1r Congrés Català d’Intel·ligència Artificial

TARRAGONA, 21 - 23 D'OCTUBRE DE 1998
Aquesta edició del bulletí recull les comunicacions que es presentaran durant el 1r Congrés Català d’Inteligència Artificial. Aquest congrés neix amb la voluntat de consolidar les dues trobades que va organitzar l’associació Catalana d’Inteligència Artificial els anys 1995 (a l’Institut d’Investigació en Intel·ligència Artificial) i 1997 (a la Universitat de Lleida).

El més de 40 treballs publicats aquí corresponen a treballs de recerca dels diferents grups d’Inteligència Artificial de les universitats i centres de recerca dels Països Catalans. El nombre de treballs rebuts i la seva qualitat són un reflex de l’activitat i la vitalitat d’aquest camp a casa nostra.

Esperem que el gran èxit de resposta que ha tingut la convocatòria del congrés permeti que els diferents grups de recerca puguin posar en comú els seus darrers resultats i intercanvi experiències. Esperem que el congrés permeti la consolidació dels resultats aconseguts i afavorixi un increment de la col·laboració entre els diferents grups ja existents.

Aquestes jornades s’han organitzat amb la col·laboració de l’associació Catalana d’Inteligència Artificial, la Universitat Rovira i Virgili, la Diputació de Tarragona, l’Escola Tècnica Superior d’Enginyeria, la Xarxa Temàtica d’Inteligència Artificial de la CIRIT, el Comissionat per a Universitats i Recerca, el Capítol Espanyol de Teoria de la Informació de l’IEEE i la GCIAT.

Per acabar, vull aprofitar aquest espai per agradir la col·laboració de tots aquells que han donat un cop de mà en la preparació del congrés. En particular, dels membres del comitè de programa, dels àrbitres addicionals, dels membres del comitè organitzador i dels dos conferenciants convidats (Nick Jennings i Bernard De Baets).

*Vicenç Torra*
## Índex

### ASPECTES FORMALS
- Flow of Information in Epistemic Contexts ......................................................... 1
  M. Álvarez, G. Núñez
- A Similarity-Based Fuzzy Modal Logic ................................................................. 8
  R. Rodríguez, L. Godo
- Interpreting a Class of Multiple-valued Datalog Languages ................................ 16
  G. Escalada-Imaz
- Propostas de modelització de primitives computacionales per a dominis discrets .......... 25
  M. T. Signes, J. M. García, J. Mora
- Defining and combining Multiple-valued Logics for Knowledge-based Systems ........ 30
  J. A. Reyes, F. Esteve, J. Puyol-Gruart
- Phase Transitions in the Regular Random 3-SAT Problem ..................................... 39
  F. Manyà, R. Béjar
- On Diagrammatic Reasoning in Category Theory .................................................. 46
  M. Schorlemmer, J. Agustí, J. Puigsegur

### SOFT COMPUTING. XARXES NEURONALS, SISTEMES DIFUSOS I ALGORISMES GENÈTICS
- Multidimensional Scaling Analysis using Neural Networks with Distance-Error Backpropagation ................................................................. 58
  L. Garrido, S. Gómez, J. Roca
- On the determination of probability density functions by using Neural Networks .......... 61
  L. Garrido, A. Jusie
- Application of Artificial Neural Networks to the Design and Implementation of Electric Offactory Systems .............................................. 68
  J. Bremsen, N. Canellas, E. Llobet, X. Vilanova, X. Correig
- A Depth Perception Algorithm based in sequential fixations of the gaze .................. 75
  J. A. Aznar, M. Moreno, J. López
- Sistemes classificadors basats en algoritmes genètics amb enfocament de Pittsburgh per problemes amb atributs de valors reals ........................................ 85
  X. Llorà, J. M. Garrell, E. Bernadó
- Una plataforma de programación genética (ModelGP) per a la construcció de models de predicción quantitativa ........................................ 94
  G. Calzada, J. M. Garrell, R. Burtí
- Fuzzy Heterogeneous Neurons for Imprecise Classification Problems ...................... 103
  L. Belanche, J. Valdés, R. Alguèzov
- Economic Series Analysis through Independent Components .................................... 111
  F. Miralles, C. Antótegui

### SISTEMES MULTI AGENTS
- A Scheme for Anonymous Electronic Commerce Using Secure Intelligent Trade Agents ................................................................. 119
  J. Domingo, J. Herrera
- Institucions Electròniques ...................................................................................... 125
  C. Sierra, P. Noriega
- Gestión de red ATM en una arquitectura multi-agente inteligente a partir de parámetros de control de congestión ......................................... 130
  A. Barba
- Distribución de la resolución de problemas al Web utilizando técnicas de satisfacción de restricciones ....................................................... 137
  M. Torrens, R. Weigel, B. Faltins
- Investigadors: un model general d'agents racionals no-ideals ................................ 145
  A. Moreno, U. Cortés, T. Sales
- Instituting Interactions Among Agents .................................................................... 154
  J. L. Arcos, E. Plaza
- JIM: A Java Interagent for Multi-Agent Systems .................................................. 163
  F. J. Martín, E. Plazo, J. A. Rodríguez, J. Sabater
- JAFDIS, a Java Framework for Dialogical Institution Specification ........................ 172
  T. Alsinet, R. Béjar, C. Antótegui, C. Fernández, F. Manyà
<table>
<thead>
<tr>
<th>Título</th>
<th>Página</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the usefulness of component-based architectures for knowledge discovery</td>
<td>177</td>
</tr>
<tr>
<td>J. Cerquides, R. López de Mántaras</td>
<td></td>
</tr>
<tr>
<td>Un modelo lingüístico para la descripción de conceptos</td>
<td>181</td>
</tr>
<tr>
<td>E. Hernández, J. Recasens</td>
<td></td>
</tr>
<tr>
<td>Representación, fusión e interpretación de atributos con técnicas fuzzy</td>
<td>185</td>
</tr>
<tr>
<td>D. Netleton</td>
<td></td>
</tr>
<tr>
<td>Generación automática de reglas a partir de la caracterización de clases</td>
<td>193</td>
</tr>
<tr>
<td>K. Gibert, U. Cortés</td>
<td></td>
</tr>
<tr>
<td>Radonament Qualitatiu</td>
<td></td>
</tr>
<tr>
<td>Applying qualitative spatial reasoning to robot navigation</td>
<td>204</td>
</tr>
<tr>
<td>T. Escrig, F. Toledo</td>
<td></td>
</tr>
<tr>
<td>A qualitative agent approach for assessing dynamic process human supervision</td>
<td>213</td>
</tr>
<tr>
<td>L. Travé-Massuyès, F. Prats, M. Sánchez, Agell, J. Pastor</td>
<td></td>
</tr>
<tr>
<td>Modal interval analysis for error-bounded semiqualitative simulation</td>
<td>223</td>
</tr>
<tr>
<td>J. Armengol, L. Travé-Massuyès, J. Veith, M. A. Saiz</td>
<td></td>
</tr>
<tr>
<td>Una generalización del modelo qualitatius d'ordres</td>
<td></td>
</tr>
<tr>
<td>de Magnitud. Aplicación a l'operació suma</td>
<td>232</td>
</tr>
<tr>
<td>A. Catalá, J. M. Moreno</td>
<td></td>
</tr>
<tr>
<td>Sistemes basats en el coneixement i llenguatge natural</td>
<td></td>
</tr>
<tr>
<td>Instance-based method for oceanographic prediction</td>
<td>237</td>
</tr>
<tr>
<td>J. M. Corchado, J. R. García, B. Lees</td>
<td></td>
</tr>
<tr>
<td>L'example distance: a new similarity measure for case retrieval</td>
<td>246</td>
</tr>
<tr>
<td>M. Sánchez, J. Roda, Q. Comas, U. Cortés, M. Poch</td>
<td></td>
</tr>
<tr>
<td>Model d'entrada natural i implementació en una base de coneixement</td>
<td>254</td>
</tr>
<tr>
<td>R. Morante, G. Vázquez, A. Fernández, I. Castellón, M. A. Marri</td>
<td></td>
</tr>
<tr>
<td>Aprenentatge</td>
<td></td>
</tr>
<tr>
<td>Reinforcement learning in constructive languages</td>
<td>264</td>
</tr>
<tr>
<td>J. Hernández</td>
<td></td>
</tr>
<tr>
<td>Efficient and comprehensible hierarchical clusterings</td>
<td>273</td>
</tr>
<tr>
<td>L. Talavera, J. Béjar</td>
<td></td>
</tr>
<tr>
<td>Dynamic local feature selection in incremental clustering</td>
<td>282</td>
</tr>
<tr>
<td>L. Talavera</td>
<td></td>
</tr>
<tr>
<td>Visió</td>
<td></td>
</tr>
<tr>
<td>Recognising natural objects in outdoor scenes: a survey</td>
<td>290</td>
</tr>
<tr>
<td>J. Freixenet, A. Cosala, J. Batlle, J. Marri</td>
<td></td>
</tr>
<tr>
<td>An active vision approach for autonomous robot navigation</td>
<td>298</td>
</tr>
<tr>
<td>A. Martínez, J. Aguerre, J. Vitrá</td>
<td></td>
</tr>
<tr>
<td>Visual ELIZA: a testbed for developing visual perception processes for conversational interfaces</td>
<td>308</td>
</tr>
<tr>
<td>J. Vitrá, J. R. Serra, E. Raso, A. Puertas</td>
<td></td>
</tr>
<tr>
<td>Mobile robots monitoring system using an agent-oriented approach</td>
<td>311</td>
</tr>
<tr>
<td>D. Ponsa, J. Vitrá</td>
<td></td>
</tr>
<tr>
<td>Robotíca</td>
<td></td>
</tr>
<tr>
<td>Reinforcement learning of sensor-based robot arm motions for multi-goal reaching tasks</td>
<td>317</td>
</tr>
<tr>
<td>P. Martín, J. Múllín</td>
<td></td>
</tr>
<tr>
<td>Estimación bayesiana de características en robots móviles mediante muestreo de la densidad a posteriori</td>
<td>326</td>
</tr>
<tr>
<td>D. Gallardo, F. Escolano, R. Rizo, O. Colomina, M. A. Cazorla</td>
<td></td>
</tr>
<tr>
<td>Programació i aprenentatge en tasques robòtiques reals de manipulació</td>
<td>335</td>
</tr>
<tr>
<td>E. Cervera, A. Pasqual del Pobí</td>
<td></td>
</tr>
<tr>
<td>Refinement of environmental maps obtained from autonomous robot exploration</td>
<td>341</td>
</tr>
<tr>
<td>M. López Sánchez, R. López de Mántaras</td>
<td></td>
</tr>
</tbody>
</table>
COMITÉ DE PROGRAMA
Núria Agell (Esade-U. Ramon Llull)
Claudí Alsina (EA-U. Politècnica de Catalunya)
Teresa Alsínet (U. de Lleida)
Alícia Casals (ESEAl-U. Politècnica de Catalunya)
Ulises Cortés (LSI-U. Politècnica de Catalunya)
Alfonso Crespo (U. Politècnica de València)
Josep Luís de la Rosa (U. de Girona)
àngel Pasqual del Poblet (U. Jaume I)
Luís Godo (I. d’Investigació en IA)
Ramon Lopez de Muntarás (I. d’Investigació en IA)
Miquel Àngel Piera (U. Autònoma de Barcelona)
Enric Plaza (I. d’Investigació en IA)
Monique Pollet (U. de Perpinyà)
David Riaño (U. Rovira i Virgili)
Miquel Sanchez (LSI-U. Politècnica de Catalunya)
Josep Maria Sopena (U. de Barcelona)
Vicente Torra (U. Rovira i Virgili)
Carme Torras (I. de Robòtica Industrial)
Louise Trave-Massuyes (LAAS-CNRS)
Jesús Vaiver (U. de les Illes Balears)
Jordi Vitià (C. de Visió per Computador)

REFEREES ADDICIONALS
Josep Domingo (URV)
Miguel Ángel García (URV)
Gonzalo Escalada (III-CSIC)
Horacio Rodriguez (LSI-UPC)
Carles Sierra (IIIA-CSIC)
Josep Aguilà (U. de Girona)
Joëlle Rey

COMITÉ ORGANITZADOR
Josep Domingo
Beatriz López
Antonio Moreno
Manel Sanromà
Aida Valls

SPONSORS
Flow of Information in Epistemic Contexts*

Matías Alvarado and Gustavo Núñez
Center of Computational Research
National Technical Institute, México.
e-mail: {matias, gustavo}@pollux.cenac.ipn.mx

Abstract

In this paper, the representation and reasoning of knowledge and belief are based on structured information over so called epistemic contexts—which are an instance of three-valued models of Kripke. The structure is generated through an iterative process, in which the possible worlds contains at least the information in the current world. Knowledge in a world in the epistemic context is every true statements in all the most informed possible worlds in the context, while beliefs does it in some of them. In order to give an adequate logical support to the process of knowledge acquisition, a generalization of the classical logic consequence is proposed. The resulting epistemic logic is complete.

1 Introduction

The formalization of the process of acquiring knowledge and believes, eminently dynamic, requires of 1) an epistemic calculus allowing gradual incorporation of knowledge, and 2) a representation frame providing, by the one hand, the application of precise axioms and inference rules on well defined information, and by the other hand, manipulation of incomplete or undefined information. Both types of information are always present in a computational or human agent.

In this paper, the representation frame proposed in order to handle mentioned process is based on the Kleene's Strong Three-valued Logic (KSL) [10], having truth values true (t), false (f) and undefined (u).

In accordance with an approach started by N. Belnap [4], t and f provide equal information, and both in turn, more than u; this fact defines an information partial order denoted ≤. Thus, is considered that true or false formulae provide more information than the undefined ones. Devoted to model the flow of information, this order between interpreted formulae was first applied by J. Van Benthem [5] and is used in the present work.

The models holding this gradual process of knowledge and beliefs acquisition are an instance of the three-valued models of Kripke [3], in which the relation of possibility is the information partial order. They are of the form $\mathcal{K} = \langle w_0, W, \leq, \nu \rangle$, where the worlds $w, w' \in W$ satisfy $w \leq w'$, if and only if the set of true or false formulae in $w$ is a subset, respectively, of the ones in $w'$, while the set of undefined formulae in $w$ is a superset of the ones in $w'$. In this case, $w'$ is more informed than $w$ and is a possible world from it. From now on these models are called Epistemic Contexts (EC).

We consider that knowledge and belief in the actual world in an EC are increased from undefined statements in this world; the possible worlds have the same or more true or false statements, and in this way, they are equal or more informed than the actual world. The most informed possible worlds in an EC are called maximals.

Whereas knowledge in a $\mathcal{K}$ is defined as the true statements in all the maximal worlds, beliefs are the true statements in at least one maximal world. Modal operators formalizing these concepts are the following: $K$ for knowledge and $B$ for belief.

The rest of the paper is organized as follows: next Section provides the elements of KSL, information order, epistemic contexts and a novel definition of logical entailment. In Section 3 knowledge and be-
Aspects of formal logic definitions are given, while in Section 4 the syntactic calculus and the logic completeness.

2 Epistemic Contexts

2.1 Kleene’s Strong Three-Valued Logic

The propositional language $\mathcal{L}$ in KSL is defined from a finite set of atom sentences $\Sigma$ together with the connectives of negation $\neg$, and disjunction $\vee$. Derived connectives from $\neg$ and $\land$ are conjunction $\land$ and implication $\rightarrow$ in the classical way [10]; all they are given in Table 1. If $\varphi$, $\psi$ are formulae in $\mathcal{L}$, then, $\neg \varphi$, $\varphi \lor \psi$, $\varphi \land \psi$ and $\varphi \rightarrow \psi$ do, and anything else is a formula in $\mathcal{L}$. Let $\mathcal{F}$ be the set of formulæ in $\mathcal{L}$.

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$\psi$</th>
<th>$\varphi \lor \psi$</th>
<th>$\varphi \land \psi$</th>
<th>$\varphi \rightarrow \psi$</th>
<th>$\neg \varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>u</td>
<td>t</td>
<td>u</td>
<td>f</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>u</td>
<td>t</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>u</td>
<td>f</td>
<td>u</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>u</td>
<td>t</td>
<td>u</td>
<td>u</td>
<td>t</td>
<td>u</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
</tbody>
</table>

A three-valued interpretation $i$, is a valuation function from $\Sigma$ to \{t, f, u\}. In accordance with the logical connectives definitions, every $i$ is extended to an interpretation $\mathcal{I}$ over the set $\mathcal{F}$. Let $\mathcal{I}$ be the set of three-valued interpretations of formulæ in $\mathcal{F}$.

From the semantical viewpoint, in KSL there are no theorems in the classical sense due to the interpretation that assigns the undefined truth-value to every formula, is not a model for any formula. Then, the set of tautologies or true formulæ for every interpretations is empty in KSL. Thus, any $\varphi \rightarrow \neg \varphi$ is not a tautology in KSL for any formula $\varphi$, and $I \models \varphi \lor \neg \varphi$ does not happen. This implies that the deduction theorem,

$$A_1, \ldots, A_n \models \varphi$$

if and only if

$$\models \varphi \rightarrow \ldots \rightarrow \varphi \rightarrow \varphi$$

is not valid in KSL. Further details are given in [3]. This fact endorses a major difference with respect to inference in two-valued classical logic, and conveys fundamental futures for inference in KSL, as well as for rules of introduction of connectives as illustrated for the introduction of $\neg$ connective above. In the next Subsection the Epistemic Contexts are defined; then, in Subsection 2.3, the theorems for our logic are defined. They are unseparated of the knowledge and belief definitions that we will introduce.

2.2 Epistemic Contexts

Formal definition of information order over three-valued interpretations is the following: The information order $\preceq$ over the set $\{t, f, u\}$ is such that: $u \prec t$, $u \prec f$ and $t, f$ are not comparable; in addition $t \preceq t$ and $f \preceq f$. That order extension over $\mathcal{I}$ is the following: let $g \in \{t, f\}$ and $I, I' \in \mathcal{I}$, then $I \preceq I'$ if and only if

- $\{\varphi \in \mathcal{F} : I(\varphi) = g\} \subseteq \{\varphi \in \mathcal{F} : I'(\varphi) = g\}$

and

- $\{\varphi \in \mathcal{F} : I'(\varphi) = u\} \supset \{\varphi \in \mathcal{F} : I(\varphi) = u\}$.

The information order is partial over $\mathcal{I}$. The epistemic use we will give this lattice structuration for interpretation in $\mathcal{I}$, is that there are alternative (branches of worlds) to be analyzed when expanding knowledge and belief. Now, definition of epistemic context is the following:

Definition 1 A three-valued model of Kripke $<W, R, \nu>$ is an epistemic context if and only if the relation of possibility among the worlds is the information order $\preceq$.

Worlds $w, w' \in W$ satisfy $w \preceq w'$, if and only if $I \preceq I'$, where $\nu(w) = I$ and $\nu(w') = I'$. It means that the set of true or false formulæ in $w$ is a subset, respectively, of the ones of $w'$, and that $w'$ is more informed than $w$. In particular, $w_0 \preceq w$ for every $w \in W$.

Figure 1 illustrates an epistemic context $\mathcal{K}$ in which $(P \lor Q) \land R$ is undefined (together with $P$, $Q$, and $R$) in $w_1$. The worlds $w_2$ and $w_3$ are more informed then $w_1$, and $w_2$ more than $w_2$. The worlds $w_2$ and $w_3$ are not comparables each other. $P \lor Q$ becomes true in $w_2$ and $w_3$, while $(P \lor Q) \land R$ remains undefined until world $w_4$. In figure 1, the true formulæ are preceded by symbol $t$, while the

$^3$Remember that $I = \{\nu(w) : \varphi \in \mathcal{F}\}$.
undefined ones not appear at all. The right formalization of this epistemic process requires a definition of logic consequence capturing this dynamic aspect.

2.3 Logic Entailment and Theorems in $\mathcal{K}$

Let $\mathcal{K}$ be an epistemic context and $\Gamma$ and $\Delta$ be sets of formulae. Now, our definitions of satisfiability and of information logic consequence (ILC) in an Epistemic Context are given.

Definition 2 A formula $\varphi$ is satisfiable in $\mathcal{K}$, whenever there is a world $w$ in $\mathcal{K}$ with $\nu(w)(\varphi) = t$. A $\Delta$ set is satisfiable in $\mathcal{K}$, whenever every formula in $\Delta$ is satisfiable in $\mathcal{K}$.

Definition 3 $\Gamma$ is an information logic consequence of $\Delta$ in $\mathcal{K}$, denoted $\Delta \models_{\mathcal{K}} \Gamma$, if and only if for every possible world being model of $\Delta$ there is a world $w'$ with $w \preceq w'$ and $\gamma \in \Gamma$, such that $\gamma$ is true in $w'$, and thus $w'$ is a model of $\gamma$.

When $\Gamma = \{\gamma\}$ is a singleton, $\Delta \models_{\mathcal{K}} \{\gamma\}$ is written $\Delta \models_{\mathcal{K}} \gamma$. In this case, for every world in $\mathcal{K}$ being model of $\Delta$, there is a possible world in $\mathcal{K}$, being a more informed model of $\Delta$, which is also a model for $\varphi$. For instance, in Figure 1, $\{P, P \lor Q\} \models_{\mathcal{K}} R$ and $\{P, P \lor Q\} \models_{\mathcal{K}} (P \lor Q) \land R$; furthermore, $\{P, P \lor Q\} \models_{\mathcal{K}} \{R, (P \lor Q) \land R\}$.

An important remark is that every model of $\Delta$ should have an extended model being model of $\gamma$. It ensures that every maximal model of $\Delta$ becomes a model of $\gamma$. Thus, in our Information Epistemic Logic, the worlds with maximal information play a fundamental role for logic consequence.

Classically, a formula $\varphi$ is a logical consequence from $\Delta$ set if and only if every model of $\Delta$ is a model of $\gamma$. This is a particular instance of the information logic consequence, because this last applies over maximal models instead of on every model. Thus, ILC generalizes the classical logic consequence.

Using ILC, the reasoning for an agent can be formalized in such a way that the acquisition of information, as a dynamic and incremental process, is well captured. A further generalization supports the formalization for the reasoning of a set of agents [3].

Although in KSL there are no theorems in the classical sense, in the epistemic logic here proposed, theorems are defined as relatives to the context. The formulae satisfiable in every maximal $w'$ in $\mathcal{K}$ is a theorem in this context. These formulae are the information logic consequence from the set of true formulae in the initial world, whenever this set is not empty.

The fact that the logic consequence and theorems are defined over the true formulae in the maximal worlds, allows the semantical handling of information that:

- Could be initially incomplete.
- Is gradually extended in a EC.
- The semantic characterization is done having maximal information at any step.

The ILC generalizes our Information Epistemic Logic semantics. Syntactically, it is developed in Section 4 resulting the formal system $\mathcal{S}_E$. In a converse way, the semantic counterpart of $\mathcal{S}_E$ is the Information Logic Consequence. In the next section definitions of knowledge and belief are given: the theorems in $\mathcal{K}$ are the knowledge statements there.

3 Knowledge and Belief

The $true$ statements in all the maximal worlds in $\mathcal{K}$—which is a global condition— are considered the knowledge statements into this context $\mathcal{K}$. They are the logic consequence of the $true$ formulae in world $w_0$.

Notice that this perspective allows to distinguish between a priori (initial) and a posteriori (latter) knowledge: Every true statement in all the maximal worlds is knowledge. However, a priori knowledge are the $true$ formulae in the current world—and thus in every world in the context; by the other...
hand, a posteriori knowledge are the undefined statements in the current world though becoming true in every maximal world. In the model $K$ of Figure 1, for instance, $R$ is a priori knowledge in $K$, while $P \lor Q$ is latter knowledge, because being undefined in the current world is true in every maximal world in $K$ (see example in Section 4).

By the other hand, true statements in at least a world are considered beliefs in $K$. The underlying intuition is that true statements in at least a part of the EC are beliefs because they are supported, at least, by particular information in $K$. Formal definitions are the following.

**Definition 4** $\varphi$ is knowledge in $K$, if and only if $\varphi$ is true in every maximal world $w$ in $K$.

**Definition 5** $\varphi$ is belief in $K$ if and only if $\varphi$ is true in a world $w' \in K$.

This definition establishes a basic difference with respect to the usual definition of knowledge, [9], [8], [13] for instance. On the other hand, knowledge, either a priori or a posteriori, is belief. Furthermore, beliefs not being knowledge in $K$, are true statements in a proper submodel of $K$, and can be considered local knowledge in $K$. For instance, in Figure 1, $(P \lor Q) \land R$ is belief (but not knowledge) in $K$, as well as $P$, $Q$ and $R$. Intuitively, the knowledge (of an individual or group) is determined by the context in which it is circumscribed. In a wider context $K'$, in general there is further knowledge. However, for statements considered knowledge in a wider context, further evidences supporting them are required.

It is noticed that in our approach, knowledge, either a priori or a posteriori, is logic consequence of a priori knowledge. But this is not the case of belief not being knowledge.

4 Derivation

In last section the semantic aspect of logic has been established. Now, the syntactic counterpart is due. The goal is to obtain a syntactic system such that every derived statement from a set of premises be an information logic consequence from this set.

Resulting system, denoted $S_K$ for epistemic system, is in Gentzen's sequent style which given a set of formulas, some formula in another set is deduced. That future is welcome here by allowing derivation between sets of formulas, which is useful concerning reasoning about knowledge and belief.

<table>
<thead>
<tr>
<th>Table 2: Rules of Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structural Rules</strong></td>
</tr>
<tr>
<td>$\Delta \Rightarrow \Gamma$ then $\Delta \neq \emptyset$ Start</td>
</tr>
<tr>
<td>$\Delta, \Gamma \Rightarrow \Delta', \Gamma'$ Cut</td>
</tr>
<tr>
<td><strong>Rules of Introduction of Connectives</strong></td>
</tr>
<tr>
<td>Right</td>
</tr>
<tr>
<td>$\Delta, \varphi \Rightarrow \Delta'$ $\Delta \Rightarrow \neg \varphi, \Delta'$ $\Delta, \neg \varphi \Rightarrow \Delta'$</td>
</tr>
<tr>
<td>$\Delta \Rightarrow \varphi, \Delta'$ $\Delta \Rightarrow \varphi \land \psi, \Delta'$ $\Delta, \psi \land \psi \Rightarrow \Delta'$</td>
</tr>
<tr>
<td>$\Delta \Rightarrow \varphi \lor \psi, \Delta'$ $\Delta \Rightarrow \varphi \rightarrow \psi, \Delta'$ $\Delta \Rightarrow \psi \rightarrow \Delta'$</td>
</tr>
<tr>
<td>$\Delta \Rightarrow \psi, \Delta'$ $\Delta \Rightarrow \varphi \lor \psi, \Delta'$ $\Delta, \varphi \lor \psi \Rightarrow \Delta'$</td>
</tr>
</tbody>
</table>

$\Delta \Rightarrow \psi, \Delta'$ $\Delta, \psi \Rightarrow \Delta'$ $\Delta \Rightarrow \psi, \Delta'$ $\Delta \Rightarrow \psi \rightarrow \Delta'$
Aspects of Formals

Let \( \varphi, \psi \) be formulae, and \( \Delta, \Gamma, \Delta', \Gamma' \) be finite sets of formulae in language \( L_\varphi \) used in these tables. There are structural rules, rules of introduction of connective and rules of inference in the system. Structural rules change the structure of the sequent; this is the type of \textit{Start} and \textit{Cut} in Table 2. Rules of introduction of connects and rules of inference in Table 2, while rules of inference establish the way to obtain further formulae in \( S \) (Table 3). The \textit{Start} rule establish that whenever there is an inference the set of premises is nonempty. That is due to there is no theorems in KSL. By the other hand, the \textit{Cut} rule provides proofs simplification.

In order to make the syntactical translation of worlds in a context, let \( W \) be a set of index, \( \leq \) a partial order over \( W \) and \( K = \{ w_0, W, \leq, \nu \} \) a subset of partially ordered index. Here \( W \) and \( K \) are sets of symbols, respectively identified with the semantical worlds and epistemic contexts. From now on we use this term to refer them syntactically as well.

For formulae in which is relevant the context and worlds, they appear explicitly in the formulae and are written in the form

\[(K : w : \varphi)\]

with \( w \) a world in \( K \), \( \varphi \) an ordinary formula and \( K \) is a partial ordered set of worlda (see Table 3); that is the semantical translation that \( \varphi \) is \textit{true} (\( \neg \varphi \) is \textit{false} ) in the world \( w \) in \( K \). In addition for sets of formulae we have that, \( (K : w : \Delta) = \{(K : w : \varphi) ; \varphi \in \Delta\}, (K : w : K(\Delta)) = \{(K : w : K \varphi) ; \varphi \in \Delta\} \) and \( (K : w : B(\Delta)) = \{(K : w : B \varphi) ; \varphi \in \Delta\} \).

Whenever context and worlds are not explicit in a formula, it means that they apply regardless them. That happen with rules, either structural or for right and left introduction of connectives.

### 4.1 Syntactical Characterization of K and B

Now, the syntactical translation of semantical definition of formula of knowledge is the following:

\[
\begin{align*}
\circ & \quad (K : w : K \varphi) \iff (K : w' : \varphi) \quad \text{for every maximal } w', \\
\bullet & \quad (K : w : \neg K \varphi) \iff (K : w' : \neg \varphi) \quad \text{for some maximal } w'.
\end{align*}
\]

In addition, whenever \( \varphi \) is \( u \) for some maximal \( w' \in K \) and is not false in no \( w' \), then \( K \varphi \) is \( u \) in \( w \in K \).

Notice that \( (K : w : \varphi) \Rightarrow (K : w : K \varphi) \), but not the converse: whenever \( \varphi \) is \( u \) in \( w \in K \) but \( (K : w' : \varphi) \) in every maximal \( w' \) with \( w \leq w' \), for definition \( (K : w : K \varphi) \).

In a similar manner, syntactical definitions for \( t \) and \( f \) belief are given:

\[
\begin{align*}
\circ & \quad (K : w : B \varphi) \iff (K : w' : \varphi) \quad \text{for a } w \leq w', \\
\bullet & \quad (K : w : \neg B \varphi) \iff (K : w' : \neg \varphi) \quad \text{for every maximal } w'.
\end{align*}
\]

In addition, whenever \( \varphi \) is \( u \) for every maximal in \( w' \in K \), then \( B \varphi \) is \( u \) in \( w \in K \). Could be that \( (K : w : B \varphi \land B \neg \varphi) \), which follows from definition (3), but \( (K : w : \neg (B \varphi \land B \neg \varphi)) \) in accordance with definition (1). By the other hand, knowledge is belief, \( (K : w : K \varphi) \) implies \( (K : w : B \varphi) \), which is established by the rule \( KB \). Concerning the heredity rule \( (He) \), it means that the formulae in a world are preserved in every further (with respect to the order \( \leq \) ) world. The rule \( Th \) is an instance of \( He \), making reference to maximal worlds. The rule of \textit{modus ponens} is the classical one.

### Table 3: Rules of Inference

<table>
<thead>
<tr>
<th>Rule</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>( He : )</td>
<td>( (K : w : \Delta) \Rightarrow (K : w' : \Delta) ) with ( w \leq w' )</td>
</tr>
<tr>
<td>( Th : )</td>
<td>( (K : w : \Delta) \Rightarrow (K : w' : \Delta) ) for maximal ( w' )</td>
</tr>
<tr>
<td>( KB : )</td>
<td>( (K : w : K(\Delta)) \Rightarrow (K : w' : B(\Delta)) )</td>
</tr>
<tr>
<td>( ModusPonens : )</td>
<td>( \varphi \land (\varphi \rightarrow \psi) \rightarrow \psi )</td>
</tr>
</tbody>
</table>

In addition to axioms and rules, \textit{provability} establishes the way in which a formula or a set of formulae is derived through the rules of inference in the formal system, \( S_\varphi \) in our logic.

**Definition 6** Formula \( (K : w : \varphi) \) is provable in the system \( S_\varphi \) if and only if there is a sequence of formulae \( (K : w_1 : \varphi_1), \ldots, (K : w_n : \varphi_n) \) in such a way that \( (K : w_0 : \varphi_0) = (K : w : \varphi) \) and \( (K : w_i : \varphi_i) \) is a premise or is derived by rules of inference in \( S_\varphi \), with \( 1 \leq i \leq n \). That is denoted...
Aspects formals

\[( \mathcal{K} : w_1 : \varphi_1 ), \ldots, ( \mathcal{K} : w_n : \varphi_{n-1} ) \vdash_{\mathcal{S}_\mathcal{E}} ( \mathcal{K} : w : \varphi ) \].

A set \(( \mathcal{K} : w : \Gamma )\) is provable in \( \mathcal{S}_\mathcal{E} \), if and only if every formula \(( \mathcal{K} : w : \varphi )\) with \( \varphi \in \Gamma \) is provable.

Notice that not necessarily \( w_i \leq w \); knowledge formulae \( K\varphi \) are satisfied in \( w_0 \) for instance. On the other hand, the provability of a set should not be confused with the notion of sequent. In the Gentzen Sequent Calculus, given two sets of formulae \( G \) and \( H \), the sequent \( G \Rightarrow H \) is a formula such that \( g_1 \land \ldots \land g_l \rightarrow h_1 \lor \ldots \lor h_m \), with \( g_i \in G \) for \( i \in \{1, \ldots, l\} \), and \( h_j \in H \) for \( j \in \{1, \ldots, m\} \). The next example is devoted to clarify the provability definition.

Example

In the following example, context \( \mathcal{K} \) is omitted; uppercases are the first letters, respectively of predicates Married, Couple, Abnormal, Responsibility and Work, while Co denotes Cohabitation. Using them, the following situation is represented: persons married or in cohabitation alive in a couple; if each in the couple is not abnormal, sons procreation is possible. A responsible couple be parents need to work, while no having sons could not work both ever.

\[( w_1 : M \lor Co \rightarrow C) \]
\[( w_2 : M), (w_2 : \neg A), (w_3 : C \land \neg A \rightarrow (S \lor \neg S)) \]
\[( w_2 : M \lor Co), (w_1 : M \lor Co \rightarrow C) \vdash_{\mathcal{S}_\mathcal{E}} (w_2 : C) \]
\[( w_3 : Co), (w_3 : \neg A), (w_3 : C \land \neg A \rightarrow (S \lor \neg S)) \]
\[( w_3 : Co), \vdash_{\mathcal{S}_\mathcal{E}} (w_3 : M \lor Co), (w_1 : M \lor Co \\
\rightarrow C) \]
\[( w_4 : A) \]
\[( w_5 : R), (w_5 : S \land R \rightarrow W), (w_5 : C \land \neg A \rightarrow S) \]
\[( w_2 : C), (w_2 : \neg A), (w_5 : C \land \neg A \rightarrow S) \vdash_{\mathcal{S}_\mathcal{E}} (w_5 : C) \]
\[( w_5 : R), (w_5 : S \land R \rightarrow W) \vdash_{\mathcal{S}_\mathcal{E}} (w_5 : W) \]
\[( w_6 : C \land (A \lor \neg A) \rightarrow \neg S), (w_6 : S \land R \rightarrow \neg W) \]
\[( w_4 : A), (w_3 : C), (w_6 : (C \land ((A \lor \neg A) \rightarrow \neg S)) \vdash_{\mathcal{S}_\mathcal{E}} (w_6 : \neg S) \]
\[( w_6 : \neg S), (w_6 : \neg S \rightarrow \neg W) \vdash_{\mathcal{S}_\mathcal{E}} (w_6 : \neg W) \]

In accordance with heredity rule, every statement in a world \( w \) happens in \( w' \) with \( w \preceq w' \). For instance, for derivation in world \( w_6 \), the formulas in \( w_3 \) and \( w_4 \) occur in \( w_6 \) too.

In the example, initial world is \( w_1 \) having \( w_2 \), \( w_3 \) and \( w_4 \) as direct successor (ds); in turns, \( w_5 \) and \( w_6 \) are ds of \( w_5 \), and \( w_3 \), while \( w_5 \) of \( w_4 \) alone. We have the following knowledge and beliefs for an agent in world \( w_0 \) of epistemic context \( \mathcal{K} \): \(( \mathcal{K} : w_1 : K(Married \lor Cohabitation)) \), \(( \mathcal{K} : w_1 : K(Sons \lor \neg Sons)) \), \(( \mathcal{K} : w_1 : B(Couple \land Sons \land Responsibility \rightarrow Work)) \), \(( \mathcal{K} : w_1 : B(Couple \land \neg Sons \rightarrow \neg Work)) \), and \(( \mathcal{K} : w_1 : B(Couple \land Abnormal \rightarrow \neg Sons)) \).

Relation between \( K \) and \( B \)

In most of the knowledge and beliefs modal logics, the operators of knowledge \( K \) and belief \( B \) fit that \( K\varphi \equiv \neg B\neg \varphi \). For our Information Epistemic Logic this fact is not an exception and \( K\varphi \equiv \neg B\neg \varphi \) means that for every knowledge formula \( \varphi \), its negation is not a belief. It is observed that it applies on every world in a context, and this is the reason to does not write explicitly the world and the context. The proof is direct: \( K\varphi \) allows \( \varphi \) in every maximal \( w \in \mathcal{K} \), which implies that \( \neg \varphi \) is not in any \( w' \), namely \( \neg B\neg \varphi \).

On the other hand, the axioms of Positive Introspection \( K\varphi \rightarrow K(K\varphi) \) is fulfilled in IEL, which follows directly from the knowledge definition. But the one of Negative Introspection \( \neg K\varphi \not\rightarrow (K\neg K\varphi) \) does not fulfill: \(( \mathcal{K} : w_0 : \neg K\varphi ) \) implies \(( \mathcal{K} : w : \neg \varphi ) \) for some \( w' \) (see the syntactical definition of knowledge), but it does not implies \(( \mathcal{K} : w_0 : K\neg K\varphi ) \). Notice that the world \( w_0 \) can be replaced for any other satisfying the condition.

4.2 Completeness in formal system \( \mathcal{S} \)

Theorem 1 \( \Delta \vdash_{\mathcal{S}_\mathcal{E}} \Gamma \) if and only if \( \Delta \vdash_{\mathcal{E}} \Gamma \).

Let \( \mathcal{S}_\mathcal{E} \) be the formal system given by the rules in Tables 2 and 3, \( \varphi \) a formula, \( \Delta \) and \( \Gamma \) be sets of formulae in \( \mathcal{L}_\mathcal{E} \) language, which is the language \( \mathcal{L} \) extended in the usual manner with operators \( K \) and \( B \). In a rough way, the completeness proof follows the next steps:

- Given a \(( \mathcal{K} : w_0 : \Delta )\) consistent set, there is a \(( \mathcal{K} : w_0 : \Sigma )\) saturated theory being extension of \(( \mathcal{K} : w_0 : \Delta )\), such that whenever \( \varphi \not\in \Delta \), then \( \varphi \not\in \Sigma \).

- There is an epistemic context \( T = w_0, W_i \leq _\mu > \) such that:
Every $z \in Z$, with $Z \subseteq W$ the most informed models (worlds) of $\Delta$ in $K$, is a model of $\Sigma$ satisfying that:

$$\mu(z)(\sigma) = t \iff \text{if and only if } \sigma \in \Sigma.$$  

$$\mu(z)(\sigma) = f \iff \text{if and only if } \neg \sigma \in \Sigma.$$  

- $\Sigma \models \varphi$ if and only if $\varphi \in \Sigma$.

Thus, every maximal world $z \in T$ is a model of $(K : w_0 : \Sigma)$. In this manner, context $T$ is an epistemic context having saturated and consistent theories $z$, being models for $(K : w_0 : \Sigma)$; these $z$ can be considered the canonical models for $(K : w_0 : \Sigma)$. And conversely, every formula satisfied in every $z$ is in $(K : w_0 : \Sigma)$. Thus, completeness follows. The detailed proof of completeness is available in an extended version of this paper.

4.3 Related Works

Elias Thijssen [13] has proposed a close framework of representation and reasoning for knowledge and belief based on Partial Logic (PL). A rough compariation between Partial and Kleene's Strong Logic is the following: in PL there are truth-valued formulae, as true or false, and the rest are not truth-valued. In KSL, true and false formulae in PL are equally valuated, while the non-truth-valued ones have undefined value. This fact determines essential differences between both approaches [3].

5 Concluding Remarks

Computational (and human) agents handle information that is organized and structured. In order to acquire knowledge and beliefs from a given information, previous organization of this information is required. This is a dynamic process, having several alternatives to explore. In this paper, the so called epistemic contexts are defined as flexible frames in order to deal with this process and, additionally, to model knowledge and beliefs expansion. The undefined truth value from Kleene's Strong Logic is used to deal with undefined or incomplete information over an epistemic context $K$. Through a gradual process, that information can become true or false. Initial true information in $K$ is apriori knowledge. Latter true information in $K$ can be a posteriori knowledge or belief. Providing formal support to this process, a new logic consequence definition, that generalize the classical, is proposed.

References


A Similarity-Based Fuzzy Modal Logic

Lluís Godo
Institut d’Investigació en Intel·ligència Artificial (IIIA) - CSIC
Campus Univ. Autònoma Barcelona
08193 Bellaterra, Spain
godo@iiia.csic.es

Ricardo O. Rodríguez
Dpto. de Computación
Fac. Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad de Buenos Aires, Argentina
ricardo@dc.uba.ar

Abstract

In this paper we are concerned with the formalization of a similarity-based type of reasoning dealing with expressions of the form approximately \( \varphi \), where \( \varphi \) is a fuzzy proposition. From a technical point of view we need a fuzzy logic as base logic to deal with the fuzziness of propositions and also we need a modality to account for the notion of approximation or closeness. Therefore we propose a modal fuzzy logic with semantics based on Kripke structures where the accessibility relations are fuzzy similarity relations measuring how similar are the possible worlds. We provide completeness results.

Keywords: similarity-based reasoning, fuzzy logic, modal logic

1 Introduction

In our opinion, approximate reasoning is one of the most fascinating branches of Artificial Intelligence, and it has generated an extensive literature. One of the aims of approximate reasoning is to provide reasoning models being more flexible than the one of classical logic. In that sense, vagueness and uncertainty are two of the main notions that usually appear in a general context of approximate reasoning. But, following [6] we make a sharp distinction between them. We shall associate the term uncertainty to a degree of belief regarding a proposition, usually crisp, and thus it can only be true or false; on the other hand, vagueness (fuzziness) is associated to a degree of truth of a proposition which may be fuzzy and thus admitting non-extremal degrees of truth. Both, truth degrees of fuzzy propositions and belief degrees of crisp propositions, are usually coded by reals from the unit interval \([0, 1]\) but they are handled in a very different way. Uncertainty, as belief, is inherently intensional (non-truth functional), that is, the belief degree of a compound proposition cannot be, in general, computed only from the belief degrees of their components (e.g. probability). In contrast, nothing prevents to truth degrees to be fully functional. From a logical point of view, systems corresponding to uncertainty are related to various generalizations of modal logics, whereas most of the logical systems corresponding to fuzziness are many-valued logics.

One type of approximate reasoning model is the so-called similarity-based reasoning. It aims at modelling notions of resemblance or proximity between propositions and at studying which kind of logical consequence relations make sense in such a setting. The kind of statements which are in the scope of similarity-based reasoning are of the form "if \( p \) is true then \( q \) is close to be true", in the sense that, although \( q \) may be false, knowing that \( p \) is true leads to that \( q \) is semantically close or similar to some other proposition which is true. Notice that the fact of \( q \) being close to be (or approximately) true has nothing to do with a problem of uncertainty, i.e. with a problem of missing information that is not allowing us to know whether \( q \) is true or false [2]. The matter here is to take into account that some situations resemble more than others. This leads to deal with degrees of truth (the closer \( q \) is to truth, the more \( q \) is true) rather than to degrees of uncertainty. But, unlike to most systems of many-valued logic, this notion of graded truth is not compositional since it is closely related to a logical modality. Actually, in a previous work [3] they have investigated systems of graded modal
logic related to similarity-based reasoning on crisp propositions.

In this paper we follow this modal approach to similarity-based reasoning but now we move from crisp to fuzzy propositions, that is, for instance, we shall now investigate what does it mean that a fuzzy proposition is close to be true. Technically, to do so we need to combine many-valued logic (to model fuzziness) and modal logic (to model similarity). Actually we shall define a modal logic over the Rational Pavelka logic (RPL) which is an extension of the propositional infinitely-valued Łukasiewicz’s logic with truth-constants. This will result on a many-valued modal system, a many-valued counterpart of the classical S5 modal system, with many-valued similarity-based Kripke model semantics.

A many-valued similarity-based Kripke model is a structure \( (W, S, e) \), in which \( W \) is a set of possible worlds, \( e \) represents an evaluation assigning to each atomic formula \( \varphi \) and each interpretation \( w \in W \) a truth value \( e(\varphi, w) \in [0, 1] \) and \( S \) is a fuzzy similarity relation on \( W \), i.e., a function \( S : W \times W \to [0, 1] \) satisfying the following properties: reflexivity \( S(w, w) = 1 \), symmetry \( S(w, w') = S(w', w) \) and t-norm-transitivity \( S(w, w') \otimes S(w', w'') \leq S(w, w'') \). A t-norm \( \otimes \) is a nondecreasing, associative and commutative function over the unit interval with identity 1 and absorbing element 0. \( S \) captures a notion of semantical proximity (or indistinguishability) between possible worlds, with value 1 corresponding to the identity of possible worlds and value 0 indicating that knowledge about one world does not provide any indication about propositions that are true in the other. With this semantics we try to cover inference patterns such as:

- From “if \( A \) then \( B \)”, and \( A' \), then it is plausible, at some extent, to conclude \( B' \) whenever \( B' \) is at least as close to \( B \) as \( A' \) is close to \( A \),

where \( A, B, A' \) and \( B' \) are fuzzy propositions. We measure the closeness between two fuzzy propositions by considering the similarity of their (many-valued) models. We provide soundness and completeness results for a modal system with respect to some classes of the above structures.

The paper is organized as follows. After this introduction we survey in Section 2 the Rational Pavelka Logic (RPL)—a generalization of the infinitely-valued Łukasiewicz’s logic discovered by Pavelka and simplified by Hájek. We will use RPL as the base fuzzy logic to model and reason about fuzzy propositions. In Sections 3 and 4 we present our similarity-based fuzzy modal system \( SLMV \), as an extension of RPL, together with its model theory. Finally in Section 5 we provide completeness results.

2 Rational Pavelka Logic

In this section we just present the main notions and properties of the infinitely-valued Łukasiewicz’s logic and its extension, Rational Pavelka Logic RPL, with rational truth-constants. A full description of these logics can be found in [6].

Łukasiewicz’s logic formulas are built from propositional variables \( p_1, p_2, \ldots \) and connectives \( \to \) and \( \neg \). Other connectives are defined from these ones. In particular

- \( \varphi \land \psi \) stands for \( \neg(\neg \varphi \land \neg \psi) \)
- \( \varphi \lor \psi \) stands for \( \neg(\neg \varphi \lor \neg \psi) \)
- \( \varphi \leftrightarrow \psi \) stands for \( (\varphi \to \psi) \land (\psi \to \varphi) \)
- \( (\varphi \to \psi) \)

An evaluation of atoms is a mapping of atomic propositions into \([0, 1]\). Such mappings can be extended uniquely to an evaluation of all formulas by putting

\[
e(\varphi \to \psi) = e(\varphi) \otimes e(\psi)
\]
\[
e(\neg \varphi) = 1 - e(\varphi)
\]

where \( x \otimes y = \min(1, 1 - x + y) \) is the well-known Łukasiewicz implication. The resulting truth functions for the derived connectives are the following:

\[
e(\varphi \land \psi, w) = \max(0, e(\varphi, w) + e(\psi, w) - 1).
\]
\[
e(\varphi \lor \psi, w) = \max(e(\varphi, w) + e(\psi, w), 1).
\]
\[
e(\varphi \land \psi, w) = \max(e(\varphi, w), e(\psi, w)).
\]
\[
e(\varphi \lor \psi, w) = \max(1 - e(\varphi, w) + e(\psi, w), 1 - e(\psi, w) + e(\varphi, w)).
\]

The following are the axioms of the Łukasiewicz’s logic L:

\[ L1: \varphi \to (\psi \to \varphi). \]
\[ L2: (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi)). \]
\[ L3: (\neg \varphi \to \neg \psi) \to (\psi \to \varphi). \]
\[ L4: ((\varphi \to \psi) \to \psi) \to ((\psi \to \varphi) \to \varphi). \]
Aspectes formals

The deduction rule of $L$ is modus ponens.

Łukasiewicz's logic satisfies an standard completeness theorem, i.e. $L \vdash \varphi$ iff $\varphi$ is a tautology over the standard MV-algebra, i.e. the real interval $[0, 1]$ with Łukasiewicz's operations as truth functions.

One inconvenience of Łukasiewicz's logic is the fact that the usual strong completeness of theories fails in this logic. Another is that Łukasiewicz's infinitely-valued logic only allows us to prove 1-tautologies, but in fuzzy logic we are interested in inference from partially true assumptions, admitting that the conclusion will also be partially true. Rational Pavelka's Logic $RPL$ is an extension of Łukasiewicz's infinitely-valued logic admitting truth constants $\overline{r}$ for each rational $r \in [0, 1]$ in the language and adding the following two bookkeeping axioms for truth constants:

\begin{align*}
r1: & \quad \neg(\overline{r}) \equiv (1 - \overline{r}). \\
r2: & \quad \overline{r} \to \overline{s} \equiv \overline{r \circ \rightarrow s}.
\end{align*}

An evaluation $e$ of propositional variables by reals from $[0, 1]$ extends to an evaluation of all formulas as in Łukasiewicz logic provided that $e(\overline{r}) = r$ for each rational $r$.

The completeness of Łukasiewicz logic extends to $RPL$ but in this case a theorem of strong completeness can be obtained. In what follows, a theory $T$ is just a set of formulas of $RPL$. An evaluation $e$ is a model of a theory $T$ if $e(\varphi) = 1$ for all $\varphi \in T$. Two principal notions will be introduced now.

**Definition 1** Let $T$ be a theory and define the truth degree of a formula $\varphi$ in $T$ as $||\varphi||_T = \inf\{e(\varphi) | e$ is a model of $T\}$, and the provability degree of $\varphi$ over $T$ as $|\varphi|_T = \sup\{r | T \vdash \overline{r} \to \overline{\varphi}\}$.

Then the completeness of $RPL$ says ([6]) that the provability degree of $\varphi$ in $T$ is just equal to the truth degree of $\varphi$ over $T$, this is, $|\varphi|_T = |\varphi|_T$.

The predicate counterpart $\forall RPL$ of the rational Pavelka logic can be developed by extending $RPL$ with the following axioms on quantifiers:

\begin{align*}
\forall 1: & \quad \forall x \varphi(x) \to \varphi(t) \quad (t$ substitutable for $x$ in $\varphi(x)) \\
\forall 2: & \quad \forall x(\nu \to \varphi) \to (\nu \to \forall x \varphi) \quad (x$ not free in $\nu$).
\end{align*}

Unlike Łukasiewicz predicate logic, a proof of strong completeness for $\forall RPL$ can be given in the same sense of above. We will also need to introduce the (crisp) equality predicate in $\forall RPL$.

Following [6] to have a theory with equality we need to introduce the following axioms:

\begin{align*}
Eq-1: & \quad (\forall x)(x = x) \\
Eq-2: & \quad (\forall x, y)((x = y) \to (y = x)) \\
Eq-3: & \quad (\forall x, y, z)((x = y) \& (y = z) \to (x = z)) \\
Eq-4: & \quad (x = y) \to (P(x, \ldots, y, \ldots) \equiv P(y, \ldots, x, \ldots)) \\
Eq-5: & \quad (\forall x, y)((x = y) \lor \neg(x = y))
\end{align*}

Eq-1, Eq-2 Eq-3 and Eq-4 correspond to usual axioms for fuzzy equality. Namely, the first three axioms correspond to the reflexivity, symmetry and transitivity of the equality predicate. The fourth is a congruence axiom of $=$ with respect to any predicate $P$. Finally, the fifth axiom ensures that the equality predicate $=$ is crisp, i.e. it can be only evaluated to 0 or 1.

3 The Logic SLMV

In this section we present the logic SLMV, for similarity-based Łukasiewicz modal logic, which will be an extension of the many-valued modal logic MV55 proposed by Hájek and Hájek and Halenková in [7]. The language of MV55 is that of RPL logic plus the modality $\Box$ ($\Diamond$ is definable as $\neg \Box \neg$). MV55 axioms are those of RPL plus the following ones:

**Axioms of S5**

\begin{align*}
\Box(\varphi \to \psi) & \to (\Box \varphi \to \Box \psi). \\
\Box \varphi & \to \varphi. \\
\varphi & \to \Box \Diamond \varphi. \\
\Box \varphi & \to \Box \Box \varphi.
\end{align*}

**Rational axioms**

\begin{align*}
\overline{r} & \equiv \Box \overline{r}.
\end{align*}

\begin{align*}
\Box(\varphi \to \psi) & \equiv (\overline{r} \to \Box \overline{r}). \\
\Diamond(\varphi \to \psi) & \equiv (\overline{r} \to \Diamond \overline{r}).
\end{align*}

**Last axiom**

\begin{align*}
\Diamond(\varphi \& \varphi) & \equiv (\Diamond \varphi \& \Diamond \varphi)
\end{align*}

and as deduction rules: Modus ponens and necessitation:

**RN:** From $\varphi$ infer $\Box \varphi$.

Now we extend the language of MV55 by including two new modal operators: $\nabla$ and $\Delta$ ($\Delta$ being definable as $\neg \nabla \neg$). We denote this language as $L$. Now, we present our SLMV modal logic.

**Definition 2** The similarity-based Łukasiewicz's many-valued modal logic SLMV is the extension of MV55 by adding the following axioms:
Aspectes formals

K: \( \nabla(\varphi \rightarrow \psi) \rightarrow (\nabla \varphi \rightarrow \nabla \psi) \).

Rel: \( \Box \varphi \rightarrow \nabla \varphi \)

T: \( \nabla \varphi \rightarrow \varphi \).

B: \( \varphi \rightarrow \nabla \Delta \varphi \).

4: \( \nabla \varphi \rightarrow \nabla \nabla \varphi \).

Equ1: \( \varphi \equiv \nabla \varphi \).

Equ2: \( \nabla(\varphi \wedge \psi) \equiv \nabla \varphi \wedge \nabla \psi \).

and the following rule:

RR: From \( \varphi \rightarrow \psi \) infer \( \nabla \varphi \rightarrow \nabla \psi \).

Note that the necessitation rule for \( \nabla \) is a particular case of RR. The notion of proof in SLMV is as usual.

Next, we define the notion of a similarity Kripke model and state the truth and validity conditions for modal sentences in a world, in a model and in a class of models.

4 Similarity Kripke Models

Intuitively, the idea is that given a non-modal fuzzy proposition \( \varphi \), the (modal) proposition \( \Delta \varphi \) is to be interpreted as \textit{approximately} \( \varphi \), in such a way that, roughly speaking, \( \Delta \varphi \) will be true at a world \( w \) if there \( \varphi \) is true in some world similar to \( w \). The following definition of similarity Kripke models formalizes this idea (see item 6 in next definition) for general \( t \)-norm-similarity relations.

Definition 3 Similarity Kripke models are structures \( \mathcal{M} = (W, S, e) \) where:

- \( W \) is a non-empty set of objects that we call worlds.
- \( S: W \times W \rightarrow [0, 1] \) is a similarity function, reflexive, symmetric and \( \otimes \)-transitive, where \( \otimes \) is a continuous \( t \)-norm.
- \( e: \mathcal{L} \times W \rightarrow [0, 1] \) is a valuation function assigning to each propositional variable \( \varphi \) in \( \mathcal{L} \) and each world \( w \) in \( W \) a truth value \( e(\varphi, w) \).

The valuation \( e \) is extended to all formulas in \( \mathcal{L} \), any world \( w \), a continuous \( t \)-norm \( \otimes \) and its respective residuum \( \rightarrow \) as follows:

1. \( e(\varphi \rightarrow \psi, w) = e(\varphi, w) \otimes e(\psi, w) \).
2. \( e(\neg \varphi, w) = e(\varphi, w) \otimes 0 \).

3. \( e(\Box \varphi, w) = \inf_{w' \in W} \{ e(\varphi, w') \} \).
4. \( e(\Diamond \varphi, w) = \sup_{w' \in W} \{ e(\varphi, w') \} \).
5. \( e(\nabla \varphi, w) = \inf_{w' \in W} \{ S(w, w') \otimes e(\varphi, w') \} \).
6. \( e(\Delta \varphi, w) = \sup_{w' \in W} \{ S(w, w') \otimes e(\varphi, w') \} \).

The usual notions of satisfiability and validity are formalized next.

Definition 4 Let \( w \) be a world in a model \( \mathcal{M} = (W, S, e) \) then: \( (\mathcal{M}, w) \models \varphi \) iff \( e(\varphi, w) = 1 \).

2. A formula \( \varphi \) is valid in a model \( \mathcal{M} \), written \( \mathcal{M} \models \varphi \), iff for every world \( w \) in \( \mathcal{M} \) it holds that \( (\mathcal{M}, w) \models \varphi \).

3. A formula \( \varphi \) is valid in a class of models \( \mathcal{C} \), written \( \models_{\mathcal{C}} \varphi \), if it is valid in every model \( \mathcal{M} \in \mathcal{C} \).

Definition 5 Given a \( t \)-norm operation \( \otimes \) on \( [0, 1] \), we define the class of structures \( \mathcal{C}_{\otimes} \) as the set of similarity structures \( \mathcal{M} = (W, S, e) \) where \( S \) is a \( \otimes \)-similarity on \( W \).

As it has already made clear, in this paper we focus our attention on a particular class of similarity models, that defined by \( \otimes \)-similarity relations, where \( \otimes \) is the Łukasiewicz's \( t \)-norm, with values on \( [0, 1] \). Now we will prove that the logic SLMV is sound with respect to the class \( \mathcal{C}_{\otimes} \) of similarity structures \( \mathcal{C}_{\otimes} \) for the Łukasiewicz's \( t \)-norm \( \otimes_{\mathcal{L}} \). But before of that let us stress that the axiom \( K \) for the modality \( \nabla \), \( \nabla(\varphi \rightarrow \psi) \rightarrow (\nabla \varphi \rightarrow \nabla \psi) \), is not valid in \( \mathcal{C}_{\otimes} \). It is not difficult to find a similarity-based Kripke model that does not validate it.

Proposition 1 The axiom schemes of Definition 2 are valid in the class \( \mathcal{C}_{\otimes} \), for the Łukasiewicz's \( t \)-norm on \( [0, 1] \). Furthermore, the rules in that definition preserve validity in \( \mathcal{C}_{\otimes} \).

Proof: We just check some of the axioms.

T: \( e(\nabla \varphi \rightarrow \varphi, w) = 1 \) if and only if \( e(\nabla \varphi, w) \leq e(\varphi, w) \), but by definition, \( e(\nabla \varphi, w) = \inf_{w' \in W} \{ S(w, w') \otimes e(\varphi, w') \} \leq S(w, w) \otimes e(\varphi, w) = e(\varphi, w) \), since by reflexivity we have \( S(w, w) = 1 \).

B: \( e(\varphi \rightarrow \nabla \Delta \varphi, w) = 1 \) if and only if \( e(\varphi, w) \leq e(\nabla \Delta \varphi, w) \). Now by using the definition we get \( e(\nabla \Delta \varphi, w) = \ldots \).
5 Completeness

We want to show that our system SLMV is complete with respect to the class of similarity Kripke structures $C^A_1$. As usual, to prove completeness it suffices to show that if a formula $\varphi$ is in $C^A_1$, valid then there is a proof of $\varphi$ in SLMV. However, the technique that we shall use here is not usual. The reason for that is because the class $C^A_1$ is not normal [\{\cal V\} does not verify axiom K] and most of the classical techniques need axiom K in some way for completing theories. Therefore we will consider a method, based on the approach of the so-called Correspondence Theory (see e.g. \cite{9,1}), that basically exploits two main ideas.

One is the possibility of looking at a propositional modal logic as a fragment of first order logic (in our case, a fragment with only one binary predicate and as many unary predicates as propositional variables of the modal logic). In this sense, one can establish a one-to-one correspondence between validity in a modal frame and validity in a subclass of first order models. Following this line, we shall show that a formula $\varphi$ in $C^A$ is valid in $C^A_1$ if and only if its transcription to VRPL is valid in the class of models that satisfy a theory $\Gamma$ containing first order expressions of all the properties required to the binary fuzzy relations $S$ in the frames of $C^A_1$.

The other idea is the existence of a correspondence between the validity of some modal formulas in a frame and the properties of the accessibility relation characterizing such frame. In particular, we shall be interested in the fact that the validity of the axioms T, B and 4 determine the reflexivity, symmetry and $\otimes$-transitivity properties, respectively, of the fuzzy accessibility relations $S$ of the frame. In that sense, we shall show that any theory, in the above mentioned fragment of VRPL, containing the translation of the axioms T, B and 4, deduces the first-order expressions corresponding to the reflexivity, symmetry and $\otimes$-transitivity of the binary predicate standing for the translation of the accessibility relation. In particular, this will be true for SLMV*}, the translation of SLMV. Finally we shall show that, from the deductive point of view, SLMV* and SLMV are equivalent.

Summarising, our aim is to prove completeness by showing that if a formula $\varphi$ is valid in the class of modal frames $C^A_1$ then, over the above theory $\Gamma$, VRPL proves $r \rightarrow \varphi^*$, for all $r < 1$. Then we will show that VRPL will still prove $r \rightarrow \varphi^*$, for all $r < 1$ over the theory containing the translations.
of axioms T, B and 4. In particular, from SLMV*, and finally, this will be equivalent to the fact that \( \varphi \) be deduced from SLMV.

In the rest of the section we present an sketch of the formalization of the above ideas (also somehow present in [6]), but still with some possible gaps that we will try to fill in future versions of this work. We start by establishing the correspondence between our modal logic and (a fragment of) the many-valued predicate calculi VRPL.

Let \( \mathcal{L}_f^V \) be the modal propositional language corresponding to SLMV built from a set of propositional variables \( V = \{p_1, p_2, \ldots \} \) and rational truth-constants. We build a corresponding VRPL predicate language \( \mathcal{L}_f^{V} \) from rational truth-constants, variables, a unique binary predicate \( R \) and a set of unary predicates \( \{P_1, P_2, \ldots \} \), one \( P_i \) for each propositional variable \( p_i \) in \( V \). Then we can define a translation between formulas of the modal language to first order formulas of \( \mathcal{L}_f^V \).

**Definition 7** Given an object variable \( x \), we define the mapping \( \ast \) from \( \mathcal{L}_f^V \) to \( \mathcal{L}_f^{V} \), assigning to each modal formula \( \varphi \) a first order formula \( \varphi^*(x) \) as follows:

\[ p_i^*(x) = P_i(x). \]

\[ \forall^* = \forall. \]

\( \ast \) commutes with connectives (i.e. \( (\varphi \rightarrow \psi)^* \) is \( (\varphi^* \rightarrow \psi^*) \), \( (\varphi \& \psi)^* \) is \( (\varphi^* \& \psi^*) \), etc.).

\( (\forall \varphi)^* \) is \( \forall y(\varphi^*(x/y)) \) where the variable \( y \) is not occurring in \( \varphi^*(x) \) and \( (x/y) \) represents the replacement of all free occurrences of \( x \) by \( y \).

\( (\exists \varphi)^* \) is \( \exists y((R(x, y) \rightarrow \varphi^*(x/y))) \) where the variable \( y \) does not occur in \( \varphi^*(x) \) and \( (x/y) \) represents the replacement of all free occurrences of \( x \) by \( y \).

Note that the formulas \( (\forall \varphi)^* \) and \( (\exists \varphi)^* \) are syntactically equivalent to \( \exists y(\varphi^*(x/y)) \) and \( \exists y(R(x, y) \& \varphi^*(x/y)) \) respectively. Furthermore, if we have a formula of first order \( \varphi^* \) which is the translation of a modal formula \( \varphi \) then \( x \) is the unique free variable in \( \varphi^* \).

**Remark 1** The mapping \( \ast \) is one-to-one and the image of \( \mathcal{L}_f^V \) by \( \ast \), denoted \( (\mathcal{L}_f^V)^\ast \), is a sublanguage of \( \mathcal{L}_f^{V} \).

It is important to observe that we can view modal models \( \mathcal{M} = (W, S, e) \) over the language \( \mathcal{L}_f^V \) as first-order models of the corresponding predicate language \( \mathcal{L}_f^{V} \) in the sense of VRPL. To do so we need to say how the predicate symbol \( R \) and the unary predicate symbols are interpreted.

**Remark 2** From now on, when talking about VRPL, it will be understood over the language \( \mathcal{L}_f^{V} \).

**Definition 8** Given a similarity Kripke model \( \mathcal{M} = (W, S, e) \) we associate to it a first order model \( \mathcal{M}^* = (M, \Theta) \) where the domain is \( M = W \) and the interpretation function \( \Theta \) is defined as follows:

\[ \Theta(P_i(x))[x/w] = e(p_i, w) \]

\[ \Theta(R(x_1, x_2))[x_1/w_1, x_2/w_2] = S(w_1, w_2) \]

\[ \Theta(\varphi \& \psi) = \Theta(\varphi) \& \Theta(\psi), \text{ being } \& \text{ a logical connective. } \]

where \( x, x_1 \) and \( x_2 \) are variables and \( [x/w] \) denotes the assignment of the element of the domain \( w \) to the variable \( x \).

**Lemma 2** Under the above notation,

\[ (\mathcal{M}, w) \models \varphi \iff \mathcal{M}^* \models \varphi^*(x)[x/w] \]

Moreover, it holds that

\[ C_L \models \varphi \iff \forall \mathcal{M} \in C_L : \mathcal{M}^* \models (\forall z)\varphi^*(z) \]

**Proof:** By induction over the complexity of the formula \( \varphi \). \( \square \)

Thus we have got the first announced result. Moreover we can go in the reverse direction as well. Let us denote by \( \Gamma \) the VRPL theory consisting of the following three formulas:

\[ (\forall x)R(x, x) \]

\[ (\forall x, y)(R(x, y) \rightarrow R(y, x)) \]

\[ (\forall x, y, z)(R(x, y) \rightarrow R(y, z) \rightarrow R(z, x)) \]

Then we have the following lemma.

**Lemma 3** Let \( \mathcal{M} = (U, \Theta) \) a model of VRPL over the language \( \mathcal{L}_f^{V} \). Define the Kripke model \( \mathcal{M}^{*-1} = (U, S, e) \), where \( S(u_1, u_2) = \Theta(R(x_1, x_2))[x_1/u_1, x_2/u_2] \) and \( e(p_i, u) = \Theta(P_i(x))[x/u] \). Then \( \mathcal{M}^{*-1} \) is a similarly Kripke model iff the formulas of \( \Gamma \) are \( 1 \)-tautologies in \( \mathcal{M} \).
Notice that the class of frames $C_L$ is fully characterized only by the set of properties over the accessibility relations $S$ (reflexivity, symmetry and $\varphi$-transitivity), and that these properties are expressible by the above set $\Gamma$ of $\forall RPL$ formulas. Then the above definition and lemmas show that $\varphi$ is a tautology of $C_L$ if and only if $\varphi^*$ is true in all models of $\Gamma$ (in the sense of $\forall RPL$). Thus, by completeness of $\forall RPL$, $\varphi$ is a tautology of $C_L$ if and only if $\|\varphi\|_{\forall RPL} = 1$, that is, $\forall r < i : \Gamma \vdash_{\forall RPL} (r \rightarrow \varphi^*)$. In other words, $\|\varphi\|_{C_L} = \|\varphi^*\|_{\Gamma}$.

But this does not give us yet any axiomatization of $C_L$.

Next, we try to prove that our logic offers a such axiomatization. With this idea in mind, we define the $\forall RPL$ theory $(SLMV)^*$ consisting of the translations by the mapping * of all axioms of $SLMV$ in $\rho_C$ as the image of $SLMV$ through the defined above mapping $*$ and prove the following theorem.

**Theorem 1** Let $\varphi$ be a model formula of $SLMVL$. Then:

$$\|\varphi\|_{SLMV} = \|\varphi^*\|_{(SLMV)^*}$$

i.e. the provability degree of $\varphi$ in $SLMV$ equals the provability degree of $\varphi^*$ in $\forall RPL$ over the theory $(SLMV)^*$

**Proof:** First of all notice that the translations by * of MVSX axioms (that axioms if $(SLMV)^*$ as well) result in formula schemes about quantifiers that are either axioms or provable in $\forall RPL$. This is the case also of the axioms $K_r$, $Rel$, $Equ$1, $Equ$2. Also, the necessitation rules $RN$ and $RR$ turn out to be derived inference rules in $\forall RPL$. So basically, in $\forall RPL$, the theory $(SLMV)^*$ can be equivalently reduced to the subtheory $Sim = \{T^*, B^*, \forall^*\}$. Then the theorem is just a consequence of the isomorphism, through the mapping $*$, we can establish between proofs in $SLMV$ and in $\forall RPL$ over the theory $(SLMV)^*$.

**Corollary 1** Let $\varphi$ be a formula of $SLMVL$. Then:

$$\|\varphi\|_{SLMV} = \|\varphi^*\|_{Sim}$$

So far we have proved:

1. $\|\varphi\|_{C_L} = \|\varphi^*\|_{\Gamma}$.

2. $\|\varphi\|_{SLMV} = \|\varphi^*\|_{(SLMV)^*} = \|\varphi^*\|_{Sim}$.

Thus, our objective will be accomplished if we prove that the theory $Sim$ is equivalent to the theory $\Gamma$. But we have to be careful because the elements of $Sim$ are schemes of special axioms and not usual formulas. In this sense when we write, for instance, $(SLMV)^* \vdash_{\forall RPL} (\forall y)(P(y) \rightarrow P(x))$, we are really meaning that this is true for any instantiation of the predicate $P$, that is, something like $SLMV^* \vdash_{\forall RPL} (\forall P)((\forall y)(P(y) \rightarrow P(x)))$. Abusing the notation, we make use this below.

Next theorem provides us with final the results we were searching for.

**Theorem 2** Let $\Sigma$ a theory in $\forall RPL$ containing the crisp equality axioms, and let $R$ a binary predicate. Then:

1. $\Sigma \vdash_{\forall RPL} \forall x \forall y (R(x, y) \rightarrow (R(y, x) \rightarrow R(x, y)))$

2. $\Sigma \vdash_{\forall RPL} \forall x \forall y (R(x, y) \rightarrow (R(y, z) \rightarrow (R(z, y) \rightarrow R(x, z))))$

3. $\Sigma \vdash_{\forall RPL} \forall x \forall y (R(x, y) \rightarrow (R(y, z) \rightarrow (R(z, y) \rightarrow R(x, z))))$

**Proof**: Note that the right-to-left directions are easy. For the left-to-right cases, we prove that each property corresponds to a first order condition on $R$ by finding a suitable instantiation of the predicate $P$. Thus in the first case we make the substitution of $P(u)$ by $R(x, u)$, yielding the equivalent formula:

$$\forall y (R(x, y) \rightarrow R(x, y))$$

By the universal validity of the antecedent, the latter may be simplified to the usual statement of reflexivity.

In the second case, we substitute $P(u)$ for the equality predicate formula $x = u$. This results in a formula of the form:

$$\forall y ((x = x) \rightarrow (\exists z (R(y, z) \& (x = z))))$$

The antecedent is a trivial tautology so the formula can be reduced to

$$\forall y (R(x, y) \rightarrow (\exists z (R(y, z) \& (x = z))))$$

Finally, by the congruence axiom for $R$ we can prove in $\forall RPL$ that

$$\forall z, y, z ((x = z) \& R(y, z) \rightarrow R(y, x))$$
or equivalently

\[(\forall z, y)((\exists x)((x = z) \& R(y, z)) \rightarrow R(y, x))\]

and thus, substituting in the above formula we finally get

\[(\forall x, y)(R(x, y) \rightarrow R(y, x))\]

To prove the third implication, we instantiate \(P(u)\) with \(S(x, u)\). Then what we get is

\[\forall x((\forall y(R(x, y) \rightarrow R(x, t))) \rightarrow (\forall y(R(x, y) \rightarrow \forall z(R(y, z) \rightarrow R(x, z))))\]

But the left-hand side of the implication is a tautology, so it can be subsequently reduced to

\[(\forall x, y)(R(x, y) \rightarrow (\forall z(R(y, z) \rightarrow R(x, z))))\]

and finally to

\[(\forall x, y, z)(R(x, y) \& R(y, z) \rightarrow R(x, z))\]

which is what we needed.

Notice that the equality axioms are only needed for the symmetry axiom. Whether this additional axiom may cause some unexpected problem is not absolutely clear to the authors, but the same situation appears in the classical case (see for instance [9, 1]) and it does not seem to cause any further difficulty.

6 Conclusions

The study here presented is related to the investigation of fuzzy logic in the narrow sense, i.e. fuzzy logic as a formal logical calculus suitable to deal with impreciseness (vagueness). Here we have proposed a propositional modal logic over the Rational Pavelka logic with Kripke models where the accessibility relation is a fuzzy similarity relation. The aim is to model patterns of reasoning involving propositions of the type \(\varphi\), where \(\varphi\) is itself a fuzzy proposition. The logic, called \(SLMV\), extends two previous proposals: similarity-based graded modal logics over crisp propositions but with fuzzy accessibility relations \([\text{and a many-valued S5 logic over fuzzy propositions but with crisp (universal) accessibility relations \([\text{[7]}\]. It is worth noticing that already Fitting [4, 5] studied finitely many-valued modal logics. Finally, a very interesting and related work is that of Liau [8] where a number of logics are explored, in particular linking rough sets and similarity, and deserves further study to get closer links.

Acknowledgements

The authors are grateful to Carlos Areces and Verónica Becher for valuable comments on the draft. Lluís Godo has been partially supported by the Spanish CYCIT project SMASH, TIC96-1038- C04001.

References


Interpreting a Class of Multiple-valued Datalog Languages

Gonzalo Escalada-Imaz
IIIA-CSIC, Campus UAM \s/n
08193 Bellaterra, Barcelona
c-mail: gonzalo@iiia.csic.es

Abstract
Interest in logic programming languages is increasing as new applications are emerging in areas such as Deductive Databases, Theorem Proving, Inductive Logic Programming, Parallel Languages, etc. However, dealing with bi-valued logics strongly the possibility of processing real applications. Some solutions to this problem related to knowledge representation have been proposed by defining some languages connected to uncertainty logics. However, little attention has been paid to deduction (interpretation) issues. Here, we describe an interpreter for a kind of multiple-valued datalog programs. The multi-valued SLD and our interpreter are illustrated and compared analytically as well as by means of an arbitrary example. The comparison result reveals the significant computation gain of our proposed interpreter.

Keywords: Datalog, interpretation, unification, multivalued logics, search space, efficient computation.

1.- Introduction

Interest in logic programming languages is increasing as new applications are emerging in areas such as Deductive Databases [Lloyd, 87], Theorem Proving [Hähnele et al., 1994] Inductive Logic Programming [Lavrac at al., 1995], Parallel Languages [J. Chasin et al., 1994], etc. Currently, Logic Programming is an important tool for developing industrial softwares being employed for instance, to design prototypes.

Concerning syntactic issues, some solutions to this problem have been proposed in the literature [Baldwin, 87], [Dubois et al., 90], [Li and Liu, 90], [Martin et al., 87], [Mukaidono, 89], [Tamburrini and Termini, 92] [F. Klawonn, 1994], mainly by basing the languages on uncertainty logics such as: possibilistic, probabilistic, evidential, etc.

Nevertheless, one of the limitations to run uncertainty logic programs in real applications is the lack of approaches to perform efficient automated deduction, namely efficient interpretation. Indeed, programming interpreters are mostly based till now, on the bi-valued classical meaning.

Here, we are going to describe an interpreter for a kind of multiple-valued datalog programs. Our proposed interpreter has been designed in such a way that the computing of unifiers and the exploration of the search space are efficiently carried out. For example, concerning the latter point, instead of using a SLD schema with backtracking, e.g. [Pereira, L.M. and Porto, 1982; Pietrzykowski, T. and Matwin, S., 1982], our interpreter follows a new strategy (all-solutions) which will be described. The multi-valued SLD and our interpreter are illustrated and compared analytically as well as by means of an example. The interpreter to be presented later notably improves the algorithms derived from the SLD schema.

Our interpreter is presented here with little mathematical formalism aiming at easing to understand the algorithmic schema, one of our highest priority objective in this article. The reader interested in more technical aspects is addressed to [Escalada-Imaz, 89] and [Escalada-Imaz, 90].

This article is structured as follows. In Section 2 we present the syntax, semantic and the logical inferences calculus. In Section 3, we extend the well-known SLD strategy to the multivalued datalog language considered in this article. Then, a fast method to process unifiers is detailed. After, in section 5, we describe our original interpreter and show its efficiency measured with respect to the SLD schema.

2.- A Class of Multiple-Valued Datalog Languages

Here below, we define the syntax, semantic and the logical inference for multiple-valued definite clauses whose terms are constants or variables.

Syntax

Definition (term): A term is a constant or a variable.
Definition (mv-atomic formula): If $p$ is a predicate symbol and $t_1, \ldots, t_n$ are terms then $p(t_1, \ldots, t_n)$ is an atom. If $p(t_1, \ldots, t_n)$ is an atom and $\alpha(0, 1)$ then $(p(t_1, \ldots, t_n); \alpha)$ is a mv-atom.

Definition (program). A program is composed by one goal and a set of multiple-valued facts and rules defined as follows:

- **fact**: it is a mv-atom $(p; \alpha)$.
- **rule**: it is a pair $(p \leftarrow \{p_1, \ldots, p_k\}; \alpha)$ where $p$, and $p_1, \ldots, p_k$ are atoms and $\alpha(0, 1)$.
- **goal**: it is a set $\{Q_1, \ldots, Q_n\}$ where $Q_i$ is a pair $((q_1, \ldots, q_i); \alpha)$, $q_i$ being atoms and $\alpha(0, 1)$.
- **subgoal**: it is pair $Q_i$ in a goal $Q$.

Definition (clause). A clause $C$ is any element of a program.

**Semantic**

Definition (valuation): A valuation $\varphi$ is a mapping from variables to a non-empty domain $D$.

Definition (interpretation): An interpretation $I$ of a multiple-valued language is a non-empty domain $D$ and a mapping $I$ that associates:

- each constant symbol $a$ with an element $I(a) \in D$;
- each predicate symbol $p$ with a mapping $I(p); D^n \rightarrow [0, 1]$, where $n$ is the arity of $p$;
- the "$\leftarrow$" and "$;$" logic connectives with mappings $[0, 1]^2 \rightarrow [0, 1]$ noted $I(\leftarrow)$ and $I(;)$, respectively.

Remark 1: By the well-known associativity and commutativity properties of $I(;)\: \text{its mapping can be denote by:} [0, 1]^k \rightarrow [0, 1]$, whenever $I(;)\: \text{is applied to n atoms.}$

Definition (semantics of terms): Let $I$ be an interpretation, $\varphi$ a valuation and $t$ a term. The meaning $\varphi(t)$ of $t$ is an element of the domain $D$ of $I$ defined as follows:

- if $t$ is a constant symbol $a$ then $\varphi(t) = I(a)$;
- if $t$ is a variable symbol $x$ then $\varphi(t) = \varphi(x)$.

Henceforth, $I = (S; \alpha)$ denotes that the interpretation $I$ is a model of the sentence $(S; \alpha)$.

Definition (semantics of wff's): Let $I$ be an interpretation and $\varphi$ a valuation. The meaning of a wff is defined as follows:

- $I \models (p(t_1, \ldots, t_n); \alpha)$ iff $I(p); \varphi(t_1), \ldots, \varphi(t_n)) \geq \alpha$ for all valuations;
Aspectes formals

- Failed Subgoal
  \[ (\beta, \alpha, Q_1, \ldots, Q_n), \beta \leq \alpha \]
  Fail

- Successful Goal
  \[ \top \]
  Yes

Notice that the multiple-valued modus ponens is a natural extension of the bi-valued classical rule.

**Theorem (Logical Correctness):** Let P be a program Q a goal: P \[\vdash\] Q if P \[\models\] Q.

The proof is not given here by lack of space.

3. The Multi-Valued SLD-like Interpreter.

In this section we design an interpreter based on the previous calculus. Given a program P and a subgoal Q\[\vdash\] \((q_1, \ldots, q_k)\); \(\alpha\) the aim of the interpreter is to find a truth degree \(\beta\) such that P \[\models\] \((q_\beta, \alpha)\) with \(\beta \geq \alpha\).

Here below, we describe an interpreter that can be seen as an adaptation of the SLD Prolog procedure to our defined multiple-valued datalog case.

**Data structures**

For each program clause C we use the following data structure:

- consequent(C)=p, where p is the consequent of a mv-rule (p \[\leftarrow\] p_1, \ldots, p_n, \alpha) or the atom p of a mv-fact (p, \alpha);
- antecedents(C)={p_1, \ldots, p_n} is the set of antecedents of a mv-rule or the empty set for mv-facts;
- val(C)=\alpha represents the truth value attached to C;

and for each atom p:

- Alternatives-Clauses(p)={C_1, \ldots, C_n} is the set of clauses in the program such that the atom or the consequent of a fact or a rule respectively, can be unified with p.

The formal description of the interpreter algorithm is as given below. We note C a rule or a fact and Q a goal formed by a set of subgoals Q_i. Also, \(\delta\) is the unifier with no substitution variables, i.e. the most general unifier among all the unifiers.

**Interpreter (Q)**

- if Q=\{\} then Halt(Yes)
- if Q=[(\beta, \alpha)] then
  - if \(\beta \geq \alpha\) then Return(Interpreter([Q_2, \ldots, Q_n]))
  - else Backtrack to last state \(\{Q_1, \ldots, Q_n\}\) with pending alternatives for atoms in Q_1
- Subgoal(Q_1, 1, \delta) \rightarrow (\beta, \alpha)
- if (\beta, \alpha)= (Fail, Fail) then Return (Fail)
- Return (Interpreter ( \{ (\beta, \alpha), Q_2, \sigma, ..., Q_n, \sigma \} ))

End

**Subgoal(Q, \sigma, \beta)**

- if Q=\{1\}, \beta then return(\beta, \sigma)
- Q=[(q_1, ..., q_k), \alpha]
- Take C in pending-alternatives(q_1)
- if C = (p, \delta) and \(\theta=\text{mg}u(q_1, p)\) \# fail do:
  - Return(Subgoal([(q_2, ..., q_k), \sigma, \theta, \theta, l, (\beta, \delta)])
- if C=(p\[\leftarrow\] p_1, ..., p_n, \delta) and \(\theta=\text{mg}u(q_1, p)\) \# fail do:
  - Subgoal([(p_1, ..., p_k), \theta, \sigma, \theta, l, (\beta, \delta)]) \rightarrow (\beta, \theta)
  - Return(Subgoal([(q_2, ..., q_k), \sigma, \theta, \theta, l, (\beta, \delta)])
- if there is not exist alternatives for Q_1 then do:
  - Return(Fail, Fail)

End

**Theorem (procedural correctness).** Given a program P, and goal Q=\(\{Q_1, \ldots, Q_n\}\), the function Interpreter(P, Q) return "Yes" if Q follows logically from P, P \[\models\] Q.

The INTERPRETER runs in a depth-first backwards manner, applying the modus ponens, i.e. a literal p in the goal Q is deduced if all the antecedent literals of a rule whose consequent unifies with p are deduced.

Thus, in every state Q of the interpreting process, a program clause (C, \delta) such that consequent(C) unifies, via an mgu \(\theta\), with the leftmost atom p_1 of the subgoal Q_1, is selected leading to the new state:

\(\{p_1, ..., p_1, \{n\}, \theta, \delta\}, \{p_2, ..., p_m\}, \theta, \sigma\}, Q_n, \theta\})

If no program clause unifies with the leftmost atom of the goal or the truth value \(\alpha\) attached to the goal is smaller than \(\delta\) then the interpreter backtracks, since from the current state, only states with truth value smaller than \(\alpha\) can be reached. This process continues until a solution state with an empty set of atoms is reached.

**Example 1:** Let P be the following datalog program:

C1: (m(x, y) \[\leftarrow\] r(x), n(y), p(x); 0.9)
C2: (r(z) \[\leftarrow\] t(z); 1)
C3: (r(u) \[\leftarrow\] s(u, w), p(w); 0.5)
C4: (r(d); 0.9)
Aspectes formals

C5: (r(c); 0.8)
C6: (s(b); d; 0.1)
C7: (p(c); 1)
C8: (p(b); 1)
C9: (n(b); 0.9)
C10: (n(c); 0.8)
C11: (t(a); 0.9)

The I,(a,b) that we will use is I,(a,b)= a x b (a well-known T-norm). Let us assume that we wish to check whether there is a solution to the goal {m(x,y)} with a degree higher than α=0.5.

Calling INTERPRETER((m(x,y); 0.5)) the interpreter returns the substitution {x/c, y/b} and the truth value 0.66. Nevertheless, this is not the only solution. This is the first solution that is found when the search space is traversed in a depth-first manner. The other solution is the substitution {x/c, y/c} with a truth value 0.58. Figure 1 in Appendix shows the search space for the goal {m(x,y)} with a threshold 0.5.

The definition of INTERPRETER can be modified to obtain all the solutions by changing the first line for

"if Q={} then print("result"); return".

The interpreter described above is based on unification steps, depth-first search and chronological backtracking. As an illustration of how it works, the different states successively reached by the interpreter are displayed below and pictured in Figure 1 in Appendix.

The initial situation is represented by state e1. The interpreter goes on working until it reaches state e5 which corresponds to the computations carried out to traverse the leftmost path of the search space.

e1 = (m(x,y), { }, 0.5); C1
e2 = (t(x), n(y), p(x), { }, 0.45); C2
e3 = (n(y), p(x), { }, 0.45); C11
e4 = (n(y), p(a), { z/a, x/a }, 0.81); C9
e5 = (p(a), { z/a, x/a, y/b }, 0.5) fail; C7
e6 = (p(a), { z/a, x/a, y/c }, 0.65), (0.5) fail; C8
e7 = (s(x, w), p(w), p(x), { u/x }, 0.45), (0.5) fail; C3
e8 = (n(y), p(d), { x/d }, 0.81), (0.5); C4
e9 = (p(d), { x/d, y/b }, 0.73), (0.5); C9
e10 = (p(d), { x/d, y/c }, 0.65), (0.5) fail; C10
e11 = (n(y), p(c), { x/c }, 0.72), (0.5); C5
e12 = (p(c), { x/c, y/b }, 0.65), (0.5); C9
e13 = (1), (x/c, y/b), 0.65), (0.5); C7

Since no solution can be found it backtracks to state e4 and generates state e6, but as this new path is also a failed, the interpreter backtracks to state e2 and generates state e7. As the truth value of state e7 is smaller than 0.5 it backtracks again to state e2 and generates states e8 and e9. State e9 represents a failed path and then the interpreter backtracks to e8 and generates state e10 which is also a failed path and hence, it backtracks to e2. From e2 states e11, e12 and e13 are generated. State e13 is a successful path (since the goal is empty and its truth value is bigger than the threshold 0.5). Thus the computed answer for the initial goal is {x/c, y/b} and its truth value is 0.65.

The previous interpreter has been defined as an adaptation of the Prolog procedure, that is, based on unification steps, depth-first search and chronological backtracking. This interpreter is particularly inefficient when it has to run on general multiple-valued logic programs instead of classical bi-valued first-order logic programs. In the latter case is not enough to find a branch ending with the empty clause, it must also have a truth value β greater than a given threshold α. Therefore, the search space that has to be examined is always bigger than or equal to the search space explored in the classical case.

We can see that the previous interpreter has several drawbacks:

- A high computational cost is paid due to inappropriate handling of substitutions as can be seen in the previous example. Thus, for each inference step:
  - first, a unifier θ must be obtained (we call it inference unifier) between a head clause consequent(C) and the literal q1 in the current state to be reduced;
  - second, this unifier θ must be applied to all the literals {q2,...,qn} in the state at hand and to all the antecedents(C);
  - finally, a composition between the previous state unifier θ and the inference unifier θ must be performed in order to obtain the present state unifier.

For instance, in the given problem, if we consider the state with unifier {x/a} and literals {t(x), n(y), p(x)} then the unification between t(x) and t(a) leads to the unifier {x/a}.

This substitution is then applied to the remaining literals of the goal and in a antecedent(C) which are instantiated in {n(y), p(a)} (in this particular case antecedents(C)={}).
Finally, the composition between the actual state substitution \(z/x\) and that just obtained \(z/a\) is performed yielding \(z/a,x/a\). Moreover, it can be shown [Escalada-Imaz, 90] that when first-order terms (not only constants) are used the size of the terms of the substitutions can grow exponentially, i.e. the complexity of a unification step can also increase exponentially.

- Each search space associated to the reduction of a given literal can be scanned many times. In the example, this is the case for literals \(p\) and \(a\). It can easily be proved that the number of times that a search space might be examined can increase exponentially [Escalada-Imaz, 90]. Thus, in addition, the elimination of redundancies in usual backtracking-based schemas [Pietrzykowsky and Matwin, 82], [Pereira and Porto, 82], [Lloyd, 87] can take an exponential time [Wolfman, 86].

To eliminate these redundancies, we aim at designing a new interpreter based on a completely different search strategy [Escalada-Imaz, 90] from the classical depth-first search and chronological backtracking, and with a different processing of unifiers. We will see that the proposed method avoids the redundancies of unsatisfying backtracks.

4. Efficient Computations of Unifiers

In the computation of unifiers, we note that the role of substitutions is to check that previous substitutions executed from the initial state until the current one represented by the state substitutions are compatible with the variable bindings imposed by the present inference unification step. That is, to check whether previous inferences are compatible with the present one.

Aiming at computing efficiently this compatibility, we define a new operation SIMPLIFIER, SPL, for short (equivalent to the rational infinite terms unification, e.g. [Colmerauer, 84]). From now on, we will deal with non idempotent substitutions \(p\) and we will associate them with a system of variable bindings \(\{x-t/x\in p\}\).

The SPL operation is applied to pairs of systems of variable bindings:
- one coming from the state substitution, \(S_\sigma=\{x-t/x\in \sigma\}\),
- the other one from the inference substitution \(S_\theta=\{y-t/y\in \theta\}\)

and, it ends successfully iff the global system of bindings \(S_\sigma\cup S_\sigma\) can be unified.

The proposed approach does not need to apply substitutions to literals.

We will now describe the SPL operation. In the literature unifiers are usually substitutions that verify the idempotent property, i.e. \(\sigma\cdot\sigma=\sigma\). To compute this kind of unifiers any first-order unification algorithm can be used. However, it has been shown [Calmerauer 84, Escalada-Imaz 90] that it is not mandatory to employ unifiers with the idempotent property.

Instead of handling idempotent substitutions, systems of bindings can be used where a variable is bound to at most one term. The compatibility between
- the previous inference steps represented by the system of bindings \(S_\sigma\) and,
- a present inference step represented by another system of bindings \(S_\theta\),
is checked throughout the application of the SPL function over the two systems of bindings. SPL\((S_\sigma\cup S_\sigma)\) obtains another system of bindings \(S"=SPL(S_\theta\cup S_\sigma)\) such that \(\{x/t/ x\in S"\}\) is an (eventually non idempotent) mgu of the global set of bindings appearing in \(S_\theta\) and \(S_\sigma\).

For example, if \(S_\theta=\{x-y,y-a\}\) and \(S_\sigma=\{x-a\}\), SPL\((S_\theta\cup S_\sigma)\) is \(\{x-y,y-a,x-a\}\). Yet, if \(S_\theta=\{x-b\}\) then SPL\((S_\theta\cup S_\sigma)\) would fail to obtain a mgu.

The SPL operation can be implemented by using only the first phase of some well-known first-order unification algorithms [Paterson and Wegman, 78], [Martelli and Montanari, 82], [Escalada-Imaz and Challa, 88]. Hence, it is obvious that SPL that use only the first phase is computationally cheaper than using all the phases of the mentioned algorithm which are necessary to obtain an idempotent unifier.

The mechanism of the interpreter for processing bindings is now the following:

A system of variable-bindings \(S_\theta=\{y_1-t_1,...,y_n-t_n\}\) is associated to each state of the interpreter. First, the inference system \(\text{SPL}(\{\text{consequent}(C)-q\})=S_\theta\) is calculated, such that \(\theta=\{x/t/x\in S_\theta\}\) is a (not necessarily idempotent) mgu of the pair of literals \(\{\text{consequent}(C)-q\}\) involved in the inference step.

Then, for each possible inference system \(S_\theta\), the bindings \(S"=\text{SPL}(S_\theta\cup S_\sigma)\) such that \(\{x/t/ x\in S"\}\) is an (eventually non idempotent) mgu of the term pairs in \(S_\theta\cup S_\sigma\) is obtained.
Aspects formals

Now the proposed method of computing systems of bindings is significantly faster than conventional methods of computing idempotent unifiers. It can be proved [Escalada-Imaz, 90] that the operation of obtaining an idempotent unifier of a set of pairs of terms is computationally considerably harder than determining systems of bindings representing non idempotent unifiers. Thus,

1. The conventional way of obtaining the idempotent mgu of \{consequent(C)-q_1\} is computationally more expensive than obtaining only the equivalent system of bindings of \{consequent(C)-q_1\}. Furthermore, literal q_1 used in the unification in conventional interpreters in the current goal is bigger than its original literal q_1 because some previous mgu's (summarised in the current state substitution \sigma) are successively applied to the literal goals instantiating the variables of the original literals q_1 by terms which make increase, state by state \...((q_1,\sigma_1),\sigma_2)...\sigma_k=q_1', their size until they become the next literal to extract from the list of goals.

2. The composition of unifiers has a higher computational cost in classical interpreters than the operations needed in \text{SPL}(S_0\cup S_\sigma) because in the proposed interpreter, as mentioned previously, the size of the unifiers can increase exponentially with respect to the size of the literals involved in inference steps (when first-order terms are used) whereas the computing of \text{SPL}(S_0\cup S_\sigma) only increases proportionally.

3. Note that no application of substitutions over set of literals in the goal is not necessary in the proposed approach.

5 A New and Efficient Multi-valued Datalog Interpreter Schema

A new search strategy is designed which is oriented towards the search for the total set of existing solutions [Escalada-Imaz, 90], instead of being oriented towards the search for one solution as backtracking-based methods do. This strategy has the merit of avoiding most of the redundancies of previous schemas. Moreover, conversely to backtracking-based methods, the search space associated to the resolution of each literal in a clause is run over only once. Indeed, in the previous example and using chronological backtracking, the search space for the solution of literals n and p are obtained more than once (3 for n and 6 for p).

The idea behind the proposed search strategy is as follows. All the possible paths beginning in a state e with literals \{q_1,q_2,...,q_n\} and ending in states e' with literals \{q_2,...,q_n\} are scanned. Hence, states of the kind \{(q_2,...,q_n),(S_1,\alpha_1),...,S_k,\alpha_k)\}, where each \{S_1,\alpha_1\}, 1\leq i\leq k, means that \text{S}_i is the system of bindings constructed traversing one path \text{e} \rightarrow \text{e'} and \alpha_i is its truth value associated to this i-th path obtained by the T-norm.

The meaning associated to each state \text{e'}=\{(q_2,...,q_n),(S_1,\alpha_1),...,S_k,\alpha_k)\} is that there exist k different paths from the initial state \text{e} with literals \{q_1,q_2,...,q_n\} to a state \text{e'} with literals \{q_2,...,q_n\}, where each \{S_1,\alpha_1\}, 1\leq i\leq k, means that \text{S}_i is the system of bindings constructed traversing one path \text{e} \rightarrow \text{e'} and \alpha_i is its truth value associated to this i-th path obtained by the T-norm.

Thus, in a general case, a state \text{e}'=\{(q_1,q_2,...,q_n),(S_1,\alpha_1),...,S_k,\alpha_k)\} contains k pairs \{(\text{system of bindings, truth value})\} recollected from the initial state until the current state \text{e}'. Now, to verify which of these systems are compatible with a possible inference applicable to this current state, the SPL function is applied to a clause C whose consequent(C) has the same predicate symbol that q_1.

Thus, the present inference set systems is obtained as follows:
- first, one calculates \text{SPL}((\text{consequent}(C_j)-q_1))=\text{S}_j, 1\leq j\leq m, where consequent(C_j) is any clause in the program P having the same predicate symbol that q_1.
- The systems \text{S}_j = “fail” are removed; the successful ones represent potential inferences applicable the current state.
- second, the compatibility between (1) systems \text{S}_j representing potential inferences, and (2) systems \text{S}_i, in the current state is checked and thus, \text{SPL}(\text{S}_j\cup \text{S}_i)=\text{S}_{j+i}, 1\leq i\leq m, is obtained.

Then, the next state visited is:
\{(\text{antecedents}(C_j)-q_2,...,q_n),(S_j',\alpha_1'),...,S_k',\alpha_k')\}, 1 \leq 1 < \text{Min}(n,m).

These states are visited and developed in a depth-first manner until they reach states \text{e}_1,...,\text{e}_w with set of
literals \{q_2, \ldots, q_n\}. These sets are then grouped in a unique state whose literals are \{q_2, \ldots, q_n\} and whose set of pairs (system of bindings, truth value) is the union of the sets of pairs in states e_1, e_2, \ldots, e_w. Note that the paths between states with literals \{q_1, q_2, \ldots, q_n\} and \{q_2, \ldots, q_n\} are explored only once.

Hence, a major improvement in efficiency with respect to conventional methods is achieved since, as it has been mentioned, the methods on backtracking could execute this search space up to an exponential number of times.

The following algorithm is a more efficient unifier interpreter with a more efficient computation of unifiers and an improved exploration of the search space. The interpreter function is almost the same that for the SLD interpreter. Hence, we show the second function Subgoal.

Subgoal(Q)
if Q = \{(S_1, x_1), \ldots, (S_n, y_n)\} then
return((S_1, x_1), \ldots, (S_n, y_n))
if Q = \{(q_1, \ldots, q_k), (S_1, x_1), \ldots, (S_n, y_n)\} then
Let \{(pi, bi) : 0 \leq i \leq 1\} with \theta_i = \mu(q_i, q_i). Do:
(SPL(S_1, S_2), \ldots, (S_n, S_n))\{\theta_1, \theta_2, \ldots, \theta_n\} \rightarrow \text{Solve}(q_1)
Let \{(pi\rightarrow p_1, \ldots, p_0, \theta_1)\} with \theta_i = \mu(q_1, q_i). Do:
(SPL(S_1, S_2), \ldots, (S_n, S_n), \theta_1, \theta_2, \ldots, \theta_n) \rightarrow \text{Solve}(q_1)
Remove Fail elements from \text{Solve}(q_1)
Subgoal((p_1, \ldots, p_n), \text{Solve}(q_1)) \rightarrow \text{Solve}(q_1)
Return(Subgoal((q_2, \ldots, q_k), \text{Solve}(q_1)))

Example 2: We will consider the program of Example 1. Its execution is depicted in figure 2 in Appendix. The interpreter begins with state e_1, namely Interpreter \((m(x, y), \{(1, 0.5)\})\).

In this state, only one inference can be performed to reduce the literal \(m(x, y)\). Applying this inference leads to state e_2. Now, to solve \(r(x)\), 4 possible inferences with clauses C_2, C_3, C_4, and C_5 can be applied.

\[ e_1 = ((m(x, y), \{(1, 1), 0.5\}) \quad C_1 \]
\[ e_2 = (((x, n(y), p(x))), \{(0, 0.9), 0.5\}) \quad C_2 \]
\[ e_3 = (((x, z), p(x))), \{(x, z), 0.9), 0.5\}) \quad C_3 \]
\[ e_4 = ((n(y), p(x))), \{(x, z, a), 0.8), 0.5\}) \quad C_4 \]
\[ e_5 = ((z(w, u), p(w), n(y), p(x))), \{(x, u), 0.45), 0.5\} \quad C_4 \]
\[ e_6 = ((n(y), p(x))), \{(x, d), 0.81), 0.5\}) \quad C_5 \]
\[ e_7 = ((n(y), p(x))), \{(x, c), 0.72), 0.5\}) \quad C_6 \]
\[ e_8 = ((n(y), p(x))), \{(x, z, a), 0.8), (x, d), 0.81), \{(x, e), 0.5\}) \quad C_9 \]

The search following this path is stopped and the other possible path reduction literal with clause C_10 is attempted yielding state e_10. These two last states are then grouped in state e_11. By applying the reduction of literals with clauses C_7 and C_9 the final solution state e_13 is determined. Note that with this schema there are not backtracking operations and the search space corresponding to each literal is scanned at most once.

6. Conclusion
One of the restrictions of logic programming languages is that each predicate can take only two values. We have presented a way of overcoming this
drawback by exposing a Multiple-Valued datalog programming.

The interpreter has been designed in such a way that the computing of unifiers and the exploration of the search space are carried out efficiently. This has been proved in an informal way, illustrating the scanning search space followed for both the SLD schema and the proposed methodology.

As future work we plan to integrate the negation as failure meta-rule and the cut operator to implement a programming environment for uncertainty logic programming.

Acknowledgements

This work has been economically supported by the project TIC 96-0721-C02-02 funded by the DGICYT.

Bibliography


Appendix

Figure 1: search space for \{m(x,y)\} and threshold 0.5

Fig. 2: Steps to solve Exemple 1 with the proposed interpreter
Proposta de modelització de primitives computacionals per a dominis discrets

Mª Teresa Signes Pont
D. Tecnologia Informática i Computació
Escola Politécnica Superior.
Universitat d’Alacant
Ctra de Sant Vicent s/n
03080 Alacant
teresa@dtic.ua.es

Juan M. García Chamizo
D. Tecnologia Informática i Computació
Escola Politécnica Superior.
Universitat d’Alacant
Ctra de Sant Vicent s/n
03080 Alacant
juanma@dtic.ua.es

Jerónimo Mora Pascual
D. Tecnologia Informática i Computació
Escola Politécnica Superior.
Universitat d’Alacant
Ctra de Sant Vicent s/n
03080 Alacant
jeronimo@dtic.ua.es

Resum
Tot allò relacionat amb el tema de la informació suscita, ara per ara, gran interès. És clar que aconseguir la informació relativa a un domini específic és la condició per a perfeccionar-ne el coneixement; d’altra banda, existeix també un altre motiu d’atenció: és quan la pròpia informació esdevé objecte formal d’estudi de la ciència i la tecnologia. Això significa que l’interés recau sobre els aspectes característics i genuïns del processament de la informació, i no sobre el domini específic. És tracta, doncs, de trobar un espai de representació supra-domini i una col.lecció d’operacions senzilles i útils. Des d’aquesta perspectiva unificadora, a més a més de la computació artificial, cal admetre que existeixen altres viés per a processar informació: s’observen en multitud de sistemes naturals, físic-químics o biològics, aquestes últims abordats de vegades des de la neurofisiologia [9]. Per tal de donar-li caràcter general al modelatge de fenòmens macroscòpics bàsics, inherents al comportament computacional de sistemes reals, naturals o artificialss, aquest treball proposa un espai de representació nou: assumim que la informació ha estat recollida en forma analògica, comprimida i quantificada degudament, i aquest és el punt de partida. Així, el que finalment considerem és una magnitud adimensional, que ens cal codificar mitjançant nombrosos que pertanyen a dominis fínits discrets. Les operacions que hi definim després són internes, però no tenen la propietat associativa, i aquesta manca planteja la necessitat d’una ordenació per a poder operar: la que proporciona el temps, per exemple. L’avantatge de modelar utilitzant dominis fínits i operacions amb poques propietats és que es guanya senzillesa en la formalització: en efecte, una formulació és minimal o canònica quan és coherent amb les característiques d’un fenòmen; així s’evidencien regularitats, invisibles quan la formulació és complicada. Postulem que es pot apagar, fins i tot, a definir una col.lecció d’operacions primitives, a partir de les quals construir funcionalitats més complexes. La implementació física de les primitives podria ser la base de màquines alternatives, especialment indicades per a computar fenòmens macroscòpics, i amb possibiltats reals d’abaratiment. Per acabar, ens centrem en el fenòmen de saturació com a primitiva, presentant alguns dels sistemes computacionals reals on apareix.

Paraules clau: informació, computació artificial, computació natural, primitiva, arquitectura, convolució.

Introducció
Ens interessen els aspectes procedimentals o funcionals del processament de la informació; així, no considerem alguna particularitat referida a la naturalesa física o estructural del sistema. Al contrari, partint de les nombroses manifestacions observades en els sistemes reals, tractem de renunciar a l’expressió essencial i comuna a tots ells, traduint-la a un model funcional abstracte.
Per raons històriques i de caire antropocèutic, el desenvolupament de sistemes computacionals artificials ha suscitat fins ara el major interès: es tractava, per damunt de tot, d’alliberar l’operador humà de l’execució de tasques intel·ligents de baix nivell com el càlcul i tot allò relacionat amb els aspectes algorítmics de la conducta.
Aspectes formals

[6]. No es pot negar, llavors, l'existència d'altres sistemes que no han estat ideats ni fabricats per l'èsser humà, però que són capaços de captar la informació de l'entorn i processar-la, produint alguna resposta. Doncs, almenys en apariència, aquests sistemes també compten. Pensem per exemple en alguns sistemes naturals, biològics o físic-químics com són el cervell humà, el sistema nerviós d'alguns mol.luscs, una colònia de formigues, una reacció molecular o un fenòmen òptic. Més encara: el cervell és capaç de percebre, comprendre, aprendre, recordar, tenir opinions o sentiments, decidir, crear [8]; el sistema nerviós d'alguns mol.luscs marins pot aprendre per condicionament [4], [1], recordar, oblidar; s'observen també en certs muscles comportaments que anomenarem proto-amorosos, com és el reconèixer la parella pròpia durant tot el temps que dura l'apariacions... Una colònia de formigues és capaç de realitzar tasques de gran complexitat com explorar el seu entorn, construir galeries o decidir entre dues fonts d'alimentació, mentre que un individu a soles mai no ho podria fer [10]. El genoma dels organismes superiors, que conté la informació codificada necessària per a l'elaboració d'unes 100.000 proteïnes diferents, actua com una xarxa de computadors, en la qual uns gens regulen l'activitat d'altres, directament o a través dels seus productes. Totes les cèl·lules d'un mateix organisme tenen aproximadament les mateixes instruccions genètiques i la diversitat d'aquestes s'aconsegueix a través de la diversitat de l'activitat genètica, no dels gens. Acabem parland del comportament de rellotge d'alguns cristals, induïts per una excitació de freqüència compatible amb la seva pròpia, del comportament de "memòria" d'alguns materials que exhibeixen cicle d'històresi o de l'aparició d'estructures fractals en una peça de ferro que es refreda, quan aplega a la temperatura de Curie.

Fent abstracció de la diversitat manifesta d'aquests sistemes, queda una essència compartida per tots ells, que és el seu comportament macroscòpic, resultat del processament de la informació rebuda de l'entorn o d'alguna guia interna. El propòsit principal d'aquest treball és trobar-ne una formalització adient: que siga independent de la naturalesa o estructura del sistema, referida essencialment a la seva funció, que pose en evidència qualitats emergents sense considerar aspectes microscòpics. Per això, situem el punt de partida allà on acaba la diversitat entre els sistemes fractals, és a dir després de captar, comprenir i quantificar la informació mitjançant sensors i acondicionadors, quan ja s'ha aplegat a una magnitud adimensional que ens cal codificar. La codificació empra nombros que pertanyen a conjunts finits discrets sobre els quals es defineixen operacions internes, però no associatives, i aquesta manca planteja la necessitat d'una ordenació per a poder operar: la que proporciona el temps, per exemple. L'avantatge de dissenyar operacions d'aquest tipus és la senzillesa de la formalització resultant. Així, per a modelar un efecte no lineal tan freqüent com la saturació, per exemple, no s'utilitzaria la funció sigmoidal, combinació d'operacions diverses com la multiplicació, la inversió i l'exponencial, definides en \( \mathbb{R} \). Com es veurà en l'exemple, es pot aplegar al mateix resultat amb una operació més senzilla, definida en un subconjunt finit d'enters. Prosseguint amb la mateixa idea, altres efectes com les oscil·lacions, l'esmentitem o la divergença es poden formalitzar també sense necessitat de recórrer a les funcions circulars o exponencials tradicionals. Cada operació, representativa d'un efecte simple, considerat elemental, s'anomena primitiva computacional i interven en la configuració d'arquitectures més complexes, representatives d'efectes més complexes també. Es postula que la contrapartida física de les primitives podria donar lloc a la realització de portes funcionals alternatives, a partir de les quals configurar màquines noves, especialment indicades en el cas de fenòmens emergents, col·lectius i/o macroscòpics.

1. Model funcional basat en operacions en dominis discrets simètrics

1.1. Marc formal

Formulació del problema

Com ja s'ha dit, es refereix a la informació de manera genèrica, tal com es presenta després de la captació, compressió i quantificació, és a dir a partir d'un conjunt, de "paquets"; cada paquet es codifica per un número i es vol definir una llei de composició genèrica (+) entre els nombros.

Formalment, \( J \) és un conjunt de nombres enteros simètric, centrat en el zero i de dimensió finita.

\[
J \times J \rightarrow J
\]

\[
(a, b) \rightarrow a + b \quad \text{opercació interna}
\]

Notació funcional de l'operació

\[
F(a, b) = F_1(b) = a + b
\]

\[
F(a, b, c) = F(F_1(b), c) = F((a+b), c) = a + (b+c) = F_{a+b}(c)
\]
Aspectes formals

Hem adoptat de fet una associativa per l’esquerra; intuitivament el que està més a l’esquerra és com si hagués aparegut abans en el temps.

Generalitzant:

Seqüència aleatòria d’elements de J:

$$\sigma = \{a_1, a_2, \ldots, a_n\}$$

$$F(\sigma)=((a_1+a_2)+ a_3)\ldots+a_n)$$, que es desglossa així:

- Pas n° 1: $$a_1$$
- Pas n° 2: $$a_1+a_2$$
- Pas n° 3: $$(a_1+a_2)+a_3$$
- ... ...
- Pas n° n: $$(((a_1+a_2)+a_3)\ldots+a_n)$$

1.2 Domini de representació i lloc de composició interna

El domini numèric és important a l’hora de codificar i de fixar el tipus de variables. En cas de voler modelar un efecte de saturació a partir de la suma no acumulativa de variables discretes, amb valors en un conjunt finit, i amb característica de mitjana aritmètica, proposem un conjunt d’enters de cardinal finit, impar, que anomenem J. El domini J = (-Z/2, Z/2). Centrar el conjunt en el 0 no és necessari, però dóna ésser expressiva per la situació intuitiva del terme mig de l’escala, i la presència del signe no afegeix complexitat computacional. Com que l’operació no és acumulativa ni cíclica, la unitat no té grandària uniforme i depen del cardinal del domini.

El fenòmen esmentat així és la saturació, que apareix en els processos de creixement. Aquest efecte es caracteritza per una evolució prou ràpida al principi, que decresce paulatinament quan el sistema que creix s’aproxima a uns punts anomenats atractors. Si hi ha dos atractors, el resultat és una bipolarització. En el domini, els atractors són els extrems -Z/2 i +Z/2. La formalització és la següent:

$$F: J \times J \rightarrow J$$

$$(f, g) \rightarrow F(f, g)$$

$$F(f +, g-) = f_+ + g_-$$

$$F(f -, g+) = f_- + g_+$$

$$F(f +, g+) = f_+ + g_+ - \text{min} \left\{ |f_+|, |g_+| \right\} \quad (1)$$

$$F(f -, g-) = f_- + g_- + \text{min} \left\{ |f_-|, |g_-| \right\}$$

El signe que segueix a l’argument indica si és positiu o negatiu. L’operació és commutativa, té element neutre que és zero i cada element té un simètric únic, no és associativa, per tant cal fixar algun criteri per tal d’operar més de dos elements. Una solució pot ser forçar l’associativa per l’esquerra.

2. Experimentació

2.1 Representació gràfica i interpretació de resultats

L’experiment consisteix en il·lustrar el fenòmen de saturació generant una seqüència aleatòria d’elements de J+, i aplicant-li l’operació que correspon de forma iterativa: s’opebra el primer terme de la seqüència amb el segon; el resultat amb el tercer, i així successivament, forçant una associativa per l’esquerra. Els resultats s’ordenen segons una seqüència indexadora de números naturals i correlatius, o passos, i es representen en una gràfica: a l’eix horitzontal el pas, i al vertical la suma parcial que li correspon. En cada pas, l’entrada és el punt blanc i el resultat, el punt negre. La gràfica indica la taula a la que es refereix, el cardinal de J, i el número de passos.

Fig.1: Representació gràfica de la funció definida per (1) 150x200

Figura 1: observem una pujada ràpida seguida de saturació definitiva que s’aconsegueix en poques etapes. S’ha imposat com a primer element de la seqüència l’extrem inferior de J+ (que és 0). Per a eixir de la saturació caldrà perllongar la seqüència de entrada amb elements de J- (o almenys de J).

2.2 Sistemes computacionales reals que exhibeixen saturacions o esmormitzents: representacions i comentaris

En el que segueix es presenten alguns exemples de sistemes computacionales, modelables en dominis J amb...
l'operació que hem definit. En cada cas es comentaràn les restriccions adicionals a formular per tal d'adequar-se al cas genèric.

2.2.1 Càrrega d'un condensador a tensió constant [5]
Encara que es tracte d'un exemple molt acadèmic, és interessant fer alguns comentaris: en el domini real, la càrrega d'un condensador pot considerar-se com una operació acumulativa en la qual la contribució d'un pas és sempre menor que la del pas precedent, fins que es fa null en algun moment. Si el pas és molt petit, l'aparició de la representació és la línia continua (Fig.2). En el domini J, les variables del model són les mateixes (tensió i temps), però canvia la manera d'operar: la suma ja no és acumulativa i tan sols contribueix al resultat el major dels sumands. Per a perfeccionar el model estan les restriccions següents:

- El primer element de la seqüència és zero, per conveni.
- La seqüència d'entrada \( v_0, v_1, v_2, \ldots \) de J+ està ordenada per ordre creixent; el resultat de la suma és doncs sempre igual a l'element que s'acaba d'inserir, i cap element de la seqüència no està repetit.
- La probabilitat d'aparició d'un element augmenta a mesura que l'element s'acosta a l'extrem superior de J+.

![Fig.2: Càrrega d'un condensador](image)

La seqüència de temps és \( t = kT \).

Així, una operació bastant complexa en \( \Re \) pot emular-se per una operació bastant simple en J, si es realitza sobre una seqüència adient.

2.2.2 Llei A de compressió logarítmica

La figura 3 representa l'aproximació per trams de la llei A de compressió logarítmica, que s'emplea en els "codecs" (PAL,93). Act, la seqüència indexadora està formada per intervals no constants d'una serral analògica, x, (ordensats per ordre creixent); però la contribució de cada pas és la mateixa sempre (equivaltent digital, n).

En aquest cas, l'operació expressada per la taula 1 també val, amb les restriccions següents.

- La seqüència d'entrada està formada per tots els elements de J+, ordenats de forma creixent.
- Es permet la repetició d'elements o la inversió local de l'ordre creixent, però d'una manera reglada.

En el cas de la figura 3, una seqüència possible seria \( 1, 2, 3, 5, 4, 6, 7 \ldots \), per exemple, on la permetissivitat augmenta en un element cada dos elements correlatius (números en negreta).

2.2.3 Aprenentatge hebbià

Per acabar, comentem breument el cas de l'aprenentatge (hebbià) d'un concepte, a partir de la presentació de patrons que contenen el concepte en major o menor grau. En 1949 Hebb introduïx algunes hipòtesis sobre els substrats neurals de l'aprenentatge i la memòria [9] que han estat verificades, posteriorment, per Bliss i Lomo, a l'any 1973 [2].

Si pensem en una tanda continuada d'experiments (passos) i que a cada experiment (pas) apareix un patró on el concepte figura en forma de "paquet" d'informació amb un númer de codi, podem utilitzar la taula 1 per aproximari-nos al fenòmen. En aquest cas no caldrà cap restricció adicional; la seqüència aleatòria del cas genèric és suficient.

La funció expressada per (1) modella en forma minimal un tipus de comportament essencial compartit per diversos sistemes reals, com és la saturació. Aspectes com la...
imposició d’un primer element, la necessitat d’un ordre en la seqüència o la probabilitat d’aparició variable dels elements són particularitats dels sistemes, no atribuïbles a l’operació, i, per tant, no crucials. L’important és “qué” s’aconsegueix i no “com” s’ha aconseguit.

3. Conclusions

- La caracterització funcional sembla un mètode prometedor per a modelar principis organitzacionals del processament de informació, perquè permet una formulació genèrica, sense cap referència a les propietats de la matèria ni a la estructura dels dispositius. Els elements canònics del model, o primitius, s’inspiren en els comportaments macroscòpics observats en els sistemes reals i es formalitzen mitjançant funcions recursives definides en els domini discrets J. S’ha presentat un exemple de primitiu, la saturació, que és un efecte observat habitualment en sistemes computacionals.

- La saturació pot modelar-se a partir d’una suma en la qual tan sols contribueix el major dels sumands i els exemples esmentats posen de manifest el caràcter universal de l’operació front a les particularitats dels sistemes, representades per restriccions de caràcter estatístic que recauen sobre la seqüència d’entrada. A més de la idoneïtat per a modelar l’emergència, la possibilitat d’implementar portes funcionals basades en aquestes primitius pareix atraient des de la perspectiva de la tolerància a l’error. Així, una seqüència d’elements positius, esquixada per un percentatge d’elements negatius podria saturar, fins i tot. Dos aspectes interessants del model són la influència qualitativa del cardinal de J sobre el caràcter brusca o suau d’un fenòmen, i la incidència, només relativa, de la manca de propietat associativa: importa el tipus d’associació únicament si ens interessem els resultats pas per pas; però no si considerem el resultat global. Paral·lelament, s’ha treballat en la línia de trobar un recolzament matemàtic que sistematitze la generació automàtica de primitius. Apostem per la convolució de funcions reals com a eina capaç de proporcionar-ne una descripció formal. Fins ara s’han obtingut bons resultats en els casos experimentals, realitzant la convolució d’una funció f per una funció \( g(n)=K^n \): fenòmens com la saturació, la divergència o l’existència d’oscil·lacions estan vinculades amb les característiques de K: si és positiu o negatiu, o si és major o menor que la unitat en valor absolut.

Referències


Aspectes formals

Defining and combining Multiple-valued Logics for Knowledge-based Systems

José Antonio Reyes  Francesc Esteva  Josep Puyol-Gruart
Artificial Intelligence Research Institute, IIIA
Spanish Council for Scientific Research, CSIC
Campus UAB. 08193 Bellaterra, Catalonia, Spain
E-mail: {reyes, esteva, puyol}@iiia.csic.es

Abstract

Multiple-valued logics are useful for dealing with uncertainty and imprecision in knowledge-based systems. In this paper, we present a tool that assist users in the declaration of such logics and in the declaration of the communication mechanism between two of these different logics by preserving inference.

Keywords: uncertain reasoning, multiple-valued logics, knowledge-based systems.

1 Introduction

The management of uncertainty and imprecision in knowledge-based systems becomes essential to model many real problems. Multiple-valued logics (MV-Logics) have been proved to be a possible way to manage them in knowledge-based systems [3; 5; 6; 10].

The aim of this work is concerned with presenting a tool which makes automatic the process of defining and communicating finite MV-Logics. This tool has been designed to be incorporated to the shell Milord If [7], an environment for developing knowledge-based systems, although it can be used in a more extensive framework. In section 2, we will introduce the concept of algebra of truth-values which defines a parametric family of MV-logics, as well as the syntax and the semantic permitted. Next, the extension of that algebra to an algebra of intervals of truth-values is described as a method for dealing with imprecision. Section 3 is devoted to establish the essential requirements needed to preserve inference in different ways when we communicate different logics. We give existence conditions for every requirement and identify different sorts of renaming functions between logics. In section 4, we describe the tool and its main features by means of an example. Finally, in section 5 we outline the conclusions of this work.

2 Defining MV-Logics

In this section we will see which features are necessary to define a suitable MV-Logic. Next, we will show how this process has been automated and which are the achieved results.

We consider a restricted family of finite MV-logics expressive enough to model the uncertain reasoning used in many rule-based systems [1]. Each logic is determined by a particular algebra of truth-values from this family.

An Algebra of truth values is a finite algebra \( A_f^a = \{ A_n, \theta, I, N_n, T, I_f \} \) such that:

1. The ordered set of truth-values \( A_n \) is a chain of \( n \) elements: \( \theta = a_0 < a_1 < \ldots < a_{n-1} = I \)

   where \( \theta \) and \( I \) are the Boolean False and True respectively, and each element \( a_i \) is a linguistic term. For example \( A_5 = \{ false, unlikely, may_be, likely, true \} \).

2. The negation operator \( N_n \) is the unary operation defined as

   \[ N_n(a_i) = a_{n-1-i} \quad (1) \]

   the only one that fulfills the following properties:

   \( N_1 : \) If \( a < b \) then \( N_n(a) > N_n(b), \forall a, b \in A_n \).

   \( N_2 : \) \( N_n^2 = Id. \)

3. The conjunction operation \( T \) is any binary operation such that the following properties hold \( \forall a, b, c \in A_n \):

   \( T_1 : T(a,b) = T(b,a) \)

   \( T_2 : T(a, T(b,c)) = T(T(a, b), c) \)

   \( T_3 : T(\theta, a) = \theta \)

   \( T_4 : T(I, a) = a \)

   \( T_5 : \) If \( a \leq b \) then \( T(a, c) \leq T(b, c) \) for all \( c \)

See an example of conjunction operator in Table 1.
**Table 1:** $T_2$ conjunction operation.

4. The implication operation $I_T$ is defined by residuation with respect to $T$, i.e.

$$I_T(a, b) = \text{Max} \{ c \in A_n \mid T(a, c) \leq b \}$$

(2)

Therefore, to establish an algebra of truth-values it is only necessary to determine the set of truth-values $A_n$ more adequate for the concerning problem, and define a conjunction operator $T$ necessary to combine and propagate uncertainty when making inference.

Forthwith, we introduce the syntax of our MV-Logic. The sentences of the language are pairs $(p, V)$, where:

- $p$ is a well formed-formula, obtained in the usual way from a denumerable set of propositional symbols and
- the connectives "not" ($\neg$), "and" ($\wedge$), and "detachment" ($\rightarrow$); and
- $V$ is an interval of truth-values.

Likewise, semantic is evaluated in an algebra of truth-values The semantic interpretation is given by the followings signification:

- **Models** are defined by valuations, i.e. mappings $\rho$ from the first components of sentences (propositional symbols) to $A_n$ provided that:

  - $\rho(\neg p) = N_n(\rho(p))$
  - $\rho(p_1 \wedge p_2) = T(\rho(p_1), \rho(p_2))$
  - $\rho(p \rightarrow q) = I_T(\rho(p), \rho(q))$

- The **Satisfaction Relation** between models and sentences is defined by:

  $$M_\rho \models (p, V) \text{ if, and only if, } \rho(p) \in V,$$

  where $M_\rho$ stands for the model defined by a valuation $\rho$.

Therefore, from now on, it is required that the logic systems used by applications (axioms and inference rules) must be sound with respect to the semantic above defined, in order to be considered as an appropriate MV-Logic. The inference rule of these logics is Modus Ponens, that gives from $(p \rightarrow q, V)$ and $(p, V')$ a sentence $(q, V'')$. In fact, modus ponens evaluates $V''$ from $V$ and $V'$. (See [9] for a theoretical study of modus ponens function and [6] for its application to expert systems).

After the definition of algebra of truth-values we extend it to an algebra of intervals of truth-values, [4]. We have some motivations to do this extension. The first one is given as consequence of imprecision, because we do not know a precise value but an interval of truth-values. A second reason is related with the modus ponens operator which requires the use of intervals to chain rules. And as we will see in the next section, it is needed to make possible mappings between different logics.

Given an algebra of truth-values $A = \langle A_n, N_n, T \rangle$, we will consider the set of intervals of $A_n$ as:

$$\text{Int}(A_n) = \{ [a, b] \mid a, b \in A_n \}$$

(3)

being $[a, b] = \{ x \in A_n \mid a \leq x \leq b \}$. As an example, the set of intervals of a chain $A_4 = \{ 0 < a < b < 1 \}$ is $\text{Int}(A_4) = \{ [0, a], [0, b], [0, 1], [a, b], [a, 1], [b, 1], [0, 0], [a, a], [b, b], [1, 1] \}$.

In the same way, we can think about extend also the algebra operators for dealing appropriately with intervals. The extensions to intervals of the above operators are:

$$N_n(\langle a, b \rangle) = [N_n(a), N_n(b)], T^\ast(\langle a, b \rangle, \langle c, d \rangle) = (T(a, c), T(b, d)).$$

### 2.1 Conjunction Generation

The search for a conjunction operator is an important task in an algebra declaration as we can deduce from the definition of algebra of truth-values. Next we shall discuss how this question has been treated and how this process has been automated.

Throughout this work we represent the conjunction operator as a matrix, where each element is a variable. Then, for a matrix $M$ of dimension $n$, we will have:

$$M = \begin{bmatrix}
V_{0,0} & V_{0,1} & \cdots & V_{0,n-1} \\
V_{1,0} & V_{1,1} & \cdots & V_{1,n-1} \\
\vdots & \vdots & \ddots & \vdots \\
V_{n-1,0} & V_{n-1,1} & \cdots & V_{n-1,n-1}
\end{bmatrix}$$

(4)

The mentioned algebra requirements (T1-T5) are properties that a conjunction operator must fit. We will try to assign agreeable values for the variables according to these properties.

These requirements act as constraints over the set of possible solutions. Satisfaction of each property or constraint, causes the following guidelines which influence the conjunction search generation problem:

- **Commutativity** allows us to consider only the set of variables $V = \{ V_i, j, i \geq j \}$.

- **Existence of absorbent and neutral elements** implies to fix the values for variables $V_{0,j}$ and $V_{i,n-1}$.
• Monotonicity implies rows and columns non-decreasing.
• Associativity test is expensive in time and memory, but taking into account that \( T(a_i, a_j) \leq \min(a_i, a_j) \) and so, each sub-matrix

\[
M_{i,j} = \begin{bmatrix}
V_{0,0} & \cdots & V_{0,i} \\
\vdots & \ddots & \vdots \\
V_{i,0} & \cdots & V_{i,i}
\end{bmatrix}
\]

is closed with respect to the conjunction operator. In this way, the full matrix will be associative if all its sub-matrices are so. This property lets us check associativity in an incremental way each time that a value is assigned to a variable of the kind of \( V_{i,j} \).

However, the number of obtained operators applying these constraints is significantly high. In particular for \( n > 5 \) (see column A in Table 2), we obtain more than 22 matrices. This is why we can consider two new options: introducing some new desirable constraints, as well as knowing several values of the matrix to reduce the search space to be explored.

Two types of additional, and optional, constraints are discussed. First, we can think that given two truth-values different than 0 (false), it does not seem reasonable that their conjunction may be 0. This is achieved by the property:

**T6 Strictness:** \( T(a_i, a_j) \neq 0 \), for all \( i, j \neq 0 \)

In the infinite case some continuity is required. In the finite case we also want that the result of the conjunction of near values has to be not very distant. This property is defined as follows:

**T7 α -Smoothness:** given \( \alpha \in \mathbb{N} \), \( T \) is said to satisfy \( \alpha \)-smoothness property if \( T(a_i, a_j) = a_k \) and \( T(a_{i+p}, a_j) = a_p \), then \( k \cdot p \leq \alpha \)

Satisfaction of these new properties (T6,T7) have the following consequences in the conjunction search generation problem:

• Strictness: the generation of strict matrices of dimension \( n \) is equivalent to generate non-strict matrices of dimension \( n-1 \).
• \( \alpha \)-Smoothness: it reduces the set of possible values for a variable taking into account column and row adjacency.

Now we consider: general constraints (T1-T5), which provide standard behavior of a conjunction operator; and additional constraints (T6,T7), that add desirable properties to a conjunction operator and decrease the number of possible solutions.

As an additional option we can pre-settled some variables values for the conjunction matrix, so it is possible to settle the more adequate behavior to the concerning problem.

Since now, we have stated the problem of define an adequate conjunction operator. In the follow, we will present how this problem had been solved.

### 2.2 Generation as a CSP Problem

There exist some options when we define a conjunction operator: it is possible to give a full matrix that will be checked to assure it as a conjunction operator; it is also possible to give a partial matrix with some pre-settled values that will be whole completed; and finally, if it is required by generating all the matrices.

The generation of operators that satisfy a determinate set of properties can be formulated as a classic CSP (Constraint Satisfaction Problem):

• Let \( V = \{V_0, \ldots, V_n\} \) be a set of variables.
• Let \( D = \{D_0, \ldots, D_n\} \) be a set of the corresponding discrete and finite domains (in general \( D \neq D_j \) if \( i \neq j \)), where variables take values.
• Let \( C = \{C_{i,j} \mid 0 \leq i,j \leq m-1 \} \) be a set of binary constraints. These constraints express the relationship that the different variables must hold among them for their values to be compatible.

The problem consists of finding a value assignment \( \{V_0, V_1, \ldots, V_{n-1}\} \) such that for every \( 0 \leq i,j \leq m-1 \), \( C_{i,j} \) holds. Then, one may be interested in finding all the solutions to the problem, or just one, or the total number of solutions or if there really exists a solution.

From the concluded search influences analyzing the requirements (T1-T7) and the additional option of pre-settled some values, the CSP problem can be reformulated according to these new domains:

• Initially, we have a set of variables with settled values. This set is composed by the values concerning the variables: \( V_{0,j} = \emptyset \), \( V_{i,n+1} = a_j \) (T3,T4) and the ones pre-settled by the user.
• We only take into account the set of variables \( V = \{V_{i,j} \mid i \geq j \text{ with } i,j \neq 0 \text{ and } i,j \neq n-1\} \) (T1).
• The initial domain for the considered variables is \( D_{i,j} = \{a_0, \ldots, a_l\} \) (T5).
• This initial domain can be modified according to strictness and/or \( \alpha \)-smoothness properties (T6,T7).

If strictness is applied, then the \( a_0 \) value is rejected. If \( \alpha \)-smoothness is applied, the domain length will have a maximum of \( \alpha + 1 \) elements, keeping the last \( \alpha + 1 \) elements of the set if this number is greater.
We solved this search generation problem with a depth-first algorithm applying backtracking and lookahead. Backtracking needs two stacks: one to keep the generated matrices and the other to keep the values that remains unassigned for each variable. The algorithm is briefly shown below, and the generated solution results, taking into account T1-T7, are shown in Table 2.

<table>
<thead>
<tr>
<th>n</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>6</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>94</td>
<td>22</td>
<td>38</td>
<td>78</td>
</tr>
<tr>
<td>7</td>
<td>451</td>
<td>94</td>
<td>118</td>
<td>306</td>
</tr>
<tr>
<td>8</td>
<td>2386</td>
<td>451</td>
<td>395</td>
<td>1274</td>
</tr>
<tr>
<td>9</td>
<td>2386</td>
<td>1404</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Matrices per number of terms.

A : T1-T5  
B : T1-T6  
C : T1-T5 and T7  
D : T1-T7

1) Initialization:
The variables with settled values are assigned and the rest will have an undefined value.
The process order for the variables is defined.
The stacks are empty.

2) Variable selection:
The current matrix is pushed upon the matrices stack.
A variable is chosen as the current one.
If it is a pre-settled value, go to step 2; otherwise all its possible values are selected.
If there are no more variables, go to step 5.

3) Value selection:
The current variable takes on a given value.
The rest of possible values are pushed upon the values stack.

4) Test:
Check the satisfaction of all the constraints on the selected variable.
If all these constraints are satisfied, go to step 2; otherwise go to step 3.

5) Solution:
Each assignment passing the test is regarded as a solution.

6) Backtracking:
The current matrix is popped out of the matrix stack.
The current variable is popped out of the other stack with all unassigned values.
If there are unassigned values left, go back to step 2 with each one.
If the stacks are empty, then stop.

Now we know how to define a logic and how we suggest to solve and automate this process. In the next section we will see how to communicate different logics and how to preserve certain inference in this process.

3 Combining Logics

We can declare different logics by varying the set of truth-values (linguistic terms) and the conjunction operator. That depends on how the expert will deal with uncertainty in each problem.

When two different logics need to exchange information, it is necessary some mechanism of translation of the linguistic terms to make the communication between those logics compatible. At such communication, we want to preserve the inference of each logic, but as occurs in real life, it is not always possible to transmit information with absolutely precision. This loss of information can be represented with an interval, and as a consequence, we need to extend the algebra of truth-values to an algebra of intervals of truth-values. Hence we map values of an algebra into intervals of the other.

Let’s take a look to which requirements we need in order to map the language of a logic into another one when they require to exchange information [2].

3.1 Mappings between MV-Logics

Let \((L, \vdash L)\) and \((L', \vdash L')\) be two logics, \(L\) and \(L'\) standing for the languages, \(\vdash\) and \(\vdash\) for the entailment relations defined on \(L\) and \(L'\) respectively. To establish a correspondence between both logics, a mapping \(H : L \rightarrow L'\) is needed. Next, we will analyze some natural requirements for the mapping \(H\) with respect to the entailment systems \(\vdash\) and \(\vdash\). We propose that at least one of the following three requirements should be fulfilled by the mapping \(H\) in order to ensure a consistent communication.

Henceforth \(\Gamma\) and \(e\) will denote a set of sentences and a sentence of \(L\) respectively. A map \(H\) is said to be a forward conservative map when,

\[ RQ.1. \text{ If } \Gamma \vdash e \text{, then } H(\Gamma) \vdash' H(e) \]
Aspects formals

For every sentence \( e \), deducible from a set of sentences \( \Gamma \), its corresponding sentence, \( H(e) \), will also be deducible from the corresponding sentences of \( H(\Gamma) \).

A map fulfilling this second requirement is said to be a **backward conservative map**:

**RQ.2**. If \( H(\Gamma) \vdash H(e) \), then \( \Gamma \vdash e \)

This is the inverse requirement of **RQ.1**. Hence, if a fact is not deducible from \( \Gamma \), then its corresponding fact from \( H(\Gamma) \) won't be deducible either. Nevertheless \( \Gamma \vdash e \) does not imply \( H(\Gamma) \vdash H(e) \).

Conditions **RQ.1** and **RQ.2**, which are very strong, can sometimes be weakened in the uncertainty reasoning framework. Formally, this can be expressed by the third and last requirement:

**RQ.3**. If \( H(\Gamma) \vdash e' \), then there exists \( e \) such that \( \Gamma \vdash e \) and \( H(e) \vdash e' \)

This requirement assures that every sentence deducible from \( H(\Gamma) \) must be in agreement with what can be deduced from \( \Gamma \). This requirement is slightly different from **RQ.2**, in the sense that it is not necessary that \( e' \) is an exact translation of a deducible sentence \( e \) from \( \Gamma \), but only something deducible from such a translation. In the framework of logics for uncertainty management, \( e' \) is interpreted as a weaker form of \( e \), i.e. a sentence expressing more uncertainty than \( e \). We will call it a weak conservative map.

Now we will consider the problem of finding inference preserving correspondences between two logics \( A = \langle A_n, N_n, T, > \rangle \) and \( B = \langle B_m, N_m, T' \rangle \). We are interested in mapping the entailment system \( (L_A, \Gamma_A) \) into the entailment system \( (L_B, \Gamma_B) \), by means of renaming functions between the corresponding linguistic term sets. This means that we will only consider those mappings translating sentences from \( L_A \) to \( L_B \) that just involve translations of truth-values, i.e. any mapping \( H : L_A \rightarrow L_B \) will be defined as \( H(e, V) = (e, h(V)) \), where \( h \) translates subsets of values of \( A_n \) into subsets of values of \( B_m \) and \( e \) (a well-formed formula) remains invariant.

### 3.2 Algebra Morphisms

As it has already been noted, to establish conservative communications, it is necessary to consider what kind of relation between the uncertainty logics is required. In [2], we can find the necessary and/or sufficient conditions for a mapping \( h : A_n \rightarrow Int(B_m) \) to satisfy these requirements. From this analysis, we deduce the following relationships between the conditions needed to satisfy every requirement (see Figure 2):

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Conditions</th>
</tr>
</thead>
</table>
| **RQ.1** \( \Leftrightarrow \) | \[
\exists h(V), h(V2) \quad h(V1) \supset h(V2) \\
\forall h(N(V)) \neq h'(N(V)), h(V) \\
h(V1) \supset h(V2) \Rightarrow V1 \supset V2
\]
| **RQ.2** \( \Leftrightarrow \) | \[
\exists h(V), h(V2) \quad h(V1) \supset h(V2) \\
\forall h(N(V)) \neq h'(N(V)), h(V) \\
h(V1) \supset h(V2) \Rightarrow V1 \supset V2
\]
| **RQ.3** \( \Leftrightarrow \) | \[
\exists h(V), h(V2) \quad h(V1) \supset h(V2) \\
\forall h(N(V)) \neq h'(N(V)), h(V) \\
h(V1) \supset h(V2) \Rightarrow V1 \supset V2
\]

**Figure 2**: Requirements conditions.

We define mappings from elements of \( A_n \) into intervals of \( B_m \), but sometimes it is possible to find mappings that translate an element of \( A_n \) into an interval \( [b_i, b_j] \) of \( B_m \) (it can be clearly consider as the element \( b_j \)). In this case, requirement **RQ.3** is satisfied and the mapping is a morpshism between the corresponding algebras. As a particular case, if a map fulfill **RQ.1** and **RQ.2**, we have not only the morphism conditions, but a one-to-one application, that is, an injective function called monomorphism. But we can not always find these kind of mappings, so in the case of a map involving intervals of truth-values, we named it a quasi-morphism.

We are mainly interested in monomorphisms because they embed \( A_n \) into \( B_m \) and because they are order preserving mappings (it is equivalent to a communication without any loose of information). Then, we are interested in morphisms, which accomplish the algebra operations (it is a transmission of information fulfilling the required properties). Finally, because of the strong conditions morphisms and monomorphisms must satisfy, it is not always possible to find these kind of renaming functions, quasi-morphisms can be useful thanks to the additional freedom of map truth-values of an algebra into intervals of the other (we allow certain loose of information).

### 3.3 Renaming Algorithm

Before seeing the process to find these algebra morphisms, let us part from the previously well-known negation operation. For a given set of truth-values \( A_n \), there exists only one negation \( N_n \) and it is defined by \( N_n(a_n) = a_{n+1} \). Then, we can partition this set into three subsets:

- the set of negative elements \( N_n = \{ x \mid x < N_n(x) \} \)
- the set of fixed elements \( F_n = \{ x \mid x = N_n(x) \} \)
- the set of positive elements \( P_n = \{ x \mid x > N_n(x) \} \)
being the subsets: \( F_n = \{ a_i \mid i < k \} \), \( N_n = \{ a_i \mid i < k \} \), \( F_n = \{ a_i \mid i > k \} \), \( N_n = \{ a_i \mid i > k \} \), and \( F_n = \{ a_i \mid i = k \} \), \( N_n = \{ a_i \mid i = k \} \). And the renaming algorithm is as follows:

1) **Initialization**:

Obtain the subsets \( N_n, F_n, N_n^* \), and \( F_n^* \).

2) **Maps Generation**:

Generate all the maps \( h_1: N_n \cup F_n \rightarrow N_m \cup F_m \) such that:

a) \( h_1(0) = \emptyset \).

b) \( h_1(F_n) \in F_m^* \).

c) \( x \leq y \) implies \( h_1(x) \nleq h_1(y) \), where \( x, y \in N_n \) and \( h_1(x), h_1(y) \in \text{Int}(B_m) \).

3) **Map extension**:

Extend each mapping \( h_1 \) with respect to the negation operation defining the morphism \( h \) as:

\[
h(x) = \begin{cases} 
    h_1(x), & \text{if } x \in N_n \cup F_n, \\
    N_m^*(h_1(N_n)(x)), & \text{if } x \in F_n.
\end{cases}
\]

4) **Conjunction checking**:

Check which ones are compatible with the conjunction operators \( T \) and \( T' \).

5) **Renaming checking**:

Finally, we check which maps \( h \) are morphisms, monomorphisms or quasi-morphism.

As an example, consider two logics declared with the sets of truth-values \( A_2 \) and \( B_3 \), and with the conjunction operations \( T_{A_2} \) and \( T_{B_3} \) (defined in Tables 1 and 3 respectively).

\[
A_2 = \{ \text{impos, few}_p, \text{pili}_p, \text{possib}, \text{quite}_p, \text{very}_p, \text{sure} \} \\
B_3 = \{ \text{false, unlikely, may_be, likely, true} \}
\]

<table>
<thead>
<tr>
<th>impos</th>
<th>few_p</th>
<th>pili_p</th>
<th>possib</th>
<th>quite_p</th>
<th>very_p</th>
<th>sure</th>
</tr>
</thead>
<tbody>
<tr>
<td>impos</td>
<td>impos</td>
<td>impos</td>
<td>impos</td>
<td>impos</td>
<td>impos</td>
<td>impos</td>
</tr>
<tr>
<td>few_p</td>
<td>few_p</td>
<td>few_p</td>
<td>few_p</td>
<td>few_p</td>
<td>few_p</td>
<td>few_p</td>
</tr>
<tr>
<td>pili_p</td>
<td>pili_p</td>
<td>pili_p</td>
<td>pili_p</td>
<td>pili_p</td>
<td>pili_p</td>
<td>pili_p</td>
</tr>
<tr>
<td>possib</td>
<td>possib</td>
<td>possib</td>
<td>possib</td>
<td>possib</td>
<td>possib</td>
<td>possib</td>
</tr>
<tr>
<td>quite_p</td>
<td>quite_p</td>
<td>quite_p</td>
<td>quite_p</td>
<td>quite_p</td>
<td>quite_p</td>
<td>quite_p</td>
</tr>
<tr>
<td>very_p</td>
<td>very_p</td>
<td>very_p</td>
<td>very_p</td>
<td>very_p</td>
<td>very_p</td>
<td>very_p</td>
</tr>
<tr>
<td>sure</td>
<td>sure</td>
<td>sure</td>
<td>sure</td>
<td>sure</td>
<td>sure</td>
<td>sure</td>
</tr>
</tbody>
</table>

\[ T_{B_3} \] conjunction operation.

First, we obtain the initial subsets \( N_0, F_0, N_0^* \) and \( F_0^* \):

\[
N_0 = \{ \text{false, unlikely} \} \\
F_0 = \{ \text{may_be} \} \\
N_0^* = \{ \text{impos, few}_p, \text{pili}_p, \text{possib}, \text{impos, few}_p, \text{impos, pili}_p, \text{impos, possib}, \{ \text{few}_p, \text{pili}_p \}, \{ \text{few}_p, \text{pili}_p, \text{possib} \}, \{ \text{pili}_p, \text{possib} \} \} \\
F_0^* = \{ \text{possib}, \{ \text{few}_p, \text{very}_p \}, \{ \text{pili}_p, \text{quite}_p \}, \{ \text{impos, sure} \} \}
\]

And following the above algorithm schema, we can find several mappings between both logics. There are here two mappings that are examples of those that hold the last requirement RQ.3:

\[
\begin{align*}
\text{false} & \rightarrow \text{impos} \\
\text{unlikely} & \rightarrow \{ \text{impos, few}_p \} \\
\text{may_be} & \rightarrow \{ \text{few}_p, \text{pili}_p \} \\
\text{likely} & \rightarrow \{ \text{very}_p, \text{sure} \} \\
\text{true} & \rightarrow \text{sure}
\end{align*}
\]

When ending the generation process, the list of inference preserving mappings is presented to the user in this way:

1) First, we offer the existent monomorphisms considered as one-to-one morphisms.

2) Next, we show which morphisms are between both algebras, if any.

3) Finally, we display the list of generated quasi-morphisms.

Due to the strong conditions monomorphisms and morphisms must fit, it is not always possible to find them. However, it is very possible that the renaming generation produces a large list of quasi-morphisms, so the possibility of giving an ordered list of quasi-morphisms to aid users in their selection may be considered. Note that in the case of morphisms, all the renaming mappings have the same evaluation, hence the selection is left to the user’s criteria.

For the purpose of producing an ordered list of quasi-morphisms, we consider a weigh among the cardinal \( c \) of the set of terms which have an atomic image (that is, an interval with the form \( \{ a_i, a_j \} \) and the length \( L \) of the remainder of intervals (number of truth-values included in this interval).

Given two chains \( A_n \) and \( B_m \), we will generate mappings between \( A_n \) and the set of intervals \( \text{Int}(B_m) \). In order to obtain an evaluation for every map, we can use the following empirical function:

\[
\eta = \frac{\sum_{i=1}^{n} L_i}{\sum_{i=1}^{n} c_i}, \quad 1 \leq L_i \leq m, \quad 1 \leq c_i \leq n
\]

\[
\sum_{i=1}^{n} c_i
\]
where \( L_i \) is the length of the interval \( i \) and \( c_i \) is the number of points which have an atomic image. Then, for every mapping we obtain a value \( \eta \), such that:

\[
1 \leq \eta \leq \frac{(n-2)m+2}{2}
\]

(7)

This method produces an ordered list of quasi-morphisms that let user to simplify the selection. User is also able to establish the behavior that can take the generated maps defining a partial map between both algebras. This definition must satisfy the set of requirements suggested previously in order to be a negation morphism. That is, our partial map must accomplish the next points:

1) It must be a non-decreasing function.

2) If our map involves the truth-value \( a_0 \), its image must be the truth-value \( b_0 \). For instance, in the above example any map must fit \( h(false) = impossible \). Therefore, to be a negation morphism, it is the same for the case of truth-value \( a_0 \), so it is required that \( h(true) = sure \).

3) If our map implies the fixed element of the first chain \( A \), we must give as its image an element belonging to the set of fixed elements \( F \). In the above example, a map of this type must fit \( h(\text{may\_be}) = \{\text{impos, few\_p, sli\_p, possib, [impos, few\_p, [impos, sli\_p, [impos, possib], [few\_p, sli\_p, [few\_p, possib], [sli\_p, possib]]}\right)

When we have defined our partial map (or total, or none), a list of renaming functions fulfilling these defined characteristics will be generated.

Now we know how to declare a MV-logic and how to establish a conservative communication between two of these logics. In the following section we will present the interactive implemented tool to define and combine MV-logics.

4 The QMORPH Tool

QMORPH [8] is a tool that allows users to define and combine MV-logics. Two different interfaces have been developed: a graphical interface for the UNIX operative system using Tcl/Tk packages, running under the X-Windows environment and developed on Sun machines; and a generic text mode interface performed in Common Lisp.

In the present, this tool is attached to Milord II, a specific expert system building environment, as an aid tool to assist experts when developing modular applications.

In this section, we illustrate the use of QMORPH throughout the use of a practical example in which we try to find existing morphisms between two MV-logics (as it include the particular case of a logic definition). For this purpose, we will make use of the most friendly graphical interface.

We consider the same example of section 3.3. Let us suppose we need to establish a communication between the logic \( A = \langle A_0, N_0, T_0 \rangle \) (belonging to the Origin Module or O-Module) and \( B = \langle B_0, N_0, T_0 \rangle \) (belonging to the Image Module or I-Module). This schema is shown if Figure 3.

![Figure 3: Communication example.](image)

Initially, it is mandatory to declare the truth-value sets of Origin and Image modules. Besides, user has to declare the \( T \) operation of the Origin Module for avoiding a computational explosion.

First of all, user must fix the sets of linguistic terms that settled both logics. For this purpose, it is necessary to decide how many linguistic terms have the logics (see Figure 4). Following our example, we must enter 5 and 7 terms for the Origin and Image logic respectively.

![Figure 4: Entering Algebra's Dimensions.](image)

Next step consists in introducing the name of the terms for each logic see (Figure 5). If user does not consider necessary to name the terms, they will be generated automatically. To declare the logics \( A \) and \( B \), we may enter here \( \{false, unlikely, may\_be, likely, true\} \) and \( \{impos, few\_p, sli\_p, possib, quite\_p, very\_p, sure\} \) respectively.

![Figure 5: Entering Truth-Values Sets.](image)

Following, it is needed to declare the conjunction operator that \( A \) uses to combine and propagate the linguistic terms when making inference. User can choose a \( T \) function depending on the meaning he
wants give to the conjunction operation. There are some matrix values that remain fixed (by properties: $T(0, a) = 0$ and $T(1, a) = a$), so these values will be presented automatically avoiding any changes. There are some possibilities here: user can give a full operator, that will be checked to assure it as a conjunction operator; it is also possible to settle the behavior of the operator and give a partial matrix with some pre-settled values; and finally, there is the possibility of not giving any value, so all the conjunction operators would be generated. To restrict the number of possible solutions, user can apply strictness and $\alpha$-smoothness constraints over the process generation. In our example (see Figure 6), we require that $T_2(\text{unlikely}, \text{unlikely}) = \text{unlikely}$.

![Figure 6: Entry O-Module AND Truth-Table.](image)

If many matrices have been generated, user must choose one of them. If user consider that there are too many matrices, it is possible to come back to redefine the matrix and/or the restrictions over it. In our example, 5 matrices have been generated (Figure 7).

![Figure 7: Number of generated matrices.](image)

To choose among the set of generated matrices, results are presented in the Matrices Reproducer (Figure 8) where we can display all the generated matrices and select the one that fits better.

![Figure 8: Browsing a conjunction operator.](image)

Once the logic of the A (O-Module) has been determined, we must define the logic concerning to B (I-Module).

The procedure for defining an I-Module conjunction operator is exactly the same that in the case of the O-Module, but it is not necessary to choose one of them. Then, for every matrix solution, all the renaming mappings between both defined algebras will be generated. It is the possible to give a partial map between both algebras. If user desire a particular truth-value from one logic to be translated to another truth-value (or to an interval), the values of the partial map will be checked to be correct (see Figure 9). It is permitted to choose a determinate criteria to be satisfied during renaming generation. There are three criteria: A, B and C, corresponding to requirements RQ.1, RQ.2 and RQ.3, respectively.

![Figure 9: Defining a Renaming Mapping.](image)

When all mappings between the two algebras have been generated, for every I-Module operation, the possible monomorphisms and morphisms, if any, and an ordered list with all the quasi-morphisms is presented.
5 Conclusions

In this paper we have defined which are the theoretical bases that allow to deal with uncertainty by means of multiple-valued logics. We have settled how to declare a suitable MV-logic from a parametric family of algebra of truth-values, and also, which are the necessary and/or sufficient requirements when we need to establish a communication between two of these logics. This problem arises in large knowledge-based systems in which different tasks need to cooperate using uncertain reasoning, as well as in distributed systems.

We have presented the developed algorithms to aid users in the generation of conjunction operations (implemented as a classic CSP) and in the search of inference preserving mappings.

The performed tool has been designed with the aim of automating these problems, as well as assisting users and offering different alternative possibilities. Beyond its particular use within Milord II, it can be deployed by any other system using MV-logics.

This work focuses on the unidirectional interaction between two logics, but the more general problem of communicating various uncertainty reasoning systems is far more complex. This work along with further research will make possible this goal.

Acknowledgments

Partially supported by project MODELOGOS funded by CICYT (TIC 97-0579-C02-01) and through SMASH by CICYT (TIC 96-1038-C04-01).

References


Phase Transitions in the Regular Random 3-SAT Problem*

Ramón Béjar and Felip Manyà

Departament d'Informàtica
Universitat de Lleida
Jaume II, 69, E-25001. Lleida, Spain
{ramon,felip}@eup.udl.es

Abstract

In this paper we investigate phase transitions in the random 3-SAT problem but we move from the usual setting of classical logic to the more general setting of multiple-valued logics. We deal with regular CNF formulas and use a generalized Davis-Putnam (DP) procedure for testing their satisfiability. We show experimentally that the location of the threshold increases logarithmically as a function of the cardinality of the truth value set and also provide a theoretical explanation of such fact. The DP procedure and the classical random 3-SAT problem appear to be a particular case of our approach.

1 Introduction

Phase transitions occur in the propositional satisfiability (SAT) problem (i.e. the problem of determining whether a classical propositional formula is satisfiable), as well as in other NP-hard problems [1]. Concerning the SAT problem, investigations have largely concentrated on the random 3-SAT problem (i.e. the SAT problem restricted to formulas whose clauses have three literals and are randomly generated).

Mitchell et al. [15] reported results from experiments on testing the satisfiability of random 3-SAT instances with the Davis-Putnam (DP) procedure [4]. They observed that (i) there is a sharp transition from satisfiable to unsatisfiable instances at a threshold of the ratio of the number of clauses to variables in such a way that about 50% of the instances around this value are satisfiable, below this value the probability of being satisfiable approaches 1 as the ratio decreases and above this value the probability of being satisfiable approaches 0 as the ratio increases; and (ii) there is an easy-hard-easy pattern in the computational difficulty of solving problem instances as that ratio is varied; the hard instances occur in the area near the threshold. Similar results about this phenomenon are described in [3]. Nowadays, there is strong experimental evidence that the value of the threshold is around 4.24, but there are no analytical results. Only lower and upper bounds on the location of the threshold are known [2, 12].

In this paper we investigate phase transitions in the random 3-SAT problem but we move from the usual setting of classical logic to the more general setting of multiple-valued logics. We deal with regular CNF formulas and use a generalized DP procedure for testing their satisfiability. We show experimentally that the location of the threshold increases logarithmically as a function of the cardinality of the truth value set and also provide a theoretical explanation of this fact. The DP procedure and the classical random 3-SAT problem appear to be a particular case of our approach.

Regular CNF formulas are a relevant subclass of signed CNF formulas (cf. Section 2). It turns out that any finitely-valued formula can be transformed into a satisfiability equivalent signed CNF formula in polynomial time [7]. Interestingly, Hähnle [6, 7] identified a wide and important class of finitely-valued logics, so-called regular logics, and showed that any formula of one of such logics can be trans-
formed into a satisfiability equivalent regular CNF formula whose length is linear in both the length of the transformed formula and the cardinality of the truth value set. Recently, Hähnle [9] has found a method for translating a signed CNF formula into a satisfiability equivalent regular CNF formula where the length of the latter is polynomial in the length of the former. Thus, a SAT problem in any finitely-valued logic is polynomially reducible to a SAT problem over regular CNF formulas.

A generalized DP procedure (hereafter called the regular DP procedure) for testing the satisfiability of regular CNF formulas was presented in [8]. In [14], we equipped such procedure with suitable data structures and showed experimentally that phase transitions occur when we run the procedure on regular random 3-SAT instances. There, we only considered truth values sets with cardinality 3, 4 and 5. The thresholds we obtained for such cardinalities were 5.92, 7.08 and 7.75, respectively. In view of these results, one could be tempted to conjecture that the location of the threshold increases linearly as a function of the cardinality of the truth value set. However, in this paper we will provide experimental evidence that such increase is logarithmic. Moreover, we will give a theoretical explanation of this fact by providing upper bounds on the location of the unsatisfiability threshold.

Our initial interest in this problem was motivated by the search of challenging benchmarks for the evaluation of satisfiability solvers for multiple-valued logics. The experimental results we report here suggest that it makes sense to test candidate algorithms on instances near the threshold. Furthermore, the random generator of instances we have implemented is able to provide a large number of such hard instances easily. Our current research is focused on encoding problems used in the evaluation of classical satisfiability solvers into regular CNF formulas and investigating the advantages derived from using regular instead of classical clausal forms.

Satisfiability checking procedures for regular Horn and 2-CNF formulas were introduced in [5, 8, 13]. See [10, 11] for an overview of proof methods for testing the satisfiability of multiple-valued formulas, as well as some applications of this work.

This paper is organized as follows. In Section 2 we define the logic of signed and regular CNF formulas; in Section 3 we describe the regular DP procedure; in Section 4 we report some experimental results about phase transitions in regular CNF formulas; in Section 5 we derive upper bounds on the unsatisfiability threshold. We finish the paper with some concluding remarks.

2 Signed CNF formulas

In this section we define the syntax and semantics of signed CNF formulas. On this basis, we present the logic of regular CNF formulas as a subclass of signed CNF formulas.

Definition 1 truth value set A truth value set $\mathbb{N}$ is a finite set $\{i_1, i_2, \ldots, i_n\}$, where $n \in \mathbb{N}$. The cardinality of $\mathbb{N}$ is denoted by $|\mathbb{N}|$.

Definition 2 signed CNF formula Let $S$ be a subset of $\mathbb{N}$ ($S \subseteq \mathbb{N}$) and let $p$ be a propositional variable. An expression of the form $S \vdash p$ is a signed literal and $S$ is its sign. The signed literal $\overline{S} = (\mathbb{N} \setminus S) : p$ denotes the complement of the signed literal $S : p$. A signed clause is a finite set of signed literals. A signed CNF formula is a finite set of signed clauses.

Definition 3 subsumption of signed literals A signed literal $S : p$ subsumes a signed literal $S' : p'$, denoted by $S : p \subseteq S' : p'$, iff $p = p'$ and $S \subseteq S'$.

Definition 4 length The length of a signed clause $C$, denoted by $|C|$, is the total number of occurrences of signed literals in $C$. The length of a signed formula $\Gamma$, denoted by $|\Gamma|$, is the sum of the lengths of its signed clauses.

Definition 5 satisfiability An interpretation is a mapping that assigns to every propositional variable an element of the truth value set. An interpretation $I$ satisfies a signed literal $S : p$ iff $I(p) \in S$. An interpretation satisfies a signed clause iff it satisfies at least one of its signed literals. A signed CNF formula $\Gamma$ is satisfiable iff there exists at least one interpretation that satisfies all the signed clauses in $\Gamma$. A signed CNF formula that is not satisfiable is unsatisfiable. The signed empty clause is always unsatisfiable and the signed empty CNF formula is always satisfiable.

The main semantic difference between classical and signed CNF formulas arises in the definition of interpretation, where the number of truth values is greater than two. Then, the concept of satisfiability of signed clauses and CNF formulas is the same as the classical one, but with respect
to multiple-valued interpretations. Thus, classical proof methods may be generalized naturally to work with signed CNF formulas provided that special care be taken at the literal level.

Observe that signed CNF formulas are clausal forms which are independent of the multiple-valued logic we are dealing with, except for the number of truth values. Such greater generality is important since, in many multiple-valued logics, it is difficult to find a clausal form in the object language of the logic.

Next, we introduce the syntax of regular CNF formulas. As they are a subclass of signed CNF formulas, the definitions of the semantics introduced above remain valid for regular CNF formulas. Roughly speaking, we assume that there is a total order over the truth value set and signs have a specific form.

Definition 6 regular sign Let \( \geq i \) denote the set \( \{ j \in N \mid j \geq i \} \) and let \( \leq i \) denote the set \( \{ j \in N \mid j \leq i \} \), where \( \leq \) is a total order on the truth value set \( N \) and \( i \in N \). If a sign \( S \) is equal to either \( \geq i \) or \( \leq i \), for some \( i \), then it is a regular sign and \( i \) is its value.

Definition 7 regular CNF formula Let \( S \) be a regular sign and let \( p \) be a propositional variable. An expression of the form \( S : p \) is a regular literal. If \( S \) is of the form \( \geq i \), then we say that \( S : p \) has positive (negative) polarity. A regular clause is a finite set of regular literals. A regular clause containing exactly one literal is a regular unit clause. A regular CNF formula is a finite set of regular clauses.

3 The regular DP procedure

In this section we describe the regular DP procedure defined by Hähnle [8]. Such procedure relies on the following rules:

Regular one-literal rule: If a regular CNF formula \( \Gamma \) contains a regular unit clause \( \{ S : p \} \), then

1. Remove all clauses containing a literal \( S' : p \) such that \( S \subseteq S' \).
2. Delete all occurrences of literals \( S'' : p \) such that \( S \cap S'' = \emptyset \).

Regular branching rule: Reduce the problem of determining whether a regular CNF formula \( \Gamma \) is satisfiable to the problem of determining whether \( \Gamma \cup \{ S : p \} \) is satisfiable or \( \Gamma \cup \{ (N \setminus S) : p \} \) is satisfiable, where \( S : p \) is a regular literal occurring in \( \Gamma \).

The pseudo-code of the regular DP procedure is displayed in Figure 1. The main function is regular-DP: it returns true if the input regular CNF formula \( \Gamma \) is satisfiable and returns false if \( \Gamma \) is unsatisfiable. Function regular-unit-resolve applies repeatedly the regular one-literal rule. Function pick-literal selects the next literal to which the regular branching rule is applied by using a regular version of the two-sided Jeroslow-Wang rule [8]. Given a regular CNF formula \( \Gamma \), pick-literal selects a regular literal \( L \) occurring in \( \Gamma \) that maximizes \( J(L) + J(\overline{L}) \), where

\[
J(L) = \sum_{\forall \overline{L} : \overline{L} \subseteq L} 2^{-|C_i|}.
\]

It is worth noting that when we only consider two truth values we get the classical DP procedure.

Example 1 Let \( \Gamma \) be the following regular CNF formula:

\[
\Gamma = \{(\leq 1 \ p_1, \geq 1 \ p_3, \leq 1 \ p_4), \{\geq 1 \ p_3, \leq 1 \ p_4\}, \\
\{\leq 1 \ p_1, \leq 1 \ p_3, \leq 1 \ p_4\}, \{\geq 1 \ p_1, \leq 0 \ p_3\}, \\
\{\geq 1 \ p_3, \leq 1 \ p_4\}, \{\leq 0 \ p_1, \leq 1 \ p_3, \leq 1 \ p_4\}\}
\]

Figure 2 shows a proof tree of the unsatisfiability of \( \Gamma \) created by function regular-DP. The root node contains \( \Gamma \) and the remaining nodes contain the regular CNF formula obtained after applying regular-unit-resolve to the formula selected for doing branching.

4 Experimental results

In [14] we showed experimentally that phase transitions occur when we run the regular DP procedure on regular random 3-SAT instances. There, we only considered truth values sets with cardinality 3, 4 and 5. The thresholds we obtained for such cardinalities were 5.92, 7.08 and 7.75, respectively. In view of these results, one could be tempted to conjecture that the location of the threshold increases
function regular-DP (Γ: set of clause) : boolean;
var L: literal;
begin
Γ := regular-unit-resolve(Γ);
if Γ = ∅ then return(true);
if ⊢ ∈ Γ then return(false);
L := pick-literal(Γ);
if regular-DP (Γ ∪ {L}) then
   return(true);
else
   return(regular-DP (Γ ∪ {¬L}))
end;

function regular-unit-resolve (Γ: set of clause)
: set of clause;
var L: literal;
var C: clause;
begin
while ∃(L) ∈ Γ and ⊢ L do
begin
Γ := {C | ∀L′ ∈ C ∈ Γ such that L ⊆ L′};
Γ := {C − {L′|L′ ∈ C and L′ ⊆ L}|C ∈ Γ}
end
return(Γ);
end;

Figure 1: The regular DP procedure

linearly as a function of the cardinality of the truth value set.

The aim of this section is to describe, on an experimental basis, how increases the location of the threshold as we increase the number of truth values. To this end, we designed a series of experiments and observed that the increase is logarithmic in the cardinality of the truth value set. After describing the generator of regular random 3-SAT instances used, we report the experimental results obtained. The regular DP procedure was implemented in C++ and the experiments were carried out with 3 Sun workstations Ultra-1.

4.1 Generator of regular random 3-SAT instances

The generator of regular random instances we implemented has as parameters the number of clauses (C), the number of propositional variables (V), and the cardinality of the truth value set (|N|). Given C and V, an instance of the regular random 3-SAT problem is produced by generating C regular clauses of length 3 which are not tautologies. Each

regular clause is produced by uniformly choosing 3 literals with different propositional variable from the set of regular literals of the form [≤p] or [≥p], where p is a propositional variable, ∈ N − {∧, ∨}, j ∈ N − {⊥}, and ∨ and ⊥ denote the top and bottom elements of N. Regular literals of the form [≤∧] or [≥∧] are not considered because they are tautologies.

4.2 Experiments

We applied the regular DP procedure to 60-variable |N|-valued random 3-SAT instances for |N| = 2, 3, 4, 5, 7, 9, 11, 13, 14, 15, 16, 17, 20, 25, 30, 35. For each of such cardinalities of the truth value set, we varied the ratio of clauses to variables from 1 to 12 for |N| < 7 and from 5 to 16 for |N| ≥ 7, incrementing the number of clauses by 5. At each setting we ran the algorithm on 300 randomly generated instances. Therefore, the number of instances executed for each |N| was 39, 900 and the total number of instances executed was 638, 400. The thresholds obtained are shown in Table 1.

The data indicates that the location of the threshold, say t, increases logarithmically as a function of the cardinality of the truth value set |N| as the following equation states:

\[ t(|N|) = 5.94638 \ln^{1.459803}(|N|) \]

This equation was derived from the experimental
| $|N|$ | 2  | 3  | 4  | 5  | 7  | 9  |
|-----|----|----|----|----|----|----|
| Threshold | 4.25 | 6.08 | 7.08 | 7.75 | 8.41 | 8.83 |

| $|N|$ | 11 | 13 | 14 | 15 | 16 | 17 |
|-----|----|----|----|----|----|----|
| Threshold | 9.08 | 9.25 | 9.41 | 9.5 | 9.58 | 9.66 |

| $|N|$ | 20 | 25 | 30 | 35 |
|-----|----|----|----|----|
| Threshold | 9.75 | 10.0 | 10.16 | 10.25 |

Table 1: Location of the threshold for different cardinalities of $N$

...results using the Levenberg-Marquardt method for getting a non-linear regression model implemented in the symbolic calculator Mathematica. In Section 5 we provide a theoretical explanation of such logarithmic increase.

In the remaining of this section we provide more details about our experimental results by means of figures:

Figure 3 is intended to give a pictorial view of the phase transition. It shows the average number of nodes in the proof tree created by the regular DP procedure when $|N| = 7$. Along the horizontal axis is the ratio of the number of clauses to variables in the regular random 3-SAT instances tested. One can observe clearly the easy-hard-easy pattern in the computational difficulty of solving instances as the ratio of number of clauses to variables is varied. The dashed line indicates the percentage of instances that were found to be satisfiable scaled in such a way that 100% corresponds to the maximum average number of nodes.

$\text{Figure 3: Regular random 3-SAT problem, } |N| = 7, V = 60$

Figure 4 is intended to give a pictorial view of how increases the location of the threshold as we increase the cardinality of the truth value set. It shows the average number of nodes in the proof tree created by the regular DP procedure when $|N| = 3$, $|N| = 4$ and $|N| = 5$. Along the horizontal axis is the ratio of the number of clauses to variables in the regular random 3-SAT instances tested. Figure 5 is like Figure 4 but for $|N| = 9$, $|N| = 11$, $|N| = 13$ and $|N| = 15$. One can observe in both figures that the average number of nodes in the proof tree increases in the area near the threshold as the cardinality of the truth value set increases.

$\text{Figure 4: Regular random 3-SAT problem, } |N| = 3,4,5, V = 60$

$\text{Figure 5: Regular random 3-SAT problem, } |N| = 9,11,13,15, V = 60$

Figure 6 displays the percentage of satisfiable instances as a function of the ratio of the number clauses to variables for all the truth value sets considered in our experiments. Figure 7 shows the location of the threshold as a function of the cardinality of the truth value set. The experimental thresholds obtained and the equation derived with
Mathematica are both plotted in the graph. One can observe in both figures that the increase of the threshold is not linear.

Figure 6: Percentage of satisfiable instances as a function of $C/V$

Figure 7: Location of the threshold as a function of $|N|$.

Remark: Since we had to run the regular DP procedure on a very large sample of regular random 3-SAT instances, we considered instances with 60 variables in order to get experimental results in a reasonable amount of time. More propositional variables occur in an instance in the area near the threshold, more time is needed to determine whether it is satisfiable.

5 Upper bounds on the unsatisfiability threshold

Let $\Gamma$ be a regular random 3-SAT instance, let $V$ denote the number of propositional variables in $\Gamma$, let $C$ denote the number of regular clauses in $\Gamma$ and let $r = C/V$ be the ratio of clauses to variables in $\Gamma$. The problem we consider is to compute the least real number $k$ such that if $r$ strictly exceeds $k$, then the probability that $\Gamma$ is satisfiable converges to 0 as $V$ approaches infinity. We say in this case that $\Gamma$ is asymptotically almost certainly unsatisfiable. A proposition stating that if $r$ exceeds a certain constant, then $\Gamma$ is asymptotically almost certainly unsatisfiable has as an immediate corollary that this constant is an upper bound for $k$. In this section we obtain upper bounds on the unsatisfiability threshold as a function of the cardinality of the truth value set. This gives us a theoretical justification of our experimental results.

We consider the following model of regular random 3-SAT instances: from the sample space of non-tautological regular clauses of length three over $V$ propositional variables and $|N|$ truth values, uniformly, independently, and with replacement select $C$ clauses to form a regular CNF formula $\Gamma$. Given an interpretation $I$, the probability that $I$ satisfies a regular random literal $L$ is $\frac{1}{2}$; observe that the number of regular literals with positive polarity satisfied by an interpretation coincides with the number of regular literals with negative polarity that are not satisfied and vice versa. The probability that $I$ satisfies a regular random clause (with three literals) is $1 - \left(\frac{1}{2}\right)^3 = \frac{7}{8}$. The probability that $I$ satisfies a regular random 3-SAT instance $\Gamma$ with $C$ clauses is $\left(\frac{7}{8}\right)^C$. Since there are $|N|^V$ possible interpretations for $\Gamma$, the expected number of interpretations that satisfy $\Gamma$ is

$$E[\{|I| \text{ satisfies } \Gamma\}] = |N|^V \left(\frac{7}{8}\right)^C.$$ 

Since the expected number of interpretations that satisfy $\Gamma$ is an upper bound on the probability that $\Gamma$ is satisfiable, it holds that

$$Pr[\Gamma \text{ is satisfiable}] \leq E[\{|I| \text{ satisfies } \Gamma\}] = |N|^V \left(\frac{7}{8}\right)^C,$$

letting $C = rV$, an upper bound for $k$ is found by choosing $r$ so that the expected number of interpretations that satisfy $\Gamma$ converges to 0 as $V$ approaches infinity. Thus, if $r > \log_\frac{7}{8} |N|$, then $\Gamma$ is almost certainly unsatisfiable.

Therefore, if an upper bound on the unsatisfiability threshold increases logarithmically in the cardinality of the truth value set, the location of the
threshold cannot increase quicker than the upper bound does. For that reason, we should get experimental thresholds increasing logarithmically or less than logarithmically in the cardinality of the truth value set. Actually, our experiments indicate that this increase is even slower than logarithmic, because the best fitting logarithmic model that the regression method found from our experimental data is very similar to the square root of the natural logarithm of the cardinality of the truth value set.

6 Conclusions

In this paper we have presented new results about phase transitions in the regular random 3-SAT problem using a generalized DP procedure. We have reported a series of experiments that indicate that the location of the threshold increases logarithmically as a function of the cardinality of the truth value set. In addition, we have provided a theoretical explanation of this fact by deriving upper bounds on the location of the unsatisfiability threshold. Since the instances near the threshold are computationally hard to solve, they provide a testbed for evaluating regular satisfiability solvers.

As future work we plan to improve the upper bounds derived by applying the method used in [12] for the classical random 3-SAT problem, and explore the usefulness of local search algorithms for model finding in the regular setting.

References


On Diagrammatic Reasoning in Category Theory*

W. Marco Schorlemmer  Jaume Agustí  Jordi Puigsegur

Institut d’Investigació en Intel·ligència Artificial
Consell Superior d’Investigacions Científiques
Campus UAB, E-08193 Bellaterra, Catalunya
{marco,agusti,jpf}@iiia.csic.es

Abstract

In the field of category theory, diagrammatic representations are omnipresent. They are not an informal aid, but a formally-defined mathematical object. Category theorists describe categories by means of ‘commutative diagrams’ and prove theorems using a technique called ‘diagram chasing’. In this paper we analyze commutative diagrams and diagram chasing from the perspective of ‘diagrammatic reasoning’. We believe that an exhaustive study in this direction may be useful for applying well-studied categorical notions like ‘sketches’ for the purposes of computation. Although we only give a first glimpse of the interesting relationship between category theory, diagrammatic reasoning, and computation, we think that further research in this direction is worthwhile.

Keywords: Diagrammatic Reasoning, Visual Declarative Programming, Category Theory.

1 Introduction

Category theory is one of those branches of mathematics that heavily uses diagrams, not as an informal aid, but as a formally-defined mathematical object. The so called commutative diagrams are the category theorist’s way of expressing equations. Categories itself are best understood having a graphical image in mind, since a category is actually a graph with some additional structure1, and can be therefore easily visualized as a graph. In addition, proofs of theorems in category theory are often constructed with a technique known as diagram chasing, based on pasting several commutative diagrams together. In fact, for many people proofs expressed in this way are much easier to understand than by applying equational reasoning inferences. The use of graphs and commutative diagrams for specifying mathematical structures has been deeply studied by means of the notion of sketch, first introduced by Ehresmann [10]. Sketches are a graph-based logic and therefore an alternative to traditional string-based logic for specifying mathematical structures [1].

Furthermore there is a well-known relationship between category theory and computer science. Category theory has provided theoretical foundations to the design of programming and specification languages, together with their semantic models. Recently, category theory, together with its diagrammatic notation, serves also as guideline for the emerging theory of information flow [7].

Very often diagrammatic notations capture a large amount of information in a much more compact and clear way than equivalent textual ones, because they somehow resemble what they represent in a much closer way than textual ones do. Barwise and Hammer suggest to provide a definition of resemblance by the use of a precise mathematical concept of homomorphism between an abstract notion of diagrammatic representations and an abstract notion of what they represent [6]. We believe that the research initiated by Barwise et al. on general axioms for logics [4, 5] may be very helpful for that purpose. In particular, the framework provided by general logics [13] may be suitable, at least in cases for logical systems based on sketches. It is in this framework that the syntax of a logic is presented as a category of signatures, abstracting from it what kind of 'symbols’ one actually uses to write down sentences. This abstraction does not restrict a signature to be of textual manner, but it allows it to be of any kind, and therefore also one of diagrammatic nature. This framework also

---

*Supported by project MODELOGOS funded by the CICYT (TIC 97-0857-C02-01)

1To be rigorous we should talk about small categories, but we shall not worry about foundational issues here.
Aspects of formals

This should not surprise us at all, because there is a strong resemblance between the mathematical notion of graph and its physical representation in Figure 1.

Since (small) categories are graphs with some additional structure, mathematicians use the same kind of representation for describing categories and reasoning about them. Let’s analyze that a little bit closer:

**Definition 2.2** A category $C$ is a graph\(^2\) together with a function assigning to each pair of arrows $f$ and $g$ in $C_1$, with $\text{target}(f) = \text{source}(g)$, the composite arrow $(f; g)^3$, with $\text{source}(f; g) = \text{source}(f)$ and $\text{target}(f; g) = \text{target}(g)$, and a function assigning to each node $A$ in $C_0$ an identity arrow $id_A$ in $C_1$, with $\text{source}(id_A) = \text{target}(id_A) = A$, such that, if $f, g, h$ are arrows in $C_1$,

\[
(f; g); h = f; (g; h) \\
id_{\text{source}(f)}; f = f \\
f; id_{\text{target}(f)} = f
\]

In a category, nodes are often called objects, and arrows morphisms.

By representing categories as graphs we clearly indicate which arrows have a common source or target, much more so than if we use a linear listing of the arrows with the source and target given for each of them, instead. Unfortunately, due to the additional structure of categories (composite arrows and identity arrows), its representations by means of finite graphs—each object represented by a node and each morphism by an arrow—can be quite complicated even for very simple categories (see Figure 2(a)), or actually impossible at all, due to infinite number of distinct composite arrows (see Figure 2(b)).

But, we can represent the morphisms of a category by means of paths in a graph instead of directly by arrows.

**Definition 2.3** In a graph $G$, a path from node $A$ to node $B$ of length $n \geq 0$ is a sequence $p = \langle f_1, f_2, \ldots, f_n \rangle$ of arrows for which

\[
\text{source}(f_1) = A \\
\text{target}(f_i) = \text{source}(f_{i+1}) \quad \text{for} \ i = 1, \ldots, n - 1 \\
\text{target}(f_n) = B
\]

\(^2\)This way we only define small categories; large categories are captured by a more general notion of graph, where nodes and arrows form collections rather than sets.

\(^3\)Usually composition is written $g \circ f$. We use its ‘diagrammatic’ notation instead. As its name indicates, it is closer to the diagrammatic representation of composition by means of paths in a graph as discussed in this section.
We will denote a path \( p \) from \( A \) to \( B \) with \( A \xrightarrow{p} B \). If \( n = 0 \), we say that the path is \emph{empty}. A path starting and ending at the same node is called a \emph{cyclic path}.

Paths in a graph capture in an elegant way the additional structure of categories, and their properties. A path represents the composition of the morphisms represented by its arrows. An empty path on a node represents the identity morphism. Figure 3 shows the categories of Figure 2 using the notion of path.

The resulting graphs, though loosing on degree of 'resemblance' with respect to the categories they represent, they remain being highly resemblant, and in addition we obtain other advantages. We gain in clear representations of categories, and are able to express equality of morphisms by means of 'diagrams' in an elegant way, as we will see in the next section.

We can now answer the question put at the beginning of this section: When mathematicians specify and reason in category theory by means of diagrammatic representations, the kind of signatures they use are actually graphs: a set of object symbols, a set of morphism symbols, and relations between them determined by the sources and targets of morphisms.

3 \textbf{Commutative Diagrams}

The basic task in categorical proof is to show the existence of a morphism, or to show the equality of two morphisms, when some other morphisms and objects are given. Category theorists use diagrams to state equality of morphisms, by a \emph{semantic convention} on these diagrams, namely that all paths with the same source and the same target are considered to represent equal morphisms.

By this semantic convention, the representation in Figure 4 states that morphism \( f; k \), represented by path \((f, k)\), and morphism \( h; g \), represented by path \((h, g)\), are equal, \emph{i.e.} \( f; k = h; g \). We call such a diagram (without the dotted arrow), together with this semantic convention, a \emph{commutative diagram}. A commutative composite diagram is a very economical way of showing several equalities simultaneously without duplica-
tion of subterms.

Although the representation of Figure 4 is that of a graph, the precise notion of diagram is much more subtle. In order to use commutative diagrams to represent equalities of morphisms, we need to express much more within one unique representation than a graph would allow. Observe the three diagrams of Figure 5.

![Diagrams](image)

**Figure 5: Diagrams**

They are different diagrams, though for all three of them their nodes are $A, B$, and their arrows are $f, g, h, k$ such that:

- $\text{source}(f) = \text{source}(h) = A$
- $\text{source}(g) = \text{source}(k) = B$
- $\text{target}(g) = \text{target}(h) = A$
- $\text{target}(f) = \text{target}(k) = B$

The diagrams of Figure 5(a) and (b) have the same shape, namely the one given by the graph of Figure 6(a): Nodes and arrows, though, are labeled differently. The diagram of Figure 5(c) has a completely different shape, namely the one given by the graph of Figure 6(b).

![Shape graphs](image)

**Figure 6: Shape graphs**

A diagram is therefore the drawing of a graph that determines the shape, and the labeling of nodes and arrows with respect to the graph we want to describe. This is captured by a graph homomorphism:

**Definition 3.1** A graph homomorphism $\phi : G \to H$ is a pair of functions $\phi_0 : G_0 \to H_0$ and $\phi_1 : G_1 \to H_1$ such that for every arrow $f$ of $G$,

- $\text{source}_H(\phi_1(f)) = \phi_0(\text{source}_G(f))$
- $\text{target}_H(\phi_1(f)) = \phi_0(\text{target}_G(f))$

**Definition 3.2** Let $I$ and $G$ be graphs. A diagram in $G$ of shape $I$ is a graph homomorphism $D : I \to G$.

For diagram of Figure 5(a), the homomorphism is $D(1) = D(3) = A, D(2) = B, D(a) = f, D(b) = g, D(c) = h$, and $D(d) = k$, taking the graph of Figure 6(a) as shape. For diagram of Figure 5(b), the homomorphism is $D(1) = D(3) = B, D(2) = A, D(a) = g, D(b) = f, D(c) = k$, and $D(d) = h$, taking the graph of Figure 6(a) as shape, too. For diagram of Figure 5(c), the homomorphism is $D(1) = A, D(2) = B, D(a) = f, D(b) = g, D(c) = h$, and $D(d) = k$, now taking the graph of Figure 6(b) as shape.

**Definition 3.3** In a category $C$, a diagram $D : I \to C$ is commutative (or commutes) if for any pair of nodes 1 and 2 of $I$ and any two paths $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_m)$ form 1 to 2, we have that the two morphisms $D(a_1); \cdots; D(a_n)$ and $D(b_1); \cdots; D(b_m)$ obtained by composition in $C$ are the same:

![Commutative Diagram](image)

Having in mind that empty paths represent identity morphisms, observe the distinct assertions we are making with the diagrams shown in Figure 5, when considered to be commutative: Diagram (a) asserts that $f; g = h$ and $k = id_B$. Diagram (b) asserts that $g; f = k$ and $h = id_A$. Finally, Diagram (c) asserts that $f; g = h = id_A$ and $g; f = k = id_B$.

### 4 Reasoning by Diagram Chasing

The conventional style of proof in category theory is diagram chasing. Diagram chasing is the technique by which category theorists reason with commutative diagrams. It is an easy, visual, reliable style of proving equality of morphisms, or constructing a morphism from others.
4.1 The ‘Free Rides’ of Diagram Chasing

Consider the diagram chasing step shown in Figure 7. The two diagrams pictured to the left state that \( f = g; h \) and \( l = h; k \). Diagram chasing consists of pasting both diagrams together along the common arrow \( h \) to obtain the new commutative diagram to the right. From this diagram, and by the semantic convention established on commutative diagrams discussed in Section 3, we can grasp a new fact not previously stated, namely that \( f; k = g; l \). This is a consequence of chasing the diagrams.

This form of diagrammatic reasoning provides us with free rides, as defined by Shimojima [14]: By chasing two diagrams, new paths with coinciding sources and targets may appear in the new diagram, without explicitly drawing them — we may get them for free! By the semantic convention of ‘commutation of a diagram’, these paths will be considered to represent equal morphisms.

4.2 Correct and Incorrect Diagram Chasings

Diagram chasing is not a mere pasting together of two diagrams along common arrows, since such pasting must be done in a correct way. Consider the diagram chasing in Figure 8. There are three different correct alternatives to chase the two original commutative diagrams together, namely along arrow \( f \), arrow \( g \), or both at once.

We do not obtain new equalities from Diagrams (a) and (b), they do not provide us any ‘free ride’, but from Diagram (c) we can grasp the equality of morphisms \( h = k \). It seems, therefore, that in order to deduce as many new equalities thanks to ‘free rides’ provided by diagram chasing, we would have to paste diagrams along as many common arrows as possible. But this is not always the case.

Observe the diagram chasing shown in Figure 9, which is almost identical to the previous one, but with the direction of arrow \( g \) reversed. As before, from Diagram (c) we would obtain that \( h = k \), but now this cannot be deduced equationally from \( f = h; g \) and \( f = k; g \). Instead, Diagrams (a) and (b) are correct diagram chasings. In order for diagram chasing to provide correct ‘free rides’ we have to paste along arrows that form together a path, which is the key notion in the semantic convention.

4.3 Cyclic Paths in Diagram Chasing

It is more subtle to state the correctness of diagram chasing when cyclic paths are involved, due to the semantic convention put on commutative diagrams that morphisms represented by paths beginning and ending at the same node are equal to the identity morphism of that node. Consider the diagram chasing shown in Figure 10.

\[
\begin{array}{c}
\begin{array}{c}
A \\
\sigma
\end{array} \xrightarrow{f} B \;
\begin{array}{c}
B \xrightarrow{k} D \\
\delta
\end{array}
\end{array}
\Rightarrow
\begin{array}{c}
\begin{array}{c}
A \\
\sigma
\end{array} \xrightarrow{f} B \;
\begin{array}{c}
B \xrightarrow{k} D \\
\delta
\end{array}
\end{array}
\]

Figure 10: Incorrect diagram chasing involving a cyclic path

The resulting diagram asserts — besides other equalities — that \( l; g; f; k = \text{id}_D \), which cannot be equationally deduced from the equations stated by the original diagrams. In general we will disallow diagram chasings on paths if some of their arrows belong to a cyclic path, though this restriction can be relaxed for some cases.

The problem arises when a node not belonging to a cyclic path before the chasing takes place ends up within a cyclic path afterwards. This is what happened to node \( D \) in our example. By the diagram chasing shown in Figure 11 we obtain that \( g; f = \text{id}_B \), which cannot be derived from the original equalities, because we have brought node \( B \) into a cyclic path, due to the loop \( h \) (which is an arrow forming a cyclic path by itself).

\[
\begin{array}{c}
\begin{array}{c}
A \xrightarrow{f} B \\
\delta
\end{array}
\end{array}
\Rightarrow
\begin{array}{c}
\begin{array}{c}
A \xrightarrow{f} B
\end{array}
\end{array}
\]

Figure 11: Incorrect diagram chasing involving a loop

Notice that everything works fine when the affected nodes already belonged to cyclic paths. The diagram...
4.4 The Formal Definition

Let's give a formal definition of diagram chasing. We will first define an equivalence relation on the set of arrows and set of nodes of both original shape graphs. The idea is to merge the two shape graphs into one, at the same time we are making equivalent those nodes and arrows on which the diagrams are chased:

**Definition 4.1** Given two diagrams $D : \mathcal{I} \to \mathcal{G}$ and $D' : \mathcal{I}' \to \mathcal{G}$ and two non-empty paths, $p = \langle a_1, \ldots, a_n \rangle$ and $p' = \langle a'_1, \ldots, a'_n \rangle$ in $\mathcal{I}$ and $\mathcal{I}'$ respectively, $n \geq 1$, such that

1. for all $i, 1 \leq i \leq n$, $D(a_i) = D'(a'_i)$, and
2. no arrow $a_i$ of $p$ and no arrow $a'_i$ of $p'$ belongs to a cyclic path, neither in $\mathcal{I}$ nor in $\mathcal{I}'$ respectively, we define the relation $\sim$ on the set of arrows $I_i \cup I'_i$ and on the set of nodes $I_0 \cup I'_0$ to be the minimal equivalence relation satisfying for all $i, 1 \leq i \leq n$:

   $a_i \sim a'_i$

   $\text{source}(a_i) \sim \text{source}'(a'_i)$

   $\text{target}(a_i) \sim \text{target}'(a'_i)$.

We are now ready to give the actual definition of 'diagram chasing', using the equivalence relation $\sim$ on nodes and arrows just defined.

**Definition 4.2** We define the diagram $D'' : \mathcal{I}'' \to \mathcal{G}$ obtained by chasing diagrams $D$ and $D'$ along paths $p$ and $p'$, and write $\{D, D'\} \rightarrow D''$, in the following way:

1. The set of nodes of $D''$ is $\mathcal{I} \cup \mathcal{I}'$. We form the set $\mathcal{I}''$ of nodes by quotienting $\mathcal{I} \cup \mathcal{I}'$ by the relation $\sim$.

   

   $I'_m = (I_0 \cup I'_0)/\sim$

   $I''_m = (I_1 \cup I''_1)/\sim$
with

\[
\begin{align*}
\text{source}'' &= (\text{source} \cup \text{source'})/' \sim \\
\text{target}'' &= (\text{target} \cup \text{target'})/' \sim .
\end{align*}
\]

2. The graph homomorphism is

\[
\begin{align*}
D_0'' &= (D_0 \cup D_0')/' \sim \\
D_1'' &= (D_1 \cup D_1')/' \sim .
\end{align*}
\]

Notice that we have made abuse of notation by defining \(\text{source}''\), \(\text{target}''\), \(D_0''\), and \(D_1''\) as quotients by \(\sim\) in the obvious way. Recall from Definition 4.1 that \(\sim\) depends on the paths we are chasing. It is easy to prove that \(D''\) is indeed a graph homomorphism.

5 Diagram Chasing vs. Equational Reasoning

We now formally prove that reasoning with diagrams and diagram chasing is indeed a valid alternative to traditional equational reasoning, by proving that every equality of morphisms derivable by diagram chasing as defined in the previous section is also derivable by conventional equational reasoning. We do this by defining a map of entailment systems in the sense of [13]. We need therefore to first define the entailment systems of diagram chasing and equational reasoning, respectively.

5.1 Entailment Systems

In Section 2 we saw that the signatures we are dealing with are graphs. The category \(\mathbf{Grf}\) of graphs and graph homomorphisms is therefore the category of signatures and signature homomorphisms in the sense of [13]. Given a graph \(G\), the diagrams in \(G\) play the role of well-formed sentences over this signature. Since diagrams in \(G\) are graph homomorphisms from a shape graph to \(G\), they form a category, namely the slice category \(\mathbf{Grf}/G\).

We can define a functor \(\text{diag} : \mathbf{Grf} \to \mathbf{Set}\) assigning to each graph \(G\) the set of all diagrams in \(G\), namely \(O(\mathbf{Grf}/G)\) where \(O : \mathbf{Cat} \to \mathbf{Set}\) is the object functor. This functor is therefore the one assigning to each signature the set of all well-formed sentences over it in the sense of [13]. Given a graph \(G\), diagram chasing determines an entailment relation \(\vdash^{\mathbf{DC}}\) between sets of diagrams and single diagrams, by defining it as the reflexive, monotonic and transitive closure of \(\vdash^{\mathbf{DC}}\). Consequently, we have that the category \(\mathbf{Grf}\), the functor \(\text{diag}\) and the entailment relation \(\vdash^{\mathbf{DC}}\), all together constitute an entailment system \((\mathbf{Grf}, \text{diag}, \vdash^{\mathbf{DC}})\), the entailment system of diagram chasing.

Theories (or to be more exact, theory presentations) are given by a signature together with a set of sentences over this signature. In our particular entailment system of diagram chasing, signatures are graphs, and sentences are diagrams in that graph. Theories are therefore linear sketches as defined by Barr and Wells [3]:

**Definition 5.1** A linear sketch is a pair \(S = (G, D)\) with \(G\) a graph and \(D\) is a set of diagrams in \(G\).

Since, given an entailment system, theories form a category [13], we have that linear sketches do so, too. Let \(\text{LinSk}\) denote the category of linear sketches.

For the entailment system of equational reasoning, the category of signatures and signature homomorphisms is \(\mathbf{Grf}\), too, because we are dealing with a set of morphism symbols, a set of object symbols, and relations between them determined by the sources and targets of these symbols. The sentences are equations, i.e., pairs of well-formed strings over these morphism and object symbols, plus the composition symbol \(\cdot\). We write equations with the infix symbol '\(\approx\)' for equality. The functor \(\text{eq} : \mathbf{Grf} \to \mathbf{Set}\) assigns to each signature the set of its well-formed equations. The entailment relation \(\vdash^{\mathbf{ER}}\) is equational reasoning with strings, capturing the reflexivity, symmetry, transitivity, and the congruence properties of equality with respect to composition. The entailment system of equational reasoning is therefore \((\mathbf{Grf}, \text{eq}, \vdash^{\mathbf{ER}})\). Analogous to linear sketches (i.e., theories of the entailment system of diagram chasing), equational theories form a category, and we denote it \(\text{EqTh}\).

5.2 Map of Entailment Systems

The following is an example of an equational reasoning process. It captures the diagram chasing of Figure 7:

\[
\begin{align*}
&f \approx g; h \\
&f; k \approx g; h; k \approx l \\
&f; k \approx g; l
\end{align*}
\]

The reasoning process done by diagram chasing, instead, spares us of many intermediate inference steps compared to equational reasoning, because the semantic

---

4 Actually we are dealing with a family of morphism symbols '\(id\)', subindexed by the object symbols.

5 The strings are well-formed according to the sources and targets of the morphism symbols.

6 To be more rigorous we should define the entailment system of equational reasoning within many-sorted equational logic with polymorphic operators (since composition of morphisms is polymorphic), but the above stated entailment system is sufficient for our purposes.
convention put on commutative diagrams — that different paths with coinciding sources and targets represent equal morphisms — implicitly captures 'graphically' the reflexivity, symmetry, transitivity and congruence properties of equality with respect to composition, without explicitly performing the inference steps, as shown in Figure 13.

Reflexivity: \( p \cong p \) \( \quad A \xrightarrow[p]{p} B \)

Symmetry: \( p \cong p' \) \( \quad A \xrightarrow[p]{p} B \)

Transitivity: \( p \cong p', p' \cong p'' \) \( \quad A \xrightarrow[p]{p} B \xrightarrow[p']{p'} C \)

Congruence: \( p \cong p'; p'' \cong p''; p'' \) \( \quad A \xrightarrow[p]{p} B \xrightarrow[p']{p'} C \xrightarrow[p']{p''} D \)

Figure 13: Equational inferences implicitly captured by commutative diagrams

Let's see these issues in a formal way by studying the map from the entailment system of diagram chasing to the entailment system of equational reasoning:

Given a graph \( G \), if \( D \) is a diagram in \( G \) of shape \( I \) and \( p = \langle a_1, \ldots, a_n \rangle \) is a path in \( I \), \( n \geq 0 \), then we denote with \( p \) the string of morphism compositions \( D(a_1); \ldots; D(a_n) \) represented by \( p \). We map a diagram \( D \) to a set of equations, by defining \( \alpha_G(D) = \{ p \cong p' \mid \text{source}(p) = \text{source}(p') \land \text{target}(p) = \text{target}(p') \} \). Being \( D \) a set of diagrams, \( \alpha_G(D) = \bigcup_{D \in D} \alpha(D) \). Theories (linear sketches) are mapped in the following way: \( \Phi(G, D) = (G, \alpha_G(D)) \).

Through the notion of 'map of entailment systems' we capture in a formal way the intuitive idea that every equality of morphisms derivable by diagram chasing is also derivable by equational reasoning.

**Proposition 5.2** The map \( (\Phi, \alpha) : (\text{Grf}, \text{diag}, \vdash_{\text{DC}}) \rightarrow (\text{Grf}, \text{eq}, \vdash_{\text{ER}}) \) is a map of entailment systems in the sense of [13].

For \( (\Phi, \alpha) \) to be a map of entailment systems, \( \alpha : \text{diag} \Rightarrow \text{eq} \circ \Phi \) has to be a natural transformation, and \( \Phi : \text{Linsk} \Rightarrow \text{EqTh} \) an \( \alpha \)-sensible functor\(^7\), such that \( D \vdash_{\Phi} D \) implies \( \alpha_G(D) \vdash_{\alpha(G, D)} \alpha_G(D) \). We refer to [13] for further details on the general definition of a map of entailment systems. In the rest of this paper we will drop the subscripts of \( \alpha, \vdash_{\text{DC}}, \text{and} \vdash_{\text{ER}} \) when the graph \( G \) is clear from the context.

That \( (\Phi, \alpha) \) is indeed a map of entailment systems is a direct consequence of Lemma 5.6 below. But let's first prove several auxiliary lemmas, in which we use the following conventions: By a path in a diagram we actually think of a path in its shape graph. We extend source and target on arrows to paths in the following way: If \( p = \langle a_1, \ldots, a_n \rangle \), source \((p) = \text{source}(a_1) \) and target \((p) = \text{target}(a_n) \). We will also write \( p = \langle s_1, \ldots, s_m \rangle \) to indicate that \( s_i \) are subpaths of \( p \), such that for all \( i = 1, \ldots, m - 1 \), target \((s_i) = \text{source}(s_{i+1}) \). If \( D' \) is the diagram obtained by chasing diagrams \( D \) and \( D' \) along paths \( p \) and \( p' \) respectively, we will call \( p, p' \) and their resulting path \( p^* \) in \( D'' \) indistinguishably the chased path. We will say that a node \( n \) is on the chased path, if it is the source or target of one of its arrows. Furthermore if \( q = \langle a_1, \ldots, a_n \rangle \) is a path in \( D'' \), we will say that \( q \) is also a path in \( D \) (or in \( D' \)), if for all \( i = 1, \ldots, n \), there exists an arrow \( a_i' \) in the equivalence class \( a_i \), and \( (a_1', \ldots, a_n') \) is a path in \( D \) (or \( D' \)). This latter can be also applied to single nodes and arrows.

**Lemma 5.3** Let \( D'' \) be the diagram obtained by chasing diagrams \( D \) and \( D' \). Every path \( p \) in \( D'' \) can be decomposed in a sequence of subpaths \( p = \langle s_1, \ldots, s_m \rangle \), with \( m > 1 \), such that for all \( i = 1, \ldots, m - 1 \), either \( s_i \) is a path in \( D \) and then \( s_{i+1} \) is a path in \( D' \), or else \( s_i \) is a path in \( D' \) and then \( s_{i+1} \) is a path in \( D \).

**Proof:** Because paths in \( D'' \) are actually finite sequences of equivalence classes of arrows of the shape graphs of \( D \) and \( D' \) (see Definitions 4.1 and 4.2), we can 'isolate' the subsequences of arrows that are completely in \( D \) or \( D' \) respectively. □

**Corollary 5.4** For all \( i = 1, \ldots, n - 1 \), \( \text{source}(s_i) \) and \( \text{source}(s_{i+1}) \) are nodes on the chased path.

**Proof:** If \( s_i \) is in \( D \), then \( s_{i+1} \) is in \( D' \), and since node \( \text{source}(s_i) = \text{source}(s_{i+1}) \), it must be both in \( D \) and \( D' \), and consequently on the chased path. □

**Lemma 5.5** Let \( D'' \) be the diagram obtained by chasing diagrams \( D \) and \( D' \). If \( p \) is a path in \( D'' \) such that its source and target is on the chased path, then there exists a subpath \( q \) of the chased path such that \( \alpha(D) \cup \alpha(D'') \vdash_{\Phi} p \cong q \).

\(^7\)\( \Phi \) is \( \alpha \)-sensible when the theorems of \( \Phi(G, D) \) are completely determined by \( \Phi(G, D) \) and \( \alpha(G, D) \).
Aspectes formals

PROOF: By Lemma 5.3 and Corollary 5.4, \( p = \langle s_1, \ldots, s_n \rangle \), such that for all \( i = 1, \ldots, n \), \( s_i \) is either in \( D \) or in \( D' \), and source\((s_i)\) and target\((s_i)\) are on the chased path. Consequently there is a subpath \( s'_i \) of the chased path between these two nodes, which must go also from source\((s_i)\) to target\((s_i)\), otherwise it would violate the condition put on diagram chasing that no arrow of the chased path belongs to a cyclic path (see Definition 4.1). Since \( s_i \) is either in \( D \) or in \( D' \) and \( s'_i \) in both, we have that \( \alpha(D) \cup \alpha(D') \vdash_{ER} s'_i \approx s_i \) for all \( i = 1, \ldots, n \), and by the properties of equationable reasoning \( \alpha(D) \cup \alpha(D') \vdash_{ER} \) \( \approx \) (proof).

Lemma 5.6 If \( D'' \) is the diagram obtained by chasing diagrams \( D \) and \( D' \), i.e. \( \{ D, D' \} \vdash_{BC} D'' \), then \( \alpha(D) \cup \alpha(D') \vdash_{ER} \alpha(D'') \).

PROOF: Let \( u \) and \( v \) be two paths in \( D'' \), and therefore \( \hat{u} \approx \hat{v} \) is in \( \alpha(D'') \). We have the following cases:

1. \( u \) and \( v \) are either both in \( D \) or else both in \( D' \):
   Then either equation \( \hat{u} \approx \hat{v} \) is in \( \alpha(D) \), or else it is in \( \alpha(D') \), and therefore \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{u} \approx \hat{v} \).

2. \( u \) is in \( D \), but \( v \) is not in \( D \) (or vice versa)\(^*\):
   By Lemma 5.3 and Corollary 5.4, and by letting \( w_1 \) be the empty path from source\((v)\) to itself when source\((v)\) is on the chased path, and \( w_3 \) be the empty path from target\((v)\) to itself when target\((v)\) is on the chased path, we can decompose \( v \) in a sequence of three subpaths \( \langle w_1, w_2, w_3 \rangle \), such that \( w_1 \) and \( w_3 \) are both in \( D \) and source\((w_2)\) and target\((w_2)\) are on the chased path. By Lemma 5.5, there exists a subpath \( q \) of the chased path, such that \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{q} \approx \hat{w}_2 \). Since \( q \) is also a path in \( D \), \( \alpha(D) \vdash_{ER} \hat{u} \approx \hat{w}_1 \hat{q} \hat{w}_3 \), and by the properties of equationable reasoning \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{u} \approx \hat{v} \).

3. Neither \( u \) nor \( v \) is in \( D \) or in \( D' \):
   By Lemma 5.3 and Corollary 5.4, we decompose \( u \) and \( v \) in the sequence of three subpaths \( \langle w_1, w_2, w_3 \rangle \) and \( \langle w'_1, w'_2, w'_3 \rangle \) respectively, such that \( w_1 \), \( w'_1 \), and \( w'_2 \) are each either in \( D \) or \( D' \), and source\((w_2)\), target\((w_2)\), source\((w'_2)\), and target\((w'_2)\) are on the chased path. By Lemma 5.5, there exist subpaths \( \tilde{q} \) and \( \tilde{q}' \) of the chased path, such that \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{q} \approx \hat{w}_2 \) and \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{q}' \approx \hat{w}'_2 \). Consequently \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{u} \approx \hat{v} \).

\( \square \)

(c) If \( q \) and \( q' \) do not overlap, then there exists a subpath \( q'' \) such that \( \langle q, q'' \rangle \) is subpath of chased path. Consequently, and by the properties of equationable reasoning \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{w}_1 \hat{q}_1 \hat{q}_1 \hat{q}_1 \hat{w}_3 \approx \hat{w}_1 \hat{q}_1 \hat{q}_1 \hat{w}_3 \hat{w}_3 \) and \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{w}_1 \hat{q}_1 \hat{q}_1 \hat{w}_3 \approx \hat{w}_1 \hat{q}_1 \hat{q}_1 \hat{q}_1 \hat{q}_1 \hat{w}_3 \), and consequently \( \alpha(D) \cup \alpha(D') \vdash_{ER} \hat{u} \approx \hat{v} \).

\( \square \)

From the diversity of cases one has to consider while proving Lemmas 5.5 and 5.6, we can see the considerable amount of equationable reasoning inferences required in order to 'emulate' one single diagram chasing inference \( \{ D, D' \} \vdash_{BC} D'' \). This observation further highlights that commutative diagrams and diagram chasing somehow embed the properties of equationable reasoning in a natural way.

If by equationable reasoning we cannot prove more equalities of morphisms than by diagram chasing, we say that the map of entailment system is conservative. Formally, the map \( \Phi_?(\delta \alpha) \) would be conservative if \( \alpha_?(D) \vdash_{ER} \hat{w}_1 \) implies \( D \vdash_{BC} \delta \alpha_?(D) \). With the current formalization of diagram chasing given in Definitions 4.1 and 4.2 it is not. This is due to the particular treatment we are giving to cyclic paths. Observe the case of Figure 11; we cannot deduce the equality of morphisms \( f; g \) and \( id_A \) by diagram chasing, because we disallow chasings along cyclic paths in order to avoid incorrect deductions, although \( f; g \approx id_A \) can be deduced by equationable reasoning.

6 Computing by Diagram Chasing

We are interested in exploiting sketches as a computing device. As an example, let’s see how we would graph-
ically describe a very simple functional programming language. Figure 14 shows the graph that represents the category providing us with the basic operators of the language together with its typing. Arrows originating at \( \mathbf{1} \) are constants of their target type (e.g. \( \text{false} \) is a constant of type \( \mathbf{Bool} \)).

\[ \begin{array}{c}
\text{Bool} \\
\text{succ} \\
\text{iszero} \\
\text{false} \\
\text{true} \\
\text{Nat} \\
\text{chr} \\
\text{ord} \\
\text{a} \\
\text{b} \\
\text{Char} \\
\end{array} \]

Figure 14: Category providing types and basic operators

The behavior of these basic operators is specified by means of the commutative diagrams of Figure 15.

The intended meaning of these operators in the obvious one. Before going any further, we need to introduce another visual formalism frequently used in category theory, namely cones. Like diagrams, they are also a particular type of graph homomorphism:

**Definition 6.1** Let \( \mathcal{I} \) be a finite discrete graph and \( \mathcal{G} \) a graph. A finite discrete cone in \( \mathcal{G} \) is a graph homomorphism \( L : \mathcal{I} \rightarrow \mathcal{G} \), a node \( n \) of \( \mathcal{G} \) and a collection of arrows \( \pi_j : n \rightarrow L(i) \), one for each node \( i \in \mathcal{I} \).

\[ \begin{array}{c}
\pi_i \\
\ldots \\
\pi_j \\
L(i) \\
\ldots \\
L(j) \\
n \\
\end{array} \]

The node \( n \) is called the vertex, the diagram \( L \) the base of the cone.

We need to specify constants as nullary operators, by determining \( \mathbf{1} \) to be a final object. This is done with the product cone (with vertex \( \mathbf{1} \) and empty base) of Figure 16.

\[ \mathbf{1} \]

**Figure 16:** Product cone specifying the final object

This description of the functional programming language is nothing else but a finite product sketch:

**Definition 6.2** A finite product sketch \( S \) is a triple \( (\mathcal{G}, \mathcal{D}, \mathcal{L}) \) where \( \mathcal{G} \) is a graph, \( \mathcal{D} \) is a collection of diagrams in \( \mathcal{G} \), and \( \mathcal{L} \) is a collection of cones.

We don’t want to enter here in more details about sketches, because our purpose is to give the basic notions in order to present our argumentation. We refer to [3] for an exhaustive study on sketches.

Let us now see how computation with commutative diagrams by means of diagram chasing may look like. Suppose we want to know if the numeric code for character ‘a’ is zero or not, i.e. we want to know to which other morphism from \( \mathbf{1} \) to \( \mathbf{Bool} \) the composite morphism \( a; \text{ord}; \text{iszero} \) is equal.

We would start with Diagram 17(a) and start to paste commutative diagrams taken from Figure 15. By chasing Diagram 15(e) on \( \text{ord} \) we get Diagram 17(b). We indicate by dotted lines those arrows on which we have just pasted a diagram. The solid-line arrows show us those arrows on which we still have to paste other diagrams. Next, by chasing Diagram 15(a) on \( a \) and \( \text{chr} \) we get Diagram 17(c). Notice that in this step we are not chasing along arrows of a common path. This would seem a violation of Definitions 4.1 and 4.2. It is not, because we are actually doing here as an ‘unchasing’ of diagrams, i.e. a reverse process of diagram chasing. You can check that the chasings done in the reverse direction (from (d) to (a)) are all in accordance to Definitions 4.1 and 4.2. Finally, by chasing Diagram 15(c) on \( \text{zero} \) and \( \text{iszero} \) we get Diagram 17(d), from which we can read that the numeric code of character ‘a’ is indeed zero.

Of course this example is in no manner a complete analysis of how sketches and diagram chasing may be applied to computation. A detailed study remains to be done. We only attempted to give a glimpse of the kind of questions that bother us, and in which directions we are trying to pursue them.

7 Conclusions and Future Work

We have formally analyzed well-known notions and methodologies of category theory, like commutative diagrams and diagram chasing, and we have motivated them from the ‘diagrammatic reasoning’ point of view, in order to lay down the basis for exploring how such reasoning methodologies may be useful for the purposes of computation. We have noticed that the formalization of diagram chasing as an entailment system and its relation to equational reasoning by means of a map of entailment systems is not that straightforward as it might appear on a first sight, due to the special treatment given
to cyclic paths. Though diagram chasing is correct—we can capture it by alternative equational reasoning—not every equationally derivable equality of morphisms can be captured by diagram chasing, at least with the formal definition given in this paper. It is therefore necessary to further study alternative formalizations of diagram chasing in order to present a more satisfactory solution. We would like to finish discussing other future work where some interesting new results may be obtained.

7.1 Homomorphism and Institutions

In this paper we have neglected the semantic component of of the reasoning process by diagram chasing, but in order to determine the resemblance of a syntax to the structures we attempt to represent with it, we have to look at the relationship between sentences and their models. The 'degree of homomorphism' of our logical system depends on the resemblance of sentences that are true in the structures under study, and these structures themselves. Let Mod(Σ) be the category of Σ-models determined by the institution [12]. If M is a Mod(Σ)-object, i.e. a Σ-model, and ϕ ∈ sen(Σ) is true in M, we write $M \models \Sigma \varphi$, where $\models \Sigma \subseteq O(\text{Mod}(\Sigma)) \times \text{sen}(\Sigma)$. We have therefore that $(M, \varphi) \in \models \Sigma$. We are actually interested in determining how 'similar' the two elements of the pairs $(M, \varphi) \in \models \Sigma$ are, that is we are interested in stating the resemblance of functors sen : Sign → Set and $O \circ \text{Mod}^{op} : \text{Sign} \rightarrow \text{Set}$ when ap-
plied to sentences that are true in structures. We want to study resemblances of representations and represented structures comparing these two functors.

7.2 Sketches for Visual Declarative Programming

Due to the close relationship between category theory and computer science, the former has provided theoretical foundations to the design of programming and specification languages. This motivates us to explore how the visual aspects of category theory provide a suitable theoretical framework for the formal analysis and design of diagrammatic syntaxes and operational semantics of visual declarative programming languages. Recently some work in this direction has been done, exploiting the category-theoretic notion of ‘sketch’ for computation. On one hand, Bagchi and Wells thoroughly studied a graph-based logic constructed over the notion of sketch. They affirm it as “a first step towards a theory that is directly implementable for the purposes of computation” [1, 2]. On the other hand, Duval and Reynaud notice that first-order structures are not well adapted to computation. They provide “a definition of sketches to deal explicitly with ‘programs’ (i.e. with operational semantics)” [8, 9]. We would like to further explore to which extend sketches fit well for the purpose of computation, by taking advantage of the experience of the diagrammatic reasoning community.

References


Multidimensional Scaling Analysis using Neural Networks with Distance-Error Backpropagation

Lluís Garrido  
Dept. ECM  
Univ. de Barcelona  
garrido@ecm.ub.es

Sergio Gómez  
Dept. Eng. Informàtica  
Univ. Rovira i Virgili  
gomez@etse.urv.es

Jaume Roca  
Dept. ECM  
Univ. de Barcelona  
roca@ecm.ub.es

Abstract

We show that Neural Networks, trained with a suitable error function for backpropagation, can be used for Metric Multidimensional Scaling (i.e. dimensional reduction while trying to preserve the original distances between patterns) and are able to outdo other standard methods mainly because of the ability to model non-linear maps.

Keywords: Neural Networks, Multidimensional Scaling, Principal Component Analysis, Backpropagation.

1 Introduction

A standard problem in Multidimensional Scaling (MDS) (see [1], for example) consists in trying to map a collection of patterns represented as points in an \( n \)-dimensional space, \( x_a \in \mathbb{R}^n \) \((a = 1, \ldots, p)\), to a lower dimensional space,

\[
x_a \rightarrow y_a \in \mathbb{R}^m \quad (m < n),
\]

in such a way that the distances between the projected points, \( y_a \), resemble as closely as possible the distances between the original ones. Typical values chosen for \( m \) are 2 or 3, since in this way the method can be used to visualize the most relevant features of the original \( n \)-dimensional configuration.

This can be set as a minimization problem once an energy function is given. Since we are concerned with the preservation of distances a natural choice is

\[
E = \frac{1}{2} \sum_{a,b} \left( d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2,
\]

where \( d_{ab}^{(n)} \) and \( d_{ab}^{(m)} \) represent the Euclidean distance between patterns \( a \) and \( b \) in the original and projected spaces, respectively.

The optimal mapped configuration will be the one satisfying the set of \( p \times m \) non-linear equations \( \partial E/\partial y_a = 0 \). Solving them directly is generally out of reach and substitute approximate methods should be addressed. The standard one is just Principal Component Analysis (PCA). With this method one can determine, in the original space \( \mathbb{R}^n \), the set of \( m \) principal directions (i.e. those directions along which the data have the highest variance). The projection onto them gives the mapped \( m \)-dimensional configuration, which makes actually the optimal solution among the restricted set of orthogonal projections. Other methods, such as Non-Linear PCA (NLPCA) [3, 4, 5], may be able to perform non-linear projections. However, NLPCA is really inappropriate for Metric MDS because it is only concerned with the (approximate) invertibility of the map but does not care at all of the distances between patterns in the projected space.

2 MDS with Neural Networks

Here we will give an alternative solution to this problem which involves the use of Neural Networks with a suitable error function for backpropagation. The main idea consists in building a net with \( n \) input units and a number of hidden layers, containing a bottle-neck layer with only \( m \) units and an output layer with \( n \) units. Backpropagation is invoked with an error function that contains, in addition to the quadratic error term between input and output, a new piece which is introduced to minimize the difference between the distances of pairs in the input and neck layers. Then, when enough iterations have been performed, the projected configuration is read
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

out from the neck layer.

In order to use the net in the most efficient way it is convenient to perform a translation and a global scaling of the initial data:

\[ x_a \rightarrow \xi_a^{in} = \lambda^{in}(x_a - a), \]

so as to make \( \xi_a^{in} \in [0, 1]^n \). Then one can use \( \xi_a^{in} \) as the input to the net. The outcome of the neck layer, \( \xi_a^{nk} \), lives in the region \([0, 1]^m\) since we are using sigmoid activation functions. This implies that \( 0 \leq d_{ab}^{nk} \leq \sqrt{m} \) while \( 0 \leq d_{ab}^{in} \leq \sqrt{n} \) for any pair of input points \((x_a^{in}, \xi_a^{in})\), where \( d_{ab}^{nk} \) and \( d_{ab}^{in} \) stand for the distances between patterns \( a \) and \( b \) in the neck and initial layers, respectively.

The error function that we have considered in the backpropagation method is given by

\[ E_{BP} = \alpha E_1 + (1 - \alpha) E_2, \]

where

\[ E_1 = \sum_a \left( \xi_a^{out} - \xi_a^{in} \right)^2 \]

and

\[ E_2 = \sum_{a,b} \left( \frac{d_{ab}^{nk}}{\sqrt{n}} - \frac{d_{ab}^{in}}{\sqrt{m}} \right)^2, \]

and \( \alpha \in [0, 1] \) controls the relative contribution of each part. The term \( E_1 \) favors those maps for which the representation in the bottle-neck layer can be best accurately inverted to recover the original configuration. The second term, \( E_2 \), is the most important one since it forces this representation in the bottle-neck to inherit, as closely as possible, the metric structure of the original configuration. The different scalings for \( d_{ab}^{nk} \) and \( d_{ab}^{in} \) in this term are introduced in order to have both numbers in the same range. In this way we can guarantee that all possible configurations can still be covered with the use of sigmoids.

The various scalings performed in this process make the outcome of the neck layer not to be directly interpretable as the final \( m \)-dimensional configuration; we can undo all those scalings by setting

\[ y_a = \lambda^{out} \xi_a^{nk}, \]

with \( \lambda^{out} = \sqrt{n/m} \lambda^{in} \). However, a slightly better solution can be obtained by choosing instead

\[ \lambda^{out} = \frac{\sum_{a,b} d_{ab}^{(n)} d_{ab}^{nk}}{\sum_{a,b} (d_{ab}^{nk})^2}, \]

since this is the value of \( \lambda \) that minimizes the function \( E(\lambda) = \frac{1}{2} \sum_{a,b} (d_{ab}^{(n)} - \lambda d_{ab}^{nk})^2 \) for the given neck configuration, which is what we are ultimately trying to achieve with the whole procedure.

In the practical use of the neural network we have noticed that the best results are obtained by letting the parameter \( \alpha \) fall to zero as the learning grows so that the error function \( E_{BP} \) reduces to \( E_2 \) after a certain number of iterations. Actually, a non-zero value of \( \alpha \) is only useful in the early stages of the learning, in order to speed up convergence. In this situation, i.e. with \( E_{BP} = E_2 \), it is easy to prove analytically that the configuration minimizing \( E_{BP} \) differs from the one minimizing directly the original energy function \( E \) in eq. (2) only by a global scaling \( \sqrt{n/m} \) of all coordinates. Thus, the (otherwise technically convenient) scalings that we have introduced above are completely harmless for the purpose of searching for the best mapped configuration.

It is well known that a network with just the input, output and neck layers, with linear activation functions and subject to self-supervised backpropagation is equivalent to PCA [2]. Our approach goes beyond PCA in two important instances. First, the presence of this new distance-error contribution, \( E_2 \), favors those configurations in the neck layer that approximate better the original distances; and second, the use of sigmoid activation functions and the addition of a number of hidden layers makes the neural net able to produce mappings which are more general (non-linear) than just the orthogonal projections of PCA. In fact, a comparative analysis of both approaches over several types of configurations shows that our method produces better results in the "tougher" situations, i.e. when some of the directions discarded by the PCA projection are still relatively important.

3 An illustrative example

As an application of both procedures we have considered a data set\(^1\) consisting of different animal species, characterized by \( n = 17 \) attributes each (15 boolean + 2 numerical). The coordinates \( x_a \) and

distances $d_{ab}^{(n)}$ have been obtained after scaling the numerical attributes to the range $[0, 1]$ in order to assign an equal weight to all attributes\(^2\) (implying that in this case we simply have $\xi_{ab}^{(n)} = x_{ab}$).

When using the neural net, the best scaling for the two-dimensional neck representation is given by $\lambda^{\text{out}} = 2.946$, which is in less than 1.1% disagreement with the expected value of $\lambda^{\text{out}} = \sqrt{17}/2$.

The projected configurations obtained with each method are drawn in figure 1. Patterns are represented by their label. As shown in the plot, both approaches produce a fairly similar configuration. However, the computation of the overall relative error, i.e.

$$\varepsilon = \left( \frac{\sum_{a,b} \left( d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2}{\sum_{a,b} d_{ab}^{(m)}^2} \right)^{1/2},$$

for each method shows that the neural network is giving out a slightly better result,

$$\varepsilon_{\text{PCA}} = 0.2728, \quad \varepsilon_{\text{NN}} = 0.2346,$$

which represents a 14.00% improvement over PCA.

Acknowledgements

This work has been supported in part by a CI-CYT contract AEN95-0590 and by a URV project URV96-GNI-13.

References


\(^2\)Although both methods are probably better suited when most of the attributes under consideration are numerical, rather than boolean, we believe that the large number of dimensions involved helps in making this issue less important.
On the determination of probability density functions by using Neural Networks

Lluís Garrido
Dept. d'Estructura i Constituents de la Matèria,
Facultat de Física, Universitat de Barcelona,
Diagonal 647, E-08028 Barcelona, Spain.
Phone: +34 3 402 11 91
Fax: +34 3 402 11 98
e-mail: garrido@ecm.ub.es

Aurelio Juste
Institut de Física d'Altes Energies,
Universitat Autònoma de Barcelona,
E-08193 Bellaterra (Barcelona), Spain.
Phone: +34 3 581 19 84
Fax: +34 3 581 19 38
e-mail: juste@ifae.es

Abstract

It is well known that the output of a Neural Network trained to disentangle between two classes has a probabilistic interpretation in terms of the a-posteriori Bayesian probability, provided that a unary representation is taken for the output patterns. This fact is used to make Neural Networks approximate probability density functions from examples in an unbinned way, giving a better performance than "standard binned procedures". In addition, the mapped p.d.f. has an analytical expression.

Keywords: Probability theory, stochastic processes, and statistics; Data analysis: algorithms and implementation, data management; Neural networks, fuzzy logic, artificial intelligence.

1 Introduction

Estimating a probability density function (p.d.f.) in a n-dimensional space is a necessity which one may easily encounter in Physics and other fields. The standard procedure is to bin the space and approximate the p.d.f. by the ratio between the number of events falling inside each bin over the total and normalised to the bin volume. The fact of binning not only leads to a loss of information (which might be important unless the function is smoothly varying inside each bin) but is intrinsically arbitrary: no strong arguments for a defined binning strategy, e.g. constant bin size versus constant density per bin, exists. More sophisticated approaches imply for instance the definition of an "intelligent" binning, with smaller bins in the regions of rapid function variation. However, the main drawback still remains: even for a low number of bins per dimension, large amounts of data are necessary since the number of data points needed to fill the bins with enough statistical significance grows exponentially with the number of variables. As it will be shown, Neural Networks (NN) turn out to be useful tools for building up analytical n-dimensional probability density functions in an unbinned way from examples.

This manuscript is organised as follows: in Sect. 2 the proposed method to construct unbinned p.d.f.s from examples is described. After a brief introduction to the statistical interpretation of the output of a Neural Network applied to pattern recognition in the case of only two classes, an expression for the mapped p.d.f. is obtained. Then, a method to quantify the goodness of the mapped p.d.f. is described. In order to illustrate the concept, an artificial example is discussed in Sect. 3, whereas Sect. 4 is devoted to the discussion of an example of practical application in High Energy Physics. Finally, in Sect. 5, the conclusions are given.

2 Method

Let us assume that we have a sample of \( N \) events distributed among 2 different classes of patterns (\( C_1 \) and \( C_2 \)), each event being characterised by a set of
\[ E [\alpha] = \frac{1}{2N} \sum_{\varepsilon=1}^{N} \left[ \alpha(x(\varepsilon)) - d(x(\varepsilon)) \right]^2. \] 

(2.1)

with respect to the unconstrained function \( \alpha(x) \), where \( d(x) \) takes the value 1 for the events belonging to class \( C_1 \) and 0 for the events belonging to class \( C_2 \), it can be shown [3, 4, 5, 6] that the minimum is achieved when \( \alpha(x) \) is the a-posteriori Bayesian probability to belong to class \( C_1 \):

\[ \alpha(\text{min})(x) = \mathcal{P}(C_1 | x). \] 

(2.2)

The above procedure is usually done by using layered feed-forward Neural Networks (see e.g. [1, 2] for an introduction). In this paper we have considered Neural Networks with topologies \( N_1 \times N_{b1} \times N_{b2} \times N_0 \), where \( N_1 \) (\( N_0 \)) is the number of input (output) neurons and \( N_{b1}, N_{b2} \) are the number of neurons in two hidden layers.

The input of neuron \( i \) in layer \( \ell \) is given by,

\[ I_i^{\ell} = \left\{ \begin{array}{ll} x_i^{(\varepsilon)} & \ell = 1 \\ \sum_j w_{ij} S_j^{\ell-1} + B_i^{\ell} & \ell = 2, 3, 4 \end{array} \right. \] 

(2.3)

where \( x_i^{(\varepsilon)} \) is the set of \( n \) variables describing a physical event \( \varepsilon \), the sum is extended over the neurons of the preceding layer (\( \ell-1 \)), \( S_j^{\ell-1} \) is the state of neuron \( j \) at layer (\( \ell-1 \)) and \( B_i^{\ell} \) is a bias input to neuron \( i \) at layer \( \ell \). The state of a neuron is a function of its input \( S_j^{\ell} = F(I_j^{\ell}) \), where \( F \) is the neuron response function. In general the “sigmoid function”, \( F(I_j^{\ell}) = 1/(1 + e^{-I_j^{\ell}}) \), is chosen since it offers a more sensitive modeling of real data than a linear one, being able to handle existing non-linear correlations. However, depending on the particular problem faced, a different neuron response function may be more convenient. For instance, in the artificial example described below, a sinusoidal neuron response function, \( F(I_j^{\ell}) = (1+\sin(I_j^{\ell}))/2 \), has been adopted.

Back-propagation [7, 8, 9] is used as the learning algorithm. Its main objective is to minimise the above quadratic output-error \( E \) by adjusting the \( w_{ij} \) and \( B_i \) parameters.

Let us now consider the situation we are concerned in this paper: we have a large amount of events (“data”) distributed according to the p.d.f. \( \mathcal{P}_{data}(x) \), whose analytical expression is unknown and which we want precisely to approximate. If a Neural Network is trained to disentangle between those events and other ones generated according to any known p.d.f., \( \mathcal{P}_{ref}(x) \) (not vanishing in a region where \( \mathcal{P}_{data}(x) \) is non-zero), the Neural Network output will approximate, after training, the conditional probability for a given event to be of the “data” type:

\[ \mathcal{P}(\text{min})(x) \simeq \mathcal{P}(\text{data} | x) \equiv \frac{\alpha_{data} \mathcal{P}_{data}(x)}{\alpha_{data} \mathcal{P}_{data}(x) + \alpha_{ref} \mathcal{P}_{ref}(x)}, \] 

(2.4)

where \( \alpha_{data} \) and \( \alpha_{ref} \) are the proportions of each class of events used for training, satisfying \( \alpha_{data} + \alpha_{ref} = 1 \).

From the above expression it is straightforward to extract the NN approximation to \( \mathcal{P}_{data}(x) \) as given by:

\[ \mathcal{P}_{data}^{(NN)}(x) = \mathcal{P}_{ref}(x) \frac{\alpha_{ref}}{\alpha_{data}} \frac{\mathcal{P}(\text{min})(x)}{1 - \mathcal{P}(\text{min})(x)}. \] 

(2.5)

As a result, the desired p.d.f. is determined in an unbinned way from examples. In addition, \( \mathcal{P}_{data}^{(NN)}(x) \) has an analytical expression since we indeed have it for \( \mathcal{P}_{ref}(x) \) and \( \mathcal{P}(\text{min})(x) \) is known once we have determined the network parameters (weights and bias inputs).

For what the reference p.d.f. is concerned, a good choice would be a p.d.f. built from the product of normalised good approximations to each 1-dimensional projection of the data p.d.f., thus making easier the learning of the existing correlations in the n-dimensional space. Since \( \mathcal{P}_{ref}(x) \) is a normalised p.d.f. by construction, the normalisation of \( \mathcal{P}_{data}^{(NN)}(x) \) will depend on the goodness of the Neural Network approximation to the conditional probability, so that in general it must be normalised a-posteriori. In the artificial (High Energy Physics) example shown below, the normalisation of the obtained p.d.f.s was consistent with 1 at the 1% (3%) level.

On the other hand, one would like to test the goodness of the approximation of the mapped p.d.f. to the true one. Given a data sample containing \( N_{data} \) events, it is possible to perform a test of the hypothesis of the data sample under consideration being consistent with coming from the mapped p.d.f. For that, one can compute the distribution of some test statistics like the log-likelihood function.
of Eq. (2.6), which can be obtained by generating Monte Carlo samples containing \( N_{\text{data}} \) events generated using the mapped p.d.f.

\[
\mathcal{L} = \log(L) = \sum_{e=1}^{N_{\text{data}}} \log(p_{\text{data}}^{(NN)}(x|e)).
\] (2.6)

Being \( \mathcal{L}_{\text{data}} \) the value of the log-likelihood for the original data sample, the confidence level (CL) associated to the hypothesis of the data sample coming from the mapped p.d.f. is given by:

\[
CL = \int_{-\infty}^{\mathcal{L}_{\text{data}}} d\mathcal{L} \mathcal{P}(\mathcal{L}),
\] (2.7)

which in practice can be obtained as the fraction of generated Monte Carlo samples of the data size having a value of the log-likelihood equal or below the one for the data sample. If the mapped p.d.f. is a good approximation to \( \mathcal{P}_{\text{data}} \), the expected distribution for \( CL \) evaluated for different data samples should have a flat distribution as it corresponds to a cumulative distribution.

3 Artificial example

In this section we propose a purely artificial example in order to illustrate how a Neural Network can perform a mapping of a 5-dimensional p.d.f. in an unbinned way from examples.

In this example our "data" will consist in a sample of 100000 events generated in the cube \([0, \pi]^5 \in \mathbb{R}^5\) according to the following p.d.f.:

\[
\mathcal{P}_{\text{data}}(\mathcal{x}) = \frac{1}{C} \left( \sin(x_1 + x_2 + x_3) + 1 \right) \cdot \left( \frac{\sin(x_4^2 + x_5^2)}{x_4^2 + x_5^2} + 1 \right),
\] (3.1)

which we want to estimate from the generated events. In the above expression, \( C \) is a normalisation factor such that \( \mathcal{P}_{\text{data}}(\mathcal{x}) \) has unit integral. The above p.d.f. has a rather intricate structure of maxima and minima in both, the 3-dimensional space of the first three variables and the 2-dimensional space of the last two variables.

In order to map the above p.d.f., we need to train a Neural Network to disentangle between events generated according to \( \mathcal{P}_{\text{data}}(\mathcal{x}) \) and events generated according to any \( \mathcal{P}_{\text{ref}}(\mathcal{x}) \) non-vanishing in any region where \( \mathcal{P}_{\text{data}}(\mathcal{x}) \) is different from zero. In order to make easier the learning of the existing correlations in the 5-dimensional space, as explained before, \( \mathcal{P}_{\text{ref}}(\mathcal{x}) \) is chosen as the product of good approximations to the 1-dimensional projections of \( \mathcal{P}_{\text{data}}(\mathcal{x}) \), properly normalised to have unit integral.

In the case of data p.d.f., it turns out that the 1-dimensional projections of the three first variables are equal and essentially flat, whereas the 1-dimensional projections for the two last variables can be parametrised as a 4th degree polynomial \( \mathcal{P}_4(\mathcal{x}) \). Therefore, we choose as reference p.d.f.:

\[
\mathcal{P}_{\text{ref}}(\mathcal{x}) = \frac{1}{\mathcal{C}'} \mathcal{P}_4(x_4) \cdot \mathcal{P}_4(x_5)
\] (3.2)

and generate a number of 100000 events according to it. As before, \( \mathcal{C}' \) is a normalisation factor so that \( \mathcal{P}_{\text{ref}}(\mathcal{x}) \) has unit integral. After the training and normalisation, the p.d.f. given by Eq. (2.5) constitutes a reasonably good approximation to \( \mathcal{P}_{\text{data}}(\mathcal{x}) \), as it is indeed observed in Fig. 1, where both are compared for different slices in the 5-dimensional space: (a) \( x_2 = x_3 = x_4 = x_5 = 0 \), (b) \( x_2 = x_4, x_3 = x_5 = 0 \), (c) \( x_2 = x_3, x_4 = x_5 = 0 \) and (d) \( x_4 = x_5 = x_2 = x_3, x_5 = 0 \).

Figure 1: Comparison between the true (solid line), the mapped (dashed line) and the reference (dotted line) p.d.f. versus \( x_1 \) for different slices in the 5-dimensional space: (a) \( x_2 = x_3 = x_4 = x_5 = 0 \), (b) \( x_2 = x_4, x_3 = x_5 = 0 \), (c) \( x_2 = x_3, x_4 = x_5 = 0 \) and (d) \( x_4 = x_5 = x_2 = x_3, x_5 = 0 \).
For that, a number of 10000 Monte Carlo samples have been generated with the mapped p.d.f., each one containing 100000 events, which is the same number of events of the "data" sample. The log-likelihood is computed for each MC sample and its distribution is shown in Fig. 2a), in which the arrow indicates the value of the log-likelihood for the original data sample \((L_{\text{data}})\). From this distribution and the value of \(L_{\text{data}}\) we have found a confidence level of 5.5% associated to the hypothesis of the data sample coming from the mapped p.d.f. This seems a low CL and needs further comments, but as we know the true p.d.f given by Eq.(2.5), we can do much better than performing a single measurement for CL and is to find out its distribution. Very same number of events as the data sample used to obtain the mapped p.d.f. \((N_{\text{data}} = 100000)\), and two with smaller statistics, one with \(N_{\text{exp}} = 10000\) and another with \(N_{\text{exp}} = 1000\).

A number of 10000 Monte Carlo samples have been generated with the mapped p.d.f., each containing \(N_{\text{exp}}\) events, for the three different values of \(N_{\text{exp}}\) and the log-likelihood is computed for each sample in all three scenarios. On the other hand, a number of 10000 data samples are generated with the true p.d.f. in the three scenarios and the confidence level is computed according to Eq.(2.7). The distribution of CL is shown in Fig. 2b) for \(N_{\text{exp}} = 1000\) (dotted line), 10000 (dashed line) and 100000 (solid line). It can be observed that for \(N_{\text{exp}} = 1000\) the distribution of CL is to a good approximation a flat distribution whereas for \(N_{\text{exp}} = 10000\) it starts deviating from being flat, which indicates that the statistics of the data sample is high enough to start "detecting" systematic deviations in the mapped p.d.f. with respect to the true one.

In the case of \(N_{\text{exp}} = 1000\) which, as mentioned above illustrates a common situation in High Energy Physics, the mapped p.d.f. turns out to be a good enough approximation when used for the smaller experimental data sample. In the other extreme, \(N_{\text{exp}} = 100000\), which illustrates the situation in which there is a unique data sample from which one wants to estimate the underlying p.d.f., it can be observed in Fig. 2b) (solid line) the existence of enough resolution to detect systematic deviations in the mapped p.d.f. with respect to the true one. It should be stressed the very complicated structure of the true p.d.f., which makes extremely difficult its accurate mapping and nevertheless the difference between both distributions are the ones observed in Fig. 1 between the solid and the dashed lines. In such situations we can not use the mapped p.d.f. for fine probability studies but it is clear that it is still very useful for other kind of studies like classification or discrimination.

4 High Energy Physics example

In order to illustrate the practical interest of p.d.f. mapping, the following High Energy Physics example is considered.

One of the major goals of LEP'200 is the precise measurement of the mass of the W boson. In November 1996, the LEP centre-of-mass en-
energy at which $e^+$ and $e^-$ collide was raised to $\sqrt{s} = 172$ GeV and an integrated luminosity of about $\int dt = 11$ pb$^{-1}$ was collected by each of the four LEP experiments. At this energy, W-pairs are produced with sufficient boost to allow a competitive measurement of the W mass by direct reconstruction of its product decays. Almost half of the times (45.6%) both W bosons decay hadronically ($W^+W^- \to q\bar{q}q\bar{q}\bar{q}q\bar{q}$) so that four jets of particles are observed in the final state. Let us concentrate in this particular decay channel.

Most of the information about the W mass is contained in the reconstructed di-jet invariant mass distribution, so that $M_W$ can be estimated by performing a likelihood fit to this bidimensional distribution. Therefore, the W mass estimator, $M_W$, is obtained by maximising the log-likelihood function:

$$\mathcal{L}(M_W) = \sum_{i=1}^N \log \mathcal{P}(s_1^i, s_2^i | M_W)$$

(4.1)

with respect to $M_W$, where $\mathcal{P}(s_1^i, s_2^i | M_W)$ represents the probability of event $i$, characterised by the two measured invariant masses $(s_1^i, s_2^i)$, given $M_W$ which, accounting for the existing background, can be expressed as:

$$\mathcal{P}(s_1^i, s_2^i | M_W) = \rho_{ww} \mathcal{P}_{ww}(s_1^i, s_2^i | M_W) +$$

$$+ \rho_{bckg} \mathcal{P}_{bckg}(s_1^i, s_2^i),$$

(4.2)

where $\rho_{ww}$ and $\rho_{bckg}$ are respectively the expected fractions of signal and background in the sample and $\mathcal{P}_{ww}$ and $\mathcal{P}_{bckg}$ are respectively the p.d.f. for signal (W-pair production) and background in terms of the reconstructed di-jet invariant masses.

In particular, for the signal there is a theoretical prediction for the 3-dimensional differential cross-section in terms of the 2 di-quark invariant masses $(s_1, s_2)$ and $x$, the fraction of energy radiated in the form of initial state photons. Because of the initial state radiation, $e^+ e^-$ collide at a reduced center-of-mass energy $\sqrt{s'} = \sqrt{s}(1 - x)$. However, the experimental distribution for signal events has been distorted because of hadronisation, phase-space restrictions, initial state radiation (ISR), reconstruction procedure (jet finding, kinematic fitting, etc.)

In order to link the parton level with the reconstructed level, a folding procedure can be applied so that the signal p.d.f. of a given event $i$, characterised by a pair of reconstructed invariant masses $(s_1^i, s_2^i)$, is given by:

$$\mathcal{P}_{ww}(s_1^i, s_2^i | M_W) = \frac{1}{\sigma_{ww}(M_W)}.$$ 

(4.3)

The integral $\int dx \int ds_1 \int ds_2 T(M_W) \frac{d^3\sigma}{ds_1 ds_2 dx}(M_W),$ where $T(s_1^i, s_2^i | s_1, s_2, x)$ represents the conditional probability of the reconstructed invariant masses given some invariant masses at the parton level and ISR. The initial state radiation is most of the times lost along the beam pipe and therefore unknown, reason for which it is integrated over.

In the case of the fully hadronic decay channel, there is the additional complication due to the fact that one does not know a-priori how to assign each jet to each W boson. It must be mentioned that this is properly taken into account when fitting the events.

Therefore, in order to determine $M_W$, we need to obtain the “transfer function” $T$ of experimental effects for signal as well as the background p.d.f. $\mathcal{P}_{bckg}$ in terms of the reconstructed di-jet invariant masses. Both will be obtained by using NN’s as described in Sect. 2.

As far as the signal is concerned, the conditional probability which is going to be used for the folding procedure is given by:

$$T(s_1^i, s_2^i | s_1, s_2, x) = \frac{f(s_1^i, s_2^i, s_1, s_2, x)}{g(s_1, s_2, x)},$$

(4.4)

where $g(s_1, s_2, x)$ is theoretically known and has a compact expression. Then, the goal is to map the 5-dimensional p.d.f. $f(s_1^i, s_2^i, s_1, s_2, x)$ of $120000$ WW $\rightarrow q\bar{q}q\bar{q}$ events were generated at $\sqrt{s} = 172$ GeV and with $M_W^2 = 80.25$ GeV/c$^2$. The transfer function has been checked to be essentially independent of $M_W$ and, as it will be shown, this approximation is not introducing a bias in the final result. The quarks were hadronised and the particles’ 4-momenta smeared out according to the typical detector resolution of a LEP experiment. Then, a typical selection procedure (82% efficiency, 79% purity) mostly based on topological variables was applied. As a result, $\approx 40000$ WW hadronic events remained, for which it is possible to obtain the variables we are interested in. Then, a 5-11-5-1 Neural Network was trained to learn the differences between the above sample of events (“data”) and a sample of Monte Carlo events (“reference”) generated according to the 1-dimensional projections of the “data” sample. The same proportions were used for the learning phase.

At $\sqrt{s} = 172$ GeV and after selection, most of the background comes from QCD. Because of the efficient selection procedure, only $\approx 60000 q\bar{q}$ (which
will be our "data" in the NN mapping) are left out of 475000 initial events and this is the statistics used to perform the NN mapping. As before, \( \sim 6000 \) "reference" Monte Carlo events were generated according to the 1-dimensional projections of the "data" sample. Both samples were used to train a 2-5-2-1 Neural Network to disentangle between them. Then, the desired p.d.f. in either case (signal and background) is computed as given by Eq.(2.5). In order to test that the event-by-event

Figure 3: Comparison between NN (solid line) and Monte Carlo (points with error bars) prediction for the average di-jet invariant mass p.d.f. for a) signal and b) background. In b), the p.d.f. as obtained by a box method (dashed line) is also shown.

p.d.f. is meaningful, the predicted 1-dimensional projection of the average invariant mass distribution is compared to Monte Carlo in Figs. 3a) and b) for both signal and background by using the obtained \( P_{\text{eww}} \) and \( P_{\text{bkg}} \), respectively. Note the overall good agreement between the distributions.

As a final check, the calibration curve for the obtained estimator by performing a large number of fits to Monte Carlo samples corresponding to an integrated luminosity of 11 pb\(^{-1}\) each, including background and generated at 5 different W masses is shown in Fig. 4. It can be observed that the result of a fit of the form \( < M_W > = P_1 + P_2(1 - M_W^{10.25}) \) is compatible with having 0 intercept and slope 1. Then, there is no need of calibrating the estimator obtained from the fit since it closely corresponds to the physical parameter we are interested in. The performance of the NN

Figure 4: Calibration curve for the estimator as obtained by performing a large number of fits to Monte Carlo samples of \( \int L dt = 11 \text{ pb}^{-1} \) generated at 5 different \( M_W \). The full line corresponds to the result of a linear fit whereas the dashed line represents the ideal case \( < M_W > = M_W^{10.25} \).

in mapping a n-dimensional p.d.f. has been compared to the "box method" [10], a standard procedure to build up binned p.d.f.'s. In the case of the background p.d.f., which is only 2-dimensional, the "box method" yielded reasonable results as shown in Fig. 3b, while in the case of the 5-dimensional p.d.f. it showed strong limitations which made impossible its application. The main reason is the time required to compute the final p.d.f. which needs an integration on top of the adjustment of the "box method" parameters (initial box size, minimum number of MC points inside each box, etc) in a space of high dimensionality and limited statistics. Is in this environment where the mapping of p.d.f.'s by means of NN's may be superior to "standard binned procedures" in terms of accuracy (the p.d.f. is determined in an unbinned way from examples) and speed (the resulting p.d.f. is an analytic function).

5 Conclusions

We have shown that Neural Networks are useful tools for building up n-dimensional p.d.f.s from examples in an unbinned way. The method takes
advantage of the interpretation of the Neural Network output, after training, in terms of a-posteriori Bayesian probability when a unary representation is taken for the output patterns. An artificial example and an example from High Energy Physics, in which the mapped p.d.f.s are used to determine a parameter through a maximum likelihood fit, have also been discussed. In a situation of high dimensionality of the space to be mapped and limited available statistics, the method is superior to "standard binned procedures".

Acknowledgements
This research has been partly supported by CICYT under contract number AEN97-1697.

References


APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO THE DESIGN AND IMPLEMENTATION OF ELECTRONIC Olfactory Systems

J. Brezmes  N. Canyellas  E. Llobet  X. Vilanova  X. Correig

Departament d’Enginyeria Electrònica, Universitat Rovira i Virgili
Av. Països Catalans s/n, 43006 Tarragona (Spain)

jbrezmes@etse.urv.es  ncanyenll@etse.urv.es  ellobet@etse.urv.es  xvilanov@etse.urv.es  xcorreig@etse.urv.es

Abstract

Recent developments in electronic olfactory systems have broadened their range of applications to many industrial fields, such as the chemical, food and cosmetic industries. Mimicking the human olfactory systems, where neural processing enhances the sensibility and selectivity of the chemosensory receptor cells, the new generation of “Electronic Noses” combine arrays of chemical sensors with powerful artificial neural network processing algorithms. This paper reviews the state of the art on these electronic instruments and describes the research activities conducted on this field by the Electrical Engineering Department of the Rovira i Virgili University of Tarragona, Spain.

Keywords: Electronic nose, neural network, tin dioxide, pattern recognition, sensor array

1 Introduction

The monitoring and control of hazardous vapours is becoming a must in many industrial activities due to the restrictions imposed by modern environmental legislation. In less dangerous industrial fields, such as the food and cosmetic industry, automated vapour identification is required to run quality control lines.

The measurement instruments used for these and other activities should be able to identify and quantify single compounds on a vapour mixture or complex odour reliably, at low cost and in real time.

Even though the industrial need for these systems exists, the methodology used nowadays lacks some of these properties. For example, the detection of hazardous substances (carbon monoxide, nitrogen oxides, sulphurs, etc. [1,2]) using traditional techniques such as chromatography or infrared spectrometry require a sampling process that can only be carried out in laboratories run by specialised personnel and equipped with expensive instrumentation. The use of electronic systems with real time operating sensors would allow the fast detection and efficient control of toxic leaks. Measurement systems based on solid state chemical gas sensors offer several advantages over traditional methods due to their low cost, portability and low maintenance.

Another field where the solid state sensors are becoming very important is in the development of the so-called Electronic nose systems [3]. These systems are designed to replace the human olfactory system in processes related to quality control in the food, beverage and cosmetic industry. Even though the solid state chemical sensors offer many advantages they also have important shortcomings that must be addressed if they are to be included in future measurement systems; their behaviour is heavily influenced by environmental parameters such as temperature and humidity, they suffer from drift and repetitiveness problems and their discrimination ability is extremely poor. Nowadays, these limitations prevent their use in commercial applications where a qualitative or quantitative analysis of a vapour mixture is required.

One of the strategies taken to overcome these limitations has been to mimic the functioning of the human olfactory system [3]. Different studies have shown that the human olfactory system receives the aromatic stimulus through chemosensory receptor cells that are kept at constant temperature and humidity. Although there are more than 100 million cells, only about 100 different types of chemosensory receptors can be differentiated. These cells are poorly selective and
their sensitivities are overlapped in the aroma spectra. The most relevant finding of these studies is the important function of the subsequent neural processing, that enhances sensitivity three orders of magnitude (from the sensitivity of ppm’s, $10^{3}$, of the chemo-sensory receptors cells to the ppb’s sensitivity, $10^{2}$, reached by the overall olfactory system) removes drift and increases the discrimination to the point that thousands of odours can be identified.

Experimental systems that take this approach are based on arrays of poorly selective sensors with overlapping sensibilities embedded with signal processing algorithms to enhance selectivity and remove drift and repetitiveness. Due to the non-linear behaviour of the majority of these sensors, artificial neural network algorithms are well suited for these type of applications.

A second approach to increase repetitiveness and remove the influence of the environmental conditions is to build a system with a sensor array composed by humidity, temperature and chemical sensors combined with an artificial neural network trained to compensate humidity and temperature effects on the chemical sensor responses.

Finally, a third way to enhance the capabilities of the semiconductor gas sensors is to find new parameters with greater selectivity and repetitiveness. Opposed to the traditionally defined stationary parameters, different efforts have addressed the dynamic characterisation of semiconductor gas sensors under different vapour samples.

All three strategies promise enhancements over the traditionally operated semiconductor gas sensors, but the most promising alternative is to combine all three methods to overcome the limitations inherent to these type of sensors.

Section 2 reviews the history and describes the state of art in “Electronic Nose” technology, while section 3 summarises the results achieved by the research studies conducted in our Department. Finally, section 4 presents the conclusions and outlines future trends.

2 History and recent developments on electronic nose instruments

The first electronic olfactory systems date back to 1964. However, the concept of an "electronic nose" as an intelligent chemical array sensor system for odour classification did not really emerge until nearly 20 years later. The term “Electronic Nose” appeared around the late 80’s and it was defined as "an instrument, which comprises an array of electronic chemical sensors with partial specificity and an appropriate pattern-recognition system, capable of recognising simple or complex odours". The first conference dedicated to the topic was held in 1990.

Since then, many works have addressed the problem and considerable success has been achieved. All the instruments developed share the same design philosophy, that is, an array of chemical sensor devices combined with pattern-recognition techniques specifically used to sense odorant molecules. The pattern-recognition methods applied include linear algorithms (Principal Component Analysis, Discriminant Function Analysis [4], etc.) and non-linear techniques such as non-linear regression, "non-linear partial least-squares", "partial model building" [5] and artificial neural networks [6].

Since the majority of chemical sensors have non-linear behaviours, the non-linear approaches are better suited than their linear counterparts. Feedforward back-propagation networks with non-linear activation functions are used in many experimental systems with considerable success [7-9]. Non-supervised neural networks, such as Kohonen autoorganizative maps are also being considered [10] in the latest designs.

Experimental electronic noses have been designed to classify perfumes and flavours, alcoholic beverages and blends of tobacco [11], roasting level of coffee grains[12], vintage of wines [13,14] and to perform quality control tasks determining the level of freshness of food(fish and meat) [15,16].

There are now several Electronic Nose systems commercially available, some of them listed in table 1. For example, “The Nose” (a system designed by Neotronics Ltd., UK) uses an array of 10 polymer
elements to monitor beer production. Due to the increasing number of commercial instruments some authors have pointed the need for standardisation methods to measure the features of each commercial product[17]. Future trends head to the development of application-specific Electronic Nose instruments (ASEN) because there is not a universal nose that can solve all odour-sensing problems.

3 Our research

The research activities conducted by the Solid State Chemical Sensor Devices Group of the Department of Electrical Engineering of the University Rovira i Virgili are centred around the design and development of measurement systems based on solid state chemical sensors. These activities can be synthesised in the following areas:

- Design and evaluation of newly developed sensing devices using new materials and structures.
- Selectivity enhancement of commercial semiconductor gas sensors by defining and measuring new dynamic parameters
- Compensation of long term drift and environmental dependencies such as humidity and temperature using artificial neural network algorithms
- Selectivity and repetitivity enhancement of commercial tin oxide sensors using sensor arrays with overlapped sensitivities and pattern recognition algorithms
- Design of application specific Electronic Nose Systems (ASEN) for commercial use in the food or cosmetic industry combining the advantages of the different approaches mentioned above.

\[\text{Array of tin dioxide gas sensors} \xrightarrow{\text{Neural Network processing algorithms}} \text{Identification} \xrightarrow{\text{Quantification}}\]

Figure 1: Schematic diagram of the general approach used in all our studies

The first results obtained by our research team relate to the last four activities. In fact, some of our results come from the combination of two or more of these approaches. The chemical sensors used so far are tin dioxide semiconductor gas sensors. These sensors are well known for their lack of selectivity and their behaviour is heavily influenced by environmental conditions.
parameters such as humidity and temperature. Figure 1 summarises the basic approach used through all of our research work. Our first three research projects are listed below:

3.1 Vapour classification and quantification under variable moisture conditions

In this study we designed a system composed by an array of two commercial tin dioxide gas sensors (Figaro's TGS-822 and TGS-842) and a polymeric type humidity sensor. The goal was to enhance selectivity and compensate for moisture effects using artificial neural networks [18]. These networks were implemented using the MATLAB mathematical environment running on a regular PC.

In a first stage our goal was to classify two different types of vapour samples (ethanol and o-xylene) using the sensor array described above. Because each gas sensor was unable to discriminate between the two vapour types alone, an artificial neural network algorithm was used to exploit the discrimination ability of the two sensor array. Using a feedforward structure with 3 input neurones, a hidden layer with 12 neurones and an output layer with 2 neurones we trained the network with 200 measurements using backpropagation algorithms. The success rate achieved approached 100% (98.5% was the worst case success rate) on a test set of 1260 measures not used for training.

Once the vapour was classified in the identification stage, our goal became to quantify the concentration of the vapour samples presented to the system. To do so, we used two separate networks. This approach was taken because it has been proven that two specialised networks, one for each vapour, give better results than using a single quantification network to determine the concentration of both vapours [19]. The combination of the humidity sensor signals and the neural network algorithms used helped to compensate humidity influence on the semiconductor sensor responses, achieving a maximum relative error of 30%, far better than 100% errors usually found with no humidity compensation. Figure 3 shows the predicted concentration against real values with and without humidity compensation. Again, the structure used was a feedforward, backpropagation trained network with a hidden layer. Out of a total 1430 measurements, 75% of the samples were used for training and 25% were used for evaluation purposes.

![Network structures for the quantification and identification stages](image)
### Table 2: Description of the sensors used to classify aromatic species

<table>
<thead>
<tr>
<th>Model</th>
<th>Mainly applications</th>
<th>Typical detection ranges (ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGS 800</td>
<td>Air quality control</td>
<td>1-10 (Cigarette smoke, gasoline exhaust...)</td>
</tr>
<tr>
<td>TGS 813</td>
<td>Combustible gas detection</td>
<td>500-10,000 (General combustible gases)</td>
</tr>
<tr>
<td>TGS 822</td>
<td>Solvent vapour detection</td>
<td>50-5,000 (Organic solvents)</td>
</tr>
<tr>
<td>TGS 824</td>
<td>Toxic gas detection</td>
<td>30-300 (Ammonia)</td>
</tr>
<tr>
<td>TGS 825</td>
<td></td>
<td>5-100 (H₂S)</td>
</tr>
<tr>
<td>TGS 830</td>
<td>Halocarbon gas detection</td>
<td>100-3,000 (R-22, R-113)</td>
</tr>
<tr>
<td>TGS 842</td>
<td>Methane detection</td>
<td>500-10,000 (Low sensitivity to interfering gases)</td>
</tr>
<tr>
<td>TGS 880</td>
<td>Cooking control</td>
<td>(Volatile gases and water vapour from food)</td>
</tr>
<tr>
<td>TGS 882</td>
<td></td>
<td>50-5,000 (Alcohol vapour from food)</td>
</tr>
</tbody>
</table>

#### Figure 3: Actual vs. predicted ethanol concentrations

#### 3.2 Discrimination and quantification of vapours using stationary and dynamic parameters:

In a second study we tried to discriminate and quantify among ethanol, toluene and o-xylene [concentration ranges: 25, 50 and 100 ppm] using an array of four thick-film tin oxide gas sensors and pattern-recognition techniques. The goal of this experiment was to combine dynamically obtained parameters, traditionally stationary measurements and neural network algorithms to enhance the selectivity and accuracy performance of the sensor array [20].

First, a Principal Component Analysis was carried out to show qualitatively that selectivity improved when the sensor behaviour was dynamically characterised. The steady-state and transient behaviour of the array components were processed with Artificial Neural Networks. In a first stage, a feed-forward back-propagation-trained ANN was used to discriminate among the studied compounds. In the second stage, three separate ANN (one for each vapour) were used to quantify the previously identified compounds.

Processing data from the dynamic characterisation of the sensor array considerably improved its identification performance, rising the discrimination success rate from a 66% when only steady-state signals were used up to 100%. After the vapour was identified, a second stage estimated its concentration. Using steady-state measurements and artificial neural networks there was an overall 95% quantification success rate for the VOC's studied (100% for ethanol and o-xylene, 84% for toluene). These results proved that dynamically obtained parameters increase the identification capability of the network., while traditional stationary parameters are good enough to quantify single vapours.

#### 3.3 Design of an electronic olfactory system to classify aromatic species

The design of application-specific Electronic noses is one of the main goals of our research. In a first tentative we tried to classify different species using an array of nine commercial tin dioxide gas sensors. Although these sensors were designed to detect different vapours, as table 2 details, their sensitivity was strongly overlapped. Mimicking the human olfactory system, the array signals were then processed by a neural network algorithm which determined the species that was introduced in the test chamber [21].

A total of five aromatic species (cinnamon, red pepper, thyme, pepper, and nutmeg) were sampled ten times each. Once more, the structure used was a feedforward network with a hidden layer. Twenty five samples were used for training the network using the standard back-propagation algorithm and the rest were used for evaluation.
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

![Diagram of patterns for two aromatic species](image)

Figure 4: Patterns for two aromatic species

Although a recognition rate of 98% was achieved with a 9 input, 25 hidden and 5 output network PCA analysis showed redundancy between three sensors. Removing the redundant inputs led to a 100% success rate using a 7 input, 20 hidden and 5 output network. The elimination of redundant information is important because it eases the training of a network if few training measurements are used.

The mentioned studies have served us to acquire first hand experience on many techniques that can be applied in the design of electronic vapour measurement systems. Our future work will be headed toward the integration of all these strategies onto specific applications in the chemical, cosmetic and food industry.

4 Conclusions

In our studies we have found that neural networks can minimise solid state gas sensor limitations such as drift, enhancing, at the same time, their selectivity in front of complex odours or vapour mixtures. Another advantage of the neural approach resides in its easy combination with new measurement techniques such as the dynamic characterisation of the sensor response.

Although the use of artificial neural network algorithms in electronic olfactory systems is broadening their range of applications, further research needs to be done. New neural network algorithms (supervised and unsupervised), the combination of different pattern recognition approaches with new measurement techniques and the design of instruments for specific applications are future trends that must be addressed to fulfil the expectations that the so-called "Electronic Nose" systems have created.

5 References


A Depth Perception Algorithm based in sequential fixations of the gaze

José Antonio Aznar*  Manuel Moreno*  Joan L. Moliner **

(*) Laboratori de Percepció visual. Universitat de Barcelona (Spain).
e-mail: jaznar@psi.ub.es.
(**) Laboratori de Percepció. Universitat Autònoma de Barcelona (Spain).
e-mail: moliner@blues.uab.es

Abstract

In this study, a depth perception model based in a sequence of fixations has been designed. As in
biological vision, each fixation allows us not only to pick up one plane more accurately, but also a
foreground (near plane) and a background (far plane). The visual cortex probably must interpolate other
depth planes, which are caught in the following
fixations.
We also proposed that the mechanism used by the
brain for combining different scales is the cortical
retina. The output shows the fixation plane in a fine
scale, while the near and far planes are portrayed in
course scales.

Keywords: Binocular disparity, stereopsis, energy
models, simple and complex cells.

1 Introduction

The visual world we view consists of surfaces
extending away into the distance. Broadly speaking,
most of the computational approaches to depth have
put forth different procedures for accounting for how
binocular matching is achieved. The different
algorithms, either correspondence algorithms or spatial
phase-based algorithms, have ignored the temporal
dimension as a relevant source of information for
constructing a three-dimensional world.

Experimental data [1] points to the fact that recovering
the third dimension consists in a process that takes
places in different time steps.

In this paper, we present a computational model of
depth vision in its initial phase. The computational
model operates at different times steps and it tries to
account for the experimental data reported in
experiments with human subjects. We assume a few
depth planes are extracted from an ocular fixation.
Therefore, the 3D scene has to be reconstructed
through time from a series of vergence ocular
movements. Each fixation gives us three planes, the
own fixation plane, the near plane and the far plane,

although accuracy features can be found only in the
first one.

2 Basis of the model

Our model for depth perception does not use one only
stereo pair, containing all possible disparities. Rather,
it considers that we fix the gaze at a point at a given
moment. The fixation point belongs to some plane of
the 3D scene and is determined by the visual attention
mechanism.

According to the photoreceptor distribution, these
stereo images have to be converted into a
representation equivalent to the format of the retina,
that is to say, a pair of images where the resolution
varies as a function of eccentricity.

Several neurophysiologists [3] verified that many
binocular cells had receptive fields (RF) in non-


Corresponding to the primary visual cortex can be described by two Gabor functions with the same Gaussian envelope but different phases in the sinusoidal modulation. The analytical form of the 2D-Gabor function, when the Gaussian envelope is centered at point (x,y) is:

\[
g(x,y) = \exp \left\{ -\pi \left[ \left( \frac{x}{\sigma_x} \right)^2 + \left( \frac{y}{\sigma_y} \right)^2 \right] \right\} \cdot \exp \left\{ -2\pi f \left[ \cos \alpha + \sin \phi \right] \right\}
\]

Where \(\sigma_x\) and \(\sigma_y\) are the standard deviation of the asymetric gaussian envelope, \(f\) is the frequency tuned, \(\alpha\) is the orientation of sinusoidal modulation and \(\phi\) is the phase.

According to others researching [7, 4], the difference in phase between RF profiles for the two eyes can be
used to quantify differences in shape between them. They argue that these differences in the RF structure is used by the brain to encode binocular disparities in the visual image.

Following to [7], our model can encode binocular disparity and, therefore, different depth planes through differences between phases of two RF (each left and right eye). The left and right RF profiles of simple cells differ in shape (or phase Gabor filter), but are centered at corresponding retinal locations. [5] found that the responses of simple cells can be modeled by first filtering the retinal image, for each left and right eye, by the corresponding RF profile and then adding the two contributions from two eyes. The analytical expression of this processing is:

$$
R_{sc} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dx \, dy \, \left[ (f_l(x,y) \ast h_l(x,y)) + (f_r(x,y) \ast h_r(x,y)) \right]
$$

Where $f_l$ and $f_r$ are the RF profiles of a simple cell of the left and right eyes, $h_l$ and $h_r$ are the left and right retinal images for a specific fixation point.

[9] verified that two adjacent simple cells were usually tuned to the same spatial frequency and orientation. Moreover, they showed that the responses of the paired cells differed in phase by approximately 90 degrees, it is a quadrature pair. [10, 1] suggest that the response of a complex cell can be modeled by adding the squared responses of a quadrature pair. Formally:

$$
R_{cc} = (R_{scl})^2 + (R_{sgr})^2
$$

Where $R_{scl}$ and $R_{sgr}$ are the responses of a simple cell,

[6] showed that negative components of a response of a simple cell to a drifting sine-wave grating is so often truncated. For this reason we will implement the half wave rectified.

According to these neurophysiological bases we will model the response of each eye by a pair of 2D-Gabor functions in quadrature of phase. The binocular response is given by three complex cells that pool signals from cortical simple cells, which will be stated by a tuned excitatory cell (zero disparity detector), a near cell (crossed disparity detector) and a far cell (uncrossed disparity detector).

The binocular neuron tuned to zero disparity is modeled as follows. The response of two simple cells for each eye is obtained by convolution of the sampled image with two 2D-Gabor filters, one having 0 degrees and the other 90 degrees. Then, two responses to filters of the same phase, one for each eye, are summed. Afterwards, each binocular response is half wave rectified and, finally, these two rectified binocular responses are added.

The implementation of binocular Near cells take in account that these neurons respond optimally to objects that are in front of the plane of fixation. Hence, this cells prefer crossed (negative) disparities. The phase difference between the RF is $\delta \phi = \phi_r - \phi_l = 90^\circ$ and can be modeled by two Gabor functions. In a similar way, binocular Far cells respond optimally to objects that are behind the plane of fixation. Hence, this cells prefer uncrossed (positive) disparities. The phase difference between the RF is $\delta \phi = \phi_r - \phi_l = -90^\circ$ and can be modeled by two Gabor functions.

### 3 Experimental results

Two cameras (left and right) pick up a stereo pair (128x128 pixels) focusing the camera on some interesting point of the 3D scene, so that the fixation point on the stereogram (left and right) is determined. We put two larger images (256x256) on the stereo pair centered at the fixation point. Afterwards, we transform them to obtain four subimages corresponding to the left retinal image and four corresponding to the right retinal image. The first subimage (16x16 pixels) correspond to the fovea and it has the same resolution that the original image. The second subimage is 32x32 pixels, the third 64x64 pixels and the fourth 128x128 pixels. The resolution of these three subimages (concentric squares around the fovea) was stated by a box filter (moving-average filter) using 3x3 pixels, 5x5 pixels and 7x7 pixels mask respectively. We will call to the pyramid composed of these four subimages the first-fixation-left-image (1Fl) and the first-fixation-right-image (1Fr).

![Figure 1](image.png)

**Figure 1.** Transformation of the stereo pair corresponding to the first fixation to a retinal image.

In order to process the subimages corresponding to the first fixation we apply a zero disparity detector (tuned excitatory cell) to the central regions of the stereo pair. The crossed disparity detector (Near cell) and the uncrossed disparity detector (Far cell) are applied 'in parallel' to the peripheral subimages (64x64 and 128x128 pixels). Figure 2 shows the depth planes obtained in this fixation after thresholding.
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

The coarse scales guide the vergence movements. Thus, the second fixation will be focused on any interesting point belonging to the far o near plane. When we focus the gaze in this second point of the 3D scene, we pick up other stereo pair (2F₁ and 2F₂), which will be processed in the same way as the first fixation. The same process is applied to the subimages extracted in this fixation. The Figure 3 shows the depth planes (after thresholding) obtained in the second fixation.

These depth planes are interpolated with those obtained in the first fixation. The next fixation and the followings are processed iteratively.

4 Conclusions

A depth perception model based in a sequence of fixations has been designed. As in biological vision, each fixation allows us not only to pick up one plane more accurately, but also a foreground (near plane) and a background (far plane). The visual cortex probably must interpolate other depth planes, which are caught in the following fixations.

In this study, we also proposed that the mechanism used by the brain for combining different scales is the cortical retina. Certainly, when we direct one’s gaze to a point of the 3D scene, we pick up only the high frequencies contained in a little region around it. Also the low frequencies, surrounding this region, are mapped to a joint representation (the cortical retina) containing both spatial and frequency information. Therefore, the fixation plane will be represented in a fine scale, while the near and far planes will be represented in coarse scales. The model takes into account long range connections between RF.

Acknowledgments

This work has been supported by the project PB95-0266 of the Spanish DGES (M.E.C.).

References

Fulls per anotar
Fulls per anotar
Fulls per anotar
Sistemes classificadors basats en algorismes genètics amb enfocament de Pittsburgh per problemes amb atributs de valors reals

Xavier Llorà i Fàbrega Josep M. Garrell i Guiu Ester Bernadó i Mansilla
xevil@salleURL.edu josepmg@salleURL.edu esterb@salleURL.edu

Departament d’Informàtica
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Passeig Bonanova 8 08022-Barcelona

Resum
En aquest treball presentem diferents propostes pel disseny de sistemes classificadors basats en algorismes genètics amb enfocament de Pittsburgh. Aquestes propostes estan orientades a solucionar problemes de classificació on els atributs descrip-
tius són nombres reals. A partir d’un sistema bàsic es proposen un conjunt d’alternatives i es mesura el seu rendiment sobre un problema real: la diagnosi automàtica de càncer a partir de mostres de teixit de giàndula mamària.

Parauls clau: algorismes genètics, sistemes classificadors, funcions de matching, representació de cromosomes, diagnosi automàtica.

1 Introducció
Darrerament els Algorismes Genètics (GA) s’han estat utilitzant en multitud de camps i d’aplicacions. Un dels camps que es pot considerar més interessant és la seva aplicació a problemes d’aprenentatge. En aquest treball els GA s’apliquen a l’aprenentatge artificial destinat a construir sistemes automàtics de classificació. Nosaltres ens centrem en sistemes on els atributs representatius del problema són nombres reals i no valors discrets.

· Dins la classificació automàtica amb GA existeixen dos enfocaments diferencials: Michigan i Pittsburgh. En aquest treball es donen propostes alter-
natives, sota l’enfocament de Pittsburgh, per solucionar problemes de classificació on els atributs són nombres reals, i el cost computacional de partida és molt elevat. Tal i com es veurà a partir dels resultats obtinguts, les propostes reduïen considerablement el temps d’execució, a la vegada que milloren el rendiment dels sistemes clàssics.

A banda d’aquesta introducció, el treball està estructurat en cinc seccions. En la segona secció es fa una breu introducció als GA i als sistemes classificadors. La tercera secció, presenta el conjunt d’alternatives proposades sota una plataforma pròpia: el sistema GENIFER. Les seccions quatre i cinquè presenten el banc de proves utilitzat i els resultats obtinguts. Finalment, es presenten les conclusions i les línies de futur d’aquest treball.

2 Els Algorismes Genètics i els sistemes classificadors

2.1 Breu introducció als GAs
Els Algorismes Genètics (GA) van ser introduïts per en Holland a mitjans dels anys setanta [6]. El model està inspirat en dos principis naturals: el principi d’evolució natural de les espècies de Darwin i les lleis de l’herència d’en Mendel. La idea tracta les possibles solucions a un problema com individus d’una població. Les característiques bàsiques de cada individu es codifiquen en els seus cromosomes, és a dir, en la seva informació genètica. Aquests individus es creuen entre si aplicant els principis
bàsics de l’herència genètica. D’aquesta manera les solucions fills hereten les característiques bàsiques de les solucions pares. Com en tot procés de creuament, s’incorpora la possibilitat de que es produueixin mutacions, petits canvis aleatoris en la informació genètica d’un individu.


2.2 Classificació automàtica amb GA
En primer lloc cal explicar què entenem per classificació. Durant tot aquest treball sempre utilitzarem el mot classificació per indicar el procés que assigna una classe a un objecte. Cal destacar que no volen dir el procés que genera classes. Partim de la hipòtesi que el conjunt de classes és conegut i el que es vol és ubicar un objecte en alguna d’aquestes classes. La classificació, vista d’aquesta manera, també es pot entendre com un procés de predicció i/o de diagnosi.
La tasca del nostre sistema, basat en GA, és aprendre a classificar a partir d’un conjunt d’exemples ja resolts.

Des de la perspectiva dels GA, els problemes de classificació s’han abordat amb dues filosofies: l’enfocament de Pittsburgh [9][8] i l’enfocament de Michigan [6][3]. Descrivim breument les característiques d’ambdós enfocaments:

Pittsburgh
- Un individu de la població representa un conjunt de regles de cardinalitat variable.
- Un individu és una solució completa al problema de classificació automàtica.
- El GA manté el seu esquema tradicional.

Michigan
- Cada individu és una regla.
- Tota la població resol el problema.
- Existeix un algoritme d’assignació de crèdit que reparteix l’avaluació a cada individu.
- El GA és sols un mètode de cerca.

En aquest treball s’ha escollit l’enfocament de Pittsburgh, deixant l’enfocament de Michigan per a treballs posteriors. El motiu bàsic és que l’enfocament de Pittsburgh és el que hereta l’estructura clàssica del GA: un individu és una solució completa al problema.

2.3 GABL: la base de l’enfocament de Pittsburgh
En aquest sistema cada individu representa un conjunt de regles. Els operadors genètics utilitzats són els clàssics: mutació a nivell de bits i creuament amb dos punts de tall, lleugerament modificat per produir individus de mida variable semànticament correctes.

Representació
Un individu de la població representa un conjunt de regles de classificació de cardinalitat variable. Aquest conjunt resol totalment el problema de classificació plantejat.
Cada regla té la forma: Condició → Concepte. La condició codifica, emprant lògica matemàtica d’ordre zero, la conjunció dels tests disjuntius que es realitzen sobre els valors nominals dels atributs. Aquestes condicions i la classe associada es codifiquen en tiges de bits tal com es pot veure a [9].

Operadors Genètics
Pel que fa al creuament entre individus, s’utilitza l’operador Modified Two-Point Crossover descrit a [9]. Respecte a l’operador de mutació, es conserva l’algorisme habitual de complementació del valor d’un bit.
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

| GENIFER-BRE | Binary Rule Encoding |
| GENIFER-MDA | Min. Distance Activation |
| GENIFER-MDAAC | Min. Distance Adapt. Activation |
| GENIFER-RA | Representative Attributes |

Taula 1: Variants actuals del GENIFER.

**Avaluació**

La funció d'avaluació dels individus en el sistema GABIL es defineix com:

\[ \text{aval}(\text{individu}) = (\text{percentatge correcte})^2 \]  

Per cada individu, s'avalua el percentatge d'exemples correctament classificats respecte al total d'exemples d'entrenament. Aquesta funció tan sols té en compte el rendiment de l'individu en la classificació i ignora altres tipus de consideracions com poden ser la longitud dels individus, la complexitat de les regles, etc.

3 Aproximacions a la classificació: Sistema GENIFER

3.1 Introducció

El GENIFER (GENnet clasiFiER) és l'amalgama de diferents propostes de classificació automàtica amb GA de problemes amb atributs descrits per valors reals.

Concretament, a partir d'un sistema de classificació basat en el GA clàssic i inspirat en l'enfocament de Pittsburgh[9], es va procedir a la corresponent adaptació per poder tractar amb atributs amb valors reals. D'aquesta proposta inicial s'observa: l'elevat temps d'execució del GA (aprenentatge) i una forta desviació del percentatge d'encert en l'etapa de test en funció dels conjunts d'entrenament emprats. Per buscar alternatives es decidí variar la representació del coneixement de les regles emprades en la codificació dels individus, sorgint així les variants que es poden trobar a la taula 1.

3.2 Aproximació inicial: GENIFER-BRE

3.2.1 Descripció del sistema

GENIFER-BRE 1 (Binary Rule Encoding) és l'adaptació del sistema GABL[9] de DeJong i Spears a un problema de classificació en el que els atributs descriptius dels exemples es representen amb nombres reals.

L'adaptació bàsica que s'ha de dur a terme consisteix en discretitzar els valors reals dels atributs de cada un dels exemples. Això es realitza fragmentant l'interval del domini de cada un dels atributs en un conjunt finit d'intervals disjunts. Un possible exemple d'aquesta tècnica seria el següent:

\[ [1,6] = [1, 2] \cup (2,3] \cup (3,4] \cup (4,5] \cup (5,6] \]  

A partir d'aquí, el valor d'un atribut deixa d'ésser un nombre real per passar a ser un valor discret: és del primer interval, o és del segon interval, etc. El nombre d'intervals aconsellable respon al compromís entre la precisió que es desitja del GA i el temps d'execució necessari per assolir-la.

El resultat d'intentar classificar un nou cas utilitzant un conjunt de regles pot ser múltiple: el nou exemple es pot classificar en una única classe, pot no classificar-se, o bé el conjunt de regles dona més d'una possible classificació. En aquest últim cas es classifica el nou exemple en funció de la primera regla que es troba al conjunt (lista d'activació).

L'avaluació d'un individu (conjunt de regles) es basa en el percentatge d'encert obtingut en la classificació del conjunt d'entrenament. També s'inicia en la classificació la capacitat de penalitzar aquells individus que tenen un nombre elevat de regles. La funció d'avaluació empra és:

\[ \text{aval}(\text{individu}) = \frac{\text{Pen}}{\#\text{regles}} (\text{percentatge correcte})^2 \]  

Pen és una constant la finalitat de la qual és esbiaixar el comportament del GA vers individus amb poques regles. Concretament s'usa sub-valor el percentatge d'encert d'aquells individus amb més de Pen regles.

1GENIFER-BRE és pràcticament equivalent a un sistema desenvolupat en el nostre grup i anomenat GeB-CS[2].
3.2.2 Problemes detectats

Un cop implementat i provat el GENIFER-BRE es van observar serioses limitacions. Cal destacar:

- La tendència a augmentar excesivament la mida del conjunt de regles, implicant un espai de cerca molt gran.
- Gran diferència entre els percentatges d'encerts obtinguts durant la fase d'entrenament i la fase d'explotació. D'una banda, es pot deduir que el sistema és molt sensible als exemples d'entrenament que se li subministren, però d'altra banda també es pot deduir que el sistema té dificultats per generalitzar i absorbi el soroll dels exemples.

Aquests problems provenen principalment de la representació dels cromosomes. És a dir, la representació utilitzant regles discretes del GABL no s'acaba d'adaptar satisfactoriament a problemes de classificació amb atributs descrits mitjançant nombres reals.

3.3 GENIFER-MDA

Dels problemes sorgits a la proposta anterior, fou necessària la revisió crítica de la mateixa. D'aquesta revisió va sortir el GENIFER-MDA (Minimal Distance Activation) que es descriu a continuació.

3.3.1 Descripció del sistema

Quan analitzem els exemples d'un problema de classificació amb atributs reals, la primera pregunta que es planteja és si es poden definir afinitats dins de l'espai n-dimensional al que pertanyen els exemples. Dit d'una altra forma, la qüestió és si es pot definir, dins d'aquest espai, regions significatives que comparteixin un mateix concepte de classificació associat.

La proposta requereix trobar alguna forma d'expressar el particionat de l'espai n-dimensional que conformen les mostres. Per dur-ho a terme, es fragmentarà l'espai emprant punts significatius d'aquest espai. Aquests punts identificaran un concepte, en el nostre cas una classe. Per saber a quina regió (classe) de l'espai pertany una mostra \( m_i \), es situarà l'exemple en l'espai i es trobarà el punt significatiu més proper. Aquest punt més proper decidirà en quina classe s'ha de classificar el nou exemple. La idea és força similar a certes funcions de similitud utilitzades en un sistema de Raonament Basat en Càsos[4], a la vegada que incorpora certes idees de les tècniques de clustering.

Així doncs, es pot començar a entreveure la redefinició de la codificació de les regles que utilitzarà el GA. Aquesta modificació afecta, a parts iguals, la representació de les regles i la política de matching i d'activació de les mateixes.

Representació

Una regla de classificació continua presentant la forma: \( \text{Condició} \rightarrow \text{Concepte} \). La condició es representa amb un conjunt de valors reals (tants com atributs tingui el problema). Aquesta condició representa un punt significatiu de l'espai. Pel que fa al concepte és manté la representació de la proposta anterior.

Matching

El procediment de matching utilitzat ara és força diferent al del GENIFER-BRE:

- Sigui \( m_i \) la mostra a classificar.
- Sigui \( R \) el conjunt de regles que descriuen el problema, i \( z \) una regla tal que \( z \in R \).
- Sigui \( \text{Dist} \) la funció de similitud entre una mostra i un punt significatiu. Un possible exemple seria la distància euclidiana:

\[
\text{Dist}(m_i, x) = \sqrt{\sum_{j=1}^{\text{Max atribut}} (m_{ij} - x_j)^2}
\]

(4)

- Buscar aquella regla \( r \) que compleixi que:

\[
\text{Dist}(m_i, r) \leq \min_{\forall z \in R} (\text{Dist}(m_i, z))
\]

(5)

- Classificar \( m_i \) en funció de la classe associada a \( r \).

Tal i com es pot apreciar, per classificar una mostra \( m_i \), s'escol·leix aquella regla la condició de la qual és més similar. Aquest canvi en la política d'activació de regles elimina el concepte de llista d'activació del GENIFER-BRE.
3.3.2 Adaptació del GA
Aquesta nova proposta de representació condiciona la introducció de lleugeres modificacions al GA:

1. Incorpora un operador que elimina regles no utilitzades. Aquest busca eliminar regles inútils degut a la proximitat dels punts significatius entre les regles del conjunt.

2. Degut a que la cardinalitat del conjunt de regles es manté estable, es recupera la funció d’avaluació del GABL.

$$aval\text{(individu)} = \text{(percentatge \ correctes)}^2$$

3. Modifica l’operador de mutació per tractar amb nombres reals.

3.3.3 Observacions
Aquesta nova aproximació soluciona els problemes que presentava el GENIFER-BRE. Es destacarien les següents característiques:

1. Obtenim una representació molt més natural.
2. El sistema és més estable.
3. El temps d’execució s’ha reduït considerablement.

3.4 GENIFER-MDAA

3.4.1 Descripció del sistema
L’aproximació anterior presenta bons resultats, tot i així es presenten certs dubtes. Un d’aquestes consisteix en preguntar quin tipus de funció de similitud s’ha d’utilitzar. Precisament en d’altres treballs relacionats[5][2], s’anàlitzen certes funcions de distància basades en la Mètrica de Minkowski:

$$\text{Dist}(m_i, x) = \sqrt{\sum_{j=1}^{\text{Max}.\text{atribut}} |m_{ij} - x_j|^n}$$

A partir dels treballs anteriors es considera la possibilitat que el propi GA busqui quina és la millor n a utilitzar en la funció de distància. Així expoltem la capacitat del GA per anar-se adaptant a l’entorn. És evident que no s’anàlitzen moltes funcions de similitud, però n’hi ha prou per n \( \in \{1, 2, 3\} \) (linial, euclidiana, o cúbica). Es manté idèntica la funció d’activació:

$$\text{Dist}(m_i, r) \leq \min_{x \in R} \text{Dist}(m_i, x)$$

A aquesta aproximació n’hem dit GENIFER-MDAA (Minimal Distance Adapative Activation).

3.4.2 Adaptació del GA
La principal modificació a introduir al GA és la codificació dels individus. Concretament s’ha d’afegir un nou gen a cada regla de classificació. Aquest gen, que sols podrà adoptar els valors enters \( \{1, 2, 3\} \), és el responsable de la tria adaptativa per part del GA de la millor mètrica de distància.

3.4.3 Comportament obtingut
Els resultats obtinguts són pràcticament idèntics als de la proposta anterior (GENIFER-MDA). Malgrat això d’aquests resultats s’ha pogut concloure que:

1. No existeix una mètrica que destaquí respecte les altres.
2. El GA és capaç d’aprendre, no sols una bona solució al problema que es resol, sinó que també pot escollir el mètode més encertat per solucionar-lo.

3.5 GENIFER-RA

3.5.1 Descripció del sistema
En aquest apartat presentarem una nova aproximació adaptativa al problema de classificació basat en GA. El procés evolutiu actual no només haurà de trobar la solució al problema, sinó que a més haurà de ser capaç de discriminar quins atributs són representatius i quins no per la classificació plantejada. Dels atributs no analitzarem el seu grau de representativitat, només si són o no són representatius \( (0/1) \). Aquesta idea és la que configura el GENIFER-RA (Representative Attributes).

Vist els resultats obtinguts en el MDAA, el RA només treballa amb la distància euclidiana, concentrant els seus esforços evolutius en la representativitat dels atributs. Aquest caràcter adaptatiu implica modificar la funció de distància tal i com apareix a continuació, per poder controlar amb el vector \( w \) el atributs que s’empren.

$$\text{Dist}(m_i, x, w) = \sqrt{\sum_{j=1}^{\text{Max}.\text{atribut}} w_j (m_{ij} - x_j)^2}$$
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

3.5.2 Adaptació del GA

La principal modificació que cal incorporar en el GA per implementar aquest concepte és la representació dels individus. Un individu es representarà com un conjunt d'attributs, un conjunt de regles (en realitat punts de l'espai i el seu concepte associat), però ara, en cada regla, caldrà afegir-hi per cada atribut un camp binari que indiqui si l'atribut és representat o no. És a dir, cada regla tindrà associat un vector de pesos $w$, on $\text{Dom}(w_i) = \{0,1\}$. Dos possibles exemples de vectors poden ser:

$$w_1 = <111 \ldots 111>$$  \hspace{1cm} (9) \\
$$w_2 = <100 \ldots 000>$$  \hspace{1cm} (10)

El primer vector indica que tots els atributs de la regla són representatius, mentre que el segon indica que només hi ha un atribut representatius, el primer.

Com en l'aproximació anterior, també cal destacar que s'ha hagut de modificar l'algorisme de mutació de tal manera que s'hi puguin treballar, tant amb els punts representatius com amb el seu pes binari, i el concepte associat a la regla.

3.5.3 Comportament obtingut

El comportament ha estat, un cop més, satisfactori. El sistema classificador no només ha trobat un bon conjunt de regles (amb bons resultats tan durant el train com durant el test), sinó que a més a més ha identificat quins atributs són representatius\(^2\).

3.6 Visió de conjunt

En aquesta secció hem presentat quatre alternatives per a la construcció de sistemes classificadors basats en GA i utilitzant l'enfocament de Pittsburgh. Totes les propostes presentades han estat sempre encarades a solventar problemes amb atributs descrits per valors reals, modificant principalment la representació dels individus de la població. A la taula 2 es pot trobar una breu visió de cadascuna d'elles.

<table>
<thead>
<tr>
<th>GENIFER</th>
<th>Codificació</th>
<th>Activació</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRE</td>
<td>Binària</td>
<td>Llista de regles + CP0</td>
</tr>
<tr>
<td>MDA</td>
<td>Punts representatius</td>
<td>Mínima distància</td>
</tr>
<tr>
<td>MDAA</td>
<td>Punts representatius + funció de distància</td>
<td>Mínima distància adaptativa</td>
</tr>
<tr>
<td>RA</td>
<td>Punts representatius + atributs representatius</td>
<td>Mínima distància descartant atributs</td>
</tr>
</tbody>
</table>

Taula 2: Resum de les variants del GENIFER

postes presentades anteriorment. El problema buscava una col·laboració entre el nostre grup de recerca i el grup de recerca de Tractament de l'Imatge d'Enginyeria i Arquitectura La Salle (Universitat Ramon Llull).

El problema general consisteix en realitzar la diagnòstica automàtica de càncer a partir de mostres de teixit de glàndula mamària. El punt de partida són imatges microscòpiques procedents de biopsies del teixit que es vol estudiar. L'objectiu final consisteix en l'obtenció d'un sistema que diagnostiqui si una mostra procedeix d'un teixit cancerígen o no.

Per a la construcció d'aquest sistema distingirem dues parts clarament diferenciades:

1. L'extracció d'un conjunt de característiques de la imatge utilitzant tècniques de morfologia matemàtica.

2. La construcció d'un "sistema intel·ligent" que, a partir d'un conjunt de mostres d'exemple, aprengui a diagnosticar (classificar) automàticament.

El nostre treball se centra en la segona part del sistema. Pel que fa a la primera part, es poden trobar resums del treball a [10] i [7].

4.2 Modelització del problema i base de dades d'aprenentatge

Per cadascuna de les imatges i utilitzant tècniques de morfologia matemàtica, en [7][10] es va procedir a l'extracció de 24 atributs descrits per valors reals. Per altra banda, per cada cas (a més dels 24 atributs) es té la classe associada (cancerígena o no) dictaminada per l'especialista. Cal tenir present que el conjunt sobre el que es realitzarà l'aprenentatge conté soroll.
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

<table>
<thead>
<tr>
<th>Paràmetre</th>
<th>Valor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mida població</td>
<td>100</td>
</tr>
<tr>
<td>Probabilitat de creuament</td>
<td>0.6</td>
</tr>
<tr>
<td>Probabilitat de mutació</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Taula 3: Valors dels paràmetres bàsics utilitzats en el GA.

5 Resultats obtinguts

A continuació es presentaran els resultats obtinguts a partir de les propostes presentades anteriorment. 

Amb l'objectiu de fer un test el més exhaustiu possible, les 1028 mostres del problema que tractarem, s'han dividit en conjunts de train i conjunts de test. Aquests conjunts utilitzats respectivament en l'entrenament i l'avaluació del sistema, s'han generat emprant diferents proportions en cadascun d'ells, així com realitzant la distribució aleatòria de les mostres 10 cops diferents, obtinguent-ne així un total de 90 parelles de conjunts. Per cada parella de conjunts el GA s'executà 3 cops, fent un total de 270 execucions per cada variant.

Els resultats que presentem són el percentatge d'encerts del sistema (màxim, mínim i mitjà) per a cadascuna de les aproximacions. També es presentarà el rendiment del sistema pel que fa a temps d'execució.

Els valors dels paràmetres bàsics del GA oscil·len al voltant dels de la taula 3. Pel que fa a la primera aproximació (GENIFER-BRE) els valors d'aquestes paràmetres són lleu germant diferents donades les particularitats del sistema comentades anteriorment.

5.1 GENIFER-BRE

Recordem que GENIFER-BRE és l'adaptació del sistema GABL[9] de DeJong i Spears a un problema en el que els atributs són reals.

A la figura 2 es poden observar els resultats obtinguts pel que fa a percentatge d'encerts de les diferents propostes del GENIFER, tant en la fase de train com en la fase de test. Cal destacar que els resultats confirmen l'aplicabilitat dels GA al problema plantejat.

Si s'observen els resultats corresponents al BRE, es pot apreciar que són bons. Malgrat això apareixen execucions amb percentatges d'encert en test baixos (ex: 54.21%). Pel que fa al temps d'execució mitjà en el problema de les biòpsies de càncer de mama, és aproximadament de 91 minuts i 39 segons⁴ en l'etapa d'aprenentatge, valor força elevat. Comparativament, a l'etapa de test el temps mitjà de classificació d'una nova mostra esdevé aproximadament de 11.38 · 10⁻⁶ segons, degut a la simplicitat de la classificació emprant un conjunt reduït de regles.

5.2 GENIFER-MDA

Recordem que GENIFER-MDA canvia la representació de les condicions de les regles, passant de l'ús de lògica matemàtica a l'ús de punts significatius. Això ha implicat modificar la representació dels individus i la funció d'avaluació, principalment.

A la figura 2 es pot observar el rendiment obtingut en les dues fases del sistema, l'aprenentatge i el test. Caldiria esmentar que el sistema no només ha guanyat en eficiència, sinó que a més ha guanyat en estabilitat, fet que es pot apreciar en l'augment del mínim obtingut en les proves realitzades.

El temps mitjà d'aprenentatge obtingut és aproximadament d'uns 16 minuts i 10 segons sobre la mateixa plataforma y les mateixes condicions, convergint vers la solució als 9 minuts. Comparant amb el GENIFER-BRE i els seus 90 minuts la reducció és apreciable.

Pel que fa a l'etapa de test, l'ordre de magnitud del temps de predicción d'una mostra és el mateix que el BRE.

5.3 GENIFER-MDAA

Passem als resultats obtinguts amb el GENIFER-MDAA. Recordem primer que el MDAA és com el MDA però amb una versió adaptativa de la funció de distància a utilitzar. Això permet que el propi GA es vagi adaptant a les circumstàncies.

A la figura 2 es poden observar els resultats obtinguts en les dues fases d'execució del sistema. Cal dir que amb aquests jocs de proves s'hagué pogut apreciar la capacitat adaptativa dels GA.

Pel que fa al temps d'execució del MDAA en front del MDA, les mesures realitzades evidencien mínimes diferències entre elles, tant a la fase d'aprenentatge com a la de test.

⁴Proves fetes sobre una estació Digital Alpha a 500MHz.
5.4 GENIFER-RA

I ja finalment passem a presentar els resultats del GENIFER-RA. Recordar abans que el RA recupera el MDA pel que fa a la funció de distància utilitzada, però incorpora el concepte d'atributs representatius. El GA ara haurà de trobar quins atributs són representatius i quins no, fent evolucionar un vector binari de pesos ω.

A la figura 2 es pot apreciar la comparació entre els resultats de les dues fases del sistema, l'aprenentatge i el test. Cal dir que amb aquests jocs de proves s'ha pogut apreciar la capacitat adaptativa dels GA per ajudar a discernir de forma automàtica quins són els atributs més representatius.

Pel que fa al temps d'execució del RA comparat amb el temps del MDA, les mesures realitzades demostraren que hi ha mínimes diferències, tant a la fase d'aprenentatge com a la fase de test.

5.5 Resum de resultats

Els resultats obtinguts durant la fase de train per les quatre aproximacions es poden veure a la taula 4 i a la figura 2. Cal observar que tots són aproximadament iguals. Pel que fa al percentatge màxim d'encerts aconseguit, les diferències són molt petites. En canvi, el rendiment mitjà i el rendiment mínim observat, és molt millor en les tres últimes aproximacions. Cal tenir present que el temps d'execució necessari per realitzar l'aprenentatge que ens ha portat a aquests resultats, és d'uns 9 minuts per les tres últimes aproximacions i de 90 minuts per la primera. Així doncs, assolim resultats iguals o millors en un temps decup còs inferior.

Pel que fa als resultats de l'etapa de test, es poden observar a la taula 5 i a la figura 2. En aquest cas el rendiment del primer sistema en front dels tres últims és força diferent. El percentatge d'encerts obtinguts millora substancialment, tant pel que fa al màxim (increment més lleuger), com pel que fa al mitjà i al mínim (aquest és el que presenta la millor més important).
6 Conclusions i treball futur

Les conclusions a les que hem arribat durant la realització d'aquest treball, es poden resumir en els punts següents:

- S'ha comprovat la viabilitat dels GA per solucionar problemes de classificació en els atributs descriptius són nombres reals.
- Les aproximacions presentades reduceixen fins a un 90% el temps d'execució del GA.
- El percentatge d'exactitud en el test ha millorat considerablement els resultats dels sistemes tradicionals, paliant també la diferència inicial entre l'aprenentatge i el test.
- Finalment, s'ha comprovat la capacitat dels GA adaptatius. Concretament, s'han ajustat paràmetres simultàniament a la cerca de la solució, millorant la convergència del mètode sense augmentar el temps d'execució.

A la taula 6 es pot trobar una breu comparativa entre diferents sistemes [10] [4] que resolen el problema de classificació presentat en aquest treball. Dels resultats es pot apreciar que els GA i el raonament basat en casos, de resultats equiparables, superen clarament els resultats obtinguts fins al moment emprant xarxes neuronals.

Les línies de treball en les que actualment ens estem centrant es poden resumir en els punts següents:

- S'està treballant en altres vessants de GA adaptatius. Per exemple, el la ponderació els pesos dels atributs.
- També s'estudien diferents propostes incrementals del GA. Això comportarà la integració de les fases de train i de test en una sola fase. Aquesta idea està inspirada en un successor del sistema GABL.
- I ja finalment, destacar que seria convenient provar les alternatives presentades sobre altres problemes de prova.

Agraïments

Voldrem agrair a Enginyeria i Arquitectura La Salle (Universitat Ramon Llull) el suport actiu que en els darrers anys ha vingut donant al nostre grup de recerca en Sistemes Intel·ligents.

Referències


Una plataforma de programació genètica (ModelGP) per a la construcció de models de predicción quantitativa

Gemma Calzada i Roca    Josep M. Garrell i Guix    Robert Barti i Domingo
Departament d’Informàtica    Departament d’Informàtica    Departament d’Acústica
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Passeig Bonanova 8 08022-Barcelona
{gemmac,josepmg,robert}@salleURL.edu

Resum

En aquest treball es presenta una plataforma de Programació Genètica (GP) per a la construcció automàtica de models de predicción quantitativa (ModelGP). La plataforma s’ha provat sobre un problema real: la construcció de models de predicción del nivell sonor de carrers. Els resultats obtinguts posen de manifest la viabilitat de la GP per a la construcció de models predictius, al mateix temps que soluciona el problema acústic plantejat.

Paraules Clau: algorismes genètics, programació genètica, modelització acústica de carrers, regressió simbòlica, predicción quantitativa.

1 Introducción

La construcció de models de predicción quantitativa és una àrea de gran utilitat en molts camps. Una de les possibles maneques d’afrontar aquest problema des de la Intel·ligència Artificial, és utilitzant Programació Genètica.

En aquest treball plantegem el disseny i la construcció d’una plataforma general de Programació Genètica (ModelGP). Aquesta plataforma havia de complir un conjunt de requeriments, com ara la transportabilitat i la eficiència. La plataforma havia de ser el més general possible, de cara a poder solucionar el màxim nombre de problemes d’aquest tipus. Finalment, i de cara a provar la potencialitat de la plataforma, s’ha abordat un problema dins d’un domini acústic. Es tracta de construir models que descriguin el nivell sonor de carrers a partir d’un conjunt de paràmetres descriptius. Aquests models no només haurien de permetre explicar les observacions realitzades, sinó que a més a més haurien de poder realitzar prediccions.

Aquest treball està estructurat en vuit seccions. La primera secció és aquesta introducció. Els Algorismes Genètics i la Programació Genètica s’introduïxen a la segona secció. A continuació, a la secció tercera, s’introdueix la plataforma que presentem (ModelGP). A la secció quarta es comenten els primers resultats obtinguts amb ModelGP sobre un banc de proves experimental. Tot seguit, a la secció cinquena, es planteja el problema que s’intenta solucionar. A la secció sisena s’introduïxen les adaptacions necessàries a ModelGP per solucionar el problema que plantegem. Finalment, a la secció setena es presenten els resultats obtinguts, i a la secció vuitena es comenten les conclusions a les que s’ha arribat i les línies de treball futur.

2 Els Algorismes Genètics i la Programació Genètica

Els Algorismes Genètics (GA) van ser introduïts per en Holland a mitjans dels anys setanta [6]. El model està inspirat en dos principis naturals: el principi d’evolució natural de les espècies de Darwin
Soft computing. Xarxes neuronals, sistemes difusos i algorismes genètics

Figura 1: Esquema bàsic de funcionament d'un GA.

3 ModelGP: Una plataforma per a la prediccio quantitativa basada amb GP

ModelGP és una eina per al desenvolupament d'aplicacions de modelització de sistemes reals. El tipus de sistemes o problemes al que va destinat, consisteixen en quantificar el valor d'una variable en funció d'una sèrie de paràmetres. La relació que existeix entre la variable independent i les variables dependents és desconeguda, i és el que pretén trobar en forma d'equació.

Prèviament el sistema a modelar s'ha hagut de sometre a l'experimentació per obtenir un conjunt de mostres que exemplifiquin el seu funcionament. Aquestes mostres es subministren a la plataforma dissenyada, i mitjançant el mètode de càlcul en què es basa la GP s'extrai l'equació que més s'aproxima als punts mesurats.

Amb el model disponible, serà possible realitzar predicccions sobre com es comportaria el sistema en noves situacions.

A continuació s'exposen les característiques del software dissenyat: la representació, els operadors genètics, i la implementació escollida.

\[1\] Val a dir que certs autors, i especialment Koza, consideren la GP una disciplina a banda dels GA, i no una variante.
Figura 2: Un exemple de la representació d’un individu en GP.

3.1 Representació: operands i ope-
radors de l’individu

Els individus es representen en forma d’arbre. L’arbre està format per un conjunt de nodes, que poden ser terminals, si es troben a les fulles de l’arbre, o funcions si són nodes interns. El valor de cadascun dels nodes (terminal o funcions) que apareixen en un individu, només poden ser un dels valors existents en el conjunt de terminals i el conjunt de funcions respectivament (figura 2). Aquest conjunt ha de complir la condició de la suﬁciència de poder formar la solució al problema i la de closure del tipus i del valor de les dades.

La condició de la suﬁciència consisteix en que el conjunt de terminals i funcions escollit ha de contenir tots aquells elements necessaris per a que es pugui formar la solució. Així, per exemple, si la solució ha de ser una relació de proporcionalitat, el conjunt de funcions haurà d’incloure el producte. És desitjable que el conjunt escollit sigui estricta-
ment suﬁcient, per tal de reduir al màxim l’espai de cerca. La condició de la closure exigeix que qualsevol funció pugui acceptar com a parametre, qualsevol valor i tipus retornat per una funció, i qualsevol valor i tipus de qualsevol terminal.

Per escollir el conjunt de terminals i funcions, tenint en compte que s’ha de satisfer aquestes dues condicions, cal tenir coneixement sobre el problema que es vol resoldre, i una idea sobre el tipus de solució que es busca.

Existeixen nombrosos variants sobre la forma d’implementació dels arbres que representen un individu. La implementació que s’ha escollit és l’aproximació prefixed based GP de Keith i Mar-
tin [7], on s’utilitza un array per enmagatzemar la informació corresponent a tots els nodes que formen un arbre genètic. S’ha realitzat un estudi comparatiu d’aquest mètode amb un mètode basat en punters, i s’ha deduït que el cost d’execució és el mateix en ambdues estructures. Amb una estructure basada en punters, s’afavoreix l’intercanvi de la informació en l’operador creament i mutació, s’osa en el cas de que els fills reemplacen als pares i la política de reproducció sigui steady state. Però resulta difícil codificar el punt de tall o de mutació. En canvi, en una estructura basada en array, es facilita la selecció del punt de tall, però cal realitzar una còpia de la informació quan es creen els fills resultants d’un creament. Referent a l’espai de memòria, és lleugerament menys costosa que la implementació en array, ja que la basada en punters s’ha d’enmagatzemar informació referent a l’estructura.

A partir de la proposta de Keith i Martin s’ha realitzat una variant per a millorar el temps d’execució. Per detectar el subarbre format a partir d’un node, Keith i Martin proposen realitzar la suma d’arits dels nodes recorreguts, i, quan aquesta adopta un cert valor (el -1), indica la fi del subarbre. Amb aquest mètode, no es pot conèixer a priori quina és la profunditat del subarbre format a partir d’un cert node. Això fa que en certs moments de l’algorisme, com per exemple, conèixer si es pot realitzar el creament entre dos individus donats degut a l’excés de la longitud màxima, no sigui possible a priori fins que no s’ha realitzat el recorregut de l’individu. En ModelGP, es coneix en tot moment la profunditat de l’arbre format a partir de qualsevol node, amb la qual cosa s’accelera el procés de detecció de possibilitat de creament entre dos individus.

La implementació en array també és més ade-
quada per a una reproducció del tipus generacional, on cal realitzar sempre nombroses còpies de la informació. La política de reproducció generacional manté dues poblacions simultàniament a memòria: la població actual i la nova població que s’està formant a partir de l’actual, en contra de la reproducció del tipus steady state. Aquesta última només manté una sola població que és una barreja d’individus pares i fills. Es va escollir la reproducció generacional perquè és la que ofereix millor rendiment, com va demostrar Alden [2] en els seus experiments sobre l’operador de selecció en el problema donut induction problem.
3.2 Operadors genètics

Els operadors genètics emprats per a fer evolucionar la població són:

Selecció

L'algorisme de selecció és el *fitness proportionate*, on la probabilitat de que un individu sigui seleccionat és directament proporcional al valor de la seva avaluació o *fitness*. El mecanisme utilitzat per implementar la selecció és el *roulette wheel selection* [11], on el nombre de cops que apareix un individu dins una ruleta és proporcional al seu *fitness*, i l'individu que surt en cada tirada passa a ser seleccionat com a part de la nova població.

Creuament i mutació

Els operadors creuament i mutació utilitzats segueixen el patró indicat per Koza [10]. A la figura 3 es pot veure un exemple de creuament.

*Constant Pertorbation*

Un altre operador que s'ha considerat important ha estat el de *constant perturbation* de Spencer [15] per a realitzar un ajust fàcil de les *ephenoral random constants* que formen part de les equacions solució. Aquest operador modifica el valor de les constants que apareixen en un individu escollit aleatoriament en un 1%.

Decimació

Per tal de millorar la diversitat de la població inicial, s'ha incorporat l'operador decimació de Koza [10]. La seva operació consisteix en sobredimensionar la població inicial en un 10% i escollir sols els millors individus per a formar la població de la primera generació.

3.3 La funció d'avaluació

La primera fase de l'algorisme de GP és l'avaluació. L'avaluació consisteix en trobar el *fitness* d'un individu. Aquest *fitness* es calcula avaluant l'individu (l'equació) per un conjunt de variables d'entrada. Llavors es compara el resultat obtingut amb el resultat esperat. Aquesta diferència és considerada com l'error puntual. El procés es repeteix per totes les mostres disponibles, i els error obtinguts es promitgen. El resultat és el *fitness* de l'individu.

Cal esmentar que l'avaluació s'ha optimitzat de forma que mai s'avalua més d'un cop un mateix individu. De les proves efectuades s'ha comprovat que aquesta optimització produeix una disminució del temps d'execució a la mitjana aproximadament.

3.4 Implementació de la plataforma *ModelGP*

La codificació del programa de programació genètica s'ha realitzat amb el llenguatge de programació C++, sota un entorn UNIX. La implementació s'ha realitzat de forma modular, implementant per separat la plataforma de programació genètica, de la part depenent de l'aplicació a la que pot ser aplicada.

La plataforma de programació genètica desenvolupada consta de les parts de l'algorisme de programació genètica que són comuns a qualsevol aplicació: les fases de l'algorisme, els operadors genètics, l'estructura de les dades d'una població, d'un individu i d'un node, i el procediment d'avaluació dels individus de forma genètica. La part que depèn de l'aplicació, i, per tant, no forma part de la plataforma, inclou la determinació dels tipus de nodes (funcions i terminals possibles), i com es calcula el *fitness* dels individus segons els tipus de nodes que el poden formar. També inclou un petit preprocessament de les dades subministrades per formar el banc de dades d'avaluació. Aquest tipus de disseny modular facilita la reutilització del *software* de programació genètica per a qualsevol altre tipus d'aplicació.
5 El problema de prova: plantejament

El problema que es vol resoldre consisteix en la generació automàtica d'un model descriptiu del nivell acústic d'un carrer d'una ciutat. Aquest nivell acústic es creu que depèn d'un conjunt de paràmetres, com ara l'amplada del carrer, el seu nivell de trànsit (descompost en diferents nivells segons el tipus de vehicles), l'algada dels edificis a banda i banda del carrer, el pendent, etc.

Des d'un punt de vista acústic, la recerca sobre un model que segueixi la realitat s'ha abordat de diferents maneres. El factor contú en totes elles és trobar el nivell de pressió acústica global en dBA. Si bé l'objectiu que es perseguix pot ser interessant, també és cert que carrers amb unes característiques molt diferenciades entre ells, ens donen uns nivells globals de pressió acústica molt similars. Per exemple, a la ciutat de Barcelona podem comparar el soroll de trànsit a la croïlla Balmes - Ronda del Mig (Balmes amb 5 carrils de circulació, i Ronda del mig amb 6 carrils de circulació). En hora punta a les 8h AM, la densitat de trànsit està al voltant dels 3.500 veh/h. El nivell obtingut és de 78 dBA. A la mateixa hora al carrer Major de Sarrià núm. 110, format per un sol carril de circulació, la densitat és d'uns 500 veh/h i el nivell mesurat és de 78 dBA. Tenim donc dos carrers força diferents i que ens donen el mateix resultat. La sensació de soroll o molèstia en un cas i altre són molt diferents. Cal, per tant, buscar un altre paràmetre que doni més informació que el nivell global de soroll en dBA.

L'anàlisi espacial en terços d'octava en pot ser la solució.

La construcció del model es vol fer de manera automàtica a partir d'un conjunt de mostres obtingudes en diferents carrers. Concretament es disposa de 144 mostres obtingudes en 12 carrers diferents. Cada mostra mostra el nivell acústic per cadascuna de les 30 bandes de terc d'octava. A la vegada, cada mostra inclou el valor dels atributs que descriuen el carrer en qüestió en el moment de la mesura. Es pot veure un resum dels atributs més representatius a la taula 1.

La construcció dels models es pot abordar des de dos punts de vista:

- La primera aproximació consisteix a construir un model que descriugui el nivell acústic total i octava a octava. És a dir, una sola equació ens donaria el nivell acústic per a qualsevol banda
Atributs
- Amplada del carrer
- Nombre de carrils
- Estacionament
- Direcció única/Doble direcció
- Pendent
- Tipus de ferm
- Activitat comercial
- Velocitat mitjana del trànsit
- Nombre de cotxes per unitat de temps
- Nombre de motors per unitat de temps
- Nombre d’autobusos per unitat de temps
- Temperatura
- Humitat

Taula 1: Alguns dels atributs que descriuen les característiques d’una mostra sonora d’un carrer.

deg de terç d’octava.

- La segona aproximació consisteix a construir un model per a cada una de les bandes de terç d’octava estudiades.

Ja a priori sembla clar que la segona aproximació ha de ser la més acurada. Com més endavant veurem, els resultats obtinguts avalen aquesta hipòtesi.

6 Solució proposada amb programació genètica

Per a provar la plataforma amb el problema de modelització acústica, es va desenvolupar la part dependent de l’aplicació: l’elecció del conjunt de terminals i funcions, i la funció d’avaluació. Es va buscar el conjunt de terminals i funcions mínim necessari per complir la condició de la suficiència i la closure. Val a dir que aquesta condició és crítica per l’eficient execució del sistema. El conjunt de funcions del que es va partir inicialment va ser les funcions aritmètiques suma, resta, multiplicació i divisió, la funció logarítmica, l’exponentzial, l’arrel nèssima i les trigonomètriques. Després d’un conjunt de proves es va trobar que el conjunt de funcions definitiu és:

\[ F = \{ +, -, \ast, /, \log \} \] (1)

El fet de que la resta de funcions no siguin necessàries és degut a que, en cap de les proves, aquestes operacions apareixen dins de l’equació solució. De la mateixa manera, del conjunt de terminals possible inicial, format per les variables corresponents a tots els paràmetres físics i ambientals d’un carrer, i la ephemeral random constant, es va decidir escollir com a conjunt de terminals final el següent:

\[ T = \{ \text{temp}, \text{humitat}, \#\text{cotxes}, \text{velocitat}, R \} \] (2)

On R és la ephemeral random constant. L’elecció es va realitzar en base a les variables que apareixien amb més freqüència a les equacions solució.

Referent a la funció d’avaluació, el conjunt de casos d’avaluació necessaris consisteixen en un conjunt de mostres de mesures acústiques realitzades a Barcelona. Un cas d’avaluació consta dels valors dels paràmetres físics i ambientals del carrer on es va realitzar una mesura, i el nivell en dBA del soroll mesurat. L’avaluació d’un individu es realitza substituint les variables corresponents als paràmetres físics i ambientals del carrer que apareixen en l’equació per un determinat cas d’avaluació, i s’executa el càlcul de l’equació que descriu aquell individu. El resultat que dóna es compara amb el que s’havia mesurat realment en el carrer, i la seva diferència dóna l’error per aquell cas d’avaluació. El procediment es repeteix per cadascun dels casos d’avaluació, i l’error mitjà entre tots ells és el fitness de l’individu.

7 Resultats obtinguts

Com ja s’ha dit, ModelGP s’ha sotmès a prova sota el banc de dades corresponent a mesures acústiques dels carrers de Barcelona. Els paràmetres que s’han analitzat per a discriminari el grau d’encert del software desenvolupat han estat els paràmetres d’error acústic absolut i el d’error de predicció. També és d’interès conèixer el tipus d’equacions que s’obtenen com a modelitzadors d’aquest sistema. Aquests valors s’exposen a continuació.

7.1 Els resultats de la modelització acústica

Les equacions resultants aconseguides són un total de 31, una per a cadascuna de les bandes de terç d’octava de l’espectre audible, una altra equació per descriure les 30 octaves amb una única equació, i dues equacions més per als paràmetres acústics A
(resposta subjectiva de l'òïda al nivell de soroll) i L (nivell equivalent de soroll). Una mostra del contingut de les equacions solució es donà a continuació amb la banda de terç d'octava número 15:

\[
Oc15 = \log(\text{humitat}) \times \log(\#\text{cotxes}) + 46
\]  \hspace{1cm} (3)

De cadascuna de les equacions obtingudes, s'han analitzat els següents aspectes:

- **Error mitjà**: és la mitjana de l'error que una equació produïx per cadascun dels casos d'avaluació o mostres. Es calcula com la suma dels errors produïts per cadascuna de les mostres dividit pel nombre de mostres. Es considera un error baix valors propers a 1 dBA.

- **Error màxim**: és el màxim dels errors que una equació produïx per tots els casos d'avaluació.

- **Error relatiu**: pretén quantificar l'error produït per una equació respecte el marge possible d'error que es podia cometre. Es calcula com la fracció entre l'error mitjà i la diferència entre el màxim i el mínim valor que podien adoptar les mostres. Per exemple, si el valor mínim que prenen totes les mostres és de 42, el valor màxim és de 73 i l'error mitjà és de 3.08, llavors l'error relatiu és de 3.5/(75 - 40) = 0.1. El marge de valors que pot prendre l'error relatiu va de 0 a 1. Es considera un error relatiu acceptable valors propers a 0.1, ja que això suposa un 10% d'error.

Els valors resultant de realitzar les proves han estat els següents.

L'error mitjà de predicció del soroll es manté en un marge de valors de 2.5 a 3.5 en la seva majoria. El valor màxim es produïx per a la primera banda de terç d'octava amb un valor de 6.04, i el mínim en la banda 18, amb un valor de 2.3 (Figura 4).

L'error màxim es troba en la seva major part entre 8 i 15. El valor màxim es dóna en la primera octava, que pren un valor de 20, i el mínim en l'octava 19, que val 7.95. (Figura 5).

L'error relatiu es troba en la seva major part distribuït en uns valors que van de 0.07 a 0.12, amb un màxim de 0.18 per a l'octava número 1 i un mínim de 0.05 per a l'octava número 30 que són valors excepcionals. Aquests errors es consideren força bons, ja que es troben al voltant del 10% d'error. És a dir, si es quantifica el valor possible que pot prendre l'error dins d'un interval que va des de 0 fins a 100, l'error que es comet és de 10. (Figura 6).
7.2 Cost temporal de la solució.

El temps global per a realitzar una execució del ModelGP per trobar una equació és variable per a cada execució. Aquest temps s’ha calculat com el temps d’execució de l’algorisme fins a arribar al criteri de finalització, que consistia en que es quedés estancat sense trobar una nova solució. En les proves realitzades fins al moment, el nombre de generacions necessàries anava d’unes pocs milers fins a centenars de milers, amb una mitjana de 50.000 generacions, la qual cosa suposa un temps mitjà d’1 hora sobre una estació Digital ALPHA 500MHz, i un màxim de 2.5 hores.

8 Conclusions i línies de futur

Les conclusions a les que hem arribat durant la realització d’aquest treball, es poden resumir en els punts següents:

- S’ha dissenyat i implementat una plataforma eficient de GP (ModelGP) per a la construcció de models de predicció quantitativa.
- S’ha observat el paper clau de la mutació per accelerar la convergència. Concretament, la mutació ha d’afectar, tant al valor dels nodes, com a la seva col·locació dins de l’individu.
- També s’ha observat la importància de l’ús d’ephemeral random constants.
- L’operador de decimació no ha donat uns resultats tant bons com s’esperaven.
- La condició de suïcència s’ha hagut d’aconseguir a base de sobredimensionar els operadors disponibles. El conjunt d’aquests operadors s’ha anat reduint gradualment.
- La representació dels individus està basada en la idea d’en Keith i d’en Martín amb una modificació que accelerà la velocitat d’execució de l’algorisme.
- Des del punt de vista acústic, la solució proposada és força esperançadora. S’observa un error mitjà elevat a les baixes i això ja que el valor de les primeres octaves no es significatiu acústicament parlant.

* * *

Aquest criteri es satisfà quan han passat un nombre de generacions igual al doble del nombre de generacions en el que es va produir l’últim canvi.
Les línies de treball en les que actualment ens estem centrant es poden resumir en els punts següents:

- Estudiar la viabilitat d’altres funcions d’avaluació que estimin millor la bondat de les solucions.
- Pel que fa a l’ús d’ephemeral random constants, caldria estudiar la possibilitat de barrejar aquestes idees amb algun métode d’optimitzacíó clàssic, és a dir, crear un sistema híbrid. La GP troba l’estructura, però li és molt més difficults ajustar les constants finals.
- Tot i que es van analitzar diferents carrers, a diferents hores i dies, cal crear una base de dades molt més amplia que contempli no solament les variacions de les característiques físiques dels carrers, sinó també del trànsit. L’objectiu serà arribar a obtenir un error per a cada banda inferior a 1 dB.

Agraïments

Voldrem agrair a Enginyeria i Arquitectura La Salle (Universitat Ramon Llull) el suport actiu que en els darrers anys ha tingut donant al nostre grup de recerca en Sistemes Intel·ligents.

Referències

Fuzzy Heterogeneous Neurons for Imprecise Classification Problems

Julio J. Valdés  Lluís A. Belanche  René Alquézar
Secció d’Inteligència Artificial.
Dept. de Llenguatges i Sistemes Informàtics.
Universitat Politècnica de Catalunya.
c/Jordi Girona Salgado 1-3
08034 Barcelona, Spain.
{valdes, belanche, alquezar}@lsi.upc.es

Abstract

In the classical neuron model, inputs are continuous real-valued quantities. However, in many important domains from the real world, objects are described by a mixture of continuous and discrete variables, usually containing missing information and uncertainty. In this paper, a general class of neuron models accepting heterogeneous inputs in the form of mixtures of continuous (crisp and/or fuzzy) and discrete quantities admitting missing data is presented. From these, several particular models can be derived as instances and different neural architectures constructed with them. Such models deal in a natural way with problems for which information is imprecise or even missing. Their possibilities in classification and diagnostic problems are here illustrated by experiments with data from a real world domain in the field of environmental studies. These experiments show that such neurons can both learn and classify complex data very effectively in the presence of uncertain information.

Keywords: Heterogeneous Neural Networks; Fuzzy Logic; Genetic Algorithms; Uncertainty and Missing Data.

1 Introduction

The classical neuron model—where inputs are continuous real-valued quantities, and net input is computed as the scalar product of the input and the weight vector—was extended by the notion of an heterogeneous neuron introduced in [14]. Model instances of this concept provide neurons accepting mixtures of real and discrete quantities (i.e. numerical and qualitative information) possibly also containing missing data. A second important feature of this class of models is that their internal stimulation is based on a similarity or proximity relation [5] between the input and the weight tuples. In particular, a family of models considering neuron outputs as a composition of a similarity function with a sigmoid-like squashing function was shown to be a reasonable brick for constructing layered network architectures mixing heterogeneous with classical neurons. These networks were shown to be capable to learn from non-trivial data sets with an effectiveness comparable, and sometimes better, than that of classical methods. They exhibited a remarkable robustness when information degrades due to the increasing presence of missing data. As was stated in [14] when discussing the general framework for heterogeneous neurons, future studies should consider more general models constructed by taking into account mappings among wider classes of sets as domains and images (i.e. inputs and output). The purpose of this paper is to consider extensions in which fuzzy sets may occur as part of the input. This will introduce more flexibility by accepting training processes using imprecise data, both in the input and the weights. This way, heterogeneous neurons of this kind may accept a mixture of real, qualitative, and fuzzy quantities,
possibly with missing information. In what follows, a neuron of this type is presented and its learning capabilities are illustrated by a real world example: an environmental study—using geophysical data processing—aimed at detecting the presence of underground caves.

2 The Heterogeneous Neuron Model Revisited

An heterogeneous neuron was defined in [14] as a mapping \( h : \mathcal{H}^n \rightarrow \mathcal{R}_{out} \subseteq \mathbb{R} \), satisfying \( h(\emptyset) = 0 \) (\( \emptyset \) is the empty set). Here \( \mathbb{R} \) denotes the reals and \( \mathcal{H}^n \) is a cartesian product of an arbitrary number of source sets. Source sets may be extended reals \( \mathcal{R} = \mathbb{R} \cup \{ \mathcal{X} \} \), and/or finite sets of the form \( \mathcal{O}_i = \mathcal{O}_i \cup \{ \mathcal{X} \} \), \( \mathcal{M}_i = \mathcal{M}_i \cup \{ \mathcal{X} \} \). Each of the \( \mathcal{O}_i \) has a full order relation, while the \( \mathcal{M}_i \) have not. The special symbol \( \mathcal{X} \) denotes the unknown element (missing information) and it behaves as an incomparable element w.r.t. any ordering relation. According to this definition, neuron inputs are possibly empty arbitrary tuples, composed by \( n \) elements among which there might be reals, ordinals, nominals and missing data.

A particular class of heterogeneous neurons can be devised by considering \( h \) as the composition of two mappings, \( h = f \circ s \), such that \( s : \mathcal{H}^n \rightarrow \mathcal{R}^n \subseteq \mathbb{R} \) and \( f : \mathcal{R}^n \rightarrow \mathcal{R}_{out} \subseteq \mathbb{R} \). The mapping \( h \) can be considered as a \( n \)-ary function parameterized by a tuple \( \hat{w} \in \mathcal{H}^n \) representing the neuron’s weights, i.e. \( h(x, \hat{w}) = f(s(x, \hat{w})) \). In particular, the function \( s \) represents a similarity and \( f \) a squashing non-linear function with its image in \([0, 1] \). Accordingly, the neuron is sensitive to the degree of similarity between the input and weight tuples, composed in general by a mixture of continuous and discrete quantities possibly with missing data. More precisely, \( s \) is understood as a similarity index, or proximity relation (transitivity considerations are put aside), that is, a binary, reflexive and symmetric function \( s(x, y) \) with image on \([0, 1] \), and such that \( s(x, x) = 1 \) (strong reflexivity). The semantics of \( s(x, y) > s(x, z) \) is that object \( y \) is more similar to object \( x \) than \( z \) is. An instance of this model uses as \( s \) function Gower’s similarity index [11]. This coefficient has its values in the real interval \([0, 2] \) and, for any two objects \( i, j \) given by tuples of cardinality \( n \), is given by the expression

\[
s_{ij} = \frac{\sum_{k=1}^{n} g_{ijk} \delta_{ijk}}{\sum_{k=1}^{n} \delta_{ijk}}
\]

where:

- \( g_{ijk} \) is a similarity score for objects \( i, j \) according to their value for variable \( k \). These scores are in the interval \([0, 1] \) and are computed according to different schemes for numeric and qualitative variables. In particular, for a continuous variable \( k \) and any two objects \( i, j \) the following similarity score is used:

\[
g_{ijk} = 1 - \frac{|v_{ik} - v_{jk}|}{\text{range} (v_k)}
\]

Here, \( v_{ik} \) denotes the value of object \( i \) for variable \( k \) and

\[
\text{range} (v_k) = \max_{i,j} (|v_{ik} - v_{jk}|)
\]

(see [11] for details on other kinds of variables).

- \( \delta_{ijk} \) is a binary function expressing whether both objects are comparable or not according to their values w.r.t. variable \( k \). It is \( 1 \) if and only if both objects have values different from \( \mathcal{X} \) for variable \( k \), and \( 0 \) otherwise.

As for the activation function, a modified version of the classical sigmoid is used, such that it maps the real interval \([0, 1] \) on \((0, 1) \).

\[
f(x, p) = \begin{cases} 
\frac{(x-0.5) - a(p)}{(x-0.5) + a(p)} - a(p) & \text{if } x \leq 0.5 \\
\frac{(x-0.5) + a(p)}{(x-0.5) - a(p)} + a(p) - 1 & \text{otherwise}
\end{cases}
\]

\[
a(p) = -0.5 + \sqrt{0.5^2 + 4 \times p} \over 2
\]

where \( a(p) \) is an auxiliary function and \( p > 0 \) is a real-valued parameter controlling the curvature, usually set in the experiments to 0.1.

2.1 A Fuzzy Extension

A step forward in generalizing the previous specific model is a relaxation of real valued inputs, by considering more flexible situations, now tolerating imprecision. According to the conceptual setting of the family of neuron models studied based on similarity, it is natural to state a fuzzy extension following the same approach. Similarity relations from
the point of view of fuzzy theory have been defined elsewhere [12], [16]. In the present case, the situation is not that of a fuzzy similarity or proximity relation defined on real values, but a relation between fuzzy entities. Let $F_i$ be a family of normalized fuzzy sets from the source set and $A, B \in F_i$ two fuzzy sets. The following similarity relation is used:

$$g(A, B) = \max_x (\mu_A \wedge \mu_B (x))$$

where

$$\mu_A \wedge \mu_B (x) = \min (\mu_A (x), \mu_B (x))$$

Clearly it is reflexive in the strong sense and also symmetric. This is a proximity relation and can be used to include extra fuzzy components in Gower’s similarity. Consider a collection of $n_f$ extended fuzzy sets of the form $F_i = F_i \cup \{x\}$ and their cartesian product $\hat{F}^{n_f} = F_1 \times F_2 \times \ldots \times F_{n_f}$. The resulting input set will then be $\hat{R}^n =< \hat{R}^{n_r}, \hat{R}^{n_o}, \hat{R}^{n_m}>$, where the cartesian products for the other kinds of source sets ($R_i, O_i, M_i$) are constructed in a similar straightforward way from their respective cardinalities $n_r, n_o, n_m$, with $\hat{R}^0 = \hat{R}^0 = \hat{R}^0 = \hat{R}^0 = \phi, n = n_r + n_o + n_m$ and $n > 0$.

![Diagram](image)

Figure 1: A fuzzy heterogeneous neuron.

The training procedure for the resulting heterogeneous neuron—shown in Fig. 1—is based on genetic algorithms ([10], [6]) and can be devised in a natural way by extending that used for heterogeneous neurons without fuzzy inputs or weights [14]. In this extension, each fuzzy weight is characterized as a tuple of reals (instead of a single one) and this only needs a chromosome enlargement, depending on the chosen functional representation for fuzzy sets (trapezoidal, Gaussian or LR).

3 An Example of Application in an Imprecise Domain

An environmental investigation in the tropics dealing with the detection of underground caves using geophysical measurements made at the surface of the earth was used to experiment with the extended approach described in the previous section. First, some words describing the problem are necessary.

*Karstification* is a peculiar geomorphological and hydrogeological phenomenon produced mostly by rock solution as the dominant process. As a consequence, earth’s surface is covered by exotic irregular morphologies, like lapiaz, closed depressions (dolinas), sinks, potholes and the like, with the development of underground caves. This implies that the surface drainage network is usually poorly developed or simply does not exist at all, while vertical infiltration of rain waters forms an underground drainage system where water flows through fissures, galleries and caves. The studied area is located 30 km to the south of Havana City (Cuba) in the so-called Havana-Matanzas Karstic Plain composed of porous, fractured and heavily karstified limestones of Middle Miocene age with abundance of a variety of clay minerals. Under the high temperatures and humidity typical of tropical conditions, weathering processes develop an overburden composed by reddish insoluble materials (terra rossa) coming from solution processes on the limestones.

Negative karst forms on the surface (the lapiaz, sinks, dolinas, etc.) are partially or totally covered by an overburden of variable depth. These forms often connect with caves in the underground, some of them big. Direct detection is very difficult or impossible and geophysical methods are necessary, as they usually are for tasks like geological mapping and construction of cross sections. This is a very important problem from the point of view of civil engineering, geological engineering and environmental studies in general in this kind of regions.

In a selected square area (340 m side), geophysical methods complemented with a detailed topo-
graphic survey [13] were used with the purpose of characterizing the shallower horizons of the geological section and their relation with underlying karstic phenomena. Targets were zones of intense fracture and karstification, filled depressions, overburden pockets and the presence of underground caves. The set of geophysical methods included the spontaneous electric potential of earth's surface, the gamma radioactive intensity and the electromagnetic field in the VLF region of the spectrum [13]. In particular, two different surveys of spontaneous electric potential were performed, in the dry and rainy season respectively, since strong negative anomalies are due to infiltration potentials associated with electrochemical processes taking place as water infiltrates into the underground via fissures and joints. These four measurements, along with the surface topography, constitute the five variables to be used by the neural models. The complexity of these measured geophysical fields in the area is illustrated, as an example, by the distribution of gamma ray intensity and the surface topography. While radioactivity is highly noisy, topography shows few features. Both are shown in Figs. 2, 3.

Figure 2: Distribution of gamma ray intensity in the studied area.

Geophysical survey methodologies consider independent sets of measurements in order to account for different kinds of errors and the natural variability of such kind of information. In order to be considered acceptable, each survey must have an error no greater than 5% when comparing the original and the independent measurements. This means that the reported values of all geophysical fields (i.e., the available data), have an inherent uncertainty which must be considered. In the area, a gentle variation in geological conditions for both the bedrock and the overburden was suspected by geologists and also a large underground cave with a single gallery was known to exist in the central part of the area. The cave has about 300 meters long with cross sections ranging from less than one square meter in the narrowest part, to chambers having 40 meters wide and 30 meters high, reaching the surface in the form of a gash in the bottom of a depression.

Figure 3: Surface topography of the studied area.

An isolation of the different geophysical field sources was necessary in order to focus the study on the contribution coming from underground targets, trying to minimize the influence of both the larger geological structures, and the local heterogeneities. According to the a priori geological ideas, each geophysical field was assumed to be described by the following additive two-dimensional model composed by trend, signal and random noise:

\[ f(x, y) = t(x, y) + s(x, y) + n(x, y) \]

where \( f \) is the physical field, \( t \) is the trend, \( s \) the signal, and \( n \) the random noise component, respectively. In order to isolate an approximation of the signals produced by the underground target bodies, a linear trend term \( t'(x, y) = c_0 + c_1 x + c_2 y \) was computed (by least squares) and subtracted from the original field. The residuals \( r(x, y) = f(x, y) - t'(x, y) \) were then filtered by direct convolution with a low pass finite-extent impulse response.
two-dimensional filter in order to attenuate the random noise component [9]. Such convolution is given by:

$$s'(x, y) = \sum_{k_1=-N}^{N} \sum_{k_2=-N}^{N} h(k_1, k_2) \cdot r(x-k_1, y-k_2)$$

where $r(x, y)$ is the residual, $s'(x, y)$ is the signal approximation and $h(k_1, k_2)$ is the low-pass zero-phase shift digital filter.

4 Experiments

In order to study the behavior of these neural models, a comparison was made w.r.t. geological-geophysical accuracy of classification. This kind of knowledge, as well as results from previous non-supervised classification techniques [15] had shown the existence of two multivariate populations within the studied area: one representing more karstified zones with large interconnected underground cavities, and another in which karstification is not so intense. Since the hypothesis of two hyparspherical classes in pattern space was tenable, and the purpose of this work is to assess the relative merits of the three considered neuron models (classical, heterogeneous and fuzzy heterogeneous) in the task at hand (imprecise classification using data which are also imprecise), a network consisting of a single neuron was the architecture selected. Clearly, other multilayer layouts are possible and should deserve future attention, but is a good reference for initial comparisons. This, together with the small training set (relative to test), should make the problem much more difficult than it really is, so the differences should be more evident.

The experiments were conceived in two phases as follows. In phase one, a comparison is made between the classical real neuron with the heterogeneous one with real inputs and weights. In a second stage, the latter is compared to the fuzzy heterogeneous neuron. Also, the experiments were designed following geological criteria. From this point of view it is known that the number of observable caves in any karstic area is only a small fraction of the actually existing ones, making class structure itself imprecise, a situation usual in complex problems like those from environmental studies. Moreover, there are no sharp boundaries between rock volumes containing caves and those containing less or none. One could say that the notion of "caveness" degrades smoothly, which is another reason to use fuzzy models.

Figure 4: The known cave borders: see text for an explanation of what is considered as cave and what is not. Dots indicate the (approximate) location of the points used for training.

Figure 5: Results of phase 1: $\alpha$-cut sets for the classical neuron.

The training was supervised (in the usual mean squared error sense) by the information given by the topographic map of a large cave present in the area, so that those surface measurement points lying exactly above the known cave were considered as
class 1 patterns and those outside as belonging to class 2 (the resulting cave is shown in figure 4). This procedure for class assignment was too conservative but, otherwise, one would have been forced to provide as output the exact caveness degree for each point. This value, besides being very difficult to estimate, would have introduced a strong subjective bias. The computation of this degree is precisely the task we want the model to perform.

Selected data from the northern half were used for training, whereas the rest was used for testing the trained network (consisting of a single neuron only). More precisely, the training set was composed by the 31 points from the northern half located exactly above the known cave (representing class 1), plus 32 others homogeneously distributed in the east-west sides. As test set we used the remaining 567 patterns from the whole area (it est, train = 10%, test = 90%).

4.1 Phase 1

Here we have a classical real-valued neuron (in this study, having scalar product as net input and hyperbolic tangent as a squashing activation function). The training procedure for this neuron is a combination of conjugate gradient with simulated annealing [1], whereas the heterogeneous neuron is trained using a standard genetic algorithm with the following characteristics: binary-coded values, probability of crossover: 0.6, probability of mutation: 0.01, number of individuals: 50, linear rank scaling with factor: 1.5, selection mechanism: stochastic universal, replace procedure: worst.

The results obtained by both models are shown in figures 5 and 6, respectively, where caveness prediction is plotted in five equally spaced α-cut sets. Clearly, the distribution of the two-dimensional sets for the heterogeneous neuron reflects much better the distribution of the known cave than the classical neuron, for various reasons. First, the classical neuron fails to detect the southernmost part of the known cave, whereas the heterogeneous counterpart does.

Second, the classical neuron predicts complete cave areas in the south-east and south-west zones, which are misleading. These are also signaled by the heterogeneous neuron, but always with a degree of 0.5 or less. The only exception is a small area located in coordinates (7 – 8, 12 – 15), where other geophysical methods (seismic and DC-resistivity) not used in this study had signaled cave anomalies. And third, the general layout of the actual cave (north-south main axis, slightly bended and narrower in the middle part) is better reflected by the heterogeneous neuron.

4.2 Phase 2

In a second stage, a fuzzy heterogeneous neuron was trained in the same experiment setting, but this time using fuzzy inputs. This means that all neuron weights were fuzzy sets (actually triangular fuzzy numbers), and both training and test vectors represented by fuzzy numbers (the mode was given by the corresponding observed value, and the spread a ±5% of it). This is in accordance with the upper bound of the measurement errors reported for the geophysical field surveys made. It should be noted that this criteria was conservative, since some surveys actually had less than 5% of error.

The results (shown in figure 7) are again qualitatively satisfactory, in what regards to the general layout of the cave. But now a quantitative factor comes into the picture: the cave is much more neatly defined, a fact that shows in two ways: first, the different α-cut sets are much closer, showing a gradual but firm transition from class 0 to class 1 of 2 units in the map on average (roughly 20m in the field) – a very reasonable value. That is, this narrow belt w.r.t. the trace of the known cave represents the transition zone between the rock vol-
Source. Other realizations of these models have been presented elsewhere, in which their possibilities are further explored, ranging from classification benchmarking [14], time-series prediction [2], [3] and system identification [4]. In the work presented, experiments made with complex multivariate space-dependent data—coming from a real world problem in the domain of environmental studies—have shown that allowing imprecise inputs and using heterogeneous fuzzy neurons based on similarity yields models more accurate (because of their greater flexibility) than those from classical crisp real-valued models, in a problem for which one is not so much interested in crude train/test set classification errors (which could well have been presented) but in its ability to model the imprecise structure of the domain. This represents only a preliminary although promising class of models that serves as an initial standpoint which is deserving further investigation.

References


Economic Series Analysis through Independent Components

Francesc Miralles
mirallesf@esade.es
ESADE - Universitat Ramon Llull
Av. Pedralbes 60-62
08034 Barcelona (Spain)

Carmen Ansótegui
ansotegui@esade.es
ESADE - Universitat Ramon Llull
Av. Pedralbes 60-62
08034 Barcelona (Spain)

Abstract

Independent Component Analysis has been widely used in Signal Processing and Automatic Control applications. A new application in Economics is reported. Independent Components have been found for economic indexes using the neural approach by Jutten & Hérault. Independent Components offer a different view for analysis of economic series.

Moreover, the Arbitrage Pricing Theory (APT) has been used to estimate security rates of return of capital markets. In this paper we also report the use of Independent Components to implement a model based on the APT.

Keywords: independent components, INCA, APT, principal components, neural networks, economic indexes, arbitrage theory.

1 Introduction

Independent Component Analysis (INCA) has been defined by P. Comon [4] as the “search for a linear transformation that minimizes the statistical dependence between its components”. INCA has been used for Blind Source Separation and other applications in Signal Processing ([4],[11]) and Automatic Control. INCA was first proposed by Jutten and Hérault in the mid-80’s as an extension of Principal Components Analysis (PCA). In our work the neuromimetic adaptive network by Jutten and Hérault ([3], [11]) has been used to implement an independent component analyser in order to find independent components of economic series and to estimate differences of return for different assets.

In this paper we report use of INCA to find components with a higher than two order of independence of some economic series [12]. The underlying idea for that application is that economic indexes are variables we are able to measure. These variables show the behaviour of the economic determinants which we are interested in ([13]).

In the second part of this paper use of Independent Components as the basis to implement a model based on the Arbitrage Pricing Theory (APT) is reported. APT explains differences of rates of return for different securities in capital markets [5] and states that rates of return are linearly related to a set of common factors. Generally, PCA has been used to implement APT models [5]. The assumption underlying these results is that independent components can bring a more reliable interpretation of the components that govern the APT model.

The so-called ‘source-separation’ problem manages to obtain signals coming from statistically independent sources that have been mixed through a linear transformation. If we receive a n-dimensional vector of signals $\mathbf{x}$, it is assumed that they have been generated from a linear transformation like

$$\mathbf{x} = \mathbf{Hs}$$

where matrix $\mathbf{H}$ is an unknown $nxn$ matrix and $\mathbf{s}$ is a n-dimensional vector of unknown original signals. The assumptions we are allowed to resort to are only the linearity of the transformation and the independence of the components of vector $\mathbf{s}$. The goal of INCA methods is to retrieve the unknown source signals from $\mathbf{x}$ by using solely their statistical independence. If a solution is obtainable, it is necessarily given by a linear transform like

$$\mathbf{x} = \mathbf{A}\hat{\mathbf{g}}$$

where matrix $\mathbf{A}$ and vector $\hat{\mathbf{g}}$ are the goals of the source separation method. This kind of approach has been solved using Principal Components analysis ([14]). The solution given by the principal components analysis provides uncorrelated signals each one with the maximum variance. Moreover, if input signals are normally distributed we get independent components. INCA methods do not require a normally distributed multidimensional signal and they provide techniques that get a greater than two level of independence between the output components.
The term source is not used in economics and other fields of social sciences. Instead of looking for the sources, in these contrast functions ([4]). Results in the work we are reporting have been obtained using the neural network method.

2 Adaptive Neural Network

The adaptive neural network to obtain independent components was proposed by Jutten and Hérault in the mid-80's ([10], [11], [3]). Network architecture is based on a non-supervised, self-organizing neural network ([7], [9]). The

Figure 1 Evolution of a coefficient during the learning period

Figure 3 Evolution of a coefficient during the learning period

Independent component methods have been included under the work on higher-order statistics ([1]). Some of the implementations of INCA methods use an adaptive neural networks ([10], [11], [3]), a four-order cumulants approach ([2]), a maximum-likelihood approximation ([8]) and a learning algorithm for the network coefficients uses higher-order moments that enables one to obtain the independent components. In figure 1, there is a diagram of the neural network. Each big dot on the left side of the diagram means the product of the output signal by the coefficient. All the
products are added up to produce the input for the unit operator (triangular element) (see [3] and [11]).

The mathematical expression for that model is

\[ \hat{x}_i = x_i - \sum_{j=1, j \neq i}^{n} c_{ij} \hat{x}_j \]

also, using a matrix notation

\[ \hat{x} = x - C \hat{x} \]

Intuitively, the adaptive procedure to get the separation subtracts from each input signal a fraction of each output signal. Coefficient \( c_{ij} \) gives the value for the fraction between output \( j \) and input \( i \). The goal of the learning algorithm is to get the values for the coefficients that isolate the independent components from each input signal. The independence level is maximised by selecting the appropriate parameters in the learning algorithm [3].

### Table 1 Correlation Coefficient between Economic Indexes and Principal Components

<table>
<thead>
<tr>
<th>Economic Indexes</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.P. Morgan Bond Index</td>
<td>0.8824</td>
<td>-0.2589</td>
<td>0.3846</td>
<td>-0.0806</td>
</tr>
<tr>
<td>Dollar/PTA exchange rate</td>
<td>-0.9417</td>
<td>-0.2245</td>
<td>-0.0365</td>
<td>-0.2480</td>
</tr>
<tr>
<td>Madrid stock market Index</td>
<td>0.8417</td>
<td>-0.4303</td>
<td>0.1262</td>
<td>-0.0444</td>
</tr>
<tr>
<td>PTA/ECU exchange rate</td>
<td>0.8757</td>
<td>0.4330</td>
<td>-0.1132</td>
<td>-0.1812</td>
</tr>
</tbody>
</table>

The learning algorithm for the coefficients must meet two conditions. First of all, the coefficients must reach a stability point. Secondly, in that point a maximum for the level of independence of components should be attained. It is not our aim to describe that algorithm here, the interested reader can refer to [3]. In figures 2 and 3 the evolution of two coefficients during the learning period have been drawn.

### 3. Independent Components of Economic Series
From our point of view, economic indexes are the observable variables that display the evolution of the Economy. These indexes can be considered as signals managed by a linear mixture of latent - non-measurable - independent components. These independent components are assumed to be the determinants of the economic forces.

Principal Components are not adequate [11] to obtain the underlying components of a non-gaussian linear mixture of signals. Independent Components offer the statistically independent traits that generate the signals of the analysis. Figure 5 and Table 2 show the correlation coefficient between indexes and independent components. The neuromimetic adaptive architecture by Jutten & Hérault, [11] and [3], has been used. The sample was replicated to get a sample large enough for convergence of INCA techniques.

In Figure 5 and Table 2 we can see that each independent component represents a specific economic trait. The second and fourth independent components are related to foreign exchange, they are highly correlated to exchange rates.

<table>
<thead>
<tr>
<th>Economic Indexes</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.P. Morgan Bond Index</td>
<td>0.6007</td>
<td>-0.3977</td>
<td>0.2578</td>
<td>0.6437</td>
</tr>
<tr>
<td>Dollar/peso exchange rate</td>
<td>0.0197</td>
<td>0.7284</td>
<td>-0.3803</td>
<td>-0.6925</td>
</tr>
<tr>
<td>Madrid stock market Index</td>
<td>0.2261</td>
<td>-0.1558</td>
<td>0.8392</td>
<td>0.5310</td>
</tr>
<tr>
<td>peso/ECU exchange rate</td>
<td>-0.2015</td>
<td>-0.4030</td>
<td>0.2702</td>
<td>0.9329</td>
</tr>
</tbody>
</table>

Table 2 Correlation Coefficient between Economic Indexes and Independent Components.

4 APT

Firstly formulated by Ross [15], the arbitrage pricing theory (APT) [6] is an arbitrage model that explains the difference
of return for different assets. The APT assumes that the rate of return on any security is a linear function of $k$ factors,

$$ R_i = \mathbb{E}(R_i) + \sum_j b_{ij} F_j + \epsilon_i $$

where $R_i$ is the rate of return of security $i$, $\mathbb{E}(R_i)$ is the expected value of the rate of return, $b_{ij}$ is the coefficient for factor $j$, $F_j$ is factor $j$ and $\epsilon_i$ is the noise associated with security $i$.

The APT [6] is derived under usual assumptions of perfectly competitive and frictionless capital markets. Furthermore, individuals are assumed to have homogenous beliefs that the random returns for the set of assets being considered are governed by the linear $k$-factor model given in the equation above. The theory requires that the number of assets under consideration, $n$, be much larger than the number of factors, $k$, and that the noise term, $\epsilon_i$, be the unsystematic risk component for the $i$th asset. It must be independent of all factors and all error terms for other assets.

APT was applied by Connor and Korajczyk using an approximated model based on PCA [5]. In our work APT has been applied under the point of view of Independent Components. The idea underlying this approach is that independent components can offer more possibilities in the interpretation of the components ([12], [13]).

5 INCA model for APT

Monthly values for the rate of return of 155 assets from the Madrid stock market have been considered for analysis. A reduction of the dimension of the asset space must be performed to apply the INCA techniques and to include only the relevant information. In order to determine the significant components, eigenvalues of the covariance matrix were obtained. Two out of 155 values gave 30% of the overall variance. The rest of the eigenvalues produced from 5% to 0% of the variance. These eigenvalues were considered to represent noise.

Two principal components were obtained and initial signals were projected on the space of these components. The new signals are the representation of the rates of return of the assets on the space of the significant components. In Table 3 correlation coefficients between the principal components and the most relevant economic indexes are shown. The first principal component is highly correlated with market index (REM) and weighted market index (REE). On the other hand, the second principal component is correlated with long term interest rate (RFJADP) and short run interest rate (RFR).

To obtain the independent components the neuromimetic approach was used. The learning of the coefficients of the neural network required the replication of the sample in order to get enough data for the convergence of the adaptive algorithm. A range of values for the learning constant (learning parameter [3]) were tested to optimise convergence rate.

6 Results

The main result is that independent components allow more possibilities in the interpretation of the factors that govern rates of return of the securities. In addition, independent components make more significant the specific traits that can be isolated from the input signals.

Economists are concerned about the meaning of the components and they identify the components with market or economic indexes. In figures 6 and 7 correlation coefficients between the two components (P.C.: principal component, I.C.: independent component) and three economic indexes (market index, weighted market index and long term interest rate) are shown.

In a further analysis, independent components of market indexes were found. Figure 8 shows the correlation coefficients of these components with the signals themselves and in Table 4 correlation coefficients between these

<table>
<thead>
<tr>
<th>Index</th>
<th>1st</th>
<th>2nd</th>
</tr>
</thead>
<tbody>
<tr>
<td>REM</td>
<td>0.9293</td>
<td>-0.1524</td>
</tr>
<tr>
<td>REE</td>
<td>0.9734</td>
<td>-0.1789</td>
</tr>
<tr>
<td>RFJADP</td>
<td>0.0499</td>
<td>0.2969</td>
</tr>
<tr>
<td>REALTB</td>
<td>-0.1570</td>
<td>0.0424</td>
</tr>
<tr>
<td>RFR</td>
<td>0.0580</td>
<td>0.2946</td>
</tr>
<tr>
<td>PREM</td>
<td>-0.0193</td>
<td>0.0084</td>
</tr>
<tr>
<td>SLOPE</td>
<td>0.0483</td>
<td>0.1366</td>
</tr>
<tr>
<td>TC</td>
<td>-0.0385</td>
<td>0.0153</td>
</tr>
<tr>
<td>IPB0</td>
<td>-0.0867</td>
<td>0.0680</td>
</tr>
</tbody>
</table>

Table 3 Correlation Coefficients between economic indexes and principal components.

The following economic indexes have been used: REM: market index, REE: weighted market index, RFJADP: long term interest rate, REALTB: real interest rate, RFR: short run interest rate, PREM: risk spread of interest rates, SLOPE: term spread of interest rates, TC: exchange rate, IPB0: industrial production index.

1 The following economic indexes have been used: REM: market index, REE: weighted market index, RFJADP: long term interest rate, REALTB: real interest rate, RFR: short run interest rate, PREM: risk spread of interest rates, SLOPE: term spread of interest rates, TC: exchange rate, IPB0: industrial production index.
7. Conclusions

Independent components are a new approach to component analysis, shifting the viewpoint from one of variance to a vision based on the differential features of the variables or signals analyzed. In economics, component models have been used to represent the behaviour of share earnings.

This paper applies independent component techniques to several macroeconomic indexes. The results reveal that these techniques provide a new perspective.

Independent component analysis is also applied to an APT model to estimate the security rates of return of the capital markets. The analysis is based on a model which allows us to isolate the independent components that might appear in a significant series of share earnings. The independent components obtained are then analyzed in terms of economic indexes that describe the behaviour of the economy.

These findings open the door to the use of independent component techniques in fields other than engineering, specifically the field of economics. Moreover, they examine ways in which independent components could be used to analyze stock market returns and other macroeconomic indexes.

Bibliography


Figure 6 Correlation coefficient between the first component and the economic indexes

Figure 7 Correlation coefficient between the second component and economic indexes

Figure 8 Correlation coefficient between independent components of market indexes and the market indexes themselves
Figure 9: Correlation coefficient between independent components of market indexes (MI) and those of securities rates of return (RR).

Table 4: Correlation coefficients between economic indexes and independent components of the market indexes.

<table>
<thead>
<tr>
<th>Indexes</th>
<th>1st</th>
<th>2nd</th>
</tr>
</thead>
<tbody>
<tr>
<td>REM</td>
<td>0.7874</td>
<td>0.6403</td>
</tr>
<tr>
<td>REE</td>
<td>0.5015</td>
<td>0.8801</td>
</tr>
<tr>
<td>RFJADP</td>
<td>0.1456</td>
<td>-0.1075</td>
</tr>
<tr>
<td>REALTB</td>
<td>-0.0611</td>
<td>-0.1569</td>
</tr>
<tr>
<td>RFR</td>
<td>0.1614</td>
<td>-0.1079</td>
</tr>
<tr>
<td>PREM</td>
<td>-0.0020</td>
<td>-0.0285</td>
</tr>
<tr>
<td>SLOPE</td>
<td>0.0101</td>
<td>0.0259</td>
</tr>
<tr>
<td>TC</td>
<td>-0.0768</td>
<td>0.0027</td>
</tr>
<tr>
<td>IPI80</td>
<td>0.0864</td>
<td>0.0045</td>
</tr>
</tbody>
</table>

Table 5: Correlation coefficient between Independent Components of the rate of return of assets and Independent Components of the market indexes.

<table>
<thead>
<tr>
<th>Market Index Components</th>
<th>1st</th>
<th>2nd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>0.5451</td>
<td>-0.1603</td>
</tr>
<tr>
<td>2nd</td>
<td>-0.8155</td>
<td>-0.3250</td>
</tr>
</tbody>
</table>
A Scheme for Anonymous Electronic Commerce Using Secure Intelligent Trade Agents

Josep Domingo-Ferrer  Jordi Herrera Joancomartí

Dept. d’Enginyeria Informàtica
Universitat Rovira i Virgili
ETSE-Autovia de Salou, s/n
E-43006 Tarragona, Catalonia
e-mail {jdomingo,jherrera}@etse.urv.es

Abstract

In the last years, the exponential growth of computer networks has created an incredibly large offer of products and services in the net. Such a huge amount of information makes it impossible for a single person to analyze all the existing offers of a product on the net and decide which of them fits better her requirements. This problem is solved with the intelligent trade agents (ITA), which are programs that have the ability to roam a network, collect business-related data and use them to make decisions to buy goods on their owners’ behalf. Known ITA systems do not provide anonymity in transactions and require an on-line trusted third party. We present a new scheme for an intelligent trade agent system allowing anonymous electronic transactions with an off-line trusted third party.

Keywords: Intelligent agents, Electronic commerce, Electronic payment systems, Untraceability, Anonymity.

1 Introduction

In the last years, the exponential growth of computer networks has given birth to a new way of doing business, namely electronic commerce. This fact has created an incredibly large offer of products and services for the users of computer networks. Such a huge amount of information makes it impossible for a single person to analyze all the offers of a product existing on the net and decide which of them fits better her requirements. This human limitation in searching the net can be solved with the use of an agent, i.e. a program that roams the net looking for the items that best satisfy the user requirements. The emerging research in this area has developed what is known as intelligent trade agents (ITA) that collect business-related data from the net and use them to make decisions to buy goods on their owners’ behalf. This new concept of agents links artificial intelligence with electronic commerce and raises important security issues. In [6], a scheme for a secure intelligent trade agent is presented that requires an on-line authorization server during the transaction and the transaction itself is made using the traditional credit card number, which does not preserve the anonymity of the buyer. We propose a new scheme for electronic commerce with secure intelligent agents that does not need the on-line authorization center and preserves the anonymity of the buyer. Section 2 sketches a general scheme for a secure intelligent trade agent proposed in [6]. In Section 3, the new proposal is described in detail. Section 4 contains the conclusion and some directions for future research.

2 A recent intelligent trade agent scheme

In the secure ITA scheme proposed in [6], the main security problem that the authors want to solve is the location of agents. From a security point of view, allowing the agent to roam the network and reside in each visited server is unsafe. The reason is
that the agent carries the credit card number of the buyer, which can be stolen if the agent is attacked while residing in the server. Another place where the agent could reside is the buyer's computer but again this is not a good solution since this computer might not be logged-in while the agent travels the network. So the solution that the authors give is the creation of an Agent Repository (AR) where all agents can reside. Using a distributed-object technology, they propose a scheme where the ITA physically remains in the protective boundaries of the AR but logically it still roams from one server to the next. Figure 1 shows the architecture and the operation of the mechanism. The operation of the scheme depicted in Figure 1 can be summarized as follows:

Protocol 1 (Van der Merwe-Von Solms)

1. The buyer identifies herself to the AR in order to instruct the agent.
2. The buyer gives the instructions to the agent regarding the goods she is interested in.
3. The ITA roams to the first server ($S_1$).
4. Once the ITA finds an item that fits the requirements specified in Step 2, it starts the transaction via the Authorization Server (AS), which is typically a financial institution such as a bank.

(a) The agent sends to the server the credit card number of the buyer encrypted with the public key of the AS, so only the AS can read this information.

(b) The agent sends to the server a description of the items and prices agreed upon with the server. This information is signed with the private key of the agent and encrypted with the public key of the AS, so the information can only be read by the AS and can only be written by the agent.

(c) The server writes a message with the same information agreed upon with the agent. He signs this information with his private key and encrypts the result with the public key of the AS. Then he forwards to the AS the two messages received from the agent together with his own message.

(d) The AS decrypts and authenticates all messages. He checks that both the descriptions of the goods and the prices are the same and in that case, he executes the payment protocol for transferring the money to the seller's account.

5. The agent roams to the next server.

6. The buyer tells the agent to come back to the AR. This can be done automatically after some conditions given by the buyer in Step 2 are met.

7. The agent returns to the AR.

8. The buyer checks what the agent has bought.

3 Our intelligent trade agent scheme

In the scheme of Section 2, there are two main weak points that can be improved:

On-line AS For each transaction, the AS has to be on-line to authorize the transaction and this could be a problem in terms of availability, computing time and bandwidth of the connections to the AS.

Lack of anonymity From his knowledge of the buyer’s credit card number and the definition of the objects bought, the AS can link a buyer’s identity with a specific purchase, so anonymity is not preserved.
As will be shown later on, both problems above can be solved by combining a secure contract signing protocol (allowing a contract to be signed without the physical presence of the parties) with an electronic payment system allowing off-line anonymous payments. Secure contract signing is dealt with in Subsection 3.1 and anonymous off-line payment are dealt with in Subsection 3.2.

3.1 Secure contract signing

In any commercial transaction where buyer and seller can see each other, the exchange of goods is not a problem: the actions of paying and receiving the goods bought (from the buyer’s viewpoint) or getting paid and delivering the goods sold (from the seller’s viewpoint) can be done almost simultaneously. In electronic commerce, a security problem arises because there is no physical coincidence during the transaction between the server and the agent. None of the parties wants to make the first step since they do not trust each other. Basically there are two ways of handling that problem:

- The first one, proposed in [6], relies on a trusted third-party (the authorization server), which is trusted by both agent and server and arbitrates the whole transaction.

- The second one, proposed in this paper, tries to avoid the need for an authorization server. Instead of an on-line AS, we use an off-line bank. The idea is to use a secure contract signing protocol based on the exchange of secrets. Secure contract signing minimizes the risk incurred by the party who makes the first step.

We next recall a secret exchange protocol described in [5] which can be used as a primitive for secure contract signing. Assume that two parties A and B each have 2n secret m-bit numbers which they wish to exchange: \( \{a_i : i = 1, \cdots, 2n\} \) for A and \( \{b_i : i = 1, \cdots, 2n\} \) for B.

Protocol 2 (Secret exchange protocol)

1. A splits her 2n secret numbers in n pairs, for instance \((a_{2i-1}, a_{2i})\) for \( j = 1, \cdots, n \). Then, she sends to B one element of each pair using a 1-2 oblivious transfer (e.g. the provably secure oblivious transfer [1] can be used), which means that B receives either \( a_{2j-1} \) or \( a_{2j} \), for \( j = 1, \cdots, n \), but A does not know which elements B received (each element of a pair has 50% probability of being transferred).

2. Simultaneously with Step 1, B does exactly the same with his 2n numbers: he splits them in pairs and sends one element of each pair to A using a 1-2 oblivious transfer.

3. A and B send to each other the first bit of all their numbers \( a_i \) and \( b_j \) for \( i = 1, \cdots, 2n, \) then the second bit, and so on. If A wants to cheat B, she only has a probability \( 1/2^n \) of success because B has already received n out of the 2n secret numbers at Step 1 and A does not know which ones. By symmetry, the same applies if B wants to cheat A.

A drawback of protocol [5] is that, if B quits the protocol after A has sent the k-th bit of her secret numbers then B has a 2 to 1 advantage, since B has \( 2^{m-k} \) choices to obtain one pair while A has \( 2^{m-(k-1)} \) (twice as much). This problem was solved by Tedrick in [7] with a modified protocol that significantly minimizes the disadvantage of the party that starts the protocol.

A secure contract signing protocol [5] can be designed by taking a secret exchange protocol such as Protocol 2 as a building block:

Protocol 3 (Secure contract signing)

1. The agent A randomly generates 2n DES-like keys \( K_i^j \) for \( i = 1, \cdots, 2n, \) and n pairs of messages \((L_j^i, R_j^i)\) for \( j = 1, \cdots, n \). Then she encrypts each message with a different key: \( P_j^i = E_{K_i^j}(R_j^i) \) for \( j = 1, \cdots, n \) and \( Q_j^i = E_{K_j^{i+1}}(L_j^i) \) for \( j = 1, \cdots, n \).

2. The server S does the same (key and message generation), so that he obtains \( P_j^i = E_{K_j^i}(R_j^i) \) for \( j = 1, \cdots, n \) and \( Q_j^i = E_{K_j^{i+1}}(L_j^i) \) for \( j = 1, \cdots, n \).

3. Either of both parties (or both) creates a contract containing the conditions of the transaction and a clause whereby the contract will be considered as signed if and only if A is able to decrypt \( P_j^i \) and \( Q_j^i \) for some \( 1 \leq j \leq n \) and S is able to decrypt \( P_j^i \) and \( Q_j^i \) for some \( 1 \leq j \leq n \).
4. A and S exchange the 2n secret keys using Protocol 2 (or any secret exchange protocol).

3.2 Off-line anonymous electronic payment

In 1982, Chaum [3] presented the concept of blind signature to preserve buyer anonymity in electronic payments by making payments untraceable. Anonymous transactions were further developed in [4]. The idea is that the buyer "mints" her own notes and presents them to the bank, in order for them to be certified and given value. The important thing is that, upon certifying a note, the bank cannot see the note number; in this way, when the note is later spent, the note number cannot be used to trace the identity of the buyer who presented the note for certification to the bank. Chaum's blind signature protocol [4] is next recalled:

Protocol 4 (Blind signature)

1. The buyer randomly chooses half the digits of a note number n and repeats these digits to form the note number n (this repeated halves property will be subsequently used to tell valid note numbers from junk). Then, the buyer picks a random integer r that will hide the note number to the bank's eyes. The buyer computes \( z = nr \) where e is the bank's public key that certifies a certain amount of money, say one euro.

2. The bank withdraws one euro from the buyer's account and uses its private key d for certifying a one euro note, that is

\[
y = x^d = ((nr)^d) = n^d r^d = n^d r
\]

Then the bank sends to the buyer the value y.

3. The buyer computes \( z = y/r = n^d \) to obtain the certified note.

In Chaum's anonymous payment system, when the buyer wants to spend her one euro note, she sends \( z \) to the shop and the shop uses the bank's one euro public key e to authenticate \( z \) by computing \( z^e = n \) and checking that \( n \) is a valid note number. Later on, the shop sends \( z \) to the bank and the bank does the same authentication check and deposits one euro in the shop's account provided that the note number \( n \) has never been used before. The bank records \( n \) to guard against double spending.

With the above approach, buyer anonymity is guaranteed but the bank still has to be on-line to prevent double-spending of notes. In [2], an untraceable off-line cash system was presented, also based on blind signatures, which becomes traceable after double-spending. Since double-spenders will be identified later on, the bank does not need to be on-line to check the note numbers of each payment transaction. The [2] payment model is based on the representation problem in groups of prime order. The basic idea is that during the withdrawal protocol, the buyer embeds her identity into the coin in such a way that nobody can obtain it, provided that computing discrete logarithms is hard. Then, during the payment protocol, the buyer has to answer a time-related challenge from the server in such a way that the answer contains some information about her identity. This information must be such that one instance of it does not reveal anything about the buyer's identity, whereas knowledge of the answers to two different challenges enables the bank to obtain the buyer's identity. In this way, double-spending is deterred because it implies loss of anonymity for the double-spender.

3.3 Architecture of the new scheme

We are now ready to define a new scheme for an intelligent trade agent. We use the solution presented in [6] for locating the agents, and thus the ITA remains physically in the AR but logically roams the network, thanks to the distributed-object technology. In the payment transaction, instead of having an AS, we propose to combine an off-line anonymous payment system with a secure contract signing protocol.

The architecture and the operation of the new scheme are shown in Figure 2. The operation of the scheme of Figure 2 can be summarized as follows:

Protocol 5 (Anonymous ITA)

1. The buyer identifies herself to the AR in order to instruct the agent.

2. The buyer gives the instructions to the agent regarding the goods she is interested in.

3. On behalf of the buyer and after proper ITA-buyer authentication, the ITA withdraws
4 Conclusion and future research

We have presented a new scheme for an intelligent trade agent system that preserves buyer anonymity and does not require any third party to be online during transactions. These are two clear improvements with respect to previous proposals. Future research should be directed to dealing with untrusted ITAs. In the scheme presented above, the buyer does not trust the bank nor the servers, but implicitly trusts the ITA, who in Protocol 5 is allowed to withdraw money on the buyer's behalf. A possible way of dealing with untrusted agents is to extend the contract signing idea to formalize the relationship between the buyer (or the buyer's smart card) and the ITA: just as the ITA and the server sign a contract, the buyer and the ITA can sign another contract specifying how much money the buyer is transferring to the ITA (of course, the relationship and the money transfer between buyer and ITA cannot be anonymous). Note that, if the ITA is untrusted, money withdrawals are performed by the buyer herself (or the buyer's smart card).

References


Sistemes Multi Agents

Institucions Electròniques

Carles Sierra, Pablo Noriega
Institut d’Investigació en Intel·ligència Artificial, IIIA
Consell Superior d’Investigacions Científiques, CSIC
08193 Bellaterra, Barcelona.
sierra@iiia.csic.es

Abstract

En aquest treball proposem la noció d’institució electrònica intermediada per agents com l’estructura social adient per a una gran varietat de sistemes multiagent. Presentem una proposta de formalitització d’aquestes organitzacions socials i esmentem tres projectes, de caire ben diferent, en els quals s’està aplicant aquesta noció de manera exitosa.

1 Introducció

La noció d’interacció entre agents forma el nucli central de tots els sistemes multiagents. Les interaccions són la part observable dels diferents mecanismes de comportament d’aquests sistemes: cooperació, coordinació, col·laboració i negociació. En un sentit molt ampli, i degut a la seva generalitat, ens referirem a la negociació en aquest treball com el mecanisme bàsic d’interacció.

El nostre objectiu és construir agents que ajudin els humans a negociar. Els humans poden, eventualment, delegar completament en agents autònoms les tasques de negociació, però, en qualsevol cas, els agents que es desenvolupin han d’estar preparats per a interactuar, no solament amb altres agents autònoms, sinó també amb humans. Per tant, els models d’interacció que ens interessen són, precisament, aquells que tenen en compte les peculiaritats de la negociació humana. En particular, i com a tret fonamental, la noció de diàleg. Considerem que la negociació, i la interacció en general, és una activitat dialògica. Els humans i els agents, quan cerquen acords, dialoguen. Es precisament per aquest fet (que els agents poden estar interactuant amb un humà) que no podrem assumir, en general, nocions d’omniscència o racionalitat perfecta en les contraparte dels nostres agents.

Aquests intercanvis dialògics presuposen que els agents són entitats capaces d’establir compromisos. En termes de negociació, aquests compromisos vinculen els participants del diàleg gràcies a convencions socials i institucions que els fan acomplir. Per exemple, un acord de divorci s’estableix mitjançant un contracte que és validat i és mediat per un jutge; les transmissions patrimonials requereixen d’un registre civil i de la intervenció de notaris; les accions són intercanviades a institucions —borses. Notem també que si bé la negociació humana pot aprofitar-se de recursos elucdatoris i explicatius de la conversa, les negociacions intermediades en les institucions esmentades a dalt obvien aquesta necessitat imposant una interpretació estricta del llenguatge utilitzat en el diàleg a través de convencions establertes per les institucions. Aquest és el cas, per exemple, de les crides de preus d’una casa de subastes, o de les sequències de nombros en una taula de borsa. De la mateixa manera, en les interaccions mediades per agents es necessita fer explícit el significatiu, encara que és imaginable que les interaccions dialògiques puguin tenir un contingut elucdatori, resulta molt més pràctic restringir la negociació, i en general les interaccions, a aquelles que es realitzen utilitzant llenguatges perfectament definits. Aquest article es concentra, precisament, en les primeres passes vers la formalització d’institucions electròniques.

La noció d’institució electrònica es fonamenta en tres pilars que contenen els elements esmentats fins ara:

1. Un marc dialògic. Alguns aspectes d’una institució són estables i constitueixen el context o marc de referència de la interacció entre els agents. En una institució dialògica els agents interactuen mitjançant il·locucions.
Les institucions defineixen quines són les il·locucions acceptables, quina és l’ontologia — incoent-hi rols, llocs i temps —, un llenguatge comú per referir-se al “món”, un llenguatge comú per comunicar-se, i un metallenguatge comú per a, per exemple, establir preferències. En molts casos, una institució pot considerar com a rellevants relacions socials entre els agents participants —en termes d’autoritat, per exemple—, o alguna mens de característica de “personalitat” que pot afectar les interaccions. Tots aquests aspectes contextuais són el que anomenem marc díalogic.

2. Una estructura performativa. Les interaccions dels agents estan articulades al voltant de trobades de grups d’agents, que anomenem escenes, amb un protocol de comunicació ben definit. Entenem per protocol d’una escena els possibles diàlegs que els agents poden tenir-hi.

3. Unes regles de comportament. Les actions dels agents en el context d’una institució, en el cas que ens ocupa, actions dialògiques, tenen unes conseqüències, normalment en forma de compromisos, que imposen restriccions sobre les actions dialògiques dels agents en les escenes en que participaren en el futur. Les regles de comportament afecten el comportament dels agents, delimitant-lo.

La intuïció darrera de la noció d’institució és que és un mecanisme molt útil en l’especificació de sistemes d’agents autònoms interactius. A la secció 2 presentem el primer esboç de formalització de la noció d’institució electrònica, a la secció 3 esmentem tres projectes en els que preveiem l’ús d’aquest concepte.

2 Una proposta formal

Com ben dit a l’apartat anterior, la noció d’institució electrònica es basa en tres pilars ben diferenciats. Definirem primer aquestes tres pilars i després les institucions electròniques com a un agrupament dels tres.

Adaptant les definicions de [2, 8] tenim que la noció bàsica per a definir els elements ontològics és el marc díalogic:

**Definition 1** Un Marc dialògic és una tupla \( DF = (\text{Rols}, L, ML, CL, Time) \), on

1. **Rols** és un conjunt d’identificadors de rol.
2. **L** és un llenguatge lògic per a representar el domini.
3. **ML** és un metallenguatge sobre **L**.
4. **CL** és el llenguatge de comunicació entre agents. Es defineix com \( CL = \{\langle a, b, \varphi, t \rangle | a, b \in L, \varphi \in L, t \in Time \} \)
5. **Time** és un conjunt discret i totalment ordenat en instants.

La noció d’estructura performativa és la més complexa i interessant del formalisme perquè és l’encarregada de modelitzar la dinàmica del diàleg. La definició es basa en la noció d’escena que presentem a continuació. Tota escena té un estat inicial on comença l’intercanvi d’il·locucions entre els participants en l’escena. En cada moment l’escena es troba en un estat del qual és dut per les transicions que van cap a nous estats. Una transició es realitza si la il·locució que l’etiqueta es produeix. Una escena es considera acabada quan es troba en un dels possibles estats terminals. El conjunt de transicions i estats determina quin és el protocol de l’escena, i.e. el conjunt d’interaccions dialògiques possibles.

**Definition 2** Donat un marc dialògic \( DF = (\text{Rols}, L, ML, CL, Time) \), una escena \( s = (A, R, W, w_0, W_f, \rho) \)

- \( A \) és un conjunt finit de variables d’agent.
- \( R : A \rightarrow \text{Rols} \) assigna un rol a cada variable d’agent en l’escena.
- \( W \) és un conjunt d’estats. \( w_0 \in W \) és l’estat inicial de l’escena i \( W_f \subseteq W \) els estats finals de l’escena.

\( \rho : W \times CL \rightarrow W \) és una funció de transició entre estats, on les transicions estan etiquetades amb una il·locució.

Una escena es pot “realitzar” quant tenim una assignació d’agents “concrets” (i.e. programes o humans) per a totes i cadascunes de les variables d’a. Un cop l’assignació és completa l’escena es pot començar a executar, començant per l’estat inicial, i fins a acabar en un dels estats finals. Un cop una escena ha finalitzat, els agents que hi han participat es redistribuiran, com veurem en la definició d’estructura performativa. Alguns poden,
però, quedar-se en l’escena per tornar a executar-la amb agents nous.

Una estructura social complexa, com és el cas d’una institució, consta de multitud d’escenes interrelacionades que s’executen de forma concurrent o seqüencial segons sigui el cas. El punt fonamental, per tal d’establir la seva interrelació, és determinar quins agents abandonen quina escena, en quin estat l’abandonen, i a quina escena s’incorporen. Les restriccions que s’imposin a aquest flux de agents determinaran en bona mesura la flexibilitat del formalisme. Hi ha una restricción necessària des del nostre punt de vista, i és la integritat dels agents que participen en una escena. Es a dir, per a produir-se una transició entre estats d’una escena donada, tots els agents de l’escena han de ser-hi presents. Dit d’una altra manera, si algun agent abandona una escena, aquesta resta congelada —no s’emeten il·locucions— fins que l’agent s’hi reincorpora. Amb aquesta restricció en ment, podem modelitzar la noció de transició entre escenes d’una manera similar a com es modelitzen les transicions en xarxes de Petri: amb una funció d’incidència previa i una d’incidència posterior. El que si permet el formalisme s que un agent concret participi en més d’una escena alhora. Això ens dona la flexibilitat necessària per modelitzar el fet que un agent pot participar en més d’un diàleg alhora.

Cal esmentar que la noció d’escena representa el concepte d’especificació d’un conjunt d’agents que segueixen un protocol de dialeg concret. Ara bé, en un moment determinat una institució pot permetre l’execució simultània de més d’una particularització d’una escena amb actors diferents.

Notarem amb subíndex S la unió per a totes les escenes del conjunt dels diferents components, per exemple, \( W_S = \{ w | (A, R, W, w_0, W_f, \beta) \in S \text{ and } w \in W \} \).

**Definition 3**

Donat un marc dialògic \( DF = (Rols, L, ML, CL, Time) \) una estructura performativa és una tupla \( PS = (S, T, \alpha, \beta) \)

\( S \) és un conjunt finit i no buit d’escenes.

\( T \) és un conjunt finit i no buit de transicions entre escenes.

\( \alpha : W_S \times T \rightarrow 2^{As} \) és la funció d’incidència prèvia.

\( \beta : T \times W_S \rightarrow 2^{As} \) és la funció d’incidència posterior.

verificant que

1. hi ha integritat dels agents: per a tota \( t \in T \), a \( \in \alpha(w, t) \) per algun \( w \in W_S \) si i només si a \( \in \beta(t, \psi(w, t)) \) per algun \( \psi(w) \in W_S \).

2. hi ha un únic estat activ en una escena: per a tota \( t \in T, \psi(w) \in W_S \), tals que \( \psi(w) \neq \psi(w') \), si \( \beta(t, \psi(w)) \neq \emptyset \) and \( \beta(t, \psi(w')) \neq \emptyset \) llavors \( s \neq s' \).

Entenem que els actes de parla d’un agent introdueixen compromisos d’actuació posteriors. Els compromisos s’han d’interpretar com a obligacions d’actuació en una certa direcció i, per tant, com a una limitació de les possibilitats d’actuació. En altres paraules, això implica imposar restriccions sobre les transicions intraescena i inter-escena de l’agent que adquireix els compromisos. Aquestes restriccions les modelitzarem amb regles d’inhibició. Una regla d’inhibició el que fa és impedir que una determinada transició es pugui realitzar. Això s’aconsegueix modificant la funció de transició de determinades escenes i prohibint certes transicions entre escenes. Per exemple, si volem eliminar una transició intraescena escriuríem \( Inhibit(\rho(w, \psi) = \psi') \) amb les particularitzacions necessàries de les variables d’agent a la fórmula \( \psi \).

**Definition 4** Donat un marc dialògic \( DF = (Rols, L, ML, CL, Time) \) i una estructura performativa, \( PS = (S, T, \alpha, \beta) \), una regla de comportament té la forma:

\[ \text{If } \varphi_1 \land \ldots \land \varphi_n \text{ then } Inhibit(\psi) \in ML \]

on \( \varphi_i \in CL \cup L \) i \( \psi \) és de la forma \( \rho(w, \psi) = w' \) o bé \( \psi \in T \).

Finalment, podem agrupar els components d’una institució electrònica:

**Definition 5** Una Institució Electrònica és una triplleta \( EI = (DF, PS, RB) \), on \( DF \) és un marc dialògic, \( PS \) és una estructura performativa i \( RB \) és un conjunt de regles de comportament.

Actualment s’està desenvolupant una eina gràfica i un llenguatge per a l’especificació d’institucions mediades per agents. L’eina gràfica es troba descrita en detall a [1].

Només queda recordar que l’execució d’una institució consisteix en la generació d’instàncies d’escena (l’escena inicial en primer terme), l’assignació d’agents concrets a les variables d’agent d’una escena, i el moviment dels agents a través de les escenes respectant les restriccions de l’estructura performativa.
3 Aplicacions

3.1 Protocols mèdics

En la medicina moderna cada procés patològic porta associat un protocol. En aquest protocol s' especifica quines accions s'han de realitzar sobre els malalts en funció de quin és el seu estat i de quins han estat els resultats obtinguts a les proves realitzades. Cada vegada més s' intenta que l'acció dels metges vagi dirigida per aquests protocols i s'estableixen normes estrictes respecte del seu compliment.

Des del nostre punt de vista, la construcció d'agents que ajudin els metges a dirigir la seva actuació respectant els protocols establerts, o bé que permetin supervisar i monitoritzar la seva actuació, és un tema de recerca de gran interès. Forma part dels objectius d'un projecte CICYT anomenat SMASH en el que participa l'IHIA.

La línia de treball en curs per a modelitzar els protocols mèdics és la de considerar que un hospital és una institució dialògica, de la que són membres professionals de diferent tipus: metges, infermeres, auxiliars, caps de servei, directors mèdics, gerents, ... La interacció entre aquests professionals es realitza dins d'una estructura performativa rica, amb multitud d'escenes: visites als malalts, realització de proves, diàlegs elucidatoris entre professionals del mateix nivell, negociacions amb serveis farmacèutics, ... El que en medicina s'entén per "protocol" no és res més —ni menys— que l'estructura ordenada al llarg del temps de les diferents escenes (estructura performativa): primer s'ha de fer una ecografia, si el resultat és x, llavors fer una biopsia, ... i un conjunt de normes (regles de comportament) sobre què fer i què no fer en determinades circumstàncies. Actualment estem en procés de formalització del protocol de càncer de mama.

3.2 Comerç electrònic

Una de les àrees en les que hem estat treballant més intensament al llarg dels últims tres anys és la de comerç electrònic intermediat per agents. És uns de les àrees en la que els sistemes multiagent tindran un impacte tecnològic més gran de manera immediata. Aquesta àrea és, de fet, la que ha inspirat les nocions bàsiques d'institució electrònica, ja que el comerç es realitza en la vida real, des de fa segles, intermediat.

L'aplicació on més hem treballat aquestes idees és la d'una casa de subhastes. En concret hem formalitzat la llotja de peix de Blanes [3]. Una casa de subhastes conté, de manera natural, totes les nocions d'una institució com les presentades en aquest treball. Determina l'ontologia (qualitat de peix, tipus de peix), identifica els participants (compradors, barques), fixa les escenes en les que els diferents participants dialoguen (admissió de compradors, de venedors, subhasta, pagament als pescadors, lliurament de mercaderies als compradors), els protocols associats (a la baixa en l'escena de subhasta), i les normes de comportament (penalitzacions per compros sense crèdit, expulsions de la llotja per mal comportament reiterat). El lector interessat pot revisar els treballs desenvolupats fins ara en aquesta àrea d'aplicació [5, 7, 6].

3.3 COMRIS

COMRIS és un projecte Europeu l'objectiu del qual és, bàsicament, establir formalismes que relacionin el món real amb un món virtual en que cada individu (real) és representat per un agent que realitza tasques mitjançant la interacció amb agents que representen altres individus. Un dels punts interessants del projecte és la interacció entre aquests dos móns, el real i el virtual. La interacció del món virtual cap al físic es farà amb un dispositiu físic portable que s'està desenvolupant actualment i que reb informació dels agents del món virtual via senyals de radiofrequència. Aquest dispositiu sintetitza veu per ser sentida per l'individu que el porta. La interacció del món sensible al món virtual es farà mitjançant monitors, teclats i programes tradicionals.

El domini d'aplicació en el que s'està experimentant és el d'una conferència científica. Els assistents, en el moment de registrar-s'hi, són provistos del dispositiu físic i se'ls crea l'agent autònom. Està previst que l'agent ajudi l'individu, a partir de la informació sobre interessos que aquest li haurà proporcionat, a decidir a quines sessions assistir, a concertar cites amb persones amb interessos similars presents a la conferència, a organitzar reunions per a discutir propostes de projectes i tasques similars.

Les activitats que els agents realitzen en el món virtual poden ser vistes dins el marc d'una institució, novament. Bèstament, els agents establiran diàlegs per a negociar dates per reunions,
per a seleccionar quin agent té el perfil més adient per a ser un soci en un projecte europeu, etc. Aquests diàlegs poden estructurar-se molt comodament mitjançant escenes, i es poden definir les regles de comportament bàsiques respecte dels compromisos que els agents hauran d’adquirir en les seves negociacions. Les primeres idees sobre l’aplicació d’institucions electròniques a una conferència es poden trobar a [4].

4. Discusió i treball futur
Les nocions presentades breument en aquest article són centrals per a la generalització de moltes aplicacions que actualment s’estan desenvolupant a laboratoris de recerca de manera ad-hoc, en particular en l’àrea de comerç electrònic. Actualment s’està en la fase inicial de desenvolupament del llenguatge d’especificació gràfic i textual. La generació automàtica d’institucions electròniques i d’agents que hi participin són els objectius a l’alç d’aquesta recerca. Per exemple, ser capaços d’especificar una casa de subastes amb els seus protocols i obtenir com a resultat tota la infraestructura de comunicacions i alguns dels agents participants: subastador, admisor de venedors, supervisador i fins i tot agents prototípic de compradors i venedors, seria l’objectiu ambiciós d’aquest llenguatge d’especificació.

References
Gestión de red ATM en una arquitectura multi-agente inteligente a partir de parámetros de control de congestión.

Antonio Barba Martí
Dept. de Matemática Aplicada y Telemática. Escola Tècnica Superior d’Enginyers de Telecomunicació Universitat Politècnica de Catalunya (UPC).
c/ Jordi Girona 1-3, Mòdul C-3, Campus Nord, 08071 Barcelona, SPAIN
email:telabm@mat.upc.es

Resumen:
Las redes de comunicaciones ATM (Asynchronous Transfer Mode) de banda ancha requieren de nuevos mecanismos de control de recursos que permitan gestionar el tráfico en tiempo real a fin de evitar situaciones de congestión. En este artículo se presenta el esquema de funcionamiento de una arquitectura para la gestión de tráfico y congestión que se basa en múltiples agentes inteligentes móviles y que se adopera como una solución que permite proporcionar una calidad de servicio óptima a los abonados.

1. Introducción
El propósito inicial de la red digital de servicios integrados de banda ancha (B-ISDN, Broadband Integrated Services Digital Network) es proporcionar una interficie digital entre el usuario y un nodo de la red para el transporte de una gran variedad de servicios de audio, video y datos dentro de la misma red.

ATM (Asynchronous Transfer Mode) es el modo de transferencia que proporciona la integración de diversos tipos de tráfico que ha de soportar B-ISDN. Esencialmente consiste en un método de transferencia a través de la red basada en la commutación rápida de paquetes. Usa paquetes de longitud fija (53 octetos), denominados celdas, formados por una cabecera de 5 octetos (bytes) y un campo de datos de 48 octetos. La cabecera de la celda es mucho más simple que las cabeceras de los paquetes de las redes de commutación de paquetes convencionales. Esto permite a los conmutadores de ATM tener una alta eficiencia y retardos de commutación bajos.

De esta forma y con el objetivo de implementar una red de alta velocidad que pueda soportar velocidades de transmisión de multimegabits, ATM no proporciona operaciones de detección de errores en el campo de información dentro de las celdas, ni tampoco servicios de retransmisión.

Dado que el tráfico de las diferentes clases de servicios en B-ISDN proviene de diversas fuentes, tiene que ser tratado de forma diferente en su preparación de generar las celdas ATM correspondientes. Por esta razón, se define el nivel de adaptación ATM (AAL) que empaqueta la información de usuario (tráfico de voz, vídeo, datos) proveniente de los niveles superiores, en segmentos de 48 octetos (bytes). Esto puede consistir en la agrupación de bits de un flujo continuo de éstos o en la segmentación de tramas en bloques más pequeños.

Posteriormente, el nivel ATM añade una cabecera de 5 octetos a estos segmentos para formar las celdas ATM de 53 octetos, que serán transmitidas por la red usando el nivel físico. Esta agrupación en celdas de la información de usuario puede ser una función proporcionada por la red o puede estar implementada en el terminal del usuario. En el extremo receptor, las celdas se desagrupan y la información se vuelve a organizar en forma de paquetes en los servicios orientados a paquetes, o bien el tráfico de celdas se suaviza en los servicios de tasa de bit constante, ya que pueden experimentar retardos de transferencia variables en las colas de los multiplexores y conmutadores.

El control de congestión en redes basadas en ATM se rige por determinadas recomendaciones sobre gestión de tráfico [1] del ATM Forum. Para los nodos conmutadores de la red dentro de la red troncal, el control del tráfico puede realizarse de dos formas distintas. Se puede usar el EF (Explicit Forward Congestion Indication) que es un bit que está situado en celdas de datos y que se activa cuando hay una

1 Este trabajo tiene el soporte de la CICYT. TEL 96-1316.
situación de congestión. Otra posibilidad pasa por utilizar celdas RM (Resource Management) a través de los bits CI (Congestion Indication) y NI (No Increase) que determinados conmutadores pueden activar según la situación. Estas mismas celdas RM tienen otro bit que se denomina ER (Explicit Rate) que tiene la función de reducir la tasa de servicio de la red, que ubicándose en determinados conmutadores se puede utilizar para reducir la congestión.

El envío de celdas RM se realiza habitualmente de forma periódica, por ejemplo, cada 32 celdas (Nrm = 32) por defecto o bien cada 100 ms como cota superior (Trm = 100 ms). Por otra parte, se pueden utilizar celdas RM de forma aperiódica cuando determinadas situaciones lo requieran, si bien llevarán asignado el bit CLP (Cell Loss Priority) a 1 dado que estarán fuera del ancho de banda ACR (Allowed Cell Rate) definido por el servicio.

Con los elementos definidos anteriormente y los ya especificados en la recomendación [1] se plantea una arquitectura que permite reducir el nivel de congestión en una red. Esta arquitectura está basada en el uso de agentes inteligentes móviles para la gestión de red.

Las redes donde se empiezan a utilizar los agentes inteligentes son redes avanzadas, es decir, redes donde la información está disponible en cualquier momento, a cualquier hora y desde cualquier terminal.

Los agentes inteligentes se pueden definir como entidades de software autónomas que se comportan de acuerdo a una inteligencia autocontenido. El Agente Inteligente es un término que define desde interfaces de usuario adaptativos hasta comunidades de procesos inteligentes que cooperan unos con otros para conseguir una tarea común. Como agentes móviles representan objetos activos o transportables que se mueven desde un sistema a otro para acceder a recursos remotos o incluso encontrar otros agentes y cooperar con ellos [3, 4].

La distribución de tareas de gestión en la gestión de red se conoce como Gestión por Delegación, adopta un paradigma de gestión descentralizada que tiene la ventaja del incremento de potencia computacional en nodos de la red y decrecimiento de la presión en sistemas de gestión de redes centralizados y en anchos de banda de red. La Gestión por Delegación habilita tanto la distribución temporal (p.e. distribución sobre tiempo) como la distribución espacial (p.e. distribución sobre nodos de red diferentes) [3].

El objetivo básico consiste en traer la inteligencia de gestión, p.e. los servicios de gestión, tan cercanos como sea posible a los recursos gestionados y a su representación lógica en forma de objetos gestionados. Esto es mediante la delegación. El agente en el nodo gestionado tiene que verse como un entorno de agente de ejecución específico, soportando la ejecución remota de scripts de gestión. Estos scripts podrían activarse según el tiempo, actividades de gestión o la aparición de eventos específicos en el agente gestionado [5].

Para obtener los objetivos definidos previamente, el resto del artículo se estructura de la siguiente forma. En el apartado dos se describe el control de congestión desarrollado junto al tipo de tráfico en consideración y los parámetros que lo definen. En el tercer apartado se describen los modos de funcionamiento de los nodos. En el apartado cuatro se desarrolla la arquitectura multi-agente de gestión integrada en los nodos de la red troncal. En el cinco se presenta el algoritmo de control de congestión para los agentes de los nodos. Finalmente se muestran las conclusiones.

2. Control de congestión

El control de congestión puede realizarse mediante el uso de celdas RM así como mediante el bit EFCI. Estos mecanismos de comunicación permiten el intercambio de información entre la estructura de múltiples agentes instalados en los nodos de la red. Para poder actuar mejor sobre la red es preciso determinar el tipo de tráfico que circula entre los nodos de la red troncal. Se va a centrar el estudio en el tráfico tipo UBR+ (Unspecified Bit Rate), de esta forma, se puede mejorar el posible descarte de celdas en determinadas estructuras de información. En este caso, basadas en el protocolo estándar MPEG2 (Motion Picture Experts Group).

2.1 Tipo de tráfico en la red

La estructura de tráfico que define el estándar MPEG2 puede utilizarse como ejemplo para poder realizar un descarte selectivo de celdas. El flujo de información enviado a la red está formado por una secuencia de imágenes, cada una de las cuales lo forman porciones (slices) separadas por las correspondientes cabeceras de señalización. Estas porciones se empaquetan en trayectos de longitud variable denominados PES (Packetized Elementary Stream). En un nivel inferior aparecen paquetes formados con trayectos de longitud fija (188 bytes) denominados TS (Transport Stream) que son los que encapsulan los bloques de control (CB) según el nivel de adaptación AAL5 (ATM Adaptation Layer) SSSC (Service-Specific Convergence Sublayer). Un bloque de control (CB) está constituido con datos desde una única porción de
video. En el nivel más bajo, una PDU AAL5 está constituida por celdas ATM. Cada una de estas celdas incorpora datos de una única porción de video.

Fig. 1 Estructura de la información en una trama MPEG2.

2.2 Parámetros para el descarte de celdas

Se pueden definir una serie de parámetros a partir de la estructura de información descrita anteriormente y que se pueden tener en cuenta para realizar un descarte selectivo de celdas.

1) Indicador de prioridad de porción de imagen.
2) Indicador del conmutador de descarte de porción de imagen.
3) Contador de celdas descartadas en un bloque de control por el nodo conmutador.
4) Indicador de descarte de celda de bloque de control por el nodo conmutador.
5) Indicador de marcaje de bit EFCI en la celda de bloque de control por el nodo conmutador.

El primer indicador es de prioridad (p.e. bit de CLP, Cell Loss Priority) y se modifica a la recepción de la primera celda de la porción (slice) de una imagen. El segundo indicador es propio del nodo conmutador e indica una situación de descarte de celdas. Se define también un contador de celdas descartadas en torno a un bloque de control por parte del nodo conmutador. El cuarto apartado hace referencia a un indicador de descarte de bloque de control por parte del conmutador. Finalmente, existe un indicador de marcaje por parte del nodo de la situación de congestión a través del bit EFCI.

El mecanismo basado en un descarte de celdas selectivo utiliza los anteriores parámetros para obtener una gestión de pérdidas eficiente. La idea consiste en repartir las pérdidas entre los nodos conmutadores según su nivel de congestión a partir de un crédito de pérdidas. Este crédito se basa en los parámetros de tráfico negociados en la fase de establecimiento de la conexión.

3. Operación en el nodo gestionado

Como ya se ha descrito anteriormente, se parte de una implementación de un control en los nodos basada en un descarte selectivo de celdas para reducir el nivel de congestión en una red.

Su funcionamiento es el siguiente, en cada nodo se analiza la ocupación de los buffers para cada tipo de tráfico. En el caso de tráfico UBR+ (para el que se aplicará el algoritmo) se diagnostican tres umbrales: L1, L2 y L3, resultando cuatro estados posibles de funcionamiento del buffer (M1, M2, M3 y M4) y dando lugar a cuatro políticas de gestión diferentes según la ocupación del buffer.

En el caso de menor ocupación, es decir, en M1, la longitud de la cola está por debajo de L1. En esta situación no hay descarte de celdas pero podría haber marcaje del bit EFCI en las celdas en el caso de detectarse una tendencia a la congestión. En el estado M2 y M3, los descartes se producen según la tolerancia asociada a cada conexión (round robin, slices). El nodo descarta las celdas de baja prioridad (CLP=1). Estas celdas pueden ser marcadas inicialmente por la fuente emisora como celdas de baja prioridad, (usadas por ejemplo para aumentar la calidad de una imagen), o pueden ser celdas que no cumplen el contrato de tráfico establecido entre el usuario y la red. Frente al peligro de congestión, dichas celdas pueden ser selectivamente eliminadas por el nodo gestionado para proteger las prestaciones del tráfico de mayor prioridad. El nodo saturado activa el bit EFCI en la cabecera de las celdas de datos de las conexiones que pasan por ese nodo. Cuando la fuente recibe celdas con EFCI=1 reduce su tasa de emisión. Esta reducción del tráfico a la entrada, debe asegurar siempre la calidad de servicio contratada por el abonado, (no estando nunca por debajo de la mínima tasa contratada por éste), y permite solucionar en buena medida la congestión en la red.

En el caso de que la ocupación del buffer alcance el estado M4, se procede al descarte selectivo de porciones de imagen o a imágenes enteras. Debido al carácter específico de los diferentes tipos de imágenes que conforman un flujo MPEG2, se pueden tratar los descartes según la prioridad de su información. Por ejemplo, las celdas pertenecientes a los cuadros B (de doble interpolación) serían de baja prioridad y por...
tanto más descartables que los cuadros de tipo P o I.
Para este descarte, se realiza un reparto de pérdidas con otros nodos según su nivel de congestión.

La arquitectura descrita permite un incremento de la calidad de servicio (QoS): es decir, una reducción en el porcentaje de pérdidas de celdas y en el retardo de transferencia de éstas por cada dominio de gestión.

4. Arquitectura multi-agente para la gestión de tráfico

El marco de funcionamiento de los controles de tráfico y congestión que se definen viene determinado por el modelo de la arquitectura de red que se presenta a continuación.

El sistema ofrece dos redes diferenciadas. Una sería la red de acceso formada por nodos con conexión directa a los usuarios abonados. Se trataría de comunicaciones multiplexadas hasta los 155 Mbps. Por otro lado está la red fija, con otro tipo de nodos, interconectados entre sí de forma malla definida diversas jerarquías que permiten a su vez un aumento de la fiabilidad del sistema. Los enlaces que unen un nodo de acceso con uno de la red fija son de 622 Mbps y los que unen dos nodos de la red fija son de 25 Gbps.

![Fig. 2. Arquitectura de la red](image)

Los nodos se han diseñado de forma que puedan procesar de manera explícita diversos tipos de tráfico independientemente. A tal efecto se disponen estructuras de colas M/D/1 con prioridades distintas según el tipo de tráfico. Ello permite definir tráfico CBR, VBR, ABR, UBR (Constant, Variable, Available and Unspecified Bit Rate respectivamente) y también señalización (celdas RM inclusive).

El mecanismo de gestión de tráfico que se propone está basado en el uso de agentes predictivos distribuidos en los nodos de la red. Estos nodos a su vez, soportan algoritmos regidos por diversas condiciones de disparo según el nivel de tráfico detectado en la red. El propósito de este control de tráfico es prevenirlo y por tanto minimizar la congestión.

Los agentes forman parte de una red inteligente que trata de predecir el tráfico anticipándose de alguna manera a las variaciones de tráfico que pudieran llevar a una situación de congestión a la red. Los agentes situados en los nodos y/o otros elementos de red son autónomos y se comunican entre sí de forma que puedan adaptarse a comportamiento del tráfico de la red.

Estos sistemas cuentan además con un control de admisión de conexiones (CAC, Connection Admission Control) que permite el multiplexado estadístico de múltiples llamadas según una calidad de servicio (QoS, Quality of Service) específica para cada llamada.

En este artículo, únicamente se profundiza en la parte correspondiente al algoritmo de los agentes, que basado en un nivel de disparo, permite adaptarse y reactivamente controlar el flujo de celdas a la entrada de los nodos de acceso para evitar la congestión.

![Fig. 3. Esquema de la arquitectura multi-agente de la red](image)

El proceso de gestión que se desarrolla en la estructura multi-agente tiene cuatro etapas diferenciadas. En primer lugar, cada nodo monitoriza la información de...
gestión relevante de forma periódica. En un segundo paso, esa información se envía al centro gestor para un procesado inteligente de la misma. Como consecuencia de ese procesado, se adoptan una serie de decisiones sobre los elementos de red. Finalmente, se procede a la aplicación de las decisiones adoptadas.

5. Cooperación entre agentes inteligentes

Los tipos de celdas que se utilizan para la comunicación entre agentes de los nodos commutadores son del tipo OAM F4 (Operations, Administration and Maintenance) de gestión de rendimiento con tipos de funciones de monitorización y de reportes. Estas celdas tienen un ámbito de actuación restringido a cada dominio de gestión, tienen un mismo camino pero distinto canal virtual que las celdas de usuario. Su período de emisión si bien tiene en cuenta la situación de congestión de los nodos, es de un orden de magnitud aproximado al retardo de circulación de la celda por cada dominio.

5.1 Agente del nodo gestionado

Los nodos commutadores de la red troncal pueden modelarse mediante buffers específicos según las clases de tráfico que soportan. De igual forma se disponen sistemas de almacenamiento de celdas compartidos a fin de disminuir las pérdidas de información al mínimo en caso de saturación de buffers o congestión de la red.

En el dibujo de la fig. 4 puede observarse la situación de un buffer con umbrales variables en función de la carga de celdas de entrada al nodo.

Fig. 4. Esquema de un buffer de nodo de la red troncal con los umbrales de funcionamiento de acuerdo al tratamiento de congestión.

Los parámetros adoptados por el algoritmo del agente del nodo gestionado son los siguientes [2]:

- Longitud media del buffer UBR (LMB).
- Espacio disponible en el buffer compartido (EDBC) con \( a, b < 1 \).
- Tasa del puerto UBR de salida (TASA).
- Umbral alto del buffer UBR (NA o L3).

En este caso, el resto de los umbrales descritos en los apartados anteriores: L1, L2 serían fracciones fijas del umbral de disparamento graves NA o L3. Es decir, L1 podría ser 0.4*L3 y L2 equivalente a 0.6*L3. Si bien estos porcentajes podrían ser fijos en una primera aproximación al problema, una realización de la solución realmente inteligente exigiría unos márgenes variables de acuerdo al status del sistema gestionado.

Por otra parte, se pueden definir unas políticas de gestión en el ámbito de los nodos gestionados de la red. Es decir, una política P1 en donde el agente del nodo sólo puede utilizar un porcentaje b del espacio compartido disponible no usado por los otros servicios además del tamaño máximo asignado al servicio UBR+. Y de forma similar, una política P2 se define como aquella en la que el agente gestor sólo puede reducir el tamaño del buffer UBR+ del nodo en un factor.

Los resultados de las operaciones que realiza el agente del nodo gestionado se transferirán al agente del nodo gestor mediante celdas OAM F4 y con una periodicidad que depende de la prioridad marcada por el centro de gestión. La estructura de las celdas puede hacerse fácilmente compatible con los estándares definidos al efecto por el ATM Forum. Los parámetros enviados en el campo de información de la celda son los siguientes:

- Retardo: Retardo de servicio medio según LMB y la TASA.
- Tmaxsum: Incremento en \( b \times (n^{\circ} \text{slots de celdas disponibles/TASA}) \).
- Tmmax: Decremento en \( a \times (\text{NA-LMB/TASA}) \).
- Tendencia: Tendencia = \( d(EDUBR) / dt \).

El uso del parámetro Tmaxsum se circunscribe a la política de gestión P1 mientras que el parámetro Tmaxsum estaría vinculado a la política de gestión P2. Hay que decir además que las medidas que se utilizan son de tipo temporal (en seg.) dado que las conexiones en la red pueden realizarse en velocidades de salida muy variadas con retardos de servicio de celdas diferentes y de esta forma se facilita el tratamiento de la información por parte del sistema.

5.2 Agente del nodo gestor

Junto a las políticas de optimización de los recursos de los nodos (P1, P2) se integran otra serie de políticas que tratan de mejorar la calidad de servicio a nivel de red y de forma global. De esta forma el agente gestor...
interrelaciona el funcionamiento de los nodos conmutadores con los requerimientos de calidad del servicio contratados por el abonado. El agente del nodo gestor es el que se ocupa de aplicar las políticas de gestión correspondientes al resto de nodos de la red. Entre estas políticas consideradas más orientadas al servicio, se pueden destacar las siguientes: Política P3 mediante la cual el agente gestor distribuye el tiempo de servicio de crédito entre los conmutadores de una forma ponderada. Una política P4 en la que el conmutador más cargado recibirá más créditos que el que lo está menos. Si es necesario, el NA se decrementaría para mantener el mismo CTD (Cell Transit Delay) especificado en la fase de establecimiento de la conexión. Finalmente, una política P5 en la que para cada dominio, el retardo de tránsito máximo experimentado por una celda al pasar por un dominio sería siempre acotado por el máxCTD negociado en el establecimiento de la conexión.

5.3 Agentes inteligentes

El centro de gestión realiza un control inteligente mediante los agentes definidos. El sistema forma una infraestructura jerárquica distribuida de agentes. Se crea una computadora en cada nodo del control basado en los buffer a nivel de celdas de información de usuario y ya descrito en el apartado 3. A continuación se documentan unos algoritmos utilizables como soporte de los agentes inteligentes.

Se construye una función de coste que se basa en el nivel correspondiente de congestión en todos los nodos de la red. Para ello se recoge la información de ocupación de las celdas de los buffers correspondientes a los diferentes tipos de tráfico (ABR, UBR, VBR). Se definen los siguientes términos:

- \( f(C_i) \): Función de coste del conmutador i (Ci).
- \( d(EDABR)/dt \), \( d(EDCBR)/dT \), \( d(EDVBR)/dt \), \( d(EDUBR)/dt \): Tendencias correspondientes a los buffers de tipo de tráfico ABR, CBR, VBR y UBR respectivamente.
- \( T_{max}^{ABR}, T_{max}^{UBR}, T_{max}^{VBR}, T_{max}^{CBR} \): Incrementos en \( b^* \) (nº slots de celdas disponibles/TASA) para los distintos tipos de tráfico y buffer.
- \( T_{max}^{RES}^{ABR}, T_{max}^{RES}^{UBR}, T_{max}^{RES}^{C} \): Decrementos en \( a^* \) ([NA-LMB]/TASA) para cada tipo de tráfico y buffer.
- \( \text{Retardo}_{ABR}, \text{Retardo}_{UBR}, \text{Retardo}_{CBR}, \text{Retardo}_{VBR} \): Retardos para los distintos tipos de tráfico y buffer.

Luego se define una función de coste genérica:

\[
 f(C_i) = K_1 f(ABR) + K_2 f(UBR) + K_3 f(CBR) + K_4 f(VBR).
\]

con \( 0 < K_1, K_2, K_3, K_4 < 1 \).

siendo cada una de las tendencias definida análogamente como \( f(UBR) = \frac{d(EDUBR)}{dt} \)

El nodo gestionado actúa proporcionando los siguientes parámetros de forma continua, en este caso para el tráfico UBR:

\[
 \text{Retardo}_{C_i} = LMB/TASA
\]

\[
 T_{max}^{RES} = b^* (\text{nº celdas libres}/TASA)
\]

Si \( LMB < NA \) entonces \( T_{max}^{RES} = a^* ([NA-LMB]/TASA) \)
sino \( T_{max}^{RES} = 0 \).

\[
 f(UBR) = \frac{d(EDUBR)}{dt}
\]

El nodo gestionado actualiza su umbral de disparo NA a la recepción de cada celda OAM siendo:

\[
 NA = NA + t^* TASA
\]

En cuanto al agente del nodo gestor, de forma continua realizaría la siguiente operación para cada buffer integrado en los nodos. Además se limita el retardo global de la conexión CTD particularizando para el tráfico UBR:

\[
 \text{Si MaxCTD} > \text{Retardo}_{C_1} + \text{Retardo}_{C_2} \text{ entonces}
\]

\[
 /* \text{Política P5} */
\]

Caso

\[
 f(UBR_{C_1}) < 0 \text{ y } f(UBR_{C_2}) > 0 \text{ hacer Políticas P4}
\]

\[
 f(UBR_{C_1}) > 0 \text{ y } f(UBR_{C_2}) < 0 \text{ hacer Políticas P4}
\]

\[
 f(UBR_{C_1}) > 0 \text{ y } f(UBR_{C_2}) > 0 \text{ hacer Políticas P3}
\]

Los agentes gestores realizarían procesados más complejos a partir de la información proporcionada por los agentes de los nodos teniendo en cuenta la función \( f(C_i) \). Actualmente se está evaluando el impacto de estos algoritmos en el funcionamiento de la gestión de una red ATM desde un punto de vista más global.

6. Conclusiones

Como conclusiones, se puede decir que se ha presentado una gestión de calidad de servicio particularizada para el tipo de tráfico MPEG2 sobre red ATM con cualquier tipo de servicio, si bien se ha
estudiado para el best effort UBR+. Esta gestión de calidad puede aplicarse también a otros tipos de tráfico. El mecanismo que salvaguarda la pérdida de celdas estudiado está basado en un descarte de celdas selectivo inteligente que tiene en cuenta la estructura de información del flujo de celdas enviado a la red. Por otra parte, todo el sistema de gestión se basa en una arquitectura de múltiples agentes inteligentes. Existe aún todo un esfuerzo por realizar en torno a mejorar la eficiencia en la definición de los umbrales de disparo en los buffers de los nodos, así como en las tolerancias de los descartes. El uso de los agentes inteligentes optimiza el comportamiento del sistema al hacer los umbrales de éstos, variables.

7. Bibliografía

Distribució de la resolució de problemes al Web utilizant tècniques de satisfacció de restriccions*

Marc Torrens i Arnal  Rainer Weigel  Boi Faltings
Laboratoire d’Intelligence Artificielle (LIA)
École Polytechnique Fédérale de Lausanne (EPFL)
IN-Eculbens, 1015 Lausanne, Suisse
Marc.Torrens, Rainer.Weigel, Boi.Faltings@epfl.ch

Resum

Els servidors Web no sempre poden proveir serveis intel·ligents i efaçós perquè han de servir molts clients simultàniament. En aquest treball presentem una metodologia on la resolució intel·ligent de problemes és distribuïda, de tal manera que les tasques dures a resoldre es porten a terme al costat del client. Hem implementat una biblioteca en Java, anomenada JCL¹ per la resolució de problemes de satisfacció de restriccions (CSPs²). La JCL permet construir fàcilment applets i aplicacions Java que utilitzen CSPs en el Web. En aquest article describim la biblioteca i dues aplicacions distribuïdes que utilitzen la JCL.

Paraules clau: Raonament basat en restriccions, Comerç electrònic, Agents intel·ligents per Internet, Configuració de productes, Raonament basat en casos.

1 Resolució distribuïda de problemes

Com a resultat de la gran expansió del WWW, els servidors d’informació interactius són cada cop més comuns. El procés de consulta de base de dades requereix un temps de resposta petit que és difícil d’aconseguir si els usuaris interaccionen directament amb un servidor. Restriccions d’amplada de banda i sobrecàrrega de servidor són els principals problemes quan els usuaris consulten informació directament a un servidor Web. A més a més, molt sovint la informació dels servidors Web no està ben estructurada per a permetre consultes complexes.

Considerem, per exemple, les pàgines Web d’una companyia aèria. Molt sovint ofereixen accés a la informació de vols demanant a l’usuari els aeroports de sortida i arribada. El resultat de la consulta és una llista de possibles connexions de vols. Una consulta més complexa, on l’usuari planifica un viatge de negocis visitant algunes ciutats en un ordre arbitrari, seria molt més complicada de realitzar amb una interfície estàndard. Això és en particular cert si l’usuari vol minimitzar hores de vol, costos, etc.

Per tal de respondre aquest tipus de consultes complexes, un sistema ha de proveir capacitats de resolució de problemes. El nostre enfocament general és el següent: considerem la resposta d’una consulta com una solució a un problema de satisfacció de restriccions (CSP). La formalització d’un problema en un CSP comporta una natural descomposició en dues tasques: generació del CSP en el servidor, i resolució del CSP a la part client. Les avantatges d’aquesta metodologia són:

a) si hi ha moltes solucions al problema, aleshores el CSP pot ser vist com una representació compacta de l’espai de solucions. Això significa que el sistema utilitzarà una quantitat mínima d’informació, i

b) per trobar les solucions només es requereix potència de comput a la part client, evitant així la sobrecàrrega del servidor. A més a més, el procés d’examinar un nombre elevat

¹ Versió reduïda de l’article aparegut a la 10th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’98, Taipei, Taiwan).
² En l’article ens referirem a la biblioteca com a JCL, Java Constraint Library.
³ En l’article ens referirem als problemes de satisfacció de restriccions com a CSPs, Constraint Satisfaction Problems.
de possibles respostes a una consulta és independent del servidor i s'executa localment.

En la figura 1 podem veure l'arquitectura que permet dur a terme aquesta metodologia. El client envia una consulta que conté les restriccions de l'usuari cap al servidor. El servidor accedirà a la seva base de dades per tal de generar el corresponent CSP considerant les restriccions de l'usuari. El CSP s'empaqueja juntament amb algortimes de recerca formant un agent que es veurià a la part client. Així, l'usuari pot consultar les diferents alternatives interactuant amb l'agent de forma local.

Descomponem aquest procés en dues parts:

1. El servidor compila tota la informació rellevant de la base de dades i les restriccions de l'usuari (la consulta) en el corresponent CSP. El CSP és una representació compacta de totes les solucions que el problema pot tenir donades les restriccions inicials de l'usuari.

2. El servidor envia un agent format pel CSP i els algortimes de recerca al costat client. Això permet a l'usuari veure totes les possibles solucions localment. Com que l'agent s'executa al costat client, el temps de resposta és petit i l'usuari pot comparar diferents alternatives sense que calgui més accessos al servidor Web.

El procés de construcció del CSP només requereix una petita fracció de temps en comparació amb el procés de resolució del CSP. Així doncs, com que l'agent s'executa al costat client, la càrrega del servidor queda significativament reduïda. Després d'haver generat el CSP, no és necessari l'accés al servidor, per tant, l'agent enviat pel servidor, és completament autònom (figura 1).

En la següent secció describim la biblioteca de restriccions en Java (JCL, Java Constraint Library) que facilita la realització de l'arquitectura prèviament descrita.

2 Java Constraint Library

Hem implementat una biblioteca de restriccions en Java (JCL) que permet empaquetar CSPs i algorismes de resolució en agents al Web. Primer farem una brua introducció a les tècniques de satisfacció de restriccions (2.1) i seguidament presentarem la JCL (2.2). Finalment descriurem el shell de la JCL (2.3), una élira per resoldre CSPs al Web mitjançant un applet Java, i veurem una petita aplicació que tracta problemes d'assignació de recursos per il·lustrar l'ús de la JCL (2.4).

2.1 Problemes de satisfacció de restriccions

Els problemes de satisfacció de restriccions (CSPs, Constraint Satisfaction Problems) són idònis per modelitzar problemes de configuració [10, 9], planificació [12, 7, 3], assignació de recursos [1, 11], scheduling [8, 2] i timetabling [6] entre d'altres. Un CSP és definït per un conjunt de variables i restriccions entre elles. Una solució d'un CSP és un vector d'assignacions de valors a totes les variables de manera que totes les restriccions es satisfin. Un CSP pot tenir diverses, una o cap solució. Les avantatges principals de la programació basada en restriccions són les següents:

a) Ofereix un marc de treball general en el qual molts problemes reals es poden formalitzar d'una manera elegant i amb èxit.

b) Una representació basada en restriccions s'utilitza tant per sintetitzar solucions d'un problema com per la seva verificació (i.e., demostrar que una solució satisfi totes les restriccions del problema).

c) La naturalesa d'aquesta representació permet una descripció formal dels problemes i una descripció declarativa d'heurístiques de búsquedas.

Un CSP finit i discret es defineix per una tupla \( P = (X, D, C, R) \) on \( X = \{X_1, \ldots, X_n\} \) és un conjunt finit de variables, cadascuna d'elles associada a un domini de valors discrets \( D = \{D_1, \ldots, D_n\} \); i \( C = \{C_1, \ldots, C_l\} \) és un conjunt de restriccions. Cada restricció \( C_i \) s'expressa per una relació \( R_i \) en algun subconjunt de variables. Aquest subconjunt de variables s'anomena \( \text{connexió de la restricció} (\text{conn}(C_i)) \). La relació \( R_i \) sobre la connexió d'una restricció \( C_i \) es defineix per
Figura 2: Els components de l'entorn de la JCL.

Figura 3: Una jerarquia dels algorismes existents en la JCL.

2.2 La biblioteca pels CSPs
La biblioteca conté algorismes per:

- crear, gestionar i representar CSPs binaris i discrets;
- aplicar algorismes de recerca i de preprocés als CSPs.

La JCL es pot utilitzar en una aplicació independent (multi-plataforma) en Java o en un navegador d'Internet en forma d'applet. En la figura 2 presentem els principals components de l'entorn de la JCL. L'objectiu de la biblioteca és proveir un marc de treball per construir agents que resolguin fàcilment CSPs al Web. La JCL es divideix en dues parts: una biblioteca bàsica de restriccions i un envolcall (shell) que possibilita l'edició i la resolució gràfica de CSPs. Així doncs, la JCL permet implementar aplicacions portables i applets utilitzant mecanismes de satisfacció de restriccions. Tot aquest entorn és accessible a partir de: http://l1aww.epfl.ch/~torrens. Per més detalls veure [13].

Els algorismes
La biblioteca conté algorismes de recerca i de preprocés pels CSPs. Els algorismes de recerca permeten trobar les solucions d'un CSP, mentre que els algorismes de preprocés s'utilitzen per simplificar un CSP eliminant valors i restriccions que no afecten a les seves solucions. Diversos algorismes de recerca de solucions són implementats en la biblioteca. Principalment hi ha tres algorismes derivats del Chronological Backtracking (BT) que són: Backmarking (BM), Backjumping (BJ) i Forward Checking (FC) [4]. Algunes combinacions d'ells són implementades en [15] i adaptades en la JCL. En la figura 3 podem veure una jerarquia dels algorismes existents en la biblioteca. Dos algorismes de preprocés són implementats en la JCL: Arc-consistency (AC) i Path-consistency (PC) [5].

2.3 L'envolcall de la JCL (shell)
El propòsit del shell és proveir una interfície gràfica que faciliti l'ús de la biblioteca. El shell pot ser executat com a aplicació independent o com a applet a través d'un navegador Web. S'han prèm en consideració els següents aspectes:

- la definició i generació de CSPs,
- l'aplicació d'algorismes i
- la gestió de solucions intermitgent i finals.

La figura 4 il·lustra les relacions entre la biblioteca, el shell i el món extern.

En la figura 5 mostrem com les restriccions entre variables són editades utilitzant el ratolí i els
Sistemes Multi Agents

Figura 4: Les relacions entre la biblioteca, el shell i el món extern.

Figura 5: Editor de restriccions en el shell de la JCL.

menus. L’usuari pot seleccionar un parell de variables i marcar les combinacions de valors permeses definint una restricció. Una altra finestra important és la "finestra de control de resolució" mostrada en la figura 6. Aquesta finestra permet a l’usuari escollir un algorisme, seleccionar les opcions de visualització de solucions i començar l’execució de l’algorisme. La sortida HTML produeix una finestra en el navegador amb els resultats de l’algorisme. La "finestra d’algorismes" permet la selecció entre els algorismes de la JCL o altres algorismes que poden ser implementats per l’usuari. Durant l’execució d’un algorisme, el shell visualitza una finestra de resolució indicant, entre d’altres coses, quantes solucions s’han trobat fins al moment. En aquesta finestra, l’usuari també pot aturar, suspendre o continuar la resolució en curs.

2.4 Una aplicació gràfica utilitzant la JCL

En aquesta secció describim una applet per a la resolució de problemes d’assignació de recursos. És una simple aplicació per demostrar la flexibilitat i adaptabilitat que té la JCL per programar aplicacions utilitzant mecanismes per CSPs al Web. Els problemes d’assignació de recursos (Resource Allocation (RA) problems) poden ser definits com segueix (veure [1]): "Donat un conjunt de tasques amb els temps per portar-les a terme fixats, i donat un conjunt de recursos que es necessiten per a cada tasca, assignar un recurs a cada tasca de tal manera que cap recurs és assignat simultàniament a dues tasques diferents". El problema d’assignació de recursos pot ser modelat com un CSP discret i binari on les variables són les tasques i els valors són els recursos. D’altra banda, es presenten restriccions d’exclusió mútua entre dues variables si les corresponents tasques s’interseccen en el temps. Dita d’altra manera: dues tasques que s’han de portar a terme al mateix temps no poden utilitzar el mateix recurs.

L’applet per resoldre aquest tipus de problemes es pot executar al Web3. En la figura 7 podem veure l’aparença d’aquesta applet. Les tasques es representen com a rectangles. D’aquesta manera, si dos rectangles s’interseccen horitzontalment representen dues tasques que s’interseccen en el temps. Cada recurs correspon a un color diferent. Quan un recurs s’assigna a una tasca, aleshores la tasca és colorejada amb el color del recurs. L’applet permet a l’usuari generar un problema aleatori d’assignació de recursos, seleccionar un algorisme de recerca, i resoldre’l. Un mode d’execució anomenat "pas a pas" permet a l’usuari seguir el rastre dels algorismes.

3http://liewww.opf1.ch/torrns/exercise/exercise.html
3 Aplicacions

En aquesta secció, describim dues aplicacions distribuïdes al Web que utilitzen la JCL: un sistema de planificació de viatges aèris (3.1) i una aplicació de configuració de productes pel comerç electrònic (3.2). Els dos sistemes utilitzen una arquitectura distribuïda per resoldre el problema de la sobre-càrrega del servidor Web.

3.1 Sistema de planificació de viatges aèirs

Tot sovint, sorgeix el problema de la planificació de viatges de negocis. Típicament, el problema consisteix en que diverses persones s’han de trobar en diferents ciutats, i per cada trobada hi ha uns cers dies disponibles. A part d’aquestes restriccions, es presenten les restriccions que ens venen imposades pels mitjans de transport que volem utilitzar (en la nostra aplicació, els vols aèris). Actualment, la informació referent als horaris només es pot obtenir via consultes a les agències de viatges o als serveis de viatgers per un itinerari i unes dates concrets. D’aquesta manera, trobar un pla òptim requereix consultar separatament cada part de tots els possibles itineraris. Degut a que cada consulta implica un temps de resposta de l’ordre d’un minut, la planificació de viatges és molt feixuga. Una millor solució és utilitzar una arquitectura client-servidor on la resolució del problema és distribuïda al Web.

El prototípic del sistema de planificació de viatges aèris du a està dissenyat per facilitar la planificació d’aquest tipus de viatges utilitzant la JCL. Definim un pla de viatges aèris com una seqüència de vols connectant les diferents ciutats que l’usuari vol visitar. Donats un conjunt de ciutats i un conjunt de possibles dates i horaris per visitar cada ciutat, el sistema genera un conjunt de plans. Aquests plans consisteixen en les dates de les trobades conjuntament amb totes les possibles connexions de vols. L’usuari pot seleccionar les connexions més adients segons les seves preferències.

Les dades d’entrada pel sistema ATP són el conjunt de trobades, on cada trobada és una ciutat i els possibles horaris pels diferents dies que la trobada es pot dur a terme (veure la taula 1 i la figura 8). Formulem el problema de trobar un pla de viatges com un CSP binari. Existeix una variable per cada trobada $M_i$. Els valors del domini d’una variable $M_i$ són els vols entre les ciutats on les trobades poden tenir lloc. Les restriccions imposen que els vols siguin disponibles de manera que la persona pugui assistir a totes les trobades. Aquestes poden ser bàsicament formulades com segueix: un vol des de la reunió $j$ a la reunió $j$ s’accepta si el vol surt després que la reunió $j$ acabi i arriba abans que la reunió $j$ comenci. Una solució del corresponsant CSP pot ser vista com una seqüència de vols entre les ciutats (de fet, els aeroports) de les diferents trobades. Per cada reunió, un dels possibles dies ha de ser assignat i aleshores s’ha de garantir que existeixi al menys una connexió de vol entre dues reuniions consecutives. Considerem com a exemple les reuniions de la taula 1.

Seguidament presentem una metodologia per resoldre el problema de la planificació de viatges al Web utilitzant la JCL. La idea principal és generar el CSP associat al problema dels viatges a la part servidor i solucionar-lo localment al costat client (veure figura 1).

Les dades d’entrada (taula 1) s’envien al servidor per tal de construir el CSP considerant les bases de dades de vols i les dades d’entrada de l’usuari. Aleshores el CSP es empaquetat juntament amb algorismes de recerca de la JCL en un agent autònom. D’aquesta manera, el CSP es resol al costat client

\[4\text{Air Travel Planning (ATP) system.}\]
Taula 1: Exemple de dades d’entrada al sistema. Per cadascuna de les trobades hi ha un conjunt de possibles dates i hores de visita. Les dades s’envien al servidor Web per tal de formar l’agent amb el correponent CSP.

- la reunió M1 (Amsterdam) es planifica pel dia 3 (de 13h a 15h)
- la reunió M2 (Barcelona) es planifica pel dia 2 (de 13h a 17h)
- la reunió M4 (Ginebra) es planifica pel dia 4 (de 9h a 12h)
- la reunió M5 (París) es planifica pel dia 5 (de 8h a 12h)

A la taula 2 mostrem els vols corresponents a aquesta solució parcial. Per tal d’accedir a les dades dels vols per construir el CSP, hem creat una base de dades MiniSQL\(^3\). Utilitzem una biblioteca en Java anomenada MsqlJava\(^6\) que permet a les aplicacions o applets accedir i manipular bases de dades MiniSQL. El servidor MiniSQL s’executa en background al costat servidor del sistema. D’aquesta manera l’aplet pot accedir a la base de dades de vols per construir el CSP.

### 3.2 Configuració de productes pel comerç electrònic

Recientment, els processos de manufacturació teneixen a passar d’una producció en massa a una producció més personalitzada cap a l’usuari. La principal raó d’aquesta evolució consisteix en que els consumidors d’avui en dia tenen unes necessitats molt específiques i individualitzades les quals no poden ser satisfetes per la producció en massa. Aquesta producció individualitzada imposa nous reutes pel que fa al marketing d’aquests productes, en particular al comerç electrònic. Amb els actuals catalàs electrònics, els usuaris poden compondren solucions els mateixos seleccionant els elements del producte. Aquest procés és actualment una tasca feixuga i difícil de realitzar ja que es porta a terme d’una manera completament manual. En un futur pròxi, les utilitats per la configuració de productes que sintetitzen productes d’acord amb les preferències de l’usuari, seran indispensables per vendre productes multi-variants a través del comerç electrònic. Aquestes utilitats seran una part essencial per una catala electrònica cada vegada més i més intel·ligents que els actuals.

La tasca general de configuració es defineix com segueix: donats

a) un conjunt de components predefinits,
### Taula 2: Possibles vols des de Barcelona el dia 2 (després de les 17h) a Amsterdam el dia 3 (abans de les 13h) i des de Ginebra el dia 4 (després de les 12h) a París el dia 5 (abans de les 8h).

<table>
<thead>
<tr>
<th>Com</th>
<th>Vol</th>
<th>De</th>
<th>A</th>
<th>Sort</th>
<th>Arri</th>
<th>Dur</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB</td>
<td>4248</td>
<td>BCN</td>
<td>AMS</td>
<td>10:15</td>
<td>12:25</td>
<td>2:10</td>
</tr>
<tr>
<td>KL</td>
<td>352</td>
<td>BCN</td>
<td>AMS</td>
<td>7:05</td>
<td>9:25</td>
<td>2:20</td>
</tr>
<tr>
<td>SR</td>
<td>724</td>
<td>GVA</td>
<td>PAR</td>
<td>12:15</td>
<td>13:20</td>
<td>1:05</td>
</tr>
<tr>
<td>AF</td>
<td>2855</td>
<td>GVA</td>
<td>PAR</td>
<td>14:10</td>
<td>15:15</td>
<td>1:05</td>
</tr>
<tr>
<td>SR</td>
<td>726</td>
<td>GVA</td>
<td>PAR</td>
<td>16:15</td>
<td>17:20</td>
<td>1:05</td>
</tr>
<tr>
<td>AF</td>
<td>2835</td>
<td>GVA</td>
<td>PAR</td>
<td>17:15</td>
<td>18:20</td>
<td>1:05</td>
</tr>
<tr>
<td>AF</td>
<td>2893</td>
<td>GVA</td>
<td>PAR</td>
<td>18:05</td>
<td>19:10</td>
<td>1:05</td>
</tr>
<tr>
<td>SR</td>
<td>728</td>
<td>GVA</td>
<td>PAR</td>
<td>18:45</td>
<td>19:50</td>
<td>1:05</td>
</tr>
<tr>
<td>AF</td>
<td>2887</td>
<td>GVA</td>
<td>PAR</td>
<td>20:40</td>
<td>21:45</td>
<td>1:05</td>
</tr>
</tbody>
</table>

b) el coneixement de còm aquests components poden ser connectats i
c) les preferències de l'usuari per una configuració específica

trobar els conjunts de components que formen un producte satisfeix els requeriments de l'usuari i respectant totes les restriccions de compatibilitat. La tasca de configuració de productes es pot modelitzar com un problema de satisfacció de restriccions (CSP).

Existen dues metodologies per a la resolució de problemes de configuració representats com a CSPs:

- **Metodologia estàndard**: Aquesta tècnica consisteix en resoldre el corresponent CSP al problema de configuració des del principi. Això vol dir que l'usuari introduceix els requeriments i aleshores, el sistema de raonament basat en restriccions intenta trobar les solucions. Hi ha tres principals problemes amb aquest mètode. El primer és que els requeriments de l'usuari poden conduir cap a un CSP sobre-restringit (over-constrained) i llavors cap producte pot ser configurat. El segon és que hi ha massa productes satisfent els requeriments i l'usuari queda sobre-carregat amb tanta informació. El tercer i més serios problema és que sovint l'usuari només té una lleugera idea del producte i no pot expressar tots els requeriments d'entrada.

- **Metodologia basada en el raonament per casos**: Aquesta metodologia permet a l'usuari seleccionar una configuració des d'un conjunt de configuracions venudes a anteriors clients (possiblement fictícies). Aleshores la configuració seleccionada es modifica interactivament fins que el producte satisfa tots els requeriments. El mètode basat en casos és especialment útil quan només hi ha uns quant productes "estàndards" (els prototipus que podem trobar en qualsevol catàleg) i quan disposem de l'adaptació basada en restriccions per modificar aquests productes. Aquesta tècnica evita tots els problemes de la metodologia descrita anteriorment.

En ambdós casos, la JCL permet executar les parts computacionalment dures del problema a la part client del sistema.

En la figura 9 mostrem l'arquitectura necessària per dur a terme la metodologia basada en el raonament per casos.

Figura 9: Una arquitectura client-servidor per distribuir la configuració de productes al Web.

### 4 Conclusions

Com a resultat del creixement del WWW, els servidors d'informació interactius estan esdevenint cada cop més importants. El procés de consulta de bases de dades al Web requereix una resposta d'usuari ràpida que és difícil d'aconseguir quan els usuaris interactuen directament amb el servidor. Hem vist una metodologia on les tècniques d'agents i el raonament basat en restriccions (Constraint-Based Reasoning) poden separar la recerca de solucions dels accessos a la base de dades. La clau del nostre mètode rau en la representació de l'espai de solucions d'un problema com un CSP. Aquest CSP s'empaqueta amb algorismes de recerca de la JCL formant un agent. L'agent permet la resolució local de problemes al costat client. Això suprimeix la necessitat de grans recursos de computació del servidor, evitant-ne així la sobrecàrrega.
Agraïments

Agraëixo al Dr. Josep M. Borrell l’acurada revisió del text d’aquest article.

Referències


Sistemes Multi Agents

Investigadors: un model general d'agents racionals no-ideals

A.Moreno*
Depart. Enginyeria Informática, URV
amoreno@etse.urv.es

U.Cortés, T.Sales
Dep. Llenguatges i Sist. Informàtics, UPC
{ia,sales}@lsi.upc.es

e és, d'algun manera, el responsable de guiar el seu comportament.

1.1 Lògiques modals doxàstiques

L'eina formal utilitzada habitualment per modelar els processos de raonament que un agent racional pot dur a terme sobre les seves creences. El model dels móns possibles i la semàntica de Kripke proporcionen una semàntica intuitiva per a les fórmules doxàstiques, però ens limiten a poder modelar agents ideals.

Nosaltes propone un model diferent, en el que els móns possibles (consistents, complet) són substituïts per situacions concebibles, que són aquells móns que l'agent modelat és capaç de considerar, encara que siguin parciaus o inconsistents. Definim una classe general de raonadors no ideals, anomenats investigadors racionals, i mostrem com l'evolució de les creences d'aquesta mena d'agents, causada per una anàlisi dinàmica multi-dimensional, pot ser modelitzada formalment en el marc de les situacions concebibles.

1 Modelització de les creences d'agents ideals

La construcció d'agents racional és el tema capdavanter dintre de la recerca en IA. En els últims anys s'han proposat diferents tipus d'arquitectures per aquesta mena d'agents (consulteu e.g. [10] o [8]). Aquestes arquitectures permeten explicar el comportament de l'agent en funció d'actituds proposicionals (com ara creences, desitjos o intencions). En totes elles es pot justificar el comportament de l'agent dient que aquest desitja assolir una sèrie d'objectius, i que va generant intencions (o plans d'acuciació) que permetran assolir-los, d'acord amb el que l'agent coneix sobre el món que l'envolta (és a dir, d'acord amb les seves creences). Així, es pot dir que el sistema de creences d'un agent racional

*Recerca recolzada pel projecte CICYT SMASH: Sistemes multi-agent i la seva aplicació als serveis hospitalaris (TIC96-1038-C04-04).
1.2 Omnisciència lògica i raonament perfecte

Proposició 2 Els agents modelitzats usant el model dels mons possibles i la semàntica de Kripke són lògicament omniscients, perquè han de creure totes les tautologies clàssiques, i també raonadors perfectes, perquè han de creure totes les conseqüències lògiques clàssiques de les seves creences.

Així, aquest no és un model realista d’agents reals (tant humans com computacionals), que sempre pateixen una limitació de recursos que els impedeix acribir a ser raonadors ideals. Així, un dels problemes amb els que s’enfronta la Intel·ligència Artificial (i la motivació principal del nostre treball) és la construcció de models teòrics que permetin explicar formalment l’evolució de les creences d’un agent no-ideal.

La resta de l’article es estructurada de la següent manera. En la secció 2 s’introdueix la noció de situació concebible, que serà clau per permetre la modelització d’agents no ideals. En la secció 3 es defineix un model abstracte d’agents no ideals, i en la secció 4 aquest model és concretat en una classe específica d’agents no ideals, anomenats investigadors racionals. En la secció 5 es mostra amb un exemple com l’evolució de les creences d’aquesta mena d’agents es pot modelitzar formalment en el marc de les situacions concebibles. L’article acaba amb un resum i les referències bibliogràfiques.

2 Monòs impossibles

L’arrel dels problemes de l’omnisciència lògica i el raonament perfecte és la visió clàssica del que és un monòs possible: una descripció completa i consistent d’un estat en el que es podria trobar l’agent. Així, un monòs possible es correspon amb un model, en el sentit lògic d’aquesta paraula. Com els monòs són complets, l’agent ha de tenir creences respecte a qualsevol fet en qualsevol món possible; com són consistent, tot el que es segueix de les creences de l’agent ha de ser necessàriament cregut.

Una idea natural és intentar solucionar aquests problemes eliminant les assumpicions de completeza i consistència que acabem de mencionar. De fet, en el camp de la Lògica Filosòfica hi han hagut diversos autors que han proposat el considerar mons possibles impossibles, en el sentit de tenir mons possibles on les connectives lògiques no tenen la semàntica habitual, o les tautologies clàssiques poden no ser certes, o es poden tenir fòrmules inconsistents. La majoria d’aquestes aproximacions allu- gen d’alguna manera el raonament ideal (consulteu [5], on es fa un amplí repàs de diverses solucions que s’han proposat a l’omnisciència lògica i el raonament perfecte). El problema bàsic que presenten és que semblen servir per modelar agents amb capacitat inferencials molt limitades (e.g. un agent pot creure P i (P ⇒ Q) i no tenir manera de creure Q, o pot creure (P ∧ Q) sense poder creure ni P ni Q). Nosaltres defensem en aquest article que es pot usar el camp dels mons impossibles per concebre un marc formal on es pugui modelar l’evolució de les creences d’un agent racional no ideal.

2.1 Situacions concebibles

L’expressió món possible no s’adapta exactament a la idea que tenim del que hauria de ser una alternativa doxàstica; en el nostre treball parlem de situacions concebibles.

Idea 1 [6], [7] Una situació concebible és qualsevol estat que l’agent pugui considerar, independentment de la seva possible parcialitat o inconsistència. Pot ser una situació que l’agent ha experimentat, o que li han explicat, o que ha somiat, o que simplement s’ha imaginat com a possible.

La única condició per a que un cert escenari pugui ser inclòs a la categoria de situacions concebibles és que l’agent pugui considerar-lo com a possible, com a concebible; no té per què ser ni consistent ni físicament realitzable. La idea principal és que no té per què ser un model lògic. A la resta de l’article la noció de situació concebible es pren com a primitiva, i correspondrà al que l’agent pot considerar com a “realitats”, sigui proves (des de la experiència o simplement imaginades). Si les alternatives doxàstiques (en el nostre cas, les situacions concebibles) són (possiblement) incompletes i inconsistencies, tant l’omnisciència lògica com el raonament ideal sembren desaparèixer. L’agent pot no creure totes les tautologies (per bé considerar com a possible una situació en una tautologia determinada no sigui certa), i no ha de creure totes les conseqüències lògiques de les seves creences (pot considerar com a possible una situació en segon certes P i (P ⇒ Q), però Q no ho sigui).

3 Agents no ideals

En aquest article es mostrarà com el model de les situacions concebibles, unit amb la semàntica de Kripke habitual (lleugerament modificada, com es veurà a la secció 5.2), ens permetrà la construcció d’un model formal de l’evolució gradual de les
creences dels agents reals, que no són ni lògicament omniscentes (no creuen totes les tautologies clàssiques) ni raonadors perfectes (poden creure una sèrie de fets sense haver de creure necessàriament totes i cadascuna de les seves conseqüències lògiques). Per això definirem una classe de raonadors no ideals, que anomenarem investigadors racionals, que ens serviran com a model abstracte de qualsevol tipus d’agent no ideal. Més endavant, en la secció 5, mostrarem com modelar formalment l’evolució de les creences d’aquesta mena d’agents.

Idea 2 [6], [7] Els investigadors racionals són agents que estan contínuament analitzant les seves creences, per fer-les els més semblants possibles als fets que són certs al món real. Intencions eliminar aquelles creences que no reflecteixen acuradament el que és cert al seu entorn, i intencions refinari lògicament aquelles que descobreixen que són cerques. A aquest procés l’anomenarem investigació racional.

A l’hora de construir un model abstracte, general, de qualsevol tipus d’agent no-ideal, el que hem fet és centrar-nos en aquelles tasques que realitza l’agent i que influeixen directament en les seves creences; més concretament, en el model dels investigadors racionals tindrem en consideració aquests aspectes:

- Capacitats deductives. Considerarem que tot agent racional ha de ser capaç de realitzar deduccions sobre les seves creences, i afegir els fets deduïts com a noves creences. El que no farem és considerar que l’agent creu d’entrada totes les conseqüències lògiques de les seves creences sense cap esforç. L’agent podrà anar deduint fets (possiblement no tots els que siguin conseqüència lògica clàssica, si té unes capacitats deductives limitades), però aquest procés inferencial haurà de ser explícit (e.g. si creu P i (P → Q), haurà de realitzar un procés deductiu abans de creure Q).

- Addició d’informació provenent de l’entorn. Tot agent realitzà la seva tasca dintre d’un entorn, en el que hi ha altres agents que es comuniquen amb el. Considerarem que els agents racionals han de ser capaços de rebre informació de l’entorn (e.g. provinent de sensors o d’altres agents), i que aquesta informació modifica directament les seves creences. De fet, farem una suposició encara més forta: que la informació rebuda és incorporada directament a les creences de l’agent.

- Demanda d’informació de l’entorn. Els investigadors racionals, com hem comentat abans, intencions continuament reflectir en les seves creences el que és cert en el seu entorn; per això, aniran analitzant les seves creences i intentaran confirmar o refutar aquelles que siguin incertes (en un estil que podríem anomenar poppetia). Així, hi haurà moments on l’agent haurà de cercar dades en el seu entorn que permetin fer aquestes corroboraclons o refutacions, i així eliminar (o conservar) una creença determinada.

- Incorporació de dubtes. Moltes vegades l’agent necessitarà saber si un fet es pot deduir a partir del seu conjunt de creences; això pot ser necessari per exemple per a que l’agent pugui decidir l’acció a prendre en un cert moment per assolir un determinat objectiu. El que considerarem és que els investigadors racionals són capaços de plantejar-se dubtes, de preguntar-se si creuen o no una certa fórmula. Aquest dubte influirà en les creences de l’agent, ja que permetrà que l’agent faci una anàlisi de les seves creences que el porti a deduir si creia o no (implicitament) la fórmula dubtada.

4 Investigadors racionals

Idea 3 Un investigador racional estarà contínuament realitzant una anàlisi dinàmica multi-dimensionals de les seves creences, per mantenir-les els més similars possibles als fets que són certs al món real. Els components d’aquesta anàlisi són els següents:

- Anàlisi lògica, on l’agent podrà realitzar inferències deductives (problematament limitades) sobre les seves creences.

- Anàlisi exploratori, on l’agent podrà plantejar-se dubtes, preguntar-se si creu (implícitament) o no un cert fet.

- Anàlisi experimental, on l’agent podrà demanar dades al seu entorn per confirmar o refutar alguna de les seves creences.

Els investigadors racionals també podran incorporar a les seves creences la informació que rebin directament del seu entorn.

4.1 Anàlisi lògica

L’agent utilitza una versió modificada del mètode clàssic dels taulers analítics ([9]) per dur a terme
l’anàlisi lògica de les seves creences. Les fórmules de cada tauler es divideixen en dos conjunts, anomenats columna esquerra i columna dreta. Les regles d’aquest càlcul de taulers es mostren a la figura 1, usant la notació uniforme de Smullyan ([9]). En aquesta figura els símbols α, β, γ i δ representen tipus especials de fórmules, φ i ψ representen qualsevol fórmula, Γ i Δ representen conjunts de fórmules, c és una constant de Skolem i o és un individu arbitrari.

Un tauler pot ser tancat lògicament per l’agent quan conté una fórmula i la seva negació a la columna esquerra o quan conté una mateixa fórmula a les columnes esquerra i dreta. Un tauler tancat lògicament és eliminat definitivament de l’anàlisi.

4.2 Anàlisi exploratòria

Una diferència important entre el nostre mètode dels taulers analítics i el clàssic és que l’agent no pot afegir tautologies arbitràriament dintre dels taulers (sinó, no es podria evitar l’omnisciència lògica). No obstant, en la dimensió exploratòria d’anàlisi l’agent sí pot afegir algunes fórmules als taulers oberts. L’agent es pot plantejar preguntes a ell mateix, pot introduir dubtes en l’anàlisi, pot considerar si una certa fórmula φ és o no el cas. Tècnicament, aquesta idea s’implementa permetent que l’agent introduïxí instàncies de l’Axioma del Tercer Exclòs, i.e. fórmules de l’estil (φ ∨ ¬φ), a les columnes esquerra i dreta de l’anàlisi lògica. L’ús d’aquesta tautologia clàssica sembla una manera natural de deixar que l’agent tingui dubtes, que es pregunti si creu (implicitament) una certa fórmula (φ) o la seva negació (¬φ). Aquesta excepció permet la introducció de la fórmula (φ ∨ ¬φ) en un tauler, que més tard és dividit en dos subtaulers que contindran φ en una columna i ¬φ en l’altra (el tercer subtauler generat en l’anàlisi lògica seria immediatament tancat, com veurem a l’exemple de la secció 5.1). D’aquesta manera l’agent pot explorar les dues alternatives de forma independent, i l’anàlisi lògica pot guiar la cerca d’exemple o contra-exemple que serveixin per donar més credibilitat a una banda del dubte que a l’altra.

4.3 Anàlisi experimental

Un agent pot cercar informació a l’entorn de moltes maneres diferents (consultant informació emmagatzemada en bases de dades, preguntant a altres agents, fent experimentes, buscant per Internet, etc.). Totes aquestes formes d’adquirir informació externa venen representades per la capacitat dels investigadors racionals d’obtenir informació de l’entorn en la dimensió experimental d’anàlisi. En aquesta dimensió els agents poden fer preguntes al seu entorn, i després poden afegir les respostes obtingudes a les columnes esquerra i dreta de l’entorn de l’anàlisi lògica. Més específicament, l’agent pot fer preguntes al seu entorn (en P_i és una proposició bàsica): Existeix algú que tingui les propietats P_1, ..., P_n i no tingui les propietats P_{n+1}, ..., P_m? Només hi ha dos tipus de respostes que puguin fer l’agent: negatives (no hi ha cap objecte conegut que tingui les característiques demandades) i positives (l’individu r les té).

L’anàlisi lògica guia l’anàlisi experimental, suggerint quines preguntes hauria de fer l’agent al seu entorn per millorar el seu coneixement. L’agent hauria d’anàlitzar fórmules lògicament fins que trobés fórmules atòmiques que contenen constants de Skolem. Aquestes constants no es refereixen a cap element concret, sinó a objectes desconeguts que han de tenir les propietats representades pels predicats atòmics. Així, l’agent pot incrementar les seves creences tancant empiricament els taulers que contenen aquestes fórmules atòmiques, si descobreix que no hi han objectes que satisfacin les propietats requerides (a la secció 5.2 es dona una explicació més detallada de les conseqüències d’un tancament empiric). Per tant, la presència de les constants de Skolem a les fórmules atòmiques dels taulers oberts de l’anàlisi lògica és el detonant de la dimensió experimental d’anàlisi.

5 Evolució de les creences

L’objectiu d’aquesta secció és mostrar, amb un petit exemple, com l’ús de les situacions concebibles permet la modelització formal de l’evolució de les creences dels investigadors racionals. Comencem mostrant com l’agent podria usar els diferents dimensions d’anàlisi de les seves creences, i després mostrem l’evolució de les creences de l’agent en aquest procés.

5.1 Exemple d’anàlisi

Suposem que el conjunt inicial de creences (Δ) de l’agent és: \{BirdTweety, FliesTweety, BirdPlatin, FliesPlatin, BirdWoddy, ∀x(Penguin_x → ¬Flies_x)\} (abreviat, \{B_1, F_1, B_2, F_2, B_3, ∀x(P_x → ¬F_x)\}). L’estat inicial de l’anàlisi de les creences vindrà representat per un tauler, T_0, que contindrà en la seva columna esquerra les creences inicials de l’agent.

L’agent pot començar l’anàlisi preguntant-se si tots els ocells volen (noteu que aquest fet ni la seva negació són deducibles a partir de les creences...
Figura 1: Regles de l’anàlisi lògica

Figura 2: Anàlisi lògica de \((\forall x (B_x \Rightarrow F_x)) \lor \neg \forall x (B_x \Rightarrow F_x))\)

inicials). Pot incorporar aquest dubte a l’anàlisi usant la dimensió exploratori, en la que pot afegir (a les columnes esquerrades dels taulers oberts) instàncies de l’Axioma del Tercer Exclòs. Així, l’agent podria afegir la fórmula \((\forall x (B_x \Rightarrow F_x)) \lor \neg \forall x (B_x \Rightarrow F_x))\) a la columna esquerra de \(T_0\), generant un nou tauler, \(T'_0\) (veure la figura 2).

L’agent pot continuar l’anàlisi de les seves creences de forma lògica, aplicant la règla que permet analitzar disjuncions situades a la columna esquerra dels taulers a la fórmula que s’acaba d’introduir, \((\forall x (B_x \Rightarrow F_x)) \lor \neg \forall x (B_x \Rightarrow F_x))\). El resultat és la generació de tres subtaulers, \(T_1\), \(T_2\) i \(T_3\) (veure la figura 2).

\(T_1\) conté una fórmula i la seva negació a la columna esquerra; l’agent podria notar aquest fet i tancar lògicament aquest tauler. Després d’aquest tancament, els únics taulers que restarien oberts serien \(T_2\) i \(T_3\).

L’agent podria decidir continuar la seva investigació fent l’anàlisi lògica de \(\neg \forall x (B_x \Rightarrow F_x)\) a \(T_3\). El resultat d’aquesta anàlisi és la generació d’un nou subtauler, \(T_4\), que conté (a banda de totes les fórmules de \(T_3\)) una instanciació de la fórmula quantificada amb una nova constant de Skolem, c. Aquest tauler es pot veure a la figura 3.

Ara es pot analitzar la fórmula \(\neg (B_x \Rightarrow F_x)\) a \(T_4\). L’agent hauria d’estar especialment interessat en analitzar les fórmules que contenen constants de Skolem, perquè les fórmules atòmiques on apareixen aquestes constants s’usaran per guiar la dimensió experimental d’anàlisi, com s’ha comentat a la secció 4.3. Si l’agent analitza aquesta fórmula generarà un subtauler, \(T_5\), que contindrà totes les fórmules
de \( T_3 \) i també \( B_c \) i \( \neg F_c \) (veure la figura 3).

L'agent podria ara adonar-se de que hi ha un tauler obert (\( T_2 \)) que conté fórmules atòmiques amb constants de Skolem. Aquestes fórmules, per tant, es refereixen a propietats d'objectes genèrics. L'agent pot intentar descobrir si existeix alguna objecte concret que tingui les propietats representades per aquests predicats atòmics. Si no en pot trobar cap, això pot ser degut a dos motius: no n'existeix cap realment, o les fonts d'informació de l'agent no són prou bones. Com l'agent no pot saber quin és el cas ha de deixar una porta oberta a la possibilitat de que, si troba més tard un individu amb les propietats desitjades, pugui afegeir aquesta informació a les seves creences. Així, els investigadors racionals podran tenir creences no monòtones (i.e. una certa creença pot ser desestimada en el futur, davant de nova informació).

![Figura 3: Arbre de taulers](image)

Figura 3: Arbre de taulers

L'agent podria fer ara la següent pregunta a l'entorn, en la dimensió experimental d'anàlisi: Existeix algun objecte que sigui un ocell i que no voli? Si la resposta és negativa, l'agent podria tancar empiricament el tauler \( T_2 \) (perquè no ha pogut donar un valor específic a la constant de Skolem \( c \)). Per tant, l'únic tauler que restaria obert seria \( T_2 \).

### 5.2 Evolució de les creences

En aquesta secció es proposarà una modelització formal de l'evolució de les creences de l'agent en aquest exemple, utilitzant el model de les situacions concebibles (que, per abreujar, a partir d'aquest punt les anomenarem casos). L'agent, en cada moment, considera una sèrie de casos com a alternatives doxàstiques; és a dir, considera que el seu món actual podria ser qualsevol d'aquestes casos. La relació entre l'anàlisi sintàctica (usant taulers) i la modelització semàntica (usant casos) queda reflectida a la següent proposició.

**Proposició 3** Un tauler és la representació (parcial) d'un conjunt de casos: aquelles on són certes les fórmules de la columna esquerra i són falses les de la columna dreta. L'agent creurà una fórmula si és certa en totes les seves alternatives doxàstiques, i no la creurà si és falsa en alguna d'elles.

Per tant, \( T_0 \) representa una classe de casos \( (\overline{w}_\alpha) \): aquelles on les sis fórmules de \( \Delta \) són certes. A l'inici de l'exemple anterior, les creences de l'agent són les fórmules del conjunt \( \Delta \); per tant, aquesta situació es pot modelitzar semànicament considerant que l'agent pensa que les seves alternatives doxàstiques són precisament les casos de \( \overline{w}_\alpha \). Aquest fet es pot representar formalment (veure la figura 4) amb una relació d'accessibilitat \( R_0 \) entre la situació on es troba l'agent (que anomenarem \( w_e \)) i totes les cases de \( \overline{w}_\alpha \). En les figures d'aquest article cada cosi (o classe de cos) ve representada amb un rectangle dividit en quatre parts, amb el següent significat:

- \( \phi \) apareix al quadrat superior/inferior esquerra si \( \phi \) és cert/fals en aquella cosi.
- \( \phi \) apareix al quadrat superior/inferior dret si \( \phi \) és/ha estat esmentat per l'agent en aquella cosi.

Les creences de l'agent a \( w_e \) es calculen amb la versió modificada de la semàntica de Kripke explicada a la proposició 3. Inicialment, com es pot veure a la figura 4, les creences de l'agent a \( w_e \) seriien les fórmules de \( \Delta \), que són aquelles que són certes a totes les cases de \( \overline{w}_\alpha \).

![Figura 4: Relació d'accessibilitat inicial](image)

Figura 4: Relació d'accessibilitat inicial

### 5.2.1 Pas 1: Anàlisi exploratòria

Per modelitzar el canvi produït en les creences de l'agent en l'anàlisi exploratòria amb que ha comen-
çat l'exemple, cal que ens adonem de que la classe de cosí \( \overline{w} \) pot ser partitionada en dues subclasses:

- \( \overline{w}_{a1} \): cosí de \( \overline{w} \) en què \( \forall x (B_x \Rightarrow F_x) \lor \neg \forall x (B_x \Rightarrow F_x) \) és certa.
- \( \overline{w}_{a2} \): cosí de \( \overline{w} \) en què \( \forall x (B_x \Rightarrow F_x) \lor \neg \forall x (B_x \Rightarrow F_x) \) no és certa.

El tauler generat en aquest pas de l'anàlisi, \( T'_0 \), representa la classe \( \overline{w}_{a1} \) (recordem la proposició 3). Però, el canvi en les creences de l'agent ve modulat per la generació d'una nova relació d'accessibilitat, \( R_1 \), que reflecteix quines són les alternatives doxàstiques de l'agent després d'aquest pas d'anàlisi. En aquesta relació es restringeix el conjunt d'alternatives considerades viables per l'agent, en passar de la classe de cosí \( \overline{w} \) a la seva subclassa \( \overline{w}_{a1} \) (veure la figura 5).

**Figura 5: Generació de \( R_1 \)**

Les creences de l'agent en aquest punt s'obtingueren aplicant la semàntica de Kripke modificada. Com les úniques creences accessibles serien aquelles de \( \overline{w}_{a1} \), el conjunt actual de creences de l'agent seria \( \{B(B_1), B(F_1), B(B_2), B(F_2), B(B_3), B(F_3), B(x) \Rightarrow \neg F_x\} \), \( B(F_3) \Rightarrow F_3 \) i \( \neg \forall x (B_x \Rightarrow F_x) \). Per tant, la restricció en el conjunt de cosí accessible explica l'increment en el conjunt de creences causat per la incorporació en l'anàlisi del dubte de l'agent.

**Pas 3:** Anàlisi lògica del dubte

El canvi de les creences de l'agent en el segon pas del seu procés d'investigació, quan analitza lògicament el dubte introduït, també es pot modelitzar formalment de la mateixa manera (amb una restricció de les situacions considerades com a possibles per l'agent). Per veure-ho, primer hem de considerar la següent partició de la classe de cosí \( \overline{w} \) representada per \( T'_0 \), \( \overline{w}_{a1} \):

- \( \overline{w}_{a1} \): cosí de \( \overline{w} \) en què \( \forall x (B_x \Rightarrow F_x) \) i \( \neg \forall x (B_x \Rightarrow F_x) \) són certes.
- \( \overline{w}_{a2} \): cosí de \( \overline{w} \) en què \( \forall x (B_x \Rightarrow F_x) \) és certa però \( \neg \forall x (B_x \Rightarrow F_x) \) és falsa.
- \( \overline{w}_{a3} \) (o \( \overline{w}_3 \)): cosí de \( \overline{w} \) en què \( \neg \forall x (B_x \Rightarrow F_x) \) és certa però \( \forall x (B_x \Rightarrow F_x) \) no ho és.
- \( \overline{w}_{a4} \): cosí de \( \overline{w} \) en què ni \( \forall x (B_x \Rightarrow F_x) \) ni \( \neg \forall x (B_x \Rightarrow F_x) \) són certes.

\( T_1 \) representa \( \overline{w}_{a1} \), \( T_2 \) representa \( \overline{w}_{a1} \) i \( T_3 \) representa \( \overline{w}_{a1} \). Arca podem donar una explicació semàntica al tancament lògic de \( T_1 \): l'agent s'havia adonat de que aquest tauler representava una classe de cosis lògicament contradictòries, i tancant el tauler elimina aquestes cosis del seu conjunt d'alternatives doxàstiques. La modelització semàntica de l'increment en les creences causat per l'anàlisi lògica de l'agent consisteix (veure la figura 6) en la generació d'una nova relació d'accessibilitat, \( R_2 \), que restringeix el conjunt de cosis considerades com a possibles per l'agent.

Les creences positives de l'agent no han canviat, perquè les úniques fórmules que es té constància de que són certes a totes les alternatives doxàstiques (toles les cosis accessibles mitjançant \( R_0 \) són les de \( \Delta \) i \( \forall x (B_x \Rightarrow F_x) \lor \neg \forall x (B_x \Rightarrow F_x) \)). En canvi, l'agent ara tindrà dues.creences negatives: no creurà ni \( \forall x (B_x \Rightarrow F_x) \) (perquè no és certa a les cosis de \( \overline{w}_{a12} \) i \( \neg \forall x (B_x \Rightarrow F_x) \) (perquè no és certa a les cosis de \( \overline{w}_{a12} \)). Per tant, les creences de l'agent en aquest moment serien: \( \{B(B_1), B(F_1), B(B_2), B(F_2), B(B_3), B(F_3), B(x) \Rightarrow \neg F_x\}, B(F_3) \Rightarrow F_3 \) i \( \neg \forall x (B_x \Rightarrow F_x) \), \( \neg \forall x (B_x \Rightarrow F_x) \), \( \neg B(\neg \forall x (B_x \Rightarrow F_x)) \).
Figura 6: Generació de $R_2$

- $\bar{w}_{R_2}$: cosí de $\bar{w}_B$ en què $\neg(B_o \Rightarrow F_o)$ és falsa, per tot individu concret $o$.

El tauler generat en aquest pas de l'anàlisi, $T_4$, representa la classe de cosí $\bar{w}_{R_2}$. L'evolució de les creences de l'agent en aquest moment es pot explicar considerant que aquest passaaria de creure que totes les situacions de la classe $\bar{w}_{R_2}$ són possibles a només creure que són factibles com a alternatives doxàstiques aquelles de $\bar{w}_{R_2}$. Formalment, aquest fet es representa amb la generació d'una nova relació d'accessibilitat, $R_3$, en la que es reduix el conjunt de cosí considerats com a possibles per l'agent. Aquesta situació es mostra a la figura 7. Com es pot veure en aquesta figura, les creences (positives o negatives) de l'agent no haurien canviat, malgrat la restricció en les cosí accessibles.

**Pas 4: Anàlisi lògica de $\neg(B_c \Rightarrow F_c)$**

En el tauler generat en el següent pas d'anàlisi lògica ($T_3$) ja apareixen tres fórmules que contenen la mateixa constant de Skolem $c$ (la que ja tenien més $B_c$ i $\neg F_c$). Així, en aquest tauler s'està postulant l'existència d'un individu que compleixi aquestes tres condicions. El canvi ocasionat en les creences de l'agent a causa d'aquest nou pas d'anàlisi pot modelitzar-se, novament, com una restricció en el conjunt d'escenaris considerats com a possibles per ell. Aquesta restricció es pot descriure amb l'ajuda de la partició de la classe de cosí $\bar{w}_{R_3}$:

- $\bar{w}_{R_3,1}$: cosí de $\bar{w}_{R_3}$ en què $\neg(B_o \Rightarrow F_o)$, $B_o$ i $\neg F_o$ són certes, per algun individu concret $o$.
- $\bar{w}_{R_3,2}$: cosí de $\bar{w}_{R_3}$ en què no existeix cap individu $o$ que faci certes les fórmules $\neg(B_o \Rightarrow F_o)$, $B_o$ i $\neg F_o$.

Després d'aquest pass d'anàlisi l'agent s'adonaria de que les cosís realment factibles no són totes les de $\bar{w}_{R_3}$, sinó només les de $\bar{w}_{R_3,1}$ (les representades pel tauler $T_5$). Aquest fet queda representat a la figura 8 amb la generació d'una nova relació d'accessibilitat, $R_4$, que torna a limitar el conjunt d'alternatives doxàstiques de l'agent. APLICANT la semàntica de Kripke modificada es pot veure que les creences de l'agent tampoc variarien en aquest punt de l'anàlisi.

**Pas 5: Anàlisi experimental**

L'últim pas que ha realitzat l'agent en l'exemple de la secció 5.1 ha estat l'aplicació de la dimensió experimental d'anàlisi, cercant (i no trobant) un individu que sigui un ocell no volador. Després d'aquest resultat l'agent havia tancat empiricament el tauler $T_5$, perquè no havia sigut capaç de trobar cap individu concret que pogués prendre el lloc de la cons.
Sistemes Multi Agents

Figura 8: Generació de $R_4$

Figura 9: Generació de $R_5$

No hi havia cap ocell que no volés; com aquest ha estat el cas ara ja creu que l’altra banda del dubte és la correcta.

6 Resum

El model dels móns possibles i la semàntica de Kripke només permeten la modelització d’agents ideals. Per solucionar aquest problema s’han introduït les situacions concebibles, que poden no contenir totes les tautologies o no ser deductivament tancades. S’ha definit un model abstracte d’agents no-ideals, i s’ha mostrat amb un exemple com el model de les situacions concebibles pot ser utilitzat per donar una explicació natural i intuitiva de com anirien evolucionant les creences d’aquesta mena d’agents.

Referències


Instituting Interactions Among Agents

Josep Lluís Arcos and Enric Plaza

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain).

{aros, enric}@iiia.csic.es, http://www.iiia.csic.es

Abstract

Social processes and agent interaction always take place in a specific context, and there is a school of thought in social studies that analyzes them in the framework of institutions [2]. We will present the notions of agent-mediated institution and structured virtual space and show how they are relevant for multiagent systems (MAS) in general and, more specifically, for MAS that include human agents and software agents involved in socio/economic interactions. We will show how the social interactions of human and software agents taking place in the COMRIS Project are developed using these notions and will present an example for the meeting people interaction.

1 Introduction

The main purpose of this paper is to present our approach in developing intelligent interfaces for interacting with information. We are interested in societies of “personal software agents” that inhabit an information space where they can access information resources and need to interact with other agents in order to achieve their specific goals. A main issue here is how this “societies” of agents can be designed and their interaction patterns can be specified (or even enforced). The approach we are taking is that social processes and agent interaction always take place in a specific context, and there is a school of thought in social studies that analyzes them in the framework of institutions [2]. We will show how the social interactions of human and software agents taking place in the COMRIS Project are developed using the notion of agent-mediated institution.

From the point of view of the user, our agent-based approach offers an intelligent interface to a specific information space. However, the user does not perform “online browsing” (the modern avatar of GUIs). In our approach the user has the following interaction modes with the agents:

1. the user instructs the agent on the task to perform and the user preferences to guide the agent’s behavior,
2. the user can access the reports on the current status of her agents, including achievements in the task she has delegated,
3. the user may get a real time notification of some agent, but this should happen only if this notification is relevant to the user in the specific context she is in that moment,
4. the user may give new instructions or preferences to the agents, and evaluate the agent’s performance.

In this paper we will only deal with the 3rd mode of agent communication in the context of the COMRIS project. Indeed, one of the major efforts in our current work is the distinction between the interestingness of a particular information and the relevance for the user of that particular information at a particular time and in a particular place. For instance, a user has delegated to an agent the task of finding a person also interested in “visual languages”, while she has also instructed other agents with other tasks. Now the question is: which information is more relevant to the user when she is...
at a particular place? Now imagine that the agent has found a person interested in "visual languages" and that this person enters the same room the user is in: clearly this information is now more relevant than it was before—and more relevant than other "competing" information items. This kind of agent interaction is the one we are studying in the COMRIS project, focusing on the particular social context of an international conference.

This particular social context (the conference) is analyzed in the framework of agent-mediated institutions (the conference institution). This analysis supports the description of the tasks to be developed by the personal agents in the virtual space (the information space containing the data of the conference organization and participants) including the interaction and negotiation protocols that are shared by the agents. Moreover, this social context allows us to determine which information is relevant for a user (immersed in a physical, social and institutional context, in a particular moment).

The COMRIS project, belonging to the European Network for Intelligent Information Interfaces\(^2\), has selected the Conference Center as the thematic space that provides the context of the work. In the Conference Center, like the Annual Esprit meeting in Brussels, many people are attending. The goals of the participants can be as diverse such as presenting their research results, contacting with people with similar interests, meeting EU officials in person, or engaging any kind of discussion. The goal of the COMRIS project is to provide a framework for improving the participation of people in these huge events.

This paper is organized as follows: first we will present the COMRIS Conference Center and the notion of agent-mediated institutions. Then, we will focus on the Virtual Space and on the software agents representing user interests and inhabiting this virtual space. Next, we will illustrate the use of the agent-mediated institution framework for designing interest-based navigation (IBN); IBN is the process by which agents identify other agents with similar interests in virtual space. The paper closes with a discussion on the main issues of the

\(^2\)COMRIS is part of the Inhabited Information Spaces schema. The aim of Inhabited Information Spaces is to develop virtual information spaces in which people, who may be geographically dispersed, interact with each other and with information sources and services. These inhabited information spaces provide a common environment for interacting human agents and software agents.

2 COMRIS Conference Center

A conference takes place in a physical setting, the conference center, where different activities (scenes) take place at different locations by people (human agents) that take different roles (speaker, session chair, participant, organization staffer, etc.) in the pursuit of their respective interests. Along the course of time during the conference people pursue their interests moving around the physical locations, abandoning some activity to take up a new one. In a moment in time people are physically distributed along the conference center and involved in some interactions with other people: this physical space is shown on the right of Figure 1. We can easily think about the spatial proximity relations that exist among people in this space. However, if we think about an informational space where the past background and current interests of the conference participants are represented, we can think of a new kind of proximity relation that is a function of the similarity among people's interests and backgrounds. This informational or virtual space is shown to the left of Figure 1.

The COMRIS project is about studying and exploiting the synergy of these two spaces, and their relationship, with the objective of supporting and improving the achievement of a participant's interests while attending a conference. The approach of COMRIS is to develop software agents inhabiting the virtual space that realize some specific activities on behalf of some interest of the participants in a conference. Specifically, a Personal Representative Agent (PRA) is an agent inhabiting the virtual space that is in charge of advancing some particular interest of a conference participant by searching for information and talking to other software agents. An example of a PRA activity is looking for other participants for setting up appointments about a particular common topic of interest.

When a PRA has gathered some information interesting for a participant it will try to push that information to him or her\(^3\). In the COMRIS project, the participant is holding a wearable computer device, nicknamed the parrot, that runs a speech-

\(^3\)As shown in the rightward arrow of Figure 1.
generation program and that is able to unobtrusively convey (push) that information to him or her using spoken language. It is clear, however, that not any information is relevant for the user (the conference participant) at any point in time. Rather, the physical situation of the person and the concrete activity in which he or she is involved will determine—or at least bias—whether a particular information is relevant for her at a particular moment in time.

Moreover, several PRA’s may compete for access to a person’s attention in their “information push” activity. COMRIS develops a competition for attention mechanism [4] that deals with both sides of the equation: several information reports that compete to be pushed to the participant and the physical situation of the participant that biases which of these information reports is more relevant for him or her at that moment in time. We can picture this competition for attention mechanism as a process that, biased by the context information of the physical space, selects a “winner” PRA and pushes the information report to his/her parrot where it will be orally rendered. For each conference participant there is a Personal Assistant (PA) agent in the virtual space that embodies this process and “owns” the channel communicating with the participant’s parrot.

2.1 Interactions, Rules, and Institutions

Determining the relevance is only a part of the issue: not only people are involved in located activities in the physical space—attending the talks of a thematic session, meeting a colleague to discuss about a topic, etc—the software agents are also meeting among them in pursue of some person’s interest. In order to analyze this complex network of agents interacting we use the notion of an agent-mediated institution [1].

However, it is first appropriate to consider the kinds of activities that take place into a regular Conference Center: a typical participant pays the registration, registers at the information desk, assists to thematic sessions, makes appointments and later attends or skips them, etc. Although we can imagine whatever activity taking place in a Conference Center it is true nonetheless that some
of them occur regularly, and the persons involved know some rules of behavior for these activities—even if most of this rules are tacit. In fact, a conference may be considered as an institution both because these recurring activities occur in it and because conferences are indeed set up for supporting and favoring most of these activities—while other activities are discouraged or plainly forbidden.

The activities the Conference Center is willing to favor and uphold are the ones COMRIS is interested in analyzing (with agent mediated institutions) and in providing a computational framework that supports and facilitates them (with the Personal Representative Agents inhabiting the virtual space).

Succinctly, an agent-mediated institution (AMI) is the computational realization of a set of explicit enforceable restrictions imposed on a collection of dialogical agent types that concur in space and time to perform a finite repertoire of satisfiable actions. The advantages of using this notion of agent-mediated institution are the following:

- it allows to describe in a comprehensive framework the roles and interactions of both human and software agents in a specific setting (the institution)
- it makes explicit the relationship between the computational framework developed by COMRIS and the existing organization of a conference (the institution we will call the Conference Center)
- it clarifies the difference between
  1. the “rules of the game” enforced by the Conference Center upon human and software agents behavior and
  2. the particular strategies the human and software agents may recur to in pursuing individual goals
- it helps to specify the social relationships such as
  1. authority, which agent types (usually roles) can order specific tasks to other agents or change specific goals of other agents
  2. entitlements, like which information is accessible and forbidden to an agent type

2.2 Agent-mediated Institutions

An Agent-Mediated Institution (AMI) is defined in terms of a collection of agent identifiers, that can play different roles, with a set of relevant social relations. These agents can participate in different scenes defined in the AMI. Each scene takes place in a location and involves different agents assuming each of them a role. Agents exchange illocutions in a common interaction protocol and a common ontology defined in the scene.

Formally, an Agent-Mediated Institution is defined as a tuple $AMI = (DF, PS, BR)$, where,

1. $DF$ is a dialogical framework
2. $PS$ is a performative structure,
3. $BR$ are the rules of behavior to which participating agents are subject to.

**Dialogical Framework** A Dialogical Framework, $DF$, is a tuple $DF = (Agents, Roles, SR, Loc, L, CL)$, where,

1. $Agents$ is a set of agent identifiers,
2. $Roles$ is a set of agent types,
3. $SR$ is a set of relevant social relations and individual distinctive characteristics that may be relevant for the description of an institution,
4. $Loc$ is a set of locations,
5. $L$ is an object language (a domain ontology), and
6. $CL$ is a communication language.

Concerning the Conference Center institution, we can summarily exemplify of a dialogical framework $DF$ as follows:

1. $Agents$ are the set of conference registered persons, conference organization staff, Personal Assistants (PAs), and Personal Representative Agents (PRA1s),
2. $Roles$ are the set of PA’s and PRA’s types, as well as the types of registered persons (presenters, listeners, invited speakers, etc) and the types of conference staff members (PC member, session chair, information desk staffers, etc),
3. Social relations are the set of relations among roles in the Conference Center (e.g., the authority of a conference chair over session chairs).

4. Locations are the collection of places where agents interact. There are locations in physical space and locations on virtual space. Physical locations (such as the main hall, a session room, a company's booth, or the cafeteria) are inhabited by human agents. Locations in virtual space, called subspaces, are inhabited by software agents. Examples of subspaces are the meeting point, special interest groups, and virtual discussion rooms. While a human agent can only be present in one physical location, a software agent may be present in several subspaces at the same time.

5. Ontology contains the elements of which the agents talk about, e.g., agendas, time-slots, projects, consortiums, topics of interests, thematic sessions, etc.

6. Communication language contains the set of performatives used in the Conference Center (e.g., INFORM, REQUEST, etc).

**Performative Structure** A performative structure, PS, is a set of interdependent located scenes. Each scene is defined as a set of agents who are each to assume a given role, each pair of agents who ever exchange an illocution are subject to a common atomic interaction protocol.

Protocols are specified as finite state machines where state transitions are labeled by illocutions and states have associated memory of commitments.

Scenes are interdependent. There are some initial scenes that can be required before starting other scenes. For instance, before arranging appointments with other participants, an agent is required to explore the collection of participants with interests similar to its own interests. We call this exploration scene the **interest-based navigation** (IBN) scene. The IBN scene precedes also other scenes such as **propagandist** scenes where, for instance, the PRA of a thematic session chairperson or an exhibition booth tries to find and attract persons that might be interested in the topics of the session or the exhibition. The IBN scene is explained in detail on Section 4.

Since many agents participate in a conference, the goal of the IBN scene is to explore the space of agents in order to identify potentially interesting interactions with other agents. The IBN scene takes place in the Meeting Point subspace. Any agent can enter the Meeting Point subspace and adopt either (i) an active role presenting itself to other agents or requesting information from other present agents, or (ii) adopt a passive role and wait for other agents to take the initiative. Then, according to agent's internal strategy, and when the agent has established a proximity relation with other agents, other actions such as arranging appointments or proposing a project consortium can take place.

**Rules of Behavior** Even though scene protocols are necessary to describe agent interactions, they may be insufficient to make fully explicit the "rules of the game" which all participating agents are supposed to follow in a given institution. These rules, BR, will be defined as the **individual rules of behavior** of each agent role. Such rules are, ideally, part of the internal model of each participating agent, but may be required and/or enforced by the institution.

The interaction protocol of a scene defines a set possible decisions (or paths) that an agent may take following its role. Rules of Behavior (RB) specify further constraints upon which decisions can be taken by an agent following its role in this scene for a given context or situation.

For instance, in the Interest Based Navigation (IBN) scene, when an agent enters the meeting point subspace, the first step in the protocol is that the agent performs a show illocution by which its appearance description is sent to all agents present in the subspace. Moreover, there is a COMRIS CC rule of behavior that forbids the agents to have more than one appearance or modify their appearance.

3. **The Virtual Space**

The virtual space is populated by software agents, called Personal Representative Agents (PRAs), that act as representatives of persons. The virtual space is not a flat space. The virtual space is composed of several locations where an agent can go. We call these virtual locations subspaces. Each subspace provides a set of different facilities and has a
collection of properties.

There are some predefined subspaces in the virtual space. Each subspace has associated a collection of properties such as:

- For each subspace only a specific collection of scenes may occur.
- A subspace may allow multicasting of messages. Multicasting facilities require notifications of when a PRA enters and leaves to the subspace.
- Some subspaces may limit the number of participants (limited capacity).
- Fairness in sending and receiving messages is assumed inside a subspace.
- Timeliness in sending and receiving messages is not a required capability in a subspace. Nevertheless, some subspaces with limited capacity can assume timeliness.

A PRA can go in and out to a subspace many times, can reside in multiple subspaces at the same time, and can participate in several scenes in each subspace.

For instance, in the Conference Center there is a subspace, called the Meeting Point subspace, where interest-based navigation takes place. A given PRA can enter, leave, and announce itself following its own strategy (e.g. sending its presentation and appearance immediately after it goes in). Moreover, since the Meeting Point subspace allows multicasting, a PRA will receive all announcements performed for other agents presents at this subspace.

Other examples of subspaces are Discussion rooms. For instance, a group of PRAs involved in a consortium and project formation can require a private subspace for negotiating. When the PRAs reach a common resolution, they can book a physical private room where the participants represented by these PRAs will physically meet to proceed with the consortium proposal negotiation.

A participant may have several PRAs pursuing her different interests. The participant has to instruct each PRA specifying a specific presentation (e.g. a topic and possibly a collection of related subtopics), an appearance for interacting with other PRAs, and a collection of actions (e.g. meeting people, attending to talks, making appointments, etc) in which the PRA can participate for achieving the participant’s interests. The collection of actions is provided by the Conference Center definition as a set of scenes and roles in which a PRA can participate.

Besides the parrot device, during the conference each participant will be able to consult the progress of their PRAs using a screen-based interface. Moreover, the participant can change the instructions of a PRA either to tune the work of the PRA or to adapt the PRA to her new interests.

4 Meeting People

In the interest-based navigation scene (IBN scene) a PRA (playing the role of initiator) explores the interests and availability of other PRAs (playing the role of receivers) in order to construct a proximity relation. This proximity relation will be the input for deciding interactions with other PRAs such as making appointments, push information, etc.

The scene takes place in the Meeting Point subspace of the virtual space. A given PRA (initiator) can maintain several interactions with several PRAs (receivers) at the same time. Each interaction initiator/receiver will be described as a different scene between the initiator and one receiver (That is to say, a given PRA can participate in several IBN scenes at the same time).

An IBN scene involves requesting two different kinds of descriptions: agents appearances and agents presentations:

- **Appearance** is a collection of features describing the view an agent wants to offer to the other agents. Appearance is unique, that is to say all the other agents perceive the same collection of features. The goal of appearance is to initiate any further potential agent interactions.

- **Presentation** is a collection of features that an agent chooses to send to another agent describing its relevant skills to that other particular agent. That is to say, a presentation is an individually tailored description made available depending of the appearance of the receiving agent. The goal of presentation is to initiate any further potential inter-agent interest investigations.

A given PRA; before starting an IBN scene with another PRA, has to go into the meeting point...
subspace (transition from D0 to S0 in Figure 2). When \( PRA_i \) goes into the meeting point subspace, it performs a show allocation by which its appearance description is sent to all agents present in the subspace. Moreover, \( PRA_i \) can look the appearance of all other present agents. Then, \( PRA_i \) can start IBN scenes with the present agents it chooses to. The scene is composed of the following phases:

(A) At the beginning (S0), the initiator \( PRA_i \) can either (B) request the presentation of a given \( PRA_i \) present in the Meeting Point (S1), or (C) can send its presentation to a present \( PRA_i \) (S5).

(B) In this phase \( PRA_i \) requests the presentation of a \( PRA_i \):

1. Given this presentation request (S1), the receiver \( PRA_r \) can answer sending it to \( PRA_i \) (S2).

2. When \( PRA_i \) receives the presentation card from \( PRA_r \) (S2), \( PRA_i \) evaluates it, according to the interests that \( PRA_i \) represents, and can either (i) decide that \( PRA_r \) is not promising enough, (ii) request a detailed interest profile to \( PRA_r \) (S3), or (iii) adopt a passive role sending its presentation to \( PRA_r \) (S5) and waiting for \( PRA_r \) to take the initiative.

3. Given a request of a detailed interest profile (S3), the receiver \( PRA_r \) can answer sending its interest profile to \( PRA_i \) (S4).

4. When \( PRA_i \) receives a detailed interest profile from \( PRA_r \) (S4), \( PRA_i \) evaluates them, according to the interests that \( PRA_i \) represents, and can either decide that \( PRA_r \) is not promising enough, or can start new scenes (D1) with \( PRA_r \) such as making appointments, pushing information, etc.

(C) In this phase \( PRA_i \) sends its presentation to a \( PRA_r \) in the Meeting Point:

1. When \( PRA_r \) receives the presentation from \( PRA_i \) (S5), \( PRA_r \) evaluates the presentation, according to the interests that \( PRA_r \) represents, and can either (i) decide that \( PRA_i \) is not promising enough, (ii) request a detailed interest profile to \( PRA_i \) (S6), or (iii) adopt a passive role sending its presentation to \( PRA_i \) (S2) and waiting for \( PRA_i \) to take the initiative.

2. Given a request of a detailed interest profile (S6), the initiator \( PRA_i \) can answer sending its interest profile to \( PRA_r \) (S7).
ates them, according to the interests that PRAs represents, and can either decide that PRA is not promising enough, or can start new scenes (D2) with PRA; this is analogous to the previous phase (B), except that now PRA will become the initiator and PRA will become the receiver.

The decision of starting a new specific scene with another PRA (D1 and D2) depends on the PRA strategy and the instructions from the user. Since a PRA can reside in multiple subspace at the same time, the interest landscape can be continuously updated while the PRA is already interacting with other PRAs in more advanced scenes. Each PRA can decide to leave—temporally or permanently—the Meeting Point subspace. Such decision implies that outside of the Meeting Point subspace a given PRA only can interact with acquaintances (previously contacted PRAs). Decisions such as remaining into the Meeting Point subspace, leaving the Meeting Point subspace, or coming back into the subspace are dependent of the internal strategy of each PRA.

5 Some other Scenes

We are designing the functionality of the COMRIS Virtual Space specifying the scenes the agents can work with in order to achieve the tasks they are charged with. This functionality includes the following scenes:

1. Making Appointments: composed by two scenes called appointment proposal scene (APS) and appointment coordination scene (ACS). First, in the APS scene a participant, playing the role of initiator, explores the interests and availability of another participant in order to identify the most promising appointments. Then, in the ACS scene, the initiator negotiates with the other participant for a specific meeting time and content.

2. Proximity Alert: in this scene a participant is informed that she is physically near to another person with similar interests. Variants of this include proximity to exhibitors booth or thematic sessions with topics similar to the user’s.

3. Commitment Reminder: this is a type of alert that reminds a participant of the proximity of a deadline she is committed to (e.g. an appointment, but also commitments with the Conference organization, like chairing a session that is about to start, or boarding a bus that is about to leave for a tour the user has paid for).

4. Consortium and Project Formation: one of the more important tasks in conferences like the ESPRIT Week. Similar to appointment scenes but involves several people with different interests that are relevant to separate aspects of a project.

5. Propagandist: in this scene a PRA from someone like a thematic session chairperson or an exhibition booth responsible tries to find and attract persons that might be interested in the topics of the session or of the exhibition. Similar to the Appointment scene but there is no need to negotiate a time-slot (session is fixed and exhibitor would prefer to remind the user when he is near and not in a particular timeslot—also similar to Proximity Alert).

6 Conclusions

In this paper we have used the notion agent-mediated institutions as the basis for developing a computational framework for social interactions of human and software agents. Agent-mediated institutions provide a way to establish a shared formalism for describing and analyzing a set of social situations where human and software agents interact in a meaningful way. These situations are described using the notions of scenes (in which different agents can be involved) and of roles. Moreover, the structure of scenes shapes the overall AMI.

The notion of agent-mediated institutions leaves open the internals of each specific agent: the preferences and the decision procedure by which a software agent decides to perform a particular action (e.g. a particular offer) are the agents own and not part of the AMI. However, only in a particular framework of interaction shared by the involved agents this decision processes can take place, and this is precisely what agent-mediated institutions has to offer.

Using this notion of notion of agent-mediated institutions, we have presented the Conference Center institution that is being developed within the COMRIS project.
Acknowledgements

The research reported in this paper is supported by the ESPRIT LTR 25500-COMRIS Co-Habited Mixed-Reality Information Spaces project.

References


JIM
A Java Interagent for Multi-Agent Systems

Francisco J. Martin, Enric Plaza, Juan A. Rodríguez-Aguilar, and Jordi Sabater

IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Barcelona, Spain
Vox: +34-3-5809570, Fax: +34-3-5809661
{martin, enric, jar, jsabater}@iiia.csic.es

Abstract

In this paper we introduce an interagent as an autonomous software agent which handles (intermediates) the communication and coordination between an agent and the agent society wherein this is situated. According to our proposal, interagents constitute the sole and exclusive means through which agents within a multi-agent scenario interact. With this aim, we have developed JIM, a general-purpose interagent that provides agents with a highly versatile range of —before and during the agent’s run-time— programmable communication and coordination services. The development of JIM lies in the framework of the SMASH project. SMASH addresses the construction of multi-agent systems to tackle complex problems of distributed nature in hospital environments. Two main benefits stem from the usage of JIM: on the one hand, it permits agents to reason about both communication and coordination at a higher level of abstraction, and on the other hand, it provides a complete set of facilities that allows agent engineers to concentrate on the design of their agents' inner and social behaviour.

Keywords: Multi-Agent Systems, Interagents, Communication, Coordination

1 Introduction

There exists a number of problems which involve multiple sources of knowledge and, thereby, can best be addressed using a multi-agent system — a computational system composed of several interacting agents which cooperate with one another to solve complex tasks. Furthermore, the deployment of multi-agent systems permits to benefit from a number of advantages — such as parallelism, robustness or scalability — that a single agent working isolatedly can not offer itself.

Currently, we are partners of the SMASH project[2], a collective, joint effort involving several research institutions that addresses the construction of a general-purpose heterogeneous rational multi-agent architecture, and the development —on a computational implementation of this architecture— of prototype multi-agent systems with learning capabilities that cooperate in the solution of complex problems in hospital environments. The development of such type of multi-agent systems poses the question of how to integrate a set of heterogeneous agents — agents developed by different people for different purposes and in different languages — within a common setting. In order to achieve this goal, two major issues need to be addressed: (highly flexible) communication and coordination among the agents composing the multi-agent system. Instead of letting agents deal themselves with such issues, we opts for introducing an autonomous software agent that we call interagent which handles (intermediates) the communication and coordination between specific

1 Cooperation with other agents is paramount: it is the raison d'être for having multiple agents in the first place in contrast to having just one" H. S. Nwana [16].
agents and the agent society wherein they are situated (see Figure 1). According to our proposal, interagents constitute the sole and exclusive means through which agents within a multi-agent scenario interact.

In our proposal, the functionality provided by an interagent will highly depend on the role played by the agent interacting with it. Thus we distinguish two distinct roles for agents making use of interagents: i) the user of an interagent regards it as the sole and exclusive means through which it can interact with the agent society thanks to the set of communication and coordination services provided by the interagent, but previously defined by the owner; ii) the owner of an interagent is provided with a wide range of facilities to either load or program into the interagent the communication and coordination services that the user is allowed to employ. Needless to say, an agent can possibly play both roles at the same time.

In this work, we present JIM, a general-purpose interagent which provides agents with a highly versatile range of programmable——before and during the agent's run-time——communication and coordination services. JIM is being implemented in Java in order to ensure platform independence.

The remainder of this paper is organized as follows. Section 2 analyzes which features distinguish interagents from related approaches. Section 3 describes the communication services offered by interagents, while Section 4 presents the coordination services. In section 5 we illustrate how JIM is being used in different agent-based applications, and finally, Section 6 draws some concluding remarks.

2 Related Work

For several years, agent-based software engineering has faced the matter of enabling heterogeneous programs written by different people, at different times, in different languages and with different interfaces to communicate and interoperate [10]. Researchers in the ARPA Knowledge Sharing Effort have proposed agent communication languages (ACLs) as the means to allow the exchange of knowledge among software agents in order to facilitate their interoperation [10]. Generally, an ACL is composed of three main elements: an open-ended vocabulary appropriate to a common application area, an inner language (KIF—Knowledge Interchange Format) to encode the information content communicated among agents, and an outer language (KQML—Knowledge Query and Manipulation Language) to express the intentions of agents.

Nowadays, KQML has become the communication language par excellence in agent-based systems. However, when several computational entities interact by exchanging messages a higher level of interaction concerned with the conventions that they share during the exchange should be addressed [6]. This level of interaction is not supported by KQML, whereas coordination languages —like COOL[6]— allow such conventions to be explicitly expressed. Making shared conventions explicit allows interdependencies among agents' activities to be managed.

Apart from COOL, other agent building tools such as AgentTalk[1] or JAFMAS [7] also provide coordination constructs. JAFMAS, for instance, provides a generic methodology for developing speech-act based multi-agent systems using coordination constructs similar to COOL. However, many more agent building tools that have not addressed this issue yet.

Interagents —like KQML facilitators[17]— are inspired by the efficient secretary metaphor already introduced in the Actors model of concurrent computation [3]. Nevertheless, interagents (unlike KQML facilitators) offer the coordination level required by agents to cooperate in non-trivial ways. On the other hand, unlike KQML facilitators, interagents have no knowledge about the reasoning capabilities of their users [21], though they are aware about their users' plans thanks to conversation protocols.

By introducing interagents, and concretely JIM, we try to make headway with respect to other agent building tools offering a programmable communication and coordination module whose behaviour can be specified either before or during its user's run-time by both the agent engineer and the (user) agent itself. In addition, another major advantage of using interagents is that they permit agents to reason about communication and coordination at a higher level of abstraction, making implementation details transparent to their users.

\footnote{For an account of other agent building tools refer to \url{http://www.ececs.uc.edu/~mnoschak/tools.html}.}
3 Communication Services

In our proposal, communications among agents are based on *message-passing* \(^3\). However, agents do not exchange messages directly but by means of interagents. Thus, an interagent is informed by its user about the message to be sent and its address, and then the interagent carries out all the operations needed to deliver it correctly.

An interagent and its user can communicate in two ways:

a) through (TCP-)stream-sockets — in case that an interagent and its user are two distinct computational processes (residing in the same computer or not)

b) through memory I/O streams — in case that an interagent and its user are two distinct threads residing in the same process space

Concerning communication, an interagent provides its user with the following communication services based upon TCP/IP:

- queueing of outgoing messages from its user and queueing of incoming messages from (the interagents of) other agents;
- *asynchronous communication* between agents;
- *synchronous communication* between agents (implemented on top of *buffered asynchronous communication* between interagents);
- agent naming services (white pages);
- handling of expired messages and automatic recovery of transmission errors.

We have devised the Inter-Agent Protocol (IAP) to be used as interaction protocol between interagents and between an interagent and its user [15]. IAP is an application-level protocol based on the Hypertext Transfer Protocol (HTTP) [8]. IAP overrides the set of request methods provided by HTTP and constrains the body of a message to be a *performative* of an agent communication language. For instance, the POST method is used to indicate that the performative enclosed in the message body must be addressed to the receiver indicated in the corresponding field of the performative, whereas the GET method is used to retrieve the performative(s) queued by an interagent such that match the pattern of performative enclosed in the message body.

On the other hand, interagents support a subset of KQML performatives, whose syntax has been

---

\(^3\)The *message-passing paradigm* provides functionality equivalent to that found in *remote procedure call* (RPC) or *tuple-space* paradigms [11].
extended with the reserved parameter keywords shown in Table 1:

- Each performative exchanged between agents is associated to a particular conversation identified by the conversation parameter.

- The delay parameter indicates how long a message queued in an interagent is postponed before this starts to process it. In this way, an agent can tell its interagent to deliver a given performative after a certain amount of time.

- The time-out parameter indicates the maximum period of time that an agent conforms to wait for receiving a reply to the performative.

- The time-to-live parameter indicates the lifetime of the performative once it has been queued in an interagent. Thus, when this time expires, the message is thrown away by the interagent where it is queued, and the sender receives back an error message.

4 Coordination Services

Interagents offer the coordination level required by agents to cooperate in non-trivial ways. An interagent allows interdependencies between agents’ communicative acts to be ordered. These interdependencies can be defined declaratively inside each interagent by means of a conversation protocol. A conversation protocol represents the conventions adopted by agents when interacting through the exchange of messages. A conversation protocol can also be seen as a agent’s plan to achieve some goal[6]. We model and implement conversation protocols as a special type of pushdown automaton. Pushdown automata, unlike finite state machines, allow context to be stored and to be subsequently retrieved for an ongoing conversation.

Conceptually, we have decomposed a conversation protocol into (see Figure 2):

- A finite state control. Each state in the finite state control represents the situation of the interagent’s user during an ongoing conversation.

- An input list continuously traversed in search of a performative which can produce a transition in the finite state control. If such message is found, it is dispatched and, thereby, removed from the input list. Note that the way of traversing the input list differs from the one employed by classic pushdown automata whose read only input tapes is traversed from left to right (or the other way around).

- A pushdown list where the context of an specific conversation can be stored and subsequently retrieved.

- A finite set of transitions. Each transition in a conversation protocol indicates:
  
  1. what message has to be either sent or received to produce a move in the finite state control; and
  2. whether it is necessary to store (push) or retrieve (pop) the context using the pushdown list.

Formally, a conversation protocol, like a pushdown automaton [4], is a 7-tuple:

\[ P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \]

where

- \( Q \) is a finite set of state symbols that represent the states of the finite state control;

- \( \Sigma \) is the finite input list alphabet composed of all possible performatives that an interagent can deal with;

- \( \Gamma \) is the finite pushdown list alphabet composed of all possible performatives that an interagent can store;

- \( \delta \) is a mapping from \( Q \times \Sigma \times \Gamma \) to the finite subsets of \( Q \times \Gamma^* \) which indicates all possible transitions that can take place during a conversation.

- \( q_0 \in Q \) is the initial state of a conversation.

- \( Z_0 \in \Gamma \) is the start symbol of the pushdown list.

\textsuperscript{4}Conversation protocols lack \( e \)-moves, what makes a significant difference with respect to classic pushdown automata.
Table 1: Reserved parameter keywords and their meanings introduced by JIM

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>:conversation</td>
<td>Identifier of the conversation wherein the performative is uttered</td>
</tr>
<tr>
<td>:delay</td>
<td>Indicates how long a message must be delayed by an interagent before it starts to deliver it to the addressee.</td>
</tr>
<tr>
<td>:time-out</td>
<td>Maximum period of time an agent accepts to wait for receiving a reply to the performative.</td>
</tr>
<tr>
<td>:time-to-live</td>
<td>Life time of the performative after being queued in an interagent.</td>
</tr>
</tbody>
</table>

- $F \subseteq Q$ is the set of final states representing possible final states of a conversation.

Messages queued by interagents can queue-jump only if they produce a transition in the conversation protocol. Therefore, in certain way interagents constrain what an agent can utter and hear, and when. For instance, Figure 2 shows instantiation c-87 of a conversation held by agents $A$ and $B$. The following transition:

$$\delta(q_6, \text{answer}, \text{ask}\alpha) = \{(q_4, \alpha)\}$$

indicates in such conversation protocol that when:

- such conversation is in state $q_6$; and
- the performative corresponding to the topmost message on the pushdown list is ask; and
- a message with the performative answer is in the input list, and its sender, receiver and the label in keyword :in-reply-to match respectively the receiver, sender and label in keyword :reply-with of the topmost message on the pushdown list.

then the following move

$$\delta(q_6, \text{answer}, \text{ask}\alpha) \vdash (q_4, \alpha)$$

takes place in the finite state control. As a result, conversation c-87 switches to state $q_4$, the corresponding message is extracted from the input list and forwarded to the corresponding addressee, and the message on the top of the pushdown list is popped out.

4.1 Conversation Protocol Definition

An interagent can support a wide range of conversation protocols that can be defined declaratively and stored into the conversation protocol library that each interagent has. Such library can be updated in two ways:

Statically. Before the interagent's user run-time, as an item provided by the agent engineer.

Dynamically. An owner can interactively define new conversation protocols (or modify existing ones) at run-time using a conversation protocol definition and manipulation language based on the set of reserved coordination performatives in Table 2. These performatives allow to either fully define a new conversation protocol or modify stored conversation protocols by adding or deleting states or transitions.

Once defined, conversation protocols must become instantiated in order to be used for coordinating the interaction between agents.

This capability of allowing agents to define and modify themselves their conversation protocols at run-time happens to be an innovative feature of our proposal, distinguishing interagents from other approaches like COOL[6] or JAFMAS[7].

5 Applications

JIM our Java-based implementation of a general-purpose interagent, is being used as the communication and coordination support for the agents developed in the framework of the SMASH project. Concretely, in this section we describe Plural –
Figure 2: Conversation protocol for the foreign evaluation capability of Plural [12]. The conversation protocol followed by Plural agents during a foreign evaluation is modelled and implemented in an agent as a pushdown automaton $P$ such that: $P = \{(q_0, q_1, q_2, q_3, q_4, q_5), \text{evaluate, ask, answer}, \{Z, \text{evaluate, ask, answer}\}, \delta, q_0, Z, \{(q_2, q_3, q_6)\})$. Messages followed by / stand for performatives sent by the interagent’s user, whilst messages preceded by / stand for performatives received by the interagent’s user.

an extension of the knowledge representation language Noos for developing agent-based systems with learning capabilities – and Fishmarket – an agent-mediated electronic marketplace⁵.

5.1 Plural Noos

Noos is an object-centered language based on feature terms [5]. Noos can furnish applications with the reasoning capabilities required by intelligent agents, nonetheless, applications developed in Noos lack communication abilities except for graphical user interface. We are developing Plural, a seamless extension of Noos based on interagents which promotes Noos to an agent-oriented language [12]. On top of the basic mechanism offered by interagents Plural extends Noos with two new constructs with the same nature as the rest of Noos constructs: defforeign and defmobile. These constructs provide Noos with two new ways of constructing feature terms –foreign refinements and mobile refinements– which, in turn, allow an agent to remotely evaluate methods owned by other agent –foreign evaluation– and send methods to other agents to solve problems on its behalf –mobile problem solving methods [14].

In this way, agents in Plural do not communicate directly with one another, instead, they rely on interagents which offer a range of programmable communication and coordination facilities. Each agent has attached its own interagent which constitutes the sole and exclusive means through which a Plural agent interacts. An interagent gives a permanent identity to its user and enforces the conversation protocols (defined for each construct) –thus establishing what messages can be forwarded, to whom and when.

The declarative fashion of the conversation protocols offered by interagents is what allows Plural
Table 2: Reserved coordination performatives, for agent A and interagent I

<table>
<thead>
<tr>
<th>Performative</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>define-conversation</td>
<td>A defines in I a new conversation protocol</td>
</tr>
<tr>
<td>add-state</td>
<td>A adds a new state to a conversation protocol residing in I</td>
</tr>
<tr>
<td>delete-state</td>
<td>A deletes a state from a conversation protocol residing in I</td>
</tr>
<tr>
<td>add-transition</td>
<td>A adds a new transition to a conversation protocol residing in I</td>
</tr>
<tr>
<td>delete-transition</td>
<td>A deletes an existing transition from a conversation protocol residing in I</td>
</tr>
</tbody>
</table>

To incorporate new capabilities which require that agents follow some convention in the exchange of messages. These conventions will be provided to interagents by means of conversation protocols. We have provided interagents with a conversation protocol for each new Plural construct incorporated at the level of the knowledge representation language. For instance, Figure 2 shows the conversation protocol for the foreign evaluation capability of Plural. Interagents allow all the underlying exchange of messages needed by those constructs to be transparent to Plural agents.

The new capabilities embedded into Plural enable agents to adequately communicate and coordinate in order to exchange knowledge. Plural can be thought as an extension of a knowledge representation language with both an agent communication language and an agent coordination language which are provided at the knowledge representation level by means of some new constructs and carried out by interagents. These constructs allow the exchange of knowledge to be performed at the knowledge representation level transparently to the agent communication and coordination languages chosen.

The capabilities that now Plural incorporates are those currently under active research by new programming paradigms, namely distributed state, foreign method invocation, and remote evaluation. In order to provide Noos with such capabilities there was no need to re-write Noos. Thus, interagents shows a way in which legacy software can also profit from mobile code paradigms.

The foreign evaluation and mobile problem solving methods capabilities of Plural have been used to devised two cooperation modes among agents with learning capabilities — Distributed Case-based Reasoning (DistCBR) and Collective Case-based Reasoning (ColCBR) [18]. These modes of cooperation are based on reusing the experience acquired by other agents. Which agent owns the similarity-based reasoning method used to solve a problem — the sender of the problem or the addressee agent — is the basic difference between both methods [18]. In DistCBR an agent is delegated to solve a task on behalf of another agent. DistCBR is supported by the foreign evaluation capability of Plural. In ColCBR, an agent in addition to the task to be achieved sends the method to solve that task. ColCBR is supported by the mobile methods capability of Plural. Such cooperation modes are being used in CHROMA a distributed system for recommending a plan for the purification of proteins from tissues and cultures [18] and in CoDiT [13], a multi-agent system for therapy recommendation in diabetic patients in the framework of the SMASH project[2].

5.2 The Fishmarket

The fish market can be described as a place where several scenes run simultaneously, at different places, but with some causal continuity. The principal scene is the auction itself, in which buyers bid for boxes of fish that are presented by an auctioneer who calls prices in descending order — the downward bidding protocol. However, before those boxes of fish may be sold, fishermen have to deliver the fish to the fish market, at the sellers’ registration scene, and buyers need to register for the market, at the buyers’ registration scene. Likewise, once a box of fish is sold, the buyer should take it away by passing through a buyers’ settlements scene, while sellers may collect their pay-
ments at the sellers’ settlements scene once their lot has been sold.

In [20, 19] we present the Fishmarket, our current implementation of an electronic auction house based on the traditional fish market metaphor, subsequently extended to become a multi-agent testbed[20]. This implementation allows to run auctions over the Internet that permit both human and software agents to participate. Thus, buyer and seller agents can trade goods as long as they comply with the Fishmarket institutional conventions. Those conventions that affect buyers and sellers have been coded into their interagents, which constitute the sole and exclusive means through which each trading agent interacts with the market institution. An interagent enforces its user (a trading agent) conversation protocols that establish what illocutions can be uttered by whom and when. Not only are interagents utilized to allow trading agents to interact with the market institution, but also to allow those agents working as market intermediaries to coordinate their activities. Figure 1 depicts a conceptual view of the Fishmarket system which differentiates trading interagents (attached to buyers and sellers) from market interagents (attached to market intermediaries). Both types of interagents communicate asynchronously through TCP-stream sockets with their users, making use of the communication services detailed in section 3.

For the current version of the system both trading and market interagents instantiate the conversation protocols stored in their libraries of conversation protocols. Nonetheless we must recall from section 4.1 that the high flexibility of interagents would permit to dynamically reconfigure Fishmarket without changing the implementation. For instance, say that the boss of the market decides, during the system’s run-time, that the auctioneer employs new bidding protocols unknown by the trading interagents. In that case, the boss would dynamically define the conversation protocol required for the new bidding protocol in the trading interagents so that buyer agents were capable of bidding under the new auction rules.

6 Conclusions

An interagent provides an agent with the basic mechanisms to interact (communicate and coordinate) with other members of an agent society. Therefore, an interagent intermediates between its user and the multi-agent environment as a whole. In this way, the overload related to the management of the communication and coordination tasks needed by an agent to live in a multi-agent system is shifted to its interagent, that relieves its user from such a tedious work. Moreover, and more importantly, interagents allow to coordinate interdependencies between agents’ activities by means of highly flexible conversation protocols.

Two major benefits are gained from employing interagents. On the one hand, it permits agents to reason about both communication and coordination at a higher level of abstraction, whereas on the other hand it provides a complete set of facilities that allows agent engineers to concentrate on the design of their agents’ inner and social behaviour.

JIM is currently being used in two directions:

- to promote the knowledge representation language Noos to an agent-oriented language[12];

- to coordinate the activities of the market intermediaries composing the Fishmarket system[22, 20] and the interaction between the market as a whole and the participating buyers and sellers (see Figure 1).

Acknowledgments

This work has been supported by the Spanish CYCIT project SMASH, TIC96-1038-C04001; Juan Antonio Rodríguez and Francisco J. Martín enjoy DGR-CIRIT doctoral scholarships FI-PG/96-8490 and FI-PG/96-8472 respectively, and Jordi Sabater enjoys a CSIC scholarship “Formación y Especialización en Líneas de Investigación de Interés Industrial” in collaboration with C.S.M (Consorci Santarri de Mataró).
References


Sistemes Multi Agents

JAFDIS, a Java Framework for Dialogical Institution Specification*

T. Alsinet, R. Béjar, C. Ansótegui, C. Fernàndez, F. Manyà
Departament d'Informàtica i Enginyeria Industrial
Universitat de Lleida,
Jaume II 69, 25001 Lleida, Spain
{tracy,ramon,carlos,cesar,felip}@eup.udl.es

Abstract

In this paper we give a detailed description of JAFDIS: a user-friendly Java framework we have implemented to specify dialogical institutions and we illustrate the use of JAFDIS by showing how to formalize a protocol of a scene of a traditional fish market.

1 Introduction

Recently, a lot of work has been done in the area of multi-agent systems. From applications such as electronic commerce [6] or e-mail filtering [3] – where agents interact autonomously, through the definition of generic models of negotiation between agents [2] – to the description of general approaches to build agent architectures [5].

The purpose of this paper is to describe our ongoing research in the project SMASH (Systems of Multi-agents for Medical Services in Hospitals). The main contribution is the design of the graphical framework JAFDIS, implemented in Java, for assisting users in the specification of dialogical institutions making easy to include in it all the possible scenes, the performing agents and the interactions between them.

The remainder of the paper is organized as follows. Section 2 summarizes the concept of dialogical institution. Section 3 describes how a dialogical institution can be modeled using JAFDIS and Section 4 illustrates the utilization of JAFDIS by formalizing the protocol of an existing dialogical institution, the Fishmarket[4].

2 Dialogical Institution

Following the approach described in [4] we can think of agent-based institutions as a computational realization of a traditional institution, which intuitively amounts to a set of clearly established conventions that somehow restrict participating agent’s interactions. Therefore, a dialogical institution can be understood as a set of roles or kinds of agents and a set of protocols so that, for every protocol, there exists a well defined behaviour of the involved roles. An agent is an entity that plays a specific institution role. Moreover, an agent can play different roles in different protocols and in different scenes of the same protocol.

In a dialogical institution agents are “dialogical entities” that interact with other agents in a multi-agent context through illocutions. An agent is an entity that expresses illocutions, reacts to illocutions addressed to it and, furthermore, only the emission and reception of illocutions constitute its observable behaviour.

For a detailed description of a dialogical institution and its components see [1].

3 Modelling with JAFDIS

JAFDIS offers the possibility of working either in graphical mode or in text mode. Thus, protocol dependence graphs can be specified either by loading a textual description written in a specific formal language or by drawing the graph using a graph editor. Both modes are totally equivalent: the changes you produce in one mode are reflected in the other mode. Change between modes is also allowed during execution. The textual description is supported by an editor with the basic editing
options. The graphical mode is supported by a full-featured graphical user-interface which incorporates technology MDI (Multiple Document Interface) in order to display the context of all the scenes that configure a protocol in a common frame.

As a general rule, we must say that the definition of a new dialogical institution can be performed in two ways that could be characterized as static and dynamic way. The static way refers to the possibility of defining all the elements by means of menu commands before drawing the related protocol, scene, etc. The dynamic way embodies our natural conduct when creating a protocol; i.e., we usually define the elements involved as we need them rather than establish a previous and closed definition.

3.1 Creating a Dialogical Institution

When JAFDIS is started the presentation screen is prompted with a menu from where a new dialogical institution or a saved one can be activated. Once a Dialogical Institution has been created or opened, its graphical representation, the Framework, is introduced. The Framework has been divided into four parts (Figure 1): A menu and a toolbar, a tab-Pane area, the WorkSpace and a Debugger area.

3.2 Dialogical FrameWork

Agents which are role occurrences are accessible from the menu, and the sequence of the dialog boxes related to them is the following. First of all, the “Agents List Dialog” is displayed (Figure 2). It shows an enumeration of the agents that have been defined and enables one to modify the list of agents or edit the characteristics of a given agent.

One of the properties that has an agent is the role to which it belongs; this property can be edited with the help of the “Role property Dialog” (Figure 3) and two lists, symbolizing the roles to which the current agent belongs and the roles to which it does not belong, are displayed.

Roles and Communication Language have a sequence of dialog boxes similar to the one of agents. Their first dialog boxes are “Role List Dialog” and “CL List Dialog”, respectively. For example, Figure 4 shows how an illocation is defined. The rest of the elements of the Dialogical Institution (L, Loc, TL, ML, T) (see [1] for a description) are generated dynamically during the generation of the protocol dependence graph.

3.3 Performative Structure

When creating a new dialogical institution, a new initial scene is created by default.

Every scene has a dual graphical representation.
3.4 Consistency checking

Every Protocol Dependence Graph must satisfy the following rules: the protocol dependence graph must be connected, must have at least one initial and one final node, and every node must be accessible and useful. A node is accessible when there is a path from the initial node to this node, and a node is useful when there is a path from this node to a final node. A depth-first-search algorithm is used to check if the graph satisfies such requirements. This algorithm marks the nodes just visited in order to avoid cycles and to optimize the algorithm. There is also a complementary test that checks if the constraints of every node are correctly satisfied.

4 An Example

In order to illustrate the utilization of JAFDIS, we describe the formalization of the protocol of a simple scene that belongs to the performative structure of the Fishmarket dialogical institution introduced in [4]. Nowadays, we are working on the application of the JAFDIS system to the definition of a medical protocol in the context of the Mataró Health Consortium.

The scene chosen is the one corresponding to the registration room (RR) scene of the Fishmarket. In this scene, sellers (s) enter to try to register a new lot of goods to be auctioned. The seller admitter (sa) is the responsible of the acceptance of the lot of goods, and he can accept or deny the request of a seller. If a new lot is accepted, the seller admitter updates the catalogue of lots (AG) for the auction and the seller is allowed to enter to the auction hall (AH), or the back office (BO).

An additional event can occur in this scene. This event corresponds to the entrance of the auctioneer (auct) in order to ask to the seller admitter about the availability of new goods to be auctioned, that occurs whenever no more goods are available in the auction room.

4.1 The Protocol Dependence Graph

We give here only the protocol in which the scene is splitted, but not all the protocols associated with all the sub-scenes that appear inside the protocol. We can see a view of the protocol dependence graph in Figure 6, where some of the illocutions contents, transition constructs and sub-scene names appear.
with a name shorter than the one we use in the following description of the protocol due to space limitations.

When the scene starts, we are in the initial state $A_1$ and only the seller admittance in the scene. Any seller who wants to enter to the scene must do it through the synchronism node $S_1$ using the transition construct $move(s, 1, market, S_1, t)$. Whenever a seller makes a request to register a new lot ($request(s, sa, register(lot), t)$) we enter to state $A_1$. We leave this state only when the seller receives one of the two possible answers ($accept(sa, s, registration(aneulot))$ or $deny(sa, s, registration(reason))$) from the seller admittance and we go back to the state $A_4$ to allow the entrance of new sellers. If the seller of an accepted lot wants to go to the scenes of the auction hall or the back office he does it through the parallel nodes $P_1$ and $P_2$, respectively, using the transition constructs $move(s, 1, P_1, AH, t)$ and $move(s, 1, P_2, BO, t)$.

Whenever the auctioneer wants to enter to the scene, to request more goods to be auctioned, he does it through the synchronism node $S_2$ using the transition construct $move(auct, 1, AH, S_2, t)$. When he makes the request ($request(auct, sa, moregoods, t)$) we go to state $A_2$ and wait the response of the seller admittance. If more goods are available, he answers with the reference to a new lot of goods ($declare(sa, auct, neutolot(AG), t)$) and we come back to the state $A_1$ and, at this point, the auctioneer can return to the auction hall, to continue the auction, through the parallel node $P_3$ using the transition construct $move(auct, 1, P_3, AH, t)$. In case no more goods are available ($declare(sa, auct, nomoregoods, t)$), we go to state $A_3$. From this state, we have two possible ways of closing the scene. With the first one the seller admittance forces a movement of all the agents of the scene to the back office through the parallel node $P_3$, using the transition construct $move((sa, s, auct), (1, all(s), 1), P_3, BO, t)$, and then the registration room scene reaches the final state $AF$. The other way of closing starts with the movement of the seller admittance to the boss office scene through the parallel node $P_4$ using the transition construct $move(sa, 1, P_4, BOFF, t)$, in order to bring the boss to the registration room, and the movement of the auctioneer to the auction hall, to initiate the closing of the auction, using the transition construct $move(auct, 1, P_4, AH, t)$. Once the seller admittance finishes successfully the boss office scene, the seller admittance and the boss enter into the registration room through the the synchronism node $S_3$ using the transition construct $move((sa, boss), (1, 1), BOFF, S_3, t)$. Then, from the state $A_4$ the seller admittance sends the illocution $declare(sa, boss, closeroom(RR), t)$ and we go to the state $A_5$. In response to this request, the boss closes the room and sends the illocution $declare(boss, sa, closed)$ and we go to the state $A_6$. Finally, once the scene is inactive the seller admittance sends a final illocution $declare(sa, boss, inactive, t)$ and we reach the final state $AF$.

4.2 The Behaviour Rules

The behaviour rules represent a set of restrictions over some of the transitions of the protocol dependence graph. The restrictions are given as a set of rules that say that the emission of some illocutions or transition constructs are not allowed when some conditions are not satisfied.

Only to give an idea of the intended utility of the behaviour rules, we give the behaviour rule associated with the movement constructs that allow a seller to go from the Registration Room scene to the Back Office or the Auction Hall. The seller can not emit the transition constructs $move(s, 1, P_1, AH, t)$ and $move(s, 1, P_2, BO, t)$ if he has not previously made a request to register a lot of goods ($request(s, sa, register(lot), t)$) and next has received a positive answer to the request ($accept(sa, s, registration(aneulot))$). We can formalize the previous behaviour rule as follows:

$$\begin{align*}
\text{if} \quad & (\neg \text{request}(s, sa, \text{register}(\text{lot}), t) \lor \\
& \quad \text{deny}(sa, s, \text{registration}(\text{reason}), t; t + \Delta t)) \\
\text{then} \quad & \neg (\text{move}(s, 1, P_1, BO, t + \Delta t) \lor \\
& \quad \text{move}(s, 1, P_2, AH, t + \Delta t + \Delta t'))
\end{align*}$$

5 Future work

Mainly, two basic lines of work will raise. The first one will be the implementation of a protocol compliance monitoring system. The second one, will be the development of software units for intelligent agent behaviour modelling, basically in two aspects. One of them will consider the agent as a mere interface with the human behaviour, applicable when the human intervention was necessary. The other, when possible, will consider the agent as an autonomous entity.
Figure 6: Protocol dependence graph of the registration room scene

References


On the Usefulness of Component-Based Architectures for Knowledge Discovery

Jesús Cerquides, Ramon López de Mántaras
Artificial Intelligence Research Institute, IIIA
Spanish Council for Scientific Research, CSIC
08193, Bellaterra, Barcelona, Spain
{cerquide,mantaras}@iiia.csic.es

Abstract

Component Software is a relatively new software technology that relies on the idea of independent extensibility. In this paper we present some recently developed concepts in component software. After that, we argue that for Knowledge Discovery Support Environments independent extensibility is one of the most important qualities. Component approaches to software development, and more concretely a component framework to rely on, would be extremely useful for the implementation of an extensible Knowledge Discovery Support Environment. We explain what are the benefits of disposing of a component framework for Knowledge Discovery. Finally, we sketch a preliminary design of KDCOM, our Knowledge Discovery Component Framework.

1 Component Software

Component software is the field of knowledge devoted to the study of the design of software systems that are structured in components.

The idea of structuring solutions in components has been followed successfully in fields such as electrical engineering and mechanics. Component software is an attempt to mimic those approaches in the software industry. This section will explain the main ideas behind the current approach to component software.

In [4], the idea of independent extensibility is introduced. “We call a system independently extensible, if it can cope with the late addition of extensions without requiring a global integrity check”. Systems designed with independent extensibility in mind can change the software market. They allow the possibility of buying small specialized components instead of huge applications. The final application will be built by composing all those components together.

In [6] a definition for component is introduced:

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.

This definition contains some technical details that should themselves be explained.

Software components are units of composition. Composition is the process by which we construct a new entity using previously existent elements. Software components have to be designed with this composition process in mind.

“Interfaces are the means by which components connect” [5]. The previously given definition of component states that interfaces should be contractually specified. A contract binds a client and a provider. It specifies the conditions that both of them have to fulfill in order to have a successful cooperation. The benefits of this idea are presented clearly in [8]. When compiling a new component, we rely only on the specifications of the contracts it subscribes with the rest of components. At run time, the composition process assigns components that fulfill the contract.
Each component offers an interface that allows the rest of components to make use of its services. Non-trivial components will surely also request a set of services themselves as, for example, the file system. The definition of component states that those services, known as context dependencies or required interfaces, should be explicit.

1.1 Component frameworks

Component frameworks are defined in [7] as “The set of rules to be obeyed by components in a certain environment”. A component framework contains the rules to be followed when designing and implementing a new component. A component framework should provide rules in aspects as how should components communicate or interact, that is, it should establish the patterns of cooperation and coordination for components. The framework itself does not implement the functionality, it only defines the battlefield where components will play, the rules of the game for components.

1.2 Dimensions of extension

If we want a system to be totally extensible we cannot fix any infrastructure. If there is no fixed infrastructure, it would be impossible for components to cooperate. An independently extensible system can only be extended in the directions that have been previously planned to be extensible. It should be clearly specified what parts of the system can be extended. Those features of a system designed to be extensible are named dimension of extension [7]. A key problem when designing a component framework is to classify possible extensions, that is, to find the dimensions of extension of the system.

2 Component frameworks and Knowledge Discovery Support Environments

The idea that extensibility is important for a Knowledge Discovery Support Environment (KDSE) is introduced in [9]. A KDSE should contain different data mining tools and should provide support for adding new ones because they are being developed constantly. In [9] the idea of plug-in is proposed as the solution to reach extensibility. The plug-ins are Data Mining Methods, that communicate with the system kernel, that deals with data management, task management and result management and visualization. The idea of plug-in is a step in the good direction; it does not take profit of all the work done in the Component Oriented Programming field but can be seen as a first attempt to develop a component framework for KDSE.

2.1 Benefits of a Knowledge Discovery Component Framework

A well designed component framework for Knowledge Discovery (KD) could provide different benefits which include:

- An increase in KDSE modularity by enforcing the different elements in the KDSE to fulfill the interfaces defined by the framework.
- An increase in reusability by allowing design reuse. The framework is the design of a KDSE, thus generating a new KDSE having the framework would be a lot easier.
- It would ease distributed execution, on the component level, because communications between components would be precisely defined.
- Obviously, it would ease extensibility, as this is its main purpose. By establishing communication interfaces and protocols, it is a lot easier to integrate a newly developed component, by making it fulfill the contract that the interface demands. It also eases the integration of legacy KD methods reducing it to the construction of a wrapper that calls the old KD method and fulfills the contract.

Most of this benefits have been presented as requirements for a KDSE in [1].

3 KDCOM: A Knowledge Discovery Component Framework

The architecture of Kepler [9] can be seen as a component framework that defines a unique dimension of extensibility. From our point of view, a
well designed component framework for KD should be extensible along more dimensions: visualization tools, data base connections and data mining process management environments. The only thing that the KDSE kernel should fix are the main interactions between KD tools.

In the design of KDCOM we have identified four dimensions of extension:

- **Data components.** They store some kind of data interesting to the application at hand. They offer a series of services to access to this data. Into this class of components we can find relational databases, rules, decision trees and any “passive” or storage component.

- **Processing components.** They have some data components as input and some data components as output. They do the main processing in our application. Into this class we find components for induction of decision trees, components for rule induction, components for discretization, components that classify an instance given a decision tree, and any component “active”, that is a component that we can run.

- **Visualization components.** They have some data components as input and normally no output other than visualization on a device of some characteristics of the input. Into this class we find components for plotting data, for showing decision trees, and any component that has visualization of information as its main purpose.

- **Process management components.** Some users will prefer a graphical process management environment while some others will prefer, for example, a textual one. As KD evolves into a more developed discipline, automated KD process management tools will be more common. A KD framework should provide extensibility along this dimension.

These dimensions of extension are not orthogonal. Data components are of main importance because all the rest of components rely upon them for the exchange of data.

In KDCOM, interfaces constitute a hierarchy starting in the most abstract and ending in the more concrete ones. Evolution of interfaces from one version of the framework to the next is made by creating new versions of the interfaces. Most of the times, new versions of an interface would be derived from the existent one. When developing a new component, the programmer has to decide the versions of each interface that the new component will support. That way interfaces have their own life cycle, being dropped by new versions of the components when they are considered out of date.

The basic interfaces have been defined by means of Java interfaces using the JavaBeans [2] component model or by means of C++ pure virtual classes. A first running version of a KDSE developed using the framework is actually being developed.

4 A Knowledge Discovery Schema

Most Knowledge Discovery tasks can be seen as an example of the schema shown in Figure 1. The schema shows the different roles that components may have in concrete tasks. Surely there are some roles missing, but the schema does not aim for completeness. Instead, it should be looked at as an example of component combination, a possibility offered by the framework. In the following, common roles for each component will be signaled.

![Knowledge Discovery component roles](image)

Figure 1: Knowledge Discovery component roles

5 Future work

In [3], a set of steps are defined for the creation of a framework. The first of them is to create three
applications in the domain area of the framework. After the first application is finished, the first version of the framework, should start growing from the generalization of pieces of code that are reused in the following applications. We are actually somewhere in this step, and we still have a lot of work to do until we get a well designed and operative Black Box Framework. From the future development of the framework we will get two benefits. First of all, we will develop a clear structure for the part of the KD process that the framework will cover. On the other hand, lots of issues are still unknown in the framework development process, so it is sure that many problems will arise and hopefully some of the solutions that we provide for them would have the possibility to be reused in the design and implementation of frameworks in other domains.

6 Summary

We have introduced some concepts of component software and the idea of component framework. We have signalled the benefits that a component framework would provide for KDSE development. We have briefly presented the four dimensions of extension that seem of main importance to us. We are now in the process of developing a first KDSE based in KDCOM.

7 Acknowledgements

This work has been partially supported by the Spanish CICYT project SMASH, TIC96-1038-C04001. Jesús Cerquides research is supported by a doctoral scholarship of the CIRIT (Generalitat de Catalunya).

References


Un modell lingüístic per a la descripció de conceptes

Enric Hernández
Secció de Matemàtiques i Informàtica
ETSAB, Avda. Diagonal 649, 08028 Barcelona
Universitat Politècnica de Catalunya
enriche@lsi.upc.es

Jordi Recasens
Secció de Matemàtiques i Informàtica
ETSAV. Passeig Serra 1-15, 08190 Sant Cugat
Universitat Politècnica de Catalunya
recasens@ea.upc.es

Resum
Cada cop és més patent la necessitat de desenvolupar metres i tècniques per extreure coneixement útil de grans bases de dades. En aquest treball presentem un algorisme capaç d’obtenir descripcions lingüístiques de conceptes definits contra una base de informació. Per reduir l’elevat cost computacional que implica conjecturar descripcions mitjançant una cerca exhaustiva, s’ha desenvolupat una tècnica basada en una ordenació particular dels atributs, que atorga simplicitat i eficiència al mètode presentat.

Paràmeters clau: concept learning, linguistic summaries, inductive fuzzy learning, rules acquisition, Data Mining, Fuzzy Logic.

1 Introducció
En els últims temps, els avanços tecnològics juntament a la seva major disponibleitat i implantació a la nostra societat, han provocat la transició de un model de societat a un altre ben diferent. En paraules de Pedrycz [5]:

“Vivim en una societat rica en informació però pobre en coneixement”.

Efectivament, la gran capacitat de “recol·lectar” informació ha originat la aparició d’ingents bancs de dades sobre els temes més diversos. Però de poc serveix tota aquesta informació si no va acompanyada de metres efectius per “navegar” a través d’ella. Aquesta consideració implica la distinció entre informació i coneixement; per aquest últim entenen una estructuració de la informació que ens permet una comprensió d’aquesta, així com un maneig efficient relatiu al problema que volen resoldre.

2 Formulació del problema
L’objectiu del mètode que presentem és, com ja s’ha dit abans, obtenir descripcions lingüístiques de conceptes.

Siguï $J = \{J_1, \ldots, J_n\}$ el conjunt de tots els atributs. Considerarem dos subconjunts de $J$: $D \subseteq J$ serà el conjunt d’atributs explicatius, o el que és el mateix, el conjunt d’atributs que utilitzarem per obtenir les descripcions dels conceptes; i $C \subseteq J$ el conjunt d’atributs a explicar, o el conjunt d’atributs a partir dels quals generarem els conceptes que volen descriure.
Exigirem que $D \cap C = \emptyset$, ja que sinó, la descripció óptima del concepte a explicar seria el mateix (ex: si volem explicar el concepte “Temperatura::Alta” a partir dels atributs Temperatura, Pressió i Volum, pretenem evitar la descripció tautològica “Temperatura::Alta si Temperatura::Alta”). $D \cap C$ no té que ser J, amb el que es permet ignorar certs atributs en el procés de generació de descripcions.

La nostra aproximació segueix la proposada per Yager a [8] en el sentit de particionar l’espai per simplificar les representacions, bastant-nos en l’ús de “templates” definits sobre els dominis dels atributs. Un “template” associat a un atribut és una col·lecció d’etiquetes lingüístiques que “cobreixen” el domini d’aquest atribut.

Concretament, per cada atribut $J_i \in J$ considerarem definir una “template” (variable lingüística) $V_i$ sobre el domini de $J_i$. A Zadeh [12] es proposa la definició d’una variable lingüística com una quíntupla $(V_i, T(V_i), U_i, G_i, M_i)$ a on:

- $V_i = \text{nom de la variable}$.
- $T(V_i) = \text{conjunt de termes àtoms de la variable (etiquetes lingüístiques)}$.
- $U_i = \text{univers de discurs}$.
- $G_i = \text{regles sintàctiques de construcció de termes vàlids}$.
- $M_i = \text{regles semanticàtiques per associar a cada terme Vàlid el seu “significat” (“fuzzy-set” definits sobre } U_i)$.

Així, per exemple, si tenim l’atribut “Anys” definit a $[0,100]$, podem associar-li la variable lingüística “Edat” definida per:

- $V_i = \text{Edat}$
- $T(V_i) = \{\text{molt jove, jove, madur, vell}\}$
- $U_i = [0,100]$  
- $G_i = \text{Molt jove i jove} \ni \text{madur i vell}$
- $M_i = \text{definicions de “Molt jove”, “jove”, “madur” i “vell” (“fuzzy-sets” definits en } [0,100])$

Les etiquetes d’aquestes variables ens serviràn per generar els concepts i les seves descripcions. Considerarem descripcions vàlides de conceptes els elements de l’Algebra de Boole generada pels termes de $U_{i \in C} T(V_i)$ (a on $T(V_i)$ són els termes vàlids de la variable lingüística $V_i$).

Les descripcions a obtenir seran cadenes produïdes per la mateixa gramàtica, amb la diferència de que $J, \in D$. És a dir, els conceptes i les descripcions seran conjuncions, disjuncions i negacions d’etiquetes dels “templates” associats als conjunts d’atributs C i D respectivament.

3 Algorisme.

Observant les definicions prèvies, el problema es reformula de la següent forma: donades dos definicions de concepts C’ i C" trobar la descripció lingüística que millor expliqui C’ i a la vegada excludi C" (CA” ha de ser entès com un conjunt de restriccions negatives imposades sobre la descripció a generar, és a dir, la representació intensional dels contraexemples). No s’exigeix que C" sigui el complementari de C’.

En aquesta secció presentarem un algorisme destinat a resoldre aquesta tasca. Aquest algorisme pot dividir-se en els passos següents:

1) Selecció de “candidates”; en aquesta primera fase, per tot atribut $A$ i per tota etiqueta lingüística $A_i$ del “template” associat a $A_i$, calculem “la proporció de C’ i C” que cauen dins $A_i$:

$$\delta(A = A_i, C') = \frac{\sum_{x \in X} \mu_{A_i}(x) \land \mu_{C'}(x)}{\sum_{x \in X} \mu_{C'}(x)}$$

$$\delta(A = A_i, C") = \frac{\sum_{x \in X} \mu_{A_i}(x) \land \mu_{C"}(x)}{\sum_{x \in X} \mu_{C"}(x)}$$

, a on X és el conjunt d’instàncies.

Aquests dos valors ens serviran per calcular la bondat de la descripció $A = A_i$ donats C’ i C". Abans, però, cal definir com evaluar la bondat d’una descripció D:

$$\sigma(D, C', C") = \frac{\sum_{x \in X} \mu_{D}(x)}{|X|} \frac{\sum_{x \in X} \mu_{D}(x) \land \mu_{C'}(x)}{\sum_{x \in X} \mu_{C'}(x)} \frac{\sum_{x \in X} \mu_{D}(x)}{|X|}$$

$$\sum_{x \in X} \mu_{D}(x) \land \mu_{C"}(x)\sum_{x \in X} \mu_{D}(x)$$

Per entendre aquesta expressió, fem la traducció al cas clàssic a on totes les funcions de pertinença ($\mu$) estan
valudes en \( \{0,1\} \). Llavors podem reescriure l’expressió, en termes probabilistes, com:

\[
\sigma(D,C^+,C^-) = p(C^+) \cdot (p(D | C^+) - p(D)) - p(C^-) \cdot (p(D | C^-) - p(D))
\]

Si la desglossem tenim que:

1. \( p(D | C^+) - p(D) \) : aquesta subexpressió és una mena de “test de dependència creixent” entre \( D \) i \( C^- \), és a dir, serà més positiu mentre més afavorit resulti el cumpliment de la hipòtesi condicionada a \( C^- \) respecte al cumpliment de la hipòtesi “a seques”.
   - No fem el test de dependència estàndard \((p(D | C^+) \neq p(D))\) perquè podria passar.
   - \( p(D | C^+) < p(D) \) és a dir, que el condicionar a \( C^- \) no només no afavorixi sinó que desfavorixi el cumpliment de \( D \), lo qual vol dir que no ens interessarà obtenir la descripció \( D \).

2. \( p(C^-) \cdot (p(D | C^-) - p(D)) \) : aquest “test de dependència creixent” el “ponderem” per \( p(C^-) \). Això ho fem perquè volen que el suport del condicionant sigui “significatiu”.

3. \( p(C^-) \cdot (p(D | C^-) - p(D)) \) : aquest terme té la mateixa interpretació que el terme anterior, només que el condicionant és \( C^- \) (concepte exclòs). Fixem-nos que resta, ja que com hem dit abans volen que la nostra descripció exclogui \( C^- \), per tant mentre \( C^- \) més afavorixi el cumpliment de \( D \), més desitjable serà la tríia de \( D \) com a descripció.

Tornant a l’explicació del algorisme, aquesta mesura \( \sigma \) ens serveix per calcular per tot atribut \( A \) i per tota etiqueta \( A_i \) de \( A \), el valor \( \sigma(A = A_i, C^+, C^-) \).

Segueixim, per cada atribut \( A \) reunim aquests valors en un vector, a on la cel.la i-éssima conté \( \sigma(A = A_i, C^+, C^-) \).

A part d’aquests vectors triem les etiquetes candidates per cada atribut. Hi han diverses formes per fer-ho:

- Aplicant un “\( \alpha \)-cut” sobre el vector, amb un llindar pre-establert.
- Triar les n cel·les (etiquetes) amb un valor \( \sigma \) més gran (té l’avantatge de que fixem el nombre de candidates per atribut).
- Les “significativament” més grans (test de significància).

2) Càlcul del grau de discriminació dels atributs: la informació continguda en el vector calculat per un atribut \( A \) és el grau de bondat corresponent a considerar cadascuna de les etiquetes associades a \( A \), per si sola, com a descripció (donat \( C^+ \) i \( C^- \)). Així, si obtenim moltes etiquetes candidates per un atribut \( A \), vol dir que les instàncies de \( C^+ \) i \( C^- \) estaran distribuïdes uniformment al ser projectades sobre el domini de \( A \), mentre que si el nombre de candidats es petita, les instàncies estaran més “localitzades”. Ditz en altres paraules, mentre més candidates tenim per un atribut, menys discriminant serà aquest atribut respecte \( C^+ \) i \( C^- \).

Definim el grau de discriminació (\( g(A) \)) d’un atribut com:

\[
g(A) = \frac{\# \text{total_etiq_de_A} \# \text{etiq_candidates_de_A}}{\# \text{total_etiq_de_A} - 1}.
\]

Aquest grau, juntament amb els valors \( \sigma \), ens permet ordenar els atributs en una llista per ordre de discriminació.

3) Refinament d’hipòtesis: ara ja està tot a punt per anar construint la descripció. Prenem com descripcions inicials les corresponents a les etiquetes candidates del primer atribut de la llista, anirém agafant la resta d’atributs per ordre, i comprovariem si la incorporació d’aquestes a la descripció actual la millorra o no (estudi del grau de discriminació d’aquest atribut condicionat a la descripció actual). El fet de que aquesta comprovació la fem només per les etiquetes candidates, fa que el procés de refinament d’hipòtesis sigui força eficient. Així:

- Primer considerem com a descripcions les etiquetes candidates pel primer atribut (\( A \)).
- Per l’atribut següent (\( B \)) calcularem els valors:

\[
\sigma(B = B_i \land D, C^+, C^-) = \sigma(B = B_i \land A = A_i, C^+, C^-)
\]

, tal que:

\[
A_i \in \text{Candidates}(A),
B_i \in \text{Etiquetes}(B).
\]

Aquests valors \( \sigma \) ens servirán per calcular \( g(B | A = A_i) \), que és el grau de discriminació de l’atribut \( B \) condicionat a les instàncies tals que l’attribut \( A = A_i \).

Si \( g(B | A = A_i) \) segueix essent discriminant, refinem la descripció \( D = A_i \) a \( D = A_i \land B_k \), tal que

\[
B_k \in (\text{Candidates}(B) \cap \text{Candidates}(B | A = A_i)),
\]

i sinó tenim dues opcions:

- Ens quedem amb \( D = A_i \) i passem al següent atribut de la llista.
• Augumentem la descripció a \( D = A_v \cup B_v \) (\( B_v \in \text{Candidates}(B) \)) i passem al següent atribut de la llista.

- Reiterem el procés amb el següent atribut fins arribar a atributs "no discriminants".

4 Conclusions i treball futur

Aquest treball s’emmarca dins els mètodes per extreure regles descriptives d’un conjunt d’instancies. En el disseny de l’algorisme s’ha posat èmfasi especial en intentar resoldre els que creiem són els punts febles d’altres aproximacions (veure secció de revisió bibliogràfica):

- Manca d’automatització en el procés de conjectura de descripcions: creiem que ha de ser el mateix sistema qui sigui capaç de "localitzar" el procés en les descripcions que "semblin millors". Això no estem dient que cal prosseguir absolutament de la figura de l’expert, ja que en última instància serà aquest l’encarregat de validar els patrons descoberts. Només critiquem la intervenció que aquest té en altres sistemes.

- Costos prohibitius en el procés de cerca de descripcions: mitjançant l’ús d’una llista ordenada d’atributs i el càlcul de les etiquetes candidates per cada atribut, reduïm el cost de la cerca i evitem la validació de descripcions que no semblin interessants. El desavantatge de fer servir un mètode heurístic és que les descripcions trobades no tenen perquè ser les descripcions òptimes. Cal assolir un equilibri entre "exhaustivitat" de la cerca i "optimalitat" de les descripcions.

Volem que aquest treball sigui el punt de partida d’un estudi més profund sobre com obtenir descripcions lingüístiques de les dades. Naturalment, resten molts aspectes oberts i subjectes a discussió, com:

- "Ajustament" automàtic de les particions "fuzzy" associades als atributs: de vegades pot succeir que la descripció d’un concepte no sigui "gaire bona" no degut a que el núvol d’instancies del concepte (representat a l’espai dels atributs) no estigui "ben distribuït", sinó perquè no "encaixi" bé amb els "templates" definides a priori. Seria desitjable que el sistema fos capaç de modificar aquests "templates" per tal d’adaptar-les millor a les dades.

- Reducció de les descripcions obtingudes: sovint, les descripcions obtingudes poden ser massa complexes.

Caldrà un "post-procés" per simplificar-les i filtrar els atributs redundants.

- Provar noves mesures per avaluar la bondat de les descripcions, i comparar-les entre si.

Referències

REPRESENTACIÓN, FUSIÓN E INTERPRETACIÓN
DE ATRIBUTOS CONTÉCNICAS FUZZY

David Nettleton
Departamento de Lenguajes y Sistemas Informáticos.
Universidad Politécnica de Catalunya
nettleton@lsi.upc.es

Resumen

Para datos mixtos de tipo difuso y tipo crisp es necesaria una representación homogénea y una forma adecuada de proceso. En este artículo, dos enfoques contrastados aportan dos visiones de los mismos datos: el primero representa los datos en una forma difusa homogénea y los procesa de forma crisp; el segundo enfoque representa los datos de forma crisp y los procesa con un clasificador difuso. Se utiliza el método para procesar los datos de prueba de Hathaway y Bezdek, junto con el proceso de datos reales de 'admisiones hospitalarias'.

Palabras Clave: Representación difusa y crisp, fusión de atributos, par parmenideano, covarianzas difusas.

1. INTRODUCCIÓN

Se presentan dos enfoques contrastados que permiten tratar los aspectos difusos de datos: (i) Representación crisp y formación difusa de clusters; (ii) Representación difusa y formación crisp de clusters. Se presenta una extensión de la representación difusa de Hathaway [4] para incluir los pares parmenideanos, y una variación del algoritmo de Gustafson [2] que permite calcular covarianzas entre variables. Estas extensiones permiten procesar diferentes tipos de datos en una forma homogénea y aprovechar la información de las características difusas.

2. REPRESENTACIÓN DIFUSA Y PROCESO CRISP

3.1 EL CALCULO DE CLUSTERS, GRADOS DE PERTENENCIA Y COVARIANZAS DIFUSAS

Definimos: \( u_i \) es la matriz de grados de pertenencia de los \( n \) casos a cluster \( i \); \( u_{ik} \) es el grado de pertenencia del caso \( k \) a cluster \( i \); \( x_k \) es el vector de características (datos) del caso \( k \); \( v_i \) es el centroide del cluster \( i \); \( m \) es el 'grado de fuzzificación'; \( T \) es la transposición de la matriz (resultante).

El algoritmo de Gustafson [2] mide el grado de relación entre una variable \( V_1 \) y el centroide de un cluster \( C_i \). También podemos medir el grado de relación de otra variable \( V_2 \) al centroide del mismo cluster \( C_i \). Así, la distancia es la diferencia entre el grado de relación de \( V_1 \) a \( C_i \) y el grado de relación de \( V_2 \) a \( C_i \), es decir \( d(V_1, C_i) - d(V_2, C_i) \).

Por consiguiente, el cálculo de las covarianzas difusas entre variables en cada cluster \( i \) se calcula con la siguiente fórmula:

\[
C_{fi} = \sum_{j=1}^{n} \Sigma_{k=1}^{q} \sum_{j=1}^{n} u_{ik}^m \left| d(V_1, C_i) - d(V_2, C_i) \right|
\]

(1)

donde

\[
d(V_1, C_i) = (x_{ik} - v_i) (x_{ik} - v_i)
\]

y

\[
d(V_2, C_i) = (x_{ik} - v_i) (x_{ik} - v_i)
\]

\( \rho \) siendo el número de variables (dimensiones) y \( n \) el número de casos.

3.2. LA FUSIÓN DE ATRIBUTOS

Se ha implementado el algoritmo de agregación de Hartigan [3] para realizar las sucesivas fusiones de atributos a partir de su matriz de covarianzas. Grosso modo: en cada paso, se fusiona el par de atributos con las covarianzas más altas para formar nuevos atributos, hasta obtener el número de atributos deseado, o hasta concluir un árbol binario con las agrupaciones realizadas.
4. DATOS DE INPUT Y RESULTADOS DE PROCESO 'DIFUSO'

Inicialmente se utilizaron los datos de prueba [4] para verificar y afinar el algoritmo, antes de procesar los datos de 'admissiones'. También se han generado covarianzas crisp con SPSS y comparado su fusión con la fusión de las covarianzas difusas.

4.1 CONJUNTO DE DATOS: 'ADMISSIONES HOSPITALARIAS'

El conjunto de datos de prueba consiste en 17 variables que perfilan el paciente (ver [5]). Algunas de las variables son: (7) Coma o estupor profundo a 24 horas (sí, no); (9) Fallo Cardiovascular (sí, no); (10) Fallo Renal Agudo (sí, no); (17) Estado vital a la salida de la UCI (muerto, vivo).

4.2 RESULTADOS DE FUSIÓN

Tabla 1. Matriz de Covarianzas Difusas para algunas variables de 'Admissiones'.

<table>
<thead>
<tr>
<th>Fallo renal agudo</th>
<th>Fallo cardiano</th>
<th>Probabilidad de Infección</th>
<th>Estado Vital</th>
<th>Coma 24hr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fallo r. a. 1*</td>
<td>14.867</td>
<td>1.432</td>
<td>1.286</td>
<td>1.80</td>
</tr>
<tr>
<td>Fallo card.</td>
<td>1.432</td>
<td>12.531</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>Prob. Inf.</td>
<td>1.286</td>
<td>1.289</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>Estado VIt.</td>
<td>1.289</td>
<td>1.286</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>Coma 24hr</td>
<td>1.807</td>
<td>1.800</td>
<td>0.799</td>
<td>0.751</td>
</tr>
</tbody>
</table>

* A la diagonal se asigna valor 1 y no se utiliza en el proceso de fusión.

Fusión con Covarianzas Difusas

Las covarianzas 'difusas' de las variables de input (Sección 4.1) fueron calculadas usando el algoritmo modificado de Sección 3.1. La matriz resultante (ver Tabla 1) fue presentada como input al algoritmo de fusión de Hartigan.

Los cuatro factores finales fueron:

Factor 1: 'Fallo Neurológico' + 'Coma 24 horas' + 'Fallo Renal'.
Factor 2: 'Sexo' + 'Nivel FIO2' + 'Probable Infección' + 'Estado Vital'.
Factor 3: 'Urea Sanguínea' + 'Tipo Paciente' + 'Nº de Sistemas de Órganos Fallando' + 'Estancia 24 horas' + 'Incremento Creatinina' + 'Estado Anterior de Salud'.
Fulls per anotar
Fulls per anotar
Knowledge Fulls per anotar
Generació automàtica de regles a partir de la caracterització de classes

Karina Gibert
karina@eio.upc.es
Dep. d’Estadística i Investigació Operativa
Universitat Politècnica de Catalunya

Ulises Cortés
ia@lsi.upc.es
Dep. de Llenguatges i Sistemes Informàtics

“L’explicació científica no consisteix en el pas de la complexitat a la simplicitat, sinó en la substitució d’una complexitat menys intelligible per una de més intelligible”.

LÉVI-STRAUSS

1 Introducció

La formació i distinció de diferents classes d’objectes és un dels mètODES d’organització que des de molt i molt antic l’èsser humà utilitza en el seu procés de comprensió del món. Actualment, existeixen tècniques automàtiques de classificació que tenen per finalitat agrupar els objectes d’un conjunt donat en classes. Però és importantíssim que aquests mètodes proporcionin també mecanismes que li permetin a un humà d’entendre quin ha estat el criteri amb què s’han format aquelles classes i quina entitat semàntica tenen. De fet, una de les principals crítiques de Michalski [15] envers els mètodes estadístics de classificació automàtica és justament la seva falta de cacapacitat explicativa, o, el que és el mateix, les seves mancances per generar interpretacions de les classes en termes semàntics.

La nostra recerca ha estat bàsicament centrada en els mètodes automàtics de classificació i, entre d’altres coses, hem desenvolupat un sistema que s’anomena KLAISs i que permet utilitzar el coneixe-ment de l’expert tot combinant-lo amb mètodes estadístics de classificació automàtica per a trobar les classes d’un conjunt d’individus. KLAISs, de fet, està evolucionant molt ràpidament cap a un sistema orientat al Knowledge Discovery i Data Mining, àrees del coneixement molt recents i on la capacitat d’interpretar de forma automàtica els resultats d’una anàlisi té quasí tanta importància com la pròpia anàlisi. D’altra banda una de les millors formes de justificar una classificació és la seva facilitat per ser interpretada per un humà o una màquina.

L’objecte d’aquest article, doncs, no és pas descriure la metodologia que s’implementa en KLAISs, sino focalitzar precisament aquesta darrera part d’interpretació automàtica de classificacions, la qual obre les portes a la generació automàtica de regles, que materialitza un altre dels objectius del Knowledge Discovery.

La proposta que es presenta es basa en trobar primer una caracterització de les classes formades. S’introduceix prèviament una descripció conceptual per les classes, a partir de la qual es construirà la caracterització conceptual d’una partició.

Per tal de facilitar el seguiment de l’article, es presenta breument una aplicació relacionada amb la diagnosi de pacients que pateixen disfuncions de la glàndula tiroides, i una segona sobre unes dades que han estat classificades anteriorment per diversos autors. Al llarg del text, ens ajudarem de les classificacions presentades per a així il·lustrant els conceptes introduïts.

Malgrat que el procés de caracterització que es descriu està desenvolupat per variables numèriques i categòriques paral·lelament, en aquest treball es farà èmfasi sobretot en el cas de les variables qualitatives.

2 Presentació de dos casos

2.1 Disfuncions de la tiroides

D’entre totes les aplicacions veus que s’han dut a terme, s’ha seleccionat una aplicació a la medicina. Les dades provenen del Hospital Clinic Setre Milosvrdice, Zagreb (Croatia) i la recollida va durar dos anys.

Es tracta d’una mostra de 1001 pacients, els quals
estan descrits per tres variables que indiquen els resultats d’una prova de laboratori —nivell de les hormones T3, T4 y TSH, que per totes tres pot ser baix (B), normal (N) o alt (A)— i 3 factors rellevants pel diagnòstic —Edat, Gènere i Teràpia—. Les dades venen directament categoritzades de l’hospital i la taula 1 mostra la classificació obtinguda amb el nostre mètode, que s’anomena Classificació basada en regles i és detallada a [4].

2.2 Microordinadors

A [15] apareix un conjunt de dades referents a microordinadors que ha estat estudiat també a [10] i finalment s’ha analitzat utilitzant KLASS. Si bé el conjunt de dades és reduït i fa referència al que podriem anomenar arqueologia de computadors, donada l’antiguitat dels models que es descriuen, resulta un conjunt interessant, fonamentalment per constituir un joc de dades ben estudiat a la literatura, i per tenir una gran dària reduïda, la qual cosa permetrà de fer un seguiment manual dels procediments que es descriurien tot seguit.

La matriu de dades fa referència a 12 microordinadors americans descrits amb 5 variables, tres de les quals són qualitatives: Display (tipus de pantalla), MP (tipus de microprocessador), Keys (tipus de teclat) (veure la matriu a la taula 2).


3 Representació conceptual de les classes

Abans de descriure el procediment que es proposa per caracteritzar les classes és necessari indicar el suport sobre el que aquest es desenvolupa, que és una descripció conceptual de les classes. A partir d’un prototipus de cada classe tindrem informació suficient per fer la caracterització posterior. Els autors són conscients de la controvèrsia associada a la representativitat dels elements de tipus prototipus. No obstant això, es veurà desseguida com a la representació que es proposa correspon a un objecte fictici que conserva bona part de la informació continguda a la classe.

L’explicitació d’un element que representa cada una de les classes formades permet proporcionar-ne una descripció conceptual (o prototípica) i fer la identificació de la classe amb els prototipus. Aquesta via obre les portes de la generació automàtica de regles de classificació per a un sistema basat en el coneixement, a més de dotar de certa capacitat expressiva al sistema, que facilitarà la interpretació de les classes resultants (veure §4).

Partint d’un conjunt d’indivíduus \( I = \{i_1 \ldots i_n\} \), la idea de prototipus \( \hat{\mathcal{C}} \) d’un subconjunt (o classe) \( \mathcal{C} = \{i_1 \ldots i_m\} \subset I \) porta a pensar en un individu, real o no, que sintetitzi les característiques dels seus representats.

Considereu la descripció d’\( \mathcal{C} \), \( \hat{x}_C = (\hat{x}_{C1} \ldots \hat{x}_{CK}) \). El càlcul d’\( \hat{x}_C \) es fa component a component, i cal estudiar per separat el cas de les components numèriques i el de les categòriques. A [6] es presenta una proposta per ambdós, però, com ja s’ha dit, en aquest article ens concentrem en les variables qualitatives.

3.1 El representant de classe

Pel que fa a les components numèriques del vector \( \hat{x}_C \), el concepte de prototipus suggereix ràpidament la idea d’individu intermedi i, per tant, descripcions de tendència central com ara la mitjana o la mediana. Pel que fa a les variables categòriques, la manera més natural de sintetitzar un grup d’indivíduus en aquest cas és detallant la funció com aquesta és distribuïxen en les diferents modalitats de la variable, bé sigui mitja quan proporcions\(^2\), o percentatges, o contingents absoluts. Utilitzar les proporcions sembla el més adequat, ja que permet fer la següent interpretació.

El representant d’un conjunt d’indivíduus en un espai de variables qualitatives és una entitat fictícia que no té una modalitat definita per a cada variable, sinó que es reparteix en les diferents categories de cada una de forma proporcional a com ha fan els objectes que el generen.

Per exemple, l’agregació de dues comunitats d’igual grandària, una catòlica i una musulmana,

\(^{1}\)Es tracta del Dr. Miquel Barceló, del Departament de LSI de la UPC, a qui, per descomptat, agraïm la seva col·laboració.

\(^{2}\)Alguns autors utilitzen la mètode com representant d’una classe però, pels nostres propòssits, aquest descriptor sintetitza massa informació de la classe.

\(^{3}\)En termes més tècnics, quan es parla de proporcions s’està fent referència, ni més ni menys, que a la distribució marginal de cada variable.
La interpretació de les classes, a la majoria dels textes, és de tipus conjuntiu amb disjunció interna. Sota aquesta aproximació, la classe de l’exemple anterior s’interpretaria com La classe C d’objectes de forma = quadrat ∧ color = blau ∨ groc. La present proposta és encara més informativa que aquestes descripcions conjuntives, pel fet d’incorporar les proporcions amb cada valor apareix en una certa classe (quadrat, (1/2) groc,(1/2 blau)).

És més, ic donaria les característiques que tindrien els objectes de la classe C si haguessin de ser tots iguals. Aquesta és precisament la interpretació de la mitjana per les variables quantitatives!

Per tant, donada una classe C = \{i_1, ..., i_{nc}\} i la variable qualitativa X_k que pren valors en el conjunt \(D_k = \{c_1^k, ..., c_{nk}^k\}\), la \(k\)-èsima component de la descripció del representant de C, que anomenem ic, és:

\[ x_{ck} = \left( (f_1^k, c_1^k) \ldots (f_{nk}^k, c_{nk}^k) \right), \quad f_i^k = \frac{i_{ck}}{n_c} (1) \]

On \(i_{ck} = \text{card}(i \in C : x_{ik} = c_i^k)\) és el nombre d’individus de la classe C de modalitat \(c_i^k\).

Tenint en compte que per una variable quantitativa, la seva mitjana aritmètica es representaria com \(\bar{x}_{ck} = \frac{x_{1k} + \ldots + x_{nk}}{n_c}\) i que, traient factor comú per a tots els valors repetits d’\(X_k\), escriuríem \(\bar{x}_{ck} = f_1 x_1 + \ldots + f_k x_k\) es pot observar un paral·lelisme clar entre aquest concepte i l’expressió (1). Així, l’expressió (1) pot considerar com una generalització de la mitjana aritmètica on les \(c_i^k\) jugen el paper de les \(x_i\) i les \(f_i^k\) el de les \(f_i\), en un cas en que, per la naturalesa de les \(c_i^k\), sumes i productes no es tan definides. Aquesta analogia permet interpretar \(\bar{x}_C\) com el centre de gravetat de la classe C.

En resum, el representant d’una classe C és un vector \((\bar{x}_C, \ldots, \bar{x}_{CK})\) on:

\[ \bar{x}_{ck} = \left( c_1^k, c_2^k, \ldots, c_{nk}^k \right) \quad \text{si} \quad x_{ik} = c_i^k, \forall i \in C \]

i s’anomena valor compacte al primer i valor extès al segon.

### 4 Caracterització de classes

Donada una partitció d’un conjunt d’individus \(T\) és interessant oferir eines que ajudin a l’usuari a interpretar-la. Una d’elles és la caracterització de
les classes formades. Partint, doncs, d'una classificació \( \mathcal{P} = \{ C_1, \ldots, C_k \} \) dels elements d'\( \mathcal{I} \), un objectiu important és la identificació de les variables més rellevants en cada una de les classes formades; dit d'una altra manera, les que han resultat més decisives en la constitució d'aquestes i, eventualment, permetriem detectar la pertinença d'un objecte a una classe determinada, excloure-lo de les restants. En estadística, sistemes comercials com SPAD[?] permeten calcular les contribucions de cada modalitat de les variables categòriques a les diferents classes, i aquelles que resulten significatives s'utilitzen per interpretar-les. També es poden calcular les contribucions dels individus a cada classe, i observar com són els individus més típics d'aquestes. En les següents seccions s'aborda el problema des d'un punt de vista conceptual i es fa ús de les descripcions conceptuals de les classes formades per extreure informació sobre la classificació en una forma tal que permeti una fàcil interpretació del significat de les classes.

4.1 Variables descriptores

S'entén que \( X_k \) és una variable descriptora de la classe \( C, \) en sentit estricte per les dades estudiateis, si pren un únic valor per a tota la classe:

\[ X_k \text{ descriptora de } C \equiv x_{ik} = c_k^i, \forall i \in C \]

Proporcionar per a cada classe \( C \in \mathcal{P} \) la llista de variables descriptores en sentit estricte i els valors que pren aquestes variables, permet descriure les classes, i inclusa batejar-les. En el cas de les tiroides, per exemple, es pot parlar de la classe dels pacients que tenen l'hormona T4 baixa (\( C_3 \)), o de la classe dels pacients més grans de 60 anys (\( C_3 \)). . . Notis que una mateixa variable pot ser descriptora en sentit estricte d'una classe, però no ser-ho en una altra de la mateixa partit. La variable Gènere caracteritza \( C_3, C_4 \) o \( C_6 \) que són classes de dones, \( C_7 \) que és una classe d'homes, però no \( C_1 \) o \( C_2 \). Es pot dir que aquest sistema de descripció està més aviat en la línia dels weak-methods de la Intelligença Artificial.

La detecció d'aquestes variables és immediata:

\[ X_k \text{ descriptora en sentit estricte de la classe } C = \{ i_1 \ldots i_n \} \in \mathcal{P} \text{ si:} \]

\[ X_k \text{ descriptora de } C \equiv x_{Ck} \text{ és un valor compacte} \]

No obstant, en alguns casos reals de gran nombre d'elements, s'observa que per particions poc fines (poques classes de molts elements), pràcticament no s'identifiquen variables descriptores en sentit estrit\( ^4 \). Per aquesta raó, s'ha relaxat aquest concepte en el de variables descriptores (en sentit ampli), que correspond a variables que, si bé no prenen un valor constant dins la classe que descriuen, hi experimenten variacions suficientment petites com per a no ser considerades. La detecció de les variables descriptores (en sentit ampli) es fa a través de la concentració de proporció sobre una determinada categoria. Aquest és, doncs, un concepte relatiu al nivell de relaxació que estem disposats a assumir en la caracterització (que es denota per \( \varepsilon \) i està associat amb la possibilitat d'oblidar un objecte en la descripció d'una classe). Així, \( X_k \) és descriptora (a nivell \( 1 - \varepsilon) \) d'una classe \( C \in \mathcal{P} \) si:

\[ \exists s = (1: n_{k}) : \]

\[ x_{Ck} = c_k^s, \]

\[ f_{C}^s \geq (1 - \varepsilon), (4) \]

i tota variable descriptora en sentit estricte d'una classe \( C, \) és també descriptora de \( C \) a nivell \( (1 - \varepsilon), \) per qualsevol valor d'\( \varepsilon \) positiu. En aquest cas, el valor descriptre d'\( X_k, \) a nivell \( (1 - \varepsilon), \) per la classe \( C \) seria \( c_k^s \). Si s'agafa \( \varepsilon = 0.1, \) s'està considerant que un 10% de variació en el valor de la variable dins la classe és negligeable, i que la variable en qüestió es pot considerar fortement determinant de la classe.

Per exemple, en l'aplicació de les tiroides, podem dir que \( C_2 \) és una classe de dones a nivell 0.95, però no ho podem fer a nivell 0.99. Si \( A = \{ X_1 \ldots X_k \} \) és el conjunt de variables o atributs utilitzats en la matriu de dades, és denotar \( A_0^C = \{ X_k \in A : X_k \text{ descriptora de la classe } C \text{ a nivell } (1 - \varepsilon) \}. \) Amb aquest procés es poden construir descripcions conceptuals de les classes de caràcter conjuntiu, del tipus:

\[ \bigwedge_{X_k \in A_0^C} (X_k = v_{Ck}) \]

estructures que tenen una interpretació clara, i proporcionen una idea més o menys precisa (en funció d'\( \varepsilon \)) de la composició de les classes.

Sobre l'exemple que estem utilitzant, en sentit estricte es pot descriure les següents classes, amb les variables indicades tot seguit:

\( ^4 \)Això depèndrà de la variabilitat de les variables observades, i no sempre serà així. Disposarem de casos reals en que, amb una mostra de 100 objectes i unes 40 variables, en particions de només cinc classes, s'identifiquen fins a un 20% de variables descriptores en sentit estrict. 

\[ A_C^{\text{disp}} = \{ \text{Tèrapia, Edat, Gènere} \} \\
A_C^{\text{disp}} = \{ \text{Edat, Gènere} \} \\
A_C = \{ \text{Tèrapia, Nivell de T3} \} \\
A_C = \{ \text{Tèrapia, Gènere} \} \\
A_C = \{ \text{Gènere} \} \\
A_C = \{ \text{Nivell de T4} \} \]

donant lloc a les següents descripcions de classes:

Per \( C_5 \): dones de més de 90 anys sense teràpia \((\text{Teràp.} = \text{Cap}) \land (\text{Edat} = > 90) \land (\text{Gèn.} = \text{Dona})\)

Per \( C_4 \): Dones entre 16 i 30 anys
Per \( C_2 \): Persones sota teràpia i amb un nivell de T3
Per \( C_5 \): Persones sota teràpia i amb un nivell normal de T3
Per \( C_7 \): Dones
Per \( C_8 \): Nivell baix d’hormona T4

Relaxant fins a \( \varepsilon = 0.05 \), trobem una descripció de nivell 0.95 per les classes restants:

Per \( C_1 \): Dones sense teràpia, \( A_C^{\text{disp}} = \{ \text{Terà., Gèn.} \} \)
Per \( C_2 \): Dones, \( A_C^{\text{disp}} = \{ \text{Gènere} \} \)

### 4.2 Variables caracteritzadores

Ara bé, si l’objectiu és extreure la informació que permeti **identificar** les classes amb un mínim d’informació, llavors l’enfoc descriptiu resulta insiificant.

Que es trobi un \( A_C^{\text{disp}} \) amb molts elements per una classe determinada, informa clarament sobre l’homogeneïtat d’aquesta, però no ajuda necessàriament a distingir-la de les altres classes. Seria el cas, per exemple de la classe \( C_3 \), les variables descriptores de la qual, o bé coincideixen amb les d’altres classes — \( C_7 \) i \( C_2 \) també són classes de dones —, o bé són disjuntes — la variable Teràpia és descriptora de \( C_4 \) amb el valor cap, però no de \( C_6 \), malgrat que \( C_6 \) té un 70% de persones que tampoc segueixen cap teràpia, mentre que la variable Nivell de T4 descriu \( C_6 \) amb el valor baix, i no \( C_8 \).

Per a fer **identificació**, cal estudiar els valors que pren una variable dins una classe en relació a les altres. Això porta a introduir alguns conceptes nous que es presenten tot seguit. Donada una partició \( \mathcal{P} = \{ C_1, \ldots, C_l \} \), es defineix, com a **valor d’\( X_k \) propi de la classe** \( C \) al valor \( c^k \) en \( D_k \) si

\[ (\exists i \in C : x_{ik} = c^k_i) \land (\forall i \notin C : x_{ik} \neq c^k_i) \]

En l’exemple de les tiroides només n’hi ha un: totes les persones amb menys de 16 anys estan a \( C_1 \) (i a més són tan poques que representen menys d’un 0.01% de la classe). Així el valor \( \leq 16 \) de la variable Edat és propi per la classe \( C_1 \).

Una variable \( X_k \) és **parcialment caracteritzadora** de la classe \( C \in \mathcal{P} \) si

- té almenys un valor propi de la classe \( C \), per bé que en pugui compartir amb algunes classes; s’anomenarà llavors \( V_C^{\text{disp}} \) al conjunt de valors parcialment caracteritzadors de \( C \): \( V_C^{\text{disp}} = \{ c^k_i : c^k_i \text{ és valor propi d’} X_k \text{ per la classe } C \} \)

La variable Edat, doncs, seria parcialment caracteritzadora de la classe \( C_1 \) i el conjunt de valors parcialment caracteritzadors de \( C_1 \) és: \( V_{C_1}^{\text{disp}} = \{ \leq 16 \} \), tot i que a \( C_1 \) hi ha persones amb edats que apareixen també en altres classes.

Si considerem la classificació dels microordinadors obtinguda amb **KLASS**, que anomenarem KL per abreviar, trobem els següents conjunts de valors propis (restringint-nos a les tres variables qualitatives):

\[
V_{C_1}^{\text{disp}} = \{ \text{Color} - \text{TV} \} \\
V_{C_1}^{\text{disp}} = \{ 6502A \} \\
V_{C_8}^{\text{disp}} = \{ \text{B&W} - \text{TV} \} \\
V_{C_8}^{\text{disp}} = \{ 53 - 56 \} \\
V_{C_1}^{\text{disp}} = \{ \text{Terminal} \} \\
V_{C_1}^{\text{disp}} = \{ 8080A \} \\
V_{C_3}^{\text{disp}} = \{ \text{HP} \} \\
V_{C_3}^{\text{disp}} = \{ 92 \} \\

Una variable \( X_k \) és **totalment caracteritzadora** de la classe \( C \), si

- tots els valors que pren \( X_k \) a la classe \( C \) són propis de \( C \). En aquest cas, es denota \( \Lambda^k_C \) el conjunt d’aquestes valors, és quals caracteritzen totalment la classe \( C \):

\[ \text{si } \forall i \in C, x_{ik} \in V_C^{\text{disp}} \Rightarrow \exists \Lambda^k_C, \text{ i } \Lambda^k_C = V_C^{\text{disp}} \]

El cas de les tiroides no presenta cap variable totalment caracteritzadora per cap classe. En canvi, la classificació KL permet identificar els següents conjunts:

\[
A_{C_1}^{\text{disp}} = \{ \text{Color} - \text{TV} \}, \quad A_{C_7}^{\text{disp}} = \{ \text{B&W} - \text{TV} \}, \\
A_{C_3}^{\text{disp}} = \{ \text{HP} \}, \quad A_{C_4}^{\text{disp}} = \{ 92 \} 
\]

\[\text{La generalització de tots aquests conceptes a variables numèriques és, actualment, objecte d’estudis. La idea bàsica es fer extensives aquestes definicions a intervals de valors en bloc de categories i treballar amb els valors màxims i mínims de cada classe.}\]
Per tant, la condició de variable caracteritzadora (total o parcial) d’una classe és relativa a una partició de referència, i es pot donar o no, depenent de la distribució d’$X_k$ en les altres classes.

Si una classe té uns o més variables totalment caracteritzadores, qualsevol d’elles, per si sola, pot identificar-la.

La caracterització parcial és una generalització de la total, i resulta d’interès quan hi ha mena de caracteritzadors totals per una classe, fet que es dóna amb freqüència, com ja s’ha dit, en particions de poques classes definides sobre un conjunt gran d’individus.

El procediment consisteix a caracteritzar totes les classes que es puguin amb una única variable totalment caracteritzadora, genera un sistema de caracterització basat exclusivament en informació positiva, mínim per les classes totals caracteritzades, però que pot no ser complet, deixant classes sense caracteritzar. Tampoc té per què ser únic. El resultat d’aquest procés és un sistema de caracterització de $P$, que es representa com:

$$S_P = \{(C, X_k, \Lambda_k^C) : C \in P \land \Lambda_k^C \neq \emptyset\}$$

Represent la classificació $KI$ pels microordinadors, un possible sistema de caracterització mínim seria:

$$S_{KI} = \{(C_1, Display, \{Color - TV\}), (C_2, Display, \{B&W - TV\}, (C_4, MP, H P))\}$$

En podríem trobar d’altres. Una segona possibilitat és:

$$S_{KI} = \{(C_1, Display, \{Color - TV\}), (C_2, Display, \{B&W - TV\}, (C_4, Keys, 92))\}$$

Cap dels dos és complet, ja que la classe $C_3$ no té caracteritzador. És diu que un conjunt $S_P$ és un sistema de caracterització completa de $P$ si

$$\forall C \in P \exists k : (C, X_k, \Lambda_k^C) \in S_P$$

Amb el procediment plantejat, no es troba una caracterització completa per $KI$. En canvi, la partició proposada per l’expert (partició $\text{Exp}$) es pot caracteritzar de forma completa, per exemple, amb el sistema:

$$S_{\text{Exp}} = $$

$$\{(C_1, \text{Disp}, \{\text{Color - TV}\}), (C_2, \text{Keys, 92}), (C_2, \text{Disp}, \{\text{B&W-TV}\}), (C_3, \text{Disp}, \{\text{Built-in}\}), (C_4, \text{Disp}, \{\text{Terminal}\})\}$$

$I S_P$ és mínim si donats $C, C' \in P$, existeix $k, k'$ tals que $(C, X_k, \Lambda_k^C), (C', X_{k'}, \Lambda_{k'}^C) \in S_P$, aleshores $C \neq C'$. En l’exemple anteriore, tractem un sistema mínim seleccionant un sol caracteritzador per cada classe:

$$S_{\text{Exp}}' = \{(C_1, \text{Disp}, \{\text{Col.-TV}\}), (C_2, \text{Disp}, \{\text{BW-TV}\}), (C_3, \text{Disp}, \{\text{Built-in}\}), (C_4, \text{Disp}, \{\text{Term.}\})\}$$

Com es pot veure, el sistema de caracterització complet no sempre existeix i, si existeix, no té per què ser únic. L’hem trobat per la partició $\text{Exp}$ dels microordinadors, però pel cas de les tireoides, en canvi, el procés anterior no porta enllà, pel simple fet que ni tan sols existeixen variables totalment caracteritzadores per cap de les classes.

El següent objectiu és completar les caracteritzacions incompletes, a fi de poder identificar les classes que faltan. De fet, en el camp de l’aprenentatge automàtic supervisat, del que IDS i els seus successors són un exemple característic, un objectiu és aconseguir mètodes robustos que permetin la identificació de la classe a la qual pertany un objecte sabent, tan sols, el valor que prenen alguns atributs. La construcció d’aquestes arbres de decisió sol tenir cost exponencial i existeixen tècniques que escurcen raonablement l’espai de cerca. Aquí s’explora una possibilitat més barata, que és l’ús d’informació condicionada de forma incremental.

### 4.3 Caracterització per condicionalment successiu

D’un sistema de caracterització $S_P$ es pot generar informació negativa també. Si $\Lambda_k^C$ caracteritza totalment la classe $C$, significa que en cap altra classe de $P$ hi ha individus que prenguin per $X_k$ valors de $\Lambda_k^C$. D’altra manera, $\Lambda_k^C$ es podria interpretar, doncs, com un caracteritzador parcial de qualsevol altra classe de $P$. En realitat, tal afirmació involucra una hipòtesis de món tancat amb tot el risc que això comporta, però és compta amb l’expert per calibrar la validesa d’aquestes$^3$.

Tot parell $(P, S_P)$ tal que $S_P$ no és complet, determina un conjunt $P' \subset P$

$$P' = \{C \in P : \forall k (C, X_k, \Lambda_k^C) \notin S_P\}$$

que és el de totes les classes que no s’han pogut

---

$^3$Ca l’ambit d’aquests tipus d’hipòtesis, s’ha de prendre les precaucions necessàries per a anar més enllà de la simple descriptiva.
caracteritzar totalment. En el cas de \( (KI, S_{KI}) \),
\[ KI' = \{ C_3 \} \]

Per la partició restringida \( P' \) es pot calcular igualment el sistema de caracterització associat \( S_{P'} \).
En l’exemple, es tractaria de caracteritzar \( C_3 \), pre-
scindint dels elements de les altres classes. En aquest cas, possibles sistemes complets serien:
\[
S_{KIP} = \{(C_3, Display, \{Terminal, Built-in\})\},
S_{KIP} = \{(C_3, MP, \{Z80, 8080A\})\}
\]

Prenem-ne un qualsevol, per exemple el primer dels
dos.
Si \( S_{P'} \neq \emptyset \), significa que hi havia solapament en-
tre els valors d’algunes variables per les classes de \( P' \) i alguna classe de \( (P - P') \). La caracterització de \( P' \) es pot fer utilitzant com a condicionant el fet
que estem en el conjunt \( (P - P') \), i això és possible
tractant informació d’\( S_{P} \) en sentit negatiu. Per ex-
emple, en la classificació \( KI \) dels microordinadors,
\( C_3 \) no es podia caracteritzar totalment amb el con-
joint \( (Terminal, Built-in) \) perquè a \( C_4 \) hi havia també ordinadors amb pantalla incorporada (\( Display=Built-in \)). Per la hipòtesi de món tancat que
qualsevol ordinador amb un microprocesador que
no sigui HP, no és de la cuarta classe, permet carac-
titzar totalment \( C_3 \) a partir de la variable \( Display \).
S’està doncs utilitzant la informació negativa “Si
(\( MP \neq HP \)) aleshores el microordinador no és de la
classe \( C_4 \)” per a caracteritzar la classe \( C_3 \), i el resul-
tat és l’ampliació del sistema de caracterització amb
l’element \( (C_3, Display, \{Terminal, Built-in\}) \), subjecte, està clar, a l’esmentada hipòtesi.
Aquest mateix procés es pot repetir per totes les
classes que no han estat totalment caracteritzades.
El resultat és una possible ampliació del sistema de
caracterització de \( P \) subjecta a l’ús d’algunes condi-
cions en sentit negatiu, el conjunt de les quals es
dena per \( \mathcal{S}_{P'}^{\text{end}} \) i es representa com:
\[
\mathcal{S}_{P'}^{\text{end}} = \{(C', X, \Lambda^b_k) \, \exists C, k : (C, X_k, \Lambda^c_k) \in S_P \land
\land \exists k', (C', X_{k'}, \Lambda^b_k') \in S_{P'} \land \exists i \in C : x_{ik} \in \Lambda^c_k' \}
\]
En l’exemple, \( \mathcal{S}_{KI}^{\text{end}} = \{(C_3, MP, \{HP\}) \} \) i \( S_{KI} =
\{(C_3, Display, \{Terminal, Built-in\})\} \)

En un segon pas, es determinaria \( P'' \) (classes de
\( P' \) que no han quedat totalment caracteritzades).
L’\( \mathcal{S}_{P'}^{\text{end}} \) hauria d’heretar els condicionants d’\( S_{P'} \)
també, i així successivament.

8Simètricament, \( P - P' \) és el conjunt de totes les classes de
\( P \) que han estat totalment caracteritzades.

La caracterització final de les classes es fa amb els
successius sistemes de caracterització trobats. Tamp-
poc així es té la garantia que la caracterització sigui
completa, però, incloent pel darrer \( P' \) totes les vari-
ables caracteritzadores parciaus, s’extreu el màxim
d’informació sobre les classes que es pot obtenir con-
sultant variables d’una en una. I amb això no es
vol pas dir que la caracterització tingui lloc amb
una sola variable en cada classe. De fet, la
imbricació de condicionants donarà lloc a jonc-
ctions tant llargues com nivells d’imbricació s’efectuïn.
L’aplicació dels microordinadors té una resposta
molt favorable al mecanisme que es presenta:
algorisme permet obtenir caracteritzacions com-
pletes de totes les classificacions que es disposa. En
aquest treball hem completat únicament \( KI \). Val a
dir, no obstant, que aquest és un procediment basat
en criteris sintàctics, i el sistema de caracterització
resultant no té perquè tenir una utilitat pràctica
directa, per bé que la seva interpretació semàntica
pugui ser clara. De vegades, la utilitat depèndrà del
cost d’observació de les variables caracteritzadores.
En aquest sentit esmentar el cas de les \( \beta \)-bandes
en colesterols que, malgrat ser uns excel·lents caracte-
rizadors d’aquestes cadenes i tenir plè significat
semàntic, s’observen amb tanta dificultat, que no
es poden utilitzar com a tals. En altres ocasions, el
que es posa en dubte és el propi significat de la clas-
ificació. Si la caracterització no és interpretable, la
classificació pot ser carent de sentit.

5 Algorisme
La caracterització de \( P \), es planteja, en una primera
aproximació, com un procés iteratiu de la forma:

- Trobar un \( S_P \) mínim.
- Determinar el conjunt de les classes que no
s’han caracteritzat totalment \( P' \) (si \( P'' = \emptyset \) fi-
nalitza el procés: s’ha obtingut una caracter-
ització completa).
- Construir \( \mathcal{S}_{P'}^{\text{end}} \) i determinar \( S_{P''} \) subjecte a
\( \mathcal{S}_{P'}^{\text{end}} \).
- Si \( S_{P'} = \emptyset \), determinar el conjunt de variables
parcialment caracteritzadores de \( P' \) i finalitzar.
En cas contrari, reduir el conjunt de classes a
les que no s’han caracteritzat totalment i reple-
tir el procés.
Pel que fa a la implementació, la detecció de variables caracteritzadores passa per l’estudi de les interseccions de modalitats de cada variable en les diferents classes. Això suposa, en principi, un cost $O(n\text{mods} \times n\text{classes})$, essent $n\text{mods}_k$ el nombre de modalitats de cada una de les variables qualitatives.

Ara bé, amb la llista de variables i la partició a caracteritzar es poden construir les estructures auxiliars:

$$\begin{align*}
X_k &\rightarrow (\text{codi}_{ij}) \\
X_k &\rightarrow (\text{mods}_{ke})
\end{align*}$$

on $\text{codi}_{ij}$ és un codi numèric que compacta una tercera dimensió corresponent a les classes, i que es construeix com segueix:

- Si la classe $c$ conté elements que per la variable $X_k$ prenen el valor $c_j$, llavors $\text{codi}_{ij}$ té un 1 a la posició $c$.

- En cas contrari, $\text{codi}_{ij}$ té un 0 a la posició $c$.

i $\text{mods}_{ke}$ representa una mena de marginal dels codis, indicant el nombre de modalitats no buides de la variable $X_k$ a la classe $C$, la complexitat del cas pitjor es reduirà sensiblement.

La construcció es pot fer en paral·lel amb un cost $O(n\text{mods} \times n\text{classes})$. La mera consulta d’aquestes estructures permet detectar de forma molt econòmica, una variable caracteritzadora d’una classe (el seu codi és potència exacta de 2) i distingir si la caracterització és parcial o total (si el nombre de modalitats caracteritzadores de la classe per la variable en qüestió coincideix amb $\text{mods}_{ke}$ la caracterització és total, i sino és parcial).

Amés aquesta representació de l’informació, s’aconseguix que el cost del procés de detecció de les classes caracteritzadores sigui $O(n\text{mods})$. Haver representat desplegada la tercera dimensió de la matriu hauria portat a un cost molt superior, ja que per cada modalitat s’hauria hagut d’estudiar a quines classes apareixia, la qual cosa és $O(n\text{mods} \times n\text{classes}^2)$. És a dir que l’estalvi tant en espai com en temps és significatiu, especialment si es treballa amb particions de baixos nivells de l’arbre.

El procediment general és un cicle que realitza caracteritzacions condicionades successives, fins que no troba cap classe caracteritzable totalment. En aquest punt, proporciona les variables parcialment caracteritzadores de les classes restants. En el pitjor cas, a cada pas es caracteritzarà totalment una única classe i s’hauria de fer $n\text{classes}$ iteracions. El cost total del cas pitjor es calcula considerant la repetició ($n\text{classes} - 1$) cops de les següents passes:

- Obtenció de la descripció conceptual de cada classe: $O(n\text{classes})$.
- Construcció de les estructures auxiliars: $O(n\text{mods} \times n\text{classes})$.
- Detecció de caracteritzadores totals: $O(n\text{mods})$
- Restringir les estructures de treball a les classes que encara no s’han caracteritzat totalment: $O(n\text{classes} - \xi)$ en el pitjor cas són totes excepte una.

I això porta a un cost total de l’ordre $O(n\text{classes}^2, n\text{mods} \times n\text{classes}^2)$.

### 5.1 Apunt a la generació automàtica de regles

La secció anterior pren com objectius purament descriptius que extreguin el màxim d’informació útil de les dades. Ara bé, el càlcul de variables caracteritzadores pot ser el punt de partida d’un sistema de generació automàtica de regles de classificació, que es situa ja en tot un altre ordre de coses.

En aquesta àrea de recerca cal citar tres direccions de treball, que són les més actives actualment: d’una banda, la generació automàtica de regles a partir de la descripció intensional i extensional de les classes, una certa teoria del dominí i l’ús de tècniques de dividir i conquerir [7]; d’una altra, la generació de regles a partir de tècniques d’IBL del tipus conquerir sense dividir (aprenentatge inductiu) [1]; finalment, l’aprenentatge algorísmic (PAC-learning) [8].

El nostre objectiu és obrir un canal que apropi l’estadística a aquest tipus de tècniques, sense pretindre, en absolut, esser exhaustiu. La nostra proposta es basa en l’ús de les variables caracteritzadores de les classes (definides a la secció anterior a partir de les distribucions marginals de cada variable en cada classe) en la generació automàtica de regles. En concret, de la detecció d’una variable totalment caracteritzadora de $C$ amb els valors
propis de \( \mathcal{C}, \Lambda_k^C \), se n'extreu una regla que identifica la classe amb el mínim d'informació, de la forma:

\[
(X_k \in \Lambda_k^C) \rightarrow \mathcal{C}
\]

com podria ser, per exemple, la regla \( r : \text{Display} = \text{Color} \rightarrow \text{TV} \rightarrow \mathcal{C}_1 \).

Els objectes seleccionats per aquests tipus de regles coincideixen exactament amb \( \mathcal{C} \) i també amb els seleccionats per la regla associada a qualsevol altra variable totalment caracteritzadora de la classe \( \mathcal{C} \).

D'aquesta forma, la nostra proposta es situaria en un punt intermíx entre les dues primeres línies de recerca mencionades.

Si es parteix d'un sistema de caracterització completa de \( \mathcal{P}, \mathcal{S}_P \), el conjunt de regles que se'n deriven permetrà identificar exactament les classes de \( \mathcal{P} \). És el cas de la classificació \( \mathcal{E}_p \), que donaria lloc al sistema de regles:

\[
\begin{align*}
\text{r}_1 : & \quad \text{Display} = \text{Color} \rightarrow \text{TV} \rightarrow \mathcal{C}_1 \\
\text{r}_2 : & \quad \text{Display} = \text{B&V} \rightarrow \text{TV} \rightarrow \mathcal{C}_2 \\
\text{r}_3 : & \quad \text{Display} = \text{Built \ - \ in} \rightarrow \mathcal{C}_3 \\
\text{r}_4 : & \quad \text{Display} = \text{Terminal} \rightarrow \mathcal{C}_4
\end{align*}
\]

En aquest cas, la quarta regla es podria substituir tranquil·lament per:

\[
\text{r}'_4 : \text{Keys} = 92 \rightarrow \mathcal{C}_4
\]

Si la caracterització completa està subjecta a un \( \mathcal{S}_P^{cond} \neq \emptyset \), llavors la generació de regles ha d'incloure la informació condicionant:

Donats \( \mathcal{S}_P^{cond} \) i \( \mathcal{S}_P \), es pot generar un conjunt de regles de la forma

\[
r : \Lambda \big( X_k \in \Lambda_k^C \big) \wedge \big( X_k \notin \Lambda_k^C \big) \quad \mathcal{C} \quad [k : (\mathcal{C}, X_k, \Lambda_k^C) \in \mathcal{S}_P^{cond}] \quad [k' : (\mathcal{C}, X_k', \Lambda_k^C) \in \mathcal{S}_P^{cond}]
\]

Així, per la classificació \( \mathcal{K} \) es poden generar les següents regles:

\[
\begin{align*}
\text{r}_K_1 : & \quad \text{Display} = \text{Color} \rightarrow \text{TV} \rightarrow \mathcal{C}_1 \\
\text{r}_K_2 : & \quad \text{Display} = \text{B&V} \rightarrow \text{TV} \rightarrow \mathcal{C}_2 \\
\text{r}_K_3 : & \quad \text{Display} \in \{ \text{Terminal, Built \ - \ in} \} \wedge \\
& \wedge (M \neq H \rightarrow \mathcal{C}_3 \\
\text{r}_K_4 : & \quad M \neq H \rightarrow \mathcal{C}_4
\end{align*}
\]

Regles similars es poden construir per les variables parcialment caracteritzadores de les classes, les quals servirien per a identificar parts d'una classe. Per bé que, en aquest cas, el sistema de regles no seria complet, es pot assegurar que no genera mai errors de selecció dels objectes de la mostra.

Si la mostra representa bé la població en estudi, la qualitat de selecció del sistema de regles així construït serà molt elevada. En aquest sentit, el sistema es comporta de forma molt \textit{conservadora} en aquesta fase, generant un conjunt de regles que no siguin massa generals.

Si, pel cas de \( \mathcal{K} \) no s'hagués volgut utilitzar la hipòtesi de món tancat, es podia haver construït el següent sistema mínim de regles, no complet:

\[
\begin{align*}
\text{r}'_K_1 : & \quad \text{Display} = \text{Color} \rightarrow \text{TV} \rightarrow \mathcal{C}_1 \\
\text{r}'_K_2 : & \quad \text{Display} = \text{B&V} \rightarrow \text{TV} \rightarrow \mathcal{C}_2 \\
\text{r}'_K_3 : & \quad \text{Display} = \text{Terminal} \rightarrow \mathcal{C}_3 \\
\text{r}'_K_4 : & \quad M \neq H \rightarrow \mathcal{C}_4
\end{align*}
\]

Aquest darrer sistema, permetria identificar la classe de pertinença d'un 84% dels microordinadors, en aquest cas, no està malament. I no hi ha assignacions errònies per la forma tan \textit{conservadora} com s'han generat el sistema de caracterització i les regles associades.

A partir de la informació mostral es genera una descripció de les classes en forma de regles, que permet identificar els objectes observats. Des d'un punt de vista estadístic, no es pretén pas, de moment, anar més enllà d'aquesta descriptiva. Tot pas d'inferència que permetés el salt al món dels sistemes experts, hauria de tenir en compte que aquest procés es realitza amb informació parcial i validar les hipòtesis de món tancat que es construeixen, així com quantificar l'error d'assignació que es pot produir per mor de la qualitat de la mostra estudiada, fent ús de les proves estadístiques pertinents. Aquest punt resta fora de l'àmbit d'aquest article, en el que es planteja la caracterització de les classes com una simple eina d'extracció de la informació útil continguda en la classificació, amb tal de contribuir a la seva interpretació.

El coneixement que hem pot extreure d'un càlcul de variables descriptors, com és el cas de l'aplicació de les trioides, pot donar lloc a un sistema de regles d'interpretació que tenen tota una altra forma. Si la descripció és a nivell \( \varepsilon \), llavors és convenient construir una base de regles probabilístiques:

\[
\bigwedge_{X_k \in \Lambda_k^C} \big( X_k = v_k \big) \rightarrow \mathcal{C}
\]

Pel cas de les trioides, es pot representar una \textit{interpretació} de les classes de la següent forma, que resulta perfectament entenedora a l'usuari:
6 Conclusions i treball futur

En primer lloc es presenta una forma de descrició de forma compacta les classes basada en l'ús de la distribució marginal de cada variable en cada classe, la qual cosa ens permet fer una certa analogia amb el concepte de centre de gravetat que s'utilitza per a variables numèriques.

A partir d'aquest prototipus de les classes, es descriu un sistema per a identificar les variables descriptores, fixant un cert nivell de classe, les quals permeten fer-se una idea de la composició d'aquesta i corresponen a un enflor maximalista, en el sentit que la descripció d'una classe es fa amb el màxim nombre de variables possible de valor constant (o quasi-constant) a ella. Aquest mecanisme anàleg en la línia dels weak-methods propis de la Intel·ligència Artificial, i és basa en un principi similar al que fan els sistemes estadístics clàssics, que consideren el màxim nombre de modalitats amb contribució significativa a una classe per a poder-la interpretar.

Plantejada la insuficiència d'aquest enflor per a generar descripcions que permetin identificar les classes, es defineix un procediment de caracterització que permet detectar conjunts mínims, ara sí, de variables que portin a distingir una classe de les altres, mecanisme que pot incidir directament en processos de generació automàtica de regles.

S'introduceix, amb aquest fi, el concepte de variable totalment caracteritzadora d'una classe $C$, que és la que pren, a la classe $C$, valors específics i diferents de les altres classes. A partir d'aquí i analitzant la completitud de les caracteritzacions, s'arriba a un procediment que utilitza informació condicional de forma imbricada, amb un cost $O(n\cdot m\cdot k\cdot n_{classes})$ i pot generar descripcions conjuntives de llargada variable, formades, sempre, d'un i només un terme positiu, acompanyat d'un seguit de condicionants negatius.

Donada l'estructura d'aquestes descripcions, no es pot garantir la completa de la caracterització resultant, per a la qual cosa caldría repetir el procés estudiant parelles, ternes, en general $n$-tuples, de variables que poguessin caracteritzar globalment les classes restants. El problema es torna combinatori i la seva complexitat creix molt. No obstant, la implementació que es proposa reduceix notablement el cost de calcular aquestes caracteritzadores (es passa de $O(n \cdot m \cdot k \cdot n_{classes})$ a $O(n \cdot m \cdot k \cdot n_{classes})$).

Una segona possibilitat consisteix a assumir un risc d'error en la caracterització i relaxar els conceptes que s'han definit. Això tindrà repercussions en la posterior generació de regles, que seran regles probabilístiques. Actualment s'està treballant en la caracterització aproximada (a nivell c) de partícules, a partir de valors quasi-propis per una classe i que quasi-caracteritzadors. Això significa passar a treballar amb proporcions per columnes (d'una modalitat en una classe respecte la mateixa modalitat en les altres classes), enflor de proporcions per files (d'una modalidad en una classe respecte les altres modalitats en la mateixa classe). Ja no és suficient el prototipus de la classe, sino que cal també el nombre total d'elements de cada classe i cal propuitar l'error d'assignació en els successius condicionaments.

La generació automàtica de regles a partir dels sistemes de caracterització definit és directa i és una contribució, de la que no en coneixem antecedents, a l'aprofundament de l'estadística cap a aquesta àrea de recerca.

D'altra banda, queda per presentar el tractament de variables numèriques, que involucra resultats força similars als introduïts en aquest treball. Com ja s'ha comentat, actualment s'està també treballant en aquest tema i la idea de base és estudiar les aglomeracions de valors contínus en les diferents classes.

References


Applying Qualitative Spatial Reasoning to Robot Navigation

M. Teresa Esug - Francisco Toledo

Universitat Jaume I, Campus de Penyeta Roja, 12071 Castellón (España)
E-mail: {esugm, toledo}@inf.ujh.es

Abstract

The integration of the qualitative concepts of orientation, distance, and cardinal directions, using points as well as extended objects as primitive of reasoning has been successfully accomplished in this paper by using Constraint Logic Programming instantiated to Finite Domains extended with Constraint Handling Rules as a tool. The resulting model has been applied to build a demonstrator: a Qualitative Navigation Simulator on the structured environment of the city of Castellón.

Keywords: Spatial Reasoning, Qualitative Reasoning, Constraint-Based Reasoning, Constraint Handling Rules, Qualitative Robot Navigation.

1 Introduction

If we attempt to simulate the human spatial reasoning process, it is necessary to represent the information included in descriptions such as "the cinema is to your right, far from here and to the north-west of the city", which contain orientation, distance and cardinal directions information, related to the same spatial landmark. Moreover, when more spatial relationships between different landmarks are provided, it is necessary to reason with them.

In the last years, many qualitative spatial models have been developed to manage properly the imprecise knowledge about different aspects of space, included in sentences such as the above mentioned. For the concept of orientation, there exist mainly three models based on orthogonal projections (figure 1a) [12] and [16] and non orthogonal projections [22] and three models not based on projections (figure 1b) [9, 10, 8] and [14]. In the models based on projections, the relative orientation among objects is obtained by drawing orthogonal or no orthogonal projections, and then reasoning in one-dimension by using the Allen's temporal reasoning [1]. In the models not based on projections, the 2-dimensional space is divided into qualitative regions by means of Reference Systems (RSs) which are centered on the referenced object. The

Figure 1. Qualitative orientation models a) based on projections and b) not based on projections.

set of models not based on projections are considered more cognitive due mainly to two reasons: (1) people do not reason about spatial orientation by doing projections on "inexistent" external axes. They rather think about egocentric RSs based on the asymmetry of the human body (front/back) and both arms (right/left) [14]; and (2) in models based on orthogonal projections, it is possible to infer inconsistent knowledge (for instance, two overlapped relationships between the projections of objects in both axes, X and Y, do not imply that the two objects overlap in the two-dimensional space). Due to all these reasons we are more interested in models not based on projections.

Although the qualitative models based on projections consider extended objects as primitive of reasoning—sometimes objects are approximated to rectangles—qualitative models not based on projections always simplify objects to points.

The qualitative orientation models not based on projections have also been extended with the concept of distance in the following models [24], [15] and [3] (see [7] for further details).

As we have pointed out, there exists several qualitative models to represent and reason about orientation and positional information. However, the integration of orientation, distance and cardinal
directions into the same qualitative model is a problem which remains unsolved. That is, there is no general model to reason with all these spatial aspects in a uniform way.

Therefore the objectives of our work are the following:
(1) the integration of several aspects of space into the same qualitative model,
(2) the use of point and extended objects as primitive of reasoning, and
(3) the application of the spatial reasoning model to Qualitative Navigation.

The bases for the integration of several spatial aspects in the Spatial Reasoning (SR) domain have been inspired in the Temporal Reasoning (TR) field.

2 Bases for the Integration of Several Temporal Aspects

Many forms of temporal relationships (such as point and interval algebra and metric information) have been integrated thanks to their consideration as instances of the Constraint Satisfaction Problem (CSP) [20]. Generate and test and backtracking are algorithms which solve the CSP, although in a very inefficient way. These algorithms have an exponential cost. Research in the field tries to improve efficiency of the backtracking algorithm (a review of the state of the art can be found in [21] and [18]). A set of these algorithms modify the search space before the search process starts, to make the search process easier. They are called algorithms which improve consistency. These algorithms are based on the idea of making explicit the implicit constraints by means of the constraint propagation process. Unfortunately the complete constraint propagation process is also hard, therefore the process is approximated by local constraint propagation, as path consistency. If the constraint graph is complete (that is, there is a pair of arcs, one in each direction, between every pair of nodes) it suffices to repeatedly compute paths of two steps in length at most. This means that for each group of three nodes (i,k,j) we repeatedly compute the following operation until a fix point is reached [11]:

\[ c_{ij} := c_{ij} \oplus c_{ik} \otimes c_{kj} \] (1)

This operation computes the composition of constraints (\( \otimes \)) between nodes ik and k and the intersection (\( \oplus \)) of the result with constraints between nodes ij. The complexity of this algorithm is \( O(n^2) \), where \( n \) is the number of nodes in the constraint graph (that is, the number of objects involved in the reasoning process) [19].

Constraint Handling Rules (CHR) are a tool which helps to write the above algorithm. They are an extension of the Constraint Logic Programming (CLP) which facilitate the definition of constraint theories and algorithms which solve them. They facilitate the prototyping, extensions, specialization and combination of CSPs [11] [2]. The composition part of the formula \( (c_{ik} \otimes c_{kj}) \) will be implemented by propagation CHR which are of the form:

\[ H_1, \ldots, H_i \Rightarrow G_1, \ldots, G_j \mid B_1, \ldots, B_k \]

where \((i>0, j>0, k \geq 0)\). It means that if a set of constraints matches the head of a propagation CHR \( (H_1, \ldots, H_i) \) and the guards \( (G_1, \ldots, G_j) \) are satisfied, then the set of constraints \( B_1, \ldots, B_k \) is added to the set of initial constraints \( (H_1, \ldots, H_i) \).

The part of the formula which refereces to intersection \( (c_{ij} \otimes \ldots) \) will be implemented by simplification CHR which are of the form:

\[ H_1, \ldots, H_i \Leftrightarrow G_1, \ldots, G_j \mid B_1, \ldots, B_k \]

where \((i>0, j>0, k \geq 0)\). It means that if a set of constraints \( H^* \) matches the head of a simplification CHR \( (H_1, \ldots, H_i) \) and the guards \( (G_1, \ldots, G_j) \) are satisfied, the set of constraints \( H_1, \ldots, H_i \) is substituted by the set of constraints \( B_1, \ldots, B_k \). The set of constraints \( B_1, \ldots, B_k \) is simpler than the set of constraints \( H_1, \ldots, H_i \) and preserves logical equivalence.

3 Bases for the Integration of Several Spatial Aspects in the SR Field

We are going to distinguish two parts in the spatial reasoning process: the Basic Step of the Inference Process (BSIP) and the Full Inference Process (FIP). For those models not based on projections, the BSIP can be defined in general terms such as: given a spatial relationship between object A with respect to (wrt) a RS, RS1, and another spatial relationship between object B wrt another RS, RS2, being object A part of the RS2, the BSIP consists of obtaining the spatial relationship of object B wrt the RS1. The RS will be different depending on the model. When more relationships among several spatial landmarks are provided, then the FIP is necessary. It consists of repeating the BSIP as many times as possible, with the initial information and the information provided by some BSIP, until no more information can be inferred. To accomplish the integration of orientation, distance
and cardinal directions into the same spatial model, we will use the following three steps which will be explained in the next sections:

1. the representation of each spatial aspect;
2. the definition of the BSIP for each aspect; and
3. the definition of the FIP for each spatial aspect.

### 3.1 Qualitative Orientation

#### 3.1.1 The Representational Model

For representing the two-dimensional orientation information, Freksa and Zimmermann's model [9, 10] has been chosen. The orientation in their RS is defined by two points, a and b, which defines the left/right dichotomy and can be interpreted as the direction of movement. The coarse RS also includes the perpendicular line by the point b, which defines the first front/back dichotomy (F/B), and can be seen as the straight line which joins our shoulders. This coarse RS divides the space into 9 qualitative regions. A finer distinction could be made in the back regions by drawing the perpendicular line by the point a. In this case, the space is divided into 15 qualitative regions (figure 2a). The point a defines the second front/back dichotomy (2F/B) of the RS. An iconic representation of the fine RS and the names of the regions are shown in figure 2b). The information which can be represented by this RS is the qualitative orientation of a point object, c, wrt the RS formed by the point objects a and b, that is, \(c \text{ wrt } ab\). The relationship \(c \text{ wrt } ab\) can also be expressed in other five different ways: \(c \text{ wrt } ba\), \(a \text{ wrt } bc\), \(a \text{ wrt } cb\), \(b \text{ wrt } ac\) and \(b \text{ wrt } ca\) which are the result of applying the inverse (INV), homing (HM), homing-inverse (HMI), shortcut (SC) and shortcut-inverse (SCI) operations, respectively (figure 2c). In our model these operations have been represented as facts of the PROLOG database. For instance, the inverse of the spatial relationship "left-front" is the spatial relationship "back-right" (\(\text{inv}(lf, frb)\)). These operations will be used for the definition of the FIP for orientation information.

#### 3.1.2 The BSIP

The BSIP for the qualitative spatial orientation is defined such that "given the relationships \(c \text{ wrt } ab\) and \(d \text{ wrt } bc\), we want to obtain the relationship \(d \text{ wrt } ab\)" (figure 2d).

In [10] two inference tables of 9x9 entries are introduced to define this BSIP. One table corresponds to the coarser division of the space into 9 regions and the other one to the inference between the relations in which the coarse back relations have been split up. These tables include the composition between the 15 qualitative regions in which the fine RS divides the space, although in an implicit way [6,7]. In that sense, we say that these inference tables are complete. However, we make explicit the implicit information included in these tables by the definition of the 15x15 inference table shown in figure 3. Disjunctions of relationships are represented by more than a black square on the iconic figure. This composition table has been implemented in our approach by facts of the PROLOG database (for example, \(\text{composition}(sf, sb, \{sm, ib, sb\})\)).

#### 3.1.3 The FIP

The above explained orientation relationships (\(c \text{ wrt } ab\)) are treated in our model as tertiary constraints \(\{c_{ab}(X_a, X_b, X_c)\}\) which relate triples of variables \(X_a, X_b, X_c\), \((1 \leq a,b,c \leq n)\). They are seen as an instance of the CSP. The operation (1) to compute
path consistency is redefined for tertiary constraints so that:
\[ c_{\text{d,ab}} := c_{\text{d,ab}} \oplus c_{\text{c,ab}} \oplus c_{\text{d,bc}} \]  \hspace{1cm} (2)

The orientation constraint \( c_{\text{c,ab}} \) is represented by the predicate \( \text{ctr}(C, A, B, \text{Rel}) \), where \( A, B, \) and \( C \) are variables which refer to spatial objects and \( \text{Rel} \) is a list which contains a disjunction of orientation relationships between the three spatial objects.

The intersection part of the formula ("\( \oplus \)") is implemented with the following simplification CHR:
\[ \text{ctr}(C, A, B, R1), \text{ctr}(C, A, B, R2) \leftrightarrow \text{intersection}(R1, R2, R3) \mid \text{ctr}(C, A, B, R3) \]  \hspace{1cm} (3)

If the constraint graph were complete, no more simplification CHRs would be needed. However, normally there is no relationship between every object in space. This lack is supplied with the definition of more simplification rules by using the five operations mentioned. By applying the five operations to the first and second constraints of the head of rule (2) independently, we have defined 11 simplification CHRs to compute the intersection part of operation (2).

The part of the operation (2) which refers to the composition ("\( \oplus \)") corresponds to the BSIP defined in the previous section. It is implemented by propagation CHRs. For instance:
\[ \text{ctr}(C, A, B, R1), \text{ctr}(D, B, C, R2) \Rightarrow \text{composition}(R1, R2, R3) \mid \text{ctr}(D, A, B, R3) \]  \hspace{1cm} (4)

Where composition/3 refers to the set of facts of the PROLOG database which defines the 15x15 inference table shown in figure 3b.

The lack of completeness of the constraint graph is also supplied by the definition of a total of 11 propagation CHRs, as it happened with the intersection part of operation (2).

3.2 Qualitative Position

3.2.1 The Representational Model

Positional information includes orientation and distance information. The orientation information will be represented using the model explained in the previous section. In the concept of distance it is possible to distinguish between comparing magnitudes of distances and naming distances. Comparing versus naming refers to the usual distinction between "relative" and "absolute" distance which is not accurate because both are relative. In this paper we focus our attention on named distances. The distance model that we are going to define is inspired in the model by Clementini et al. [3], although with several differences which we will point out.

The positional relationships which we are going to represent are (figure 4): (1) the orientation \( c \) with \( ab \), and the distance measured from the first point of the front/back dichotomy ("c wrt ab from 1st" for short), and (2) the orientation \( c \) wrt \( ab \) and the distance measured from the second point of the front/back dichotomy ("c wrt ab from 2nd" for short). Both relationships can also be expressed of three different ways by using the operations defined for the orientation relationships which make sense for positional relationships, that is, INV, SC and SCI for c wrt ab from 1st and INV, HM, and HMI for c wrt ab from 2nd [7].

For named distances it is necessary to define a reference system (which we will call Distance Reference System (DRS)). It is composed of a list of distance relations and a set of acceptance areas. The list of distance relations is a set of distance symbols in increasing order, i.e. \( Q = \{q_0, q_1, ..., q_n\} \) where \( q_0 \) is the closest distance to the reference object, and \( q_n \) is the one farthest away, going to infinity. The number of distance symbols will depend on the granularity of the task. The acceptance areas will be described by a set of intervals, i.e. \( Dr = \{d_0, d_1, ..., d_n\} \). Each distance symbol \( q_i \) is associated with an acceptance area \( d_i \). An example of a DRS is shown in figure 5 where the boundaries of regions defined by the acceptance areas overlap (representing the difficulties of humans in determining the lines which divide regions).
symbols, from the lower bound to the upper bound, is obtained in the BSIP. See [7] for details of the algorithms.

3.2.3 The FIP
Positional relationships are also considered an instance of the CSP, where the constraints are of the form: $c_{c,ab}$ from 1st and $c_{c,ab}$ from 2nd. The operation (1) to compute path consistency is redefined for positional information by the following formulas (5), (6) and (7) which correspond to the three cases in which the BSIP for positional information has been split up:

$c_{d, ab}$ from 1st := $c_{d, ab}$ from 1st ⊕ $c_{c,ab}$ from 1st ⊕ $c_{d, bc}$ from 1st
$c_{d, ab}$ from 2nd := $c_{d, ab}$ from 2nd ⊕ $c_{c, ab}$ from 2nd ⊕ $c_{d, bc}$ from 1st
$c_{d, bc}$ from 1st := $c_{d, bc}$ from 1st ⊕ $c_{c, ab}$ from 1st ⊕ $c_{d, bc}$ from 2nd

The positional relationships are represented by the predicate $ctr\_dist(C,A,B,Reldist, Relorient, Rs, ForS)$ where A, B and C represent spatial objects, $Reldist$ is the disjunction of distance relationships, $Relorient$ is the disjunction of orientation relationships, $Rs$ refers to the distance RS and $ForS$ is set to "1st" in the case that the distance is measured from the 1st point of the front/back dichotomy of the RS and to "2nd" if it is measured from the 2nd point of the front/back dichotomy of the RS.

The intersection ("⊕") and composition ("⊗") parts of formulas (5) to (7) are implemented with simplification and propagation CHRs, respectively. For supplying the lack of completeness of the constraint graph, 13 simplification and 21 propagation CHRs have been defined in the CS for positional information [5,7].

3.3 Cardinal Directions

3.3.1 The Representational Model
The representational model for cardinal directions is based on Frank's work [8], with several differences which will be pointed out. Cardinal direction is a binary function from two objects in space (P1, P2) that map onto a symbolic direction. The set of symbolic directions depends on the granularity. Usually humans use two levels of granularity: $d_4 = \{N, E, S, W\}$, and $d_8 = \{N, NE, E, SE, S, SW, W, NW\}$. The information represented is dir(P1,P2), where P1 will refer to the reference object and P2 to the referenced object.

For $d_4$, Frank introduced the identity symbol, O, with the meaning "two points too close each other for a direction to be determined" which increases deductive
power. Besides its original meaning we are going to use $d_s$ with identical symbol to deal with extended objects. When the extension of an object cannot be simplified to a point, it will be delimited by the identity region which also will determine the rest of the cardinal direction regions, as it is shown in figure 7 a). Therefore the identity region is adapted to the size of the extended object.

Given a cardinal direction for a line segment between points P1 and P2, the direction from P2 to P1 can be deduced by using the inverse operation $(\text{inv}(\text{dir}(P1,P2)) = \text{dir}(P2,P1))$ which is implemented in our approach as facts of the PROLOG database, i.e., $\text{inv}(n,s)$.

3.3.2 The BSIP

Given two cardinal direction relationships, between objects P1 and P2 and between objects P2 and P3, the cardinal direction relationship between objects P1 and P3 can be inferred by the BSIP ($\text{dir}(P1,P2) \bowtie \text{dir}(P2,P3) = \text{dir}(P1,P3)$). As example we are going to assume that P1, P2 and P3 are extended objects and $\text{dir}(P1,P2)=\{N\}$ (P2 is to the north of P1) and $\text{dir}(P2,P3)=\{E\}$ (P3 is to the east of P2), therefore we can deduce $\text{dir}(P1,P3)=\{NE\}$ (P3 is to the north-east of P1) as it is shown in figure 7 b). However, it is possible that P2 does not occupy all the north region of P1 (as it happens in figure 7 c), therefore P3 can be to the north-east of P1 or to the north of P1, that is, $\text{dir}(P1,P3)=\{NE,N\}$.

The BSIP for the combination of all the cardinal directions is represented by a table in [8]. This table only reflects the inference for the case in which all the identity regions are of the same size, and contains many lower case values which represent approximate reasoning (for instance "e" might mean the disjunction $\text{NE,N}$ or the relation $\{NE\}$). Variable identity regions (adapted to the size of the extended objects) and exact disjunctive reasoning is represented in our approach [7].

3.3.3 The FIP

Cardinal direction relationships are also considered in our approach as instance of the CSP where the constraints are binary, of the form $C_{ij}$ (i and j refer to point or/and extended spatial objects). Therefore, to compute path consistency we can repeatedly computed operation (1). The intersection ($\bowtie$) and composition ($\bowtie$) parts of formula (1) are implemented with 3 simplification and 3 propagation CHR's, respectively.

It is interesting to notice that the cardinal directions model explained in this section is equivalent to the orientation model defined by [14], therefore the corresponding FIP will be define in a very similar way. Further details can be found in [7].

4 The use of point and extended objects as primitive of reasoning

In this section, a new approach for representing and reasoning, in a uniform way, with point and extended objects as primitive of reasoning, in those spatial models not based on projections, is presented. Constraint Logic Programming instantiated to Finite Domains [13] — CLP(FD) — (which is also included as a library in [2]) has been used as a tool to define the model. This tool permits the definition of variables with a finite and integer domain associated, and linear terms which are linear integer combination of
intended domain variables. The model is based on:

- The definition of a grid with a context dependent size, which is implemented by two finite domain variables (one for each axis X and Y). All the spatial objects are projected onto the grid. Moreover, the infinite number of points belonging to a qualitative spatial region is simplified to a finite and small number of points, i.e., the physical space is discretized.
- The 2-D objects are represented by two finite domain variables, one for each axis of the grid.
- The straight lines which define the RSs are represented by a set of linear terms (an example is given in figure 9). In order to determine the set of linear terms, it is necessary to distinguish the case in which both objects which define the RS are points from the case in which they are extended objects.

The goal of this module is to obtain the set of points belonging to the grid which also belongs to any qualitative region, independently of the fact that the RSs are formed by point or/and extended objects.

5 Application: QNavSim

Traditional methods for robot navigation need a precise and reliable metric of the robot trajectory. These methods provide good results in certain, known and indoor environments where all the work space is perceived from any position. However, they fail in other environments [23]. Qualitative navigation is an alternative way to solve the autonomous robot navigation problem, which tries to make use of uncertain and imprecise spatial information, as human beings do [17] [4] (see [7] for a review of the state of the art in Qualitative Navigation).

In this section, the qualitative spatial model which deals with several spatial aspects in a uniform way, described in the previous sections, is used to build, as demonstrator, a qualitative robot navigation simulator, in a structured environment. The aim is the simulation of the movement of a robot through the structured environment of the city of Castellón, from a starting to a goal position, using a set of qualitative spatial relationships as initial information.

The system architecture proposed consists of two parts: the spatial reasoning module and the Qualitative Navigator Simulator (QNavSim) module. The spatial reasoning module is formed by a set of constraints solvers which manages the different spatial aspects which have been integrated into the same spatial model; and a set of processors of point and extended objects, one for each spatial aspect considered. This module has been explained in the previous sections. In this section, the features of the QNavSim will be introduced.

The grid with context dependent size, introduced in the previous section, is used to define the world model which consists of streets and spatial objects or landmarks. The streets are approximated to intersections of the grid and defined as facts of the PROLOG data base. For instance, the fact street(22,16,23,17) means that there is a portion of the street from the intersection, X=22, Y=16, of the grid, to the neighboring intersection, X=23, Y=17. The name and position of any important landmark are also stored as facts of the PROLOG data base. For instance, the fact position(restaurant,20,17) means that the restaurant is positioned at X=20, Y=17, corresponding to the grid.

The initial information given to the simulated robot is its initial position in terms of the grid, the name of the objective and a set of qualitative spatial relationships. For instance, the bull ring is somewhere to the left of the RS formed by Plaza La Paz and Plaza Santa Clara, and so on. It is possible to draw different cognitive maps which satisfy the initial relationships. In these cognitive maps, the objective might be in different positions—even qualitatively very different—with respect to the initial position. This means that, with the initial spatial relationships, it is possible that the problem of going from a starting position to an objective is not directly solved, i.e. the objective region is not small enough for the robot to directly arrive. Another feature of the simulator is the knowledge acquisition of the robot. When the robot arrives to any intersection of the grid, the PROLOG data base is sought to check if a landmark exists at this position.
In summary, the QNavSim algorithm consists of, first of all, a call to the spatial reasoning module, providing as input the initial set of qualitative spatial relationships. The output is the set of relationships inferred from the initial knowledge. The best relationship between the current position and the goal is used to compute the region where it is possible to find the goal. Then the loop of the algorithm starts. The agent moves one step forward (which corresponds to a grid unit) if there is no crossroad or it chooses as its next step the one which is "closer" to the goal when there is a crossroad. At every new position, the agent checks whether a landmark exists there. If there is no landmark, the program control goes back to the beginning of the loop. If a landmark is perceived at this position, it is checked whether the goal region can be narrowed down by using some inferred spatial relationship, before going back to the beginning of the loop. The algorithm finishes when the robot arrives to the goal. The complete description of the QNavSim algorithm can be found in [7].

An example of the execution of the QNavSim in different steps is shown in figure 9. As the movement of the robot through the streets is simulated, the acquisition of new knowledge will allow the reduction of the objective region, until it will be small enough for the robot to get to the goal.

6 Conclusions and future work

The contributions of the work presented in this paper are the following:
• The definition of an approach for integrating several aspects of space into the same model. For each aspect of space to be integrated we have defined: (1) its representation; (2) the Basic Step of the Inference Process (BSIP); and (3) the Full Inference Process (FIP). Constraint Logic Programming extended with Constraint Handling Rules (CLP+CHR) provides the suitable level of abstraction for the definition of incremental, flexible and general-purpose user-defined constraint solver (CS) which solves in a straightforward way the FIP for each aspect of space to be integrated. Another important feature of these CSs are their efficiency —the complexity of each CS is $O(n^3)$ where $n$ is the number of spatial landmarks considering in the reasoning process—. Qualitative orientation, positional information and cardinal directions have been integrated into the same qualitative spatial reasoning model in this paper. However, we argue that this approach can also be used to integrate further aspects into the same model, for instance space and time which will permit to qualitatively deal with the concept of motion.
• Constraint Logic Programming instantiated to finite domains CLP(FD) has been used as tool to deal with point and extended objects as primitive of reasoning for spatial models not based on projections (which are considered more cognitive). Finite domain variables and linear terms are suitable for representing spatial objects and the straight lines which describe the reference systems, respectively.
• The development of a Qualitative Navigation Simulator (QNavSim) where the movement of a robot through the structured environment of the city of Castellón is simulated. It is assumed that the robot does not know the city nor has a map. Its aim is to arrive to a goal, using a set of qualitative spatial relationships which contains a lot of imprecision. The spatial reasoning module will obtain the set of relationships which can be inferred from the initial information. The use of these inferred relationships and the acquisition of new landmarks while the robot is moving, will allow further reduction of the goal region.

In this paper, an interpretation of qualitative spatial labels like "right-front" or "close" has been used for a qualitative navigation simulator. The interpretation of
the reasoning process for a real mobile robot will be very different. This is our current interest.

References


A Qualitative Agent Approach For Assessing Dynamic Process Human Supervision

LAAS-CNRS & LEA-SICA  MA2-UPC & LEA-SICA  ESADE-URL & LEA-SICA  INSERM U455
7, Av. Colonel Roche  Pau Gargallo, 5  Av. Pedralbes, 62  Service de Neurologie.
31077 Toulouse Cedex  08028 Barcelona  08034 Barcelona  CHU Purpan
louise@laas.fr  {prats, monicas}  agell@esade.es  josette@purpan.inserm.fr

Abstract

The WAHRPS (Worlds for Assessing Human Reasoning in Process Supervision) project aims at assessing the reasoning of patients suffering from a frontal syndrome. This patients present deficiencies in their reasoning about dynamic features, sometimes in a similar way as those reported for industrial operators in stress or fatigue situations. The methodology is based on comparing their performances to those of various reference Reasoning Artificial Operators, or Agents, when performing the supervision of a dynamic micro-world. This paper presents the concepts underlying the design of a qualitative agent as well as a method for comparing the behaviour of a human operator with the one proposed by the artificial agent.

Key words: Supervision Reasoning Assessment, Qualitative Reasoning, Micro-Worlds, Dynamic Systems.

1 Introduction

Understanding and assessing human reasoning about dynamic situations is a highly complex task, which is central to several scientific areas such as neuropsychology, ergonomics and artificial intelligence. A typical case is the supervision of dynamic systems, for instance in industrial production plants environments or in the air traffic control domain [3]. Several accidents occurred during the last decade have pointed out the increasing need for better understanding the different cognitive components of human supervision reasoning – anticipation, diagnosis, decision making, etc. – as well as the human-machine interface impact.

This has a link with the medical domain as it has been noticed that some human errors arising from information overload, stress or fatigue in the process supervision domain [11] can also be observed on patients suffering from Parkinson Disease [1]. As a matter of fact, some parkinsonian patients present a “frontal-type syndrome”, which is suspected of impairing many cognitive components involved in their reasoning about dynamic situations. Whereas direct observation of the operators at work is one of the most commonly used method in the human factors and knowledge engineering communities, it is admitted that experimental environments based on simulators or micro-worlds [5] offer significant advantages such as the reproducibility of the experiments.

The WAHRPS (Worlds for Assessing Human Reasoning in Process Supervision) project is a project conducted by INSERM U455, which aims at assessing the reasoning of patients suffering from a frontal syndrome. The methodology is based on comparing their performances to those of various reference Reasoning Artificial Operators, or Agents, when performing the supervision of a dynamic micro-world [9]. A user-friendly and powerful micro-world shell has been specifically designed for this purpose [7]. The aim of this paper is to present the concepts underlying the design of a naïve qualitative agent [10] as well as a method for comparing the behaviour (i.e. the sequence of actions) of the patients that undergo the test with the one proposed by the artificial agent.

2 The micro-world shell

The WAHRPS environment includes a generator of micro-worlds that allows one to define and build dynamic micro-worlds and a software that simulates
their behaviour [7]. The micro-worlds are in the form of waterworks which may undertake all the characteristics of complex dynamic systems, such as decomposability and non-linearity.

A WAHRPS micro-world is composed of a set of tanks, linked by a set of pipes, each of which may or may not include a binary action valve. The micro-worlds fulfill several important requirements:

- The micro-worlds physics domain (hydraulics) is sufficiently grounded in everyday life concepts (gravity flowing water) so as to allow one to test patients;
- It is of sufficient "industrial realism" so as to be accepted by industrial operators;
- It allows one to build highly flexible micro-world structures, which may easily vary in the type of situation and their complexity.

2.1 Case study: the micro-world configuration

A micro-world is defined by the following parameters:

- Number of tanks: \( n \).
- For each tank \( T_i \), \( i = 1, \ldots, n \), \( V_i \) represents its capacity, \( H_i \) its height, and \( W_i \) its width.
- All the tanks are assumed to have the same depth, \( D_i = D, i = 1, \ldots, n \).
- For each pipe \( P_{ij} \), \( T_i \) represents the upstream tank, \( T_j \) the downstream tank, \( H_{ij} \) the height, \( L_{ij} \) the "diagonal" length (i.e., the distance between the bottom of \( T_i \) and the top of \( T_j \)), \( W_{ij} \) the mean width and \( B_{ij} \) a constant that depends on the pipe \( P_{ij} \) and on the gravitation constant \( g \).
- If a valve controls the pipe \( P_{ij} \), flow, the valve is denoted by \( V_{ij} \).

By assumption, a pair of tanks \( T_i \) and \( T_j \) cannot be connected by more than one pipe.

Two tanks must be differentiated from the other ones: the source tank, always referred to as \( T_1 \), which is at the very top of the configuration and the sink tank, always referred to as \( T_n \), which is at the very bottom.

Both are set to have the same capacity. The other tanks are referred to as the intermediary level tanks.

At the beginning of a session all the tanks are empty but the source tank \( T_1 \). The instruction given to the operator, called THE INSTRUCTION in the following, is that he/she must convey the maximum of water from the source tank \( T_1 \) to the sink tank \( T_n \) hence avoiding the intermediary tanks to overflow, in minimum time by acting on the binary valves controlling the pipes. Water propagates through the pipes under the effect of gravity, the propagation being only constrained by the closing of the ON/OFF valves.

The set of top tanks, which evolves according to the dynamics of the system, is defined as the set of tanks whose upstream tanks are all empty but still contain water themselves.

A micro-world configuration instance at a given time of a test session is given in Figure 1. It will be referred to this particular micro-world configuration all along the paper to illustrate the proposed concepts, explanations, and ideas.

![Fig. 1: A micro-world configuration](image)

2.2 Formal representation of the physical system

The physical tank configuration system and the flowing processes occurring in it can be formally represented by an oriented graph \( G = (V, A) \). The set of vertices \( V = \{ T_1, \ldots, T_n \} \) correspond to the tanks and the set of arcs \( A = \{(T_i, T_j) \mid T_i, T_j \in V \} \) and there exists a pipe \( P_{ij} \) from \( T_i \) to \( T_j \) if and only if \( T_i \) is the source tank and \( T_j \) is the sink tank. For sake of clarity, the arc \( (T_i, T_j) \) is hence denoted by \( P_{ij} \). The arcs of \( G \) are oriented in such a way that the source tank \( T_1 \) and the sink tank \( T_n \) are the source and the sink of the graph respectively.

Weights 0 or 1 are associated to each arc of \( G \), depending on whether the corresponding valve is closed or open, respectively. If a pipe includes no valve, its weight is always 1. The arcs with weight 1 (weight 0) are graphically represented by continuous lines (discontinuous lines) (cf. Fig. 2).

The state of the system is assessed within a linear temporal scale provided by a logical clock. The arcs weight values evolve in time as the operator acts on the valves.

A path in \( G \) is defined as a sequence \( (P_{i1}, P_{i2}, \ldots, P_{in}) \) of arcs. It starts from tank \( T_i \) and ends at tank \( T_n \).
At some time point, an open path in $G$ is defined as a path such that all arcs have weight 1. Figure 2 represents the oriented graph corresponding to the system in Figure 1.

![Fig. 2: The oriented graph](image)

3 A Naïve Qualitative Agent (NQA)

Three specific agents, based on anticipating the process behaviour by means of a simulation and on choosing the best action according to a three steps ahead prediction strategy, have already been built [8] [9]. The first one, referred as the numeric agent, uses a numerical fluid mechanics model. The other two were implemented using the causal simulator Ca-En [2]. These two later ones differ in that the first is based on a semi-qualitative version of the fluid mechanics model whereas the second relies on abstracted qualitative laws. The action choice strategy underlying the three agents was shown to be similar to that used by some humans [9].

On the short-term, prediction of the system's future states by human subjects has been observed to be as precise as the numeric agent and certainly much better than the (semi)-qualitative simulation based agents. This does not mean that humans really compute numerically the states. More probably, their performance is due to a kind of "perceptual prediction" : the water heights in the tanks being continuously displayed, humans may anticipate the immediate future heights. Qualitative simulation of the states cannot therefore compete with humans at this level. However, it is known that human brain shows an extraordinary ability to categorise in order to perform the most rapid and efficient reasoning. We assume therefore that they draw from the predicted precise states the qualitative information that is accurate for the problem, i.e. "levels of danger" (alarms). This paper proposes an artificial agent built along these lines: starting from a numeric perception, it abstracts this information into qualitative concepts that are processed with qualitative reasoning techniques [10]. The primary perceptions that a human being undergoes when facing up a WAHRPS micro-world are the heights of water in the tanks and their tendencies, i.e. whether they are increasing, decreasing or steady. Our assumption is that these are the main factors that a human being takes into account for deciding about an action to perform. This agent, referred as the Naïve Qualitative Agent (NQA), is to be added to the existing agents to constitute an initial library of artificial operators.

Human operators have mostly a qualitative representation of the world, elaborating generally on a causal mental representation of the process at hand [Kirwan 92]. On the other hand, psychologists commonly agree on the fact that anticipation is a crucial aspect in process supervision reasoning [Cellier 91]. The operators are hence suspected to use the causal mental model to perform predictions that are qualitative in nature. The NQA also implements these observations.

3.1 The agent perception model

According to the precedent considerations, a qualitative representation of the height of water in the tanks and the tendency of these heights is introduced by means of two qualitative variables associated to each tank $T_i$, $i=1,...,n$, at each instant $t$:

1. The height of water $h_i(t)$ may take four possible qualitative values:
   - $h_i(t)$ is EMPTY (0) when $h_i(t) = 0$.
   - $h_i(t)$ is LOW if it is below a given threshold $\alpha_i$, which specifies a criticality level and is used to trigger an alarm. This alarm is an anticipatory indicator in the control strategy (cf. Section 3.2). The alarm threshold is determined such that the volume of water admissible in tank $T_i$ before it overflows, i.e. $V_i - h_i(t)W_iD$, equals a given constant $K$, the same for all tanks. The value of the landmark $\alpha_i$ is hence given by $V_i - \alpha_iW_iD = H_iW_iD - \alpha_iW_iD = K$, therefore $\alpha_i = H_i - \frac{K}{W_iD}$.

This way of setting the alarm thresholds guarantees that the alarms correspond to the same level of criticality for all tanks. It indeed captures the natural perception of the width of the tanks, which is definitely involved in the human operator decision making process.
   - $h_i(t)$ is HIGH when $\alpha_i \leq h_i(t) < H_i$. 


- \( h_i(t) \) is FULL when \( h_i(t) = H_f \)

\[ \begin{align*}
\text{EMPTY} & \quad \text{LOW} & \quad \text{HIGH} & \quad \text{FULL} \\
0 & \quad \alpha_i & \quad \beta_i & \quad \gamma_i 
\end{align*} \]

Fig. 3: Qualitative values of height

2. The tendency of the water height in the tank, \( \partial h_i(t) \), may take three possible qualitative values: \( \text{inc} \), \( \text{dec} \) and \( \text{std} \) (meaning “increasing”, “decreasing” and “steady” respectively):
- If \( h_i(t) - h_i(t-1) > 0 \), then \( \partial h_i(t) = \text{inc} \).
- If \( h_i(t) - h_i(t-1) < 0 \), then \( \partial h_i(t) = \text{dec} \).
- If \( h_i(t) - h_i(t-1) = 0 \), then \( \partial h_i(t) = \text{std} \).

At each instant \( t \), each valve has an associated value 1 or 0, depending on whether it is open or closed, respectively.

3.2 Control strategy

At each instant \( t \), a tank \( T_i \) is said to be alarming (Al) if its height of water \( h_i(t) \) is HIGH or FULL, and its tendency \( \partial h_i(t) = \text{inc} \).

The intuition advises to distinguish two cases: the case with alarms in which there is one or several tanks overflowing or about to overflow (i.e. there are alarming tanks) and the case without alarms (i.e. no alarming tanks). In each of these two cases, the goals are different, and so are the corresponding strategies that the NQA carries out.

3.2.1 Case without alarms

3.2.1.1 Goals

When there are no alarms, the main objective is to accelerate the process, i.e., to transport the maximum quantity of water from the top tanks to tank \( T_n \) at each instant. This objective is a direct answer to the minimum time requirement of THE INSTRUCTION and can be interpreted in practical terms by the following instructions:

1. Achieve and maintain an increasing tendency for the sink tank \( T_n \) i.e. \( \partial h_n(t) = \text{inc} \).
2. Increase at the most the number of open paths from the top tanks to the sink tank \( T_n \).

The first instruction aims at avoiding the situation in which there are no open paths. The second instruction expresses the acceleration goal.

The action to be performed is hence chosen according to the number of supplementary paths that it opens. If two actions have the same consequence to this respect, a criterion based on the maximal flow allowed by the new open paths is applied. This is described in more details in the next section.

3.2.1.2 Method

The method uses the graph \( G \) and the weights associated to its arcs. At each instant \( t \) with no alarm:

Step 1: Compute the paths starting from a top tank and ending tank \( T_n \):
- If \( t = 0 \), compute the set of all possible paths starting from the source tank \( T_i \) and ending at the sink tank \( T_n \). This set is called \( \mathcal{P}_0 \).
- If \( t = k \neq 0 \), update the paths in the set \( \mathcal{P}_{k-1} \) as follows: the path \((P_{i1}, P_{i2}, ..., P_{i\text{dir}}) \in \mathcal{P}_{k-1} \) is updated by removing the head sub-path \((P_{i1}, P_{i2}, ..., P_{i\text{dir}}[z \leq i]) \), corresponding to EMPTY tanks.

*/As \( T_i \) and the following tanks of the oriented graph are getting EMPTY, the paths must be updated so as to start at the first not empty tank, i.e. a top tank/*

Step 2: Label the paths in \( \mathcal{P}_k \) with the number of arcs having a weight equal to 0.
*/This number corresponds to the number of actions to be performed by opening the path/*

Step 3: Determine the action to be performed.

1. If all the paths are open, the action of the NQA is to do nothing.
2. Otherwise, consider the paths with a minimum label.
- When there is only one path, the action of the NQA is to open the closed valve of this path which is the closest to \( T_n \).
- When there are more than one path, the NQA uses the "min-max criterion" given below in order to

\[ \text{An alternative strategy is to open the closed valve closest to the top tank of the path. This option is more risky, in the sense that it may result in alarming situations more often than the chosen one.} \]
select one of them; the action of the NQA is to open the closed valve of this path that is the closest to \( T_n \).

**Min-max criterion:** Given a set of paths \( p_i \), \( i=1,...,k \), compute for each \( p_i \) the minimum width of its pipes (min-width); select the path that has the maximum (min-width) value.

### 3.2.2 Case with alarms

#### 3.2.2.1 Goals

When one or several alarms are active, the main objective is to come back to a non-alarming situation; nevertheless, the general objective of accelerating the process is maintained. Therefore the instructions are the following, ordered by importance:

1. Do not enlarge any alarm.
2. Reduce at the most the number of alarms.
3. Maintain the tank \( T_n \) increasing.
4. Increase at the most the number of open paths to tank \( T_n \).

The first instruction means that when a tank is alarming, the operator is required not to open (close) any valve containing an upstream (downstream) pipe. The other three instructions have clear aims.

#### 3.2.2.2 Method

The given method is based on a qualitative one-step ahead prediction after the computation of all the possible actions which may remove alarms.

**Step 1:** Computation of the set of possible actions:
For each alarm \( A_l \) (alarming tank \( T_l \)) compute all the possible actions that may remove \( A_l \). These actions are: to close the valve of any pipe arriving at tank \( T_l \), or to open the valve of any pipe going out from tank \( T_l \).

**Step 2:** Qualitative one-step ahead prediction:
For each possible action, a one-step ahead qualitative prediction is performed on the bases of the graph \( G \).

---

It could also be considered the opening or closing of a valve more distant from the alarming tank, but the inertia of the system makes this type of action much less intuitive than the chosen ones. Hence they are not considered.

Also note that this naive strategy does not take into account the case in which none of the proposed actions is possible due to the absence of valves.

---

That can be viewed as a representation of the causal influences underlying the flow processes.

An action on a valve \( V_{ij} \) influences a subset of tanks whose vertices define a sub-graph of \( G \), say \( G_{V_{ij}} \). The vertices of this sub-graph are \( T_i \) and \( T_j \), together with all tanks of the open paths starting from \( T_j \). Then, the prediction consists in:

1. **Adding a qualitative descriptor (\( i \) or \( c \))** to the label of tendency (\( inc \) or \( dec \)) of the tanks in \( G_{V_{ij}} \) if the label is not steady, otherwise changing the \( std \) label to \( inc \) or \( dec \) in the following way:

| \( inc \rightarrow inc^c \) | \( inc \rightarrow inc^c \) |
| \( dec \rightarrow dec^c \) | \( dec \rightarrow dec^c \) |
| \( std \rightarrow inc \) | \( std \rightarrow dec \) |

**Table 1:** Opening a valve

1. **Closing a valve** \( V_{ij} \):

| \( inc \rightarrow inc^c \) | \( inc \rightarrow inc^c \) |
| \( dec \rightarrow dec^c \) | \( dec \rightarrow dec^c \) |
| \( std \rightarrow inc \) | \( std \rightarrow dec \) |

**Table 2:** Closing a valve

These increased labels correspond to the following intuitive idea, given through an example: if a valve that is upstream an increasing tank is opened, this tank increases even more, hence the new label \( inc^c \); if it is closed, the height of water in the tank increases less, hence the new label \( inc^c \). The other ten cases can be explained in the same way.

2. **Adding a qualitative descriptor (\( i \) or \( c \))** to the label \( AI \) of the alarming tanks in the sub-graph. When the action consists in opening the valve of a pipe upstream the tank in alarm, or closing the valve of a downstream pipe, the alarm is obviously enlarged. In these cases the new label is \( AI \). In the opposite cases, the new label \( inc^c \) could be \( std \) as well but the qualitative nature of the prediction does not allow one to distinguish the two cases.
is $A^t$ (in certain cases, $A^t$ may mean that the alarm is eliminated):

<table>
<thead>
<tr>
<th>Opening a valve $V_j$</th>
<th>Other tanks in $G_{V_j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tank $T_i$ (upstream the valve)</td>
<td>$A^t$ (downstream)</td>
</tr>
<tr>
<td>$AI \rightarrow A^t$</td>
<td>$AI \rightarrow A^t$</td>
</tr>
</tbody>
</table>

Table 3: Opening a valve

<table>
<thead>
<tr>
<th>Closing a valve</th>
<th>Other tanks in $G_{V_j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tank $T_i$ (upstream the valve)</td>
<td>$A^t$ (downstream)</td>
</tr>
<tr>
<td>$AI \rightarrow A^t$</td>
<td>$AI \rightarrow A^t$</td>
</tr>
</tbody>
</table>

Table 4: Closing a valve

The following example shows the process of this one-step prediction for a particular case with two alarming tanks. It is considered the micro-world given in Fig. 1 and that at instant $t$ tanks $T_1$ and $T_2$ are alarming, as represented by the graph in Fig. 4.

Fig. 4: Two alarming tanks

The possible actions that may remove $AI_3$ are: to close valve $V_{13}$, to open $V_{34}$ or to open $V_{23}$, and the only possible action for removing $AI_4$ is to close valve $V_{24}$. Then, for each one of these four possible actions, the one-step qualitative prediction is performed in the following way:

Fig. 5: Closing $V_{13}$

In each case, the influenced sub-graph is encircled by a dotted line.

Step 2: Choice of the action to be performed:

The possible actions are evaluated according to the goals on the basis of the predictions of step 2.

Four grades $G_1$, $G_2$, $G_3$ and $G_4$, corresponding to the instructions 1, 2, 3 and 4 given in 3.2.2.1 are assigned to every possible action, although the action consists in doing nothing. Positive grades represent an improvement, and negative ones represent deterioration of the situation.

1. If the action generates $n$ alarms $A^t$, $n \geq 0$, then $G = -n$. 
2. If the number of alarms that have been labelled with AI is \( n \), and the number of new alarms AI is \( m \), then \( G_e = n - m \).
3. If the state of tank \( T_\alpha \) is \( \text{inc}^\ast \), then \( G_f = 2 \). If the state of tank \( T_\alpha \) is \( \text{inc or inc}^\ast \), then \( G_f = 1 \). If the state of tank \( T_\alpha \) is \( \text{std} \), then \( G_f = 0 \).
4. If the number of open paths to the sink tank has been increased in \( n \), then \( G_e = n \). If it has been decreased in \( n \), then \( G_e = -n \). If it has remained constant, then \( G_e = 0 \).

After these grading, there are two different ways for the NQA to choose the action to be performed. In both, the order of importance of the goals is taken into account. However, in the first option only the order of importance is considered whereas in the second option each goal is associated a weight and the four goals are evaluated as a whole by means of a weighted sum.

**OPTION 1**

Every action is characterised by the 4-tuple \((G_b, G_d, G_p, G_h)\); the set of all 4-tuples is ordered in the lexicographical order, and the greatest one is chosen. In case of ambiguity, the NQA chooses randomly any of the actions with a maximal 4-tuple.

**OPTION 2**

Weights \( p_b, p_d, p_p, p_h \) such that \( p_1 > p_2 > p_3 > p_4 > 0 \) and normalised such that \( p_1 + p_2 + p_3 + p_4 = 1 \), are respectively assigned to each goal and the weighted sum \( p_b * G_b + p_d * G_d + p_p * G_p + p_h * G_h \) is computed for every action. The NQA chooses the action that obtains the greatest weighted sum.

In case of ambiguity, the NQA chooses randomly any of the actions with equal weights.

In the above example, the four grades \( G_b, G_d, G_p, G_h \) assigned to every possible action are the following:
- Closing \( V_{13} \): \( G_1 = 0, G_2 = 1, G_3 = 1, G_4 = 0 \).
- Opening \( V_{23} \): \( G_1 = 1, G_2 = 2, G_3 = 1, G_4 = 0 \).
- Opening \( V_{35} \): \( G_1 = 0, G_2 = 1, G_3 = 1, G_4 = 0 \).

In OPTION 1 every action is characterised by: Closing \( V_{13} \) \( \leftrightarrow \) \((0,1,1,0)\), Opening \( V_{34} \) \( \leftrightarrow \) \((-1,1,2,1)\), Opening \( V_{15} \) \( \leftrightarrow \) \((0,1,2,1)\), Closing \( V_{24} \) \( \leftrightarrow \) \((0,1,1,-1)\). They are ordered in the following way: \((-1,1,2,1) \leq (0,1,1,-1) \leq (0,1,1,0) \leq (0,1,2,1)\). Hence the NQA chooses the greatest one, that corresponds to Opening \( V_{34} \).

In OPTION 2, with weights \( p_1 = 0.4, p_2 = 0.3, p_3 = 0.2, p_4 = 0.1 \), every action is characterised by the sums: Closing \( V_{13} \) \( \leftrightarrow \) 0.5, Opening \( V_{34} \) \( \leftrightarrow \) 0.4, Opening \( V_{35} \) \( \leftrightarrow \) 0.8, and Closing \( V_{24} \) \( \leftrightarrow \) 0.4. Hence the NQA chooses the action that obtains the greatest sum, i.e. Opening \( V_{34} \).

4 Refining the agent perception and reasoning (QA)

A more refined way, though still qualitative, for the computation of the "tendencies of the water height" of each tank is presented in this section. The resulting qualitative agent (QA) is able to quantify the tendencies and to account for alarm levels, depending on the relation between the widths of the open pipes arriving to a tank and those going.

The advantage of the QA with respect to the former NQA is that it is more accurate without refining many much more computations. The kind of quantification based on the pipes width is obviously an aspect of the reasoning involved in humans decision making.

4.1 The refined perception model

As in the NQA, at each instant \( t \), two qualitative variables associated to each tank \( T_\alpha \) are considered:
1. The height of water \( h(t) \) is the same as in NQA. It may take four qualitative values: \( \text{EMPTY (0)}, \text{LOW}, \text{HIGH}, \text{FULL} \) (or equal to the total height \( H_\alpha \) of the tank \( T_\alpha \)).
2. The tendency of the water height in the tank: \( \partial h(t) \), may now take five qualitative values: \( \text{incL}, \text{incS}, \text{decL}, \text{decS} \) and \( \text{std} \) (meaning "increasing a lot", "increasing slightly", "decreasing a lot", "decreasing slightly", and "steady" respectively).

They are defined in the following way:

Let \( W_{\alpha}, W_{\alpha}^l, W_{\alpha}^m, ..., W_{\alpha}^r \) be the widths of the open pipes arriving to tank \( T_\alpha \) and \( W_{\alpha}, W_{\alpha}^l, W_{\alpha}^m, ..., W_{\alpha}^r \) the widths of the open pipes going out from it at instant \( t \), and consider the quotient:

\[ Q(t) = \frac{W_{\alpha}^l + W_{\alpha}^m + ... + W_{\alpha}^r}{W_{\alpha}^l + W_{\alpha}^m + ... + W_{\alpha}^r}, \]

that gives the relation between these two sums. The possible values of these quotients determine the five possible qualitative values of the tendency \( \partial h(t) \) of the water height in the tank \( T_\alpha \) as given in the following table:

<table>
<thead>
<tr>
<th>( Q(t) )</th>
<th>( Q(t) \geq 2 )</th>
<th>( 1 &lt; Q(t) &lt; 3 )</th>
<th>( Q(t) \leq 1/3 )</th>
<th>( \frac{1}{3} &lt; Q(t) &lt; 1 )</th>
<th>( Q(t) = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( h(t) = \text{incL} )</td>
<td>( h(t) = \text{incS} )</td>
<td>( h(t) = \text{decL} )</td>
<td>( h(t) = \text{decS} )</td>
<td>( h(t) = \text{std} )</td>
</tr>
</tbody>
</table>

Table 5: Qualitative values of the tendency
If \( Q(t) > 1 \), there is more water going into the tank than going out from the tank, so the height of water in the tank is increasing. It seems reasonable to assume that a human being is able to distinguish the pipes total widths when they are in a proportion above 3. Therefore, 3 is taken as a threshold for the two different levels for increasing, incL and incS. If \( Q(t) < 1 \), the height of water in the tank is decreasing. Following a similar reasoning as before, two different levels for decreasing, decL and decS, are considered for a threshold at \( 1/3 \).

If \( Q(t) = 1 \), the assigned label is std.

As in the NQA, at each instant \( t \), each valve has an associated value 0 or 1, depending on whether it is closed or open, respectively.

### 4.2 The refined control strategy

As the NQA, the QA distinguishes two cases: with or without alarms.

At each instant \( t \), a tank \( T_i \) is said to be \textit{ alarming} when its height of water \( h_i(t) \) is HIGH or FULL, and its tendency \( \partial h_i(t) \) is either incL or incS. It is said to be \textit{largely alarming} in the first case and \textit{smoothly alarming} in the latter case.

When there are no alarms, both operators NQA and QA have the same strategy for deciding at each instant which action to perform. Nevertheless, the reasoning in the case of alarms is more accurate in QA because of the consideration of more refined levels of alarm.

#### 4.2.1 Case with alarms

##### 4.2.1.1 Goals

As in the case of the agent NQA, the instructions of QA are the following, ordered by importance:

1. \textit{Do not generate large alarms.}
2. \textit{Reduce the number of alarms.}
3. \textit{Maintain the tank \( T_i \) increasing.}
4. \textit{Increase at the most the number of open paths to tank \( T_w \).}

These are the same instructions as for NQA, except for the first one, which is now formulated by means of the levels of alarms.

##### 4.2.1.2 Method

The method remains similar to the one for the NQA in its principles. It includes the same steps.

\textbf{Step 1:}\hspace{1cm} Computation of the set of possible actions (identical to NQA):

For each alarm \( Al \) (alarming tank \( T_j \)), compute all the possible actions that may remove \( Al \). These actions are: to close the valve of any pipe arriving at tank \( T_i \), or to open the valve of any pipe going out from tank \( T_j \).

\textbf{Step 2:} Qualitative one-step ahead prediction:

For each possible action, a one-step ahead qualitative prediction is performed on the bases of the graph \( G \) that can be viewed as a representation of the causal influences underlying the flow processes.

An action on a valve \( V_j \) influences a subset of tanks whose vertices define a sub-graph of \( G \), say \( G_{V_j} \). The vertices of this sub-graph are \( T_i \) and \( T_j \), together with all tanks of the open paths starting from \( T_j \). Then, the prediction consists in:

1. \hspace{1cm} Computing the new quotients \( Q(t+1) \) that would result at the instant \( t+1 \) from doing the action for each one of the tanks \( T_i \) in the mentioned sub-graph.
2. \hspace{1cm} To assign new different labels of tendency to each tank in the sub-graph, by means of these new values \( Q(t+1) \). In this way, the label of tendency of a tank that, for example, at instant \( t \) was incS, at instant \( t+1 \) can become into any of the possible labels incS, incL, decS, decL, or std.
3. \hspace{1cm} To assign new different labels of alarm by means of the new labels of tendency to each tank \( T_i \) in the mentioned sub-graph; perhaps, some alarms have disappeared and some have appeared.

\textbf{Step 3:} Evaluation of the possible actions

The possible actions are evaluated according to the goals on the basis of the predictions in order to choose one of them.

Four grades \( G_1 \), \( G_2 \), \( G_3 \) and \( G_4 \), corresponding to the instructions 1, 2, 3 and 4 given in 4.2.2.1 are assigned to every possible action:

1. If the action generates \( m_i \) \( AL \), n \( \geq 0 \), then \( G_1 = -m_i \).
2. If the number of \( AL \) that have been eliminated is \( n_1 \), the number of new \( AL \) is \( m_i \), the number of \( AL \) that have been eliminated is \( n_2 \), and the number of new \( AL \) is \( m_2 \), then \( G_2 = n_1 + \frac{1}{2} n_2 - m_i - m_2 \).

These grades are assigned to each possible action.
3. If the state of tank \( T_n \) is \( \text{inc}L \), then \( G_1 = 2 \). If the state of tank \( T_n \) is \( \text{inc}S \), then \( G_2 = 1 \). If the state of tank \( T_n \) is \( \text{std} \) then \( G_3 = 0 \).

4. If the number of open paths to the sink tank has been increased in \( m \), then \( G_4 = m \). If it has been decreased in \( m \), then \( G_4 = -m \). If it has remained constant, then \( G_4 = 0 \).

After these grading, the same two different ways for the NQA to choose the action to be performed are used for the QA.

5 Comparing the agents with the human actions

The comparison is carried out on the basis of the sequence of actions performed by the human operator, performing the comparison at each sample instant. The absence of action is considered as a "no-action" action.

The test session is organised so that, at each time instant, the human and the artificial operator make a decision on the next control action and these actions can be compared directly. Each time instant, hence, provides an independent experiment sample.

The human operator action is always executed.

The comparison is performed on every sample, and the final evaluation of how similar the reasoning of the two operators is can be obtained statistically on the experiment sample set.

The evaluation criteria must capture "how well" the instructions used in the decision making process are fulfilled by the performed action.

For an action performed at instant \( t \), these criteria are hence the following:

1. (Only for the QA) Difference in the number of large alarms \( \text{ALL} \) between \( t \) and \( t+1 \), denoted by \( C_t \).
2. Difference in the number of alarms (including \( \text{ALL} \) and \( \text{ALL} \)) for the QA between \( t \) and \( t+1 \), denoted by \( C_t \).
3. Tendency of the height of tank \( T_n \) at \( t+1 \).
4. Difference in the number of open paths to tank \( T_n \) between \( t \) and \( t+1 \), denoted by \( C_t \).

Four grades \( G_1, G_2, G_3, G_4 \) are defined:

\( G_1 = C_t \) for \( t = 1, 2, 4 \) and

\( G_2 = 2 \) if \( \Delta h_d(t+1) = \text{inc} \) \( \Delta h_d(t+1) = \text{inc L} \) for the QA,

\( G_3 = 1 \) if \( \Delta h_d(t+1) = \text{inc} \) or \( \text{inc} \) \( \Delta h_d(t+1) = \text{inc S} \) for the QA,

\( G_4 = 0 \) if \( \Delta h_d(t+1) = \text{std} \).

The actions are evaluated by a weighted sum, using the same weights as in section 3.3.2.2:

\[ G(t) = p_1G_1 + p_2G_2 + p_3G_3 + p_4G_4 \]

Notice that for each action, the grades \( G_1, G_2, G_3, G_4 \) take their values in the following sets:

- \(-n-2 \leq G_1 \leq 0\)
- \(-n-2 \leq G_2 \leq n-2\)
- \(G_3 \in \{0, 1, 2\}\)
- \(0 \leq G_4 \leq n\).

where \( n, n \geq 3 \), is the total number of tanks and \( n \) is the maximum number of paths ending at \( T_n \), that can be opened all at once opening a valve in the particular configuration of the micro-world. Hence \( G(t) \) takes its value from \( x = -(n-2)(p_1+p_2) \) to \( y = p_3(n-2)+p_4+2p_3 \).

The difference \( G_0(t)-G_0(t) = d_0(t) \) (for the agent, \( G_H \) for the human), is computed for each instant from \( t = 0 \) to \( t=t_f \) where \( t_f \) is the instant defining the end of the test session (there is no more water in the intermediary tanks). Hence, a vector \( D = (D_0, \ldots, D_m) \) is obtained. Each component of this vector takes values from \( x-y \) to \( y-x \). There are two natural ways of measuring the similarity between the performances of the human operator and the reference artificial operator, both in the form of a norm of the \( (1+t)\)-dimensional vector \( D \):

\[ d_1 = \sqrt{\frac{\sum_{t=0}^{n} (D_t)^2}{t+1}} \quad d_2 = \frac{\sum_{t=0}^{n} |D_t|}{t+1} \]

These two distances come from two classical norms (the first one, from the Euclidean norm) in \( R^{n}\). They evaluate a type of mean value over the \( t_f \) experiment samples.

Although these two distances are different, both of them vary from 0 to \( y-x \). When \( d_1 = d_2 = 0 \) all the actions of the human operator coincide with the actions of the artificial agent, i.e. they have identical behaviour.

In order to interpret the results, the interval \([0, y-x] \) is split up into four sub-intervals corresponding to the situations "very similar behaviour", "similar behaviour", "different behaviour", and "very different behaviour", such as it is shown in Figure 8:

\[ 0 \quad \text{very similar} \quad \text{similar} \quad \text{different} \quad \text{very different} \quad y-x \]

where the thresholds \( a_1, a_2, a_3, a_4 \) can be \((y-x)/4, (y-x)/2 \) and \((y-x)/4, \) or other values between 0 and \( y-x \) chosen by the user.
CONCLUSION

This paper presents an on-going work, which provides the concepts and decision strategies for implementing qualitative artificial agents to be used as reference agents for assessing human reasoning in the process supervision domain. It is intended to contribute to the WAHRPS project, which is conducted by INSERM U455, in the medical domain for testing parkinsonian patients.

The paper builds on the observations that several concepts coming from the qualitative reasoning area of Artificial Intelligence match cognitive features outlined by psychologists and ergonomics about the way human operators perform the supervision task (causal mental models, qualitative anticipation, etc.)

It is our opinion that this research direction is a promising perspective and that a lot of work still needs to be done.

The proposed agents (NQA and QA) have not been fully implemented yet. This will be the next step of our work, followed by a series of tests on human patients to be undertaken by our medical experts.

REFERENCES


Modal Interval Analysis for error-bounded semiqualitative simulation

Joaquim Armengol\textsuperscript{1}, Louise Travé-Massuyès\textsuperscript{2}, Josep Vehí\textsuperscript{3}, Miguel Ángel Sáinz\textsuperscript{3}

\textsuperscript{1}Institut d'Informàtica i Aplicacions. Universitat de Girona & LEA-SICA
Campus Montilivi. E-17071 Girona. E-mail:{armengol,vehi}@eia.udg.es

\textsuperscript{2}LAAS-CNRS & LEA-SICA. 7 Avenue du Colonel-Roche. F-31077 Toulouse. E-mail:louise@laas.fr

\textsuperscript{3}Departament d'Informàtica i Matemàtica Aplicada. Universitat de Girona.
Campus Montilivi. E-17071 Girona. E-mail:sainz@sina.udg.es

Abstract

Imprecision and uncertainty in systems can often be expressed with interval models. The result of the simulation of these models can be represented in the form of envelope trajectories. These envelopes can be characterised by several properties such as completeness and soundness which lead to the concept of overbounded and underbounded envelopes. Simulation of such interval models can be performed by several means including quantitative, qualitative and semiqualitative techniques. Whereas existing simulators do not provide any information about the "error" with respect to the exact envelope, a method to obtain error-known envelopes is proposed. It is based on the simultaneous computation of an underbounded and an overbounded envelope. Both envelopes are computed by means of Modal Interval Analysis. A way of controlling the error of the envelopes and adjust it to the desired value is presented.

Keywords: semiqualitative simulation, interval analysis.

1 Introduction

Most of the existing simulators need a mathematical model in which the values of the parameters are real numbers. This implies that the user must have a totally deterministic knowledge of the system. However, complex systems are often subjected to uncertainties that make such a model difficult if not impossible to obtain. A precise model cannot represent the behaviour of such systems which require an explicit representation of imprecisions and uncertainties. A special case is when the uncertainties are structured: only the parameters undergo imprecisions but the model structure is known. This case can be handled with interval models in which the equation parameter values are allowed to vary within numeric intervals. For instance, such an interval model could be given by the following transfer function with interval parameters which represents a linear differential relation between an input \( u(t) \) and an output \( y(t) \):

\[
F(s) = \frac{Y(s)}{U(s)} = \frac{[2, 3] s + [1, 3]}{[1, 2] s^2 + [3, 5] s + [2, 4]}
\]

in which \( s \) is the Laplace variable and \( U(s) \) and \( Y(s) \) are the input and output Laplace transforms, respectively. Actually, a precise model can be viewed as an interval model in which the interval widths are zero. As interval widths decrease, precision increases [12].

The results of the simulation of such interval models is in the form of envelope trajectories (or envelopes for short)[6] [24]. The most common use is for Fault Detection (FD) as it provides a way to compute automatically and in a model-based sound manner adaptive alarm thresholds for every variable [22]. The envelopes can be characterised by several properties, the main ones being completeness and soundness which lead to the concepts of overbounded and underbounded envelopes which have radical consequences on the robustness and sensitivity of the FD system.

The simulation of interval models can be performed by several means including quantitative, qualitative and semiqualitative techniques. Existing simulators do not provide any information about the "error" with respect to the exact envelope. A method to obtain error-bounded envelopes
is proposed. It is based on the simultaneous computation of an underbounded and an overbounded envelope. Both envelopes are computed by means of Modal Interval Analysis. A way of controlling the error of the envelopes and adjust it to the desired value is also provided.

The next section defines the envelopes and their properties in relation to the fault detection problem. Section 3 discusses the related work and provides a summary of the existing simulators that can be used to generate envelopes, putting special emphasis on simulators based on interval arithmetic. In section 5 a method to generate error-bounded envelopes is presented. This method is based on Modal Interval Analysis, which is presented in section 4, and applied to the envelope generation problem in section 6. Finally, some conclusions and directions for the future work are discussed.

2 Envelopes and their properties in relation to Model-based Fault Detection

When simulating an interval model, which actually represents a whole set of models, the state space is a compact set which can be represented by an envelope trajectory for every variable. All the possible behaviours starting from a specific initial state are compacted within a unique curve. This envelope hence includes a whole family of temporal curves, like the one displayed in figure 1.

![Figure 1: The bounds of the envelopes are two curves](image)

The size of the envelope is critical. If it is too tight there are systems belonging to the model set whose output is outside the envelope. Such an envelope is not complete, taking the definition that a complete envelope is one that includes all possible behaviours [21]. On the other hand, if the envelope is too wide it includes zones that cannot be reached by any of the systems belonging to the set. Such an envelope is not sound. Our definition of a sound envelope is that every point inside the envelope belongs to the output of at least one of the systems belonging to the set. A complete but not sound envelope is an overbounded envelope. A sound but not complete envelope is an underbounded envelope.

The ultimate goal is to generate a complete and sound envelope, that is the exact envelope. However, more realistic goals are to produce either a minimally overbounded envelope or a minimally underbounded envelope.

3 Related work: envelope simulators

There are many simulators that can be used to generate envelopes. They can be classified into different groups depending on the information used for the simulation: quantitative, qualitative or semi-qualitative [1]. A detailed survey of these simulators can be found in [2]. Among the semiquantal simulators there are methods based on quantitative simulation ([3] [4] [5] [11] [12]), methods based on qualitative simulation ([3] [12] [13]) and methods based on interval arithmetic, which are the ones that are directly related to the approach exposed in this paper.

Interval arithmetic [18] allows to consider the whole continuous range of possible instances represented by an interval model. This is due to the natural extension and one of its properties: monotonic inclusion.

**Definition 1** The natural extension of a rational function is the one obtained by substituting real arguments by intervals and rational operations by their arithmetic interval extensions.

**Theorem 2 (Monotonic inclusion)** Given $f(x_i)$, a real function, and $F(X_i)$, its natural extension, then $x_i \in X_i$ implies $f(x_i) \subseteq F(X_i)$.

In consequence, the natural extension gives a guarantee on the result: no function in the class can take values outside the range computed using interval arithmetic. Unfortunately, it does not provide the exact result (complete and sound) in the general case. This comes essentially from the two following problems:
The multi-incidence problem: interval arithmetic considers that each incidence of a variable in a function is independent of the other. Similarly, it is unable to take into account other dependencies. The compiled range is hence overbounded.

The wrapping problem: the state, at some time point, of a system with interval parameters may be represented by a hypercube. However, it may be that the system’s state does not evolve into another hypercube at the next time point. In figure 2 an example with two state variables is shown: the hypercubes is a rectangle. It transforms into a rhombus at the following time step (it could actually evolve into any two dimensional shape). The projection of this rhombus on the variable axis leads to a new rectangle which obviously includes spurious states, as shown in the figure. Therefore, the obtained envelopes are overbounded.

Given the set of closed intervals of $R$, $I(R) = \{[a, b] | a, b \in R, a \leq b\}$, and the set of logical existential and universal quantifiers $\{E, U\}$, a modal interval is defined by a pair

$$X := (X', Q)$$

where $X' \in I(R)$ and $Q \in \{E, U\}$. $QX$ is the modality and $X'$, called extension, is an interval in the classical sense, i.e. a set of real numbers. The canonical notation for modal intervals is:

$$[a_1, a_2] := \left\{ \begin{array}{l}
([a_1, a_2], E) \text{ if } a_1 \leq a_2 \\
([a_2, a_1], U) \text{ if } a_1 \geq a_2
\end{array} \right\}$$

A modal interval $([a_1, a_2], E)$ (the classical one) is called "existential interval" or "proper interval" and denotes one value within the interval bounds. On the other hand, a modal interval $([a_2, a_1], U)$ is called "universal interval" or "improper interval" and denotes all the values within the interval bounds. Thus, the proper interval $[1, 3]$ points at some real number between 1 and 3 and the improper interval $[3, 1]$ points at any real number between 1 and 3.

The rational operations between modal intervals are extensions of classical interval arithmetic with the addition of the dual operator defined by:

$$Dual([a_1, a_2]) = [a_2, a_1]$$

The dual formulation of the modal intervals allows to define two modal interval extensions of a continuous function $f$: $f^\ast (X)$ and $f^{**}(X)$. The modal interval extension represented by $f^\ast (X)$ may be interpreted as

$$f^\ast (X) \subseteq F (X) \iff U (x_p, X^\ast_p)$$

$$Q (x, F (X)) E (x_i, X^i) (z = f(x_p, x_i))$$

where $X_p$ and $X_i$ are the proper and improper components of $X$, respectively. This interpretation can be read: "For all elements belonging to the proper intervals there exists at least one element in the improper intervals that fulfill the function".

**Example 3** $[10, 20] + [20, 15] = [30, 35]$ means

$$U (a, [10, 20]) E (f, [30, 35])$$

$$E (b, [15, 20]) (a + b = f)$$

**Example 4** $[10, 20] + [15, 20] = [25, 40]$ means

$$U (a, [10, 20]) U (b, [15, 20])$$

$$E (f, [25, 40]) (a + b = f)$$

Figure 2: The wrapping problem

Some semiquantitative simulators based on interval arithmetic are presented in [6] [12] [14] [16] [18] and [24].

All these simulators produce overbounded envelopes. Some methods use fuzzy sets: all of them take the α-cut of the fuzzy sets and end up using interval arithmetic. Some of them are presented in [7] and [20].

4 Modal Interval Analysis

Modal Interval Analysis [8] [9] [10] extends real numbers to intervals, identifying the intervals by the predicates that the real numbers fulfill, unlike classical Interval Analysis which identifies the intervals with the set of real numbers they contain.

In the following, some of the properties of modal intervals that are interesting for envelope generation are stated.
On the other hand, the semantic interpretation of the $f^{**}(X)$ extension is dual:

$$F(X) \subseteq f^{**}(X) \iff U(x_i, X_i)$$ (8)

$$Q(z, D^{\text{Dual}}(F(X))) \mathcal{E} (x, X_x)$$

$$(x = f(x, x_i))$$

The computation of $f^*(X)$ and $f^{**}(X)$ is not always possible. The usual procedure is to find overbounded computations of $f^*(X)$ and underbounded computations of $f^{**}(X)$ which maintain the semantic interpretations. To this respect, an aspect to be taken into consideration is the rounding of computations. Computers work with digital numbers, not with real numbers. In order to maintain the semantic interpretations, direct roundings (up or down) have to be used.

If $f$ is a rational function, there are some theorems in Modal Interval Analysis that allow to obtain the exact range of $f$ in some cases or overbounded computations of $f^*(X)$ and underbounded computations of $f^{**}(X)$ in other cases.

**Definition 5** A modal interval extension $fR(X)$ of $f$ in $X$ is optimal if

$$f^*(X) = fR(X) = f^{**}(X)$$ (9)

The following theorem provides the conditions and the way to obtain optimal extensions.

**Theorem 6** Given $fR(X)$, a rational interval function defined in a parameter space $Prop(X)$, tree-optimal and totally monotonic for each multi-incident component of $X$. Let $XD$ be an enlarged vector of $X$ obtained considering each multi-incident component as independent and transforming it into its dual if it is antitonic (the monotonicity of this incidence and the monotonicity of the component have opposite senses). In this case,

$$f^*(X) = fR(XD) = f^{**}(X)$$ (10)

**Example 7** Given $f = x^2 - xy$ and the parameter space $x = [2, 4]$ and $y = [1, 2]$, the range of the function obtained by its natural extension is $f^*(X) \subseteq [-4, 14]$. The exact range can be computed applying theorem 6 and is $f^*(X) = f^{**}(X) = [2, 4]^2 - Dual([2, 4])[1, 2] = [0, 12]$

If the function is not monotonic for each multi-incident component, theorem 6 can be partially applied in order to reduce the complexity of the problem. For instance, given an $n$ variable function, the problem of finding the range of this function in a domain in which the function is monotonic with respect to $r$ variables, can be reduced to evaluate the range of an interval function of $n - r$ variables. Therefore, the problem complexity has a lower order.

**Example 8** Given $f = xy - x^2 - 2y$ and the parameter space $x = [1, 2]$ and $y = [3, 4]$, the range of the function obtained by its natural extension is $f^*(X) \subseteq [-9, 1]$. The function is totally monotonic with respect to $y$

$$\frac{\partial f}{\partial y} = x - 2 = [-1, 0] \leq 0$$ (11)

but it is not totally monotonic with respect to $x$

$$\frac{\partial f}{\partial x} = y - 2x = [-1, 2] \geq 0$$ (12)

**Theorem 6** can be applied to $y$ and hence a better approximation of the range of the function is obtained:

$$f^*(X) \subseteq x^{Dual}(y) - x^2 - 2y = [-8, -1]$$ (13)

A way to obtain even a better approximation is by splitting the parameter space. The advantage is that now only the variable $x$ must be split. Moreover, the range in each sub-space can be computed more exactly because the modality of each incidence of the variable $y$ is already known.

As a conclusion, the number of sub-spaces to be made in order to compute an approximation of the range of the function is smaller when modal intervals are used. This is illustrated in [23], in which modal intervals combined with a branch-and-bound algorithm have been applied to the analysis and design of robust controllers.

5 Using modal intervals for generating and controlling error-bounded envelopes

The properties of the simulators presented in section 3 can be exhibited only in a binary manner. For example, all the interval based simulators are complete but not sound, i.e., they provide over-bounded envelopes. A step forward would be to produce some kind of measure of the degree of over-bounding, in other words to be able to evaluate the "error" of the obtained envelope with respect to the
exact one. However, the exact envelope is of course not known. The exact envelope problem is actually highly complex as it requires global optimisation tools for non linear and non convex functions.

The alternative way proposed in this paper is to bound the error, i.e. to determine the maximum distance. This is achieved by computing both an underbounded envelope and an overbounded envelope. The distance between these envelopes indeed gives the maximum error. These envelopes are defined as error-bounded envelopes.

Moreover it is shown that the maximum error can be controlled by widening the underbounded envelope or by tightening the overbounded one.

The generation of the underbounded and overbounded envelopes is approached in an original way by using Modal Interval Analysis.

5.1 The multi-incidence problem in the simulation task

Interval based methods are based on the reformulation of the simulation problem into an optimisation problem.

The behaviour of a $n$-th order dynamic system can be represented by the following difference equation:

$$y_{t+1} = \sum_{i=0}^{n} a_i y_{t-i} + \sum_{j=0}^{m} b_j u_{t-j}$$

(14)

in which it can be observed that the output of the system at any time point ($y_{t+1}$) depends on the values of the previous outputs ($y_{t-i}$) and inputs ($u_{t-j}$). This dependency is given by the parameters of the system ($a_i$ and $b_j$). Hence, finding the limits of the envelope at $s$ given time point is equivalent to finding the maximum and the minimum of a function into a parameter space. This is a global optimisation problem. Nevertheless, the computational cost of a global optimisation algorithm is too high to use them for this task.

There are many different methods for global optimisation, but many of them have no guarantee of finding the global optimum. In the case of the envelope generation problem, a local optimum results in an incomplete (underbounded) envelope. Moreover, the computational effort needed by these methods is very high because they search the optimum trying to minimize the error as much as possible. For most of the applications it is not needed to have a very small error and therefore it is not necessary to make such a computational effort.

Conversely, global optimisation methods based on the interval arithmetic obtain overbounded results as stated by the monotonic inclusion property (see. section 3). This comes from the multi-incidence problem. In the following, two types of multi-incidences are pointed out.

The first type of multi-incidence comes from the fact that, in a difference equation like the one shown above, some parameters may appear several times. For instance, the transfer function representation of a generic first order system is:

$$F(s) = \frac{Y(s)}{U(s)} = \frac{k}{\tau s + 1}$$

(15)

in which $k$ is the static gain and $\tau$ is the time constant. If this transfer function is discretised by the Euler method, the difference equation representation of this system is:

$$y_t = \left(1 - \frac{T}{\tau}\right) y_{t-1} + \frac{kT}{\tau} u_{t-1}$$

(16)

in which $T$ is the sampling period. As it can be seen, $\tau$ appears more than once in this equation. If the equation is rewritten, renaming the parameters as follows:

$$a = \left(1 - \frac{T}{\tau}\right)$$

$$b = \frac{kT}{\tau}$$

(17)

the situation is even worse because the multi-incidences do not appear explicitly. This is the reason for avoiding intermediate operations: each time an intermediate operation is performed some information is lost. Going deeper, $k$ and $\tau$ themselves include implicit multi-incidences with respect to the physical parameters of the device!

The second type of multi-incidences is particular to the simulation mechanism as the equations are taken at different time points. For instance, the difference equation at the time point $t + 1$ of the generic first order system used above is:

$$y_{t+1} = \left(1 - \frac{T}{\tau}\right) y_t + \frac{kT}{\tau} u_t$$

(18)

As it can be seen, $y_t$, $\tau$ and $k$ appear both in the difference equation for time point $t$ and $t + 1$. It should be noticed that these multi-incidences can be treated as independent variables for systems whose parameters are known to vary in time. On the other hand, they must be treated as so if the physical system is assumed to be invariant.
5.2 The proposed method

The proposed approach is to make the multi-incidences explicit by merging the different equations starting from 0 into a unique expression on which the optimisation is performed. This expression is obtained in a recursive way by substituting $y_i$ within equation 18 down to $y_0$. For instance, in the case of a first order system the following expressions will be used:

\[
y_1 = \left(1 - \frac{T}{\tau}\right)y_0 + \frac{kT}{\tau}u_0
\]

\[
y_2 = \left(1 - \frac{T}{\tau}\right)^2y_0 + \left(1 - \frac{T}{\tau}\right)\frac{kT}{\tau}u_0 + \frac{kT}{\tau}u_0
\]

\[
y_3 = \left(1 - \frac{T}{\tau}\right)^3y_0 + \left(1 - \frac{T}{\tau}\right)^2\frac{kT}{\tau}u_0 + \\
\quad + \left(1 - \frac{T}{\tau}\right)\frac{kT}{\tau}u_1 + \frac{kT}{\tau}u_2
\]

... Modal Interval Analysis then provides an efficient tool to perform the optimisation task taking the multi-incidences into account. Moreover, the two semantic interpretations provided by modal intervals are directly applicable to compute the overbounded and the underbounded envelopes:

- Overbounded envelope. Its semantics is: "For every (universal quantifier) model parameter, input and initial state, the output belongs to the envelope (existential quantifier)", which corresponds to $f^*(X)$ (see equation 5).

- Underbounded envelope. The semantics is dual: "For every output belonging to the envelope there exist parameter, input and initial state values that produce this output", which corresponds to $f^{**}(X)$ (see equation 8).

Therefore, Modal Interval Analysis can be used to compute both envelopes.

6 Practical implementation and examples

In section 4, some tools of Modal Interval Analysis that are useful for envelope generation have been described. One limitation of these tools is that it is necessary to differentiate the function in order to apply theorem 6, hence restricting the method to differentiable functions. However, the discrete representation of the system used in our simulation problem is differentiable.

A simulator based on these tools has been implemented. It uses Matlab version 5.1 for Unix [17]. Symbolic computations are performed with Maple V r4 [16] through the Symbolic Math Toolbox. Moreover, it uses C++ programs as MEX-files to perform modal interval computations with direct roundings and to accelerate the branch-and-bound algorithm.

The implemented simulation algorithm is the following:

```
Modal Interval Branch-and-bound Algorithm
Given a function in a space
/*/ The exact range of the function can be computed if there are not multi-incidences */
Modality of uni-incident variables is not changed
IF all variables are uni-incident THEN
  Exact result
END
ENDIF
Calculate internal approximation
  DO
    External = internal
    DO /* The monotonicity of the function is studied in order to apply theorem 6*/
      Get subspace
      FOREACH variable with unknown modality
        IF 0th derivative THEN
          fix modality
        ENDIF
      ENDFOREACH
    IF all modalities are known THEN
    /* Application of theorem 6 where possible */
      Calculate partial exact
      Internal = internal V partial exact
      External = external V partial exact
    ELSE
      /* Division of subspaces if necessary */
      Calculate partial internal
      Internal = internal V partial internal
      Calculate partial external
      External = external V partial external
      Divide subspace
    ENDIF
    WHILE remaining subspaces
      IF error < epsilon THEN
        final = 1
      ENDIF
      WHILE final = 0
    END
```

END
As an example, figure 3 shows the envelopes obtained for a generic first order system with the following parameters:

- static gain: $k = [0.95, 1.05]$
- time constant: $\tau = [5, 20]$
- initial state: $y_0 = 0$
- sampling time: $T = 1 \text{ s}$
- input: steps of different lengths and magnitudes
- maximum error of the envelopes $\varepsilon \leq 0.2$

![Figure 3: Example of simulation](image)

Figure 3: Example of simulation

In this figure, the solid line is the overbounded envelope and the dotted line is the underbounded one. The error between the two envelopes is not very small, but sufficient in most cases, for instance for fault detection. This allows to obtain useful results with a computational effort much lower than the one needed to obtain similar results using global optimisation algorithms.

Another example is the one shown in figure 4, in which the same system as above is excited with a sinusoidal input. On the right, a high frequency white noise has been added to the input, whereas on the left the input is the same but without the noise. The interesting comment about this example is that, as it can be seen on the figures, a significant amount of noise has been "absorbed" by the envelopes. Indeed, the envelopes produced with or without noise have a much higher similarity than the noisy and non-noisy inputs. In particular, the envelopes corresponding to the noisy input are not wider than the other ones. The width depends essentially on the imprecision of the model. This means that semiqualitative simulation acts as a high frequency filter. This property has not been carefully studied yet but should deserve attention in view of fault detection applications.

![Figure 4: Simulation without (left) and with (right) noise](image)

7 Conclusions and future work

In this paper it has been shown that the existing simulators for systems with structured uncertainties provide envelopes which may or may not have properties like completeness, soundness, stability, etc. or not. Sometimes the properties are not even known.

When the properties are known, the error of the envelopes with respect to the exact one is unknown. A method to obtain error-bounded envelopes is proposed. It is based on the simultaneous computation of an underbounded envelope and an overbounded one. Two ways to control this error are proposed as well: tightening the overbounded envelope or widening the underbounded one. Both can be achieved by means of Modal Interval Analysis. The error of our envelopes can hence be adjusted to the desired value. The computation effort of course increases when the error decreases.

This method has been implemented in Matlab and uses Maple and C routines. This facilitates its future integration into a supervision framework that is being developed based on Matlab and Simulink. It is also planned to be used to improve
the prediction and fault detection algorithms of the Ca"En simulator [22].

An inconvenient of this method is that the computation effort increases at each step of the simulation. The exact envelope at a time point \( t \) can only be obtained by computing the range of the function that relates the current time point to the initial one, i.e. all the previous states must be considered in order to obtain the exact envelope. This means that the procedure is not incremental. A solution to overcome this problem is the use of a shorter temporal window. Due to the dynamics of the systems, it has been shown that the influence of the previous states over the current one decreases with time [19]. Hence, it should be possible to obtain close results with a shorter temporal window length. Saludes [19] recommends a window length which essentially depends on the time constant of the system.

A related problem is about the semantics of these results. For instance, given an overbounded envelope and an underbounded one at a time point \( t \), both can be used as the initial state for the function at time point \( t + m \). This means that four approximations of the range of the function can be obtained: over and underbounded approximations of the range of the function using over and underbounded approximations of the initial state. Each one of these four approximations has different semantics and the most suitable one, if it exists, has to be decided. This problem is still under investigation. It seems that the solution is simple for overbounded envelopes. This envelope can be obtained computing an overbounded range of the function and using an overbounded initial state. As the length of the temporal window increases, the envelope gets closer to the exact one. The case of underbounded envelopes is more complex and more work remains to be done.

As it has been shown, a very interesting feature of modal intervals is the semantics. A future work is to study whether envelopes with different semantics can be used not only to detect the faults but also to localize the faulty parameter. For instance, if a system has two physical parameters \( a \) and \( b \), envelopes with the semantics "for every \( a \) there exists \( b \) so that..." or "for every \( b \) there exists \( a \) so that..." can be obtained. If a system is faulty and its output belongs to only one of these two envelopes, it should be possible to determine whether it is \( a \) or \( b \) that is faulty.

Finally, in [4] it is claimed that it is not necessary to study the evolution of all points belonging to an uncertainty region to know the evolution of the region. The study of the evolution of points belonging to its surface is enough. The possible application of this result to modal interval simulation gives another direction for research.

Acknowledgements

This work has been funded by the autonomous government of Catalonia, Generalitat de Catalunya, through CIRIT 1996 XT 00023 and Grup de Recerca Consolidat ref. 1995GR00474 and by the Spanish government through co-ordinated research project CICYT TAP96-1114-C03-03, "Plataformas integradas de CAD de Supervisión y Metodologías".

References


UNAGENERALITZACIÓDELSMODELSQUALITATIUSD’ORDRESDE MAGNITUD.
APLICACIÓ A L’OPERADOR SUMA.

ANDREUCATALÀMALLOFRÉ
Dept.d’Enginyeria de Sistemes Automàtica i Informàtica Industrial
E.U.P.V.G. Av.Victor Balaguer
08800 Vilanova i la Geltrú
(Barcelona)
andreu@esat.upc.es

JOSEP MARIA MORENO FURNER
Dept.de Matemàtica Aplicada III
E.U.P.M. Av. de les Bases de Manresa, 61-73
08240 Manresa
(Barcelona)
jmoreno@pie.xtec.es

Resum

Partint del model dels signes i del model d’ordre de magnitud de granulació 7, es tracte de dotar-los d’una nova nomenclatura que permeti fàcilment la seva generalització a models d’ordre de magnitud superior.

Paraulas clau: raonament qualitatiu, ordres de magnitud.

1. Introducció

Es poden situar els orígens del raonament qualitatiu pels anys 60, quan s’observa que una anàlisi qualitativa dels sistemes econòmics permet sovint arribar a conclusions significatives [1]. L’evolució dels diferents models teòrics porten a trobar-hi aplicacions ens camps com ara l’automàtica, la biologia, la física, la psicologia, entre d’altres. Dels diferents models qualitius que es poden trobar, àlgebra des signes [2], àlgebras intervalàries [3] [4], àlgebras híbrids [5], ordres de magnitud absoluts [6] [7], ordres de magnitud relatius [8] [9] [10] [11], models mixtes (absolut-relatius), ordres de magnitud amb relacions difuses [12] [13],... es considerarà com a base de l’estudi el model complet dels ordres de magnitud [6] [14], utilitzant unes dades totalment qualitatives. La filosofia que es pretén seguir alhora de considerar les dades totalment qualitatives és la de la discretització de la recta real.

2. Models d’ordres de magnitud

2.1. El model dels signes

El model dels signes s’obté mitjançant la divisió de la recta real en tres classes d’equivalència on cada classe vindria definida per la relació tenir el mateix signe. A cada una de les tres classes d’equivalència que s’obtenen se li associa l’etiqueta “-”, “0”, “+”.

Es denota per S = { - , 0 , + } el conjunt de les classes d’equivalència

De seguida es va veure amb la necessitat d’introduir una nova etiqueta si el que es pretenia era que a l’operar etiquetes el resultat fos una etiqueta dels mateix conjunt.

D’aquesta manera s’obté S = { - , 0 , + , ? } anomeat espai de quantitats en el qual es poden considerar l’operador suma qualitatiu i l’operador producte qualitatiu i que estan definits segons les taules 1 i 2.

<table>
<thead>
<tr>
<th>-</th>
<th>0</th>
<th>+</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-</th>
<th>0</th>
<th>+</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Taules 1 i 2: operador suma qualitatiu i operador producte qualitatiu.

Les propietats d’aquest operadors es troben ampliament estudiades en [14].

2.2. Model d’ordres de magnitud de absoluts

Els models basats en els ordres de magnituds absoluts parteen dels de la idea que si dos objectes es poden distingir qualitativament, llavors han de tenir etiquetes diferents. Així doncs, dues quantitats positives diferents han de tenir etiquetes qualitatives diferents. Per això cal definir diferents etiquetes que representin els diferents ordres de magnitud suggerits pel propi context on es treballi. Això implica la partició de la recta real en 2n + 1 classes, simètriques respecte el zero. Per facilitar la seva interpretació ens centrarem en un dels models descrit per [6] [7] on es divideix la recta real en set classes corresponents a les etiquetes:
Nomament qualitatiu

NG = negatiu gran; NM = negatiu mitjà; NP = negatiu petit; 0 = zero; PP = positiu petit; PM = positiu mitjà; PG = positiu gran; fig. 1

Es denota per S 1 al conjunt format per les etiquetes de la partició.

S 1 = \{ NG, NM, NP, 0, PP, PM, PG \}

Donats dos elements, \( x \) i \( y \), de S 1 es defineix l'etiqueta \([x, y]\) de la manera següent:

\[
[x, y] = \begin{cases} 
    x & \text{si } y = 0 \\
    y & \text{si } x = 0
\end{cases}
\]  

El interval més petit que conté \( x \) i \( y \)

Es defineix aleshores el que s'anomena Univers Complet de Descripció i que es denota per:

\( S = S 1 \times \{ [x, y] \} \times q. x, y \in S 1 \times [x, y] \times \leq \}

En S es pot considerar una relació d'ordre induïda per la inclusió, i s'introduïen sis nivells precisió diferents, fig. 2, que ajuden a determinar millor l'ordre de magnitud [15].

La majoria dels treballs basats en els ordres de magnitud defineixen la igualtat qualitativa. En S la igualtat qualitativa \( q\)-igualtat s'ha formalitzat de la manera següent:

Sigueix \( x \), \( y \in S \) direm que, \( x \) i \( y \) (igualtat qualitativa) si existeix un \( z \in S \) tal que \( z \leq x \) i \( z \leq y \).

Un dels problemes més importants que presenta aquesta igualtat qualitativa és que es tracta d'una relació reflexiva, simètrica, però no transitiva [3], per exemple si tenim: \( x = [\text{PP}, \text{PM}], y = [\text{PM}, \text{PG}] \), i \( z = \text{PG} \) aleshores es podrà dir \( x \leq y \) i \( z \leq x \) i això no és veritat.

En l'Univers Complet de Descripció ( S ) es defineix també una suma qualitativa \( q\)-suma (taula 3) i un producte qualitatiu \( q\)-producte [16]. Aleshores tenim que la quaterna formada per ( S, , , ) presenta una estructura d'algebra qualitativa, això és,

- els operadors suma qualitativa i producte qualitatiu són qualitativament associatius
- els operadors suma qualitativa i producte qualitatiu són commutatius
- l'operador producte qualitatiu és qualitativament distributiu respecte l'operador suma qualitativa

<table>
<thead>
<tr>
<th>NG</th>
<th>NG</th>
<th>NM</th>
<th>NP</th>
<th>0</th>
<th>PP</th>
<th>PM</th>
<th>PG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td></td>
<td>NG</td>
<td></td>
<td></td>
<td>[NG, NM]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[NG, NM]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>[NP, NM]</td>
<td></td>
<td>[NP, PP]</td>
<td>[PP, PM]</td>
<td>[PM, PG]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[NP, NM]</td>
<td>[NP, PP]</td>
<td>[PP, PM]</td>
</tr>
<tr>
<td>PP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[PP, NM]</td>
<td>[PP, PM]</td>
<td>[PM, PG]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[PP, NM]</td>
<td>[PP, PM]</td>
<td>[PM, PG]</td>
</tr>
</tbody>
</table>

Taula 3 En les cases on hi trobem més d'un valor, aquests corresponen als casos en que \( b < 2a \), \( b = 2a \), i \( b > 2a \) on a i b són els llindars que es prenen ahora de discretitzar la recta real

Val a dir, que el model dels signes, estudiat en l'apartat anterior, es pot considerar un model d'ordre de magnitud format per dos nivells de precisió (fig. 3)

3. Aportacions al model qualitatiu d'ordres de magnitud

Amb la idea de facilitar la generalització de la nomenclatura del model d'ordres de magnitud abs vocals descrit per [6] [7], en la recta real es divideix en set classes, introduïm el concepte d'espai qualitatiu, Q(2n+1), on els elements passen a tenir una interpretació vectorial i per tant molt simple.

3.1. L'espai qualitatiu Q ( 2n + 1 ). Elements i base de l'espai Q ( 2n + 1 )

Definim un nou espai qualitatiu que representem per la lletra Q(2n+1), on (2n+1) indica la granulació de l'espai. Tant el model de l'algebra dels signes com el model dels ordres de magnitud absoluts [6] [7] esdevenien casos particulars d'aquest nou espai qualitatiu Q(2n+1).

Sigui (2n+1) la granulació de l'espai qualitatiu Q(2n+1) on n .

Observació: la granulació de l'espai qualitatiu Q(2n+1) ens indica el grau de discretització que estem considerant de la recta real.

Un element qualitatiu de l'espai qualitatiu Q(2n+1) vindrà representat per:
A = (a₁, a₂, a₃, ..., aₙ, aₙ₊₁)

i) aᵢ ∈ {0, 1}

ii) aᵢ = {n, n₊₁, ..., n₋₁, n}  
    t.q. aᵢ = 1

iii) sigui j.t.q. aᵢ = 1 i aⱼ = 1 aleshores:

    aⱼ = 0  
    t.q.  
    aⱼ = 1

Sigui (2n+1) la granulació de l'espai qualitatiu

Q(2n+1) on n. Llavors al conjunt d'elements de la forma:

\[ e − n = (1, 0, 0, 0, 0, 0, 0, 0, 0) \]

\[ e + n + 1 = (0, 1, 0, 0, 0, 0, 0, 0, 0) \]

\[ e + 0 = (0, 0, 1, 0, 0, 0, 0, 0, 0) \]

\[ e + n − 1 = (0, 0, 0, 1, 0, 0, 0, 0, 0) \]

\[ e + n = (0, 0, 0, 0, 1, 0, 0, 0, 0) \]

direm que formen una base de l'espai qualitatiu Q(2n+1)

Notarem per eᵢ els elements de la base on n i n
Notarem per qₛₜ a la resta d'elements aₙ − n s t n

Observació: utilitzarem el signe \( \{-, +\} \) per acompanyar als subíndexs dels elements de qualitatiu de Q(2n+1) tal com es mostra en els exemples anteriors.

El primer subíndex (s) ens indica la posició que ocupa el primer d ins el vector de granulació (2n+1) i el segon subíndex (t) ens indica la posició que ocupa el darrer t. Cal tenir present que els subíndexs del vector de granulació (2n+1) van de (−n) fins a (n) passant pel 0.

Exemples:

Si n = 1 el model que s’obté és el corresponent a l’algebra dels signes.

La granulació de l’espai qualitatiu és 3 (Q₃), i els seus elements són:

\[ e − 1 = (1, 0, 0, 0) \]

\[ e + 0 = (0, 1, 0) \]

\[ e + 1 = (0, 0, 1) \]

Aquests tres elements formen la base de l’espai Q₃.

Altres elements seran:

\[ q − 10 = (1, 1, 1, 0) \]

\[ q + 0 + 1 = (0, 1, 1, 1) \]

\[ q + 1 + 1 = (1, 1, 1, 1) \]

Si n = 3 obtenim el model dels ordres de magnitud absolut

La granulació de l’espai qualitatiu és 7 (Q₇), i els seus elements són:

\[ e − 3 = (1, 0, 0, 0, 0, 0, 0) \]

\[ e − 2 = (0, 1, 0, 0, 0, 0, 0) \]

\[ e − 1 = (0, 0, 1, 0, 0, 0, 0) \]

\[ e + 0 = (0, 0, 0, 0, 0, 0, 0) \]

\[ e + 1 = (0, 0, 0, 0, 0, 0, 0) \]

\[ e + 2 = (0, 0, 0, 0, 0, 0, 0) \]

\[ e + 3 = (0, 0, 0, 0, 0, 0, 0) \]

Aquests set elements formen la base de l’espai Q₇.

Altres elements seran:

\[ q − 3 − 2 = (1, 1, 1, 0, 0, 0, 0) \]

\[ q − 1 + 1 = (0, 0, 1, 1, 0, 0, 0) \]

3.2. L’operador suma en l’espai qualitatiu Q(2n + 1)

Definirem la suma de dos elements qualsevol de l’espai qualitatiu Q(2n+1) a partir de la suma dels elements de la base de l’espai. Per això, tractarem en primer lloc la suma dels elements de la base per passar a continuació a la suma de dos elements qualsevol.

3.3. Suma qualitativa dels elements de la base

En base a un factor de salt α, que té la finalitat de ponderar la discretització de la recta real per tal d’aconseguir una millor adaptació dels elements qualitatives als diferents problemes reals, es podria dir que el factor de salt α és un indicador de la magnitud o força dels elements de l’espai qualitatiu Q(2n+1), definint el que és la suma qualitativa dels elements de la base. En fase de desenvolupament, per facilitar la seva interpretació expressarem els resultats a través de taules.

exemple de taula corresponent a \( \alpha = 1 \)

els elements e−1, e0 i e1 de l’espai qualitatiu Q7 tenen una magnitud inferior a la resta d’elements de l’espai. Això vol dir que alhora de ser sumats amb qualsevol altra element de l’espai no juguen cap paper fonamental i que al ser sumats entre ells no fa variar la magnitud del resultat. ( *)
La introducció del factor de salt ens permet adaptar l'espai qualitatiu a les necessitats específiques de cada problema concret.

La implementació de les operacions qualitatives en un llenguatge de programació (matlab) es preveu senzilla. Com a línia de treball tenim:
- Estudi de la consistència i de les propietats dels operadors, definitos en $Q(2n+1)$

- Definició de nous operadors en $Q(2n+1)$ com ara la distància, útilis pel desenvolupament de la teoria qualitativa per a sistemes connexionistes.

- Aplicacions a problemes reals dels tipus:
  - diagnosis
  - classificació [17]

Referències:


INSTANCE-BASEDMETHODSFUROOECEANOGRAPHICPREDICTION

1J. M. Corchado, 2J. R. Garcia Bermejo Giner, 1B. Lees
1Department of Computing and Information Systems
University of Paisley, UK.
2Departamento de Informática y Automática
Universidad de Salamanca, Spain
corc-ci0@wpmail.paisley.ac.uk

Abstract

This paper presents an Instance-based reasoning system that has proved capable of creating a prediction detailing the physical evolution of a rapidly changing oceanographic environment. The aim of the system is to identify and forecast the thermal structure of the water ahead of an ongoing vessel. The work focuses on the development of a system for forecasting the behaviour of complex environments, in which the underlying knowledge of the domain is not completely available, the rules governing the system are fuzzy and the sets of data samples are limited and incomplete. This paper presents an approach that shows the ability of an Instance-based reasoning (IBR) system to solve the above mentioned problem.

The structure of the paper is as follows. The paper presents a brief overview of the basic concepts of case-based reasoning and Instance-based reasoning and an introduction to the forecasting problem. The IBR forecasting system is then explained, and, finally, an outline of some of the results obtained to date is presented.

Index of Abbreviations: IBR: Instance-based reasoning; CBR Case-based Reasoning; AI: Artificial Intelligence; KBS: Knowledge-based System; ANN: Artificial Neural Network; RBF: Radial Basis Function; ARIMA: Autoregressive Integrative Moving Averages.

1. Introduction to Instance-based Reasoning Systems.

Knowledge-Based Systems (KBS) are one of the most popular and successful branches of the Artificial Intelligence (AI) world. Nevertheless, developers of these systems have met several problems [13]. In many cases it can be difficult to deal with the knowledge elicitation part of the problem, implementation of KBS is also complicated and once implemented KBS are often slow and are unable to access or manage large volumes of information. Therefore they are normally difficult to maintain.

With the intention to overtake these problems, Kolodner (1993) proposed a revolutionary model: Case-based reasoning (CBR) model which is in fact a model of human reasoning. The idea behind a CBR is that people rely on concrete previous experiences when solving new problems. This fact can be experimented in any day to day problem by simple observation or even by psychological experimentation. Since the CBRs were proposed, they have been proved successful in a wide range of application areas [8]. A case-based reasoning system solves new problems by adapting solutions that were used to solve old problems. The case base holds a number of problems with their corresponding solutions. Once a new problem arises, the solution to it is obtained by retrieving similar cases from the case base and studying the solutions given to them. A CBR is a dynamic system in which new problems are added to the case base, redundant ones are eliminated and others are created by combining existing ones.

CBR systems analyse a problem, and by means of indexing algorithms, retrieve previously stored cases, along with their solution, match them and adapt them to a given situation with the intention of abstracting a solution, from the knowledge stored in the form of cases in the case-base. All of these actions are self-contained and can be represented by a cyclical sequence of processes, in which human intervention may be needed.

A typical CBR system is composed of 4 sequential steps that are recalled every time that a problem needs to be solved [1, 13]. This cyclical process involves four major steps:

Retrieval the most relevant case(s).
Reuse the case(s) to attempt to solve the problem;
Revise the proposed solution if necessary, and
Retain the new solution as a part of a new case.
As any other problem solving mechanism, the aim of any CBR is to find the solution for a problem. The mission of the retrieval algorithms is to search the case base and to select from it the most similar problems, together with their solutions to the new problem to be solved. Cases should therefore represent accurately problems and their solutions. Once one or more cases are identified in the case base as very similar to the new problem, they are selected. Those cases are reused to generate a proposed forecast (i.e. normally using an adaptation technique). This solution is revised (if possible) and finally the new case (the problem and the obtained solution) is stored. Cases can also be deleted if they are proved to be inaccurate, they can be merged together to create more generalised ones and they can be modified. This is an effective way of learning that represents the general learning method used by humans.

CBR systems are able to utilise the specific knowledge of previously experienced and concrete problem situations. A CBR is an incremental learning approach because every time that a problem is solved a new experience is retained, making it immediately available for future retrievals.

Human intervention is common in the CBR life cycle, in particular case revision and retuning are often undertaken by human experts. This is a weakness for CBR systems and one of their major challenges.

Aamodt and Plaza [1] differentiate between different types of CBR systems.

- Instance-based reasoning.
- Exemplar-based reasoning.
- Memory-based reasoning.
- Case-based reasoning.
- Analogy-based reasoning.

Although all of them share their main features, each of them is more appropriate for a particular type of problem. Some do not require all the typical phases of the CBR life cycle, some are more appropriate for a particular type of cases than other, etc.

Instance Based Reasoning systems can be considered a CBR focused in problem with high numeric or syntactic connotations [1]. They are used in problems in which there are a large number of cases (instances) that represent the whole range of the domain. These large numbers of instances (which can be represented as simple feature vectors) are normally used when the general background knowledge about the problem is fuzzy or insufficient. These types of CBR are as much automated as possible to eliminate human intervention [2, 11].

2. The Oceanographic Problem

Every system, even the most complicated ones, such as the oceans of the world, can be modelled if its behaviour is fully known and understood. However the current knowledge of the ocean structure is still too weak to create a full model of its behaviour [12]. Ocean water masses are extremely dynamic. Each water mass has certain properties that differentiate it from the others, and the convergence area between different water masses can be very noisy. For example the Arctic and Antarctic convergence zones are extremely heterogeneous and very variable. These water masses have their own general characteristics that can be described and partially simulated. Artificial Intelligence (AI) approach to the problem of describing the ocean environment scene potentially offers advantages over a conventional algorithmic data processing approach, as it is able to deal with uncertain, incomplete and even inconsistent data.

The intention of this research is to develop a successful method to predict the temperature of the water several km ahead of an ongoing vessel. Different types of ANN network have been tested, ranging from unsupervised models [4] to supervised models such as typical Radial Basis Function ANN [3]. Experiments have also been carried out with statistical models such an ARIMA [4] and other simpler regression techniques. Several simple CBRs [9] have also been used.

After experimenting with over a dozen of techniques, it was found that an IBR could handle and make better use, than any other technique, of the huge amount of data needed for the development of an Universal forecasting method. Universal in the sense that the forecasting mechanism has to be capable of providing successful predictions anywhere in the ocean and at any time of the year.

The research presented in this paper is part of a major project that aims to develop a methodology for predicting the values of physical parameters in three dimensions. The project concentrates in predicting temperature of the water around a sea going vessel from data acquired in real time and also from past records of sea temperature. The characteristics of the problem under investigation require the use of an Instance-based reasoning system because:
• Cases or instances can be represented as numerical feature vectors.
• The number of cases is huge (millions of them).
• The general knowledge about the problem is weak.
• There are no prototypical cases.
• The solution of the problem requires a fully automated system.

3. Instance-based Forecasting Model

This section presents the IBR model used to forecast in real time. For simplification it only shows the technique used to forecast the temperature of the water 5 km ahead of the vessel. In the operational environment, oceanographic data (e.g., sea-surface temperature) are recorded in real time by sensors in the vessels; also, satellite pictures of the sea surface are received on a weekly basis. The satellite pictures are stored in a centralised database. Cases or instances are created by extracting vectors of temperature from the satellite pictures (see Figure 1). These vectors are used to construct the cases or instances, which are stored in the instance base (or case base).

![Satellite Picture and vector of temperature](image)

**Figure 1:** (a) Satellite Picture and (b) vector of temperature extracted from it.

A problem case is generated every 2 km using the temperatures recorded by the vessel during the last 40 km and consists of a vector of 40 temperature values, recorded at 1 km intervals (see Figure 2).

![Forecasting data](image)

**Figure 2:** Forecasting data.

During the retrieval phase, the $m$ instances that most closely match the problem case are selected from the instance-base using K-nearest neighbour matching algorithms. Each of the cases (or instances) stored in
the instance-base is defined by an Input Vector $I_k$ ($1 \leq k \leq 40$) of 40 water temperature values, a Forecast Value $F$ (representing the value of the water temperature 5km ahead of the point at which $I_k$ ($k = 40$) was recorded) and several parameters. These parameters define the geographical location of the instance, the number of times it has been retrieved, the average error in the forecast every time that the instance has been retrieved, etc. A vessel following a straight line records both $F$ and $I_k$.

The $m$ retrieved instances are adapted in real time by an artificial neural network (ANN) to obtain an initial (proposed) forecast during the reuse phase. In the revise phase the final (revised) forecast is obtained by modifying the Proposed Forecast taking into account the accuracy of the previous predictions. Each instance has associated an accumulated average error which is a measure of the average error in the previous predictions for which this instance was used to train the ANN. Then the confidence limits are calculated by averaging the accumulated average error of the $m$ instances used to train the ANN to produce the current forecast.

Learning (retaining) is achieved by storing the proposed forecast, modifying some of the constituent parameters of the cases (such as the forecasting error) and storing knowledge (ANN weights and centres) acquired by the ANN after the training and instance adaptation.

A database (Forecast Database) records all the forecasts during the last 5km and all the instances used to train the ANN to obtain these forecasts. These forecasts are eventually compared with their corresponding real values (there is a 5km lag between the point at which the forecast is made and the point for which the forecast is made). The forecasting errors are then used to modify the relevant features of the instances, to prune the instance-base and to determine the confidence limits.

4. Neuro-Adaptation Method

The adaptation of instances is one of the most complex tasks of the CBR life cycle. In this particular problem it is possible to select a number of instances that are similar to the problem instance using k-nearest neighbour metrics. Nevertheless extracting from them the best instance or creating from them a new instance successfully in not a trivial task. Several experiments have been made with averaging algorithms and statistical methods. However none of these techniques has been able to implement a universal adaptation algorithm, capable of producing successful and stable results anywhere in the ocean and at any time of the year.

The interest in developing a successful system took us to investigate into the use of Artificial Neural Networks. CBRs are flexible systems able to absorb the beneficial properties of other technologies [7]. During the last decade an increasing number of scientists have been researching in the hybridisation of CBRs and ANNs.

ANN are not especially appropriated for stepwise expert reasoning and their explanation abilities are extremely weak. Nevertheless their learning and generalisation capabilities can be very useful. Therefore they can only be used as part of CBRs in those areas that do not involve knowledge explanation and reasoning and can be used in areas involving knowledge generalisation. Learning is an excellent ability of most of the ANN, and learning forms an intrinsic part of many stages of the CBR Life Cycle. So why not using ANNs to learn to retrieve the closest instance to a particular situation, or in other words to learn to identify the closest distance between several instances. Indeed, for an ANN it should be reasonable easy in most cases to learn new instances and to learn how to generalise (adapt) an instance from a pool of instances. In our particular problem, the set of $m$ instances selected by the metrics during the retrieval step of the IBR Life Cycle is used to train an ANN, the output of which is the Proposed Forecast. The Radial Basis Function ANN [6] is retrained in real time to produce the forecast; during this step the weights and the centres of the ANN, used in the previous prediction, are adapted, based on the new training set. The goal of the ANN is to extract a generalised solution from the $m$ instances (the Proposed Forecast). The aim of the IBR system is to organise the information, and perform the initial selection. The ANN on the other hand is used to refine that initial selection and generate the final solution. The ANN acts as a function that obtains the most representative solution from a number of instances that are the ones most similar to a given situation. Both techniques successfully complement each other and the hybrid system absorbs their beneficial properties. The Radial Basis Function is extremely good at learning and generalising, IBRs in the other hand have good knowledge explanation and reasoning abilities.
5. Architecture of the RBF ANN

The algorithm developed for the construction of radial basis function (RBF) networks is a variation of the more general one described by Haykin [6]. RBF ANNs, in general, offer an interesting alternative to multi-layer perceptrons since they can be trained much faster. Training involves the placement of the localised units in the input vector space [5]. This is often done by some clustering method. Then, the weights to the output units are determined by the accuracy of the output. The delta rule determines the value of the weights since there is only one layer of connections. The complexity of this ANN depends on the difficulty to determine which centres to use and where to locate them. The architecture presented in this paper automates this process and guarantees a number of centres very close to the minimum number that gives optimum performance.

5.1 Initialisation

Initially, two vectors are randomly chosen from the input data set, and used as centres in the middle layer. All the centres are associated with a Gaussian function. The width of the Gaussian for all the functions is set to the mean length of the Euclidean distance between each centre and the 2 centres closest to it. The rationale for this idea is to overlap to some extent the set of input vectors associated to close vectors and to improve the generalisation process. This applies when there are at least 3 centres.

5.2 Centre and Weight adaptation

Presenting pairs of input and desired output vectors carries out the training. After an input vector activates every Gaussian unit to some degree, these activations are propagated forward through the weighted connections to the output units, which sum all incoming signals. The comparison of actual and desired output gives error information for this input pattern that is used in the Least Mean Square Rule.
The closest centre to each particular input vector is moved toward it by a percentage $\alpha$ of the present distance between them. By using this technique the centres are positioned close to the highest densities of the input vector data set. The aim of this adaptation is to force the centres to be as close as possible to as many vectors from the input space as possible. An adaptation of this type is particularly important for high-dimensional inputs. The delta rule is used to adapt the weighted connections from the centres to the output neurons. In particular, for each presented pair of input and desired output vectors, one adaptation step, according to the delta rule, is made.

5.3 Insertion of new units

A new unit is inserted after a number of adaptation steps. A new centre is inserted when the error does not fall for a predefined number of iterations. Calculating the average error for all the patterns used to train the network without any feedback does the error assessment. If this error does not fall any more then - for the current network size - the delta rule and the centre adaptation are unable to improve the performance of the network. In order to determine the most distant centre $P$, the Euclidean distance between each centre and each input vector is calculated and the centre whose distance from the Input data vectors is largest is chosen. A new centre is inserted in between $P$ and the closest centre to it.

5.4 Termination of training

A maximum number of units is defined in advance; however, the output error may be small enough to stop the learning process of the NN without including the maximum number of centres. Training and insertion continue until either the output error is smaller than a predefined value or the number of centres is equal to the maximum stated before training starts. This criterion makes the size of the ANN often smaller than a RBF network produced by other methods.

![Temperature AMT5 - 1997](image)

**Figure 4:** Data Profile recorded in September 1997, from a vessel going from the UK (latitude: 50.48, longitude: -1.25), to the Falkland Islands (latitude: -51.94, longitude: -58.19).

6. Results

The IBR system developed as a result of the work presented in this paper has been tested in the Atlantic Ocean in September 1997 on a research cruise going from the UK to the Falkland Islands. The cruise crossed several water masses and oceanographic fronts (areas where two water masses with different characteristics converge). Figure 4 shows the temperature values recorded by the vessel during this cruise.

The average error in the forecast was found to be 0.020 °C. Only 4.51% of the forecasts have an error higher than 0.5 °C, 8.33% higher than 0.04 °C and 32% higher than 0.020 °C. These figures indicate that the IBR system is capable of producing a forecast with an
average error of 0.020 °C and with a probability of 95.9% that the error in the forecast is smaller than 0.05 °C. Although the test has been done using a limited data set (11000 km between the latitudes 50° North and 50 °C South), eleven water masses with different characteristics were crossed including six fronts.

Figure 5 shows the absolute value of the difference between the actual temperature value of the water and the forecast value obtained using the RBF neural network for the instance adaptation in the IBR system. This graph does not take into account any improvement that may be obtained using confidence limits during the review phase of the IBR life cycle.

The prototype system used in this experiment was set up to forecast the temperature 5 km ahead of the vessel. Figure 5 illustrates the error in the forecasts over a total distance traversed of 10500 km.

A similar experiment was carried out using the data recorded by the vessel (Figure 4) during the cruise, but this time using instances obtained from satellite images recorded more than one week before that the oceanographic data set presented in Figure 4. The experiment was carried out in 25% of the data shown in Figure 4, this limitation being due to the availability of the satellite images. Also, from a computational point of view, the computing resources available were insufficient to allow these tests to be run over the whole data set. Table 1 shows how the average errors were found to vary when satellite images of different ages were used.

Table 1 shows that using satellite images which are 1 or even 2 weeks old, the accuracy of the forecast is substantially similar. Table 1 also shows that using pictures that are 3 or more weeks old, the error in the forecast can be similar to the error obtained using pictures collected exactly one year back. This is the reason why data up to one year old is kept in the database and in the instance-base. If for technical reasons (i.e., clouds covering a certain area or problems in data telecommunications) recent satellite images cannot be obtained, data recorded one year back can be used by the system and may in fact produce better results that data recorded three or four weeks back. This is due to the annual cycle of most of the water masses. However, such results can not be guaranteed, as there are also other factors that determine the pattern of ocean temperature variations.

![CBR-ANN Hybrid System](image)

**Figure 5:** Absolute value of the error using the hybrid system.
Table 1: Average error in the forecast outside the confidence limits.

<table>
<thead>
<tr>
<th>Number of Weeks</th>
<th>Average Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.020</td>
</tr>
<tr>
<td>2</td>
<td>0.024</td>
</tr>
<tr>
<td>3</td>
<td>0.034</td>
</tr>
<tr>
<td>4</td>
<td>0.048</td>
</tr>
<tr>
<td>52</td>
<td>0.033</td>
</tr>
</tbody>
</table>

Experiments were carried out to compare the performance of the IBR system with other forecasting approaches. These include standard statistical forecasting algorithms and the application of several neural network methods. The results obtained from these experiments are given in Table 2.

Table 2: Average forecasting error using the cruise data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type</th>
<th>Average Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR</td>
<td>ANN</td>
<td>0.096</td>
</tr>
<tr>
<td>RBF</td>
<td>ANN</td>
<td>0.114</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>Statistics</td>
<td>0.174</td>
</tr>
<tr>
<td>ARIMA</td>
<td>Statistics</td>
<td>0.129</td>
</tr>
<tr>
<td>CBR</td>
<td>CBR</td>
<td>0.12</td>
</tr>
<tr>
<td>CBR-ANN hybrid</td>
<td>CBR - ANN</td>
<td>0.020</td>
</tr>
<tr>
<td>CBR-ANN hybrid</td>
<td>CBR - ANN</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

The forecasting error using the IBR is smaller than with any of the other forecasting methods. In particular, the average forecasting error outside the confidence limits (0.0009) is significantly smaller than the value produced by any of the other forecasting methods. Forecasting in the adequate direction is a crucial factor in these particular systems. It is important that the IBR system do not forecast that the temperature is increasing when it is decreasing and vice versa. In this sense the average percentage of misleading forecasts is between 3% and 6% for the IBR system, depending on the data used to create the instances. These percentages increase three or more times when other models are used. The standard deviation of the error is between 0.08 and 0.106 for the IBR system depending on the data used to create the instances. These errors also increase by more than 50% when using the above mentioned models.

6. Conclusions

This paper presents a real-time forecasting system that combines an instance-based reasoning system and an artificial neural network. Forecasting the temperature 5 km ahead of an ongoing vessel is a difficult task for two reasons: the complexity of the media in which the forecast must be done and the fact that the forecast has to be done in real time.

The methodology presented here, is capable of producing a forecast within an adequate degree of accuracy and within the time constrains imposed by the real time specification of this problem. Although the accuracy of the forecast depends a great deal on the quality of the instances and on the date in which those instances have been collected, it has been shown that a good quality forecast can be obtained even with data collected one year before the forecast is done.

The ANN plays an important role in the system, it adapts the instances selected by the CBR, absorbs the knowledge of the data and generates the prediction. The radial basis functions ANN adapts its structure, in an unsupervised way, to the characteristics of the environment in which the system is immersed and acts as a function that maps a number of instances onto an instance representative of them all, taking into consideration previously learned instances. The results here presented may be extrapolated to forecast further ahead on an ongoing vessel. Obviously the further ahead the forecast is done, the more unreliable will be the forecast.
Sistemes basats en el coneixement i llenguatge natural

- The forecast can only be done while the trajectory of the vessel describes a straight line. The present system is not able to forecast while the vessel is changing its trajectory (or has changed it in the last 40 km), nevertheless, the instance structure could be changed and adapted to forecast in these situations.

- The system (as it is) can work if there are data discontinuities longer than 2 km. The forecast can be resumed by interpolating the missing data with data from both ends on the instance vector after a discontinuity. This strategy has been proved successful if the discontinuity is no longer than 5 km.

  - The system can not work in a particular area if there are not instances from that area. In that case the only solution is to use a back up system. In that respect it is believed that the FIR model [4] is the most appropriate.

The method we propose uses the ability of the CBR to index, organise and retrieve relevant data with the generalisation, learning and adaptation capabilities of the radial basis function ANN. This hybrid system combines the best properties of both connectionist and symbolic techniques. It is believed that the ideas presented in this paper and in particular the idea of combining a CBR with an ANN, in the way presented here, may be the solution for any problem that shares the same features as the one presented here.

Acknowledgements

The author wish to express his gratitude to Professor Jim Aiken and Mr Nigel Rees at Plymouth Marine Laboratory, for their support in this research.

References


L’Eixample: a New Similarity Measure for Case Retrieval

Miquel Sánchez-Marrè, Ignasi R.-Roda, Quim Comas, Ulises Cortés and Manel Poch

Abstract

This paper introduces l’Eixample distance, a new similarity measure for case-based retrieval. This measure tries to improve the competence of a case-based reasoning (CBR) agent, providing flexibility and adaptation to complex real-world application domains. This measure has been tested against some other related and well-known measures with good results. The testing has been done in real domains, with real cases.

Key words: similarity assessment, case retrieval, case-based reasoning.

1 Introduction

Over the last decade an important progress has been made in the Case-based Reasoning (CBR) field. Specially, because problems are more clearly identified, and research results have led to real applications where CBR performs well. As noticed by Althoff & Aamodt [1], in this situation it has also become clear that a particular strength of CBR over most other methods is its inherent combination of problem solving with sustained learning through problem solving experience. In CBR, similarity is used as a focusing mechanism as every important method is somehow related to similarity. And similarity measures attract the attention of many researchers in the field. In this paper, we introduce l’Eixample distance: a new similarity measure for case retrieval. This measure tries to improve the competence of a CBR-agent, providing flexibility and adaptation to real application domains. This retrieval measure has been tested against some other related and well-known measures with good results. The testing has been done in real domains, with real cases.

1.1 Wastewater treatment plants domain

The main objective of wastewater treatment is to maintain natural water systems at as high a quality level as possible. The discharge bounds established by the Catalan government are no longer intended to just prevent discharges that exceed the self-purification capacity of the streams, but also are concerned to reach some additional benefits which includes to ensure equilibrium between supply and demand through a rational use and management of water resources and the attenuation of rivers as biological corridor, that means to ensure a good life quality for animals and vegetals living into the water.

Wastewater coming from different municipal uses contains a wide variety of contaminants. Among them, the most commonly found in MWW are suspended solids, biodegradable organics and nutrients.

Wastewater treatment processes [2] are usually designated as pre-treatment, primary and secondary, depending on the degree of purification. See figure 1. Pre-treatment includes simple screening to remove large floating objects and grit removal. Primary treatment utilise physical and chemical processes, that is sedimentation and coagulation-flocculation to settle from wastewater most of the suspended solids. Although pre-treatment and primary effluent gets removals of about 35% BOD (a global organic messier) and 60% of the suspended solids, still has enough organic matter to cause dissolved oxygen depletion problems and enough nutrients (nitrogen and phosphorus) to accelerate eutrophization in the rivers.

1 L’Eixample is an architectonic neighborhood of Barcelona, characterized by its square city-blocks.
A secondary treatment involving the microbial oxidation of wastes is required to meet discharge permits. Such biological treatment mimics nature by using microorganisms to oxidise biodegradable organics into stabilized, low-energy compounds. The most commonly used approach of secondary treatment is activated sludge process. When properly designed and operated, secondary treatment facilities remove about 90% BOD and 90% SS. Recently, secondary treatment also includes nutrient removal, if necessary, depending on the effluent requirements.

1.2 WWTP operation and control problem

Correct control and operation of WWTP is not an easy and well established task. Classical controls methods, based on mathematical modelling, shows some limitations when trying to control the activated sludge process, mainly when the plant is not working under the ideal state. Some of the aspects that difficult the WWTP operation and control are the following:

- the biological nature of the process, involving the presence of a complex mix of microorganisms,
- the dynamic state of the process,
- the lack of on-line sensors and signals (there are only a few sets of variables able to get on-line to be useful for a classical control model),
- the ill-structure of the WWTP domain (difficulty on the description of cause-effect relationships between different processes that takes place between microorganisms and the different substrates),
- the use of subjective information,
- the relevance of many qualitative variables and human's experience,
In this paper we address the problem of similarity assessment between the input case and the retrieved cases from the case library.

2. Case Retrieval and Similarity Assessment

The task of retrieving cases in the case library is slightly more difficult than typical retrieval in databases. In database systems the recalling algorithms use an exactly matching method, whereas in a case library retrieval, because the very nature of the structure, a partial-matching strategy should be used. A retrieval method should try to maximise the similarity between the actual case and the retrieved one(s). And this task implies most of the time the use of general domain knowledge.

The retrieving process of a case (or a set of cases) from the system’s memory strongly depends on the case library organisation. Major case library structures are flat memories or hierarchical ones. Flat memories have an intrinsic problem of bad performance in time, so that the retrieval time is proportional to the size of the case library. Hierarchical memories are very effective in time retrieval because only a few cases are considered for similarity assessment purposes, after a prior discriminating search in the hierarchical structure. Although sometimes could not reach optimal cases because is exploring a wrong area of the hierarchy.

The retrieval process in hierarchical case libraries, usually consists of two main substeps:

- Searching the most similar cases to the new case: the goal of this stage is recalling the most promising cases—given that the sub-system has a goal and therefore the relevance of the cases depends upon that goal—based on using some direct or derived features of the new case as indexes into the case library.

- Selecting the best case(s): the best case(s) among those ones collected in the previous step are selected. Commonly, this selection is made by means of a case ranking process through a similarity or distance function. The best retrieved case is the closest one (most similar) to the new case.

Selecting the best similar case(s), it is usually performed in most Case-based reasoning agents by means of some evaluation heuristic functions or distances, possibly domain dependent. They are usually named as nearest neighbour (NN or k-NN).
algorithms [23]. The evaluation function usually combines all the partial-matching through a dimension or attribute of the cases, into an aggregate or full-dimensional partial-matching between the searched cases and the new case. Commonly, each attribute or dimension of a case has a determined importance value (weight), that is incorporated in the evaluation function. This weight could be static or dynamic depending on the Case-based reasoning agent purposes. Also, the evaluation function computes an absolute match score (a numeric value), although a relative match score between the set of retrieved cases and the new case can also be computed.

Most Case-based reasoners such as REMIND [24], MEDIATOR [25], PERSUADER [26], etc., use a generalised weighted distance function (NN) such as,

\[
\text{dist} (C_i,C_j) = \sum_{k=1}^{n} w_k \cdot \text{attr-dist} (C_{ik}, C_{jk}) / \sum_{k=1}^{n} w_k
\]

but this kind of distance functions, sometimes, does not capture the significant differences among the attributes, because they are a lineal weighted combination of one-dimensional distances, that can be lost as the number of retrieved cases increases.

3 Case retrieval in our CBR approach

We have assumed that a non-linear weighted multidimensional distance function would be required for a better matching performance. After a competence study, we have developed a normalised exponential weight-sensitive distance function, that we have called \( L'\text{Eixample distance} \). It takes into account the different nature of the quantitative or qualitative values of the lineal (ordered) attributes, and the modalities of categorical (not ordered) attributes.

\( L'\text{Eixample distance} \) is sensitive to weights, in the sense that, for the most important attributes, that is weight \( > \alpha \), the distance is computed based on their qualitative values, i.e. maintaining or amplifying the differences between cases, and for those less relevant ones, that is weight \( \leq \alpha \), the distance is computed based on their quantitative values, i.e. reducing the differences between cases. \( L'\text{Eixample distance} \) used to rank the best cases is:

\[
d (C_i,C_j) = \sum_{k=1}^{n} c^{W_k} \cdot d (A_{ki},A_{kj}) / \sum_{k=1}^{n} c^{W_k}
\]

where,

\[
d (A_{ki},A_{kj}) = qtv (A_{ki}) - qtv (A_{kj}) / (\text{upperval} (A_k) - \text{lowerval} (A_k))
\]

if \( A_k \) is an ordered attribute and \( W_k \leq \alpha \)

\[
d (A_{ki},A_{kj}) = qtv (A_{ki}) - qtv (A_{kj}) / (\#\text{mod} (A_k) - 1)
\]

if \( A_k \) is an ordered attribute and \( W_k > \alpha \)

\[
d (A_{ki},A_{kj}) = 1 - \delta_{qtv (A_{ki}),qtv (A_{kj})}
\]

if \( A_k \) is a non-ordered attribute and,

\( C_i \) is the case i; \( C_j \) is the case j; \( W_k \) is the weight of attribute \( k \); \( A_{ki} \) is the attribute \( k \) in the case i; \( A_{kj} \) is the attribute \( k \) in the case j; \( qtv (A_{ki}) \) is the quantitative value of \( A_{ki} \); \( qtv (A_{kj}) \) is the quantitative value of \( A_{kj} \); \( A_k \) is the attribute \( k \); \( \text{upperval} (A_k) \) is the upper quantitative value of \( A_k \); \( \text{lowerval} (A_k) \) is the lower quantitative value of \( A_k \); \( \alpha \) is a cut point on the weight of the attributes; \( qtv (A_k) \) is the qualitative value of \( A_{ki} \); \( qtv (A_{kj}) \) is the qualitative value of \( A_{kj} \); \( \#\text{mod} (A_k) \) is the number of modalities (categories) of \( A_k \); \( \delta_{qtv (A_{ki}),qtv (A_{kj})} \) is the \( \delta \) of Kronecker.

Our CBR agent computes the distances between each case retrieved by the discrimination tree search and the new case. Then, it ranks the list of cases by increasing distance. So, the first case in the list is the closest case to the new case. The output of this process is a comparative table of all the retrieved cases and the new case, describing the distance and the attribute’s values.

4 Experimental testing

To test the performance of the new proposed similarity measure, we have done a comparative study of different distances, mainly derived from the Minkowski’s metric:

\[
d(x_i, x_j) = \left( \sum_{k=1}^{n} |x_{ik} - x_{jk}|^r \right)^{1/r}
\]

All the distances were tested in three different ways depending on which value of the ordered attributes were used in the computation of the distance value: discrete or quantitative values, continuous or quantitative values, and weight-sensitive or discrete/continuous values depending on the weights of the attributes. The 3-order or superior order Minkowski’s distance were disregarded after an initial study of performance. Thus, the final comparison study has been performed among the distances shown in
table 1, normalised over the interval [0, 1]. The general formula for its computing is the following:

\[
d(C_i, C_j) = \left( \sum_{k=1}^{n} \text{weight}_k \cdot |d(A_{ki}, A_{kj})|^{r_k} \right)^{1/r} / \sum_{k=1}^{n} \text{weight}_k
\]

where for not ordered attributes, their contribution to the distance is,

\[
d(A_{ki}, A_{kj}) = 1 - \delta qv(A_{ki}), qv(A_{kj})
\]

<table>
<thead>
<tr>
<th>similarity Measure</th>
<th>r</th>
<th>weight</th>
<th>(W_k )</th>
<th>(q_{\text{qv}}(A_{ki}) - q_{\text{qv}}(A_{kj})) / ((#\text{mod}(A_k) - 1))</th>
<th>(W_k \leq \alpha)</th>
<th>(W_k &gt; \alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disc. Manh. (MD)</td>
<td>1</td>
<td></td>
<td></td>
<td>([a])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disc. Eucl. (ED)</td>
<td>2</td>
<td></td>
<td></td>
<td>([a])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disc. Exp.-w. Manh.(EMD)</td>
<td>1</td>
<td></td>
<td></td>
<td>([a])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cont. Manh. (MC)</td>
<td>1</td>
<td></td>
<td></td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cont. Eucl. (EC)</td>
<td>2</td>
<td>(e)</td>
<td>([b])</td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cont. Exp.-w. Manh. (EMC)</td>
<td>1</td>
<td>(e)</td>
<td>([b])</td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W.-sens. Manh. (MW)</td>
<td>1</td>
<td></td>
<td>([a])</td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W.-sens. Eucl. (EW)</td>
<td>2</td>
<td>(e)</td>
<td>([b])</td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W.-sens. Exp.-w. Manh. (EIX)</td>
<td>1</td>
<td>(e)</td>
<td>([b])</td>
<td>([b])</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Similarity measures tested

4.1 WWTP description

Both examples used, Girona (70000 inhabitants) and Lloret (20000 inhabitants in winter and 150000 in summer) – a touristic resort located in Costa Brava (Catalonia) – are municipal WWTP and have similar configuration based on primary treatment of raw water followed by the activated sludge process. The goal is to remove organic load and suspension solids from an inflow of 45000 m³/day in the Girona’s WWTP and an inflow of 13000 m³/day in winter or 45000 m³/day in summer in the Lloret’s WWTP.

Both plants collect a large amount of information corresponding to an exhaustive characterisation of the plant to control the process. It has been distinguished between the quantitative and the qualitative variables (only available in Lloret). The quantitative information includes analytical results of water and sludge quality at different sample points, together with on-line signals coming from sensors. Qualitative data includes some subjective information about plant performance, quality of biomass, and daily microscopic examinations.

In the first domain for evaluating the performance of the different distances, the case library was initialised with a representative set of cases obtained from a previous classification of 396 real cases, using Linneo’, an unsupervised classification tool [27]. Those cases are real daily working situations of the Girona’s WWTP coming from the 1995/1996 period. In the second domain, the case library was initialised with a representative set of cases obtained from a previous classification of 234 real daily working situations of the Lloret’s WWTP coming from the the 1996/1997 period.

In both domains, 10 cases were used as a training set. These cases were a representative sample of all possible cases in the domain to assure the validity of the results. For each case, the case library was searched to retrieve the most similar cases to it, using all the similarity measures (10). The experts evaluated each retrieval table for each possible distance (100 for each domain), giving the results described in the next section.

5 Evaluation

The experts have used two evaluation criteria, to validate the obtained results. The first one is to compute the percentage of times that the optimal
case was retrieved using a concrete similarity measure [28]. The results are shown in the table 2.

From the experiments done and from the table 2, it can be observed that in the Girona’s WWTP (1) the discrete distances are worse than the continuous, and these are worse than the weight-sensitive distances. But in the Lloret’s WWTP (2) happens the oposed fact. We think that this is due to the fact that in Girona’s WWTP there are only ordered continuous attributes while in the other domain, there are some important non ordered categorical attributes. Also, it seems that Manhattan distances are slightly better than Euclidean ones in the first domain (1), and the oposed fact in the second domain (2). At a first look, the Exisample distance – Weight-sensitive Exponential-weighted Manhattan – seems not to substantially improve the other weight-sensitive distances, but another deeper and more accurate study must be done to extract some conclusions.

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>% Optimal retrieval Lloret’s WWTP</th>
<th>% Optimal retrieval Girona’s WWTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disc. Manhattan (MD)</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Disc. Euclidean (ED)</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>Disc. Exp.-w. Manhattan (EMD)</td>
<td>80</td>
<td>70</td>
</tr>
<tr>
<td>Cont. Manhattan (MC)</td>
<td>50</td>
<td>90</td>
</tr>
<tr>
<td>Cont. Euclidean (EC)</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>Cont. Exp.-w. Manhattan (EMC)</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>Weight-sens. Manhattan (MW)</td>
<td>60</td>
<td>90</td>
</tr>
<tr>
<td>Weight-sens. Euclidean (EW)</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td>Weight-sens. Exp.-w. Manhattan (EIX)</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 2. Percentage of optimal retrieval in both domains

A second criterion has been considered by the experts as much important as the first one, and recognised by some researchers as [29]. It can make clearer the behaviour of L’Exisample distance. Another feature to take into account is not only if the optimal case has been retrieved, but if the retrieval ranking and distance evaluation of the other retrieved cases are as good as expected. In the figure 2 there are the comparative results in the first domain (1) –for an example— after retrieving the most similar cases for a concrete input case. The results are the computed distances, using all the similarity measures, for the first three more similar cases.

The figure 3 outlines that L’Exisample distance really is the best measure, because the optimal case and the second nearest case are assessed as the most similar to the input case while the other distances are not as good as it. On the other hand, the third case, has been signalled by the experts as a very different case from the input one, and L’Exisample distance is one of the best measures in distinguishing that case.

Figure 3. Optimal retrieval ranking in Girona’s WWTP

In the same way the second domain (2) was tested. The comparative results are depicted in the figure 4 after retrieving the most similar cases for a concrete input case. The results—for an example—are the computed distances, using all the similarity measures, for the first five more similar cases.

The figure 4 confirms that L’Exisample distance really is the best measure, because the optimal case is assessed as the most similar to the input case while the other distances are not as good as it. On the other hand, the second and third cases, have been signalled by the experts as similar cases but slightly more different than the optimal case, and L’Exisample distance is one of the best measures discovering this fact. Moreover, the fifth case has been also pointed out by the experts as a very different case from the input one, and again, L’Exisample distance is one of the best measures in distinguishing it from the optimal case.

Other conclusions can be stated from these figures and other experiments done. The continuous distances are not as good as it seemed from the comparison of the optimal retrieval percentages, because they cannot easily distinguish the worse cases from the optimal ones. Continuous distances are best suited for domains, such in (1), described with many ordered quantitative attributes. Also, the Euclidean distances are best suited for domains, such in (1), having zero or few non-ordered categorical attributes, while the Manhattan distances...
are best suited for domains, such as (2), having more non-ordered categorical attributes. Therefore, \( L'\text{Ex}\text{ample} \) distance derived from Manhattan distance, and combining discrete and continuous assessing of one-dimensional matching between attributes gets an improved performance for case retrieval.

![Figure 4. Optimal retrieval ranking in Lloret's WWTP](image)

### 6 Conclusions and future work

Most existing approaches to similarity assessment do not have an extensive comparison using several measures. Also most of them work assuming that variables or attributes used to assess the similarity are independent. But they some how reflect the expert’s knowledge and could be considered as a declarative bias and therefore \( L'\text{Ex}\text{ample} \) distance reflects this fact.

The main results of this paper are that on average the results obtained using \( L'\text{Ex}\text{ample} \) distance improve the results of other similarity measures for case retrieval in a CBR agent environment. We have done more experiments that have confirmed this fact. We have to stress that the case libraries used to test all the measures are composed by real cases from real applications.

Our CBR agent, is being improved in several aspects that constitute some future work: doing more tests with other domains extracted from the UCI Machine Learning Repository of data bases, to confirm these results, although we have the problem that we need the expert knowledge to weight the attributes and to set a discrimination ordering of them, and this knowledge is not encoded in the the UCI Machine Learning Repository of data bases. Another point is solving some discretisation problems that can provoke unsuccessful searches of optimal cases in hierarchical case libraries, by means of some re-exploration techniques, and by means of what we have named as meta-cases [6], that can bias the searching in a set of case libraries.

### Acknowledgements

This research is been partially supported by the Junta de Sanjament de la Generalitat de Catalunya and the Spanish CICYT projects TIC96-0878 and AMB97-0889. Also the authors wish to acknowledge the cooperation of managers of the Girona’s and Lloret’s wastewater treatment plants.

### References


Sistemes basats en el coneixement i llenguatge natural


Model d’entrada lèxica verbal i implementació en una base de coneixement*

Roser Morante¹, Glòria Vázquez², Ana Fernández³, Irene Castellón¹ i M. A. Martí¹

¹Secció de Lingüística General
Departament de Filologia Romànica
Universitat de Barcelona
{morante, castel, marti}@lingua.fil.ub.es

²Departament d’Anglès i Lingüística
Universitat de LLeida
gvazquez@dal.udl.es

³Universitat Autònoma de Barcelona
a_fernandez@minerva.uab.es

Resum

En aquest article es presenta un mètode d’anàlisi dels verbs que tracta d’explicitar el coneixement que es requereix tant per al procés de comprensió o anàlisi com per al de síntesi o generació del lenguatge. El treball es basa en el estudis de tres aspectes bàsics per a la caracterització de les entrades verbals: el semàntic (components de significat), el sintàctic (alternances de diàtesis) i l’estructura esdevenimental. L’objectiu de la recerca és construir una base de coneixement lèxic per ser usada en els sistemes de processament de lenguatge natural. Amb aquesta finalitat s’ha dissenyat un model d’entrada lèxica, que es representarà la informació concreta.

Pàrraules clau: entrada lèxica verbal, components de significat, alternances de diàtesis, estructura esdevenimental, classe semàntica, estructures de trets, base de coneixement lèxic.

1 Introducció

El marc general en què s’inserta aquesta investigació¹ és la construcció d’una base de coneixement lèxic verbal multilingüe, que inclou informació morfològica, sintàctica i semàntica sobre un ampli nombre de verbs, en la línia de l’obra de Levin [8]. Des d’un punt de vista metodològic, la primera qüestió plantejada ha estat la definició del model d’entrada lèxica, tant d’un punt de vista formal com teòric.

Pel que fa a la primera qüestió, la representació de la informació, s’introduïxen uns requisits estrictes d’explicititat del coneixement, de manera que el model proposat sigui tractable computacionalment; és a dir, la informació que s’especifica en cada entrada està estricament codificada i guardada en un format recuperable per qualsevol sistema de processament del lenguatge natural (Copestake [3]).

Quant al fonament teòric del model, aquest té un caràcter ecològic. En primer lloc, cal destacar l’obra de Levin [8]. Aquesta autora, com al-

¹Aquest treball ha estat realitzat gràcies als recursos obtinguts amb els projectes PB-94 0830 de la DGICYT, l’acció APC-96 0125 de la DGICYT, ITEM TIC-96 1243-C03-02, el projecte Llevec de la Universitat de Lleida i les beques predoctorals del Comissionat per a Universitats i Recerca de la Generalitat de Catalunya (ref. PI 96/0008 i PI 97/00306 PG).
tres lingüistes en el camp de la semàntica lèxica (Levin i Pinker [9] i Levin i Rappaport [10]), parteix de la hipòtesi que la interrelació entre sintaxi i semàntica es manifesta en les característiques de les classes verbals: els membres de cada classe no solament comparteixen els mateixos components de significat, sinó que també mostren un comportament sintàctic semblant.

Així doncs, per analitzar adequadament el comportament verbal cal estudiar els predicats des d’aquestes tres perspectives: components de significat, alternances de diàtesi i estructura esdevenimental. Partint d’aquesta hipòtesi s’han utilitzat aquests tres aspectes com a criteris per a la classificació dels verbs estudiat. A més, prenent com a punt de referència el lèxic generatiu de Pustejovsky [13], es parteix del postulat que el conjunt d’aquesta informació ha d’incorporar-se en una entrada lèxica.

Tant el model teòric adoptat com el desenvolupament del lèxic computacional multilingüe implica que es defineixin com es conceben cada un d’aquests mòduls i com interactuen.

1.1 Mòduls

1.1.1 Components de significat

La definició dels components de significat en aquest marc no es basa a determinar el tipus de relació semàntica que s’estableix entre el verb i els arguments, sinó que es considera que el verb i els seus arguments formen una unitat de caràcter conceptual, que pot expressar-se mitjançant diversos recursos. Aquesta concepció dels components es troba en la línia de la proposta de Talmy [18] per a l’anàlisi dels verbs de moviment. Aquest autor determina uns components i estableix diferents recursos de realització: es poden expressar amb constituents, en el lexema verbal o poden quedar sobreentèsos.

En la proposta que aquí es presenta es defineixen els components de significat a tres nivells. En el primer se situen els components necessaris per a la realització d’una peça lèxica verbal, que són els que fan referència a les coordenades espaciotemporals i la manera en què es realitzan els esdeveniments. Aquests components es caracteritzen per ser optatius pel que fa a la seva realització sintagmàtica.

En el segon nivell se situen els components que són necessaris per a la realització del predicat com un esdeveniment: l’entitat implicada i l’iniciador. L’entitat implicada pren unes determinades característiques en funció de la classe verbal (per exemple, en el cas dels verbs de trajectòria, aquesta entitat es correspon amb l’entitat desplaçada, mentre que en els de canvi, amb l’entitat afectada). L’iniciador fa referència a l’element que inicia l’esdeveniment, i pot coincidir o no amb l’entitat.

Per últim, es poden diferenciar classes verbals específiques en funció del requeriment d’altres components de significat: en el cas dels verbs de trajectòria, per exemple, es defineix el component trajectòria i, per als verbs de canvi, el component canvi. Aquests components es consideren definidors de les classes, ja que són els que les diferencien de la resta.

Com es veurà en l’apartat 3, en la BCL aquest tipus d’informació es codifica en el tret M-COMP, definint a través de l’atribut COMP, que té com a possibles valors la llista de components de significat predefinida (iniciador, entitat...) i un índex que permet la coïndexació de cada un d’ells amb l’estructura argumental.

1.1.2 Alternances de diàtesi

Les alternances de diàtesi s’entenen com les diferents possibilitats d’expressió sintagmàtica que té un predicat pel fet de focalitzar unitats diferents de l’estructura esdevenimental o dels components de significat. Cada una de les possibilitats d’enfocament dóna lloc a diverses oposicions de significat, encara que sempre es manté la mateixa accepció del verb.

Darrere aquesta definició hi ha la concepció que les oracions són representacions semàntiques d’un verb més uns participants (components), la relació dels quals forma el significat del predicat. En cada representació semàntica els participants poden realitzar-se sintagmáticament, poden aparèixer lexicalitzats o poden no realitzar-se, és a dir, estar sobreentèsos o infraespecificats. Aquest fenomen tindrà repercussions en el comportament sintàctic del verb.

En una primera aproximació (Castellón et al. [2], Fernández i Martí [4]), s’han considerat quatre possibles oposicions semàntiques basades en criteris d’enfocament, que es presenten seguidament. Per a cadascuna se n’especificarà l’esquema bàsic i se n’indicaran alguns exemples. Les tres primeres fan
Sistemes basats en el coneixement i llenguatge natural

referència als participants, la quarta, a l'estructura esdevenimental:

1. Canvi d'enfocament genèric: aquesta oposició implica que sintagmáticament es produeixi un creuament argumental, sense pèrdua de cap argument. L'esquema corresponent és el següent:

<table>
<thead>
<tr>
<th>SN1</th>
<th>V</th>
<th>SN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN2</td>
<td>V</td>
<td>SP1</td>
</tr>
</tbody>
</table>

(1)
El sol irradia calor
La calor irradia del sol

2. Canvi d'enfocament amb pèrdua d'informació: en aquest grup s'inclouen les diàtesis que presenten una alternança entre una estructura transitiva i una d'intransitiva. L'estructura intransitiva té com a subjecte el sintagma nominal en posició d'objecte en la transitiva. L'estructura transitiva pot ser de caràcter agentiu o causatiu. En el primer cas alternar amb una construcció passiva, en el segon, amb una d'anticausativa. No obstant això, de vegades la construcció intransitiva admet la presència d'un sintagma preposicional que procedeix dels subjectes de la transitiva i que té un caràcter opcional. Els esquemes corresponents són els següents:

<table>
<thead>
<tr>
<th>SN1</th>
<th>V</th>
<th>SN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN2</td>
<td>V</td>
<td>(SP1)</td>
</tr>
</tbody>
</table>

(2a)
En Joan va vendre moltes flors
Es van vendre moltes flors

(2b)
El vent va tancar la porta
La porta es va tancar (a causa del vent)

3. Generalització: consisteix en la no-especificació d'un argument pel fet que és irrelevat del punt de vista comunicatiu, de manera que deixa d'enfocar-se un element determinat. La construcció intransitiva no implica creuament argumental. Un cas específic d'aquest tipus d'alternança és el dels verbs que incorporen l'objecte en la forma lèxica, el qual pot aparèixer en la sintaxi quan està especificat. L'esquema corresponent és el següent:

<table>
<thead>
<tr>
<th>SN1</th>
<th>V</th>
<th>SN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN1</td>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>

(3a)
L'Imma baixa
L'Imma baixa les escales

(3b)
En Pere beu
En Pere beu un refresc / *una beguda

4. Oposició aspectual: s'alterna l'expressió d'un esdeveniment i la d'un estat. Els recursos de què disposa cada llengua per expressar aquesta oposició són diversos. En general, un predicat que expressa un esdeveniment pot passar a ser estatiu quan s'usa en un temps no marcat, preferentment amb un modificador.

D'alta banda, els esdeveniments que participen en l'oposició de 'canvi d'enfocament amb pèrdua d'informació' tenen una lectura estatiu quan s'usen en un temps imperfecció (4c):

(4)
a. La meva mae va tallar la carn per al rostit
b. Es va tallar la carn per al rostit
c. La carn es talla amb facilitat

Aquesta classificació de les alternances de diàtesis en quatre oposicions és provisional i oberta, per la qual cosa no es descarten modificacions posteriors. De moment, però, permet progressar en l'elaboració del model d'entrada lèxica.

En la base de coneixement, la informació sobre les diàtesis es representa en el tret SINTAX, que fins ara sols conté l'estructura de trets SUBCAT. A SUBCAT es defineix la llista de possibles complements, per a cadascun dels quals s'expressa el correlat amb l'estructura argumental i la categoria sintàctica que el realitza.

1.1.3 Estructura esdevenimental

Tal com s'ha observat (Pustejovsky [13], Rosen [15] i Ramchand [14]), les diferències en l'estructura esdevenimental dels predicats intervenen en el comportament sintàctic d'aquests. Per aquest motiu, es considera fonamental tractar aquest aspecte, ja que permet donar compte de fenòmens que, si no, quedarien inexplicats.

Per al tractament correcte d'aquest nivell és bàsic tenir en compte que les peces lèxiques no es poden
Sistemes basats en el coneixement i llenguatge natural

considerar àïlladament, sinó que han de tractar-se sempre d’acord amb el context lingüístic en què apareixen. Així, un verb no expressa un estat o un esdeveniment en si mateix, sinó que depèn del temps verbal en què s’usa i del tipus de complement que l’acompanyen:

(5)

a. El vidre es treu amb facilitat (estat)
b. El vidre es va treure (esdeveniment)

Com a tipologia de predicats es presenta una versió modificada de la proposta de Vendler [20]. Següent aquest autor s’estableixen dos tipus bàsics, els estats i els esdeveniments, que, al seu torn, se s’alí Altaixen en esdeveniments simples (processos) i esdeveniments complexos (Verkuyl [21] i Pustejovsky [13]).

S’entén per estat una representació semàntica que descriu les propietats d’una entitat en un moment determinat:

(6) El nen ja té dues dents

(7) En Joan escriu

Els esdeveniments simples, per la seva banda, descriuen accions o activitats que s’estan realitzant en un lloc, en un temps i d’un mode determinats, independientment de si són o no puntuals:

(8)

a. El caçador va disparar l’animal
b. El caçador va còrrer l’animal

Els esdeveniments complexos expressen un procés i un estat o bé dos processos. Un exemple del primer tipus és el dels verbs de canvi. La relació que s’estableix entre cada subcomponent és de diferents tipus, com veurem més endavant.

Per a l’expressió d’aquest mòdul en la base de coneixement, s’ha optat per implementar la proposta de Pustejovsky [13]: es declaren les subestructures implicades, les relacions que en puguin establir entre si i quin és l’esdeveniment en què s’està focalitzant.

2 Proposta de classes


La proposta de classificació agrupa les classes enumerades en dos grups principals, els verbs de trajectòria (v. ap. 2.1) i els de canvi (v. ap. 2.2). Per als verbs de cadascuna d’aquestes classes s’especifiquen els components que comparteixen les alternances de diàtesis, així com el tipus d’estructura esdevenimental. La caracterització semàntico-sintàctica verbal s’estableix a un nivell més elevat que en l’obra de Levin [8], ja que l’abstracció es considera imprescindible per extreure generalitzacions explicatives sobre les classes semàntiques i obtenir una classificació que trascendeixi petites agrupacions de peces lèxiques intuitivament relacionades.

2.1 Els verbs de trajectòria


1. Verbs de transferència:

(a) de possessió: algú dóna alguna cosa a algú

Aquesta proposta sols ha estat contrastada amb un nombre de dades reduït, per la qual cosa està subjecta a modificacions.

2 Els verbes d’una coincidència entre aquesta classe verbal i els verbs que Shank [16] defineix com a caracteritzats per les primitives ATRANS, MTRANS i PTRANS.
2. Verbs de moviment:

(a) autònim: algú va a algun lloc
(b) no autònim: algú posa alguna cosa en algun lloc

El component específic d’aquesta classe de verbs és la trajectòria (T), entenent que aquests verbs expressen un canvi de situació d’una entitat (E), promogut per un iniciador (I), des d’un punt origen (To) passant per un punt/s punt/s intermedi/s (Tm) per arribar a una destinació (Td). L’origen, la destinació i qualsevol dels punts intermedis poden ser físics o abstractes i són els subcomponents de la trajectòria.


La característica sintàctica que defineix aquesta classe és l’acceptació de l’alternança de trajectòria (Morante i Vázquez [11]). Aquesta alternança consisteix en la possibilitat d’expressar, com a mínim, un subcomponent de la trajectòria. El constituent que l’expressa ocupa la posició de complement verbal i es correspon normalment amb un SP o, a vegades, amb un SN. Normalment la realització d’aquest complement és opcional i en la majoria dels predicats expressa la destinació (10a, b). L’expressió de l’origen es dóna, per exemple, en el cas d’alguns verbs de transferència (10c) i de moviment (10d).

(10)

a. En Manel (els) explicava contes
b. L’Agnès va guardar el talismà (en la cova)
c. La Mònica va comprar el pis (a una agència)
d. En Joaquim va sortir de Madrid.

Alguns verbs de moviment (11) requereixen obligatòriament la presència del sintagma que expressa la destinació, si no hi ha informació pragmática que completi la interpretació. En els predicats que requereixen l’expressió del component trajectòria, si focialten 4 alguns subcomponent, serà aquest el que es realitzi sintàgmatícamen:

(11)

a. En Joan va posar els expedients (en l’arxiu)
b. En Gabriel va anar (a Saragossa)

Una característica semàntica de tots els membres de la classe és que, encara que no admeten la incorporació 5 dels components definitoris (iniciador, entitat i trajectòria), sí que admeten la d’altres components, com l’instrument (12) o la manera (13):

(12)

a. La Carolina va [telefonar] a tota la família
b. La grua va [remolcar]; el cotxe fins al taller

(13)

a. El lladre va [arribar] a la bossa a la senyora
b. En la sèrie els actors van [cantar] un sol cop, en el primer episodi

Quant a l’estructura esdevenimental aquests verbs presenten, normalment, una estructura d’esdeveniment simple (procés). Aquest procés es correspon amb la trajectòria que recorre l’entitat d’un punt a un altre. Hi ha alguns verbs que expressen sols una de les fases del procés, normalment la final (collocar, internar), mentre que d’altres es refereixen al procés en si (caminar).

En el primer cas, es tracta de verbs que tradicionalment s’han considerat achievements. Els verbs del segon tipus pertanyen als processos pròpiament dits (activities en la terminologia de Vendler [20]) i als accomplishments.

La diferència entre aquests dos tipus de processos, parciaus i totals, respectivament, té repercussions importants en la sintaxi: per una banda, els verbs que expresen processos parciaus sols poden expressar un subcomponent de la trajectòria, depenent de la fase que estiguin descritva (14a), i, per l’altra, no accepten les preposicions fins a, des de, per, entre (14b); els verbs que expressen processos totals (15) poden expressar normalment més d’un subcomponent i, depenent del verb, accepten les preposicions anteriors.

(14)

a. Ha collocat [els llibres] [del calaix] [al terra] 6
b. Ha collocat els llibres fins al terra

(15)

a. Va des de casa scva fins al treball
b. Anirà a Tarragona per l’autopista

4 Es defineix la focalització a partir del concepte proposat per Iegam [7]. Així, alguns dels predicats definit mitjançant components de significat compostos per subcomponents admeten que se n’emfatitzi algun.

5 Per incorporació s’entén l’expressió d’un component en la part lèxica (Talmy [17]). Quan aquest fenomen es produïx a través d’un procés morfològic de derivació parlarem de lexicalització, que és un tipus d’incorporació.

6 Aquesta frase seria gramatical amb una altra estructura sintàctica, on el primer SP fos complement del nom i no del verb: Ha collocat [els llibres [del calaix]] [al terra]
Per últim, els verbs del tipus *intercanviar* o *conversar* formen un grup especial en relació amb l'estructura esdevenimental ja que presenten una estructura esdevenimental complexa, formada per dos o més processos que tenen lloc a la vegada en sentits oposats.

### 2.2 Els verbs de canvi


Els verbs de canvi es defineixen com aquells l'objecte dels quals és afectat per l'acció del verb, iniciada per un element causal. Com a conseqüència d'aquesta afectació l'objecte pateix un canvi, que pot ser físic o psicològic (Vázquez et al. [19]).

Els components de significat bàsics d'aquests verbs són l'entitat, el canvi i l'iniciador. L'entitat es refereix a la persona o l'objecte que resulta afectat per l'acció del verb, a diferència dels verbs de trajectòria, en què l'entitat desplaçada no queda modificada.

El canvi és el component semàntic específic d'aquesta classe i es refereix a l'estat que resulta de l'acció del verb: l'entitat es troba en un estat (indefinit) i passa a un altre (estat resultat) més o menys permanent en dur-se a terme l'esdeveniment. Aquest és el cas de (16a), on els nens és l'entitat que passa d'un estat no específic a un altre d'avortament. En la majoria dels verbs d'aquesta classe, com *avorrir*, el component canvi està incorporat en la peça verbal i no sol expressar-se sintagmáticament. La frase (16b) exemplifica el cas contrari.

(16)

a. Aquesta pel·lícula avorreix els nens
b. La bruixa ha transformat el príncep en granota

Finalment, el tercer component de significat és l'iniciador (*la pel·lícula* i *la bruixa* en les frases anteriors). Mitjançant aquest component s'expressa la causativitat, concepte que es defineix en termes de la no rellevància del caràcter voluntari. Aquest component, com veurem, sempre pot realitzar-se sintagmáticament.

La diàtesi determinat per definir els verbs de canvi és l'anticausativa, en què l'entitat ocupa la posició argumental de subjecte (17a). Aquesta estructura alterna, en molts casos, amb la variant transitiva, en què l'iniciador ocupa la posició argumental de subjecte i l'entitat la d'objecte (17b). Algunes verbs requereixen la perifrasi amb el verb *fer* (17c) per expressar la variant causativa:

(17)

a. L'aigua bull
b. El cuiner bull l'aigua
c. Les seves paraules van fer envermellir la Maria

c. Els verbs d'aquesta classe són, des del punt de vista de l'estructura esdevenimental, el resultat de la combinació de dos tipus esdevenimentals, procés i estat (és a dir, un esdeveniment complex). Cada un d'aquests tipus expressa una relació semàntica en què intervenen components diferents: en el procés, l'iniciador i l'entitat, i en l'estat, el canvi. Així doncs, en frases com (18) es distingeix un procés previ a l'estat final:

(18)

a. Aquesta notícia va sorprendre la Núria
b. L'aplicació directa de la calor va assecar la roba en mitja hora

c. Els verbs de canvi poden focalitzar en l'estat resultant, tant en els casos en què aquest se situa a l'inici de l'acció (18a) com en els que se situa al final (19b):

(19)

a. La Núria se sorprén ràpidament
b. La roba es va assecar en mitja hora

c. En els exemples dels verbs *sorprendre* i *assecar* es focalitza el procés o l'estat a través d'una determinada construcció per a cada cas, l'anticausativa (18) i la causativa (19), respectivament.

---

7V. Morante i Vázquez 98 [11] per a una explicació de la doble classificació que presenta la classe 50.

8Els tres semàntics relacionats amb la voluntariat que caracteritzen aquest component no són objecte d'estudi en aquest treball i, a més, s'han mostrat irrellevants per a la caracterització de la classe. Per a més detalls, v. Fernández i Martí [4].
3 Implementació del model en la base de coneixement

Per a la construcció de la base de coneixement s'ha utilitzat el llenguatge de representació lèxica Laurel d'A. Copestake [3]. La informació s'ha estructurat prenent com a referència el model veïnic de l'HPSG (Pollard i Sag [12]) i incorporant alguns dels elements proposats per Pustejovsky [13].

3.1 Laurel (LRL)

Laurel és un llenguatge de representació del coneixement lèxic basat en la unificació d'estructures de trets tipificades. Aquest llenguatge permet diferenciar quatre nivells de coneixement: el sistema de tipus, en què s'aplica l'herència monotònica; els lèxic, on es permet l'herència per defecte; les regles lèxiques, que permeten generar noves entrades a partir de les existents; i, finalment, els t-links o enllaços multilingües, que permeten definir relacions de traducció entre els lèxic de llengües diferents.

El sistema de tipus està organitzat de manera jeràrquica a través de la relació de subsumpció, la qual cosa permet l'aplicació de procediments d'herència monotònica i així els subtipus hereten restriccions de rang: es possibilita la transferència entre un nus superior i un d'inferior de manera global. El sistema admet l'herència múltiple ortogonal, és a dir, un tipus pot estar subsumit per més d'un nus, tot i que la informació dels nusos superiors no pot ser incompatible.

Les regles lèxiques s'apliquen sobre un signe lèxic (input) i en creen un de nou (output) a partir d'unes transformacions realitzades sobre l'entrada original. Aquestes regles permeten generar noves entrades en la BCL i poden ser de tipus morfològic (generació de plurals, femení dels adjectius i flexió verbal), lèxic (obtenició de derivats) o bé semàntic (tractament de la polisèmia). Laurel possibilita també la definició de regles gramaticals que combinen signes lèxics per formar signes frasals.

3.2 Representació de l'entrada lèxica

En la representació del model d'entrada lèxica s'han distingit les estructures de tres següents: SENSE-ID, ORT, FON, MORF, SINTAX, EE, SEM, M-COMP i FORLÒG.

SENSE-ID és un identificador de cada una de les entrades amb un sentit merament referencial, on s'indica la llengua de l'entrada, el númer de l'accepció i la classe semàntica a la qual pertany (v. ap. 2). Per exemple, per al verb *trencar* seria:

| SENSE-ID: | LLENGUA: català |
| N-SENTIT: | 1 |
| CLASS-SEM: | canvi |

A ORT s'expressa la forma ortogràfica del verb i a FON, la transcripció fonètica. MORF és un tret simple on s'expressa la informació referent al model flexi de l'entrada (Márquez et al. [6]).

El tret SINTAX informa sobre la llista de complements possibles d'una entrada i les realitzacions diàstiques. La primera informació s'expressa en el tret SUBCAT. Com el nombre de complements possibles és obert s'ha definit com una llista recursiva on l'element bàsic és COMPL. A DIAT s'indiquen les diferents maneres en què una forma verbal pot realitzar les alternances diàstiques.

| SINTAX: | SUBCAT: COMPL-RESTC |
| DIAT: |

Cada COMPL es defineix segons dos trets: CAT i SEMREF. A CAT s'indica la categoria sintàctica del complement, una llista tancada (SN, SP, VFIN...). El valor de SEMREF és un índex que relaciona el complement amb un element de l'estructura argumental. L'obligatorietat dels complements es resol mitjançant l' especificació de les diàtesis que accepta l'entrada.

| SUBCAT: | COMPL: CAT=cat |
| SEMREF= |
| <SEM:ARGS:ARG1> |
| RESTC: ... |

Seguint l'exemple anterior de *trencar*, aquesta estructura quedaría de la forma següent:

| SUBCAT: | COMPL: CAT=SN |
| SEMREF= |
| [1]nat-force/animate |
| RESTC: COMPL: CAT=SN |
| SEMREF= |
| [2]PHYS-OBJ |
| RESTC=F(alse) |

DIAT consta de quatre trets que fan referència a les quatre oposicions bàsiques de caràcter semàntic al voltant de les quals s'organitzen les diferents realitzacions sintàctiques (v. ap. 1.1.2). Cadascuna
d’aquestes oposicions es defineix en termes de quatre trets. El primer, DÍAVALE, és un boolean que indica la participació no del verb en aquella oposició. Els altres tres, D-LEX, D-SINT i D-MORF, s’usen per activar o bloquejar l’aplicació de les regles lèxiques que generen les alternances de diàtesis. D-LEX s’activa quan la peça lèxica no pateix cap tipus d’alteració en la realització de la diàtesi (butar); D-SINT correspon a aquells verbs que necessiten una forma perifràstica per expressar una alternança (sonar/fer sonar); finalment, D-MORF adquireix un valor positiu quan el verb es modifica morfològicament (per exemple, mitjançant la presència del pronom es, trecrar, trecar-se):

| DÍAVALE:  | boolean |
| D-LEX:    | boolean |
| D-SINT:   | boolean |
| D-MORF:   | boolean |

Pel que fa a l’estructura esdevenimental (EE), se segueix la proposta de Pustejovsky [13] tant en el plantejament com en la representació. En primer lloc, es declaren els predicats implicats i s’indica el tipus (procés, estat, esdeveniment complex). En aquest últim cas, s’especifica la relació entre les subestructures mitjançant el tret RESTR, que té com a valors una llista predefinida. Els valors de la llista són: a, encavalçament, <, precedència, i ≤, precedència i encavalçament. Finalment, mitjançant el tret HEAD s’indica quina és la subestructura enfocada.

L’estructura corresponent als predicats simples és la següent:

| EE: e1=tip-pred |

L’estructura corresponent als esdeveniments complexes és aquesta altra:

| EE: e1=tip-pred  
| e2=tip-pred  
| RESTR=relació  
| HEAD=e1/e2 |

En el tret SEM s’especifica el nombre d’arguments del predicat i les restriccions selectives. Cada argument està coïndexat amb els constituents sintagmàtics i amb els components de significat. És en aquesta estructura de trets on s’especifica la interrelació entre sintaxi i semàntica.

| ARGSTR: |
| arg1: entity[1]  
| arg2: entity[2] |

En el cas del verb trecrar seria:

| ARGSTR:  |
| arg1: physobj[1]  
| QUALIA:FORMAL: animat/força-nat  
| arg2: top[2] |

En l’estructura MCOMLIST es declaren els components de significat que corresponen a cada predicat. Els components de significat es tracten com una llista recursiva, formada per l’estructura MCOM i per la llista dels components restants, RESTCOM. MCOM construix d’un tret, COMP, on s’especifica el valor (iniciador, entitat, etc.) i un índex que coïndexa amb l’argument corresponent.

| MCOMLIST:  |
| MCOM: COMP=valor  
| INDEX=[1] |

El tret FORLOG inclou la fórmula lògica associada a les entidades verbals. La fórmula lògica té com a objectiu relacionar els esdeveniments amb el predicat i els arguments implicats.

FORLOG té dos possibles valors en funció de la complexitat esdevenimental del predicat. En el cas que el verb sigui un predicat simple (procés o estat), la fórmula es compondrà mitjançant un predicat (PRE), un esdeveniment (EVE) i la llista dels arguments implicats (ARGLIST):

| Predicat(e,x,y) |

que en la representació proposada s’anota tal com s’indica a continuació:

| FORLOG:  |
| PRE=ORT  
| EVE=e[1]  
| ARGLIST: ARG=[2]  
| ARG=[3]  
| ARGLIST... |

En el cas del verb oferir, la fórmula quedaría de la forma següent:

| FORLOG:  |
| PRE=ofrir  
| EVE=e[1]  
| ARG=[3]objecte  
| ARG=[4]  
| ARGLIST= nul |
En el cas de les estructures esdevenimentals complexes, la fórmula lògica ha de ser prou expressiva com per poder codificar diverses fórmules i la relació que mantienen entre si. Així, el valor complex de FORLOG constaria de dues fórmules simples (del tipus que hem vist abans) i l’anotació sobre el tipus de connectiva que les relaciona. El que es pretén és associar una interpretació a cada subestructura amb l’objectiu que una regla lèxica sigui capaç de produir de forma independent la fórmula adequada.

\[
\text{(trençar}(e1,x,y))\land(\text{trençar}(e2,y))
\]

on la primera expressió codifica la relació entre l’estructura esdevenimental i els arguments en el procés i la segona en l’estat.

4 Conclusions

S’ha presentat un model teòric d’entrada que integra tot tipus d’informació lingüística pertinent per donar compte de les propietats i del comportament de les peces lèxiques: fonologia, ortografia, morfologia, sintaxi i semàntica. Aquest model ha estat desenvolupat de manera exhaustiva per expressar de manera pormenoritzada els aspectes rellevants en la caracterització de les entrades lèxiques verbals, concebudes com el producte de la integració de diversos mòduls interconnectats: els components de significat, l’estructura esdevenimental i les alternances de diàtesis.

Es considera, així mateix, que les estructures de trets enriquitdes amb mecanismes d’hèrència monotònica i per defecte constitueixen un formalisme adequat per a la representació del coneixement lèxic. En aquesta línia, el llenguatge de representació de Copestake [3] posseeix els requisits necessaris per expressar tant les generalitzacions com les particularitats lèxiques.

Aquest model d’entrada formalitzat en termes d’estructures de trets pot ser utilitzat per a la generació d’un lèxic per a qualsevol aplicació basada en el processament del llenguatge natural, així com per a l’obtenció de dades rellevants per a la confecció d’un diccionari d’ús comú.

L’objectiu global del projecte és la descripció dels verbs del català i el castellà a través d’aquest model teòric. De la combinació dels mòduls proposats s’obtindrà una classificació dels verbs. Fins al moment, s’han analitzat uns 2.000 verbs de les dues llengües esmentades i s’han definint dues grans classes semàntiques: els verbs de canvi i de trajectòria. Cada classe tindrà associat un subtipus de l’entrada prototipus i els membres de cada grup heretaran els trets que són comuns al model general i instanciaran els que són propis de l’entrada. D’aquesta manera s’economitza la representació de les dades i es capten generalitzacions rellevants des del punt de vista lingüístic, ja que cada classe constituïrà un model de comportament verbal.
Referències


Reinforcement Learning in Constructive Languages

Josep Hernandez-Orallo
Universitat Politecnica de Valencia. Departament de Sistemes Informàtics i Compatoació,
Camí de Vera 14, Aapt. 22.012 E-46071, Valencia (L'Horta)
E-mail: jorallo@dsc.ucv.es

Abstract

We present different ways of measuring reinforcement for eager learning methods and constructive languages. The problem of propagating reinforcement from the evidence into the theory has been shown especially troublesome in high-level languages, like ILP, but the same problem pervades other representations that allow redescription (e.g. neural networks).

In this work, we present an operative measure of reinforcement for general theories, studying the growth of knowledge, theory revision and abduction in this framework. Our approach performs an apportionment of credit wrt. the 'course' that the evidence makes through the learnt theory. The result is compared with other evaluation criteria, like the MDL principle.

Finally, we study a more common view of reinforcement, where the actions of an intelligent system can be rewarded or penalised, and we discuss whether this should affect the distribution of reinforcement.

The most important result of this paper is that the way we distribute reinforcement into knowledge results in a rated ontology. In this way, one of the most difficult dilemmas of inductive learning, the choice of a prior distribution, disappears.

Keywords: Reinforcement Learning, Incremental Learning, Ontology, Apportionment of Credit, Abduction, Induction, MDL principle, Knowledge Acquisition and Revision, ILP.

1 Introduction

The study of reinforcement learning in restricted representations has been especially popular in this decade (see [19] for a survey) and it has been recently related with EBL [7]. One of the main problems of reinforcement learning is that it is increasingly more difficult to assign and 'propagate' the reinforcement (or apportionment of credit [18]) depending on two factors (which are as well related): (1) how eager is the inductive strategy (vs. lazy methods like instance-based and case-based reasoning [27]) and (2) how expressive is the language where induction must take place. Explanatory Based Learning (EBL) and Inductive Logic Programming (ILP) are two areas where the propagation of reinforcement faces these issues in a more anidious way.

In this paper we shall address the problem of reinforcement with eager learning methods. Eager learning methods extract all the regularity from the data in order to work with intensional knowledge (instead of the extensional knowledge of lazy methods [1]).

Additionally, we will consider the problem with constructive languages. A constructive language is a language that allows dynamical change of its representational bias (what is sometimes known as the possibility of 'redescription'), i.e., new constructed terms can be created to express more compactly the evidence. This is usually known in ILP as predicate invention.

In decision trees or attribute languages, no invented terms are induced and the reinforcement is distributed among the initial attributes. The main drawback of these approaches is the lack of flexibility: when arrived to a 'saturation' point, the data are not abstracted further and the mean reinforcement arrives to a limit. Consequently, the ontology must be given and not constructed (a model of the 'world' is embedded in the system) and the possible extensions of this world are very restricted.

In the case of learning in highly expressive frameworks, a main problem is presented (apart from efficiency): the ontology of the new constructed concepts is indirect. The usual solution to this problem is the assumption of a prior probability. Once the probabilities are assigned, a bayesian framework can be used to 'propagate' the distribution.

In general, there is no justification at all of which prior distribution to choose. In the absence of any knowledge, the most usual one is the MDL (Minimum Description Length) principle [34][35]. The MDL principle is just a formalisation of Occam's razor. Theoretically, its close relation with PAC-learning [41] has been established by [4]. Some high-level representation inductive methods have adapted these ideas (e.g. U-learnability in ILP [32]).

All of them are based on the assumption of a prior. However, there are many riddles with the management of probabilities and, in particular, the best choice, the MDL principle, has additional ones.

As we will see, most of these difficulties would disappear if no prior distribution is assumed and the
knowledge is constructed by reinforcement, as the data suggest. However, the translation of these ideas to general representational frameworks seems difficult. First, the length of the structures which supposedly are to be reinforced is variable. Second, and more importantly, it seems we can always invent ‘fantastic’ concepts that can be used in the rest of knowledge. Consequently, these ‘fantastic’ concepts are highly reinforced, increasing the reinforcement ratio of knowledge in an unfair way.

An immediate way out is the combination of reinforcement learning with some prior, mainly the MDL principle, essayed under the name of ‘incremental self-improvement’ [36] using syntactic minimality to restrict the appearance of these inventions.

Notwithstanding, our approach also avoids ‘fantastic’ concepts but it is based exclusively on reinforcement. Consequently, compression turns out to be an ‘a posteriori’ consequence of a well-established reinforcement, instead of an ‘arbitrary’ assumption.

The paper is organised as follows. Section 2 presents some prior distributions usually assumed in machine learning, especially the MDL principle. Section 3 introduces our framework for incremental knowledge construction. Section 4 essays a first adaptation of reinforcement to realise the problems of ‘fantastic’ concepts. Section 5 remarks the approach and introduces the idea of ‘course’ to measure reinforcement. Section 6 discusses the extension of these ideas to wider notions of reinforcement with the presence of reward and penalties. Section 7 considers the length of the reinforced ‘units’ or ‘rules’ showing the relation with the MDL principle in the limit. In the same section it is introduced a balanced reinforcement suitable for EBL. Section 8 presents two methods for computing effectively these measures and deals with their limitations and complexity. Section 9 closes the paper discussing the results and the open questions.

2 Prior Selection in Machine Learning

The aim of Machine Learning is the computational construction of hypothetical inferences from facts, as Michalski have pointed out [28]: “inductive inference was defined as a process of generating descriptions that imply original facts in the context of background knowledge. Such a general definition includes inductive generalisation and abduction as special cases”.

However, given some evidence $E$, infinite many hypotheses $H$ can be induced ensuring $H \models E$. Obviously, some selection criteria are needed. Depending on different applications, some criteria have been used (e.g. the most specific hypothesis, the most general one, the shortest one, the most informative one, ...). In general, this choice implies the assumption of a prior distribution which can be used to derive the likeliness of the hypotheses.

The principle of simplicity, represented by Occam’s razor, selects the shortest hypothesis as the most plausible one. This principle was rejected by Karl Popper because, in his opinion (and at that moment) there was no objective criterion for simplicity. However, Kolmogorov complexity [43], denoted $K(x)$, is an objective criterion for simplicity. This is precisely what R.J. Solomonoff proposed as a ‘perfect’ theory of induction [26]. Algorithmic Complexity inspired J. Rissanen in 1978 to use it as a general modelling method, giving the popular MDL principle [34], recently revised as a one-part code [35] instead of the initial two-parts code formulation.

It is remarkable (and often forgotten) that Kolmogorov Complexity just gives consistency to this theory of induction; Occam’s razor is assumed but not proven. Nonetheless, some justifications have been given in the context of physics, reliability and entropy, but, in our opinion, it is the notion of reinforcement (or cross validation) which justifies the MDL principle more naturally. At a first sight, it seems that the higher the mean compression ratio $(\text{length}(E) / \text{length}(H))$ the higher the mean reinforcement ratio.

Summing up, the MDL principle says that, in absence of any other knowledge about the hypotheses distribution, we should select the prior $P(h) \propto 2^{-K(h)}$, prevailing short theories over large ones. However, this prior has many riddles. First of all, (1) it is not computable, so the prior must be approximated (e.g. using the time-weighted variant $K_T$ of Kolmogorov complexity [24]) or must dynamically change as the learner knows that something can be further compressed. Second, (2) it presents problems with perfect data; the MDL principle usually ‘underfits’ the data, because sometimes it is too conservative. Third, the reliability of the theory is not always increasing with the number of examples which have confirmed the theory (e.g., a string of $10^{10}$ is more compressible than a string of $78450607356$’s). Moreover, (4) is difficult to work with different and non-exclusive hypotheses, because if we have $T_a$ and $T_b$ intuition (and logic) says that $T = T_a \lor T_b$ should have more probability, but MDL assigns less probability to $T$ because it is larger.

Finally, (5) the MDL principle has shown problems for explanation, because, for the sake of maximum mean compression, some part of the hypothesis can be not compressed at all, resulting in a very compressed part plus some additional extensional cases. This extensional part is not validated, making the whole theory weak. An ontology is difficult to construct from here if they are unrelated (not explained) with the other facts. This is closely related with the differentiation between Enumerative Induction and Best Explications.
nation [13] [14] [8] and the distinction between Induction
and Abduction [10].

We intend to handle these difficulties with a dyna-
mical reinforcement. However, our approach has
additional advantages: (1) no prior assumption has to
be made (apart from how to distribute this reinforce-
ment, which is the topic of this paper), i.e. knowledge
is constructed just as the data suggest, and (2) rein-
forcement can be more flexibly managed than prob-
abilities, and allows further insight on the relation
between the evidence and the theory.

3 Preliminaries

With this section we just present the schema of incremen-
tal learning and the languages we aim to address
in the following sections.

3.1 Incremental Knowledge Construction

The field of knowledge construction gathers many
other related subfields and usually makes use of very
complex techniques for the organisation and revision
of the data. We will tackle exclusively the inductive or
learning task in knowledge construction.

Incremental knowledge construction (which in-
cludes acquisition and revision) generates a theory
from an evidence that is gradually supplied example
by example. From the very beginning, with an empty
knowledge $T=\emptyset$, when new observations or evidences
$e$ are received, we can have three possible situations:

- **Prediction Hit** (or ‘matter of course’). The observa-
tions are covered without more assumptions, i.e.,
$T \models e$. The theory $T$ is reinforced.

- **Novelty**. The observation is uncovered but consist-
tent with $T$, i.e. $T \models e$ and $T \cup e \not\models \Box$. Here, the
possible actions are:
  1. **Extension**: $T$ can be extended with a good ex-
    planation $A$, (i.e. $T \cup A \models e$).
  2. **Revision**: revised if a good explanation cannot
     be found.
  3. **Patch**: quoted as an extensional exception
     (i.e. $T'=T \cup e$), or
  4. **Rejection**: regarded as noise and ignored.

- **Anomaly**. The observation is inconsistent with the
theory $T$, i.e., $T \not\models e$ and $T \cup e \not\models \Box$. In this case, $T$
cannot be extended and there are three possibilities:
  1. **Revision** or **Patch**.

An eager but still non-explanatory approach to theory
formation is Kuhn’s theory of changing paradigms [23]
which basically matches with the MDL principle: as
too many exceptions to the paradigm are found, they
are increasingly lengthy to quote (patch) and the
whole paradigm (or part of it) must be changed.

In the preceding sketch, abduction appears as an
extension of current knowledge with some assumption
(usually one or more facts) and induction is also an
extension or revision which performs some kind of
generalisation. Nevertheless, this characterisation is
not sufficient for a clear distinction (see [15] for
sounder considerations about how to distinguish
them). In fact, it is a topic of current discussion (for a
state of the art see [10]). In this way, abduction has
been commonly seen as belief revision [5], usually
combined with induction [2]. In other cases is related
with validation, justification or ontology [9] in the way
the part of the theory where abduction supports must
be reliable. Unavoidably, this reliability must come
from a reinforcement produced by the previous evi-
dence.

The previous schema is general enough to include
explanatory and conservative knowledge construction.
Explanatory knowledge construction should minimise
the exceptions, so patches and revisions should not be
allowed. Thus, the revisions are much more frequent.
Even more, the goal is anticipating, investing, finding
more informative and easily refutable hypotheses [33],
in contrast to what many approaches to minimal revi-
sions aim for (see e.g. [30]), supported by the obvious
fact that a minimal revision is usually less costly, in
short-term, than a deep revision.

3.2 Representation Languages

For the study of reinforcement we need to introduce
some basics for the representation to which it can be
applied. A ‘pattern’ of languages is defined as a set of
chunks or rules $r$ which are composed of a head (or
consequence) and a body (or set of conditions) in the
following way $r = \{ h \leftarrow t_1, t_2, \ldots, t_n \}$.

Since no restriction of how $h$ and $t$ can be (there
may be variables, equations, boolean operators...), our
definition could be specialised to propositional lan-
guages, Horn theories, full logical theories, functional
languages, some kind of grammars, and even higher-
order languages. In the following, we leave unspeci-
fied the semantics of the representations and we just
say that $e$ is a consequence of $P$, denoted $P \models e$ (in
other words, there is a proof for $e$ in $P$, or, simply, $P$
covers $e$).

4 Reinforcement wrt. the Theory Use

Whatever the approach to knowledge construction,
the revision of knowledge must come from a partial or
total weakness of the theory or, in other words, a loss
of reinforcement (or apportionment of credit [18]). We
present a way to compute the reinforcement map for a
given theory, depending on past observations.
DEFINITION 4.1
The pure reinforcement $r(r)$ of a rule $r$ from a
type $T$ w.r.t. some given evidence $E = \{ e_1, e_2, \ldots, e_i \}$ is computed as the number of proofs of $e_i$ where
$r$ is used. If there are more than one proof for a
given $e_i$, all of them are reckoned, but in the same
proof, a rule is computed only once.

DEFINITION 4.2
The (normalized) reinforcement $r(r)$ is $1 - 2^{-\text{mp}(r)}$.
Definition 4.2 is motivated by the convenience of
maintaining reinforcement between 0 and 1. The mean
reinforced ratio $\mu r(T)$ is defined as $\sum_{e \in T} r(r)/m$, being
$m$ the number of rules. These definitions show that, in
general, the most (mean) reinforced theory is not the
shortest one as the following example shows:

EXAMPLE 4.1
Given the evidence $e_1, e_2, e_3$, consider a theory $T_0 = \{ r_1, r_2, r_3 \}$ where $r_1$ covers $e_1$, $r_2$ covers $e_2$ and $r_3$ covers $e_3$. Then, a theory $T_0 = \{ r_1, r_2, r_3, r_4 \}$ where $r_1$, $r_2$ cover $e_1$, $r_2, r_3$ cover $e_2$, and $r_3, r_4$ cover $e_3$.

From here, $T_0$ is less reinforced than $T_0$.
In the first case we have $\mu r_0 = 2^2 = \mu r_0 = 1$ and
$\mu p(T_0) = 0.5$. For $T_0$ we have $\mu r_0 = 2^3 = \mu r_3 = 1$.
$\mu p(T_0) = 0.5938$.

In addition, redundancy does not imply a loss of mean
reinforcement ratio (e.g., just add twice the same rule).

However, measuring reinforcement of the theory
presents problems of fantastic (unreal) concepts:

THEOREM 4.3
Consider a program $P$ composed of rules $r_i$ of the
form $\{ h : t_1, t_2, \ldots, t_k \}$, which covers $n$ examples $E
= \{ e_1, e_2, \ldots, e_n \}$, if the mean reinforcement ratio
$\mu p < 1 - 2^{-n}$ then it can always be increased.

PROOF
A fantastic rule $r_i$ can be added to the program by
modifying all the rules of the program in the fol-
lowing way $r_i' = \{ h : t_1, t_2, \ldots, t_k, r_i \}$. Obviously, all
the other rules maintain the same reinforcement
but $r_i$ is now reinforced with $\mu p' = n$. Since $\mu p > \mu p'$
then the new $\mu p'$ must be greater than $\mu p$.

One can argue that these fantastic rules could be
checked out and eliminated. However, there are
many ways to ‘hide’ a fantastic rule; in fact, cryptography
relies on this fact.

5 Reinforcement w.r.t. the Evidence
It can be derived from this problem that reinforcement
must be combined with a simplicity criterion in order to
work (maybe neural networks theory is the field
where this avoidance of overfitting, ensured by sim-
licity, has been more thoroughly studied in combi-
nation with reinforcement).

However, there is solution without explicitly mak-
ing use of simplicity. The idea is measuring the valid-
ation w.r.t. the evidence.

DEFINITION 5.1
The course $\chi(e)$ of a given fact $e$ w.r.t. to a theory
is computed as the product of all the reinforce-
ments $r(r)$ of all the rules $r$ used in the proof of $e$. If
a rule is used more than once, it is computed once.

If $f$ has more than one proof, we select the greatest
course.

In this case, we can select the theory with the greatest
mean of the courses of all the data presented so far, as
defined as $m(T, e) = \sum_{e \in T} \chi(e)$, being $n$ the number
of facts (examples) in the evidence. We can use the
geometric mean instead, denoted by $\mu \chi$. The following
example shows the use of this new criterion for knowl-
edge construction:

EXAMPLE 5.1
Using Horn theories as representation (Prolog), suppose
we have an incremental learning session as follows:

$T_0 = \{ r(X,Y,Z) : s(X, Y) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_2 = \{ r(a,b,c) : p = 0.75 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_3 = \{ r(X,Z,Y) : s(X,Y) : p = 0.75 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_4 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_5 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_6 = \{ r(X,Y,Z) : s(X,Y) : p = 0.75 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_7 = \{ r(X,Y,Z) : s(X,Y) : p = 0.75 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_8 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_9 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.

At this moment, $P_1$, $P_2$, and $P_3$ are the best options and $P_4$ and $P_5$ seem risky theories, according to the evidence.

$e(a,b,c,d)$ is observed.

$P_1$ does not cover $e(a,b,c,d)$ and it is patched:
$P_1 = \{ r(X,Y,Z) : s(X,Y) : p = 0.875 \}$
$\mu(p, a,b,c) = 0.75$.
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$\mu(p, a,b,c) = 0.75$.
$P_4 = \{ r(X,Y,Z) : s(X,Y) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_5 = \{ r(X,Y,Z) : s(X,Y) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_7 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_9 = \{ r(X,Y,Z) : t(X,Y,Z) : p = 0.875 \}$
$\chi(e) = \chi(e) = \chi(e) = 0.75$.
$P_1$ is reinforced $\{ r(X,Y,Z) : p = 0.75 \}$. 267
The example illustrates that in general, and using this new reckoning of reinforcement, the shortest theories are not the best ones. More importantly, it also shows that as soon as a theory gains some solidity, abduction can be applied.

The way reinforcements are calculated makes that very complex programs are avoided, but redundancy is possible. But now there is not any risk of fantastic concepts. As said before, for any program \( P \) composed of rules \( r_i \) of the form \{ \( h \vdash t_1, t_2, \ldots, t_n \) \}, which covers \( m \) examples \( E = \{ e_1, e_2, \ldots, e_m \} \) and their reinforcements \( r_i, \) a fantastic rule \( r_i \) could be added to the program and all the rules could be modified in the following way \( r_i = \{ h \vdash t_1, t_2, \ldots, t_n, r_i \} \). The following theorem shows that now it is not reinforced over the original one:

**Theorem 5.1**

The course of any example cannot be increased by the use of fantastic concepts.

**Proof**

Since the fantastic concept \( r_i \) now appears in all the proofs of the \( m \) examples, the reinforcement of \( r_i \) is exactly \( 1 \leq 2^m \) and the reinforcements of all the \( r_i \) remain the same. Hence, the course of all the \( m \) examples is modified to \( \chi(e_i) = \chi(e_i) \cdot r_i = \chi(e_i) - \chi(e_i) \cdot 2^m \). Since \( m \) is finite, for all \( e_i \in E, \chi(e_i) \) can never be greater than \( \chi(e_i) \).

These ideas are being used by [17] in an incremental learning system using Curry as a representation language (a logic functional programming language based on narrowing with some higher-order constructs). The results demonstrate that the intended hypothesis is found sooner than when using the MDL principle.

Another advantage of this approach is that a "rated" ontology can be derived directly from the theory. In this way, the parts which are sound or weak are easy to detect. Intuitively, if a rule only covers just one example, it suggests that the rule is not very real.

### 6 Rewarded Reinforcement

In reinforcement learning, it is usually assumed that the learner receives some reward (or penalty) value of its actions. In other words, prediction hits can receive different degrees of reward and prediction errors (including novelties and anomalies) can receive different degrees of penalty (or negative reward).

Usually, this broader view of reinforcement is suitable for frameworks where reasoning about action is necessary. The rewards are assigned depending on the actions that the agent performs for each situation. Temporal languages are used for representation, like event calculus or situation calculus [22]. The important issue here is that our model can be used in these cases, by asking the learning system to predict the following situation \( s_{n+1} \), after every possible action it
can perform in a certain situation \( s \). The task of the system is just selecting the one with the greatest reward. In the case the result of the action matches with the evidence, a positive hit happens with the predicted reward. In the case a prediction error occurs, the action may have no awful consequences (no penalty) or it may be fatal. The question is how ontology and 'hedonism' must be combined. It is commonly accepted in psychology the claim that hedonism motivates ontology, and this is stronger the earlier the stage of development of a cognitive system. In our opinion, this motivation does not imply that they must be mixed. Moreover, rewards should be learned as well because they may change.

From here, the choice of the best action must take into account both the reliability of the prediction (i.e. the reinforcement) weighted with the reward, not the action with the best reward alone (because it may be a very weak guess).

Finally, there can be degrees of reliability in the evidence. This degree may come from different reliabilities of the sensors of the system or from intermediate recognition or sensor preprocessing subsystems. Indeed, this should affect ontology in the following way: every fact of the evidence is assigned a real number as a reliability degree, \(-1 \leq \alpha \leq 1\). In this framework, the completely reliable positive examples are assigned a value of \(\alpha = 1\) and the completely reliable negative examples are assigned a value of \(\alpha = -1\).

**DEFINITION 6.1**

The 'grounded' course \( \chi'(f) \) of a given fact \( f \) w.r.t. to a theory is computed as the normal course \( \chi(f) \) multiplied by the reliability degree of \( f \). More formally, \( \chi'(f) = \chi(f) \cdot \alpha \).

### 7 Balanced Reinforcement

With the approaches introduced in section 5 and section 6 there is a tricky way of increasing reinforcement: joining rules. If a high-level representation mechanism allows very expressive rules, larger rules can be made in order to stand for the same that was expressed with separated rules, with the advantage of increasing reinforcement:

**EXAMPLE 7.1**

For instance, the following extended functional programs are equivalent:

\[
T_0 = \{ \begin{array}{l}
    r_1 = (f(X,a) \rightarrow g(b)) \\
    r_2 = (f(X,c) \rightarrow i(d))
\end{array}
\]

\[
T_0 = \{ r = (f(X,Y) \rightarrow \begin{array}{l}
    \text{if} (Y=a) \text{ then } g(b) \\
    \text{if} (Y=c) \text{ else } i(d)
\end{array})
\}
\]

but \( T_0 \) would be more reinforced than \( T_0 \).

The solution to this problem requires the introduction of a factor inversely related with the syntactical length of a rule. It is important to clarify that this syntactical measure is not a prior and it can be effectively computed, in contrast to the MDL principle.

With length \( r \) we denote the length of a rule \( r \) for the concrete language which would be used. The only restriction for length \( r \) is that for all \( r \), length \( r \) \( \approx 1 \). Thus we extend the definitions of section 5:

**DEFINITION 7.1**

The extended pure reinforcement is defined as:

\[
\rho^p(r) = \rho(r) / \text{length}(r).
\]

The extended normalised reinforcement \( \rho^*(r) \) and the extended courses \( \chi^*(r) \) are defined in the obvious way using \( \rho^p(r) \) and \( \rho(r) \).

It is obvious that if length \( r \) simply assigns 1 to every rule of the program, these definitions are equivalent to those of section 3.

With this extension, it is easy to show that —in the limit \( k \rightarrow \infty \)— the MDL principle is an excellent principle for achieving reinforcement:

**THEOREM 7.2**

If the data \( E \) are infinite and a theory \( T \) is finite, the mean course \( m\chi^*(T,E) = 1 \).

**PROOF**

Given some infinite data as evidence \( E = \{ \, e_1, ..., e_n \} \), without loss of generality, consider that \( T \) can be exclusively composed of two rules: \( r_1 \), which covers all \( E \) except \( e_i \) and, independently, \( r_2 \), which covers \( e_i \). The reinforcements are \( \rho(r_1) = (1-2^{-\alpha t_{\chi}t_{\chi}(r_1)}) \) and \( \rho(r_2) = (1-2^{-\alpha t_{\chi}t_{\chi}(r_2)}) \) and the mean course \( m\chi^*(T,E) = \frac{(n-1)!}{(n-1)!} \cdot (1-2^{-\alpha t_{\chi}t_{\chi}(r_1)}) + (1-2^{-\alpha t_{\chi}t_{\chi}(r_2)}) / n \). For infinite data, we have that \( \lim_{n \rightarrow \infty} m\chi^*(T,E) = 1 \).

This theorem shows that maximum reinforcement matches with maximum compression in the limit (simply because both are saturated). However, when the data are finite we have many cases where they differ. The most blatant case occurs when some exception is covered extensionally (as \( r_2 \) which covers \( d_1 \) in the proof of theorem 7.2) and there is an important loss of reinforcement vs. a slight loss of compression. The following example illustrates this point:

**EXAMPLE 7.2**

Consider the following evidence \( e_1, ..., e_n \):

\[
E = \{ \begin{array}{l}
    e_1: e(4) \rightarrow \text{true}, \quad e_2: e(12) \rightarrow \text{true}, \\
    e_3: e(3) \rightarrow \text{false}, \quad e_4: e(2) \rightarrow \text{true}, \\
    e_5: e(7) \rightarrow \text{false}, \quad e_6: e(7) \rightarrow \text{false}, \\
    e_7: e(20) \rightarrow \text{true}, \quad e_8: e(0) \rightarrow \text{true}, \\
    e_9: e(3) \rightarrow \text{true}, \quad e_{10}: e(2) \rightarrow \text{false}
\end{array}
\]

and that natural numbers are represented using the function \( s \) as the symbol for successor, e.g. \( s(0) = 1 \). The length \( n \) of a rule is computed as \( 1 + n \cdot n_{\alpha} \), where \( n_{\alpha} \) means the number of functions (including constants as functions with arity 0) and \( n \) the number of variables.

From here, the following theories are evaluated:

\[
\begin{array}{l}
    \vdash l \\
    \vdash \rho^p \rho^p \rho^* \rho^*
\end{array}
\]
The extended course are $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6) = 0.5 \cdot 0.5647 = 0.28235$, $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6) = 0.5 \cdot 0.3402 = 0.1701$, $m\chi^*(e_9) = 0.0943$ and $m\chi^*(e_{10}) = 0.1091$.

The extended course $m\chi^*$ is 0.2125.

The extended course are $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6, e_7) = 0.5 \cdot 0.5647 = 0.28235$, $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6, e_7) = 0.5 \cdot 0.3402 = 0.1701$, $m\chi^*(e_9) = 0.1797 \cdot 0.1294 = 0.02325$ and $m\chi^*(e_{10}) = 0.1797 \cdot 0.1591 = 0.02859$.

The extended course $m\chi^*$ is 0.1974.

The extended course are $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8) = 0.5898$.

The extended course are $m\chi^*(e_1, e_2, e_3, e_4, e_5, e_6, e_7) = 0.5898 \cdot 0.4257 = 0.2511$, $m\chi^*(e_9) = 0.2063 \cdot 0.5898 \cdot 0.4257 = 0.1591 = 0.00824$ and $m\chi^*(e_{10}) = 0.2063 \cdot 0.5898 \cdot 0.6464 \cdot 0.1591 = 0.0125$.

The extended course $m\chi^*$ is 0.2681.

Note that the lengths ($l(T_2)=29$, $l(T_3)=32$, $l(T_4)=30$) would not give many hints about which theory to select. The example also shows the advantages of this approach for explanation-based learning. Since all the data must be explained, if a part is left in an extensional way (or unrelated with the rest), it is penalised. On the other hand, we have seen in the preceding sections that fantastic concepts are also avoided, so it results to be a balanced criterion for the 'intensionality' of theories, without falling into fantasy.

Regarding $T_2$ of example 7.2, our measure can be adapted to situations where a more compensated theory is required, using a geometric mean instead of an arithmetic mean. In addition, and concerning $T_3$, if we do not want exceptions (extensional parts) at all, we can discard theories where a fact has a course value less than the mean divided by a constant. Moreover, this case should trigger theory revision in an incremental framework in order to integrate (or reconcile) the example with the theory.

Finally, another straightforward extension to our approach is considering the length of the examples, too. However, it could also be included in the reliability value which was discussed in section 6.

8 Computing Reinforcement

First of all, it should be stated clear that our theory of reinforcement is not an inductive learning method. We have not dealt about how the theory could be constructed from the evidence. On the contrary, this paper presents a set of measures that allow a detailed study of the relation between the theory and the evidence, assisting the evaluation, the selection, and the revision of theories.

A general method of computing reinforcement is as it has been used in the examples:

**GENERAL METHOD:**
Consider the theory $T$, with $m$ rules $r_1, r_m$, and the evidence $E$, with $n$ examples $e_1, e_n$ such that $T\models E$. First we must prove all the examples and compute $\rho^p$ and $\rho^e$ for each rule. In a second stage, we prove again the $n$ examples, computing $\chi'$ from the $\rho'$ obtained in the first stage.

The complexity of the previous method seems to be, in the worst case, in $\mathcal{O}(m \times n)$. However it is not, because we have not stated any restriction about the computational cost of the theory, and each proof has its cost.

However, it would be more realistic to consider the computing of reinforcement in an incremental setting:
INCREMENTAL METHOD:
We will use four arrays: \( l_{m} \), \( \rho^{*}_{m} \), \( \rho^{t}_{m} \), and \( \chi_{l_{m}} \) for the lengths, the pure and normalised reinforce-
ments and the courses, respectively. An additional boolean bidimensional array \( U_{m\times m} \) assigns true to \( U_{i,j} \) if \( e_{i} \) uses \( r_{j} \) in its proof and false otherwise.

For each new example \( e_{m+1} \) which is received we have different possibilities:
1. If it is a hit, we remake \( \rho^{*}_{m}, \rho^{t}_{m}, \chi_{l_{m}} \), according to the proof of \( e_{m+1} \). \( U \) is extended with \( U_{m\times m} \) and \( \chi_{l_{m}} \), is updated using \( U \).
2. If it is a novelty and no revision is made to \( T \), only an extension \( T = T \cup \{ \rho^{*}_{1, \ldots, m}, \rho^{t}_{1, \ldots, m}, \chi_{l_{m}} \} \), the steps are very similar to the previous case, except that the arrays must be extended to \( m+k \).
3. Finally, if it is a novelty or an anomaly and the theory is revised in some rules \( \{ r_{1}, \ldots, r_{p} \} \) and extended in others \( \{ r_{m+1}, \ldots, r_{m+k} \} \), only the \( U_{i,j} \) which does not use any rule from \( \{ r_{1}, \ldots, r_{p} \} \) can be preserved. The rest must be remade.

The previous method ignores two exceptional cases: that a hit could trigger a revision of the theory to re-
adjust reinforcements and that case 2. could produce alternative proofs for previous examples.

Further optimisation could come from a deeper study of the static dependencies (i.e. some rule always
depends on others) and the topology of dependencies that the theory generates. On the other hand, an appro-
piate approximation could be used. Even more, some of the past evidence can be ‘forgotten’ if it is covered
by very reinforced rules, so minimising the cost.

However, in the case that an inductive learning method uses reinforcement for evaluating the theories
it is constructing, the complexity of these methods would surely be very modest compared to the usual
huge costs of machine learning algorithms.

Moreover, reinforcement measures are a very ade-
quate tool to guide a learning algorithm. For instance, in [17], the examples and rules with low reinforcement
were mixed in order to ‘conciliate’ them and to obtain
more compact and reinforced theories.

9 CONCLUSIONS
We have presented a framework to distribute or propa-
gate reinforcement into a theory depending on the
observation (or evidence). The advantage of this
approach is that it makes no assumptions about the
prior distribution. Also in this framework, knowledge
can have alternative descriptions, without reducing the
evidence’s courses. Moreover, “deduction in the
knowledge” can affect positively to reinforcement,
something that the MDL principle or other syntactic
priors avoid because the theory cannot change its syn-
tax without changing its a posteriori probability.

Reinforcement allows a more detailed treatment of
exceptions and provides different ratings for different
parts of a theory, not the single probability value given
by the priors which is assigned to the whole theory.
Moreover, different predictions or assumptions are
provided with different reliability values.

We have seen it working in the context of knowl-
edge construction, showing that abduction is feasible
as long as the theory gets reinforced. We think that the
role of reinforcement in induction and abduction in
knowledge acquisition is portable even from expert
systems and diagnostic systems to neural networks
(training=induction, recognition=abduction). It is more
obvious the relation of this work with the distribution
of reinforcement in neural networks, and the problems
of overfitting and underfitting in the learning of linear
functions. It even resembles some popular algorithms,
like back-propagation. However, a symbolical frame-
work seems an extremely adequate tool to advance and
combine different areas and applications: ILP, EBL,
Analogical Reasoning, Reinforcement Learning and
some kinds of non-monotonic reasoning.

As future work, the measures could be extended to
consider time-complexity and/or negative cases in the
courses. In addition, a deeper study of how deduction
affects reinforcement could be of capital interest in
knowledge-based systems which use inductive and
deductive reasoning techniques. Finally, we plan to
apply our ideas in domains with actions, probably us-
sing situation or event calculus [22][37], and treating
rewards in a more direct way (connecting with the
work of [7]), in order to re-associate our notion of re-
forcement with more classical notions of reinforce-
ment learning.

Acknowledgements
Voldria agra als revisors de la CCIA’98 per llurs
comentaris, especialment per seguir el millorament de la
secció 7 i la introducció de la secció 8.

References

tive Reasoning in Philosophy and AI” in M. Denecker, L. De

Raedt, P. Flach and T. Kakas (eds) ECAI’96 Workshop on
Abductive and Inductive Reasoning, pp. 7-9. 1996.
[5] Beetier, C.; Becher, V. “Abduction as belief revision” Ar-

“The computational complexity of abduction” Artificial Intel-

[7] Dietrich, T.G.; Flann, N.S. “Explanation-Based Learning
and Reinforcement Learning: A Unified View” Machine


Efficient and Comprehensible Hierarchical Clusterings

Luis Talavera  
Departament de Llenguatges i Sistemes Informàtics  
Universitat Politècnica de Catalunya  
Campus Nord, Mòdul C6, Jordi Girona 1-3  
08034 Barcelona, Catalonia, Spain  
talavera@lsi.upc.es

Javier Béjar  
bejar@lsi.upc.es

Abstract

Clustering is an important data mining task which helps in finding useful patterns to summarize the data. In the KDD context, data mining is often used for description purposes rather than for prediction. However, it turns out difficult to find system that help to ease the interpretation task to the user in both, statistical and Machine Learning fields. In this paper we present ISAAC, a hierarchical clustering system which employs traditional clustering ideas combined with a feature selection mechanism and heuristics in order to provide comprehensible results. At the same time, it allows to efficiently deal with large datasets by means of a preprocessing step. Results suggest that these aims are achieved and encourage further research.

Keywords: Clustering, comprehensibility, feature selection, data preprocessing.

1 Introduction

Clustering is one of the primary data mining tasks aiming to the goal of finding a useful set of categories or clusters to summarize the data. As in other inductive tasks, clustering results may serve for two different purposes, namely, prediction and description. As pointed out in [2], in the context of Knowledge Discovery on Databases (KDD), description tends to be a more important task, since the main focus in this discipline is on finding interpretable patterns. Traditionally, when applying clustering algorithms, the interpretation step is usually left to users. So, they have to evaluate the results and change the appropriate settings of the clustering system if the results does not suit their needs. However, the settings of a clustering system may be difficult to interpret for an average user who has not deep knowledge about metrics or control strategies.

Therefore, it may be desirable to have clustering methods which not only perform a partition of the data, but also facilitate the interpretation task. The weaknesses of traditional statistical clustering methods to cope with this requirement, gave raise to the development of conceptual clustering methods in the Machine Learning (ML) community [3, 4, 5, 7]. These methods are intended to combine the clustering and interpretation tasks thus making easy the latter to the external user.

Furthermore, data mining poses additional problems for the inductive tasks such as database size, high dimensionality or the need for user interaction, which make the process even harder, since usually, conceptual clustering systems are not specifically designed to be efficient under these circumstances.

In this paper we present the ISAAC system, an approach that combines traditional clustering concepts with some heuristic procedures in order to provide comprehensible results and, at the same time, efficiently deal with large amounts of data. It also allows the user to decide the structure of the cluster hierarchy with regard to the number of levels and their generality.

2 ISAAC

ISAAC is a conceptual clustering system that accepts vectors of nominal attribute-value pairs and summarizes these objects in probabilistic concept hierarchies. A probabilistic concept is represented
by a summary description that lists attribute values and its associate probabilities. For each concept $C_k$, we have a prototype which stores the probability $P(A_j = v_{ij} \mid C_k)$. This is the conditional probability that a value $v_{ij}$ for a feature $A_j$ will occur in an object of the cluster $C_K$. Probabilistic descriptions are not traditionally used in statistical clustering approaches, but they are more common in conceptual clustering systems [3, 4]. Probabilistic representations allow gradual updating of clusters descriptions and should be more robust than logic-based representations in the face of noise or graded concepts. In addition, probabilistic representations allow a unified view of objects and clusters. An object can be seen as a particular case of a probabilistic concept where all the probabilities are either 1.0 or 0. This is sometimes referred to as the ‘single representation trick’.

ISAAC is intended to allow users to model the construction of the cluster hierarchy which better suits their needs. Typically, hierarchical clusterings are arranged in a binary tree or dendrogram. From this tree, the user has to extract a useful partition, or apply some automatic procedure to select the best level or levels. In our approach, the system allows the user to specify the number of levels of the hierarchy and their generality. This is done via the $NG$ parameter which is in the $[0,1]$ range. As the $NG$ value increases, the system creates more general partitions with few clusters. Lower $NG$ values instruct the system to build more specific partitions. A complete hierarchy is built by specifying a set of increasing $NG$ values to indicate the desired levels. The user can interact with the system experimenting with different sets of values for this parameter. Since the effect of modifying the $NG$ values is semantically clear to the user, it should be easier to deal with this parameter than, for instance, to specify distance thresholds to decide cut points in a tree.

The ISAAC clustering process, which is depicted in Fig. 1, consists of three stages: Preprocessing, Reflection and Refinement which will be detailed in the following sections.

2.1 Preprocessing

Typically, complexity of hierarchical clustering algorithms is $O(n^2)$ where $n$ is the number of objects in the dataset. This complexity may be acceptable if one desires just a ‘one-shot’ clustering. However, for large datasets, it may result in a relatively slow processing, particularly if the user needs to interact with the clustering system by changing some setting in order to obtain interesting results.

The Preprocessing stage is intended to cope with this problem by exploiting incremental clustering algorithms, which can efficiently deal with large datasets. An incremental clustering algorithm can construct a flat partition in a $O(nk)$ time, where $k$ is the number of clusters created. Note that this complexity is equivalent to $O(n^2)$ only if $k \approx n$. The problem with incremental algorithms is that they are sensitive to ordering effects. To cope with this problem while maintaining the efficiency of incremental processing, we developed a buffering strategy which has shown to be fairly robust under bad object orderings and it is detailed in [10].

We chose a nearest neighbor algorithm with uses a similarity measure and an $\alpha$ threshold to form clusters. For each instance the similarity with every existing cluster is computed. If the maximum similarity found is greater than $\alpha$ then the instance is incorporated into the cluster. Otherwise, a new cluster is created. With this preliminary step, the size of the database can be compressed in a variable amount dependent on the value of $\alpha$.

By defining the similarity measure over probability distributions, we can take advantage of the unified representation of objects and clusters and use the same measure to compare objects, objects and clusters and clusters. Intuitively, we can consider two probabilistic descriptions as similar if their probability distributions are also similar. Viewing probability distributions as histograms, a natural way for representing the degree of coincidence between the distributions is to take the intersection of the histograms. Specifically, we can obtain a numerical measure of this coincidence by computing the area of this intersection. More formally, the
similarity between two clusters \( C_m \) and \( C_n \) for a feature \( i \) is computed with the expression

\[
Sim(C_m, C_n, i) = \sum_j \min\{P(A_i = V_{ij} \mid C_m), P(A_i = V_{ij} \mid C_n)\}
\]

where \( j \) indexes the different values of feature \( i \). The total similarity between two clusters is then computed averaging the similarity computed for each feature and normalizing by dividing for the total number of features considered:

\[
Sim(C_m, C_n) = \frac{\sum_i Sim(C_m, C_n, i)}{\#i}
\]

This measure is into the \([0,1]\) interval and, therefore, this is also the allowed range of values for the \( \alpha \) parameter. Moreover, when applied to individual objects, this measure gives 1 if the value for a given feature is the same in both objects, and 0 otherwise, thus reducing to the inverse of the Hamming distance.

This procedure helps to detect very similar or even identical objects in a fast manner. It is worth to notice that although very low \( \alpha \) values will result in a greater compression of the dataset, it also will produce very general clusters, letting little room for improvements to the next stages. The user may explore a suitable trade-off between these two aspects by using different \( \alpha \) values.

We can see this stage as similar to the typically used feature selection steps which reduce the set of features used in the learning process. The difference here lies in that, instead of selecting objects, the preprocessing step transforms the initial data vectors into new vectors summarizing compact groups of objects.

2.2 Reflection

This stage is intended to extract useful information about the clustering process in order to guide the following steps. It actuates at every step of the agglomerative clustering procedure which will be explained below. It takes the current set of clusters as the basis for computing the relevance of each feature. For this purpose the distance measure [6], a relevance measure used for attribute selection in decision tree induction, is used. Note that by using this sort of measure we are assuming that the more relevant features are those which better discriminate among the clusters of a given partition. As a result of the process, an ordered set of features is obtained and those which do not score high enough are discarded. A heuristic procedure is used to determine this subset of useful features. The rationale behind this procedure is the assumption that the set of attributes which can discriminate among the clusters of a given partition is smaller for more general levels than for specific ones. The level of generality of the level which is being built is indicated by the NG value specified by the user, so it can be used to heuristically define the set of useful features \( U \) as follows:

\[
U = \{ a \in A \mid Rel(a) \geq m \cdot NG \}
\]

where \( A \) is the initial set of features, \( Rel(a) \) is the relevance of feature \( a \) and \( m \) is the maximum computed relevance for features in \( A \). This procedure does not guarantee in any case that the selected subset is neither the best nor the minimal for any particular task. We are just using a relatively naive approach to make a conservative selection in an unknown domain allowing the system to dynamically discard features which are very likely to be irrelevant.

2.3 Refinement

This stage consists of an agglomerative procedure which iteratively selects the best two candidates to merge and creates a new cluster. However, a number of differences with traditional clustering procedures exist. First, the starting point for this procedure is not the original dataset, but the clusters obtained in the Preprocessing stage or in the previous
generalization if a set of $NG$ values is given. Secondly, the procedure ends when the level of generality indicated by current $NG$ value is achieved and the resulting level does not store all the intermediate pairs of mergings. Therefore, the system does not necessarily produce binary trees. And finally, at each merging step, the current set of clusters is passed to the Reflection step to obtain a weighted subset of useful features which is used in the following computations.

To be able of determining when a given level of generality is reached, we need a measure to characterize cluster generality. ISAAC measures the generality of a cluster using probabilistic analogs of logical sufficiency and necessity. These measures are interpreted as degrees of sufficiency and necessity of the probabilistic descriptions used by the system. From this point of view, they represent a continuous valuation over the sufficiency and necessity properties analogous to the binary one of classical logic, but allowing a greater flexibility. These measures are called continuous sufficiency ($CS$), and continuous necessity ($CN$), and are defined, for a cluster $C_k$ as follows:

$$CS(C_k) = \sum_i \sum_j P(C_k \mid A_i = V_{ij})^2$$

$$CN(C_k) = \sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2$$

$i$ indexes the features of the objects, and $j$ indexes the values of each feature. Both measures can be easily generalized to partitions by simply averaging the results for the set of clusters. These two measures evaluate the average degree of sufficiency or necessity for the features contained in the prototypes of the clusters of a given partition.

Each of the measures is biasing the selection of partitions in a different direction as regards generality of a partition. Favoring $CS$ over $CN$ tends to reward partitions with more general concepts and favoring $CN$ over $CS$ tends to reward partitions with more specific concepts. This property allows the user to bias the process towards the type of partition required, linking the bias to the $NG$ parameter by means of the formula:

$$Gen(P, NG) = (1 - NG) \times CS(P) - NG \times CN(P)$$

This measure indicates whether a given level of generality defined by a given $NG$ value has been achieved. The measure is used to allow merging until a certain level of both $CN$ and $CS$ measures is reached. The initial partition usually will have a negative $Gen$ value due to the high score of $CN$ and, as the generalization progresses, the generality of the partition will tend to zero according to the evolution of the $CS$ and $CN$ measures.

The control structure for the Refinement stage is shown in Table 1. As mentioned before, it follows a typical agglomerative schema which merges pairs of clusters until the desired $NG$ level is reached. However, unlike most of its statistical counterparts, our algorithm does not construct a similarity matrix to decide which pair of clusters should merge. Instead, the algorithm takes advantage of the generality measure defined above and always chooses as the first cluster to merge the one who has the lower generality score. The similarity of this cluster with the rest of the clusters in the partition is then computed, by means of expression 2, to find the most similar one and perform the merging operation. All of this computations (similarity and generality) are done in the context provided by $U$, the subset of useful features obtained in the Reflection stage. This means that only features included in $U$ are considered when computing both measures, which also use the available weights.

The reason for using the generality measure as a heuristic to decide one candidate to merge is twofold. If we use a similarity matrix, since we want to dynamically adjust similarity computations with feature weights, we are forced to update the whole matrix at each step of the process. These

---

**Function** Refinement($P$, $NG$)

$U = \text{Reflection}(P, NG)$

while Generality($P$, $NG$, $U$) $< 0$ do

Let $C$ be the least general cluster in $P$

Compute the similarity between $C$ and the rest of clusters in $P$ using $U$

Merge $C$ with the most similar concept in $P$

$U = \text{Reflection}(P, NG)$

endwhile

---

Table 1: Algorithm for the Refinement stage of ISAAC

---
computations will result in a cubic complexity for the algorithm with respect to the number of clusters initially considered. The heuristic used, allows to maintain a quadratic complexity for the algorithm. On the other hand, the heuristic should bias the algorithm to reach levels in which each cluster approximately corresponds to the same level of abstraction, and hence improve the understandability of the obtained clusters.

3 A simple example

In this section, we present a very simple example taken from [3] to illustrate the basic behavior of the proposed method. The example corresponds to a simple animal domain containing 8 objects and 5 features and is showed in Table 2.

First, the Preprocessing step is performed. Assume that the α value selected is 0.80 and that objects are presented in the order shown in Table 2. This means that only objects/clusters with a similarity between them greater or equal than 0.80 can be merged. The incremental nearest neighbor clustering proceeds incrementally creating one cluster for the initial object, and processing the rest of the objects one by one. The second object is not similar enough to the first object, so it is placed in a separate cluster. The same holds for the second, third and fourth objects. However, the similarity between the fifth object -reptile2- and the first -reptile1- is 0.80, so they are merged into a unique cluster. The next object is again placed in a separate cluster. When observing mammal2, the similarity with mammal1 is again 0.80, so this pair of objects is also merged. The same holds for amphibian1 and amphibian2. Therefore, as a result of the Preclustering step, the initial set of 8 objects is reduced to the following 5 clusters:

<table>
<thead>
<tr>
<th>mammal1</th>
<th>hair</th>
<th>four</th>
<th>regulated</th>
<th>internal</th>
<th>brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>reptile1</td>
<td>corn-skin</td>
<td>imp4</td>
<td>unregulated</td>
<td>internal</td>
<td>green</td>
</tr>
<tr>
<td>amphibian1</td>
<td>moist-skin</td>
<td>three</td>
<td>unregulated</td>
<td>external</td>
<td>red</td>
</tr>
<tr>
<td>bird</td>
<td>feathers</td>
<td>four</td>
<td>regulated</td>
<td>internal</td>
<td>red</td>
</tr>
<tr>
<td>reptile2</td>
<td>corn-skin</td>
<td>imp4</td>
<td>unregulated</td>
<td>internal</td>
<td>gray</td>
</tr>
<tr>
<td>fish</td>
<td>scales</td>
<td>two</td>
<td>unregulated</td>
<td>external</td>
<td>red</td>
</tr>
<tr>
<td>mammal2</td>
<td>hair</td>
<td>four</td>
<td>regulated</td>
<td>internal</td>
<td>gray</td>
</tr>
<tr>
<td>amphibian2</td>
<td>moist-skin</td>
<td>three</td>
<td>unregulated</td>
<td>external</td>
<td>green</td>
</tr>
</tbody>
</table>

Table 2: A simple animal domain.

This set of clusters is the input to the Refinement and Reflection stages. Initially, the later computes a first set of feature weights and selects the most relevant features according to the NG value. Let's assume that the user provides a single NG value of 0.5, that, roughly, requires a balance between necessity and sufficiency. Using this value, two features, BodyCover and HeartChamber, are selected for the following computations. From the CN and CS scores of each cluster, Generality scores are computed and the following values are obtained:

\[ Gen(C_1, 0.5) = -0.24 \]
\[ Gen(C_2, 0.5) = 0.00 \]
\[ Gen(C_3, 0.5) = 0.00 \]
\[ Gen(C_4, 0.5) = -0.38 \]
\[ Gen(C_5, 0.5) = 0.00 \]

The global Generality of the partition is -0.62, indicating that the level of abstraction is under that indicated by the NG parameter. Generality scores suggest that some clusters posses the desired level of abstraction, but these scores are computed by using only the currently selected features and weighting computations, so that they can be modified in later steps. Clearly, by only considering the two first features in the dataset, \( C_2, C_3 \) and \( C_5 \) are defined by necessary and sufficient conditions and, therefore, they appear to have the required level of abstraction. Analyzing the Generality scores, Cluster \( C_4 \) is selected as the best candidate to merge. Note that by considering only the two features mentioned above, this cluster shares one value with \( C_1 \), thus
decreasing its CS score and, hence, giving a lower Generality score.

The next step consists in selecting the most similar cluster to $C_i$ according to set of useful features. This cluster turns out to be $C_1$ and both clusters are merged, producing a partition with 4 clusters:

$$C_1 = \{\text{mammal1, mammal2, bird}\}$$
$$C_2 = \{\text{reptile1, reptile2}\}$$
$$C_3 = \{\text{amphibian1, amphibian2}\}$$
$$C_4 = \{\text{fish}\}$$

Again, the Reflection stage computes feature relevances. Two additional features have increased their relevance enough to be selected, BodyTemp and Fertilization. Their relevances are not as high as that of the other two selected features, thus reflecting the conservative strategy of the feature selection heuristic. New Generality scores are computed as follows:

$$\text{Gen}(C_1, 0.5) = 0.45$$
$$\text{Gen}(C_2, 0.5) = -0.42$$
$$\text{Gen}(C_3, 0.5) = -0.35$$
$$\text{Gen}(C_4, 0.5) = -0.46$$

The cluster combining the mammals and the bird scores a Generality above 0, because it possesses features that allow to discriminate between this cluster and the others, therefore increasing the CS score, but shows a slight internal variability that decreases the CN score. The system has dynamically adapted its perspective of the current set of hypotheses by adding features that in the first step were deemed as irrelevant. This new perspective is reflected in the change of the Generality score in the selected cluster $C_4$ compared to the previous score.

The selected cluster is merged with the most similar one, in this case, $C_3$. In the resulting partition, only one cluster scores a Generality under 0, but it is compensated by the scores of the other clusters, giving a global Generality score above 0, and stopping the process. Heart-Chamber, BodyCover, BodyTemp and Fertilization are returned, in this order, as the most relevant features, completely discarding Color. Actually, the first feature is a necessary and sufficient condition for the resulting clustering structure, but again, the system makes a conservative estimation. The final returned partition is the following one:

$$C_1 = \{\text{mammal1, mammal2, bird}\}$$
$$C_2 = \{\text{reptile1, reptile2}\}$$
$$C_3 = \{\text{amphibian1, amphibian2, fish}\}$$

None of the intermediate clusters are stored, so that the resulting tree consists of two levels, the bottom level with the initially created 5 clusters, and the top level with the final 3 clusters. Clusters at the bottom level are the children of clusters in the top level. If the user specifies an additional $NG$ value, a new level will be created having the current top level clusters as its children.

### 4 Empirical evaluation

The framework outlined in this paper may suggest several lines for evaluation, but in our experiments we focused on two aspects, the effect of the Pre-processing stage, and the effect of the feature selection mechanism. Particularly, we want to check the degree of compression that can be achieved in the first stage of the process and if this compression can decrease the quality of the final results. Also, we are interested in analyzing the results from the point of view of comprehensibility to confirm the utility of feature selection. Next, we present two sets of experiments, the first using artificial data in order to explore the scalability of the system under controlled conditions, and the second with a real dataset.

#### 4.1 Artificial data

A first set of experiments were conducted using artificially generated datasets. Several datasets with 1000, 2000, 3000 and 4000 objects, were constructed by using 15 features (denoted A..J) with 3 values each. Datasets are generated representing an uniformly distributed underlying structure of 2 classes. Six feature-value pairs were randomly chosen in order to represent necessary conditions for class membership as follows:

$$D = 2 \land G = 2 \land I = 1 \rightarrow \text{class1}$$
$$A = 2 \land G = 1 \land J = 3 \rightarrow \text{class2}$$

The goal of these experiments was to test the scalability of the clustering algorithm to the number of objects in the dataset. In order to construct a complete hierarchy, ISAAC was run with a set of $NG$ consecutive values with an 0.1 increment until achieving a two-class top level partition. Although a two-class level can be created by selecting the appropriate $NG$ value, by constructing several levels we get a more realistic simulation of the utilization of the system. Figure 3 depicts the averaged
time performance of 25 independent runs with random orderings and different \( \alpha \) values. Results show clearly a linear increase in time as the number of objects increase, far from the quadratic complexity of agglomerative hierarchical clustering algorithms.

To assess the performance of the system we computed the average percentage of misclassified objects with respect to the original classes for each of the runs. For each cluster in the top level of the hierarchy, class labels are designated according to the majority value for the label in the cluster. Objects not corresponding to the designated class are deemed as misclassified. This measure serves as an indicator of the purity of the resulting clustering structure. ISAAC consistently obtains very high quality clusterings, with a 90% of purity. Despite the regularity of the domain still there are a few misclassified objects, but this is possible due to the order sensitivity of the preprocessing step. These results demonstrate that the preprocessing step can greatly reduce computational complexity yet maintaining a good performance. In addition, all the hierarchies constructed by the system showed a low average number of features per node, around 5. Actually, at the top level just one feature was selected most of the times, feature G, which according to the previously showed rules is a necessary and sufficient feature for describing the classes in this dataset.

4.2 Real data

An additional set of experiments were carried out using the mushroom dataset from the UCI repository. The mushroom dataset consists of 8124 mushroom descriptions represented by 22 nominal features belonging to two classes, edible and poisonous. As before, ISAAC was run with a set of \( NG \) consecutive values with an 0.1 increment until achieving a two-class top level partition. Depending of the initial partition considered, the number of levels ranged between 7 and 9 in order to get the desired number of clusters.

To evaluate the quality of the discovered hierarchies, we measured the purity of the resulting clusters in the top level as regards the original mushroom division into edible and poisonous by computing the percentage of misclassified objects as in the previous experiment.

Table 3 shows the results for 50 ISAAC runs on the mushroom dataset using different \( \alpha \) values. Some data from the table is graphically depicted in figure 4, showing that a great amount of compression may be achieved by just using \( \alpha \) values around 0.85, which are relatively high. As expected, decreasing the \( \alpha \) value allows for greater data compression. Obviously, the amount of data compression is directly correlated with the running times of the system. Figure 4 also shows the average purity scores and standard deviations achieved by the system for each different \( \alpha \) value. Clearly, there is no relationship between these two factors since results appear to be somewhat variable. In fact, we cannot expect to find such a relationship because different initial partitions may bias the subsequent feature selection steps in a different and not easily predictable manner. Actually, the interest here was in demonstrating that the system is able to reach high scores with some of the compressed datasets. The impact of the compression in the final results may vary between different datasets and users should experiment with different \( \alpha \) values until obtaining a suitable partition.

For comparison purposes, we also ran the well-known AUTOCLASS program [1], obtaining a purity score of 0.90. Since ISAAC results do not appear to follow a very homogeneous distribution, table 3 shows an additional column with the percentage of ‘good clusterings’ obtained with each \( \alpha \) value, considering as good clusterings those with a purity score over 0.85. It is not possible to establish a direct comparison between the two systems, since ISAAC generates hierarchical clusterings as opposed to the flat clusterings of AUTOCLASS. However, the AUTOCLASS score indicates that, despite the variability of its results, our approach may achieve good quality clusterings with a reasonable amount of compression.

Table 3 also gives us a picture of the capabilities of the feature selection mechanism. In average, the
system only uses between a 10-20 % of features from the initial feature set in order to discriminate between the different hierarchy nodes. This demonstrates the ability of ISAAC to bias the clusterings towards simplicity and, hence, provides results which should be easier to interpret.

5 Concluding remarks

We have presented ISAAC, a conceptual clustering system which combines statistical procedures with symbolic learning oriented heuristics. The system partitions the clustering process into three stages. The first one, accomplishes two goals, namely, to achieve a compression of the dataset, and to provide an initial set of hypotheses for the rest of the process. The following stages are performed in a collaborative manner. One of them is responsible of obtaining a weighted subset of features from the actual cluster structure. The second, uses this information to generalize the current partition and generates new cluster structures, from which new weights may be computed.

One novel feature in our approach is the definition of a cluster generality measure. This measure allows to parameterize the system in order to allow the user to bias the resulting structures towards the desired level of generality. We think that this sort of parameters are more meaningful to users than those in traditional statistical clustering. Users should be more comfortable interacting with systems which have parameters with a semantically clear interpretation [9].

Our preliminary experiments suggest that this approach achieves two interesting goals from the standpoint of data mining tasks. First, the system should efficiently deal with large datasets by means of the initial Preprocessing step. Second, the resulting hierarchical clusterings should be easy to understand since the system provides descriptions with an important reduction in the number of features used.

On the other hand, results seem to show a large variability with respect to the original labeling in the used dataset. However, we cannot assume the
original labeling to be the only interesting underlying structure present in the dataset. Also, variability may be due to the global interaction between the different biases provided by the preprocessing and the feature selection mechanisms. This suggests a future research to explore the isolate and joint influences of all these biases.

Some aspects of the system deserve further research such as the similarity metrics used, the feature weighting measure or the feature selection strategy. Currently, some of these components are chosen largely based on intuitions and empirical results. Probably, a more accurate study of the properties of the different measures used can lead to a better understanding of the behavior of the system.

The modular design of ISAAC easily allows to extend the system. We plan to explore different feature selection measures and their impact in performance tasks like prediction, given some promising results obtained with the system in this task [8]. Also, a set of discretization procedures will be added to the Preprocessing stage to provide the system with the ability to deal with numerical valued features. Finally, we are working on extending the Reflection stage in order to constraint the Refinement process using declarative knowledge provided by the user.

Acknowledgments

We thank the anonymous referees for many valuable comments on an earlier draft.

References


Dynamic Local Feature Selection in Incremental Clustering

Luis Talavera
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6, Jordi Girona 1-3
08034 Barcelona, Catalonia, Spain
talavera@lsi.upc.es

Abstract

In this paper we describe a preliminary study into the use of feature selection in incremental hierarchical clustering. Our aim is to add this capability to the clustering system, still maintaining the incremental nature of the learning process. This constraint lead us to consider a dynamic feature selection mechanism which is performed parallel to the clustering process. In addition, feature selection is performed in a local manner for every individual node. We propose a mechanism based upon an ordering scheme that relies on feature relevance computations. Preliminary experiments suggest that feature selection can provide a significant speed-up in the learning and prediction tasks without decreasing neither the quality of the obtained clusterings nor their predictive accuracy.

Keywords: Clustering, feature selection, incremental learning.

1 Introduction

The widespread use of information technologies produces an growing amount of data which is too huge to be analyzed by manual methods. For instance, data mining problems on the World Wide Web provide volumes of data containing both, many features and many examples. Inductive learning methods appear to be a useful method for automatically extracting useful information from this data or for assisting humans in this process. A problem related to this sort of data is the presence of a large number of features that might tend to decrease the effectiveness of learning algorithms, especially if most of these features appear to be irrelevant with regard to the learning task. In fact, feature selection is a central problem in inductive learning as suggested by the growing amount of research in this area [2, 3, 8, 9, 10, 12].

Most of the work concerning feature selection has been carried out under the supervised learning paradigm, paying little attention to unsupervised learning tasks and, particularly, clustering tasks. Clustering is a form of unsupervised learning used to discover interesting patterns in data. Hierarchical clustering methods construct a tree-structured clustering where sibling clusters partition the observations covered by their parent. Hierarchical structuring provides a view of data at several levels of abstraction, yet flexibly allowing different flat partitions to be extracted during data analysis. In addition, incremental hill-climbing approaches seem to be an inexpensive way of constructing hierarchical clusterings, since the approximate cost of incorporating new observation tends to be logarithmic with respect to the number of observations.

In this paper we describe a preliminary study into the use of feature selection in incremental hierarchical clustering. We propose an implementation based upon the well-known COBWEB system [5]. Our aim is to add this capability to the clustering system, still maintaining the incremental nature of the learning process. This constraint lead us to consider a dynamic feature selection mechanism which is performed in parallel with the general clustering process and in a local manner for each node. We propose several dimensions to evaluate the benefits of feature selection other than accuracy improvement. Particularly, these benefits are concerned with the efficiency of the learning task, the efficiency of the performance task and comprehensibility.
2 Feature selection in hierarchical clustering

Typically, the primary goal of feature selection is intended to make inductive learning algorithms more robust in the face of irrelevant features. That is, the number of training cases needed to reach a given level of accuracy has to grow slowly with the number of irrelevant features. However, feature selection may provide other benefits besides improving predictive accuracy. We may, for instance, focus in a description task rather than in prediction. In such a case, it is likely that we would have some cluster quality measure and the goal would be to optimally partition the data with respect to this measure. On the other hand, it is possible to take advantage of feature selection by improving the prediction task without necessarily increasing accuracy if we focus in improving the efficiency of the task. We briefly introduce three different dimensions to evaluate the particular benefits of feature selection in the hierarchical clustering task:

- **Efficiency in the learning task.** Commonly, hierarchical clusterings are polythetic classifiers, that is, they divide objects based on their values along multiple features. Particularly, they tend to use the whole feature set at each node to decide how to cluster a new object. Since this decision has to be made along several nodes in the tree, the number of features present in the data directly influences the complexity of the clustering process. If we apply feature selection to reduce this complexity, we should expect to obtain clusterings of similar or better quality that we would have obtained by using all the available features.

- **Efficiency in the performance task.** The number of features may also influence a performance task such as prediction. When using a hierarchical clustering to classify an unseen observation in order to infer unknown properties, the number of features still has a strong influence in the complexity of the process in the same manner we have described above. Again, selecting an appropriate subset of features may reduce this complexity. However, in this case the concern is not in cluster quality, but in maintaining or raising the level of performance achieved by the system.

- **Comprehensibility of the results.** As we noted before, with the exception of approaches that operate on logical descriptions, clustering systems usually make use of all the available features at each node of the hierarchy. Reducing the number of features used in the clustering process allow to provide shorter cluster descriptions to the user. Short descriptions tend to be more readable and, hence more comprehensible.

Another interesting remark to be made here is that hierarchical clusterings can take advantage of local feature selection at the node level. This sort of feature selection occurs at every individual node of the hierarchy independently of the others. In this manner, the clustering system may globally use all the feature set and select different feature subsets among different local parts of the observation space. This view of relevance can be viewed as similar to the local weighting approaches used in lazy learning [1, 16].

3 An overview of COBWEB

COBWEB [5] is a hierarchical clustering system that constructs a tree from a sequence of observations. The system follows a strict incremental scheme, that is, it learns from each observation in the sequence without reprocessing previously encountered observations. An observation is assumed to be a vector of nominal values $V_{ij}$ along different features $A_i$. COBWEB employs probabilistic concept descriptions to represent the learned knowledge. In this sort of representation, in a cluster $C_k$, each feature value has an associated conditional probability $P(A_i = V_{ij} | C_k)$ reflecting the proportion of observations in $C_k$ with the value $V_{ij}$ along the feature $A_i$.

The strategy followed by COBWEB is summarized in Table 1. Given an observation and a current hierarchical clustering, the system categorizes the observation by sorting it through the hierarchy from the root node down to the leaves. At each level, the learning algorithm evaluates the quality of the new clustering resulting from placing the observation in each of the existing clusters, and the quality resulting from creating a new cluster covering the new observation. In addition, the algorithm considers two more actions that can restructure the hierarchy in order to improve its quality. Merging attempts to
Function COBWEB(\text{observation}, \text{root})

1) Incorporate observation into the root cluster.
2) If root is a leaf then
   return expanded leaf with the observation.
else choose the best of the following operators:
   a) Incorporate the observation into the best host
   b) Create a new disjunct based on the observation
   c) Merge the two best hosts
   d) Split the best host
3) If a), c) or d) recurse on the chosen host.

Let $A$ be a set of features
Let $\tau$ be the feature selection threshold
Function select\_features($A$, $\tau$)
    compute feature weights($A$)
    $\text{max}_w = \max \{ \text{weight}(A_i) \mid A_i \in A \}$
    return $\{ A_i \mid \text{weight}(A_i) \geq \text{max}_w \times \tau \}$

Table 2: A method for feature selection based on an ordering scheme.

combine the two sibling clusters which were identified as the two best hosts for the new observation; splitting can replace the best host and promote its children to the next higher level. The option that yields the high quality score is selected and the procedure is recurred, considering the best host as the root of the recursive call. The recursion end when a leaf containing only the new observation is created.

In order to choose among the four available operators, COBWEB uses a cluster quality function called \emph{category utility} defined for a partition $P = \{C_1, C_2, ..., C_n\}$ of $n$ clusters as

$$\sum_k P(C_k) \sum_i (P(A_i = V_{ij})^2 - P(A_i = V_{ij})^2)$$

This function measures how much a partition $P$ promotes inference. It rewards clusters $C_k$ that increase the predictability of feature values within $C_k$. Generally, the system is evaluated in terms of its predictive accuracy along all the features. However, CU may also be interpreted as a trade-off between intra-cluster similarity and between clusters dissimilarity, so that, often, the first level is regarded as the optimal partition of the data.

4 A dynamic feature selection mechanism

Feature selection can be viewed as a search through a space of possible feature subsets. The search, may be performed either outside the induction algorithm as a \emph{preprocessing} step, or may be \emph{embedded} within the induction algorithm and be performed parallel to the general learning process. The former methods can be deemed as \emph{static}, since they select a subset of features which is used during all the learning process, while the later are \emph{dynamic} in the sense that they can adapt the subset of selected features at every step of the learning process. Batch clustering systems may perform either static or dynamic selection. However, in incremental systems, feature selection has to be performed dynamically as new observations are processed.

John et al. [8] make a further division of feature selection methods into filter and wrapper methods. Filter methods are independent of the induction algorithm used, while wrapper methods evaluate alternative feature sets by running the induction algorithm on the training data. The major disadvantage of wrapper methods is the high computational cost resulting from calling the induction algorithm. Since dynamic feature selection in an incremental clustering system has to be performed every time a new instance is observed, application of wrapper methods appears to be unfeasible.

We propose a filter method of feature selection based on an \emph{ordering} scheme. A weight is individually computed for each feature and features are ordered according to these weights. Then, the $k$ features with the highest value are selected. Instead of directly choosing a value for $k$, we define a \emph{feature selection threshold} $\tau$ in the $[0,1]$ range such that the weight required for a feature to be selected is higher for higher $\tau$ values. Our method uses the maximum computed weight as a baseline to determine which features are selected as shown in Table 2. Note that, if we assume relevances to be positive, when $\tau = 0$ there is no feature selection at all, so reducing the original algorithm to a special case of our approach.

This method can be easily incorporated into COBWEB by slightly modifying the control strategy showed in Table 1. First, we need to add an ad-
ditional step between steps 2 and 3 of the existing algorithm. In this step a call to the select_features function is performed, obtaining a subset of relevant features to be stored in the current root node. Second, at each classification step, the computation of the quality function must be modified in such a way that only the subset of relevant features stored in the current root node is used.

5 Feature relevance in clustering

In order to choose the more relevant features, we need to obtain an ordering of the feature set. To determine an ordering among the features we need to define a feature relevance measure and, hence, we must commit to some definition of relevance. There are a number of definitions in the literature for feature relevance [3], but, intuitively, we can say that a feature is relevant if it cannot be removed without loss of prediction accuracy. Since these definitions are almost universally applied to supervised learning, often prediction accuracy is measured over the class labels. In clustering tasks, there are no labels available, but we can still assume that each discovered cluster represents a different class and apply the same definition. However, it is not clear how successful this approach could be for a clustering task, since the classes are not known in advance, rather they are created during the learning process. Moreover, several authors have abstracted out an unsupervised performance task from that of supervised systems; unsupervised learning supports prediction not only of a class label, but of every feature missing from the observations. Therefore, if we would want to apply the former definition of relevance, we should consider as relevant those features that cannot be removed without loss of the average prediction accuracy over all the remaining features. Note that if we define relevance as regards the feature ability to predict the cluster in which a new observation would be classified, we are stressing description as the primary goal of the learning process while if we consider relevant those features that promote inference of other features, we are focusing on a prediction task.

We will consider two alternative measures of feature relevance. The first measure was proposed by Gennari [7] in the context of his CLASSIT system, an extension of COBWEB to deal with numeric features. Gennari refers to this measure as salience. He defines the relative salience of a feature as its contribution to category utility (see equation 1) in a clustering. More formally, for a given feature $A_i$, salience is defined as follows:

$$\sum_{C_k} P(C_k) \sum_{j} |P(A_i = V_{ij}|C_k) - P(A_i = V_{ij})|^2$$

Category utility is defined as a function measuring the predictive power of a partition over all the features, so it is reasonable to think that features that maximize the above measure will maximize the predictive ability of the clustering.

We can also give a second definition of feature relevance related to the descriptive ability of features. We can identify this ability with the information that a feature provide for discriminating among the existing clusters in a partition. This idea was proposed in [13] and has been used in the ISAAC clustering system with some success [15]. Specifically, ISAAC measures feature relevance by using the following expression:

$$-\sum_{C_k} P(C_k) \log_2 P(C_k) - \sum_{i,j} P(A_i = V_{ij}) \log_2 P(A_i = V_{ij}) - \sum_{i,j} P(A_i = V_{ij} \wedge C_k) \log_2 P(A_i = V_{ij} \wedge C_k) - 1$$

The measure is known as the distance measure and was proposed in the context of the induction of decision trees [11]. Adapting this measure appears to be a natural approach since research on attribute selection measures for inducing decision trees is closely related to the idea of relevance as a measure of descriptive power.

An aspect worth to mention is that in the context of the COBWEB system, prediction and description might not be so different as it could seem. Theoretically, the category utility metric leads the learning process towards clusterings with provide good predictive ability over all features. However, as we have pointed out before, the metric may also be interpreted as a trade-off between within cluster and between clusters similarity. Therefore it is possible to consider that the formed predictive clusterings are at the same time good domain descriptions, thus blurring the difference between the two concepts.

6 Preliminary results

In order to evaluate our feature selection mechanism, we ran experiments on three datasets from the UCI repository. Our aim is to test the two proposed relevance metrics as regards to the different
goals we mentioned in section 2. Also, we want to gain some insight into the effect of different degrees of focus that could be modeled by tuning the $\tau$ parameter.

In our first experiment, we performed ten runs with different random orderings for each data set. The goal was to check if we could speed-up the learning process by focusing only on relevant features, while still maintaining the same level of quality. Usually, it is assumed that COBWEB finds the optimal partition of data in the top level of the hierarchy, so the category utility score for this level is used as a quality measure. In addition, we can consider this score as an indicator of the estimated predictive power of the clustering over all the features. Therefore, by using this quality measure we are assessing, in some sense, two different potential benefits of feature selection. In order to evaluate the efficiency of the learning process, we computed the average number of feature comparisons needed to cluster the observations in the data set. This number is calculated by summing the total number of features involved in evaluating the category utility metric for the different clustering choices. For instance, if an observation is being clustered in a root node with three children and using a subset of $n$ features, we need to perform $3n$ feature comparisons to evaluate the CU of incorporating the observation to each of the siblings; $n$ feature comparisons to evaluate the creation of a new class; $n$ feature comparisons to evaluate merging and $n$ more to evaluate splitting. This addition is performed for all the classification path followed by each observation. We think that this way of measuring efficiency give us a better empirical approximation of the complexity of the clustering process than, for instance, the average of features per node. On the other hand, we use this later measure as a measure of simplicity of the obtained clusterings which is usually deemed as a measure of comprehensibility.

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>Soybean small</th>
<th>Soybean large</th>
<th>Zoo database</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>$\tau$</td>
<td>$\tau$</td>
<td>$\tau$</td>
</tr>
<tr>
<td>$0.00$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.05$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.10$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.15$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.20$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.25$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.30$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.35$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.40$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.45$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
<tr>
<td>$0.50$</td>
<td>$1.47$</td>
<td>$1.47$</td>
<td>$1.02$</td>
</tr>
</tbody>
</table>

Table 3: CU scores, efficiency and comprehensibility for several $\tau$ values.
Results are shown in Table 3 for different $\tau$ thresholds, including the null value, which corresponds to the original COBWEB algorithm without feature selection. In all the data sets, even with a small $\tau$ value, the amount of speed-up achieved is impressive. The feature selection mechanism consistently improves efficiency without significantly decrease cluster quality by reducing the number of feature comparisons needed to build the hierarchy in 55-70%. Further improvements can be achieved at the expense of decreasing the CU scores obtained. The number of features per node drops in a similar amount, thus providing simpler and, therefore, more comprehensible, cluster descriptions.

As regards to $\tau$ values, it appears that values between 0.10 and 0.20 guarantee a nearly optimal quality and a reasonable speed-up. On the other hand, none of the feature relevance measures seem to work significantly better than the other. Perhaps, it is remarkable that the salience measure shows a less marked decrease in the CU scores than the distance measure. However, this could be explained by the fact that the former measure is derived from the CU metric itself. In fact, it could appear more surprising that guiding feature selection with a measure not derived from CU, still gives high CU scores.

In our second experiment, we measured the effect of feature selection in improving efficiency in a performance task. We computed learning curves for each dataset by measuring the accuracy of predicting the class label of previously unseen testing observations. The class labels were masked during training and used only for evaluation purposes. Although the proposed task is a predictive one, we can also thing about it as measuring the ability of the system of obtaining good domain descriptions, assuming that this ability can be assessed by measuring the power of the system in ‘rediscovering’ the existing classes. As before, we recorded the number of feature comparisons needed to cluster each observation, but this time only for the testing set.

Figure 1 shows the learning curves for the best $\tau$ values of both relevance metrics and for the original COBWEB algorithm without feature selection. The graphs suggest that the distance measure provides a faster learning rate than salience and the original algorithm. As we remarked earlier, we are interested in obtaining an additional benefit from feature selection over an accuracy improvement. Figure 2 shows the efficiency curves corresponding to the learning curves in Fig. 1. As it could be expected, the original algorithm increases the complexity of adding a new observation as more instances are seen. This is because the number of internal nodes created grows as more observations are seen, forcing the system to perform more feature comparisons in order to classify each new observation. Interestingly, feature selection compensates for this increase of the number of nodes by reducing the number of features indexing those nodes. Figure 2 shows this effect and how it helps to dramatically reduce the cost of assimilating new observations.

7 Related work

Few research has been carried out on the feature selection problem in incremental hierarchical clustering. We have already referenced Gennari’s work incorporating a feature selection mechanism into CLASSIT [7]. This work appears to be an attempt to define a dynamic and local feature selection scheme as defined here. It also shares our efficiency concerns, but its results are very limited. More recently, Devaney and Ram [4] proposed an attribute-incremental concept creator which efficiently deals with feature selection. This approach differs to the one presented in this paper in that it is a pre-processing approach and do not operate incrementally as learning proceeds.

8 Concluding remarks

We have presented the problem of feature selection in hierarchical clustering. Although this task may benefit from the traditional aim of feature selection, that is, increasing accuracy, we have proposed other dimensions for evaluating this benefits mainly concerned with efficiency and comprehensibility. In addition, hierarchical clusterings suggest a local feature selection scheme for each node in the hierarchy.

An implementation of these ideas under the framework of the COBWEB system has been proposed based upon an ordering scheme that orders the feature set according to some relevance measure and selects those that appear to be more relevant. To maintain the incremental nature of the system we have applied a dynamic feature selection scheme that runs parallel to the clustering process instead of being a preprocessing step. Experiments have been carried out using two different relevance measures and varying the degree of focus applied. Preliminary results suggest that important benefits
may be obtained from the efficiency standpoint in both, learning and prediction tasks. In addition, by reducing the size of the feature set at each node we obtain more readable, and hence, comprehensible results.

We have shown that it is possible to reduce the number of features per node in the constructed hierarchies, yet obtaining high quality clusterings. We have assumed that, given the nature of the category utility metric, these hierarchies should also provide a good predictive power over all the feature set. However, it has been pointed out that category utility may not be always well correlated with the predictive ability of the clusterings. This suggest future work in testing the proposed method on a more general prediction task than predicting the class label, the average prediction of every feature present in the data. We have demonstrated elsewhere [14] that feature selection may improve efficiency on this task. In addition, future research should extend this work by testing and compare additional feature relevance measures and feature selection schemes, as for instance, schemes relying on performance feedback. These points deserve further investigation since multiple feature prediction has been more commonly used to evaluate unsupervised learning systems [6] than label prediction.

References


Recognising Natural Objects in Outdoor Scenes: A Survey.

J. Freixenet†, A. Casals‡‡, J. Batlle† and J. Martí†

† Computer Vision and Robotics Group.
Institute of Informatics and Applications.
University of Girona.
Av. Lluís Santaló s/n, 17071 Girona, Spain.
(jordif,jbatlle,joanm)@eia.udg.es

‡‡ Dep. of Automatic Control and
Computer Engineering.
Polytechnical University of Catalonia.
C/ Pau Gargallo 5, 08028 Barcelona, Spain.
casals@esaii.upc.es

Abstract

This paper surveys some significant vision systems dealing with the recognition of natural objects in outdoor environments. The main goal of the paper is to present a classification depending on how the approaches perform the segmentation and the recognition processes. The motivation of this study is to overview the state of the art with the aim to advance a little further in the process of outdoor scenes description.

keywords: computer vision, scene understanding, outdoor scenes, control strategies, natural object recognition.

1 Introduction

Scene Understanding is one of the most interesting subjects involving the fields of Computer Vision and Artificial Intelligence. Most of the efforts in this discipline have been focused on improving image segmentation techniques and finding efficient knowledge representations, that have permitted to attain some relevant results. In particular, the increasing interest in developing applications in outdoor environments can be pointed out. However, outdoor scenes are particularly hard to treat in terms of lighting conditions, due to weather phenomena, day time and seasons.

The study has been motivated mainly by the need to advance a further step in the process of scene description in the frame of the work refereed in [25,26], initially restricted to urban environments. In this previous work we looked for some pre-defined kind of objects in the scene: buildings, vehicles, persons... Now, we are interested in increasing the number of kind of objects to detect and recognize to describe the scene better. Our goal is to detect new objects that present a wide variety of features depending on the class, season or weather conditions, as for example trees or bushes.

This paper discusses the most relevant approaches developed in recent years, related to the recognition of natural objects in outdoor environments. The approaches have been classified as General Purpose Vision Systems (GPVS) or Specialised Purpose Vision System (SPVS) according to their purpose. In general, GPVS and SPVS can be considered as knowledge-based vision systems; that is, systems that perform the following steps: segmenting the input image, looking for "object hypothesis", and validating them by semantic restrictions. The surveyed vision systems make use of a large variety of knowledge representations, from simple vectors to complex structures. Note that there is no attempt to survey the different knowledge representation techniques; that task has already been accomplished in [9,18,25]. The analysed approaches use static, colour images as input. We have decided to exclude the approaches that use other sensor systems, such as sonar, lasers or scanners. Although it is widely known that these sensors are very useful to obtain 3D information, they do not provide colour information which is especially relevant to know the nature of the objects.
Some other surveys about outdoor scene understanding can be found in the related literature. However, these surveys emphasise more on the concepts involved than on the application itself. Following this philosophy, Haralick and Shapiro [18] emphasised knowledge representations, control strategies and information integration. They discussed several key systems in order to illustrate various techniques and applications. Recently, Buxton [4] reviewed several vision systems with the idea to give promising directions for future research. She emphasised the role of context, control and learning. She argued that reasoning is the main focus of the work in visual interpretation and distinguishes four major approaches: constraint-based vision, model-based vision, formal logic and probabilistic frameworks. Several key systems illustrate the ideas of these four approaches. Finally, Crevier and Lepage [9] reviewed the role of knowledge in knowledge-based vision systems. They classified the kinds of knowledge required for image understanding and examined how these various kinds of knowledge have been represented. They pointed out promising trends such as a standardised image understanding environment, agent-based representation or automatic learning techniques.

2 GPVS versus SPVS approaches for Objects Recognition

The objective of GPVS consists on recognising and localising the significant imaged objects in the scene and identifying the relevant object relationships [11]. The typical scheme that a GPVS often uses was defined by Fischler in 1978 [14] and is currently assumed: first, it is necessary to partition a scene into regions or primitive objects; these objects are then characterised by a fixed set of attributes, and the scene itself is characterised by linking the objects to each other. The paradigm thus involves the decomposition or segmentation, the naming (labelling) or classification, and the synthesis or description.

On the other hand, SPVS are conceived to carry out specific tasks, defining, structuring, and applying knowledge relevant to their task domain. A typical SPVS applies specialised segmentation and recognition methods in order to find specific objects. Hanson et al. refer to SPVS as systems to cover a more humble goal, and as succeed systems [16].

Summarising, the main difference between GPVS and SPVS for objects recognition, is its purpose; while GPVS try to interpret the global scene, SPVS is centred only on a specific object. But, besides its purpose, it is possible to differentiate GPVS from SPVS by its initial step: the segmentation. Segmentation is a key issue in object recognition, scene understanding and image understanding, as pointed out by several authors [16,24]. The better the segmentation process is, the easier the process of understanding and recognising will be. Segmentation in GPVS is done without using semantics related to the expected objects, whereas in SPVS, the segmentation is specialised and uses all the available knowledge related to the expected object.

Fig. 1 advances a classification of the studied approaches according to their purposes, corresponding to the works carried out during the last 20 years. Note that GPVS approaches, characterised by using general purpose segmentation methods, are classified into two groups, depending on whether they are based on regions or objects. The approaches dealing with regions, try to relate each region with the appropriate object-model. The approaches dealing with objects, apply specific procedures for each object seeking regions which match with the object. So, specific feature vectors are selected for each object model.

3 Outdoor scenes vision systems

This section describes the main characteristics of the most significant vision systems intended to recognise natural objects in outdoor scenes. The different approaches are arranged following the proposed classification outlined in Fig. 1.

3.1 GPVS based on regions

GPVS based on regions are approaches characterised by the following: the input image is segmented by using non-purposive techniques; the resulting regions are characterised by a number of features; regions are labelled as a result of a matching operation; all the expected objects (labels) are recognised by using the same features.

In 1978, Bajcsy and Joshi presented a world model for natural outdoor scenes in the framework of production rules [2]. They are interested in finding relationships in the real world, which impose a partial order in a set of objects. In order to accomplish with these relationships, the system is structured in a database, a set of rules and an interpreter. The database contains information about the state of the world: it expresses specific object knowledge, spatial relations
between objects, and subclasses (objects which are considered as a part of other objects). The recognition is performed by checking only partial features; that is, recognising by matching partial information. The results they present demonstrate that, without any tuning, the system finds the ground, the sky and the horizon skyline in scenes which are visually very different.

Levine et al. proposed a scheme consisting of a collection of analysis processors, each of which is specialised in a particular task. The processors compete and cooperate in an attempt to determine the most appropriate label to each region [23,24]. They agreed with the VISIONS group, in that the process to follow to analyse a scene in order to understand its content, should be separated in two steps: the segmentation process, and the interpretation of the segmented results. This assumption is a widely accepted concept. With respect to the low level processor, in 1981 they proposed a region one-pass segmentation algorithm. Regions grow by attempting to merge as many adjacent pixels as possible, taking into account colour features. Due to the importance of the segmentation, they proposed a more sophisticated algorithm which deals simultaneously with both edges and regions [28]. The system is based on the design of a rule-based expert system in the form of a modular set of processes and two associative memories. The input image, the segmentation data and the output are stored in a short term memory (STM). A long term memory (LTM) embodies the rules that are based on regions, lines and areas. The region analysis rules merge and split regions by taking into account colour features (R, G, B), adjacencies, sizes and variances, whereas the line analysis rules join, extend and delete lines. The first experiments carried out over a typical house scene demonstrates the reliability of this scheme.

Ohta et al. in 1980 proposed a set of colour features, $I_1=(R+G+B)/3$, $I_2^\prime=(R-B)$, and $I_3^\prime=(2G-R-B)/2$ [29]. The effectiveness of their colour features set was discussed by a comparative study with other sets of colour spaces. In 1985, they developed a region analyser for colour images of urban scenes [30]. The knowledge of the task world is represented by rules. The knowledge associated to the models, which describes properties and relations among objects, is organised as a semantic network. The objects are characterised by colour, texture, position and shape information. The analyser is divided into a preliminary bottom-up process and a top-down process that permits to produce more detailed descriptions. The bottom-up process can be summarised in the next steps:

- The image is segmented by a region splitting algorithm using one-dimensional multi-histograms. The partitioning is performed in the over-segmentation state, and the segmented image is organised into patches.
- The largest patches are selected in order to generate a "plan". These patches are multiple labelled and a
score indicates the degree of correctness associated with each label value.

- In order to evaluate the plan, the rules are activated to examine every region in the plan image. Each rule has a fuzzy predicate which describes a property of a certain object or a relationship between objects.

On the other hand, the bottom-up analysis uses a production system where each production rule is a condition action pair. The algorithm is reliable and can correctly label regions in urban scenes using only 57 rules.

In 1981, Douglass proposed a system named EYE [10], in order to interpret a house scene. It emphasises the integration of depth with semantic information to form a 3D model of a scene. Firstly, images are pre-processed and segmented into a set of regions of approximately uniform colour and texture. The segmentation algorithm is general-purpose and combines the advantages of edge detection and region growing. The scene model is built from the region descriptions in four steps. First, the regions are labelled with an object name as an initial hypothesis. Next, an initial estimate of the regions’ 3D structure is made using a set of depth cues. The initial depth estimates are corrected in an iterative relaxation process, and finally, regions are labelled as belonging to a specific object. In order to label regions, Douglass uses multiple criteria such as occlusion analysis, size evaluations, texture gradients, shadows and highlights analysis, and linear perspective. The EYE system represents object models with an associative net where nodes are objects and the links are logical and spatial relationships among objects. The EYE system goes beyond most other existing image understanding systems that only intend to label regions. Indeed, EYE is the first system that attempts to interpret regions as 3D surfaces and to join the surfaces into a consistent model of the environment.

Celenk carried out a colour analysis study and proposed to operate with the CIE (L*,a*,b*) uniform colour co-ordinate system L*,H*, and C* (Luminance, Hue and Chroma) [6]. In 1994, Celenk proposed a colour scene method for the analysis of object surfaces appearing in the noisy and imperfect images of urban scenes [7]. The proposed system is based on the fact that the uniformly coloured object surfaces are recognised by their colour monomodal distributions, using the colour parameters (L*,H*,C*). The algorithm has been used to recognise and separate regions visually different enough in four images containing uniform or high textured regions, such as the sky, the river or a set of buildings. A scene labelling algorithm was later proposed in [8]. In this approach the segmented image is represented by an adjacency graph description and the scene is modelled by means of 3D to 2D constraints and adjacency relations. The labelling is performed by finding the function which relates the two graphs. In order to do so, the relational distance measure is used to match the relational graphs of the scenes and the respective image.

She and Huang presented a knowledge-based method to classify pixels of colour images representing road scenes [33]. Their method is based on colour and texture, and consists on computing a weighted sum. For each object class there is a database entry as a set of texture and colour descriptions. An object’s texture is characterised by the fractal behaviour of its colour bands (R,G,B). This fractal dimension measure is relatively invariant to changes in the image scale and contrast. An object’s colour is characterised by a chromaticity region in CIE-uv space, in order to reduce the effects of shadows and illumination gradients.

In 1995, Bhanu et al. presented an adaptive image segmentation system which incorporates a genetic algorithm to adapt the segmentation process to changes in image characteristics caused by variable environmental conditions [3]. They consider an unstructured road scene belonging to a natural environment. As a base of their system, they use the Phoenix segmentation algorithm developed at Carnegie-Mellon University. The Phoenix algorithm is a recursive region splitting technique which contains seventeen different control parameters. Based on experimentation, they select the two most critical parameters that affect the overall results of the segmentation process. The idea of their approach is to infer these parameters automatically. Once the image statistics and external variables (time of the day and weather) have been obtained, a genetic learning component selects an initial set of segmentation algorithm parameters. The system needs a module to carry out the task of evaluating the segmentation results. Bhanu et al. proposed five different quality measures to determine the overall fitness for a particular parameter set. The algorithm was tested and provided high quality segmentation results.

Mecocci et al. presented a system capable of giving a simple outdoor and indoor scene description in order to provide help for blind people navigation [27]. The system carries out a preliminary scene components classification into two classes: the man-made object class characterised by planar surfaces and straight lines; and the fractal class, representing the objects, like natural ones, without simple planar surfaces. The
two main inputs of the interpretation are regions and segments. The regions are obtained by a region growing algorithm which acts in two steps and takes into account grey levels, edges, mean values, and standard deviations. In order to carry out the interpretation, regions are then classified in accordance to their related segments. In this way, the system recognises fractal regions, lateral vertical surfaces, frontal vertical surfaces, horizontal surfaces and unrecognised regions. For example, one horizontal surface normally appears in the image as a region with many horizontal and vanishing point segments. In other cases, there are many oblique segments in a region that do not converge to any vanishing point, the related scene entity is classified as fractal. The interpretation ends with constraint region criteria and with criteria for propagating adjacent regions.

Lasserre [22] presented a GPVS approach for the identification of the objects nature in natural scenes. Firstly, the input colour image is segmented by a method combining both region growing and clustering (based on general histograms shapes of R, G and B) techniques. Then, regions are characterised by their colour (R,G,B) and texture edge parameters, and afterwards identified. Finally, a probabilistic method based on bayesian classification, is used to determine the nature of the elements present in the environment. This final stage can be divided into three phases. The first one consists in building the knowledge database. This phase, named learning phase, permits to define a model for each object class. The second phase, named evaluation phase, consists in assigning to each region a probability of belonging to one class. The third phase consists in merging neighbour regions which are labelled with the same class and in verifying each labelled region using a rule method, which takes into account spatial relations.

3.2 GPVS based on objects

This section describes vision systems characterised by the following procedure: the input image is segmented by using non-purposive techniques; the objects are identified by specialised procedures which search for a set of selected features in regions; each object is characterised by its own features; a final step consists on gathering the information provided by the algorithms related to the different classes of objects in the scene, so as to obtain a globally consistent description of the scene.

From the mid seventies, the VISIONS group has been evolving a GPVS for understanding images of road and house scenes [11,16,17]. The VISIONS Schema System provides a framework for building a general interpretation system as a distributed network of many small special-purpose interpretation systems. Each schema is an "expert" at recognising one type of object. The system's initial expectations about the world are represented by one or more "seed" schema instances which are active at the beginning of an interpretation. As these instances predict the existence of other objects, they invoke the associated schemas, which in turn may invoke still more schemas. Schema instances run as independent concurrent processes, communicating asynchronously through a global blackboard. The goal of a schema is to collect the evidence necessary to post a hypothesis of confidence level or belief or higher.

Concerning natural object recognition, the VISIONS group emphasises the following points:

- Object recognition requires flexible matching on a variety of characteristics, with the ability to handle exceptions and deviations.
- Not all object classes are defined in terms of the same attributes, and these attributes may be used in various ways within the matching or interpretation process.
- A close-up view of an image of a typical urban setting, illustrates the utility of attempting to recognise objects from data obtained at the pixel level.

In 1996, the VISIONS group reviewed with a critical eye the core issues of knowledge-based vision systems [12]. They argue that low- and mid-level vision procedures that were utilised to perform the basic tasks of vision, were immature at the time to support the ambitious interpretation goals of these systems.

From the late eighties, Asada, Shirai et al. developed a GPVS that interprets colour images representing urban scenes using 3-D information [1,20]. First, using a general purpose algorithm the image is divided in regions having uniform brightness and colour. Second, the interpretation process is based on the search for specific objects in a specific order. For instance, the interpretation system starts from extracting road, sky and trees, independently. The characterisation of natural objects includes colour distributions (Brightness, Hue and Saturation), position, orientation, size and relations among the objects. Each object model is represented as a frame and the object models are related to each other by a network of frame structures. In 1994 they presented a new approach in which 3D information was inferred from a stereo vision method [37].
Strat and Fischler proposed a method, called CONDOR, to recognise objects in natural scenes [34,35,36]. The central idea of their architecture is a special-purpose recognising method designed for each object-class. Despite the fact that their architecture looks for specific objects, its goal consists on understanding of the whole scene using non-purposive segmentation methods. In CONDOR the interpretation of an image involves the following four process types:

- Candidate generation: Condor associates a collection of simple procedures to each object model. Each procedure is competent only in some restricted contexts, but collectively, these procedures offer the potential to recognise a feature in a wide range of contexts. The output of each procedure is treated as an hypothesis.
- Candidate comparison: Condor makes use of multiple evaluators to evaluate each candidate for a given class, and if all such evaluators prefer one candidate over another, a preference ordering is established between them.
- Clique formation: A search for mutually coherent sets of candidates is conducted by building incrementally cliques of consistent candidates, beginning with empty cliques.
- Clique selection: One clique is selected as the best semantic labelling of the image on the basis of a portion of the image that is explained and the reliability of the operators that contributed to the clique.

The idea of Strat et al. was that each of these processes acts like a demon, watching over the knowledge base and invoking itself when its contextual requirements are satisfied. The knowledge is embedded in rules as conditions actions pairs, named “context sets”. Context sets are employed in three types of rules: candidate generation, candidate evaluation and consistency determination. The addition of a candidate to a clique may provide a context that could trigger a previously unsatisfied context set. For example, once one bush has been recognised, it is a good idea to look specifically for similar bushes in the image. The considered natural object features include colour (R,G,B), texture, geometric shapes and spatial relations among objects.

3.3 SPVS

This section describes two kind of SPVS. On the one hand, the approaches of Campani, Parodi et al. and Hild and Shirai, which goal is to understand the whole scene. On the other hand, the approaches of Graefe et al. and Ide et al., which are only interested in recognising specific objects. The difference between GPVS based on objects and SPVS, is its initial step. SPVS segment the input image by using purposive methods. Then some specific procedures are used to look for specific objects, and finally the results provided by the procedures are merged to check a coherent description of the scene.

Graefe et al. proposed an object-oriented approach for detecting and classifying objects, that could be obstacles for a mobile robot operating in an outdoor environment [13,32]. Their approach is based on the principles of the monocular dynamic vision for sensing the environment [15]. Their goal is to reach tangible results in reasonable time, as required by mobile robots operating on roads. They decompose the global task into several independent sub-tasks, or modules, including the initial detection of possible candidates, the classification of candidates into false alarms and real physical objects. The key of their approach, in addition to the use of a image sequences rather than a single one, is to apply specialised segmentation and recognition methods in order to find specific objects. The scheme used to recognise objects is roughly based on the following: Firstly, limiting the search area. Secondly, detecting objects in one image. The detected objects are considered candidates. Thirdly, tracking the candidates in order to validate them. Although neither colour nor texture features are used in their approach, they achieve fantastic results working in real time.

In 1992, Ide et al. presented a method for measuring 3D positions of target objects using stereo vision [21]. In particular, using trunks or poles in urban scenes as targets, they attempted to automate recognition and stereo correspondence process. Their method follows the SPVS paradigm: it uses knowledge in order to limit the search area, to obtain candidates (pairs of vertical edge sequences) and finally, to validate them (using specific characteristics of the poles such as diameter and height).

In 1993, Hild and Shirai proposed a two-stage interpretation SPVS applicable to hard natural scenes [19]. The first stage carries out specific object classification for each image pixel on the basis of known default models for local feature. It then searches purposively for regions which satisfy qualitative constraints in the classified object images and selects object region candidates, using multiple features (hue, shape, orientation, position...). The second stage extends and refines the result of the first stage by exploiting knowledge about the object.

Parodi, Campani et al. proposed a method for the interpretation of traffic scenes [5,31]. The presented
method is feature-based in the sense that the objects are identified and localised by exploiting some of their characterising features and, only later, their shape is reconstructed more thoroughly. First, the general structure of the 3D scene is recovered exploiting a priori scene information and edge analysis. As a result, the image is partitioned into four regions: roadbed, sky and two lateral regions. Second, the objects are identified by specialised procedures which search for a set of selected features in determinate areas. Basically, natural objects are characterised by their supposed spatial disposition in the scene, their colour and edge parameters. A final step consists in gathering the information provided by the algorithms related to the different classes of objects in the scene, so as to obtain a globally consistent description of the scene in the form of a simple line drawing.

4 Conclusions

This work reviews knowledge-based vision systems intended to recognise natural objects in outdoor scenes. A classification of the analysed approaches is given according to whether the segmentation is general purpose or special purpose and whether the recognition process is based on labelling regions or recognising objects by specific procedures. Related to the proposed classification, several vision systems have been described. Advantages and disadvantages of GPVS with respect to SPVS have also been discussed. Moreover, the paper emphasises on the segmentation proposals due to its importance in further scene interpretation.

Concerning natural object recognition, it is remarkable how some approaches, especially GPVS based on objects and SPVS, perform flexible matching on a variety of characteristics; how they use different attributes to model different objects; or how they take into account the variability in the measured chromatic values due to outdoor lighting conditions. Anyway, adaptive models that change their attributes automatically depending on the outdoor conditions and seasons, are missing. For instance, it seems interesting to select different colour space transformations in function of the time of the day and the weather conditions; an adaptive model seems interesting for objects such as trees because they change heavily their appearance due to seasons and other phenomena.

Developing a vision system with the aim to understand the whole scene, where all the levels of vision are specific purpose oriented (like an SPVS), is an important research direction. Although such systems report interesting results, there is still a long-term work to build a vision system with similar performances as the human eye and visual perception system.

References

An Active Vision Approach for Autonomous Robot Navigation

Aleix Martínez Margarit  
Computer Vision Center  
and  
Robot Vision Lab  
Purdue University  
aleix@ecn.purdue.edu

Jaume Aguirre  
U.A.B.  
Dept. D’Informàtica

Jordi Vitriá  
Computer Vision Center  
U.A.B. Edifici O  
08193 Bellaterra  
jordi@cvc.uab.es

Abstract

The ever increasing need to build more intelligent robots, have prompted scientists to search for new areas of research which can aid in the building of better (or, let us say, more intelligent) robots. In this article we present a novel approach to the design of active vision systems which can help our robot to better interact and behave in uncertainty environments (such as an office building). The system is based on visual scanning [Ya67]. The idea of the whole system is inspired in ecological vision [Gi86] which helps us to build simple mechanisms able to interact with our world (real environment) and overcome very difficult tasks such as: (i) room recognition, (ii) localization and identification of specific landmark, (iii) recognition of people, (iv) navigation, etc. In this paper the whole system and the applications to room recognition (where results obtained where up to 95%) and landmarks localization are presented. This implementation shows how complex information processing tasks can be solved by using active visual behavior based on simple mechanisms.

Key Words: Active Vision, Ecological vision, Behavior based, Autonomous Agents, Learning.

1 Introduction

Many of the actions of an agent are quite separable, coherent intelligence can emerge from subcomponents interacting in the world [Br91]. That is, the intelligence of an agent is not determined by the complexity of a unique behavior algorithm but by the combination of several simple behavior algorithms. Actually, this is not a new statement, the concept of organized complexity became the very subject of the systems approach in the early 20s. At that time, the philosopher C.D.Broad coined the term emergent properties for those properties that emerge at a certain level of complexity but do not exist at lower levels [Ch81]. This new way of thinking went on for quite a long time and helped to give birth to a new way of thinking: systems thinking. According to the systems view, the essential properties of an organism (or living systems) are properties of the whole, which none of the parts have. They arise from the interactions and relationships among parts. These properties are destroyed when the system is dissected, either physically or theoretically into isolated parts (as a recommended bibliography see [Ha76]). Other scientific disciplines have also contributed with interesting new ideas to the whole new way of thinking. In quantum physics, for example, these relationships are expressed in terms of probabilities, and the probabilities are determined by the dynamics of the whole system (i.e. whereas in classical physics the properties of the parts determine those of the whole, in quantum physics the whole that determines the properties of the parts) [He71]. All these new disciplines end up in the creation of a new scientific understanding of life called ecology (known also as systems thinking, or gestalt). In ecology we define three kinds of living systems: organisms, parts of organisms, and communities of organisms; all of which are integrated wholes whose essential properties arise from the interaction and interdependence of their parts. In nature, evolution has created a great diversity of species with such features, where the whole system resembles a large, multi-creadured organism (bees and ants are good examples of this).

In artificial intelligence many scientists have adopted also such a philosophy of thinking. Since
Brooks published his first papers on autonomous agents and subsumption architecture in mid-eighties, a significant progress has been made. Rewriting the previous ideas (of different ecological sciences), scientists have started a new discipline (behavior based) where agents behave in a sophisticated way by using a combination of some simple tasks, rather than modeling complex behavior in a unique very complex program. Intelligence emerges from the system, we do not really build it. Some good examples of this philosophy can be found in [Br89] and [Be90].

However, during these last few years little progress has arisen in this area. It seems difficult to find simple tasks which can easily be combined in a synergetic manner to produce a complex behavior. This is the reason why many researchers have been looking for new research areas, where new, simple behavior tasks might help the whole system to yield better results.

In these last years we have also been studying different possibilities for incorporating new knowledge and behavior capabilities in our agents [MV95] [MV96] [MVL97]. To that end, we have studied visual systems, and based on visual scanning criteria [Ya67], a visual mechanism has been built. The main ideas of our system are based in a ecological vision. One of the main topics of ecological vision, is that assume that agents (robots) cannot be understood as a unique entity, but rather as a system that is integrated in a whole, its environment [Gi86]. In other words, vision cannot be understood as a single system, but as a part of the whole. In computer vision some scientists have also started to work with some of those ideas. Along these lines it is worthwhile to note the contributions of Aloimonos [AWB88], Ballard [Ba91] and Tsotsos [TC98]

In this article we present a new approach to the design of autonomous robots which make use of a behavior based control system and ecological vision systems which help the whole to yield better results. That will allow us to build better robots or, let us say, more intelligent robots.

Another important point to be mentioned here, is the low cost of our system. The use of ecological techniques has allow us to build system which not only are able to act in real time, but also to do it by using low-cost hardware (based on PCs and cheap cameras).

In the following sections we will present the visual scanning method we have implemented and some of the applications we are working on. We are also studying other possible applications such as: the location of other agents, location and identification of people [MV97], localization of specific objects, etc.

This paper is organized as follows. In Section 2 we introduce the environment where the robot lives, Nat (our small mobil robot), and the system capable of interacting in this environment. Section 3 presents the experiments we have performed to test the accuracy of the method. Finally, in Section 4 we discuss the method and the experimental results.

2 Recognizing the Environment

2.1 The environment

Nat is a robot that goes around in the first floor of our building (figure 1a), that is, a total of 20 different offices. It is supposed to navigate around this floor and being able to recognize the place were it is, when necessary. As soon as Nat localize an open door it enters to the room and tries to recognize it. No-one can grab the robot and place it into another place of the room, even though small displacements may not affect the recognition algorithm. However, due to possible slippage in the wheels or any other environment distortion, the orientation of the robot, when it comes into the office, is not known. An scheme of that can be appreciated in figure 1b.

Another characteristic of our offices is that they might have a special landmark which indicate that the robot must go to that place to do some specific task. Those landmarks are positioned by people that want the robot to do some administrative task in that specific office.

2.2 The robot and its applications

To test our ideas and the system developed, we build a small robot (figure 4a) capable of moving around the first floor of our building (figure 1a).

The robot is supposed to navigate around this floor and help people with some administrative tasks. To do so, it is capable to localize open doors, recognize the office it is in, and (if necessary) localize a specific landmark (see figure 6) that people from the lab can label when they need help. When the robot localize such a landmark in one offices it moves forwards towards it until it is
correctly positioned (in front of the landmark). At this point the person who was requiring its help can tell it what is his wish. Of course, the robot cannot yet understand what that person says (and just runs away), but in the futures it is suppose to understand few commands which might be of some help for the people of the lab (or any other company).

2.3 Visual Scanning

The developed system can be divided into the following four parts: (i) focus of attention, (ii) selection of a fixation point, (iii) foveal information extraction, and (iv) pattern matching (e.g., compare the information extracted from the new image with some stored DB of rooms).

Focus of attention: With modest hardware requirements, it would be impossible to build a real-time system which analyzes an entire image (that is to say, the whole visual area). In order to decrease the complexity and computational cost of the algorithms, an attentional mechanism is used [HuB91] [Ba91]. The attentional mechanism should select which visual areas are important enough to examine closely and which are not. That means, it must decide where the relevant information is in the raw image. To do this, three attentional mechanisms are used; one which detects symmetrical points in the image (figure 2a), and two others which detect the most important L and X unions of the image (figure 2b-c). There is some biological evidence that suggests that these three operators play an important role in the human pre-attentive visual system [MRW95].

The symmetry operator we have chosen [RWY95] is an operator capable of finding the most important symmetrical points in an image independently of its orientation. The operator’s description is as follows:

\[
C_S(i, j) = D_s(i, j) P(i, j)r_i r_j, \tag{1}
\]

where, \(D_s = \frac{\exp(-\frac{D}{\tau})}{r \exp(-\frac{D}{\tau})}, \quad P(i, j) = [1 - \cos(\theta_i + \theta_j - 2\alpha_{ij})] \times [1 - \cos(\theta_i - \theta_j)], \) and, where \(\theta_i\) and \(\theta_j\) are the angles of the gradient in the i and j points of the image, and \(\alpha_{ij}\) is the normal angle of the line that goes from point i to point j.

The union detectors use the scale-space operators described by:

\[
C_L(x, y) = \frac{2L_{\partial L_y L_{xy}} - (L_x^2 L_{xx} + L_x L_{xy}) L_y^2}{L_x^2 + L_y^2}, \tag{2}
\]

and,

\[
C_X(x, y) = -(L_{xxx} L_{yy} - 3 L_{xxy} L_{xyy})^3 + 27(L_{xxx} (L_{xxy} L_{yy} - L_{xyy} L_{xyy}) - L_{xx} L_{xyy} L_{xy} + L_{yy} (L_{xxy} (L_{xxx} L_{xyy} - L_{xyy} L_{xy}))^2, \tag{3}
\]

where \(L_x\) is the first derivative of the image in the x direction, \(L_y\) the first derivative in the y direction, \(L_{xx}\) the second derivative in the x direction, and so on.

Selection of a new fixation point: The first point to be selected will be the one which has the highest symmetry value (we found experimentally that the symmetry operator was more important, i.e. more useful for the first steps of the algorithm than the union operators were). For the rest of the symmetry points, a visual scanning [Ya67] strategy based on motivational autonomy [MB93] is proposed. For that purpose, each attentional point is labeled as follows: the symmetry points are weighted with twice their value, the X-unions with their own value, and the L-unions with half of their value (that weight is referred to as motivation and is denoted by \(m_{ot}\)). Now, the probability for going from each point to the
next will depend on this number plus a proximity function. The next point to look at (to fixate on) is the one that maximizes the formula:

$$S(x_{\text{new}}, y_{\text{new}}) = \frac{1}{(||x_{\text{old}} - x_{\text{new}}|| + 1)} \times \frac{1}{(||y_{\text{old}} - y_{\text{new}}|| + 1)}$$

where, old means the point we are looking at currently, and new the point we might go to; k is a constant. Notice that the nearest points are given much more importance than those that are further away. Each time a point is selected, its motivational value is decreased to 10% of its previous value. An example of this visual scanning strategy can be appreciated in figure 2d.

Foveal information: It is obvious that some information must be extracted from each fixation point in order to compare that information with some stored database of rooms. As biological systems do [Ca88], we use a foveal mechanism for environment information extraction. On our system, this implies that only nine points are used (figure 3a-b). Since the system is foveated, the central point extracts more information than the peripheral ones, i.e. it uses more scales (see figure 3a).

In each of these points, a high dimensional vector is extracted. For this purpose, an iconic representation [RB95] is used. The first, second, third and fourth derivatives of the gaussian filter (a well-known basis filter for natural images1 [HaBS92]) at different scales and orientations are used. We also use four scales around the central point, three around each of the peripheral points, and we use k+1 orientations at k-order derivatives (as a well-known base of steerable filters [FA91]). Thus, a 504-dimensional vector is built. In a previous work, we have demonstrated that these vectors have high classification capabilities even when changes in the target or in the illumination conditions are presented [MV97].

Pattern recognition: The obtained vectors in the previous step are too complex to work with, and although they are highly discriminant vectors, the processes (algorithms) needed for pattern matching with such a vectors would be computationally intractable in real time and modest hardware requirements. Fortunately, we have tools which can find where the relevant descriptive information is (inside of those vectors, e.g. which are the dimensions with highest variance). One of the most used techniques in pattern matching, and perhaps the most accepted into the computer vision community, is the Principal Components Analysis (PCA) [Fu90]. Applying such a simple technique we can obtain much smaller vectors which correctly represents the information obtained before. In such a way we solve two basic problems: (i) the amount of memory used is reduced drastically, and, even more important, (ii) the computational cost of matching vectors decrease enormously.

To do that, we first store all images in a vector by reading pixel brightness values in a raster scan manner, $\mathbf{x} = \{x_{1,1}, \ldots, x_{m,n}\}$ (where m is the number of rows and n the columns). After that, the vectors are normalized to have total energy equal to unity (i.e. $||\mathbf{x}|| = 1$). Then, the average of all images in the database is subtracted from each image. This ensures that the eigenvector associated with the largest eigenvalue represents the dimension in eigenspace in which the variance of images is maximum in the correlation sense. In other words, it is the most important dimension of the eigenspace. Let us call to this new vector

1Hancock et al. extract the first few principal components (which can be seen as eigenvectors) of a neural network built up from natural images and found that regardless of the scale of analysis, these eigenvectors were very close approximations of different orientation derivatives of gaussian operators.
\( X = \{x_{1,1} - c_{1,1}, \ldots, x_{m,n} - c_{m,n}\} \). The projecting matrix (from the high-dimensional space to the low-dimensional one) is obtained from the \( p \) most important eigenvectors (i.e., the \( p \) eigenvectors associated with the \( p \) largest eigenvalues) determined by the well-known eigenstructure decomposition problem:

\[
\lambda_i e_i = Q e_i
\]  

(4)

where, \( Q \) is the covariance matrix (i.e. \( Q = XX^T \)), \( e_i \) the \( i \)-th eigenvector, and \( \lambda_i \) the \( i \)-th eigenvalue.

Thus, the projecting matrix will be mathematically expressed as:

\[
[e_1, e_2, \ldots, e_p]
\]  

(5)

Learning (database) and identification:

First of all, a database of rooms has to be built. In order to do that, the twenty most important attentional points of each room are chosen using the criteria described above. Then, the iconic representation of each room is calculated and projected into the new low-dimensional space (by applying the PCA algorithm).

Then, the ten vectors of each room that are the most discriminating are selected for the database; that is, those vectors which are best separated from the vectors that describe the other rooms. For this purpose, the between-class measure of Discriminant Analysis [Fi83] is used. Mathematically speaking we can define the between-class scatter matrix as:

\[
S_b = \sum_{i=1}^{c} (M_i - M)(M_i - M)^T
\]  

(6)

where, \( M_i \) is the mean feature vector of class \( i \), and \( M \) the mean feature vector of all classes.

The most discriminating vectors will be those which maximize the between-class scatter matrix, \( S_b \).

Finally, we add all these previous points together to come up with the final algorithms. The first algorithm describes the learning process.

**Algorithm I:** Learning algorithm.

- 0. Take the first room, \( i = 1 \), of all \( n \).

- 1. Take the 20 most important attentional points of the \( i \)-th room \( i \) (the ones which have highest \( mot \) value).

- 2. If there are more rooms, \( i < n \), go to 1.

- 3. Obtain the \( S_b \) for all possible combinations of 10 points per room. That implies putting 10 of the 20 points into the database and leaving the remaining 10 out (without using them).

- 4. Select the combination that maximizes \( S_b \).

- 5. When finish, calculate the PCA of the whole database of rooms and store the projection matrix (5) in memory.

Notice the high computational cost of the third step of this algorithm. It is an exponential problem (precisely, it is of the order \( \left(\frac{q}{p(q-p)}\right)^n \), where \( q \) is the number of different attentional points extracted from the image, \( p \) the number of points added to the database, and \( n \) the number of different rooms). However, if the number of rooms is not very high and as learning should only take place once in a while, the computational cost of this step can be ignored.

Unfortunately, if a large number of rooms are to be used, this point becomes critical and should be studied further. A possible solution to this problem, could be to first learn the \( m \) most important rooms (or any random \( m \) rooms, if any is more important than the others) where \( m \) is small (normally \( m << n \), and then, build an iterative (i.e. recursive) process for the remaining \( n - m \) rooms, where the computational cost will only be of the order \( (n-m) \frac{q^m}{p(q-p)^m} \). And so, the final cost will be of the order \( \left(\frac{q}{p(q-p)}\right)^m (n-m) \frac{q^m}{p(q-p)^m} \).

The second step consists of identifying new rooms.

**Algorithm II:** Identification algorithm.

- 0. For each room \( R_k \) initialize the evidence array \( E(R_k) \) to 0. Then, choose the most important attentional point of the current image.

- 1. Move the focus there and extract the iconic representation of the point.

- 2. Project the new obtained vector into the PCA-space (use the projecting matrix, formula (5), stored before to do this step).

\textsuperscript{2}The evidence array is a voting space, where each position of the array represents a known room.
3. Match the information with all the points stored in memory (by using either the $L_2$ norm or any other dissimilarity technique).

4. Add one to the $E(R_e)$ value for all rooms $R_e$ which have an iconic vector similar to the one we extracted from the image (this implies that a threshold will be applied to the distance calculated previously).

5. If any model $R_w$ has the $E(R_w) > T$ (where $T$ is a threshold value) then the model recognized is $R_w$.

6. Otherwise, select the next attentional point and go to 1.

2.4 Landmark recognition

Once the robot has entered to a new room, some additional tasks might be necessary. To achieve that, Nat (our robot), is able to determine if a demanding landmark (specifying that some task is required) is present or not. To achieve that an algorithm to localize specific, easily-detected landmarks is proposed. The landmark used is shown in figure 6. The use of this specific landmark is justified because it can be easily detected by using simple and quick algorithms of computer vision [Mu93].

To correctly localize those landmarks Nat makes a look around all the offices it is in (starting from one side to the offices and rotating until it reach to see the other side). Every 10 degrees it stops and grabs an image. In each of them, Nat calculates the four more important corners (L-unions) and applies an affine transformation to see if those points are equi-distant and forming an ellipse. If that is the case, the landmark has been localized. Otherwise, we conclude that the landmark is not present (see figures 6a-b).

Once the landmark has been localized, we move the robot forwards until it reaches the final position (i.e. when Nat is in front of the landmark). To achieve that, we calculate the deformation of the previous ellipse to determine the orientation of the landmark. This orientation will determine the next direction that Nat has to do. The position of the landmark in the current grabbed images will also inform us either if Nat has to move to the right or to the left and how many degrees. The quantity of degrees is calculated qualitatively but no quantitatively (that would make no sense to us). The results of Nat approaching to one of those landmarks can be appreciated in figures 6c-f. Figure 6g describes the path that Nat made in this previous example.

3 Nat: a mobil robot

To test the system described above, we developed an autonomous robot: Nat (see figure 4a). Nat is a wheel robot equipped with a QuickCam, a small camera. Both the robot and the camera, are connected to a Pentium PC through the serial port and the parallel port respectively. The robot has two stepper motors which allow it to move forwards, backwards, right and left as it wishes. The camera provides images of 256 by 256 pixels with 6 bits of depth.

The robot lives on the first floor of our building (figure 1a) and, in the future, it is supposed to be able to help people to do some administrative tasks. Currently, it is only capable of moving around (either wandering or making use of a map of the building), avoiding obstacles, detecting open doors, and recognizing offices. Its architecture has been developed using the subsumption architecture [Br86] and is briefly described in figure 4b.

3.1 Experiment 1:

First, we acquired a unique image from each room and made Nat learn them using the Algorithm 1 described above. These images were frontal images, which meant that the robot was at orientation zero (see figure 1b). The next day we grabbed a new frontal image from each room and tested whether Nat was able to recognize them or not. The success rate was 100%. Notice that some
small pieces of furniture in the rooms may have been moved during the previous day, and that the illumination conditions on both days might not be the same. Even so, Nat successfully recognized these frontal images.

3.2 Experiment II:

The second experiment consisted of testing the on-board visual system. For that purpose, we ran a program which drove Nat to a randomly chosen room. As soon as the robot entered the room, the visual system tried to recognize it. As Nat cannot know its exact orientation when it comes into a new room - due to differential slippage in the wheels or other environment distortions - a third algorithm was proposed. The underlying idea consisted of acquiring different images from different view-points until the robot could successfully recognize the room (as we human beings do). That is, if the image that the robot has acquired is not similar to any room that it has in its database, it will not make a match. Thus, if after a certain number of fixations (on each image), the robot could not identify the room, it moves a few degrees (either to the right or to the left, as necessary) and starts the process again.

Algorithm III: Active identification of rooms.

1. For each room $R_k$ initialise the evidence array $E(R_k)$ to 0.
2. In the current position grab an image, and choose the most important attentional point.
3. Move the fovea there and extract the iconic representation of this point.
4. Match the information with all the points stored in memory (by using either the $L_2$ norm or any other dissimilarity technique).
5. Add one to the $E(R_k)$ value for all the rooms $R_i$ which have an iconic vector similar to the one we extracted from the image.
6. If any model $R_w$ has the $E(R_w) > T$ (where $T$ is a threshold value) then the model recognized is $R_w$.

3.3 Experiment III:

A total of 20 tests were done to check either the algorithm of landmark localization and approximation works well or not. In all our experimtens the successful results was 100%. As an example see figure 6. Notice, the landmark is sometimes near the limits of the image. That is because a qualitative algorithm is used instead of a quantitative one. It is an easy algorithm which can both be more easily calculated through many different cues, and it has lower computational cost.
Figure 5: (a) An example of the different images used for Nat until it could recognize the room. At the beginning of the sequence Nat could not identify anything; however, as soon as it was turning around, it started to recognize the room. Finally, the identification result was positive. (b) Another example of a successful identification. In this sequence two images were enough.

Figure 6: (a-b) searching for a landmark. In (a) no landmark is localized. Whereas, in (b) a landmark is correctly localized. (c-f) Moving towards the landmark. (f) Landmark reached. (g) The path that the robot made to go from the door to the landmark.
4 Conclusions

We have shown that visual scanning can help to build real-time platforms which allow an agent to achieve complex visual tasks like environment recognition with a high degree of accuracy. When implementing the attentional mechanism, we tested three different operators and found that symmetries play an important role in the recognition task.

Finally, we have shown that pre-attentive scanning, i.e. without context or high-level information, can yield high success rates in recognition tasks. That last point is very important, since it means we can use the same algorithm for different recognition tasks (such as a face recognition algorithm based on similar ideas which we developed [MV97]). Another important consequence is that although high-level processes might lead to better results, there is no need to use them in order to build agents which are able to interact with their environment. This is consistent with the ideas of Brooks [Br91] concerning artificial intelligence. A reasonable characterisation of the field is that it is intended to make computers do things, that when done by people, are described as having indicated intelligence.

Further details of this project can be found in [Ma88]

Acknowledgements

This research was supported in part by grant TAP94-0401 from the CICYT. We are in debt with Sergio Gonzalez for his early work on the project and with Craig von Land for his help when writing this document. The authors also wish to thank Professor Avi Kak for their help, comments and support during the realization of this project.

References


[MV95] A. Martínez and J. Vitrià. Designing and Implementing Real Walking Agents using Virtual Environments. EPIA'95 - Seventh Portuguese Conference on Artificial Intelligence, Madeira (95).


Visual ELIZA: a testbed for developing visual perception processes for conversational interfaces.

Jordi Vitrià, Joan Ramon Serra, Enric Raso, Agustí Puertas
Centre de Visió per Computador
Edifici O, Universitat Autònoma de Barcelona
08193 Bellaterra, Barcelona
jordi@cvc.uab.es

Abstract
In this paper, we present a new implementation of ELIZA, the AI-classical program that simulates a Rogerian psychotherapist. The result is an augmented version of the program called Visual ELIZA. In addition to the “keywords and frames” approach that were used in the old version to control the conversation between the user and the program, Visual ELIZA can use visual data to behave more realistically. We think that this approach can be very useful for developing, testing and comparing visual processes for conversational interfaces while maintaining a relative simplicity of the application.

Key words: intelligent multimodal interfaces, visual perception, face detection, face recognition, face tracking, conversational interfaces.

1. Introduction
It seems quite likely that intelligent conversational interfaces will be present among computer applications in the next years [7,8,10,12]. Many of the involved technologies, including speech processing and natural language understanding are mature enough to think that the goal of constructing a conversational interface for complex applications has become a realistic challenge. Visual perception plays an important role in some of the processes that are involved in communication [9,11]. Some work has been developed for classifying and recognizing facial expressions and gestures from video, but little has been done on integrating and studying the dynamics of these tasks in real interfaces. Visual ELIZA is a prototype system that allows the integration of visual and graphical technologies into conversational interfaces while maintaining global application simplicity. The basic idea is to increase the “key words” repertory of ELIZA [1] by introducing “key visual percepts” that can control textual as well as graphical outputs during human-computer dialogue.

Our work has focused on defining the basic building blocks of Visual ELIZA, as well as to consider the role that faces can play in a conversational interface. Faces are accessible “windows” into the mechanisms that govern our social and emotional lives [2]. Visual technology is now in hand to develop systems that are capable of tracking and recognizing faces in real time, but more advanced developments are needed in order to use more subtle information. In this paper we firstly present an overview of facial image analysis and summarize those challenges that can be crucial for developing useful conversational interfaces. Next, the architecture of Visual ELIZA is exposed. At last, we outline our research program in this field.

2. Visual perception for communication
Faces can provide information about affective state (emotions, moods), cognitive activity (perplexity, concentration, boredom), temperament and personality, truthfulness, and psychopathology [2]. This information is conveyed via four general classes of sign vehicles: static facial signals, slow facial signals, artificial signals and rapid facial signals. All four classes contribute to facial recognition, being the last the most important for conversational interactions. Rapid changes in facial muscular activity are brief, lasting a few seconds.

There is a long history of research into computer face recognition and interpretation. Much of the work has focused on detecting individual features (eyes, mouth...) and defining a face model based on the relative position of these features. Such approaches have proven to be difficult to extend to general imaging conditions. Recent approaches to face identification have derived to develop methods (using neural networks or the eigenface approach [6]) that seek to capture the configurational nature of the task. These more global methods have proven more successful and robust, but several problems remain not completely solved:
a) Finding faces and facial features in complex backgrounds. Detection and location of faces encounter two major problems: pose and scale.

b) Recognizing and interpreting facial expressions. Most work has involved frontal images of static faces under good illumination. No robust methods exist for classifying facial expressions and our comprehension of the role that these expressions can play in communication is limited.

c) Tracking faces and temporal interpretation. Facial movements are often more rapid than our (physical and artificial) visual systems. Hence robust and specialized tracking strategies and temporal integration models are needed.

d) Detection of additional information and events: detecting when a person is speaking and distinguishing those lips movements required for speech articulation from additional movements of the lips which are signs of emotion; detecting position changes; detecting brow movement; detecting smiles; recognizing age and sex; etc.

In spite of the fact that recognizing and interpreting facial expressions may seem the most important point for developing conversational interfaces, we think that this is not the whole truth if we consider the technologic state of the art. Even in the case that this point would be solved in laboratory conditions, other (in theory) “minor” factors can make useless a real world conversational interface. The best intelligent interface can be useless when it does not know if its user is speaking to it or to a telephone!

Visual ELIZA is an augmented version of ELIZA [1], the program developed by Joseph Weizenbaum in 1966 to study natural language communication. In addition to the “keywords and frames” approach that were used in ELIZA to control the conversation between the user and the program, Visual ELIZA will use visual data to behave more realistically. The conversational nature of ELIZA remains at the same level in this new implementation. This fact enables a rich communication environment where visual information can be easily integrated and tested.

Our approach for developing visual capabilities for Visual ELIZA is bottom up. We will start by augmenting its perceptual intelligence by incorporating reactive and low-level informative processes, like face detection/tracking, and face recognition.

3. Initial system

In its initial development stage, Visual ELIZA has resulted in an augmented version of ELIZA that presents an anthropomorphic graphic interface [3,5]. The interface character is capable of directing its sight towards the user and reacts to his/her movements by using saccadic as well as smooth sight movements. This simple behavior (which clearly increases the effectiveness of ELIZA with inexperienced users) is based on two basic visual processes: face detection and motion perception. The first uses color information as well as an eigenface approach to face localization. The second detects temporal changes in image luminance and determines in which area of the character (virtual) visual field there is some activity. The interface character has mobile eyes that can rotate when indicated by the visual processes. The face detection process is in charge of providing an attention area where to look (that can suffer from smooth location changes), and the motion detector is a “disturbing” process that can redirect attention to high activity areas during short time periods. This visual-guided behavior is executed independently during the conversation between the program and users.

3.1 Software architecture

Visual ELIZA has been designed as a multiprocess application. Three processes run simultaneously: a LISP interpreter loaded with a LISP version of ELIZA; a graphical user interface; and a visual process. These processes do a synchronous communication to provide just in time responses. The visual process retrieves images from a color camera. Some facial characteristics are extracted to take them in account when the communication with the ELIZA process is achieved.

3.2 Hardware environment

The environment runs on a 300MHz Pentium PC under Windows NT, with a color camera. The system is coded in C++ and LISP.

4. Current development directions

The next steps towards a complete Visual ELIZA will be the following:

1. Incorporating a face recognition module for “personalized” conversations. This will be the first “non-reactive” visual process, and its information
will be stored and modeled as "key visual percepts" by the ELIZA process.

2. Developing ages and sex recognition strategies. Age and sex can be used in many ways in the ELIZA framework for making its behavior more realistic.

3. Studying important facial events' recognition, like turn-taking protocols or "busy flags" (do not talk to me when I am at the telephone).

![Figure 1: Visual ELIZA uses an anthropomorphic graphical user interface that tracks user face and reacts to his/her movements.](image)

4. Conclusions

Visual perception will play an important role in the near future interfaces, but some basic problems remain to be solved before its generalized use. We have centered these problems in our "emotional and social window": the face, although we can not forget other expressive cues that we use (i.e. gestures) [4]. Some basic visual processes must be more deeply studied in order to be able to build reliable "face analyzers for conversational interfaces", and this study will be more accurate if it is made in context. For this reason we have developed a simple conversational environment, based on the AI-classic ELIZA, that can be easily augmented with visual perception capabilities. This "third way", between the development of highly specialized visual processes in aseptic laboratory conditions and simple visual capabilities for highly complex conversational interfaces, allows to concentrate on visual problem solving while maintaining the proper problem embodiment.

References


---

1 In the sense that processes are developed without any problem "embodiment", using "aseptic" data.
Mobile Robots Monitoring System Using an Agent-Oriented Approach

Daniel Ponsa Mussarra
Computer Vision Center, Edifici O,
Universitat Autònoma de Barcelona,
08193 Bellaterra, SPAIN.
e-mail: daniel@cvc.uab.es

Jordi Vitià Marca
Computer Vision Center, Edifici O,
Universitat Autònoma de Barcelona,
08193 Bellaterra, SPAIN.
e-mail: jordi@cvc.uab.es

Abstract

In this paper we present an agent-oriented monitoring framework applied to the development of an application to control an autonomous guided vehicle fleet in an industrial environment. The framework has been defined from the analysis and the requirements of the basic needs of a general monitoring system, with the objective of establishing a platform where different strategies based on visual processes could be easily developed and tested using an agent-oriented approach. The framework allows different agents to work cooperatively on the same task as well as to distribute competitive agents in several areas of interest. The paper first describes the possibilities that offers a vision-based monitoring system to control and manage an autonomous transportation system. Then the structure of the multi-agent monitoring framework is presented. Its validity is shown by the implementation of a first prototype of a mobile robot monitoring station. This application runs in real time under Windows NT and has been developed in Visual C++, using the threads mechanism to implement agency.

Keywords: Multi agent system, mobile robots, tracking, dynamic scene annotation.

1 Introduction

In the last years computer vision has focused on a new research area: the study of scene dynamics. Some of the work done is oriented to the field of computer-human interface (American sign language recognition [9], face expression recognition, ...). The extraction of the dynamics of a scene is also of interest for applications such as automatic surveillance systems [8]. The common factor for all these applications is that the dynamics of the mobiles detected in the scene are monitored over time, and the information extracted is synthesized in higher level concepts that intend to describe the actions happening in the scene [1].

The aim of the work described in this paper is to construct a monitoring system for a mobile robot fleet in an industrial store. To deal with this objective an agent-oriented framework has been proposed generic enough to satisfy the basic requirements of any fixed camera scene monitoring system. The validity of the approach is tested by monitoring scenes without using any a priori knowledge of the mobiles characteristics, with the objective of showing the good operation of the agent community proposed.

2 Problem description

Autonomous Guided Vehicles (A.G.V.) have been introduced in the industrial environment to cope with the requirements of transportation lines for flexible production systems. The flow of the raw material in a production line has changed from being fixed, to depend on the kind of product required in a certain moment. AGV have become a kind of dynamic pipes between the material store and the production cells, enchainning the subproduct flow between cells until arrive to the product store. AGV require a minimum conditioning of its

This work has been supported by a grant from the Generalitat de Catalunya (Centre de Referència en Tecnologies Avançades de la Producció).
environment to be operational. They have odometric sensors that allow themselves to calculate in which spatial coordinates they are, and which are the components of their movement. Nevertheless this odometric measures suffer from accumulative errors that bring them to follow erroneous trajectories, so that their internal spatial coordinates have to be readjusted periodically. This is commonly solved by conditioning the environment in a higher degree and following landmark-based navigational strategies or using expensive mechanisms for absolute positioning.

One of the more common lacks of AGV fleets is that their components work individually instead of as a team. Individual AGV can establish communication to solve simple problems as, for example, the preference when there are conflicts in their trajectories, but they don’t go furthermore. In order to have a more productive behavior of the transportation system a global communication between its integrant parts is needed. This can be achieved having a global view of the transportation system so more optimal decisions can be taken.

Our work proposes a vision based monitoring system that can be useful to solve in one go some of the problems previously described. The system is connected to a central processing station that controls the state of all AGV in the system. This station receives the transportation needs from a user or a high level manufacturing process and decides which AGV is more suitable for a transportation task, and plans its trajectory minimizing the path cost and avoiding conflicts with other working AGV. To get the global state of the transportation system several vision-based monitoring stations are installed strategically in the AGV transited corridors. They have a zenithal vision of the corridors and inform the central station about the transiting AGV, their position, orientation, velocity and incidences. This station can readjust the AGV internal spatial coordinates too. Other interesting functions can be managed by this stations: the AGV load control, the traffic conflicts arbitration (for example, determine the preference in a crossroad), or the detection of stranges in the scene (for example load lost by an AGV, persons, etc).

The next sections focuses in the design and development of a part of the AGV control system proposed: the vision-based monitoring stations. The system as a whole is still in development, and its integration is planned for early 1999.

3 The vision-based monitoring station

In the design of the vision-based monitoring station we have been faced with the problem of defining a complex architecture of visual processes and tasks to cope with all the possible situations in our environment. We have stated the problem as a generic dynamic image annotation problem. The objective is to establish a basic framework that could be used as the starting point for any specific mobile monitoring application. Analysing the problem from the agent-oriented paradigm, a community of interacting processes is proposed.

3.1 The overall system

Any dynamic scene monitoring system has three basic needs to satisfy:

- Obtaining and maintaining a description of the static parts of the scene.
- Detecting the presence of new dynamic elements on the scene.
- Controlling (Annotating) the evolution along time of the dynamic elements in the scene.

They way that each task is done depends on the requirements and the specificity of the monitoring system. We propose a simple architecture based on a community of reactive agents that can be used in any monitoring system.

Figure 1: Monitoring agent architecture.

To achieve a concrete task (for example tracking arbitrary mobiles) sometimes doesn’t exist an optimal solution, and many techniques can be applied. We propose in these cases to accomplish the task by using several
agents with different behaviors but with the same objective, and determine the global solution integrating partial results.

The structure proposed permits the easy integration of higher level modules to the system. Some applications require a more conceptual description of the dynamics in the scene. Work in this area has been done using techniques as Bayesian Networks [8] and Hidden Markov Models [9].

Next the basic modules of a monitoring station are described.

**The acquisition agent** The agent responsible of this module has to acquire the images to be processed by the rest of the agents of the system. In simple applications its mission will be to acquire the images at the latency required to understand the dynamics of the scene. In multiple camera monitoring systems this module could take care of acquiring different images and constructing a unique representation of the scene by mosaicing the images of each camera.

**The vigilance agent** Given an static view of the scene, we can have a priori knowledge of the zones of the scene where dynamic objects can appear or disappear (image borders, doors). In order to detect new mobiles in the scene, we propose to have an agent dedicated to each one of these zones (frontier zones).

To detect the appearance of new mobiles, we use the robust change detector proposed in [3]. Having the mean and the standard deviation of the grey levels of the frontier zones from a reference image, the vigilant agents check periodically the variation of these values with respect to the ones calculated from the current image, alerting of a new mobile in the scene when the variation is outstanding. As a measure of variation we use the following likelihood ratio test:

$$\lambda = \frac{\sigma_1^2 + \sigma_2^2 + (\frac{\mu_1 - \mu_2}{2})^2}{\sigma_1 \sigma_2}$$  \hspace{0.5cm}(1)$$

When a vigilance agent detects changes in its controlled region, it continuously checks the region until it doesn’t detect more changes. Then it decides what have happened:

- An object have left the scene.
- An object have entered in the scene.
- A false alarm.

First of all the agent queries the state of the current tracking agents, checking if any of them have tracked a mobile near of its controlled zone. In this case, it orders the tracking agent to finalize, registering the information computed for the mobile in a public structure that can be analized by a higher order agent.

If any tracking agent have not arrived to the vigilance agent controlled zone, then a new mobile may have entered in the scene. Substracting the current image from a reference image, a zone of change can be detected ( besides the zones of change corresponding to the current tracked mobiles). If any new change region appears, then the situation is considered a false alarm and no action is performed. If a new change region is detected, its bounding box is calculated and passed to the tracking module, where different tracking agents will control the same object at the same time.

Note that this way of working makes the implementation of the system indifferent to the number of mobiles to be tracked.

**The tracking agent** Tracking is a complex task and there is not an ideal tracking algorithm to be applied in all situations. Tracking the same mobile with different agents that are based in different tracking algorithms allows the best tracking information for each situation. Actually each agent works in a non-cooperative fashion, and the tracking results are integrated a posteriori, but this is not the only alternative. It could be interesting that tracking agents for a given mobile could collaborate and share their mobile dynamics description, resynchronizing their values periodically. This will give more robustness to each tracking agent, and consequently to the overall system.

We have tested the feasibility of this proposal using three simple different agents.

**Blob Tracking Agent**: Since the system has a periodically updated reference image, a very simple generic tracking algorithm could be to follow the difference over time between the current image and the reference image. Having an a priori knowledge of the projected shape of the objects to be tracked, its very simple to characterize the position, orientation and velocity of the mobile.

**Correlation Tracking Agent**: When this agent is created, it establishes a greylevel pattern from the change region to be tracked
along the image. We use cross-correlation for the detection of the object from frame to frame. The position given by cross-correlation is calculated more precisely finding the affine transformation that better superpose the model with the image, starting from the cross-correlation results. This method allows to take into account at the same time the changes in orientation of the object frame to frame, and the optics deformation of the image, considered locally affine.

**Optical-Flow Tracking Agent**: This agent works without the use of a reference image. Having the knowledge that when we analyse a region with a mobile object we find two kinds of movement, the object motion and the background motion (static in a monitoring application), the mobile in the image can be segmented, and its dynamics computed. Mobile data is separated from background data using the scheme proposed in [4], and then the movement of the mobile is calculated using an affine approximation. The movement of the scene is formulated from an statistical point of view as a mixture of movements, and the Expectation-Maximization Algorithm is used to cluster the data in two possible coherent motions.

\[
\hat{x}_t = -(\alpha + \beta - 2)\hat{x}_{t-1} - (1 - \alpha)\hat{x}_{t-2} + \alpha \hat{x}_t + (-\alpha + \beta) \hat{z}_t \\
\hat{v}_t = -(\alpha + \beta - 2)(\hat{v})_{t-1} - \\
(1 - \alpha)(\hat{v})_{t-2} + \beta(x_t - x_{t-1})
\]

\[
\hat{v}_{t-1} = \hat{x}_t - \hat{x}_{t-1}
\]

where \(\hat{x}_t\) represents the estimated pose of an object at frame number \(t\), and \(\hat{v}_t\) represents the measured pose. \(\hat{v}_t\) is the estimated rate of change of pose per frame processed, and we use this value to determine the zone of interest at the frame \(t + 1\). \(\alpha\) is a real positive scalar less than 1 and \(\beta\) a real positive scalar less than \(\frac{\alpha^2}{1-\alpha}\).

Obviously the tracking here described is valid in a prototyping phase. For a concrete final application the mobile dynamics have to be more precisely modeled, the system optics have to be calibrated, and all the possible 3D information of the tracked objects has to be incorporated in the tracking algorithms [7].

![Figure 2: Photo assembly of the mobile detection and tracking process.](image)

**The image-reference agent** This module gives support to the vigilance and tracking modules supplying them a reference image. This image takes into account the variation of the scene along time not due to mobile objects (i.e. illumination changes). This slow dynamic changes are incorporated to the reference image using the algorithm proposed by [5]. Basically it is based on the generation of a median image over time, avoiding the zones of the

![Figure 3: Basic monitoring station succeeds in tracking people.](image)
image where a mobile object is being controlled. If the monitored environment can have unexpected changes (for example, sudden changes in the global scene illumination), alternative agent(s) can control the scene checking for them, suggesting the reference-image module a faster actualization.

4 Agency Implementation

The system has been implemented under Windows NT. It supports multiprocess applications that are ideally suited to implement agency. Furthermore, Windows NT is becoming a kind of standard in the industrial environment, due to its networking facilities and by the fact that the majority of PC expansion cards (frame grabbers, AD/DA converters, ...) supply drivers for it. Given that the real time requirements of the system are not very hard, Windows NT is appropriate. An accurate study of Windows NT real time possibilities can be found in [6].

To codify the system we used Microsoft Visual C++. We find appropriate to use an object-oriented language because the inheritance concept, and the virtual mechanism allowed us to establish an environment where changes could be easily made given that a basic communication interface where fixed for each class of agents.

The synchronization of the shared data access has been done using critical section objects, and the communication between agents has been achieved by shared data and the event communication mechanism.

5 Experiments

The system has been implemented using a standard Pentium 200 MHz MMX with 64 Mb of RAM. Using an standard interlaced camera, several image sequences were taken. Different lenses were used (3.5 mm, 8.5 mm, 12mm), causing different degrees of distortion to the images, and different kinds of mobiles were tracked (rigid and non-rigid ones). The images of the analysed sequences were sampled by a factor of two (remaining as 384x288 pixel images), avoiding the image disparities that due to motion appear between the image fields.

We have tested the performance of our architecture by varying the computational load of the system. The critical part of the application is the "soft" real time response that tracking agents can support. Hence, several tests have been realised in order to quantify the response time variations when the system is faced with multiple mobiles.

4 Figure 4: Image from an interlaced camera with a 8.5 mm optics and its sampled version.

The following graphics show the system response when the number of tracking agents in the scene varies. The delay between tracking cycles is shown, and its statistics displayed. The tracking agents were implemented using Normal Priority Class threads.

5 Figure 5: Tracking cycle delay in milliseconds depending of the number of tracking agents.

6 Conclusions and future work

The paper has presented a basic framework to prototype vision-based monitoring systems. We used this framework to develop a simple monitoring station in an standard PC. We have described how the system has been designed and presented some examples to illustrate the performance of the system. New work is under way to apply the framework proposed for controlling a real AGV system, incorporating camera calibration and 3D model-based tracking. Cooperation between tracking agents and a high-level description of the scene is studied too.
References


Reinforcement Learning of Sensor-based Robot Arm Motions for Multi-Goal Reaching Tasks

Pedro Martín
Dept. of Computer Science, University of Jaume I
12071 Castellón, Spain
martin@inf.uji.cs

José del R. Millán
ISIS, Joint Research Centre, European Commission
21020 Ispra (VA), Italy
jose.millan@jrc.it

Abstract

Sensory information is fundamental for autonomous robots that must face unknown environments. We present a real sensor-based manipulator, namely a Zebra ZERO, that has a sonar sensing skin mounted along its links. Our work deals with the learning of goal-oriented obstacle-avoiding motion strategies for such a manipulator in unknown 3D environments. We have developed a controller that is made up of two main modules: an action generator (AG) and a goal vector generator (GG). The controller acquires motion strategies through reinforcement learning from local sensory data. The actions given by the AG are interpreted with regard to a goal vector in the robot joint space obtained by the GG. The AG has two reinforcement-based neural modules: one for negotiating obstacles and another one for moving to the goal. The GG uses a neural differential inverse kinematics module to produce suitable goal vectors. This kind of controller allows the manipulator to learn suitable motion strategies while trying to perform a high-level task that requires the end-effector to be placed sequentially at different goal locations.

Keywords: Differential inverse kinematics; Neural networks; Sensor-based manipulators; Multi-Goal reaching tasks; Reactive systems; Reinforcement learning.

1 Introduction

In order to be able of executing high level descriptions of tasks, an autonomous robot manipulator must have the ability of performing efficient motions to reach a desired goal location from some initial location while avoiding obstacles. Traditional approaches to that problem require a previous geometrical model of the workspace [6]. They address the problem in two steps: (a) the computation of the configuration space of the robot arm [9] and (b) an off-line planning process over the configuration space to find a collision-free path before any motion can take place. These methods are suitable for robot manipulators working in the same perfectly known and static environment. However, the real world - even if known a priori - is subject to unexpected occurrences or rapidly changing situations. There are also robotics tasks in which a robot arm must face continuously different environments, e.g., a robot arm mounted on a mobile robot platform that is commanded to carry out several manipulations at different places of a building - what resembles much more the typical demand that our human arms must undertake. Moreover, robot arms are sometimes needed to operate in hazardous environments (e.g., remote hazardous/radioactive waste cleanup), which are usually unknown the first time robots face them. All those situations claim the use of sensors for a more flexible operation of robot manipulators. On-line sensing allows a robot arm to modify their motion in real-time. Recently, there are attempts to achieve collision-free motions of the end-effector by making use of visual information [15]. Yet the use of cameras is not suitable for a constant survey of the whole area surrounding the manipulator. In fact, the robot arm itself often can hide critical areas and visual processing can be time-consuming. Whole arm sensing can be achieved by sensitive skins with proximity sensors like infrared [4] or tactile sensors [2]. Proximity sensors are specially interesting as they permit to prevent collisions before any real contact can occur, and so they preserve the physical integrity of both the robot arm and of the environment. Such feature make them be appealing as it allows robot arm manipulators to operate with humans side by side in assisting tasks or in constrained environments with critical and/or valuable hardware.

Reactive systems (e.g., [3]) are appropriate to generate on-line motions from local sensory data. A reactive controller can be implemented automatically by using artificial neural networks and reinforcement learning (RL) [1,14,17]. RL allows a neural network to acquire reaction rules while the robot arm interacts
with its environment. We have previously applied RL successfully to autonomous mobile robots [13]. We have also demonstrated the feasibility of RL to acquire sensor-based reaching strategies for robot arms in our previous works with simulated two-link [12] and three-link planar manipulators [11]. In this paper, these last works are extended for a real manipulator, namely a Zebra ZERO, that has a whole-arm sensitive skin with sonar proximity sensors (see Fig. 1). We describe a neural reactive controller that learns goal-oriented obstacle-avoiding motion strategies for such a manipulator in unknown 3D environments. The controller is made up of two main modules: a reinforcement-based action generator (AG) and a goal vector generator (GG). The AG uses local sensory data and position information to determine an appropriate deviation from the goal vector given by the GG. The task of collision-free reaching can be decomposed into two sequential subtasks: Negotiate Obstacles (NO subtask) and Move to Goal position (MG subtask). When the robot arm is not near the goal position and detects an obstacle in its way to the goal, the best strategy is to focus on negotiating the obstacle—moving along an efficient trajectory is not so important. The robot arm deals with the MG subtask later on, when the goal position is nearby. As each subtask may require different motion strategies, the AG has two different reinforcement-based modules. When the robot is dealing with the NO subtask, the GG provides a fixed goal vector in the joint space that allows the AG to learn goal-independent motion strategies. For the MG subtask, the GG efficiently computes a goal vector in the configuration space by using a differential inverse kinematics (DIV) neural network.

2 The sensor-based manipulator

The Zebra ZERO (see Fig. 1) is a 6 dof manipulator with a gripper and a force sensor attached to the wrist. All its six joints are revolute: three joints are associated with the arm (first three links) and the other three with the wrist. The controller generates angular motions for the three joints of the arm $(q_1, q_2, q_3)$.

In the experiment, the angular movements of the joints are constrained in degrees as follows: $(q_{1\text{min}}, q_{1\text{max}})=(-180,180)$, $(q_{2\text{min}}, q_{2\text{max}})=(10,65)$ and $(q_{3\text{min}}, q_{3\text{max}})=(-50,0)$.

The sensitive skin has 13 ultrasonic smart sensor modules that are allocated in two sections. The sensors provide a number in the range $[1,255]$ that corresponds to the proximity of an object between 0.1 and 25.5 inches approximately. The value 255 is also used when no obstacle is detected. Both the manipulator and the sensitive skin are commanded from a PC with a 80386 microprocessor working at 36 MHz and with 640 Kb of memory. While performing the action given by the neural controller at each step, the robot stops and retracts if any of the sensors detects an obstacle closer than a prespecified safety distance $d_s=2$ inches.

Figure 1: The Zebra ZERO robot arm with the sonar skin.

3 Controller architecture

The controller has two main components: an action generator (AG) and a goal vector generator (GG). The action generator produces actions that are interpreted with regard to a goal vector (GV) previously computed by GG. The AG is a modular reactive neural system that acquires the motion strategies through reinforcement. A neural differential inverse kinematics (DIV) module is employed in the GG.

Fig. 2 shows how the controller interacts with the robot arm environment. The AG receives 25 real-valued inputs that range in the interval $[0,1]$. These inputs are divided into three groups: goal, position and sensory inputs. Goal inputs $g$ provide information about how close the robot arm is getting to the goal position. They correspond to the components of the workspace shortest path vector (WSPV) normalized between 0 and 1. Given the robot end-effector location $(\phi_1, \phi_2, P)$ and the goal location $(\theta_1, \theta_2, P, g)$, both in spherical coordinates, the WSPV is given by $WSPV=(\theta_1-\phi_1, \theta_2-\phi_2, P-g)$. The NO module of the AG does not make use of the goal inputs, as we want the NO module to learn goal-independent motion strategies. Position inputs $p$ give information about the current joint configuration of the robot arm: $p_i=\theta_i-q_i'$, $q_{i\text{min}}$, $q_{i\text{max}}$, $i=2,3$. The
input $p_i$ is not supplied to the NO module since the way in which the robot can negotiate a certain obstacle while getting closer to the goal is independent of the angular value of the first joint. Three kinds of sensory inputs $d$ can be distinguished: front, back and joint-limit. The front and back components correspond to the sonar readings from the sensing skin. They are computed as follows: $d_i = 1 - (v_i/255)$, $i=1,13$, where $v_i$ is the range value given by the corresponding sonar sensor. The front inputs come from the sensors that lie in the direction of rotation that the first joint must follow to approach the goal. The joint-limit components are associated with virtual sensors. They give information about the proximity of the joints to their physical limits. There are two joint-limit inputs for each joint: $d_{min} = 1 - (q_i - q_{min})/17.2$, $d_{max} = 1 - (q_i - q_{max})/17.2$, $i=1,2,3$. The joint-limit inputs are forced to be zero when their computation gives a negative value. A sensory input has a value close to 1 when its associated sensor is detecting a very near obstacle.

Finally, the AG gives an action vector $a=(a_1, a_2, a_3)$ that specifies a movement in relative spherical coordinates with regard to the goal vector (GV) given by the GG. All the components are real values in the interval $[-1,1]$. $a_1$ and $a_2$ represent the horizontal and the vertical angular deviation from the GV, respectively. $a_3$ corresponds to the length of the action vector as a proportion of the length of the GV. The actual angular increments $\Delta q=(\Delta q_1, \Delta q_2, \Delta q_3)$ supplied to the robot joints are calculated from the action vector.

The reinforcement signal $r$ is used to update the neural modules of the AG. It is computed in the same manner for both the NO and MG modules. The AG is severely punished when it runs into a virtual collision (i.e., if any sensor detects an obstacle closer than a safety distance): $r=20$. For collision-free situations, $r$ pays attention to approach the goal properly: $r=15e^{-pr}$, where $pr = (1 - |\theta_i - \theta_r|)/|\theta_i - \theta_r|/|GV|$ measures the progress of the robot to the goal. Note that the reinforcement signal is only based on local information.

3.1 Action generator

The decomposition of the collision-free reaching problem in the NO and MG subtasks is achieved by defining a suitable region of reachability around the goal position. Given the end-effector location $(\phi_r, \phi_p, \rho)$ and the goal location $(\theta_r, \theta_p, \rho)$, both in spherical coordinates, the robot arm is considered to be inside the region of reachability when $|\theta_i - \phi_i| < 20$ degrees. As each subtask may require different motion strategies, the action generator has two connectionist reinforcement-based modules, the NO module and the MG module (see Fig. 3). In addition, the AG has another simple reactive module that allows the robot to follow the GV (FG module). The FG module is used when the robot’s sensors do not detect any obstacle along the direction to the goal. The NO module is in operation when an obstacle is detected along the direction to the goal and the robot arm is outside the region of reachability. The MG module is switched on when the end-effectors enters the region of reachability. These last two modules have an actor-critic architecture [1].

The actor of each module aims at learning to perform those actions that optimize the total reinforcement received along the trajectory to the goal (i.e., the cumulative reinforcement). The actor must discover which actions yield the highest reward through a trial-and-error search. Hence there must be a source of randomness so as to explore the action space. For this purpose, the actor is made up of two artificial neural networks: the policy network, which codifies the situation-action rules, and the exploration network (see Fig. 4). Both networks have one hidden layer of 50 units with logistic activation functions. This
The policy network has Gaussian units that generate the three action components by using random Gaussian distributions $N(\mu_i, \sigma_i)$, $i=1,2,3$. The mean value $\mu_i$ for the Gaussian unit $i$ is the net input coming from the hidden layer to that unit. The standard deviation $\sigma_i$ of each Gaussian unit is provided by the corresponding output unit of the exploration network, which is linear. Figure 4 shows how the action components are finally calculated.

Figure 4: Actor, Policy and exploration networks

In order to deal with the temporal credit assignment problem, the critic uses temporal differences (TD) methods [14] to learn the cumulative reinforcement. In particular, we use TD(0). The critic is also implemented as an artificial neural network with one hidden layer of 50 units. The activation functions of the hidden units are logistic with asymptotes at $-1$ and $1$, while the output unit is linear. As a measure of the cumulative reward at time $t$, the discounted sum of future reinforcement is normally used:

$$z_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (2)$$

where $\gamma$ is the discount factor, with $0 \leq \gamma < 1$. The discount factor determines the degree to which long-term consequences of the actions are taken into account. The weights of the critic are updated by backpropagating TD(0) errors:

$$\Delta \theta_t = r_t + \gamma V_t(s_{t+1}) - V_t(s_t) \quad (3)$$

where $V_t(s)$ is the prediction of the cumulative reinforcement for situation $s$ made by the critic with the state of its weights at time $t$. It is worth noting that the TD error indicates whether the last action was better (or worse) than expected and so it is also used to update the weights of the actor. In particular, the TD error is employed as the reinforcement offset term $(r-b)$ of Expression (1).

### 3.2 Goal vector generator

The GG provides suitable goal vectors for the different modules of the AG. Whenever the NO module of the AG is in operation the following goal vector in the joint space in degrees is provided: $GV=(20,0,0)$, if the angle of the first joint must be increased to approach...
the goal location and $GV = (-20, 0, 0)$, in the other way around. For the MG module, the GG produces suitable goal vectors with a differential inverse kinematics (DIV) neural network (see next section).

## 4 Differential inverse kinematics network

In general, a goal vector that points to one of the arm goal configurations can be obtained by using the pseudoinverse of the Jacobian matrix [16]. This approach is computationally expensive, as the pseudoinverse of the Jacobian matrix must be recalculated for every new joint configuration of the robot arm. In addition, it can not be applied to arms whose kinematics are unknown or difficult to be specified mathematically. On the other hand, neural networks can be used to learn the kinematics relationship of manipulators during an action-perception cycle without requiring explicit knowledge of their geometry. This latter approach is more biologically plausible as it tries to reproduce the hand coordination ability of human beings.

### 4.1 Distal learning approach

Our differential inverse kinematics neural method is based on the distal learning approach to the inverse kinematics problem [7]. In this approach, the forward kinematics of the manipulator $q \rightarrow x$ is learned by a feedforward neural network through backpropagation. For on-line backpropagation, the typical cost functional to be minimized during learning corresponds to

$$E(q) = \frac{1}{2} (x^* - F(q))^T (x^* - F(q))$$

$$= \frac{1}{2} (x^* - x)^T (x^* - x)$$

where $x^*$ represents the desired output for the network and function $F(q) = x$ stands for the approximation of the forward kinematics carried out by the network.

Then, the forward model is used to train another feedforward network for the inverse kinematics $x \rightarrow q$. The outputs of the inverse model are connected to the inputs of the forward model (see Figure 5). The objective of learning is to update the weights of the inverse network such that the composite system converges to the identity mapping. During the training of the inverse network, the weights of the forward model are kept fixed and the same cost functional (4) is considered. At the training step $t$, the weight vector $w$ of the inverse model is incremented by $\Delta w = -\alpha \nabla_w E$, where $\alpha$ represents the learning rate. The gradient of the error function is computed in the following manner by applying the chain rule to (4)

$$\nabla_w E = -\frac{\partial q^T}{\partial w} \frac{\partial x^T}{\partial q} (x^* - x) = \frac{\partial q^T}{\partial w} \nabla_q E$$

$$\nabla_q E = -\frac{\partial x^T}{\partial q} (x^* - x)$$

This expression describes the propagation of the distal error $\Delta x = x^* - x$ backward through the forward model; this effectively multiplies the distal error by an estimate of the transpose Jacobian matrix $\partial x/\partial q$. Such computation corresponds to the gradient of the error function $\nabla_q E$ with regard to the inputs $q$ of the forward model. As the composite system is forced to converge to the identity mapping, this approach assures that the inverse model learns one inverse kinematics solution, even for redundant manipulators.

### 4.2 The differential inversion algorithm

The gradient $\nabla_q E$ of the error function with regard to the inputs $q$ of the forward model can be seen as an estimate of training error $\Delta q$ for the inverse model

$$\Delta q = \frac{\partial x^T}{\partial q} \Delta x = J^T \Delta x$$

This equation shows that the backpropagation procedure over the forward model carries out an indirectly computation of the Jacobian transpose of the manipulator throughout the weights of the network.

From difference vectors in the workspace, the estimate of the Jacobian transpose of the manipulator in (6) can be used to obtain difference joint vectors that make the end-effector of the manipulator approach the goal position. In addition, the gradient $\nabla_q E$ can be employed to minimize the error function in the input space of the forward network in order to obtain an inverse kinematics solution. Our DIV approach is based on such inversion of a neural network through backpropagation [8].

**Figure 5:** Distal learning approach for inverse kinematics learning.
Let $q$ and $x$ be the current configuration and the current end-effector locations of the robot arm. Let $x^*$ be the goal location where the end-effector must be finally placed. A goal configuration $q^*$ is a configuration that makes the end-effector be placed at the goal location $x^*$, that is $F(q^*)=x^*$. The set of all goal configurations $q^*$ for the goal location $x^*$ is denoted by $Q(x^*)$. The inverse kinematics problem can be stated from a differential inverse kinematics framework as follows: “Given the vectorial difference $\Delta x=x^*-x$ in the workspace between the current location of the end-effector $x=F(q)$ and a goal location $x^*$, obtain the closest goal configuration $q^*=\arg\min_{q^* \in Q(x^*)} \|q^*-q\|$ to the current configuration $q$.”

A gradient search in input space of the forward kinematics network is conducted by the backpropagation inversion algorithm. The search generates a sequence of input vectors $q=q_0, q_1, ..., q_n$ such that the error $E(q^n)$ is minimal. This is the same error function (4) that has been used to train the neural network with the forward kinematics. The sequence starts with the current configuration of the robot arm. Each successive input configuration $q_i$ is changed by adding an amount $\Delta q_i$ proportional to the components of the gradient $\nabla_{q_i} E$ of the error function $E$ with respect to $q_i$,

$$\Delta q_i = -\eta \frac{\partial E}{\partial q_i} = -\eta \sum_j \frac{\partial^2 E}{\partial q_i \partial q_j} \Delta x = \eta \left( J^T \right)^T \Delta x \quad (7)$$

where $\eta$ is the inverse rate. By backpropagating the error, each input component must be updated as follows:

$$q_i^{i+1} = q_i^i + \Delta q_i = q_i^i - \eta \delta_{oi}$$

$$\delta_{oi} = \sum_j w_{oji} \delta_{1j} \quad (8)$$

where $\delta_{mi}$ is the backpropagated error associated with the output unit $i$ of layer $m$, with $m=1,2, ..., M$, and $w_{oji}$ represents the weight of layer $m$ that connects unit $i$ of this layer with the unit $i$ of the previous layer $m-1$. The first hidden layer is referred by $m=1$ and $M$ corresponds to the output layer. Such error is expressed as:

$$\delta_{mi} = \sum_j \delta_{m-1j} w_{mij} f'(h_{mi}) \quad (9)$$

where $h_{mi}$ is the net input of the $i$th unit of layer $m$.

The inversion of the neural network is carried out in an iterative way (see Figure 6). At each iteration, the configuration $q^*$ is fed to the network to generate its corresponding output $F(q^*)=x^*$. Then, the corresponding error $E(q^*)$ is computed and backpropagated to have the next configuration of the sequence $q^{n+1}$. This procedure finishes when $\|x^*-x_i\| < \xi, \forall i$. The error tolerance $\xi$ must be small enough to guarantee that the last input vector $q^n$ to the network corresponds to a goal configuration $q^*$. A greater error tolerance lead to a configuration $q^n$ that will make the robot arm get closer to a goal configuration. From the last configuration of the sequence, the difference vector $\Delta q=q^*-q$ can be obtained. In order to approach a goal configuration, the AG of the controller does not need an accurate goal vector. A vector $\Delta q$ that indicates a proper direction to follow is sufficient. Consequently, a greater error tolerance can be used to obtain such a goal vector with a little number of iterations. In this way, suitable goal vectors for both the MG and FG modules of the AG are obtained in just one iteration of this neural inversion by using an inversion rate $\eta = 1.0$.

The DIV algorithm can be viewed as a neural network implementation of the CLIK Jacobian transpose method [5]. This method is derived by finding a relationship between $q$ and $\Delta x$ that ensures error convergence to zero. The Lyapunov direct method is utilized to determine such relationship. Given the Lyapunov function

$$E(\Delta x) = \frac{1}{2} \Delta x^T K \Delta x \quad (10)$$

where $K$ is a symmetric positive definite matrix, it can be proved that the “control law”

$$q = J^T K \Delta x \quad (11)$$

ensures that the end-effector tracking error $\Delta x$ is ultimately bounded into an attractive ball. The radius of the ball can be made arbitrarily small by suitably increasing the minimum eigenvalue of the positive definite matrix $K$. In the DIV algorithm the cost functional to be minimized during the inversion is also a Lyapunov function with the identity matrix as the $K$ matrix. Other $K$ matrix can also be used for the DIV method. The block scheme of the inverse kinematics algorithm with Jacobian transpose is equivalent to the DIV one shown in Figure 6; but now the error propagation stage is carried out by an explicit transpose of the manipulator Jacobian and the forward kinematics equations must be known in order to implement the forward computation stage.
4.3 Forward kinematics learning

We have implemented the DIV method for the three first joints of the manipulator to be used in the GG of the controller. Although the solution to the inverse kinematics of the Zebra ZERO arm is easy to compute, we have used the DIV module in the GG to demonstrate its functionality.

We have employed a feedforward neural network with a single hidden layer of 50 units to approximate the forward kinematics of the robot arm in spherical coordinates. The network is trained through backpropagation with momentum. Different learning rates have been utilized for the output and the hidden layers. The output learning rate is 0.01 and the hidden learning rate is 0.02. A 3D grid with a resolution of 3 degrees are employed to collect the samples. The accuracy of the trained network is of 5% of the workspace radius with regard to Cartesian coordinates. The components of the training samples \((q,x)\) are scaled between \(-1\) and 1. The desired outputs \(x\) in the training pairs are computed from the forward kinematics equations of the manipulators. However, it is worth noting that the forward kinematics relationship is not indispensable to be known for the method. Workspace locations can be properly specified in image coordinates of cameras. For planar manipulators, just one camera is enough to obtain suitable image coordinates with two components. Two cameras are needed for the 3D case. In such case, the target locations are identified with a four dimensional vector that results from the image coordinates of the two cameras [15]. The training pairs can be collected through an action-perception cycle by moving the manipulator to different joints configurations and computing the corresponding image coordinates from the cameras. The forward model will learn the manipulator forward kinematics as the relationship between joint configurations and image coordinates.

The inversion of the neural network guarantees to obtain the closest goal configuration \(q^* \approx q^f\) if the error tolerance \(\xi\) is sufficiently small and, therefore, the difference vector \(\Delta q^f - q^f\) corresponds to the Shortest Path Vector (SPV) in the joint space. In order to obtain the SPV, training the network to approximate the forward kinematics in spherical coordinates is fundamental to avoid undesirable local minima that appear when using Cartesian coordinates [10].

5 Experimental setup and results

A high level task for a manipulator generally requires the end-effector of the manipulator to be placed sequentially at different goal locations. A suitable learning method for acquiring collision-free motions must allow the robot to learn appropriate motions while trying to accomplish high level tasks. Our controller copes with such a realistic demand. An experiment has been carried out in the environment shown in Fig. 7 to test the performance of the controller. Three different goal locations were considered. The experiment consists of 10 runs, differing in the initial seed of the random number generator. All runs are made of 300 trials. At the beginning of a trial, the robot is commanded to reach one of the goal locations. Once the arm arrives there, a new trial starts whose target is one of the remaining goal locations (selected at random). Thus, the starting and target locations change from trial to trial.

![Figure 7: Environment used in the experiments (profile and top view).](image)

Different learning rates have been employed for the two layers of all the networks in the actor-critic modules. We have used the same learning rates for the policy and exploration networks. For these networks, the learning rates in the output and hidden layers are 0.03125*κ and 0.015625*κ, respectively. The learning rates for the output and hidden layers of the critic networks are 0.0625*κ and 0.125*κ, respectively. The symbol κ designates a parameter that has been used to tune proportionally the learning rates in a series of initial experiments. In all the cases, κ is set to 0.05. In the experiment, a discount factor γ of 0.95 has been utilized.

For each run, the performance of the controller is measured every group of 30 trials according to two
parameters: the number of collisions and the rate of steps (see Fig. 8). The length of the motion that can perform the robot at each step is limited to a maximum. The minimum number of steps that the robot can perform to reach the goal location corresponds to the number of steps needed by the robot to follow the shortest path in the configuration space (i.e., the straight line between the starting and goal configurations) when applying the maximum motion at each step. This is the optimum path for the robot to get to the goal location, provided that no obstacle blocks such path. The number of steps performed by the robot at each trial is divided by that minimum number of steps in order to obtain the rate of steps. Collisions fall down below 1 per trial after only 150 trials (around 4 hours of robot operation) and the rate of steps is also reduced to 2.3. The performance of the controller keeps on improving as learning proceeds. It is worth noting that since the learning task is defined by the goal location and the obstacles as perceived by the sensors, in this experiment the robot arm faces a different environment at each trial. However, the controller succeeds in learning environment-independent motion strategies. This is mostly due to two factors, namely, the NO module is supplied with local goal-independent information and the reinforcement signal. The input codification, which exploits the symmetry of the sonar information (front and back sensory inputs), has also been essential to achieve such a performance of the controller.

6 Conclusions

We have proposed a controller to learn obstacle-avoiding reaching strategies for sensor-based multilink robot arms. This approach has been tested on a real robot arm, namely a Zebra ZERO, that has a sonar sensing skin mounted along its links. The controller consists of two main modules: a reinforcement-based action generator (AG) and a goal vector generator (GG). The collision-free-reaching task is decomposed into two sequential subtasks: negotiate obstacles (NO) and move to goal (MG). The AG is implemented with two connectionist actor-critic modules, one for each subtask. The GG makes use of a differential inverse kinematics (DIV) neural module to obtain suitable goal vectors for the MG module. The NO module is supplied with a fixed goal vector that is independent of the goal location. The use of this goal vector, as well as the environment-independent input codification and reinforcement schedule for the controller, facilitates the learning of the motion strategies in realistic situations with different goal locations at each trial.

This method has been only tested for the three-first link of the Zebra ZERO. However, the use of the DIV module in the controller allows the method to be applied to redundant arm robots [11]. The Zebra ZERO becomes redundant if the wrist and the gripper together are considered as a fourth link. We have also tested the controller with a DIV module for this case but without obstacles since, unfortunately, the wrist and the gripper are not covered by the sensitive skin so far. Due to the lack of sensory rings, the robot task consisted in reaching the goal position from different initial configurations avoiding to exceed the limits of the joint angles. No obstacles were considered. The controller brought successfully the robot arm to the goal position since the very first trial along acceptable trajectories and allowed a real-time operation of the manipulator.

Acknowledgments

Pedro Martin was at the Joint Research Centre of the European Commission in Ispra (Italy) while most of this research was done. His stay was supported by a grant from Fundacio Caixa-Castelló (1997). He also acknowledges partial support from the Fundacio Caixa-Castelló (1998) research project leaded by Dr. Mª Teresa Escrig.

References


Estimación bayesiana de características en robots móviles mediante muestreo de la densidad a posteriori

Domingo Gallardo, Francisco Escolano, Ramón Rizo, Otto Colomina, Miguel Angel Cazorla
domingo,sco,rizo,otto,miguel@dccia.ua.es
Grupo VGIÁ: Visión, Gráficos e Inteligencia Artificial
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

Resumen
La obtención de características estables y robustas del entorno en un robot móvil es un elemento clave para la extracción autónoma de mapas del entorno, la localización en el mismo o la navegación de una localización a otra.

Proponemos en este trabajo la estimación y el seguimiento de características topológicas de forma robusta y estable mediante la utilización de técnicas de muestreo de la función de densidad a posteriori, a partir de la formulación de un modelo probabilístico del sensor y del movimiento del robot. Estas técnicas de muestreo permiten representar una densidad arbitraria de un espacio paramétrico continuo, frente al filtro de Kalman, que necesariamente se aplica a distribuciones gaussianas, o las redes bayesianas temporales, en las que es necesario definir un conjunto discreto de estados.

Por último, comprobamos los resultados sobre lecturas de sonar obtenidas en entornos simulados, en distintas condiciones de ruido y variabilidad.

Palabras claves: estimación y seguimiento temporal, seguimiento no gaussiano, muestreo de densidad de probabilidad, características geométricas estables y robustas, robots móviles.

1 Introducción
La obtención de características estables y robustas del entorno en un robot móvil es el paso previo para una posterior extracción autónoma de mapas del entorno, localización en el mismo o navegación de una localización a otra [11]. Las características se suelen extraer del entorno a partir de sensores de ultrasonidos, elementos estándar de captación de información en la robótica móvil. Estos sensores se utilizan normalmente en arquitos de 12 o 24 transductores distribuidos de forma equidistante alrededor del robot.

La popularidad del sonar se debe fundamentalmente a que se trata de un sensor de muy bajo coste, su consumo es muy pequeño y el procesamiento que se realiza a su señal es muy sencillo, haciendo posible una respuesta casi inmediata y permitiendo una reacción rápida del robot.

Sin embargo, frente a estas ventajas, el sonar tiene como principal inconveniente el elevado nivel de ruido e incertidumbre presente en sus lecturas, debido a diversos factores [7].

Existe una amplia colección de trabajos en los que se proponen métodos para filtrar los resultados obtenidos por estos sensores y obtener características geométricas elementales [3, 1, 12]. Sin embargo las características obtenidas en todos ellos son muy locales, como aristas, esquinas o segmentos, y esta propia localidad hace que sean muy sensibles al ruido, poco robustas y poco estables. Esto último se acentúa en entornos dinámicos y variables, del tipo en los que suelen evolucionar estos robots.

Frente a este planteamiento local aparece como alternativa la utilización de enfoques bayesianos en los que se va acumulando en el tiempo la evidencia asociada a la característica. Aunque este enfoque ha sido aplicado con éxito desde los primeros momentos de investigación en el campo de la robótic-
ca móvil [4], es ahora cuando se ha despertado un gran interés en su uso para aplicaciones de mayor alcance [14, 2, 5, 15, 10]. En esta línea, nuestro trabajo se basa en características topológicas de más alto nivel, como son pasillos o finales de pasillo (ver figura 1). Estas características se extraen de los datos del sonar utilizando un enfoque bayesiano que presentamos en la siguiente sección. Esta metodología es genérica y permite ser aplicada a otro tipo de características topológicas, como conexiones entre pasillos, etc.

En concreto, el núcleo de la propuesta se basa en la utilización de técnicas iterativas de muestreo para estimar la densidad a posteriori de estas características topológicas. Estas técnicas permiten representar la densidad de probabilidad de forma explícita, mediante un conjunto \((m_1, m_2, \ldots, m_N)\) de muestras extraídas de dicha función de probabilidad. Para obtener estas muestras, utilizaremos el algoritmo CONDENSATION [9], una versión del algoritmo de rechazo [13] aplicada al muestreo de distribuciones a posteriori. Presentamos este algoritmo en la sección 3. En la sección 4 formulamos en detalle el modelo dinámico del sistema y el modelo de verosimilitud. Por último, en el apartado 5 analizamos los resultados de aplicar este enfoque a la detección de características obtenidas por el sonar en entornos con ruidos y con abundante variabilidad, comprobando que las características son estables y robustas.

2 Técnicas para la estimación bayesiana temporal

Un problema estándar en reconocimiento estadístico de patrones es encontrar un objeto parametrizado con el vector \(x\) con una distribución de probabilidad a priori \(p(x)\) utilizando unos datos \(z\) [6]. Aplicando este modelo genérico al problema que tratamos en el artículo, el vector \(x\) representa la parametrización de una característica a buscar en el entorno (pasillo, final de pasillo, etc.) y los datos \(z\) representan las lecturas del sonar.

Para estimar \(x\) a partir de \(z\) es necesario conocer la distribución condicional \(p(z|x)\) que mide la verosimilitud de que unas lecturas \(z\) correspondan a un hipotético objeto \(x\). La densidad a posteriori \(p(x|z)\) representa todo el conocimiento sobre \(x\) que puede ser deducido a partir de los datos obtenidos. Esta densidad se calcula utilizando la regla de bayes

\[
p(x|z) = \alpha p(z|x)p(x)
\]

donde \(\alpha\) es una constante de normalización independiente de \(x\).

2.1 Estimación bayesiana temporal

La versión dinámica del planteamiento bayesiano supone que tenemos una secuencia de \(t-1\) lecturas en el tiempo \(z_{t-1} = (z_1, z_2, \ldots, z_{t-1})\) que han sido producidas por una evolución del objeto \(x\), cuya historia denotamos por \(x_{t-1} = (x_1, x_2, \ldots, x_{t-1})\). La dinámica de dicho objeto ha sido producida por una secuencia de acciones también conocidas \(a_{t-1} = (a_1, a_2, \ldots, a_{t-1})\) (en nuestro caso cada acción \(a\) será un valor de velocidad lineal \(v\) y angular \(\omega\) del robot móvil). La figura 2.1 representa gráficamente el proceso.

Para realizar una estimación temporal debemos formular el modelo dinámico del sistema y el modelo de medida. En los siguientes subapartados presentamos las restricciones generales que deben cumplir ambos y en la sección 4 detallaremos su expresión para el caso del seguimiento de características del entorno a partir de datos del sonar.

2.1.1 Modelo dinámico

El modelo dinámico proporciona una expresión de la probabilidad de un determinado estado a partir

Figura 1: Características topológicas usadas en el trabajo: pasillos y finales de pasillo.

Figura 2: Estimación bayesiana temporal.
de una secuencia de estados y acciones previas

$$p(x_t|x_{t-1}, a_{t-1}).$$  

(2)

Asumimos que este modelo dinámico cumple la condición de Markov

$$p(x_t|x_{t-1}, a_{t-1}) = p(x_t|x_{t-1}),$$  

(3)

esto es que el nuevo estado depende únicamente del estado y de la acción anterior. Asumimos también que este modelo es estacionario, esto es que no depende del instante de tiempo, por lo que deberemos formular más adelante la expresión $p(x'|x, a)$.

### 2.1.2 Medidas

Suponemos que las observaciones $z_t$ son con respecto al proceso dinámico, lo que se expresa probabilísticamente como

$$p(x_{t-1}, x_t|x_{t-1}, a_{t-1}) = p(x_t|x_{t-1}, a_{t-1})p(x_{t-1}|x_{t-1}, a_{t-1}).$$  

(4)

También suponemos que son independientes entre sí,

$$p(x_{t-1}|x_{t-1}, a_{t-1}) = \prod_{i=1}^{t-1} p(z_i|x_t, a_t).$$  

(5)

y que la observación únicamente depende del estado, y no de la acción realizada

$$p(z_t|x_t, a_t) = p(z_t|x_t).$$  

(6)

Suponemos, por último, que este modelo también es estacionario y que no depende del instante de tiempo, quedando una densidad de probabilidad $p(z|x)$ que define la verosimilitud de unas lecturas dado una hipótesis característica del entorno y que detallaremos más adelante.

### 2.1.3 Propagación de la probabilidad a posteriori

Asumiendo las consideraciones anteriores, la propagación de la probabilidad a posteriori en los modelos de propagación bayesiana temporal viene dada por

$$p(x_t|z_t, a_t) = \alpha_t p(z_t|x_t) p(x_t|x_{t-1}, a_{t-1}),$$  

(7)

siendo

$$p(x_t|z_{t-1}, a_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1}, a_{t-1}) p(x_{t-1}|z_{t-1}, a_{t-1}).$$  

(8)

Estas expresiones son el equivalente temporal a la regla de Bayes y determina de forma iterativa la distribución de probabilidad a posteriori en función de la última lectura del sensor $z_{t-1}$, la última acción $a_{t-1}$, el modelo dinámico $p(x'|x, a)$, el modelo de sensor $p(z|x)$ y la probabilidad a posteriori efectiva anterior.

El problema de estimar esta densidad a posteriori se ha venido resolviendo usualmente mediante el filtro de Kalman o mediante las redes bayesianas temporales.

### 2.2 Filtro de Kalman

El filtro de Kalman se ha usado con gran éxito para realizar una estimación temporal en los casos en que $x$ es un vector en un espacio de parámetros continuo y la verosimilitud $p(z|x)$ tiene una distribución gaussiana. Este filtro estima de forma iterativa la media y la varianza de las $x$ en cada instante de tiempo y esto sirve para caracterizar perfectamente la distribución $p(x|z)$.

La aplicación del filtro de Kalman al seguimiento de características del entorno es problemática, debido fundamentalmente al problema del aliasing del sensor que provoca que distintas configuraciones del entorno $x$ produzcan parecidas lecturas del sensor $z$. Esto hace que la densidad $p(z|x)$ sea mucho más complicada que una gaussiana, siendo en muchas ocasiones multimodal. Los entornos dinámicos y el ruido en las lecturas tampoco son bien soportados por esta técnica.

### 2.3 Redes bayesianas temporales

Las redes bayesianas temporales se utilizan como una alternativa al filtro de Kalman, ya que permiten representar distribuciones de densidad arbitrarias. Para representar la distribución $p(x|z)$ se discretiza
el espacio de parámetros $X$ en un conjunto de estados y se utiliza la regla de bayes para propagar sus probabilidades.

Si llamamos $s_i$ a cada uno de los estados que se definen en $X$ y suponemos que en el instante $t$ se ha realizado la lectura $z_i$ después de realizar la acción $a_{i-1}$, las probabilidades de los estados $s_i$ se actualizan iterando la expresión

$$ p(s_i) \leftarrow p(z_i|s_i) \sum_{s_j \in X} p(s_i|s_j, a_{i-1}) p(s_j) \quad (9) $$

para todos los $s \in X$. Después de actualizar las probabilidades asociadas a cada uno de los estados, éstas se normalizan para asegurar que

$$ \sum p(s_i) = 1. \quad (10) $$

El problema fundamental de las redes bayesianas es la necesidad de discretizar el espacio de parámetros, lo que añade incertidumbre e introduce una complejidad espacial (y temporal al realizar la propagación) exponencial con el número de parámetros.

### 3.1 Algoritmo CONDENSATION

El algoritmo CONDENSATION proporciona una representación explícita de la densidad a posteriori mediante un conjunto de $N$ muestras ($m_1, \ldots, m_N$) y sus probabilidades asociadas ($\pi_1, \ldots, \pi_N$). Estas muestras se generan en cada instante de tiempo a partir de las muestras del instante anterior y de las nuevas mediciones del entorno ($z_i$) y las nuevas acciones realizadas por el robot ($a_i$).

Inicialmente, el conjunto de muestras se distribuye uniformemente en el espacio de parámetros $X$. Posteriormente, en cada instante de tiempo, se actualizan las $N$ muestras de la siguiente forma. Se escoge una muestra $m_i$ con probabilidad $\pi_i$ a partir del conjunto de muestras anteriores. Esta muestra es una muestra correcta de la distribución $p(x_{i-1}|z_{i-1}, a_{i-1})$. A partir de ella se realiza una proyección del modelo del sistema según la acción $a_i$ realizada por el mismo, obteniendo la nueva muestra $m'_i = p(x_i|x_{i-1} = m_i, a_i)$. Por último, como probabilidad asociada a $m_i$ se calcula su verosimilitud con las lecturas realizadas, mediante el modelo de probabilidad del sensor, $\pi_i = p(z_i|x_i = m_i)$.

### 4 Modelo de movimiento y de sensor

Para aplicar cualquier propagación temporal es necesario formular correctamente tanto el modelo de acción del sistema como el modelo del sensor.

#### 4.1 Modelo dinámico del robot

Esta sección describe las ecuaciones de movimiento fundamental de un robot *synchro-drive*, modelando su posición $x(t)$, $y(t)$ y su orientación $\theta(t)$ mediante las siguientes ecuaciones

$$ \theta(t_n) = \theta(t_0) + \int_{t_0}^{t_n} \omega(t)dt \quad (11) $$

$$ x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cos \theta(t)dt \quad (12) $$

$$ y(t_n) = y(t_0) + \int_{t_0}^{t_n} v(t) \sin \theta(t)dt \quad (13) $$
Robótica

Figura 3: Simulación de las lecturas de un anillo de sensores de ultrasonidos.

Las ecuaciones anteriores pueden simplificarse si se asume que el robot se mueve de forma discreta, con intervalos de control de tiempo $\Delta t$ en los que la velocidad lineal es $v_t$ y la angular $\omega_t$. Esta suposición es asumible cuando los intervalos de muestra de las velocidades y los datos del sonar no es demasiado grande en comparación con la velocidad del móvil. En nuestro caso, se han realizado todas las pruebas con el móvil simulado moviéndose a 25 cm/s y se han tomado lecturas cada 0.25 segundos.

La generación de muestras de $p(x_t|a_t = x, a_{t-1} = a)$ es directa a partir del modelo dinámico anterior. Sólo hay que aplicar el cambio de posición y orientación del robot sufrido tras sus movimientos $v_t$ y $\omega_t$ a los parámetros $x$ que definen las características. Además introducimos en la distribución un ruido gaussiano para modelar la incertidumbre en la posición real del robot. Este ruido gaussiano es fundamental a la hora de dotar al modelo de robustez.

4.2 Modelo del sensor

El modelo del sensor nos proporciona la verosimilitud de unas lecturas $z$ obtenidas en una hipotética configuración del entorno $x$. Las lecturas $z$ son distancias a los obstáculos más próximos en la dirección definida por los sensores de ultrasonidos distribuidos alrededor del robot. En nuestros experimentos hemos utilizado una configuración de 24 sensores simulados, con lo que $z = (l_1, l_2, \ldots, l_{24})$.

El enfoque para el cálculo de la verosimilitud ha sido similar a las propuestas de reconocimiento a partir de sensores [8]. Se ha construido un modelo realista del sensor de ultrasonidos, capaz de simular correctamente la mayor parte de características de las lecturas reales, incluyendo las que han sido despreciadas hasta el momento como fuentes de error, como los múltiples rebotes. Estos errores no pueden modelarse correctamente suponiendo un modelo gaussiano del ruido.

Para implementar el modelo se ha realizado una modificación del clásico algoritmo de trazado de rayos, adaptándolo a superficies 2D y al cálculo de las distancias de alcance [7]. Un ejemplo del resultado del algoritmo de simulación puede verse en la figura 3.

Para calcular la verosimilitud $p(z|x)$ se simulan hipotéticas lecturas obtenidas por la característica en la configuración $x$ y se comparan con las lecturas reales, utilizando una medida de distancia.

Formalmente, la función de verosimilitud se expresa como

$$p(z = (l_1, l_2, \ldots, l_{24})|x) = e^{-d(0, k)^2/t^2},$$

(14)

siendo $l_i$ las lecturas reales del sonar y $l_{24}$ las lecturas simuladas producidas por la característica $x$. En los experimentos posteriores, la función $d$ se ha implementado como una distancia euclidiana entre las posiciones finales de las lecturas.

5 Resultados

Para comprobar la técnica propuesta se han realizado un serie de experimentos en los entornos simulados que aparecen en la figura 4. En todos los experimentos el robot se mueve evitando obstáculos a una velocidad de 25 cm/s y realiza una lectura de sensores cada 0.25 segundos. Cada vez que se realiza una lectura se ejecuta un paso del algoritmo CONDENSATION. La velocidad del algoritmo es aceptable, funcionando a 1/4 del tiempo real con $N = 300$ muestras en un procesador Pentium II. Futuras optimizaciones del código harán posible la ejecución del algoritmo en tiempo real.

En los siguientes apartados veremos ejemplos del funcionamiento del algoritmo en distintos instantes de tiempo. Dibujaremos las muestras gene-
Figura 4: Algunos de los entornos de prueba en los que se han realizado los experimentos. El entorno 1 consiste en un pasillo con dos obstáculos y el 2 un pasillo con múltiples puertas.

radas por el algoritmo sobre el entorno real, y cada muestra tendrá un tono de gris proporcional a su probabilidad.

5.1 Inicialización

En la figura 5 se puede observar el proceso de inicialización del conjunto de muestras de la característica pasillo. Recordemos que cuanto más oscura es la muestra mayor probabilidad asociada tiene. El número de muestras utilizadas es \( N = 300 \). El tiempo de una instantánea a otra es de 1 segundo.

5.2 Seguimiento de pasillos con obstáculos

En la figura 6 se puede ver la continuación de la situación anterior. Una vez centrado el conjunto de muestras alrededor del pasillo real todas las muestras bajan en verosimilitud al pasar el robot frente a un obstáculo (instantánea 8). En la instantánea 9, el ruido gaussiano del modelo de movimiento genera algunas muestras de pasillos más cercanos al obstáculo, pero la media de la distribución no cambia de forma sensible. En las instantáneas 10 y 11 el robot superado el obstáculo, vuelven a realizarse lecturas del pasillo real y la distribución se mueve otra vez hacia el pasillo real.

5.3 Seguimiento de finales de pasillos

En la figura 7 se comprueba el funcionamiento del algoritmo siguiendo finales de pasillo en un entorno complicado como el número 2. La dificultad de este entorno se debe a que el robot no obtiene ninguna lectura del pasillo cuando pasa frente a las puertas. El número de muestras de este experimento es el mismo que el anterior (\( N = 300 \)). En la instantánea 1 se ven las muestras ya inicializadas. De la instantánea 1 a la 9 el robot pasa frente a diversas puertas, con lo que las muestras han evolucionado según el modelo dinámico del robot, sin ser reforzadas por lecturas del sonar. Sin embargo, cuando el robot vuelve a detectar el pasillo vemos como en las instantáneas 10 y 13 el algoritmo vuelve a reforzar las muestras correctas.

6 Conclusiones

Se ha presentado la aplicación del algoritmo CONDENSATION al problema del seguimiento de características topológicas a partir de datos de sensores de ultrasonidos. Este algoritmo permite estimar una distribución a posteriori arbitraria en un espacio de parámetros continuo, resolviendo las deficiencias de los métodos empleados usualmente como son el filtro de Kalman o las redes bayesianas temporales.

Se han mostrado resultados experimentales realizados sobre entornos simulados en los que se comprueba la robustez y estabilidad de las características extraídas.

Referencias

[1] Billur Barshan and Roman Kuc. Differentiating sonar reflections from corners and


Figura 5: Inicialización de la característica *pasillo* en el entorno 1.
Figura 6: Seguimiento del pasillo moviéndose el robot en el entorno 1. El obstáculo puede verse como un segmento recto paralelo al pasillo.


Figura 7: Seguimiento de finales de pasillo moviéndose el robot en el entorno 2.
Robòtica

Programació i Aprenentatge en Tasques Robòtiques Reals de Manipulació*

Enric Cervera, Angel P. del Pobil
Departament d'Informàtica
Universitat Jaume I, Campus de Penyeta Roja
12071 Castelló
{ecervera, pobilia}@inf.uji.es

Resum

La programació de robots dins d'entorns amb incertesa esdevé molt complexa. Es necessiten sensors per a obtenir informació sobre l'estat real del món. Per a adaptar-se a noves situacions, els robots han d'aprendre d'exemples o de la seua experiència. L'aprenentatge pot ser més ràpid si hom utilitza el coneixement a-priori disponible. En l'arquitectura basada en sensors que es proposa, es combina l'aprenentatge amb mòduls programats que representen el coneixement injectat al sistema. S'apriú la correspondència entre estats i accions qualitius. La programació reduceix la complexitat de la tasca d'aprenentatge. El processament numèric i qualitatiu s'integren de manera apropriada per a un ampli espectre de problemes. S'inclouen exemples d'una tasca real d'inserció, on els sensors de força i la posició relativa són informació suficient per a aprendre la tasca. D'aquesta forma, la solució és vàlida per a qualsevol posició de l'objectiu.

Paraules clau: robòtica, aprenentatge, tasques reals.

1. Introduction

While modern robots are capable of performing a broad range of tasks, the required effort to program them to do the desired tasks has been vastly underestimated. If there is any uncertainty in the motion or in the world model these hard coded robot programs will fail miserably. The robot needs to be endowed with sensors for perceiving its environment, adapting to changes, and reducing the uncertainties [1]. Additionally, the robot should be capable of adapting to new situations by learning from examples or from its own experience. Current learning approaches are limited to simulated problems or simple real tasks. In order to achieve complex real-world tasks, the system should not ignore the available task knowledge. The objective of this work is to build complex systems using force sensing, task knowledge and learning capabilities inspired on biological systems. The desired architecture should get benefits from both programming and learning.

2. Uncertainty makes it difficult

In the absence of feedback from the environment, a robot program must rely only on its internal programmed model. The quality of this model is the key factor for the success of the robot operation, but precision tasks are very sensitive to small inaccuracies. For example: if the location of the elements for an insertion task is known with an error greater than the clearance between the elements, the model cannot guarantee that the task will be successfully accomplished.

In a broad sense, the problem is to enable the robot to perform tasks despite its uncertain position relative to external objects. The use of sensing significantly extends the range of possible tasks [2]. Initially, persons rely heavily on their senses, primarily vision and tactile, to accomplish a certain task. However, upon proficiency, the task becomes mechanistic and routine.

A robotic system should increase its skills in a similar way: beginning with simple exploratory actions, and upon completion of successive trials, it should incorporate the learnt knowledge to its strategy, thus becoming more skillful.

A learning approach for robot tasks is proposed in [3]. However, learning from scratch is inefficient in all but the simplest tasks. Advanced control techniques should be considered as primitive elements for such a

skilled system: for example, the compliant motion techniques [4], or the use of vision in motion control as studied in [5].

3. A hybrid approach

The proposed method is a framework which integrates robot programming, sensing, and learning in a modular architecture.

A-priori task knowledge should be programmed or hard coded, while learnt knowledge is acquired by sensing and experience. The a-priori knowledge must take into account all of the system uncertainties. As the system learns and becomes more skilled, its uncertainty decreases progressively.

The nature of the a-priori knowledge is not restricted. Qualitative reasoning is a natural way of integrating sensory information and common-sense knowledge in the system. If a-priori knowledge is very sparse, exploration techniques should be considered. In the absence of precise guidelines, random exploration has been proven to be a valid choice for real tasks [6]. The system architecture is depicted in Fig. 1. In the application to robotic manipulators, force and position sensors are considered. Modules are described below in more detail.

3.1 Signal processing

Raw sensor signals can be used to generate the state of the system directly. In the general case, filters are necessary for extracting the relevant information of time-dependent data and reducing high-frequency noise.

3.2 State identification

There are three steps in this phase (Fig. 2): feature extraction, discretization and, optionally, a finite state automaton (FSA).

The relevant features of the signals are not necessarily their absolute values (e.g., filtered values from noisy signals). If there is some a priori information about the important features, it should be used at this level. The appropriate extracting algorithms, either fixed or adaptive, should be incorporated, in order to facilitate the work of the following levels. In a previous work [7, 8] the authors demonstrated the feasibility of unsupervised learning algorithms for extracting relevant feature information from sensor data in robotic manipulation tasks.

The feature vector should be converted into qualitative information before further processing. This is a requirement for the proposed learning algorithm which is suited for discrete state and action spaces. This conversion can be achieved by defining sets in the feature space, and assigning labels to each set. The label of the set which contains the feature vector is chosen. If no task knowledge is available, a random partition can be chosen, or an unsupervised clustering algorithm can be applied [7, 8].

In the simplest case, the qualitative information obtained from the feature vector is considered as the
system state. If past history is important, a FSA can be defined based on task knowledge. Its alphabet is the set of qualitative labels. Its states and transitions are defined by the high-level qualitative dynamics of the task. The state of the system is determined by the state of this automaton upon processing of the qualitative input.

3.3 Programming vs. learning

The system generates an output based on its current state. This output can be programmed for a given state, or it can be learned from the experience, by means of a reinforcement process, which indicates the quality of the actions performed by the robot at each state. In this case, the Q-learning algorithm [9] is proposed, which learns the state-action pairs which maximize a scalar reinforcement value or reward. Initially, the system randomly explores the action space. As it becomes more experienced, it learns the correct association between each state and the best action for such state.

3.4 Motion translation and compliant control

Only discrete actions are issued by the learning algorithm. A look-up table is used to translate this discrete actions to robot motions (Cartesian velocities). Discretization is usually considered a major drawback for real tasks. However, in our experimental results, it is shown that only eight different actions are sufficient for successfully performing a real insertion task. Well-known algorithms are used for position control, force control, compliant motion [4], etc. A separate control law can be used for each distinct qualitative action.

3.5 Kinematic modules

The rest of the modules are programmed with the robot kinematic equations; this model allows to extract the Cartesian location and the Jacobian matrix from the robot joint angles, which are measured by optical encoders. Since our task domain is manipulation and fine-motion tasks, it is assumed that the robot kinematic model exists and is suitable for these computations.

4. A real-world example

An example of dexterous task is the insertion of a peg into a hole with low clearance and uncertainty in the location (see Fig. 3). Two shapes are considered: cylinder and cube. Since the uncertainty is higher than the clearance, a pure position control cannot achieve the task. A force sensor is attached to the robot wrist. It senses the force and torque vectors applied to the end-effector. Three degrees of freedom (translations) are considered for the cylinder, and four (rotation around z-axis) for the cube. This problem is difficult to model analytically and exemplifies the trouble raised by uncertainty in real-world control problems.

First, a high-level qualitative plan is derived (Fig. 4). It should give insight for the design of the control architecture. This plan is a formal specification of the task, and consists of the following steps:

1) Approach hole until contact with surface.
2) Move compliantly on the surface towards the hole until insertion.
3) Move compliantly downwards into the hole until contact with bottom.

Surface contact is detected simply by a threshold in the z component of the force signal. Compliant
motions are hybrid control motions. Force control is used for some degrees of freedom and position control for the other degrees [4]. In the 2nd step, $z$ is force-controlled and $(x, y)$ are position-controlled. In the 3rd step, $z$ is position-controlled and $(x, y)$ are force-controlled.

Location uncertainty is modeled as an error ball. If the actual location of the system is given by the point $\mathbf{x}$, then the sensor will return a sensed value $\mathbf{x}_s \in B_\varepsilon(\mathbf{x})$, where $B_\varepsilon(\mathbf{x})$ is the sphere of radius centered at $\mathbf{x}$. One should note that the clearance (difference between hole and peg width) is much smaller than the uncertainty radius, but the hole depth is much greater than this radius. Consequently, the location of the hole cannot be detected due to uncertainties in the coordinates $(x, y)$, but the insertion of the peg can be detected with the value of the $z$ coordinate. The key problem is how to move towards the goal with position uncertainty. A random walk is one possible choice [6, 10], but a sensor-guided strategy is likely to be more efficient. Physical studies suggest that torque signal provides enough information about the direction to the hole [11]. However, when using real sensors, analytical solutions are quite difficult to derive. Thus, a learning scheme will be introduced at this stage.

Based on all the previous task knowledge, our proposed architecture is now applicable to this problem. Each independent module is described in the following.

4.1 Sensor signals

The system inputs are the six force and torque signals from the wrist sensor, and the joint angles which are measured by internal encoders.

4.2 Signal processing

Sensor measurements are processed by a second-order low-pass Butterworth filter with cutoff at 4 Hz (the sampling rate is 140 Hz).

The encoder values are used to calculate the joint angles. Since the motion commands are given in Cartesian coordinates, the Jacobian matrix has to be determined from the joint angles. The Jacobian matrix will only be used by the control submodules to convert the Cartesian commands to joint velocities.

4.3 State identification

Task knowledge states that only the uncertainty in translation along $x$ and $y$ axes is important in our setup. Contact information is extracted from the reaction force (nearly parallel to the $z$ axis). Depending on the point of contact, a different torque will be sensed. Thus the features extracted from the processed sensor signal are the current values of $F_z, M_x, M_y$. Additionally, $M_z$ is also considered in the cube example.

Numerical values are converted to descriptive symbol labels by means of selected thresholds. Two categories are defined for $F_z$ (contact and no-contact), and from three to five categories for each component of the torque. No effective method for threshold selection exist, so the selection is ad-hoc (see section 5).

The symbolic label and the previous state are used to determine the current state. The programmed task plan implicitly defines a simple FSA. When learning takes place, the state is directly identified by the discretized values of the sensor signals.

4.4 Programming and learning

The high-level programmed plan can solve the task, but the learning process dramatically improves the performance of the system if, when the peg is contacting with the surface, perception is used to move the peg towards the hole. The relationship amongst contacts and actions is learned by means of a reinforcement algorithm, namely Q-Learning [9] (Fig. 5). A discrete number of actions is chosen, i.e. different directions of motion. Each pair $(\text{State}, \text{Action})$ or Q-value is initialized to zero. The learning algorithm updates these values according to the reinforcement signal obtained when each action is executed in a given state. A simple action-penalty representation is used [12]. The system is penalized with the same value for every action that it executes. The method converges to the values which minimizes the accumulated penalization (which will correspond to the minimum insertion time). Initially, the system chooses actions randomly. After learning, the action with higher value for the current state is chosen.

4.5 Action translation and control

![Fig. 5. State-action learning.](image-url)
Each action is translated to a Cartesian velocity command with a programmed compliant behavior, i.e., some degrees of freedom are position-controlled and other degrees are force-controlled, as stated in [4]. For example, motions on the surface are z-compliant, and the inserting motion is xy-compliant.

5. Experimental results

The system has been implemented in a Zebra Zero robot arm with a wrist-mounted force sensor. The task is the insertion of a peg into a hole. Two shapes are considered: cylinder and cube. The cylinder is 29 mm in diameter, while the hole is chamfered and 29.15 mm in diameter. The clearance between the peg and the hole is 0.075, thus the clearance ratio is 0.005. The peg has to be inserted to a depth of 10 mm into the hole.

The controller is trained in a sequence of trials, each of which starts at a random position within the uncertainty ball. The uncertainty radius is 3 mm, and the exploration area is a 5 mm square, centered at the starting position.

The discrete qualitative value of $F_z$ is used to detect a surface contact by means of a threshold. For the cylinder, four signals are used: $M_x$ and $M_y$ from the force sensor, and the relative $x$ and $y$ with regard to the initial location. These features do not depend on the absolute position or orientation of the target. Force signals are discretized in 5 values, while location ones are discretized in 3 values. The number of states is 75. The action space is discretized and only eight directions of motion across the XY-plane are allowed: from 0 to 315 degrees with steps of 45 degrees. For the cube, two additional signals are used: $M_z$ and the relative z-angle with regard to the initial orientation. Since the number of signals has increased, only three discrete values are considered for each signal in order to limit the curse of dimensionality. The number of states is 729. Ten actions are considered, the eight previous translations and two rotations around the z-axis. In [13] it is suggested that qualitative reasoning may be an effective means by which humans understand the sensory force information they receive during a manipulation task.

The peg approaches the surface until a contact is detected. Next, a sequence of compliant motion steps across the surface is executed. The action at each step is determined by an exploration scheme. Initially, it relies on random choices, but progressively the learnt values are used to select the action. The z-motion is determined by a compliant controller.

After the execution of each step, the learning algorithm updates the Q-value associated with the starting state $i$ and executed action $a$ as follows:

$$Q_{s+1}(i, a) = (1 - \alpha) \cdot Q_s(i, a) + \alpha \cdot \left[ r + \gamma \cdot V_{Q_s}(j) \right]$$

where $j$ is the successor state reached, $r$ is the immediate cost or reinforcement value, $\alpha$ is a learning rate, and $\gamma$ is a discount factor ($\gamma=0.99$). $V_{Q_s}(j)$ is the value of state $j$, i.e., the largest value of its actions. In the action-penalty representation, the system is penalized for every action it executes ($r = -1$ always). Since reinforcement-learning methods determine policies that maximize the total reward, the system will learn to insert the peg in the minimum possible number of steps.

The search phase ends up when the hole is detected or it is aborted after a time limit (20 seconds). The detection is achieved by a threshold in $F_z$ (loss of contact). A motion along the z-axis is performed until a contact in $F_z$ is detected again. Despite the low clearance, no problems of jamming or wedging arose in the experiments, possibly due to the passive compliance of the gripper and the inherent vibration of the process.

Experimental results for the cylinder are shown in figures 6 and 7. The critical phase is the surface-compliant motion towards the hole. The system must learn to find the hole based on sensory information.

Fig. 6 depicts the insertion time versus the number of trials. A moving average window of 100 trials is used in all the plots. It should be taken into account that failures are given a time of 20 seconds. In addition, the system can perform about 40% of the insertions at the beginning only with random motions. But the learning process improves dramatically the performance of the system. In fig. 7 the probability of success is shown. After 1000 trials, there is a steep increase, and this probability approaches 1. During
the last trials, 97% of successful insertions are achieved in less than 20 seconds. The mean insertion time for the last 500 trials is 7.8 seconds and the standard deviation is 3.2. It must be noted that the nominal velocity of the end-effector is kept slow (5 mm/s) for security reasons. Our system takes more trials to train than that of [3], but one should note that our trials are time-limited, besides other differences like velocities and sensing times. In addition, our system only uses relative information regarding the location, thus making possible to use the same controller for any position or orientation of the target. The complete learning process took about 20 hours of dedicated robot work.

Results for the cube are depicted in Fig. 8. Intuitively this is a difficult task due to the lack of symmetry, and this is confirmed by the low probability of insertion (about 15%) with random motions during the first trials. The increased difficulty slows down the learning process and the final results. It takes 4000 trials to reach a 66% success rate, and it does not improve further. We suspect that the architecture is suitable, but the system lacks the necessary information for solving the task. New features are needed in order to achieve better results for the cube and, in general, for non-cylindrical shapes.

Experimental results show that the system learns incrementally from a random strategy, by improving its performance based on its own experience. A significant amount of task knowledge is embedded in the architecture, which simplifies the learning problem. However, the imprecision, noise and inherent difficulties of a real robot are dealt with a discrete learning algorithm.

6. Conclusions and future work

A structured robot control architecture has been introduced. Sensor processing, task programming, learning from experience and qualitative knowledge are the key elements of the system. The modular structure provides a clean integration of the different paradigms. The goal applications are those tasks which cannot be fully programmed due to uncertainties and incomplete knowledge. The proposed scheme differs from other learning approaches in that it clearly states the difference between previous knowledge (programmed) and learned knowledge (association amongst states and actions). The qualitative treatment of information makes it suitable for the analysis of system behavior, knowledge extraction and generalization to other more complex tasks. Future research includes the study of the convergence and stability of the algorithm, and how to improve the results for non-cylindrical shapes.

Acknowledgments

This paper describes research done in the Robotic Intelligence Laboratory. Support for this laboratory is provided in part by the CICYT under project TAP95-0710, by the Generalitat Valenciana under project GV-2214/94, by Fundació Caixa-Castelló under P1A94-22, and by a scholarship of the FPI Program of the Spanish Department of Education and Science.

References


Refinement of Environmental Maps Obtained from Autonomous Robot Exploration

Maite López-Sánchez and Ramón López de Mántaras
Institut d'Investigació en Intel·ligència Artificial, IIIA-CSIC
Campus UAB, 08193 Bellaterra, {maite, mantaras}@iiiia.csic.es

Abstract
This paper presents some improvements of a previous environment mapping work. On one hand, a previous restriction about the environment orthogonality is relaxed and, on the other hand, an extension of detected segments is done in order to increase the environment coverage. Finally, results in different environments illustrate how this map extension yields to safer paths when planning over the extended map instead of over the non-extended one.

Key words: autonomous robots, map generation, planning.

1 Introduction
The background of this paper is a previous mapping work [5] which consisted of two different steps: first, a group of small autonomous robots explore an unknown office-like orthogonal environment, and second, a host computer receives exploration information from the returning robots and uses it to build a map. Robots explore moving randomly in clear areas and following walls (or obstacle edges) when detected by their infrared (IR) sensors. The computer host models the environment generating a grid map in terms of degrees of possibility and necessity [2] of the position of the detected walls and obstacles. Since a total coverage cannot be guaranteed, the host also computes the shortest path between two positions so that a robot can use it as a guide to reach a less explored area by combining the path information with reactive behaviour.

The paper describes two improvements of the mapping: one is that the assumption about the orthogonality of the environment is partially discarded and the other is that detected segments are enlarged on the basis of implicit map information. This map extension process increases the map coverage of the real environment and gives more conservative and informed paths that reduce the robots use of reactivity when following them.

2 Non-orthogonal environments
Our mapping process only considers environmental features with edges long enough to be followed by the robots. Small obstacles, such as chair or desk legs, are not represented because they generate spurious IR sensor readings that are considered as noise. This does not represent an important disadvantage because small obstacles can be easily avoided using a non-informed reactivity. That is, information about a small obstacle is not crucial due to the fact that whether the robot surrounds it by its left or by its right it will not imply a significant change in the global performance of the robot.

Therefore, considering only relatively-large environmental features, we can assume that office-like environments are highly orthogonal. On the one hand, walls are usually connected by right angles. And on the other hand, human made objects in an office such as bookshelves or drawers tend to have rectangular shapes.

In case a robot follows a wall or an edge obstacle without an exactly parallel trajectory, the resulting detected segments will not be orthogonal. The orthogonal assumption presents the advantage that it can be used to correct those non-orthogonal segments. Nevertheless, half opened doors appear so often in office-like environments that the mapping process has been refined to consider non-orthogonal features (the host only orthogonalizes those segments that are almost vertical or horizontal and keeps the oblique segments as they are). This relaxation of the orthogonal assumption means a significant improvement to the mapping process because, as Yamauchi points out in his recent work [10], it is still quite common to find in the literature systems that can only handle parallel or perpendicular walls.
3 Wall segment extension

Robots describe trajectories that are sequences of turns and straight movements. Trajectories are thus represented as adjacent line segments. When a robot detects a wall or obstacle, it follows it along a random distance and marks the trajectory segment so that the host can compute the corresponding wall segment.

Following a wall along a random distance implies that very often the robots leave the wall that they are following before reaching its end. This is done in order to increase the number of discovered features and to avoid that robots keep forever following the contour of a feature once it is detected. Thus, in general, random distances force robots to leave walls before reaching their end-in order to look for other detections- but when the end is reached, the corresponding end point of the segment is labelled as "singular point" to mean that there is a corner or a door.

The host computer discretizes the environment in a two-dimensional grid where each cell has a degree of possibility and necessity of being occupied. Trajectory segments give information of free space (with necessity equal to zero and possibility less than one) and wall segments correspond to occupied cells (necessity is positive and possibility is one). Since robots have odometry errors-they do not know their positions with accuracy-, these values are not assigned to single cells corresponding to a position but to rectangular areas whose sizes grow proportionally to the error (which grows in proportion to the robot displacement).

In this way the host builds a map having all detected wall segments and robot trajectory segments in a reliable but limited manner. Actually, wall segments without singular points correspond to longer walls-or obstacle edges- because robots leave them before they find their ends. Unfortunately, there is not enough information in the map to know how long real walls are but some criteria help in limiting the extension of wall segments: since trajectories represent free space, they are used as upper extension bounds. Furthermore, it seems reasonable to stop extending a wall segment when it meets either a detected wall segment or another segment extension. In these cases, if they have the same direction they will be considered as being part of the same wall, while if they have perpendicular directions they can be taken as corners and therefore labelled as "singular points". Extension is done locally by propagating small constant certainty values of been occupied for those cells in the segment ends. These small certainty values reflect that they are just assumptions and do not correspond to actual robot detections.

The host only extends orthogonal wall segments. Oblique ones usually correspond to doors-which are not longer than the minimum random following distance- or unusual objects- whose shape is more difficult to predict.

Wall extension has two main advantages: first, it increases the coverage of the real environment and second, the planning [8] over the resulting maps give more conservative and safer paths than the ones obtained considering just detected features. These paths are computed from a visibility graph of the free space and guide robots towards less explored areas, reducing the robot use of reactivity when following them.

4 Results

With the aim of testing the coverage of an unknown environment, five robots have been sent to explore it (see Fig 1). Since the host builds its map (see Fig 2) in an incremental way, the data in Table 1 shows the environment coverage each time one more robot delivers its exploration information to the host. Columns show the percentage of the coverage of the walls in the environment—which sum up 30.4 meters, first column stands for detected wall segments, second for correctly extended wall segments and the third represents the incorrect extensions.

<table>
<thead>
<tr>
<th># robot maps</th>
<th>Detection coverage</th>
<th>Extension coverage</th>
<th>Incorrect extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.3%</td>
<td>32%</td>
<td>1.97%</td>
</tr>
<tr>
<td>2</td>
<td>28.2%</td>
<td>48.99%</td>
<td>1.97%</td>
</tr>
<tr>
<td>3</td>
<td>44.6%</td>
<td>58.7%</td>
<td>0.4%</td>
</tr>
<tr>
<td>4</td>
<td>58.8%</td>
<td>78.1%</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>65.2%</td>
<td>82.1%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: percentage of the environment coverage.

Figure 1: five robots explore an unknown environment (which is 5.3m x 3.8m)
These results considers all the walls in the environment; nevertheless two walls of the upper right room are unreachable because of its almost closed door. Hence, the percentages can be recomputed to obtain a reachable wall coverage of 74.45% for detection and 93.88% for extension.

Figure 2: extended map: wall detections in medium grey, trajectories in dark grey, singular points in very light grey and wall extensions in light grey (darker scales mean more certainty).

Figure 3: Paths planned considering: a) a non-extended map and b) an extended map.

The extension process is purely based on the application of heuristics and, therefore, it is hard to evaluate how it helps in the coverage of the environment. This process is highly dependent on the environmental structure as well as on detections that result from random explorations. In order to further illustrate the improvement, we have applied the same heuristics to a different environment (shown in Fig 4). In this case, the detections of four robots are shown in Fig 3a and they cover 45.3% of the total walls. Fig 3b contains the resulting extensions, from which 70.1% are correct and 6.4% are incorrectly extended.

This second example has been also used to compare the planning in extended and non-extended maps. Fig 3a shows planned paths in the non-extended map and Fig 3b the paths in the extended map. Notice that although paths labelled with the same number have the same initial and goal position, they describe different trajectories because they consider different maps.

The three images in Figure 4 depict the trajectories actually followed by the robots –some of them present common trajectories. Obviously, in areas with the same information both perform equally (see path p1 and p2). When one map is more accurate than the other, incorrectly extended information yields to paths longer than necessary, while non-detected walls imply a high use of reactivity.

Sometimes, an incorrect extension closes narrow gaps that require the use of reactive capabilities to get through. In that way, an initially shorter path in the non-extended map can result in a trajectory whose length is in fact comparable to the length of a trajectory from the extended map (this is the case of p3 and p3e). Of course, this is not always the case: p4e avoids an actually non-existing piece of wall causing an extra displacement of 0.37m. (p4 is a 6.45m long trajectory). The worse case of avoiding a non-existent wall is p5e, which turns around the corner where a previous robot went. Although this tendency of going over previous robot trajectories can be a disadvantage, already explored free space is also the source of safer paths.

Finally, when extension is correct, resulting paths are more informed and therefore reactivity is less often needed (see p6 and p6e). Reactivity is less efficient because it causes the robot to decide how to avoid an obstacle without knowing its shape. This can yield the problem that appears in p7, in which the reactivity makes the robot to take so many wrong decisions that the robot stops before reaching the target (the robot position error grows so much due to the many displacement movements that the area it covers already includes the target coordinates even when the robot is still quite far from the target).
and assume known distributions, Fuzzy Set theory is a good alternative when these conditions are not met. For example [3] uses fuzzy numbers to model the uncertainty of the parameters of geometric primitives and of co-ordinate transformations used to describe natural environments and [9] uses fuzzy sets to assign to each point in the grid map a degree of being empty and a degree of being occupied. This last work is the closest to ours, but our use of proximity infrared sensors in short distances gives better results than their imprecise information coming from ultrasonic sensors. In fact, most of the works mentioned here use ultrasonic sensors for obstacle detection. These sensors have problems that increase the location error due to the wide radiation angle and false reflections, and furthermore, the ultrasonic beam can easily be lost due also to reflections when hitting obstacles with a wide angle of incidence.

Regarding the coverage percentage, it is not possible to compare our results with the rest of works because they establish their own settings and they do not provide enough data.

References

Fulls per anotar
Fulls per anotar
Fulls per anotar
ASSOCIACIÓ CATALANA D'INTEL·LIGÈNCIA ARTIFICIAL

Adhesió Particular
Nom: Cognoms:
Afiliació:
Adreça:
Nº de Telèfon: Nº de fax:
Adreça electrònica:
☐ Adhesió simple: ......................................................... 5.000 Pts.
☐ Adhesió estudiant: ......................................................... 2.000 Pts.

Adhesió Entitat
Organisme:
Nom del representant: Cognoms:
Funció:
Adreça:
Nº de telèfon: Nº de fax
Adreça electrònica:
☐ Adhesió: ................................................................. 20.000 Pts.

Presentat pels socis
Nom soci 1: Signatura:
Nom soci 2: Signatura:

☐ Accepto que les meves dades apareguin en el butlletí de l’ACIA
☐ Accepto que les meves dades siguin comunicades a l’ECCAI per a la
constitució d’una base de dades europea d’Intel·ligència Artificial.
☐ M’oposo a qualsevol difusió de les meves dades personals.

Enviar aquest full a: Miquel Sánchez, Tresorer de l’ACIA, LSI, UPC, Pau Gargallo, 5. 08028 Barcelona,
Tf.: (93) 401 7334, Fax.: (93) 401 7014, E-mail: miquel@lsi.upc.es

Dades bancàries
Autoritzo l’Associació Catalana d’Inteligència Artificial a presentar rebuts amb càrrec al meu compte bancari.
Banc/Caixa: Adreça:
Entitat: Oficina: Codi control: Nº Compte:
Signatura: