Rovira i Virgili University

Master on Artificial Intelligence

Background knowledge to improve induction of decision structures in medicine

Author:
Joan Albert López Vallverdú

Advisor:
Dr. David Riaño

June 3, 2008

This work has been carried out within the Research Group on Artificial Intelligence (Banzai) of the Rovira i Virgili University.

I am greateful for the support received from Maria,
this thesis is dedicated to her.

Abstract

In this tesis, an analysis is carried out on which are the medical criteria involved in a decision process and how they can be applied in the automatic generation of decision structures in medicine. The drawbacks of the current approaches to the induction of medical decision trees are discussed and solved with our proposed model. This model includes the background knowledge from the physicians which contains explicit criteria like economic cost of the tests and implicit criteria like the degree of adjustment to the common medical practice. These criteria are represented by means of cost functions or partial orders and are integrated in a multi-criteria decision model which is used in the generation of the decision structures. The methodology has been tested on real data of some medical domains like diagnosis of diabetes, thyroid malfunctioning, heart diseases and management of post-operative patients obtaining more medically comprehensible decision trees.

Contents

1 Introduction **8**

2 The state of the art **10**
2.1 Optimization 10
2.2 Ordering 11
2.3 Decision 12
2.4 Cost functions 12
2.4.1 The sense of a cost function 12
2.4.2 The range of a cost function 13
2.4.3 The weight of a cost function 13
2.4.4 Combining cost functions 14
2.4.5 The cost function in the optimization problem 14
2.5 Partially ordered sets 14
2.5.1 Definitions 15
2.5.2 Layered partial orders 17
2.5.3 Transforming partial orders into cost functions 19
2.5.4 Transforming cost functions into partial orders 20
2.5.5 The weight of a partial order 20
2.5.6 Combining partial orders 20
2.5.7 Combination of partial orders through cost functions 22
2.5.8 Statistical combination of partial orders 22
2.6 Decision trees 23
2.6.1 Definitions 23
2.6.2 Using decision trees 24
2.6.3 Inducing decision trees 26
2.7 Modeling medical decisions 28
2.7.1 The information gain approach 28

2.7.2 The cost-sensitive approach . . . 31
2.7.3 The order-sensitive approach . . . 35
2.8 Evaluating medical decisions . . . 38
2.8.1 Evaluation of the decision structure . . . 39
2.8.2 Evaluation of the decision cost . . . 40
2.8.3 Evaluation of the decision order . . . 41

3 Methodology proposed 42
3.1 Decision model proposal . . . 43
3.2 Medical criteria . . . 45
3.3 Representation of criteria . . . 46
3.3.1 Criteria on the attributes . . . 46
3.3.2 Criteria on the decisions . . . 53
3.4 Combination of criteria . . . 58
3.4.1 N cost functions . . . 58
3.4.2 N partial orders . . . 59
3.4.3 N cost functions and M partial orders . . . 60
3.5 Medical decision . . . 61

4 Tests and results 66
4.1 Tests performed and evaluation measures . . . 66
4.2 Medical domains . . . 70
4.3 Diabetes domain . . . 71
4.3.1 The dataset and the background knowledge . . . 71
4.3.2 Results and analysis . . . 73
4.4 Heart disease domain . . . 79
4.4.1 The dataset and the background knowledge . . . 79
4.4.2 Results and analysis . . . 80
4.5 Post-operative domain . . . 87
4.5.1 The dataset and the background knowledge . . . 87
4.5.2 Results and analysis . . . 90
4.6 Thyroid domain . . . 95
4.6.1 The dataset and the background knowledge . . . 95
4.6.2 Results and analysis . . . 97

5 Conclusions and future work 103

Bibliography 105

List of Figures

2.1 Hasse diagrams corresponding to the partial order of the stages of breast cancer 17
2.2 Not layered partial order 18
2.3 Decision tree on the echocardiogram domain 24
2.4 Path followed by the patient s_1 on the echocardiogram domain 25

3.1 Scheme of the decision model 43
3.2 Main medical criteria 45
3.3 The partial order $\leq_c'$ 51
3.4 The preorder $\lesssim_c'$ 51
3.5 The partial order $\leq_c$ 52
3.6 The partial order $\leq_m$ 53
3.7 The partial order $\leq_{Ie-h}$ 55
3.8 The partial order $\leq_{IIe-h}$ 57

4.1 The partial orders $\leq_x$ for the diabetes domain 73
4.2 Results of the general test on the diabetes domain 75
4.3 Decision tree obtained for the diabetes domain (with pruning) 77
4.4 The partial orders $\leq_x$ for the heart disease domain 82
4.5 The partial order $\leq_{IIe-e}$ / $\leq_{IIe-h}$ / $\leq_{IIe-c}$ for the heart disease domain 83
4.6 Results of the general test on the heart disease domain 84
4.7 Decision tree obtained for the heart disease domain (with pruning) 86
4.8 The partial orders $\leq_x$ for the post-operative domain 89
4.9 The partial order $\leq_{Ie-h}$ for the post-operative domain 89
4.10 The partial orders $\leq_{IIe-x}$ for the post-operative domain 90
4.11 Results of the general test on the post-operative domain . . . 91

4.12 Decision tree obtained for the post-operative domain (with pruning) 93
4.13 The partial orders $\leq_x$ for the thyroid domain 97
4.14 Results of the general test on the thyroid domain 99
4.15 Decision tree obtained for the thyroid domain (with pruning) . 101

List of Tables

3.1 Tests needed for each attribute 49
3.2 The values of V_t and κ_t 49
3.3 The values of K_t 49
3.4 The values of W'_t 56
3.5 The values of IIe_t 57

4.1 Priority and relevance for attribute criteria 66
4.2 Priority and relevance type II error criteria 67
4.3 Priority and relevance for attribute criteria from the economic vision 68
4.4 Priority and relevance type II error criteria from the economic vision 68
4.5 Priority and relevance for attribute criteria from the medical vision 68
4.6 Priority and relevance type II error criteria from the medical vision 68
4.7 Priority and relevance for attribute criteria from the acceptability vision 69
4.8 Priority and relevance type II error criteria from the acceptability vision 69
4.9 Summary table of the datasets of the domains tested 71
4.10 Decription of the attributes of the diabetes domain 71
4.11 Decription of the decisions of the diabetes domain 71
4.12 Tests needed for each attribute in the diabetes domain 72
4.13 The values of K_x for the diabetes domain 72
4.14 The values of Ie_h for the diabetes domain 73
4.15 The values of IIe_x for the diabetes domain 74
4.16 General test for the diabetes domain 74
4.17 General test (pruning) for the diabetes domain 74

4.18 General test (crossvalidation) for the diabetes domain 76
4.19 General test (crossvalidation/pruning) for the diabetes domain 76
4.20 Medical aspects test for the diabetes domain 78
4.21 Medical aspects test (pruning) for the diabetes domain 78
4.22 Decription of the attributes of the heart disease domain 79
4.23 Decription of the decisions of the heart disease domain 79
4.24 Tests needed for each attribute in the diabetes domain 80
4.25 The values of K_x for the heart disease domain 81
4.26 The values of Ie_h for the heart disease domain 81
4.27 The values of IIe_x for the post-operative domain 81
4.28 General test for the heart disease domain 83
4.29 General test (pruning) for the heart disease domain 83
4.30 General test (crossvalidation) for the heart disease domain . . 85
4.31 General test (crossvalidation/pruning) for the heart disease domain 85
4.32 Medical aspects test for the heart disease domain 86
4.33 Medical aspects test (pruning) for the heart disease domain . . 86
4.34 Decription of the attributes of the post-operative domain . . . 87
4.35 Decription of the decisions of the post-operative domain . . . 87
4.36 Tests needed for each attribute in the post-operative domain . 88
4.37 The values of K_x for the post-operative domain 88
4.38 The values of IIe_x for the post-operative domain 89
4.39 General test for the post-operative domain 90
4.40 General test (pruning) for the post-operative domain 92
4.41 General test (crossvalidation) for the post-operative domain . 92
4.42 General test (crossvalidation/pruning) for the post-operative domain 92
4.43 Medical aspects test for the post-operative domain 94
4.44 Medical aspects test (pruning) for the post-operative domain . 94
4.45 Decription of the attributes of the thyroid domain 95
4.46 Decription of the decisions of the heart disease domain 96
4.47 Tests needed for each attribute in the diabetes domain 96
4.48 The values of Ie_h for the thyroid domain 97
4.49 The values of IIe_x for the thyroid domain 98
4.50 General test for the thyroid domain 98
4.51 General test (pruning) for the thyroid domain 98
4.52 General test (crossvalidation) for the thyroid domain 100
4.53 General test (crossvalidation/pruning) for the thyroid domain 100

4.54 Medical aspects test for the thyroid domain 102
4.55 Medical aspects test (pruning) for the thyroid domain 102

Chapter 1

Introduction

The aim of this thesis is to investigate which criteria are involved in the medical decision process and use them to improve the medical coherence of the decision structures used to assist the physicians.

Usually, in machine learning, decision structures are generated automatically without any kind of background knowledge of the domain using criteria like the information gain whose objective is to obtain simple and efficient structures [26, 27, 28]. Information gain can be enough in some domains where we only need the final decisions determined by the structure. However, there are other application domains where several criteria have to be considered and where we are not only interested in a final correct decision but in decision as a sequential process [29]. In these cases, obtaining structures as efficient as possible is not useful because the decisions made along the decision process may be incomprehensible (or incorrect) by the experts and the misclassification errors can be critical. The alternative is to acquire all the substantial background knowledge of the domain from the experts and use it in the generation of the decision structures. With this approach, *decision theory* is not the only theory involved. Usually, when generating decision structures considering background knowledge of the domain we also have to deal with *optimization* and *ordering theories*.

Our area of application is medicine. We want decision structures representing the different steps followed by physicians as they perform medical tasks (e.g., during diagnosis). Our approach uses *decision trees* as decision structures because they are a simple way for representing a decision process. In order to apply the medical background knowledge in the inductive generation of decision trees we use cost-sensitive and order-sensitive decision trees.

Cost-sensitive decision trees [3, 7, 14, 15] are generated using algorithms that try to minimize some kind of cost function. *Order-sensitive decision trees* [16] are automatically generated using algorithms that consider some kind of partial order among the local decisions in the whole decision problem. Both, cost functions and partial orders represent different domain criteria used in the decision process. In our case, we consider health-care criteria related to the length of the medical process (response time), the economy (economic cost), medical sense (adherence to medical standards and coherence with the way of doing in medicine) and acceptability (risk for the patient's health or influence on the patient's comfortability).

Each criterion has a different logical way of representation. For example, the economic cost of a certain test could be measured numerically (e.g., in euros) but the health risk is probably better by levels (e.g., {*no risk, low risk, moderate risk, high risk, unacceptable risk*}). One of the main problems that we will deal with is to find the best way to represent each one of the previously mentioned criteria assuming that we want a trade off between how exhaustive is the information represented and how easy is for the physician to provide this information. Finally, each representation of a criterion has to be transformed into a cost function or into a partial order to be applied in the decision tree generation.

Another important point is the combination of criteria. We want to integrate the knowledge from cost functions with the knowledge from orders into the algorithm of generation of decision trees letting the user specify the relevance or priority of each criterion.

The algorithms developed in this thesis are tested on real data coming from the Hospital Clinic (Barcelona, Catalonia, Spain), the Hospital Consortium SAGESSA (Reus, Catalonia, Spain) and the UCI Repository of Machine Learning [19].

Chapter 2

The state of the art

This chapter introduces the state of the art of the decision process in medicine. It first describes three mathematical *problems*: optimization, ordering and decision. Later, it presents the three *structures* that we use to implement the functions of optimization, ordering and decision. These are cost functions, partial orders and decision trees, respectively.

Once the problems and the structures we deal with are presented, we explain the three main *approaches* that, according to the specialized bibliography, have been applied to model the medical decision process and expose the drawbacks of these approaches when they are confronted to the construction of decision structures in medicine.

The first of the approaches is based on the *information gain* criterion and considers the medical decision process as a decision problem using no kind of background knowledge. The second approach is the *cost-sensitive* one. In this case, background knowledge is included in the form of cost functions. The problem is seen as a combination of optimization and decision. Finally, the *order-sensitive* approach includes background knowledge as an order relationship. In this approach, the problem is seen as a combination of ordering and decision.

2.1 Optimization

Optimization is the mathematical problem of finding out the values of a set of variables that minimize (or maximize) a numeric function. Formally speaking, an optimization problem (S, f) consists of a domain S and a func-

tion $f : S \rightarrow \mathbb{R}$. The objective is to find an element $a \in S$ such that $\forall b \in S, f(a) \leq f(b)$ (*minimization*) or such that $\forall b \in S, f(a) \geq f(b)$ (*maximization*).

Typically, S is some subset of the Euclidean space $\mathbb{R}^n$, often specified by a set of constraints, equalities or inequalities that the feasible solutions must satisfy. The domain S of the feasible solutions is called the *search space*. The elements of S are called *feasible solutions*. The function f is called *cost function* and a feasible solution that minimizes (or maximizes, if that is the goal) the cost function is called an *optimal solution*.

In the context of medicine, the problem of optimization is repeatedly observed in tasks like finding the treatment that has a lower risk on the patient's health (i.e., propose the treatment t such that risk(t) is minimal), or prescribing the drug that maximizes the patient's recovery process (i.e., prescribe the drug d such that Pr(recovery$|d$) is maximal).

2.2 Ordering

Ordering is the mathematical problem of sorting out the elements of a set. Formally speaking, an ordering problem $(S, \leq)$ consists of a domain S and a binary relation $\leq$ such that for any pair of elements $a, b \in S$, $a \leq b$ means that a precedes or it is at the same position that b, and $\leq$ satisfies the following properties:

- Reflexivity: $\forall a \in S, \{a \leq a\}$
- Antisymmetry: $\forall a, b \in S, \{a \leq b \wedge b \leq a \Rightarrow a = b\}$
- Transitivity: $\forall a, b, c \in S, \{a \leq b \wedge b \leq c \Rightarrow a \leq c\}$

In health-care, the problem of ordering is frequent like, for example, in the common practice of starting a treatment with soft drugs when that is possible, before prescribing more aggressive (and unpleasant) treatments (i.e., $t_1 \leq t_2$ if t_1 is a soft treatment and $t2$ is an aggressive treatment), or asking for some analysis and, according to the results, request some other analysis (i.e., $a_1 \leq a_2$, where a_1 is the first analysis and a_2 the second one).

2.3 Decision

Decision is the mathematical problem of, given a situation, choose one action out of a set of actions to be performed for that situation. Formally speaking, a decision problem (S, D, f) consists of a domain S, a set of decisions D, and a decision function $f : S \to D$. The problem of decision is equivalent to the problem of *classification* [11, 12, 38].

In health-care, the problem of decision is so frequent that sometimes it receives special names as *diagnosing* (i.e., decide on the sort of disease), *assessing* a patient condition (i.e., decide the severity of a disease sign or symptom) or prescribing a treatment (i.e., decide on the proper therapy).

2.4 Cost functions

In the problem of optimization, the *cost function* (sometimes called objective function) plays the main role in the *search process* that is used to find the optimal element in the domain. In medicine, cost functions can be used to represent the economic cost of some medical procedures, the risk of a treatment on the patient's health, the response time of certain drugs in urgent treatments, etc.

Optimization can be based on a single cost function $f : S \to \mathbb{R}$ where S is the search space where we have to find the element $a \in S$ such that for all $b \in S, f(a) \leq f(b)$; also denoted as $a = \arg\min_b f(b)$. For example, finding out the most economic treatment among a set of possible treatments a patient may receive. However, optimization in medicine uses to be based on the combination of several cost functions (e.g., economic cost and risk factor). The combination of cost functions is a complex topic that may require the application of three steps: convert cost functions to the same sense, convert cost functions to the same range, and weight the cost functions.

2.4.1 The sense of a cost function

A cost function representing *health risk* is in the correct sense since the higher the risk is, the higher the global cost should be, so the option with less risk should be chosen (i.e., minimization problem). On the contrary, a cost function representing patient *comfortability* goes in the opposite sense of the global cost since the more comfortable the patient is, the lower the

global cost should be, so the option with a greatest comfortability should be the one chosen in the optimization process (i.e., maximization problem).

Formally speaking, a cost function in the correct sense $f : S \to \mathbb{R}$ can be directly combined in a global cost function, but a cost function in the opposite sense $f : S \to \mathbb{R}$ must be converted to a function $f' : S \to \mathbb{R}$ such that $f' = -f$ before it is combined with other cost functions.

2.4.2 The range of a cost function

On the one hand, some cost functions as the economic cost of medical procedures may be given in several units as hundreds or thousands of euros. This means that the increment of the cost of a procedure in one year may affect the global cost function in different ways. For example, an increment of €100 will increase the global cost function in 1 if it is given in hundreds but in 0.1 if it is provided in thousands.

On the other hand, different cost functions might be represented in different magnitudes as for example, currency units or time units. In any of these cases, the direct combination of cost functions with different ranges should be avoided.

The way of dealing with these situations is to adjust all the cost functions to the same range. Formally speaking, in cases where the cost function has a maximum value, a possible adjustment is to convert the cost function $f : S \to \mathbb{R}$ into the cost function $f' : S \to [0, 1]$ using equation 2.1.

$$f'(x) = \frac{f(x)}{\max\limits_{s \in S}(f(s))} \tag{2.1}$$

A particular case of this is when the domain S is a finite set.

2.4.3 The weight of a cost function

When two or more cost functions are combined, the contribution of all cost functions to the global cost function is not necessarily the same. This is usually taken into account with the introduction of the concepts of relevance and priority. The *relevance* of a cost function $f : S \to [0, 1]$ is normally represented by a coefficient $\alpha_f \in [0, 1]$ such that $\alpha_f > \alpha_{f'}$ means that f is more relevant than f'.

The *priority* of a cost function $f : S \to [0, 1]$ is normally represented by a natural number $p_f \in \mathbb{N}$ such that $p_f < p_{f'}$ means that f is more priority than f'.

2.4.4 Combining cost functions

Suppose N cost functions $f_1, ..., f_N$ in the same range, the same sense and the same priority with α_i representing the weight of the cost function f_i $(i = 1..N)$ such that $\sum_{i=1}^{N} \alpha_i = 1$, then the most common way of combining these cost functions is by linear combination, as it is described in equation 2.2.

$$g(x) = \alpha_1 \cdot f_1(x) + \alpha_2 \cdot f_2(x) + ... + \alpha_N \cdot f_N(x) \tag{2.2}$$

When there are cost functions of several priorities to combine, the equation 2.2 is applied on the cost functions with priority 1 to obtain a global cost for the higher level g_1. The element that has the lowest value for g_1 is considered the optimal. If there are several elements in S that are optimal for g_1, then the global cost g_2 for the functions of priority 2 is calculated for these elements repeating the process for each priority level until one of the elements obtains a lower cost. If all the levels provide the same global cost for several elements then they are considered to have the same cost and so all of them are optimal solutions.

2.4.5 The cost function in the optimization problem

For small domains, the optimal solution of an optimization problem can be calculated using an exhaustive search. This means evaluating the global cost function g for each feasible solution in S and choosing the optimal one. For big domains, other more complex methodologies must be applied [9]. In this work, only optimization for small domains is required and, therefore, the optimization problem is solved with an exhaustive search process.

2.5 Partially ordered sets

Ordering is the second problem discussed in this work. Orders can be total, partial or preorders. In a total order, each element is related to each other element (i.e., for each $a, b \in S$ either $a \leq b$ or $b \leq a$, or both), while in a partial order or in a preorder, this condition is not mandatory (i.e., for each

$a, b \in S$ either $a \leq b$, or $b \leq a$, or both, or a and b are not related). In partial orders, for each $a, b \in S$ such that $a \leq b$ and $b \leq a$ we have that $a = b$ while in preorders this condition is not mandatory. In medicine, an example of total order is fever, an example of partial order is the stage of a breast cancer patient [1] and an example of preorder is the preference of selection of attributes according to their health risk. In the first case, fever can be expressed as the body temperature in $^\circ$ C and, therefore, it is always possible to determine which is the patient with a higher temperature among a group of patients. In the second case, the possible stages of breast cancer are 0, I, IIA, IIB, IIIA, IIIB and IV and the comparison of stages among the patients must be made according to figure 2.1 and it is not always possible to determine which patient is worst according to these stages (e.g., patients in stage IIA are not comparable with patients in IIB). In the third case, an attribute a can be more risked than an attribute b (i.e., $b \leq a$) or this relation can be uncertain (i.e., neither $a \leq b$ nor $b \leq a$). But if a and b are obtained using a same medical test we know that the health risk of a and b is exactly the same (i.e., $a \leq b$ and $b \leq a$).

2.5.1 Definitions

Given an ordering problem $P = (S, \leq)$, S is called the *ground set* of P and $\leq$ is called a *partial order*. The ground set equipped with the partial order is called *partially ordered set*.

Two elements $a, b \in S$ are *comparable* if either $a \leq b$ or $b \leq a$ or both.

Given three elements $a, b, c \in S$ such that $a \leq b \leq c$ then b is said to be *between* a and c.

Given two elements $a, b \in S$ we say a *covers* b if $a \leq b$ and there is not any $c \in S$ such that $a \leq c \leq b$ or if $b \leq a$ and there is not other $c \in S$ such that $b \leq c \leq a$. In the first case, a is a *lower cover* of b ($a \prec b$) and in the second case a is an *upper cover* of b ($b \prec a$). We denote $c(a)$ the set of covers of a (i.e., $c(a) = \{b \in S : a \prec b \text{ or } b \prec a\}$).

A subset $C \subseteq S$ is called a *chain* in $P = (S, \leq)$ if and only if for any pair $a, b \in C$, $a \leq b$ or $b \leq a$ or both (i.e., C is a totally ordered subset of S).

The *length* of a partially ordered set $P = (S, \leq)$ is the cardinality of the biggest chain in P.

On the contrary, a subset $C \subseteq S$ is called *antichain* in $P = (S, \leq)$ if and only if for any pair $a, b \in C, (a \neq b)$ neither $a \leq b$ nor $b \leq a$ (i.e., C is a totally unordered subset of S).

The *width* of a partially ordered set $P = (S, \leq)$ is the cardinality of the biggest antichain in P.

An element $a \in S$ is called a maximal (or minimal) element if there is none $b \in S$ for which $a \leq b$ (or $b \leq a$). The set of maximal (or minimal) elements of a partially ordered set $P = (S, \leq)$ is denoted $MAX(P)$ (or $MIN(P)$).

Hasse diagrams [24] are used to represent partially ordered sets. These diagrams are a graphical rendering of a partially ordered set displayed via the cover relation of the partially ordered set with an implied upward orientation. A point is drawn for each element of the ground set of the partially ordered set, and line segments are drawn between these points according to the following two rules:

1. If $a \leq b$ in the partially ordered set, then the point corresponding to a appears lower in the drawing than the point corresponding to b.

2. The line segment between the points corresponding to any two elements a and b of the partially ordered set is included in the drawing if and only if $b \in c(a)$.

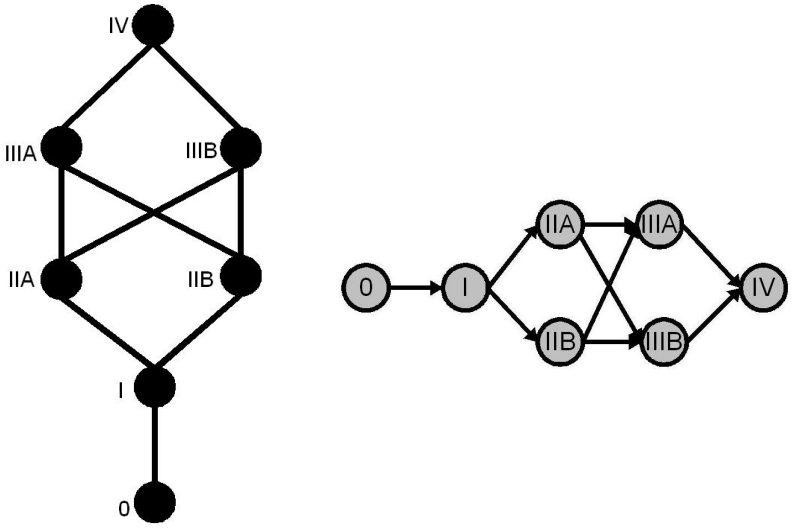
Figure 2.1(a) depicts an example of a Hasse diagram of the partially ordered set with $S = \{0, I, IIA, IIB, IIIA, IIIB, IV\}$ representing stages of breast cancer according to [1]. It is observed, for example, that $I \leq IIA$ because I appears in a lower position of the diagram and there is a way of going up from I to IIA. This represents the meaning that stage I is better than stage IIA, as far as breast cancer is concerned. By transitive property, we may conclude also that $I \leq IIIB$, $0 \leq IV$, etc. but it is impossible to say whether $IIA \leq IIB$ or $IIB \leq IIA$.

In health-care, these diagrams are usually represented as left-right Hasse diagrams as the one depicted in figure 2.1(b) obtained from [1].

Given an ordering problem $P = (S, \leq)$ the order relation $\leq$ is called a *preorder* or a *quasiorder* if it does not necessarily satisfy the antisymmetry property. A preorder relation is usually denoted by $\lesssim$. Notice that partial orders are particular cases of preorders.

Given an ordering problem $P = (S, \leq)$ the order relation $\leq$ is called a *total order* if it satisfies the trichotomy law:

- Comparability (trichotomy law): $\forall a, b \in S, \{a \leq b \vee b \leq a\}$



(a) Hasse diagram (b) Left-right Hasse diagram

Figure 2.1: Hasse diagrams corresponding to the partial order of the stages of breast cancer

A total order relation is usually denoted by $<$. A set S equipped with a total order relation $<$ is called a *totally ordered set* or *linearly ordered set* [10]. Notice that total orders are particular cases of partial orders.

Suppose a partially ordered set $(S, \leq)$, then an *extension* of $\leq$ is defined as a partial order $\leq^*$ such that for any elements $a, b \in S$ with $a \leq b$, it is also the case that $a \leq^* b$, but there can be relationships $a \leq^* b$ between elements $a, b \in S$, with $a \not\leq b$. A *linear extension* of a partial order is an extension which is a total order.

2.5.2 Layered partial orders

Given an ordering problem $P = (S, \leq)$, we say it is a *layered partial order* (LPO) if it satisfies the following property:

- $\forall a, b \in S$, if a and b are not related (i.e., neither $a \leq b$ nor $b \leq a$), then $c(a) = c(b)$

They are called layered partial orders because the elements in S are strictly arranged in layers. A layered partial order determines n disjoint antichains: $C_1, C_2, ..., C_n$ such that $\bigcup_{i=1}^{n} C_i = S$ and for each pair of elements

$(a, b) \in C_i \times C_j$, where $i < j$, we have that $a \leq b$. Notice that if $j = i + 1$, we also have that $a \prec b$. Each antichain C_i $(i = 1..n)$ is called a *layer* of the partial order. An element $a \in S$ is in layer i-th if $a \in C_i$. The layer of an element $a \in S$ is denoted as ℓ_a. A layered partial order with n layers is called n-layered partial order (n-LPO).

For example, the partial order depicted in figure 2.1 is an LPO (concretely a 5-LPO). As it can be observed, the elements of this partial order are arranged in the following layers:

$$\{0\}, \{I\}, \{IIA, IIB\}, \{IIIA, IIIB\}, \{IV\}$$

Observe that the covers of the elements in the same layer are identical and different from the covers of the elements in other layers. For example, $c(IIA) = c(IIB) = \{I, IIIA, IIIB\}$ and $c(IIIA) = \{IIA, IIB, IV\}$. The elements within each layer are not related to each other (e.g., $IIA \not\leq IIB$ and $IIB \not\leq IIA$) and they are all related to each of the elements of the next (or previous) layers (e.g., IIA is related to each element of the previous layers ($0 \leq IIA$, $I \leq IIA$) and to each element of the next layers ($IIA \leq IIIA$, $IIA \leq IIIB$, $IIA \leq IV$).

In figure 2.2 there is a partial order which is not layered. There are elements which are not related but which do not share the same set of covers (e.g., $e \not\leq f$ and $c(e) \neq c(f)$ because $c(e) = \{b, c, h\}$ and $c(f) = \{c, d, h\}$).

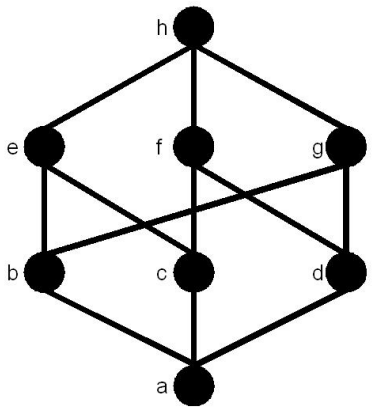


Figure 2.2: Not layered partial order

To represent medical knowledge in the decision process we always use LPOs because their composition in strict layers of priority makes them more

natural to medical problems. Therefore, in the rest of the document, when we are referring to a partial order we are actually meaning an LPO.

Ordering can be based on a single partial order $\leq$ over a ground set S. For example, *triage* in medicine is defined as the sorting of patients according to their needs when the available resources are insufficient for all the patients. The sorts of possible patients (i.e., needs) according to the triage system define the ground set S and the selection procedure is the partial order $\leq$.

Sometimes, an ordering in medicine cannot be solved with a simple LPO but with a combination of several LPOs, each one providing alternative ordering criteria that have to be taken into account. For example, in the problem of dealing with post-operative patients several attributes can be determined in order to decide where to send each patient (Intesive Care Unit, general hospital floor or home). In accordance to the criterion of comfortability, a good decision would be to measure the internal temperature of the patient as it is not an uncomfortable test. Nevertheless, according to the criterion of medical adherence, there are attributes like the oxygen saturation which are more prioritary although they may be more uncomfortable. Thus, deciding which attribute to ask first is not a trivial problem and must consider a combination of all the criteria involved. The combination of several LPOs based on a same ground set S is a complex topic. In the next sections, we introduce several operations that may be employed to combine LPOs.

2.5.3 Transforming partial orders into cost functions

We may need to transform an LPO $(S, \leq_S)$ into a cost function $f : S \rightarrow [0, 1]$ and therefore, converting an ordering problem into an optimization problem. In order to transmit the information contained in the partial order to the cost function, the transformation has to guarantee the following properties:

- $\forall a \in S, \{0 \leq f(a) \leq 1\}$
- $\forall a, b \in S, \{a \leq_S b \Leftrightarrow f(a) \leq f(b)\}$

As it can be determined from the above properties, the lower the layer of $a \in S$ is, according to the partial order, the lower the value of $f(a)$ is. As the partial order does not provide information about the distance between elements, we may assume that it is always the same, although other approaches could be considered. So, let $(S, \leq_S)$ be an n-LPO, we define equation 2.3 the transformation function that converts it into a cost function.

$$f(a) = \begin{cases} \dfrac{\ell_a - 1}{n-1} & \text{if } n > 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.3}$$

For example, for the partial order in figure 2.1, $f(0) = 0$, $f(I) = 0.25$, $f(IIA) = f(IIB) = 0.5$, $f(IIIA) = f(IIIB) = 0.75$ and $f(IV) = 1$.

2.5.4 Transforming cost functions into partial orders

Given a finite set S, a cost function $f : S \to [0, 1]$ can be transformed into an n-LPO $(S, \leq_S)$ by partitioning the elements in S in n chunks in accordance to their values of f. As we imposed that the distances between the layers of a partial order are always the same, each chunk c_i, $(i = 1..n)$, contains all the elements $a \in S$ such that $\frac{i-1}{n} \leq f(a) < \frac{i}{n}$ and chunk c_n also contains all the elements $a \in S$ such that $f(a) = 1$. Once the n chunks $c_1, ..., c_n$ are determined, the final LPO is the only one that fulfills that each element $a \in c_i$ is in layer i.

2.5.5 The weight of a partial order

As it also happens with cost functions, when two or more partial orders are combined, it may happen that not all the partial orders are equally important. In such cases, the level of importance of a partial order can be measured in terms of the concepts of relevance and priority. The *relevance* of a partial order $\leq_i$ is normally represented by a coefficient $\alpha_i \in [0, 1]$. We say a partial order $\leq_i$ is more relevant than a partial order $\leq_j$ if and only if $\alpha_i > \alpha_j$.

The *priority* of a partial order $\leq_i$ is normally represented by a natural number $p_i \in \mathbb{N}$ such that, given two partial orders $\leq_i$ and $\leq_j$, $p_i < p_j$ means that $\leq_i$ is more priority than $\leq_j$.

2.5.6 Combining partial orders

Supposing N partially ordered sets $P_i = (S, \leq_i)$ $(i = 1..N)$ with the same ground set S and the same priority, a function P_{ij}=*combine*$(P_i^{\alpha_i}, P_j^{\alpha_j})$ may be defined where $P_{ij} = (S, \leq_{ij})$ is a partially ordered set obtained from the combination of P_i and P_j with relevances α_i and α_j.

The function *combine* must fulfill some properties in order to be considered a valid procedure of combination of partial orders:

- Idempotence: $combine(P_i^{\alpha_i}, P_i^{\alpha_{i'}}) = P_i$
- Commutativity: $combine(P_i^{\alpha_i}, P_j^{\alpha_j}) = combine(P_j^{\alpha_j}, P_i^{\alpha_i})$
- Associativity: $combine(P_i^{\alpha_i}, combine(P_j^{\alpha_j}, P_k^{\alpha_k})) =$
  $= combine(combine(P_i^{\alpha_i}, P_j^{\alpha_j}), P_k^{\alpha_k})$
- Isotonicity: $\forall a, b \in S$, such that $a \leq_i b$ and $a \leq_j b$, then $a \leq_{ij} b$

We have identified three alternative approaches to the combination of N partial orders $P_i = (S, \leq_i)$, $(i = 1..N)$ if they have the same priority: structural combination, combination by transformation, and statistical combination:

- Structural combination takes the Hasse diagrams of the partial orders that have to be combined and produces a new Hasse diagram by means of structural operations. We did not found any of such methods in the revised bibliography.
- Combination through transformations: We have an injective transformation function $T : PO \rightarrow CS$ which gets a partial order $P_i \in PO$ and returns a certain structure $q_i \in CS$ which can be combined with other structures using the function $c : CS^N \rightarrow CS$. Then the process of combination is solved by the combination function $T^{-1}(c(T(P_1), ..., T(P_N)))$. In section 2.5.7, we present a way of carrying out this approach transforming partial orders to cost functions.
- Statistical combination: The priority of each element in S is statically estimated using the information provided for each partial order. In section 2.5.8, we present a method based on the linear extensions of a partial order.

When there are partial orders of several priorities to combine, we apply one of the previous procedures for partial orders with priority 1 to obtain a global partial order for the higher level $\leq_{g_1}$. This partial order is used to sort the elements of the ground set. If there are elements in S that cannot be ordered with $\leq_{g_1}$, then the global partial order $\leq_{g_2}$ obtained with

the combination of the partial orders of priority 2 is used to order these elements. The process is repeated for each level until the elements we want to sort are totally ordered or the last level of priority is reached and the remaining elements cannot be ordered.

2.5.7 Combination of partial orders through cost functions

Considering N partially ordered sets $P_i = (S, \leq_i)$ $(i = 1..N)$ with the same ground set S and the same priority, the first step of this method consists in transforming each partial order P_i into its corresponding cost function f_i with the procedure explained in section 2.5.3.

Then, these cost functions are combined as described in section 2.4.4. With this approach we are allowed to give a certain importance to each partial order by means of weights.

Finally, the cost function g is transformed into an LPO as section 2.5.6 indicates.

2.5.8 Statistical combination of partial orders

Considering N partial orders $P_i = (S, \leq_i)$ $(i = 1..N)$ with the same ground set S and the same priority, we define Ω_i as the set of all possible linear extensions of $\leq_i$. In total orders, the layer of an element is also called *rank*. Supposing the linear extension $\omega_i \in \Omega_i$, we denote the rank of an element $a \in S$ according to ω_i as $\omega_i(a)$ [22]. Therefore, the probability for a certain element $a \in S$ to be in rank r according to linear extension of the partial order $\leq_i$ is the one calculated by equation 2.4.

$$Pr_i(a, r) = \frac{\#\{\omega_i \in \Omega_i : \omega_i(a) = r\}}{\#\{\omega_i \in \Omega_i\}} \tag{2.4}$$

The probability of a certain element $a \in S$ to be ranked in a position r by each partial order according to its relevance is calculated with equation 2.5.

$$Pr(a, r) = \sum_{i=1}^{N} \alpha_i Pr_i(a, r) \tag{2.5}$$

Given this probability we can decide the most probable rank $\omega(a)$ with equation 2.6 where $[x]$ is the nearest integer to x.

$$\omega(a) = \left[\sum_{r=1}^{|S|} r \cdot Pr(a, r) \right] \tag{2.6}$$

These ranks are used to obtain the final LPO, where each element $a \in S$ is situated in the layer $\omega(a)$. If a layer has no elements it is removed. Thus, the final LPO has a maximum of $|S|$ layers.

Although this approach can be applied to the combination of partial orders in general, our application is only for layered partial orders. We observe that the resulting partial order is always an LPO.

2.6 Decision trees

The decision problem was introduced in section 2.3 as a tuple (S, D, f) where S is a domain, D is a set of decisions, and $f : S \to D$ is the decision function. Artificial intelligence has a long tradition in the generation of decisional structures to solve decision problems. One of the most used is decision trees. In this section, we provide an introduction to decision trees.

2.6.1 Definitions

A *tree* is a mathematical concept that denotes a simple, undirected, connected and acyclic graph. The edges are known as *branches*, the vertices of order 1 are called *leaves* and the rest of the vertices, *internal nodes*. A *rooted tree* is a tree in which a special node is singled out. This node is called *root*. In such kind of trees, nodes which are one edge away from a given node n are called *successors* of n.

Decision trees [25] are rooted trees used as decisional structures to solve a decisional problem (S, D, f). Each internal node contains an attribute taken from a set of attributes $\{a_i\}_{i=1..p}$ on the elements of S that represent the functions $a_i : S \to D_i$ such that any element s in the domain S is given an attribute value $a_i(s)$ in the attribute domain D_i. Each internal node with attribute a_i represents a partition of the domain D_i and it has as many successors as parts are in that partition. Each branch leading from an internal node a_i to a successor of a_i is labeled with one of the possible parts of the partition that the internal node represents. The leaves of the decision tree contain final single decisions from the set D.

Figure 2.3 depicts an example of a decision tree on the echocardiogram domain which is used to determine whether a patient who suffered a heart attack is still alive or not after 1 year [19]. For example, the root node, partitions the domain according to attribute *wall-motion-index*. Concretely, it separates the patients whose value for *wall-motion-index* is equal or lower than 1.333 from the patients whose value for *wall-motion-index* is greater than 1.333. Observe that each of the leaves of the decision tree contains one of the possible final decisions {*dead*, *alive*}.

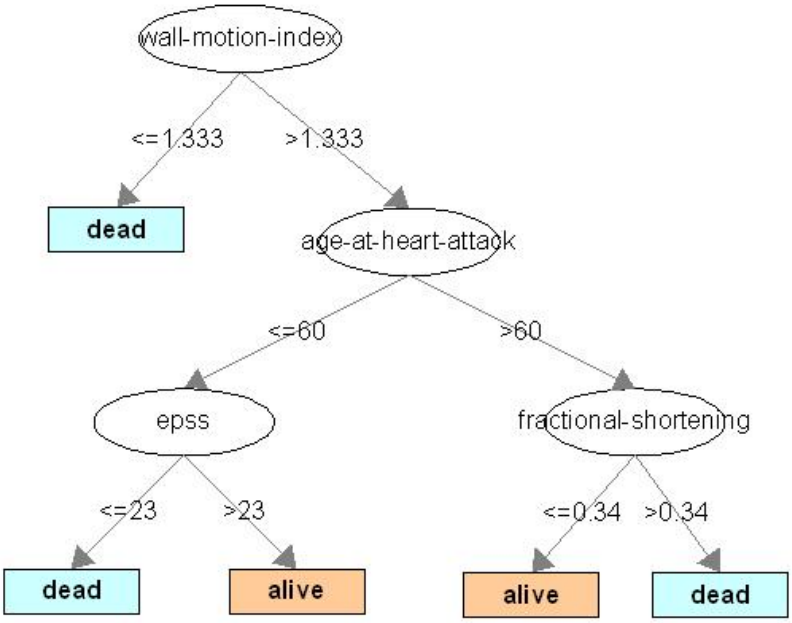


Figure 2.3: Decision tree on the echocardiogram domain

2.6.2 Using decision trees

In a decision problem (S, D, f), a decision tree may act as the decision function f. Given an element $s \in S$, usually called *instance*, such that $a_i(s) = v_i$ where v_i is the value of s for attribute a_i, the decision tree determines a path from the root node to a certain leaf. To decide which path corresponds to instance s, at each internal node n_i containing an attribute a_i, the next node of the path $n_{i'}$ is the successor connected to n_i by the branch labeled p_j such

that $p_j \subset D_i$ and $v_i \in p_j$ (i.e., at each internal node the branch whose label matches the value of the attribute for s is followed). This process is repeated until the path is completed and, thus, a leaf with a final decision d is reached.

Each path from the root to a leaf might be seen as a constraint over the values of the attributes in the path expressed as a conjunction. It is equivalent to a rule $\{a_1 \in p_1\} \wedge \{a_2 \in p_2\} \wedge ... \wedge \{a_n \in p_n\} \rightarrow d$ where a_i is the attribute corresponding to the i-th node of the path, each p_i is a different part of the partition $\{p_1, p_2, ..., p_n\}$ of D_i, and $d \in D$ is the final decision contained in the leaf confirming that all the elements $s \in S$ arriving to this leaf satisfy $f(s) = d$.

For example, considering the example of decision tree in figure 2.3, we suppose that we want to make a decision over a patient $s_1 \in S$ whose values for attributes *wall-motion-index*, *age-at-heart-attack*, *epss* and *fractional-shortening* are 1.8, 77, 16 and 0.13 respectively (i.e., *wall-motion-index*$(s_1) =$ 1.8, *age-at-heart-attack*$(s_1) = 77$, *epss*$(s_1) = 16$ and *fractional-shortening*(s_1) $= 0.13$). In this case, the patient would follow the path remarked in figure 2.4.

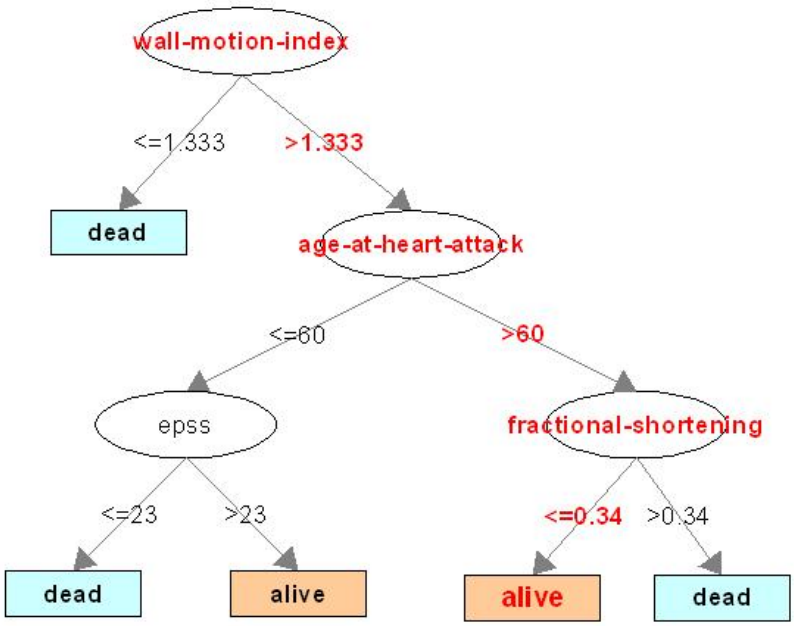


Figure 2.4: Path followed by the patient s_1 on the echocardiogram domain

In the root, the instance s_1 follows the branch on the right hand because

its value for *wall-motion-index* is 1.8 which is greater than 1.333. In the next node, the attribute evaluated is *age-at-heart-attack* which is 77. The instance s_1 follows the branch labeled > 60. Finally, it follows the left branch because its *fractional-shortening* is 0.13 which is lower than 0.34. At this point, the instance s_1 reaches a leaf *alive*. Therefore, the decision tree concludes that this patient will be alive after 1 year (i.e., $f(s_1) = alive$).

2.6.3 Inducing decision trees

Although decision trees can be build by hand, one of the main processes to automate the construction of decision trees is *induction* or *supervised learning*. This means that, given a set of instances $S' = \{s'_1, s'_2, ..., s'_m\}$ where $s'_i \in S$ and a set of decisions $S'_f = \{d_1, d_2, ..., d_m\}$ where $d_i \in D$ and $d_i = f(s'_i)$, the decision tree is automatically built using an inductive learning algorithm. The pair (S', S'_f) is usually called the *dataset*.

Most of the algorithms used in supervised learning of decision trees are *greedy* and *top-bottom*. An algorithm is said to be greedy when it follows the metaheuristic of making the locally optimum choice at each stage with the hope of finding the global optimum. Top-bottom is the strategy of starting from the root node (top) and generating the successive internal nodes of the tree until reaching the leaves (bottom). Some of the most typical algorithms for inducing decision trees are ID3 [28], C4.5 [26] and C5.0 [27]

The typical structure of a greedy and top-bottom algorithm for inducing decision trees is algorithm 1. At each node a final decision is taken if the condition for placing a leaf is reached (or if the set of attributes that can be used A is empty, i.e. $A = \emptyset$). In this case, the best decision at this moment is determined according a certain criterion. If the above condition is not reached, an attribute $a_i \in A$ is selected for partitioning the dataset. A new branch is created for each one of the possible parts of a partition of the domain D_i of the attribute a_i (e.g., each value of the domain is separated in a different part of the partition, as algorithm 1 is doing) and the function makes a recursive call with the instances in the current dataset that fulfill the condition in the branch in order to create a decision subtree for each one of the alternatives branches.

Algorithm 1: Induce_Tree

Input: (S', S'_f) : dataset, A : attributes
Output: DT : decision tree
$DT \leftarrow$ create decision tree with a root node;
if *(condition for placing a leaf reached)* $\vee$ $(A = \emptyset)$ **then**
    $d \leftarrow$ *selection of the best decision*;
    label the root node of DT with d;
    return DT;
else
    $a_i \leftarrow$ *selection of the best attribute*;
    label the root node of DT with a_i;
    foreach *value* v_j *of attribute* a_i **do**
        add a new branch b_j below the root of DT labeled v_j;
        $(S'(v_j), S'_f(v_j)) \leftarrow$ dataset of elements in S' such that $a_i = v_j$;
        if $S'(v_j) = \emptyset$ **then**
            $d \leftarrow$ *selection of the best decision*;
            add a leaf labeled d below b_j;
        else
            $DT' \leftarrow$ Induce_Tree$((S'(v_j), S'_f(v_j)), A - a_i)$;
            add the subtree DT' below b_j;
        end
    end
end
return DT;

2.7 Modeling medical decisions

Decision making is a common activity in medicine. So, for example, diagnosis, drug and therapy prescription, or prognosis are about deciding on the patient disease, treatment, or evolution. Several approaches have been applied to deal with the problem of modeling medical decisions in order to create *decision support systems* (DSS) [13, 23, 31, 37].

In the previous sections, we have considered the problems of ordering, optimization and decision. In this section, we explain three approaches to model medical decisions which consider these problems. The first one is based on the information gain criterion and considers the medical decision process only as a decision problem and using no kind of background knowledge. In the cost-sensitive approach, background knowledge is included in the form of cost functions. Therefore, the problem is seen as a combination of optimization and decision. Finally, the order-sensitive approach includes background knowledge as an order relationship. In this case, the problem is seen as a combination of ordering and decision. For each one of the approaches, we analyze their application in the domain of medicine.

2.7.1 The information gain approach

Several techniques for inducing decision trees from datasets have been carried out in the field of machine learning. The most simple yet effective one is the information gain approach. This technique generates the decision tree calculating the *amount of information* gained for each one of the alternatives in each node of the decision tree. Information gain based algorithms obtain simple and accurate decision trees and are one of the best approaches for simple domains. For complex domains such as medicine their effectiveness is more debatable.

Definitions

The *information gain criterion* measures the amount of information [30] gained by partitioning the training set in accordance with the mutually exclusive values of a single attribute. This is to say, the information gain of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y when the value of X is known. The uncertainty about the value of the class attribute Y is measured by the *en-*

tropy $E(Y)$. Let X and Y be discrete variables with values $\{x_1, ..., x_n\}$ and $\{y_1, ..., y_m\}$ respectively (in case that X and Y were numeric variables some finite partition of the values should be determined). We define the entropy of Y in equation 2.7.

$$E(Y) = -\sum_{i=1}^{m} Pr(Y = y_i) \log_2(Pr(Y = y_i)) \tag{2.7}$$

If the value of X is already known, we define the uncertainty about the value of Y by the conditional entropy of Y given X, $E(Y|X)$ in equation 2.8.

$$E(Y|X) = \sum_{j=1}^{n} Pr(X = x_j) E(Y|X = x_j)) \tag{2.8}$$

Thus, the information gain of X with respect to Y is defined as equation 2.9.

$$I(Y; X) = E(Y) - E(Y|X) \tag{2.9}$$

One of the most famous decision tree inductive algorithms based on the concept of information gain is ID3 [28]. Essentially, it builds the tree by computing at each internal node the information gained when splitting the training set using each of the attributes and selecting the one that maximizes the gain. The structure of the ID3 algorithm is the one of algorithm 1. In this case, the condition for placing a leaf is that all the elements in the dataset must have the same final decision (see algorithm 2).

Algorithm 2: Condition for placing a leaf in ID3 approach

Input: S'_f : final decisions
Output: boolean
return $\forall d \in S'_f, d = d_i$;

In order to select the best of the decisions, the mode of the final decisions among the elements of the dataset is determined (see algorithm 3).

In order to select the best attribute, the ID3 algorithm chooses the one which maximizes the information gain (see algorithm 4).

The main disadvantage of using the information gain is that it has a strong bias in favor of the attributes with many values. Another algorithm called C4.5 [26] solves this by using the so-called *gain ratio criterion*. The

Algorithm 3: Selection of the best decision in ID3 approach

Input: S'_f : final decisions
Output: d : final decision
return $mode(S'_f)$;

Algorithm 4: Selection of the best attribute in ID3 approach

Input: (S', S'_f) : dataset, A : attributes
Output: a : attribute
return $\arg\max_{a \in A} I(D; a)$;

idea is to use the gain ratio $GR(Y;X)$ in equation 2.10 to select the best attribute instead of the information gain.

$$GR(Y;X) = \frac{I(Y;X)}{S(X;Y)} \tag{2.10}$$

In equation 2.10, $S(X;Y)$ is known as the *split information* which is sensitive to how wide and uniform the partition induced by an attribute is. It is defined in equation 2.11.

$$S(X;Y) = \sum_{j=1}^{n} Pr(X = x_j|Y) \log_2 \frac{1}{Pr(X = x_j|Y)} \tag{2.11}$$

The C4.5 algorithm has another important improvement with respect to ID3. This is the incorporation of pruning strategies to simplify the decision tree.

Another version has been published called C5.0 [27]. It improves C4.5 in speed, memory usage and it includes new characteristics as boosting techniques or weighting techniques which allow us to weight different attributes and misclassification errors.

Application of information gain based decision trees in medicine

The approach of using conventional decision trees generated using the information gain criterion has some drawbacks when it is applied in complex scopes like medicine. In medicine, good decisions are not only those which could obtain good results but also those which have a medical sense. Thus, in the context of producing decisional structures in medicine, success can be

measured at the level of the decision structure (i.e., is the structure taking good decisions?) and at the level of meaning (i.e., has the decision process of the structure a medical sense?).

Information gain based algorithms are exclusively centered in the construction of decision structures that take good decisions not necessarily avoiding the generation of medically incomprehensible or inapplicable decision models. One may argue that these models that are obtained from the evidence on the data, may hide decisional aspects that medical doctors may accept and adopt after a deeper analysis, but what reality shows is that what it normally happens is that medical doctors do not trust these decisions [18].

Moreover, in this approach no differences are considered among the possible misclassification errors. Usually, the relevace of making a wrong final decision is different for each possibility. Therefore, this fact can lead to models that have a great percentage of correct decisions but whose misclassification errors are critical.

Along the years, multiple works have proved ID3 based algorithms to be efficient machine learning algorithms to generate decision trees that obtain good decisions. However, less works have been published on the analysis of their quality in the generation of meaningful and practical results [7, 15, 16].

2.7.2 The cost-sensitive approach

This approach has been proposed as a solution to incorporate background knowledge in the induction of decision trees. Cost-sensitive decision trees are a kind of decision trees which have been generated considering background knowledge represented as a cost function. This cost function may have as the domain, for example, the set of attributes and it represents the knowledge about a criterion of the real world which is not explicitely given in the dataset (e.g., the economic cost of obtaining the values of each attribute).

Definitions

When experts make decisions in real-world they use to attach more importance to background knowledge which is not always explicit in a training set. In the bibliography, some approaches have been made that assist the induction of decision trees using background knowledge represented as a cost function [3, 7, 14, 15]. The models obtained with these approaches are called *cost-sensitive decision trees*. The concept of cost can refer to several criteria

such as economic cost or response time. Moreover, costs can be applied in different ways during the generation of decision trees. Among other applications of costs [36], the most typical ways of applying costs in the induction of decision trees are: specify the cost of obtaining the value of an attribute and specify the cost of making a misclassification error.

The cost of the attributes

In some domains, there is a cost related to obtain the value of an attribute (e.g., in medicine, a blood test has an economic cost). This cost can be constant for each attribute [20, 21, 32, 33, 34] or it can be conditioned on prior information [35]. In the case that it is conditioned it can depend on the attributes used before, on the values obtained for these attributes, on prior knowledge about the instance, etc.

The cost of the misclassification errors

Another way of applying costs in the generation of decision trees is considering misclassification errors. These errors occur when the tree assigns a final decision d_i to an instance s'_j when actually $f(s'_j) = d_j \neq d_i$. Often, there are misclassification errors which are more critical than others and therefore, having a cost function to represent this knowledge is essential. One of the ways for representing this knowledge is using a matrix. Letting $||D||$ be the number of final decisions, we define M as a $||D|| \times ||D||$ matrix where the element $M(d_i, d_j)$ specifies the cost of assigning the final decision d_i to an instance s'_j such that $f(s'_j) = d_j \neq d_i$. Usually, $M(d_i, d_i) = 0$. Once again, this cost can be constant [2, 8] for each misclassification error or conditioned [4, 5, 6] on prior information.

Application of cost-sensitive decision trees in medicine

In the bibliography, we have found a few applications of cost-sensitive decision trees as a feasible structure to represent the medical decision process. An interesting approach carried out in this area is the work done by Ling et al. [3, 14, 15]. This approach induces decision trees with an algorithm similar to C4.5 but changing the concept of information gain by the concept of economic cost. It associates a constant cost $C_A \in \mathbb{R}^{||A||}$ to obtain each attribute and it also incorporates costs for misclassification errors by means of a constant cost matrix $C_M \in \mathbb{R}^{||D||} \times \mathbb{R}^{||D||}$. Using these costs, the algorithm has the

structure of algorithm 1 where the condition for placing a leaf is whether it has a lower economic cost than selecting an attribute (see algorithm 5).

Algorithm 5: Condition for placing a leaf in Ling et al. approach (simplified)

Input: (S', S'_f) : dataset, D : decisions, A : attributes,
C_M : misclassification costs, C_A : attribute costs
Output: boolean
$T \in \mathbb{R}^{||D||}$, $T_A \in \mathbb{R}^{||A||}$;
foreach $d \in D$ **do**
  $T(d) = cost\_placing\_leaf((S', S'_f), C_M, d)$;
end
foreach $a \in A$ **do**
  $T_A(a) = cost\_selecting\_attribute((S', S'_f), C_M, C_A, a)$;
end
return $min(T) \leq min(T_A)$;

The way that vectors T and T_A are calculated is detailed in [15]. Essentially, to calculate the cost of placing a leaf, the function *cost_placing_leaf* uses the distribution of final decisions in S'_f to weigh the misclassification costs in C_M. So, the cost of placing a leaf for decision d depends on the number of elements s'_i such that $f(s'_i) \neq d$ and on the costs in C_M for making this misclassification error. On the other hand, in order to calculate the cost of selecting an attribute a, the function *cost_selecting_attribute* partitions the dataset according to the values of a and a prediction is made about the cost $T'_A(a)$ of placing a leaf after this partition. Finally, the cost to obtain attribute a (contained in C_A) is added to $T'_A(a)$.

Therefore, the best of the decisions $d \in D$ is the one such that placing a leaf for d is the cheapest alternative according to vector T (see algorithm 6).

The best of the attributes $a \in A$ to be selected is the one which minimizes the value $T_A(a)$ (see algorithm 7).

Ling et al. uses several strategies in order to obtain decision trees with minimal costs and it achieves good results. Nevertheless it is unrealistic to consider only economic costs in the medical decision process.

One of the major drawbacks of the previous procedure is that it does not consider any kind of criterion related to medical knowledge. When physicians make decisions (e.g., during the process of diagnosis) they are based above

Algorithm 6: Selection of the best decision in Ling et al. approach (simplified)

Input: (S', S'_f) : dataset, D : decisions, C_M : misclassification costs
Output: d : final decision
$T \in \mathbb{R}^{||D||}$;
foreach $d \in D$ **do**
  $T(d) = cost\_placing\_leaf((S', S'_f), C_M, d)$;
end
return $\arg\min_{d \in D} T(d)$;

Algorithm 7: Selection of the best attribute in Ling et al. approach (simplified)

Input: (S', S'_f) : dataset, A : attributes, C_M : misclassification costs, C_A : attribute costs
Output: a : attribute
$T_A \in \mathbb{R}^{||A||}$;
foreach $a \in A$ **do**
  $T_A(a) = cost\_selecting\_attribute((S', S'_f), C_M, C_A, a)$;
end
return $\arg\min_{a \in A} T_A(a)$;

all on other issues as health-care standards, or the health risk for the patient, etc.

Freitas et al. [7] expand the number of criteria used in the induction of medical decision trees. They consider the economic cost and the information gain of the attributes and introduce some medical knowledge using the concept of *risk*. The risk is a constant value that indicates whether obtaining a certain attribute has health risk or it is not patient-friendly. Some attributes will have a low value of risk if the medical tests to obtain their values are harmless (e.g., obstetric echography). The risk value will be higher for attributes whose tests can be dangerous and put patient's life at risk (e.g., cardiac catheterism) and for those which are not risked but uncomfortable (e.g., digestive endoscopy).

Once again this work is based on the C4.5 algorithm. It includes misclassification error costs and adapts the splitting criterion with the cost function of equation 2.12.

$$\frac{\Delta I_i}{(C_i \Phi_i)^\omega} \tag{2.12}$$

where, for each attribute a_i, ΔI_i is the gain ratio, C_i is the economic cost, Φ_i is the risk factor and ω is a constant called cost scale factor.

The work done by Freitas et al. is very exhaustive because it includes attribute costs, misclassification error costs and several important criteria. Nevertheless, the so-called risk criterion involves in a single value the health risk and the patient comfortability which are different concepts. Moreover, in misclassification error costs only economic costs are considered whereas medical criteria such as health risk or comfortability should also be included.

2.7.3 The order-sensitive approach

Another approach to include domain background knowledge in the induction of decision trees is by means of orders. In this case, the information about a criterion used in the real world is represented as an order among attributes, final decisions, etc. Usually, these orders are partial (see section 2.5) so they are useful to represent implicit and approximated orders (e.g., the medical coherence of asking for the value of an attribute).

Definitions

Order-sensitive algorithms that induce decision trees incorporate background knowledge of the domain represented as an order. The ground set of the order is usually the set of attributes but it can also be, for example, the set of final decisions. During the generation, in each node, only the alternatives which are more priority in accordance to the order provided are considered. Then, one of these alternatives is selected using another criterion. The trees obtained with the previous procedure are called *order-sensitive decision trees*.

The orders used can be total or partial. Normally, these algorithms deal with partial orders. This is because this kind of orders is ideal to represent implicit and approximated orders. An example of implicit and approximated knowledge could be the medical coherence of using an attribute during the process of diagnosis, it is obvious that this information is not exact. This knowledge depends on a combination of several concepts like the know-how and experience of the physician, the medical standards, etc. Usually, there are situations where it is definitely more medically coherent to determine the value of attributes a and b before asking for attribute c but it is indistinct the order of asking for a or b. In such cases the introduction of cost functions will provide more information than the proper medical information required, since medically senseless comparisons, when converted to cost functions, may acquire a mathematical sense that is wrongly interpreted as medical.

Application of order-sensitive decision trees in medicine

As far as we are aware, the only work that has been carried out within the order-sensitive approach is [16]. In this work neither the economic cost nor the misclassification error costs are considered. The criterion is a trade off between information gain and adherence to medical knowledge. The structure used to represent this background knowledge is a partial order over the set of attributes provided by a physician. This partial order aims to summarize all the medical knowledge related to this decision process. The partial order represents the relation "determining the value of the attribute a before b is more coherent from a medical point of view".

The algorithm (PS-C4.5) [16] is based on the C4.5 algorithm. It follows a structure similar to the structure of algorithm 1. As no misclassification error costs are considered, the condition for placing a leaf and the way it selects the best leaf is essentially the same that in algorithms 2 and 3.

The main difference respect to C4.5 algorithm is in how it selects the best attribute to split the dataset (see algorithm 8).

Algorithm 8: Selection of the best attribute in PS-C4.5 approach (simplified)

Input: (S', S'_f) : dataset, A : attributes, P : partial order over the set of attributes A
Output: a : attribute
$F \leftarrow PSAttributeSelection(P, A)$;
$a \leftarrow \emptyset$;
while $a = \emptyset \wedge notEmpty(P)$ **do**
    $a \leftarrow C45AttributeSelection(F)$;
    if $I(D; a) < \delta$ **then**
        Remove all the attributes in F from the P;
        $F \leftarrow PSAttributeSelection(P, A)$;
        $a \leftarrow \emptyset$;
    end
end
return a;

Firstly, it applies the *PSAttributeSelection* criterion (i.e., partial order criterion) to get the subset F of attributes with a higher priority (i.e., the attributes located in the first layer of the partial order). The next step consists in finding the best of the attributes in the subset F in accordance to the *C4.5AttributeSelection* criterion (i.e., gain ratio criterion). There can be several trade offs between PS and C4.5AttributeSelection. This is implemented by using an information gain threshold called δ in the algorithm. The information gained by the selected attribute must be greater or equal than δ. If $\delta = 0$ there will be no constraints and the order of selecting the attributes proposed by the partial order will be strictly respected. Another possibility is to set δ to the average value of the information gain. In this case, there will be a balance between both criteria. If it finds an attribute whose information gain is greater or equal than δ, the algorithm returns this attribute. Otherwise, it removes all the attributes in F from the partially ordered set P and begins again.

Although PS-C4.5 obtains accurate and meaningful results, it still can be improved. On the one hand, misclassification error costs should be included as well as other important criteria like the economic cost of obtaining

each attribute. On the other hand, representing all the knowledge related to a medical decision process as one partial order may be troublesome to physicians.

2.8 Evaluating medical decisions

In a decision problem (S, D, f), when we choose whether to make or not a certain medical decision $d \in D$ according to an induced decision function f_{DT}, we can distinguish among the following four cases:

	$f(s) = d$	$f(s) = \overline{d}$
$f_{DT}(s) = d$	True positive	False positive *(Type I error)*
$f_{DT}(s) = \overline{d}$	False negative *(Type II error)*	True negative

True positives (TP) are the cases such that $f_{DT}(s) = d$ and $f(s) = d$, *false positives* (FP) are the cases such that $f_{DT}(s) = d$ and $f(s) \neq d$, *false negatives* (FN) are the cases such that $f_{DT}(s) \neq d$ and $f(s) = d$ and *true negatives* (TN) are the cases such that $f_{DT}(s) \neq d$ and $f(s) \neq d$.

Observe that true positives and true negatives are the correct cases (i.e., $f_{DT}(s) = f(s)$). Thus, the rest of cases are errors. In some domains, like medicine, false positives are also known as errors of *type I* and false negatives, errors of *type II*.

Usually, to represent the relation between the real decisions and the decisions made by the system we use a similar matrix called *confusion matrix* C of the form $(n + 1) \times (n + 1)$:

	$f(s) = d_1$	$f(s) = d_2$	...	$f(s) = d_n$
$f_{DT}(s) = d_1$	...	...	...	...
$f_{DT}(s) = d_2$	...	...	...	...
...	...	...	...	...
$f_{DT}(s) = d_n$	...	...	...	...

where $d_i \in D$ $(i = 1..n)$ are the different possible decisions and each cell $C(i, j)$ contains the number of cases s that fulfill the conditions in $C(i, 0)$ and $C(0, j)$. Observe that the number of true/false positives/negatives can be extracted from this matrix for each decision. For example, for d_i we have:

	$f(s) = d_1$	...	$f(s) = d_i$	...	$f(s) = d_n$
$f_{DT}(s) = d_1$	TN_{d_i}	TN_{d_i}	FN_{d_i}	TN_{d_i}	TN_{d_i}
...	TN_{d_i}	TN_{d_i}	FN_{d_i}	TN_{d_i}	TN_{d_i}
$f_{DT}(s) = d_i$	FP_{d_i}	FP_{d_i}	TP_{d_i}	FP_{d_i}	FP_{d_i}
...	TN_{d_i}	TN_{d_i}	FN_{d_i}	TN_{d_i}	TN_{d_i}
$f_{DT}(s) = d_n$	TN_{d_i}	TN_{d_i}	FN_{d_i}	TN_{d_i}	TN_{d_i}

where the cells labelled as TP_{d_i}, TN_{d_i}, FP_{d_i} and FN_{d_i} contain respectively true positives, true negatives, false positives and false negatives when making the decision d_i.

2.8.1 Evaluation of the decision structure

In order to evaluate the results obtained by a decision structure, several measures have been defined based on the concepts of true/false positives/negatives. These measures only evaluate whether the induced decision function makes correct decision or errors according to the dataset without considering any kind of additional background knowledge. Therefore, these measures can be applied in any approach.

The *accuracy* is the degree of conformity of a measured or calculated quantity to its real value and it is formally defined in equation 2.13.

$$Accuracy = \frac{Correct\ decisions}{All\ the\ decisions} = \frac{\sum_{i=1}^{n} C(i,i)}{\sum_{i=1}^{n}\sum_{j=1}^{n} C(i,j)} \tag{2.13}$$

The accuracy is a general measure to evaluate the decision process. The following measures are related to each one of the possible decisions. The *predictivity* is represented by means of two measures: the *positive predictive value* and the *negative predictive value*. The positive predictive value for decision $d_i \in D$ is the proportion of correct results over all the cases for which our system has accepted the decision d_i and it is formally defined in equation 2.14.

$$PPV_{d_i} = \frac{TP_{d_i}}{TP_{d_i} + FP_{d_i}} = \frac{C(i,i)}{\sum_{j=1}^{n} C(i,j)} \tag{2.14}$$

The negative predictive value for decision $d_i \in D$ is the proportion of correct results over all the cases for which our system has rejected the decision d_i and it is formally defined in equation 2.15.

$$NPV_{d_i} = \frac{TN_{d_i}}{TN_{d_i} + FN_{d_i}} = \frac{\sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} C(j,k)}{\sum_{j=1,j\neq i}^{n} \sum_{k=1}^{n} C(j,k)} \tag{2.15}$$

The *sensitivity* for decision $d_i \in D$ is the proportion of correct results over all the cases for which the correct decision is to accept d_i and it is formally defined in equation 2.16.

$$Sensitivity_{d_i} = \frac{TP_{d_i}}{TP_{d_i} + FN_{d_i}} = \frac{C(i,i)}{\sum_{j=1}^{n} C(j,i)} \tag{2.16}$$

The *specificity* for decision $d_i \in D$ is the proportion of correct results over all the cases for which the correct decision is to reject d_i and it is formally defined in equation 2.17.

$$Specificity_{d_i} = \frac{TN_{d_i}}{TN_{d_i} + FP_{d_i}} = \frac{\sum_{j=1}^{n} \sum_{k=1,k\neq j}^{n} C(j,k)}{\sum_{j=1,j\neq i}^{n} \sum_{k=1}^{n} C(k,j)} \tag{2.17}$$

2.8.2 Evaluation of the decision cost

In the cost-sensitive approach, the previous measures are not enough. This approach includes background knowledge represented as a cost function and so we also have an optimization problem that must be evaluated. The most

logical measure is the *Average Cost* of the decision process formally defined in equation 2.18.

$$AC = \frac{1}{\|S\|} \sum_{s \in S} cost(f_{DT}(s)) \tag{2.18}$$

where $cost(f_{DT}(s))$ is the cost of the path of the decision tree that the instance s follows.

2.8.3 Evaluation of the decision order

In the order-sensitive approach a measure is needed to compare the decision tree with the partial order that contains the background knowledge. In [16], a measure called *Doctor's Satisfaction* (DS) was introduced. This measure scores a decision tree according to how well it follows a certain order of selection. Given a decision tree, the first step consists on transforming it into a partially ordered set. Beginning at the root, each level of the tree matches to a level of priority in the poset. The process continues until all the attributes in the tree have been treated. If there are attributes which do not appear in the tree, they are situated in the last level of the partially ordered set.

Then, supposing the partial order $\leq_1$ provided by the physician and $\leq_2$ obtained from a decision tree on the set of attributes A, we define $A_i = \{(a, b) \in A \times A | (a \leq_i b)\}$ the set of comparable attribute pairs in a partial order $\leq_i$. The symmetric difference between A_1 and A_2 is $A_1 \Delta A_2 = (A_1 \cup A_2) - (A_1 \cap A_2)$, its cardinality is a measure of how different $\leq_1$ and $\leq_2$ are, and DS in equation 2.19 is a measure of the similarity between the partially ordered set provided by the physician and the order used to select the attributes in the induction of the decision tree.

$$DS = 1 - \frac{card(A_1 \Delta A_2)}{card(A_1) + card(A_2)} \tag{2.19}$$

Chapter 3

Methodology proposed

We have observed that decisions in medicine involve problems of ordering, optimization and decision and our aim is to use all these concepts to model the medical decision process carried out by physicians. We built a system which sets out from a *dataset* of patients and some *background knowledge* of the domain and induces a medical decision tree.

As it was explained in section 2.6, in a decision problem (S, D, f), each element $s \in S$ is called an *instance* and $s = \{v_1, v_2, ..., v_n\}$ where v_i is the value of s for an attribute a_i. In our model we denote the set of attributes A. The *dataset* is the pair (S', S'_f) such that $S' = \{s'_1, s'_2, ..., s'_m\}$ where $s'_i \in S$ (instances) and $S'_f = \{d_1, d_2, ..., d_m\}$ where $d_i \in D$ and $d_i = f(s'_i)$ (final decisions).

The *background knowledge* is all the information which is not included in the dataset but physicians use implicitly or explicitly in their decisions. For example, this knowledge may include the information about the costs of using a certain attribute in the decision process.

In section 3.1 our decision model is proposed and explained. Section 3.2 introduces the medical criteria used in our approach. Section 3.3 explains in detail how the criteria are represented in order to be considered in the model. In section 3.4, we explain the procedures of combination of the criteria in order to be applied in the multi-criteria decision process specified in section 3.5.

3.1 Decision model proposal

The general decision model used in our approach is oriented towards the generation of a decision tree. In each node of the decision tree, the dataset and the background knowledge are processed and combined in order to decide whether to use an attribute to split the dataset or to make a final decision.

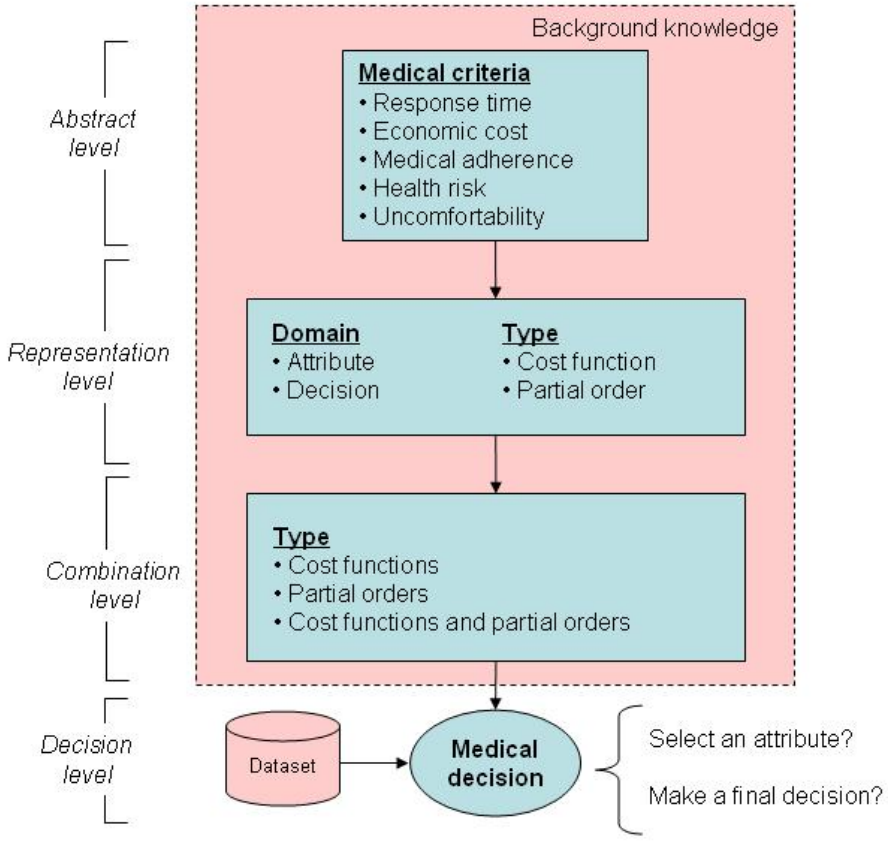


Figure 3.1: Scheme of the decision model

Figure 3.1 depicts how the background knowledge is treated in order to make decisions in a certain node. The top level is an abstract level containing the medical *criteria* considered in the decision process. In information gain based algorithms (see section 2.7.1) the criterion would be only information gain (which we do not consider in our model) while, as we have seen in

section 2.7.2 and 2.7.3, there are other approaches which consider criteria like the economic cost or the medical adherence of obtaining a value of an attribute in a certain moment.

In the representation level, the criteria of the abstract level are provided a concrete representation. A distinction is made according to whether the criteria is related to the set of attributes A or to the set of decisions D. These sets are called the domains of the criteria.

- Attributes: Some criteria are related to the attributes A and they are used to decide which attribute would be better to be obtained in a certain node. For example, if we have attributes of patient's surface temperature, stability of patient's surface temperature and stability of patient's internal temperature, the response time criterion will make us to ask for the patient's surface temperature because the evaluation of this attribute is the fastest one.

- Decisions: Other criteria are related to final decisions in D and distinguish between:

  - Type I error: These are criteria related to the relevance of rejecting the correct final decision. For example, the risk for health when a patient that deserves admission to a *general medicine* unit is not admitted.

  - Type II error: These are criteria related to the relevance of accepting a final wrong decision. For example, the risk for health of a wrong *discharge* of a patient.

When a criterion is represented our model may use a cost function or a partial order. These are called the types of background knowledge at the representation level.

In order to make a multi-criteria decision about attributes and decisions, the different criteria are combined in the combination level. We are able to combine cost function with cost functions, partial orders with partial orders and cost function with partial orders together.

Finally, in the decision level, the multi-criteria knowledge obtained from the combination level is used together with the dataset to decide whether to select an attribute (i.e., to split the dataset according to the values of a certain attribute) or to make a final decision (i.e., to place a leaf with a decision for all the patients in the dataset).

3.2 Medical criteria

The abstract level of the decision model gathers the medical criteria that will be used in the construction of decision trees. After an exhaustive analysis on how physicians make decisions we have created the semantic classification of criteria that is shown in figure 3.2.

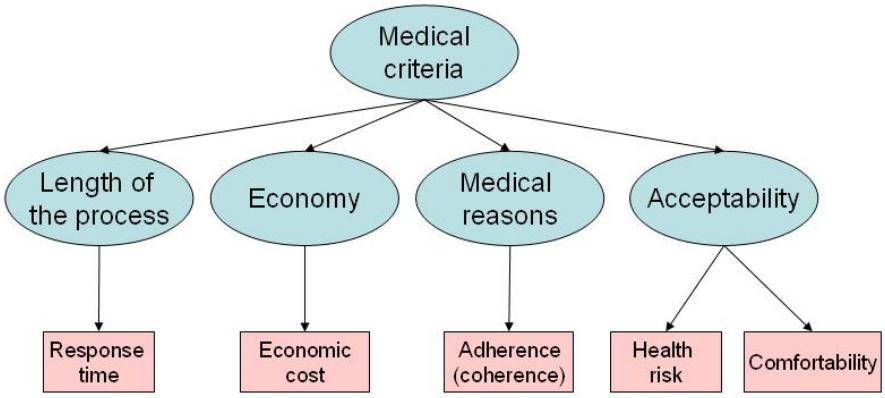


Figure 3.2: Main medical criteria

- Length of the process: The criteria which are related to the response time of the structure generated. Simple and brief decision structures are preferred.
- Economy: The economic cost of the tests and the treatments. Cheap decision structures are preferred.
- Medical reasons: They evaluate whether the decisions are more or less adjusted to the common medical practice. Decision structures with higher adherence to health-care standards are preferred.
- Acceptability: It contains the risk of the actions performed over the patient's health and their comfortability. Comfortable and safe decision structures are preferred.

In the decision model, each criteria is represented by a letter: response time (t), economic cost (e), medical adherence (m), health risk (h) and comfortability (c). The set containing each one of these criteria is denoted by O.

We use the medical criteria explained in this section since we have determined that these are the most relevant criteria in medicine. Nevertheless, the decision model is able to consider additional medical criteria that can be easily introduced if they are needed.

3.3 Representation of criteria

All the criteria in the previous section can be represented in different ways in the decision model. The knowledge supporting each criterion must be extracted and represented in some way considering a trade off between how exhaustive the information represented is and how easy for the physician is to provide this information.

For each criterion, we have two options to represent its background knowledge: a cost function or a partial order. Usually, we will work with cost functions when the information is directly extracted from hospital databases or clinical measurements and, with partial orders, when we are not able to provide exact values for the criteria. For example, for economic cost we may have the cost attached to each one of the procedures in the treatment of hypertension and be able to decide in terms of a cost function (i.e., optimization problem), but for health risk we may not have the absolute numbers but a pair-wise comparison of the risks of the procedures related to hypertension. In this case, the decision is taken in terms of a partial order (i.e., ordering problem). Nevertheless, there are some situations where physicians can represent better a implicit criterion like health risk using numeric values in a certain scale.

3.3.1 Criteria on the attributes

Some criteria have the set of attributes A as domain (see section 2.7.2) and are used to decide which attribute would be better to be selected in a certain node of the decision tree during the induction process. However, these criteria do not depend directly on the attributes themselves. Carefully observing their meaning it can be determined that they may depend on:

- The test needed to obtain the value of the attribute: response time, economic cost, health risk and comfortability.
- The context: medical adherence.

In the first case, some criteria are not related directly to their corresponding attributes but to the tests producing the values of these attributes (e.g., the attribute *sodium_on_blood* has no economic cost itself, but related to the cost of a blood test that provides the value for this attribute). A *test* is a procedure used to obtain the value of an attribute for a certain patient. In our model, tests are represented by greek letters as α, β or γ. The set of tests is represented by T. The function $test : A \rightarrow T$ returns the test corresponding to a certain attribute (i.e., $test(a) = \beta$ denotes that the value of attribute a is determined by the test β). In this decision model, a test may provide the value of several attributes but each attribute can only be obtained by a unique test (i.e., the *test* function is not injective). The criteria that depend on the test performed are response time (t), economic cost (e), health risk (h) and comfortability (c). Notice that the knowledge corresponding to these criteria has to be provided only once for each test (i.e., if a test is useful for the diagnosis of n diseases we will only need to obtain its information about response time, economic cost, health risk and comfortability once because it will always be the same). These criteria are also called *structural criteria*.

In the second case, some criteria may depend on the context of the medical decision process. Obtaining the value of an attribute may be the most logical decision from a medical point of view during the treatment of a certain disease but it may be irrelevant if the context of the treatment is different For example, determining the stability of the patient's blood pressure is a logical decision from a medical point of view when deciding where to send a post-operative patient but this attribute is not a logical decision in other domains like determining whether the patient is hypothyroid or not. In our model, only the criterion of medical adherence (m) depends on the context. For this kind of knowledge, the relevance of obtaining the value of an attribute depends on the context (i.e., the disease). The disease is the context of the medical adherence and, therefore, this knowledge must be provided for each disease. These kind of criteria is known as *problem criteria*.

As we said, each criterion can be represented as a cost function or as a partial order. In our model, all the structural criteria can be represented as a cost function (concretely response time (t), economic cost (e), health risk (h) and comfortability (c)). Although we do not have exact real values for health risk and comfortability, some physicians may find it easy to decide numeric values in a certain scale to indicate whether a test is risked/uncomfortable or not. When a criterion is represented as a cost function, we deal with an optimization problem (A, K_x). The cost function is defined as $K_x : A \rightarrow [0, 1]$

where $x \in O$ is the corresponding criterion. As we always deal with structural criteria, to obtain the cost K_x of a certain attribute a, we will have to know the cost of its corresponding test. The cost of a test is determined by the function $\kappa_x : T \rightarrow [0, 1]$ and so, we will always have that $K_x(a) = \kappa_x(test(a))$.

The κ_x function extracts information from vectors V_x of $|T|$ elements. $V_x(\alpha)$ contains the real cost in terms of criterion x for the test α. The transformation of the values in V_x consists in a unification of units (if needed) and a normalization. Since our cost functions return values between 0 and 1, κ_t, κ_e, κ_h and κ_c must normalize the values provided by the physician by applying equations 3.1, 3.2, 3.3 and 3.4 respectively.

$$\kappa_t(x) = \frac{V_t(x)}{max(V_t)} \tag{3.1}$$

$$\kappa_e(x) = \frac{V_e(x)}{max(V_e)} \tag{3.2}$$

$$\kappa_h(x) = \frac{V_h(x)}{max(V_h)} \tag{3.3}$$

$$\kappa_c(x) = \frac{V_c(x)}{max(V_c)} \tag{3.4}$$

where $max(V_x)$ id the maximum of the values in the vectors.

As an example of how we will deal with criteria on the attributes represented as a cost function we will suppose the set of attributes $A = \{L - CORE, L - SURF, L - O2, L - BP, SURF - STBL, CORE - STBL, BP - STBL, COMFORT\}$ and the set of tests $T = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta\}$ of the post-operative domain (detailed later in section 4.5). In table 3.1, we indicate which tests are needed to obtain each attribute.

In this example we consider the response time (t), therefore, the values of vector V_t provided by the doctor and the values of κ_t are shown in table 3.2 and finally, as $K_x(a) = \kappa_x(test(a))$, the values of the cost function K_t for each attribute are shown in table 3.3.

The values contained in vectors V_x (and so, the κ_x values) will change during the decision process. Obviously, the costs of performing the test will not vary along the generation of the tree, but when an attribute $a \in A$ is selected to split the dataset in a node n of the tree then, since this node, the test $\alpha \in T$ such that $test(a) = \alpha$ is supposed to have been performed and,

Table 3.1: Tests needed for each attribute

	α	β	γ	δ	ϵ	ζ
$L-CORE$	$\times$					
$L-SURF$	$\times$					
$L-O2$		$\times$				
$L-BP$			$\times$			
$SURF-STBL$				$\times$		
$CORE-STBL$				$\times$		
$BP-STBL$					$\times$	
$COMFORT$						$\times$

Table 3.2: The values of V_t and κ_t

	α	β	γ	δ	ϵ	ζ
V_t	1 min	0 min	2 min	1 h	1 h	0 min
κ_t	0.02	0	0.03	1	1	0

Table 3.3: The values of K_t

	K_t
$L-CORE$	0.02
$L-SURF$	0.02
$L-O2$	0
$L-BP$	0.03
$SURF-STBL$	1
$CORE-STBL$	1
$BP-STBL$	1
$COMFORT$	0

therefore, if the cost of α is needed later in the subtree whose root is n, it will always be 0 because the test α has already been carried out.

For example, in the previous example of the $post - operative$ domain, if the attribute $L - CORE$ is selected to split the dataset, the test α has been performed and, hence, in the next node, the response time related to attribute $L - SURF$ will be 0 because its value was actually obtained before (i.e., $K_t(L - SURF) = 0$).

We also observed that some criteria may be represented as a partial order. In our model, both structural and problem criteria can be represented as a partial order (concretely response time (t), economic cost (e), medical adherence (m), health risk (h) and comfortability (c)). When a criterion is represented as a partial order, we deal with an ordering problem $(A, \leq_x)$ where $\leq_x$ is an LPO that contains the knowledge about the criterion x. We use the partial order representation when the physician is unable to provide numeric values for a certain criterion. For example, concepts like health risk and comfortability do not have an evident unity of measurement and it is possible that the physician prefers to represent them as a partially ordered set. We may also use partial orders to represent exact criteria like response time or economic cost when the information is not available and the physician provides an approximation of the criterion.

For structural criteria, as the criteria depend on the tests, we have an initial partial order which is defined over the set of tests. So, instead of K_x, κ_x and V_x we will have an LPO (see section 2.5.2) denoted $\leq'_x$ where $x \in O$ is the corresponding criterion. We want to transmit this order to the domain of the attributes. Therefore, we define a preorder $\lesssim'_x$ over A such that for $a, b \in A$ and $\alpha, \beta \in T$ with $\alpha \Rightarrow_t a$, $\beta \Rightarrow_t b$ we have that $a \lesssim'_x b$ if and only if $\alpha \leq'_x \beta$. Finally, we transform the preorder $\lesssim'_x$ into an LPO $\leq_x$ over the set of attributes A such that $a \leq_x b$ if and only if $a \lesssim'_x b$ and $b \not\lesssim'_x a$. Although there is a loss of information, this transformation is valid for our approach because, as we work with LPOs, if $a \lesssim'_x b$ and $b \lesssim'_x a$, a and b will be located in the same layer of the partial order $\leq_x$ and so, the procedures of transformation into a cost function (see section 2.5.3) and combination of partial orders (see section 2.5.6) will maintain the coherence.

Setting out from the same example than before (i.e., the post-operative domain), now we deal with the uncomfortability criterion (c). In figure 3.3, the partial order $\leq'_c$ over the set of tests T is depicted.

The preorder $\lesssim'_c$ is depicted in figure 3.4 and the final partial order $\leq_c$ over the set of attributes A is shown infigure 3.5.

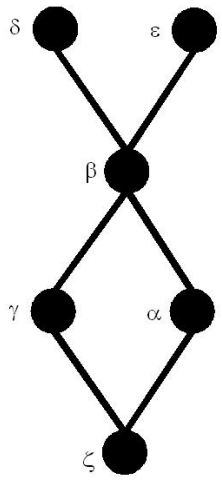


Figure 3.3: The partial order $\leq'_c$

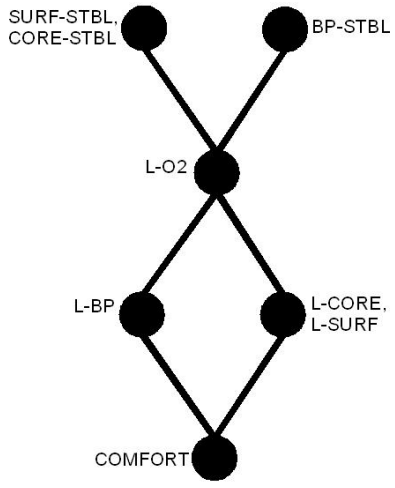


Figure 3.4: The preorder $\lesssim'_c$

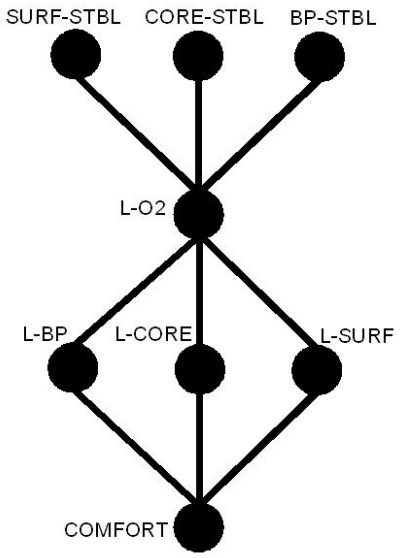


Figure 3.5: The partial order $\leq_c$

For structural criteria represented as a partial order, when a test is performed it is automatically translated to the layer with a highest priority of each partial order $\leq'_x$. This is because, like with cost functions, when the test has already been done it has no cost.

For example, in the previous example, if the attribute $L - CORE$ is selected to split the dataset, the test α has been performed and, hence, this test is translated to the layer with a highest priority of $\leq'_c$ and the attribute $L - SURF$ will finally be placed also in the first layer of $\leq_c$ because has already been obtained.

When we want to represent a problem criterion as a partial order the procedure is simpler. The problem criteria are not defined over the set of tests but over the set of attributes so the physician will directly provide the LPO $\leq_x$ over the set of attributes A.

Continuing in the post-operative domain, supposing that we want to represent the medical adherence (m) of the attributes (which is actually the unique problem criterion considered in our model), we will directly have a final partial order $\leq_m$ like the one in figure 3.6.

For problem criteria, the partial order is static during the decision process.

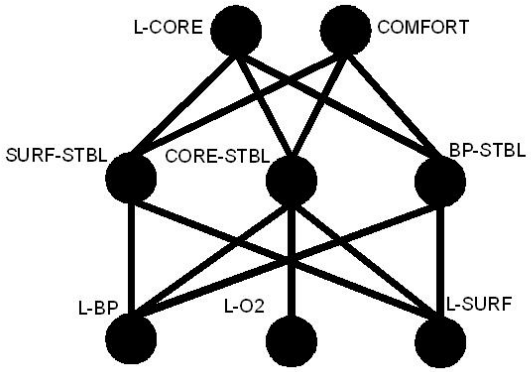


Figure 3.6: The partial order $\leq_m$

Therefore, in the previous example, if the attribute $L-CORE$ is selected to split the dataset, although the test α is performed and, so, the attribute $L - SURF$ is obtained, the partial order $\leq_m$ will be the same because the medical adherence of asking for attribute $L - SURF$ is a problem criterion (i.e., it does not depend on the test but in the context which is the post-operative domain).

3.3.2 Criteria on the decisions

Decision criteria are related to misclassification errors (see section 2.7.2) and they have as the domain the set of decisions. Although in the bibliography misclassification errors only imply economic costs, other criteria are usually involved. A wrong classification may imply a longer response time, risk on the health of the patient or a higher uncomfortability for him. Final decision criteria are always structural criteria because they depend directly on the final decision made (e.g., a treatment for a certain disease) and, therefore, their costs are always the same. We explained that decision criteria are divided into *type I error criteria* and *type II error criteria*, both used to determine the magnitude of a misclassification error.

As it is explained in section 2.7.2, usually, misclassification error costs depend on the final decision selected and the final decision that should have been selected. The most typical way of representing this knowledge is by a matrix where the element $M(d_i, d_j)$ specifies the cost of assigning a final decision d_i to an instance s'_j when actually $d_j = f(s'_j)$. Thus, in this case,

we would need a cost function (we call it M) of the form $M : (D, D) \rightarrow [0, 1]$ where D is the set of final decisions. This representation of M has drawbacks. First of all, the physicians will have to provide the different costs for each of the combinations of misclassification errors. This means a spatial cost $S(d) = \theta(d^2 - d)$ (considering $\forall d_i \in D, M(d_i, d_i) = 0$) which is often impracticable. Moreover, this representation is not medically realistic. Physicians do not consider the cost for each possible combination. This knowledge would be redundant. The costs considered for misclassification errors fulfill that the error of having classified wrongly an instance is independent of the real class of this instance and, also, the error of not having selected the correct class is independent of the class that has been actually selected (i.e., $\forall d_i, d_j, d_k \in D, M(d_i, d_j) = M(d_i, d_k)$ and $M(d_j, d_i) = M(d_k, d_i)$).

Therefore, we separate the criteria related to the error of having rejected the correct decision from the criteria related to the error of having selected a wrong decision. The first case involves the type I errors and the second case involves the type II errors. With this approach we obtain a more realistic representation and an increment on efficiency (the spatial cost is only $S(d) = \theta(2d)$).

Type I error criteria

Type I error criteria are related to the relevance of rejecting the correct decision for the dataset (e.g., the risk on the health of a patient who has not received the correct treatment). Health risk (h) is the only criterion involved in type I error criteria because not performing the correct treatment does not imply economic costs, losses of time, etc. directly.

We may represent the knowledge of health risk as a cost function or as a partial order. If the physician chooses the cost function representation, we will deal with an optimization problem (D, Ie_h) where D is the set of decisions and the cost function is defined as $Ie_h : D \rightarrow [0, 1]$ where h is the health risk criterion. These cost functions extract information from a vector W_h of $|D|$ elements which contain the values provided by the physician for criterion h for each final decision.

The transformation of W_h into the respective cost function is the same as explained for structural attribute criteria. We unify units of measure and apply equation 3.5 to normalize.

$$Ie_h(x) = \frac{W_h(x)}{max(W_h)} \tag{3.5}$$

where $max(W_h)$ is the maximum of the values in the vector.

However, if the physician chooses the partial order representation we will deal with an ordering problem $(D, \leq_{Ie-h})$, where D is the set of decisions and $\leq_{Ie-h}$ is an LPO over D where h is the health risk criterion.

As an example of how we will deal with type I error criteria we will define the set of decisions $D = \{I, S, A\}$ corresponding to the decisions of the post-operative domain (detailed later in section 4.5) of sending a patient to the Intensive Care Unit, to home or to the general hospital floor, respectively. Supposing that we want to represent as a partial order the health risk (h) (which is actually the unique type I error criterion considered in our model), we will directly have a final partial order $\leq_{Ie-h}$ like the one in figure 3.7.

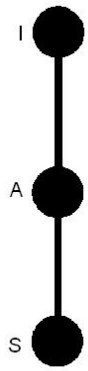


Figure 3.7: The partial order $\leq_{Ie-h}$

For instance, from $\leq_{Ie-h}$ we observe that the most risked error is not to send the patient to the Intensive Cure Unit (I) when it is actually the correct decision.

The type I error criteria are related to the final decisions of the decision tree, thus, the cost function Ie_h or the partial order $\leq_{Ie-h}$ will be static during the decision process.

Type II error criteria

Type II error criteria are those related to the relevance of the error of misclassification done when accepting a wrong decision for the dataset (e.g., the amount of time lost performing a wrong treatment). The criteria of response time (t), economic cost (e), health risk (h) and comfortability (c) are involved in type II error criteria. As in structural attribute criteria we can choose between cost function or partial order representation.

For cost function representations, we deal with an optimization problem (D, IIe_x) where D is the set of decisions and the cost function is defined as $IIe_x : D \rightarrow [0, 1]$ where $x \in O$ is the corresponding criterion. These cost functions extract information from vectors W'_x of $|D|$ elements which contain the real cost in terms of criterion x for each final decision.

The transformation of W'_t, W'_e, W'_h and W'_c into their respective cost function is done by unifying units of measure and applying equation 3.6, 3.7, 3.8 or 3.9 to normalize.

$$IIe_t(x) = \frac{W'_t(x)}{max(W'_t)} \tag{3.6}$$

$$IIe_e(x) = \frac{W'_e(x)}{max(W'_e)} \tag{3.7}$$

$$IIe_h(x) = \frac{W'_h(x)}{max(W'_h)} \tag{3.8}$$

$$IIe_c(x) = \frac{W'_c(x)}{max(W'_c)} \tag{3.9}$$

where $max(W'_x)$ is the maximum of the values in the vectors.

Continuing the example of the previous section, we consider the decisions in $D = \{I, S, A\}$ and we may want to represent the amount of time lost when sending the patient to the wrong place.

We have the vector W'_t with the real values (in days) in table 3.4.

Table 3.4: The values of W'_t

	I	S	A
W'_t	1	7	2

The function IIe_t transform the W'_t values into the cost values shown in table 3.5.

Table 3.5: The values of IIe_t

	I	S	A
IIe_t	0.14	1	0.29

According to the values of IIe_t, we observe that allowing the patient to go home (S) wrongly is the decision which will cause a higher loss of time.

If the partial order representation is chosen, we define the ordering problem $(D, \leq_{IIe-x})$, where D is the set of decisions and $\leq_{IIe-x}$ is an LPO over D where $x \in O$ is the corresponding criterion.

As an example on the previous set of decisions from the post-operative domain, we suppose that we want to represent the risk on the patient's health (h) when performing each wrong treatment. We will directly have a final partial order $\leq_{IIe-h}$ like the one in figure 3.8.

Figure 3.8: The partial order $\leq_{IIe-h}$

In the partial order $\leq_{IIe-h}$ we observe that, for example, the less risked decision is to send the patient to the Intensive Care Unit (I) wrongly.

The type II error criteria are related to the final decisions of the decision tree, thus, both the values in vectors W'_x (and so, IIe_x values) and $\leq_{IIe-x}$ partial orders will be static during the decision process.

3.4 Combination of criteria

In the decision model proposed, several criteria are defined to decide on the most appropriate attribute or final decision, from a health-care point of view. The structures that implement these criteria define the background knowledge of the system as cost functions and partial orders. In order to define a global decision criterion for the model, mechanisms for combining cost functions and partial orders are proposed. The aim is to finally obtain a global criterion for attributes, a global criterion for type I errors and a global criterion for type II errors. Our approach lets us choose carefully the priority and relevance of each criterion in the medical decision process. Therefore, we may, for example, generate a decision tree highly influenced by economic costs or another one that prefers healthsafe decision trees.

The combination is made by considering both the priority and the relevance of each criterion, following the steps described in sections 2.4.4 and 2.5.6. Broadly speaking, during the first step, the priority of each criterion is considered and, in a second step, all the criteria of the same priority are combined in a single decision criteria according to their relevance.

As some criteria may be represented as cost functions and others as partial orders, each level of priority may contain only cost functions, or only partial orders or both cost functions and partial orders. The combination of criteria in each one of the three situations is explained in the next sections.

3.4.1 N cost functions

Let $O = \{t, e, m, h, c\}$ be the set of criteria and $O_i \subset O$ be the set of criteria in the level of priority i where all the criteria in O_i are represented as cost functions.

For criteria on attributes we have an optimization problem which is defined as (A, AK_i) where A is the set of attributes and $AK_i : A \to [0, 1]$ is the global cost function for the level of priority i which is calculated with equation 3.10.

$$AK_i(x) = \sum_{y \in O_i} \frac{\alpha_y}{\sum_{y' \in O_i} \alpha_{y'}} K_y(x) \tag{3.10}$$

where α_y is the relevance of a criterion y.

For criteria on type I errors we have an optimization problem which is defined as (D, IEK_1) where D is the set of decisions and $IEK_1 : D \to [0,1]$ is the global cost function for the level of priority 1. There is only one criterion involved in type I errors so there is only one level of priority and no combination is needed (see equation 3.11).

$$IEK_1(x) = Ie_h(x) \tag{3.11}$$

For criteria on type II errors we have an optimization problem which is defined as $(D, IIEK_i)$ where D is the set of decisions and $IIEK_i : D \to [0,1]$ is the global cost function for the level of priority i which is calculated with equation 3.12.

$$IIEK_i(x) = \sum_{y \in O_i} \frac{\alpha_y}{\sum_{y' \in O_i} \alpha_{y'}} IIe_y(x) \tag{3.12}$$

where α_y is the relevance of a criterion y.

3.4.2 N partial orders

Let $O = \{t, e, m, h, c\}$ be the set of criteria and $O_i \subset O$ be the set of criteria in the level of priority i where all the criteria in O_i are represented as partial orders.

We apply the procedure of combination described in section 2.5.6.

For criteria on attributes we have an ordering problem which is defined as $(A, \leq_{A_i})$ where A is the set of attributes and $\leq_{A_i}$ is the global LPO for the level of priority i which is calculated with equation 3.13.

$$P_{A_i} = (A, \leq_{A_i}) = combina(P_{y_1}^{\omega_{y_1}}, combina(P_{y_2}^{\omega_{y_2}}, combina(..., P_{y_m}^{\omega_{y_m}}))) \tag{3.13}$$

where $P_{y_i} = (A, \leq_{y_i})$ and $y_1, ..., y_m \in O_i$ and $\omega_{y_i} = \frac{\alpha_{y_i}}{\sum_{y_j \in O_i} \alpha_{y_j}}$ where α_{y_i} is the relevance of criterion y_i.

For criteria on type I errors we have an ordering problem which is defined as $(D, \leq_{IE_1})$ where D is the set of decisions and $\leq_{IE_1}$ is the global LPO for the level of priority 1 which is calculated with equation 3.14.

$$P_{IE_1} = (D, \leq_{IE_1}) = \leq_{Ie-h} \tag{3.14}$$

No combination is required because there is only one criterion involved and there is only one level of priority.

For criteria on type II errors we have an ordering problem which is defined as $(D, \leq_{IIE_i})$ where D is the set of decisions and $\leq_{IIE_i}$ is the global LPO for the level of priority i which is calculated with equation 3.15.

$$P_{IIE_i} = (D, \leq_{IIE_i}) = combina(P_{y_1}^{\omega_{y_1}}, combina(P_{y_2}^{\omega_{y_2}}, combina(..., P_{y_m}^{\omega_{y_m}}))) \tag{3.15}$$

where $P_{y_i} = (D, \leq_{IIe-y_i})$ and $y_1, ..., y_m \in O_i$ and $\omega_{y_i} = \frac{\alpha_{y_i}}{\sum_{y_j \in O_i} \alpha_{y_j}}$ where α_{y_i} is the relevance of criterion y_i.

In our model, in order to use criteria on type I errors and criteria on type II errors in the medical decision procedure that will be explained in section 3.5, we will always need to finally work with an optimization problem. Thus, partial orders $\leq_{IE_1}$ and $\leq_{IIE_i}$ must be finally transformed into cost functions $IEK_1 : D \to [0, 1]$ and $IIEK_i : D \to [0, 1]$ applying equation 3.16 and equation 3.17 respectively.

$$IEK_1(x) = f_{IE_1}(x) \tag{3.16}$$

$$IIEK_i(x) = f_{IIE_i}(x) \tag{3.17}$$

where $f_{IE_1} : D \to [0, 1]$ and $f_{IIE_i} : D \to [0, 1]$ are the functions of transformation into a cost function corresponding to the partial orders $\leq_{IE_1}$ and $\leq_{IIE_i}$ respectively.

3.4.3 N cost functions and M partial orders

Let $O = \{t, e, m, h, c\}$ be the set of criteria and $O_i \subset O$ be the set of N criteria in the level of priority i where all the criteria in O_i are represented as cost functions and $O_j \subset O$ be the set of M criteria in the same level of priority i where all the criteria in O_j are represented as partial orders.

For criteria on attributes, the first step is to consider the ordering problem $(A, \leq_{O_j})$ where A is the set of attributes and $\leq_{O_j}$ is the partial order obtained by combining the partial orders of criteria in O_j using the combining procedures (see section 2.5.6) as for the previous section. The next step is to transform the ordering problem into an optimization problem (A, f_{O_j}) where $f_{O_j} : A \to [0, 1]$ is the cost function obtained by the transformation of the

partial order $\leq_{A_{O_j}}$ (as described in section 2.5.3). Finally, the final optimization problem (A, AK_i) is defined where the cost function $AK_i : A \rightarrow [0,1]$ is calculated with equation 3.18.

$$AK_i(x) = \sum_{y \in O_i} \alpha_y K_y(x) + \frac{\sum_{y \in O_j} \alpha_y}{\sum_{y \in O_{ij}} \alpha_y} f_{O_j}(x) \tag{3.18}$$

where i is the current level of the priority procedures, α_y is the relevance of a criterion y and $O_{ij} = O_i + O_j$.

The same way, for criteria on type II errors, we transform the ordering problem $(D, \leq_{O_j})$, where D is the set of decisions and $\leq_{O_j}$ is the partial order obtained by combining the partial orders of criteria in O_j, into an optimization problem (D, f_{O_j}) where $f_{O_j} : A \rightarrow [0,1]$ is the cost function obtained by the transformation of the partial order $\leq_{O_j}$. Then, the final optimization problem $(A, IIEK_i)$ is defined where the cost function $IIEK_i : D \rightarrow [0,1]$ is calculated with equation 3.19.

$$IIEK_i(x) = \sum_{y \in O_i} \alpha_y IIe_y(x) + \frac{\sum_{y \in O_j} \alpha_y}{\sum_{y \in O_{ij}} \alpha_y} f_{O_j}(x) \tag{3.19}$$

where i is the current level of the priority procedures, α_y is the relevance of criterion y and $O_{ij} = O_i + O_j$.

We do no include criteria on type I errors because it only involves one criterion (health risk).

3.5 Medical decision

By this moment, we have seen how medical criteria are represented in our model and how we combine their knowledge. In this section, we use this knowledge to generate a decision tree making a multi-criteria medical decision in each node. We have to decide between two options:

- Selecting an attribute (i.e., splitting the dataset according to the values of a certain attribute)

- Making a final decision (i.e., placing a leaf with a decision for all the patients in the dataset)

Our decision tree induction algorithm follows the same structure as algorithm 1. Therefore, the medical decision is based on the functions *condition for placing a leaf*, *selection of the best decision* and *selection of the best attribute*.

The function *condition for placing a leaf* calculates a total cost for making each of the final decisions. If the minimum of the costs is lower than a certain threshold, the condition is reached. We have to consider type I error criteria and type II error criteria. From the combination level, we have for each level of priority, a cost function IEK_i for type I error criteria and cost function $IIEK_i$ for type II error criteria. These costs correspond to making a final decision over a unique patient. However, to calculate the cost of placing a leaf we have to take in account the probability distribution of the final decisions over the whole dataset (see equation 3.20).

$$P_{S'_f}(d) = \frac{\#\{d' \in S'_f : d' = d\}}{\#\{d' \in S'_f\}} \tag{3.20}$$

where S'_f contains the final decisions of the dataset. The cost of placing a leaf FDK is calculated then using equation 3.21.

$$FDK_i(d, S'_f) = P_{S'_f}(\overline{d}) \cdot IIEK_i(d) + \sum_{d' \in D, d' \neq d} \left(P_{S'_f}(d') \cdot IEK_1(d')\right) \tag{3.21}$$

where $P_{S'_f}(\overline{d})$ is the probability of not having d as final decision (i.e., $P_{S'_f}(\overline{d}) = 1 - P_{S'_f}(d)$). If the minimum of the costs for level of priority 1 is lower than a threshold δ, the condition for placing a leaf is reached (see algorithm 9).

Algorithm 9: Condition for placing a leaf in the proposed model

Input: S'_f : final decisions, δ : threshold
Output: boolean
return $min_{d \in D}(FDK_1(d, S'_f)) < \delta$;

If the condition for placing a leaf is reached, the algorithm has to select one of the final decisions using the function *selection of the best decision*.

This function is similar to the previous one but, as there may be several final decisions with the same cost at the first level, it has to consider the other levels of priority. Concretely, the function determines the optimum final decision using the procedure described in section 2.4.4. If a unique final decision is minimum for FDK_1 this final decision is the one selected. Otherwise, if more than one final decision is minimum, the cost function FDK_2 is calculated only for these elements. The procedure is repeated for each of the levels of priority until there is only one optimal final decision. If for the last level of priority we do not obtain a unique optimal final decision, one of the optimal final decisions will be selected randomly. This procedure is detailed in algorithm 10.

Algorithm 10: Selection of the best decision in the proposed model

Input: S'_f : final decisions, p : levels of priority
Output: d' : final decision
$i \leftarrow 1$;
$FD \leftarrow \emptyset$;
while $|FD| \neq 1 \wedge i \leq p$ **do**
  $FD \leftarrow \arg\min_{d \in D} FDK_i(d, S'_f)$;
  $i = i + 1$;
end
return $random(d'), d' \in FD$;

If the algorithm decides not to place a leaf, it has to select one of the attributes to split the data. The procedure, as in the previous function, consists in selecting the optimum attribute using the method described in section 2.4.4 and 2.5.6. However, the selected attribute must have a expected cost lower than a certain threshold. The *expected cost EC* of an attribute a is the average of the cost of making a final decision for each of the datasets that are obtained by splitting the current dataset (S', S'_f) using the values of a (see equation 3.22).

$$EC(a, (S', S'_f)) = \frac{1}{n} \sum_{v=1}^{n} min_{d \in D}(FDK_1(d, Sv'_f)) \tag{3.22}$$

where (S', S'_f) is the dataset at the current node and Sv'_f contains the final decisions of the dataset that contains all the patients in (S', S'_f) such that their attribute a is valued v.

The function *selection of the best attribute* is detailed in algorithm 11.

Algorithm 11: Selection of the best attribute in the proposed model

Input: (S', S'_f) : dataset, A : attributes, p : levels of priority,
ϵ : threshold
Output: a' : attribute
$A_c \leftarrow \emptyset$;
foreach $a \in A$ **do**
  if $EC(a, (S', S'_f)) < \epsilon$ **then** $A_c \leftarrow a$;
end
$i \leftarrow 1$;
$A_s \leftarrow \emptyset$;
while $|A_s| \neq 1 \wedge i \leq p$ **do**
  if *for level p the general structure is a cost function* **then**
    $A_s \leftarrow \arg\min_{a \in A_c} AK_i(a)$;
  else
    $A_s \leftarrow \arg\min_{a \in A_c} \ell_a$;
  end
  $i = i + 1$;
end
return $random(a'), a' \in A_s$;

We observe that, first of all, we put in A_c each attribute whose expected cost is lower than the threshold ϵ. Then, the optimum attribute in A_c is determined. For each level of priority, if we have a cost function, the selected attributes are those whose value for AK_i is minimal and, if we have a partial order, the selected attributes are those situated in the lowest layer (ℓ_a is the layer of a in $\leq_{A_i}$).

Finally, we have enriched our algorithm with two different procedures of pruning. The first one, consists in specifying a percentage ζ which will refer to the minimum number of instances needed to split the dataset. If the previous algorithm decides not to place a leaf, then if the number of instances in the current dataset is lower than $\zeta\%$ of the initial number of instances, a leaf will be placed. The leaf selected will be the one given by the algorihtm 10. This kind of pruning is used to build more generalized decision trees. When a subtree is getting too specific (i.e., there are only very few instances) it is pruned.

The second kind of pruning is carried out after the decision tree has been generated. It detects subtrees whose leaves contain the same final decision. These subtrees are replaced by the corresponding leaf. With this pruning, we reduce the cost related to the attributes because in these kind of subtrees all the internal nodes are unnecessary.

The first kind of pruning is optional. We can choose whether to use it or not. The second kind of pruning is mandatory because the presence of subtrees that lead to the same leaf has absolutely no sense.

Chapter 4

Tests and results

In this chapter, the methodology proposed in chapter 3 is tested with data of real medical domains.

4.1 Tests performed and evaluation measures

We have performed two kinds of tests (called *general test* and *medical aspects test*). The general test compares our methodology ($MEDBK$) to the information gain approach (IG) using measures at the level of decision structure and measures at the level of meaning. To perform this test, a certain priority and relevance for each medical criterion has been decided. These weights are shown in tables 4.1 and 4.2 for attributes criteria and type II error criteria respectively (remember that type I error only involves health risk and so there is no combination of criteria).

Table 4.1: Priority and relevance for attribute criteria

Priority	Criterion	Relevance
1	Medical adherence	0.5
	Comfortability	0.3
	Economic cost	0.2
2	Health risk	0.8
	Response time	0.2

In order to evaluate the decision tree obtained with the general test we will use the measure introduced in section 2.8.1 called accuracy (equation 2.13).

Table 4.2: Priority and relevance type II error criteria

Priority	Criterion	Relevance
1	Health risk	0.9
	Comfortability	0.1
2	Economic cost	0.5
	Response time	0.5

We also want to evaluate our model at the level of meaning so the measure of accuracy is not enough. It is necessary to evaluate the adherence to the criteria provided. As our attribute criteria and decision criteria cannot be considered together we will not use a global measure like AC (equation 2.18) but two different measures: AC_A (the average attribute cost of the decision process for a patient) and AC_D (the average decision cost of the decision process for a patient) defined in equation 4.1 and 4.2 respectively.

$$AC_A = \frac{1}{\|S\|} \sum_{s \in S} cost(f^A_{DT}(s)) \tag{4.1}$$

$$AC_D = \frac{1}{\|S\|} \sum_{s \in S} cost(f^D_{DT}(s)) \tag{4.2}$$

where $cost(f^A_{DT}(s))$ is the cost of the attributes used in the path of the decision tree corresponding to s and $cost(f^D_{DT}(s))$ is the cost of the final decision for s.

In our tests, we will not use the rest of measures introduced in section 2.8.1 (positive predictive value, negative predictive value, sensitivity and specificity) because they are strongly related to the measures of AC_D.

In the medical aspects test we generate 3 versions of decision trees using our methodology with different weights for the criteria and compare them to the information gain approach (IG). Each one of these versions strongly considers a certain medical aspect. The first version is from an economic vision (ECO) (tables 4.3 and 4.4), the second one is from medical vision (MED) (tables 4.5 and 4.6) and the third one is from the patient's acceptability vision (ACC) (tables 4.7 and 4.8).

To evaluate the decision trees from the medical aspects test we also use the accuracy (equation 2.13). But for the meaning evaluation we have defined the AC^x_A and the AC^x_D measures which evaluate the same cost that the AC_A and

Table 4.3: Priority and relevance for attribute criteria from the economic vision

Priority	Criterion	Relevance
1	Economic cost	0.7
	Response time	0.3
2	Medical adherence	0.33
	Comfortability	0.33
	Health risk	0.33

Table 4.4: Priority and relevance type II error criteria from the economic vision

Priority	Criterion	Relevance
2	Economic cost	0.7
	Response time	0.3
1	Health risk	0.5
	Comfortability	0.5

Table 4.5: Priority and relevance for attribute criteria from the medical vision

Priority	Criterion	Relevance
1	Medical adherence	0.7
	Health risk	0.3
2	Economic cost	0.33
	Comfortability	0.33
	Response time	0.33

Table 4.6: Priority and relevance type II error criteria from the medical vision

Priority	Criterion	Relevance
1	Health risk	1
2	Comfortability	0.33
	Economic cost	0.33
	Response time	0.33

Table 4.7: Priority and relevance for attribute criteria from the acceptability vision

Priority	Criterion	Relevance
1	Comfortability	0.7
	Health risk	0.3
2	Economic cost	0.33
	Response time	0.33
	Medical adherence	0.33

Table 4.8: Priority and relevance type II error criteria from the acceptability vision

Priority	Criterion	Relevance
1	Comfortability	0.7
	Health risk	0.3
2	Economic cost	0.5
	Response time	0.5

the AC_D but only considering the criterion x. So, for example, the measure AC_A^e is the average economic cost for a patient related to the attributes used in the decision process. Then, using AC_A^x and AC_D^x we can calculate AC_A^{ECO}, AC_D^{ECO}, AC_A^{MED}, AC_D^{MED}, AC_A^{ACC} and AC_D^{ACC} which consider combination of costs of the first level of priority in each of the versions. These measures are defined in equations 4.3, 4.4, 4.5, 4.6, 4.7 and, 4.8 respectively.

$$AC_A^{ECO} = 0.7 \cdot AC_A^e + 0.3 \cdot AC_A^t \tag{4.3}$$

$$AC_D^{ECO} = 0.7 \cdot AC_D^e + 0.3 \cdot AC_D^t \tag{4.4}$$

$$AC_A^{MED} = 0.7 \cdot AC_A^m + 0.3 \cdot AC_A^h \tag{4.5}$$

$$AC_D^{MED} = AC_D^h \tag{4.6}$$

$$AC_A^{ACC} = 0.7 \cdot AC_A^c + 0.3 \cdot AC_A^h \tag{4.7}$$

$$AC_D^{ACC} = 0.7 \cdot AC_D^c + 0.3 \cdot AC_D^h \tag{4.8}$$

The comparation of approaches must be done independently for each of the measures. For example, we cannot compare the cost AC_A^{ACC} to the cost AC_A^{ECO}. This is because the scale of values of cost for each criterion is different. It depends on the representation structure used for this criterion, the distribution of the values of cost for each attribute (or decision), etc.

For each general test we will consider 4 alternatives:

1. Without crossvalidation and without pruning
2. Without crossvalidation and with pruning ($\zeta = 0.02$)
3. With crossvalidation (n=5, %=90) and without pruning
4. With crossvalidation (n=5, %=90) and with pruning ($\zeta = 0.02$)

The pruning procedure is the one specified in 3.5 which places a leaf when the number of instances in a certain node is below ζ% of the initial number of instances of the dataset. With pruning the decision tree is smaller and more generalized.

The crossvalidation is performed in order to evaluate the robustness of our decision trees. This procedure consists in inducing the decision tree n times (in our case, $n = 5$) and for each case the dataset is divided into 2 parts. A certain percentage of the dataset (in our case, 90%) is used to induce the decision tree and the rest of instances are used to evaluate the decision tree.

The medical aspects test will be carried out with and without pruning but always without crossvalidation (i.e., the dataset used to build the tree will be the one used to test the tree).

4.2 Medical domains

In order to test our model we have applied it in several health-care domains. The datasets are provided by the UCI Repository of Machine Learning [19]. Table 4.9 contains a brief summary about the main characteristics of the datasets of each domain.

The medical background knowledge for each of the domains has been provided by the Hospital Clinic (Barcelona, Catalonia, Spain) and the Hospital Consortium SAGESSA (Reus, Catalonia, Spain).

Table 4.9: Summary table of the datasets of the domains tested

	Instances	Attributes	Decisions
Diabetes	768	8	2
Heart disease	303	13	2
Post-operative	90	8	3
Thyroid	3772	20	3

4.3 Diabetes domain

4.3.1 The dataset and the background knowledge

In this domain we have to decide whether a patient is tested positive for diabetes or not. The attributes used are described in table 4.10.

Table 4.10: Decription of the attributes of the diabetes domain

	Description	*Values*
$times - pregnant$	Number of times pregnant	integer
$plasma - glucose$	Plasma glucose in an oral glucose tolerance test	integer
$diastolic - blood$	Diastolic blood pressure (mm Hg)	integer
$triceps - skin$	Triceps skin fold thickness (mm)	integer
$serum - insulin$	2-Hour serum insulin (mu U/ml)	integer
$body - mass$	Body mass index (weight in kg/(height in m)^2)	real
$diabetes - pedigree$	Diabetes pedigree function	real
age	Age (years)	integer

The possible final decisions are described in table 4.11.

Table 4.11: Decription of the decisions of the diabetes domain

	Description
0	Tested negative for diabetes
1	Tested positive for diabetes

In this domain each attribute is provided by a different test (see table 4.12).

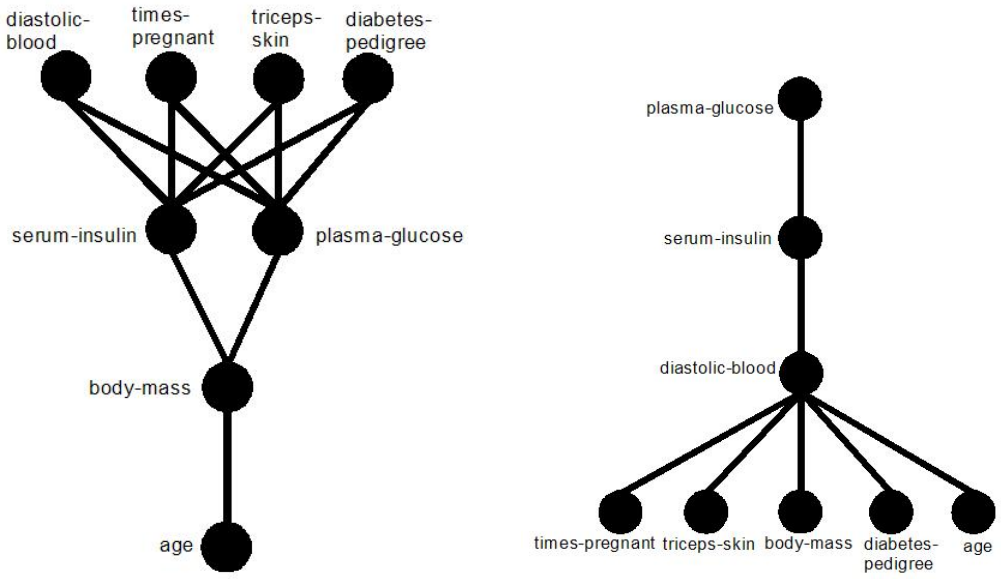
For attribute criteria, the final values for the cost functions K_e and K_t are detailed in table 4.13 and the final partial orders $\leq_m$ and $\leq_c$ are depicted in figure 4.1. We do not show the partial order $\leq_h$ because of the lack of risked tests in this domain.

Table 4.12: Tests needed for each attribute in the diabetes domain

	α	β	γ	δ	ϵ	ζ	η	θ
$times - pregnant$	$\times$							
$plasma - glucose$		$\times$						
$diastolic - blood$			$\times$					
$triceps - skin$				$\times$				
$serum - insulin$					$\times$			
$body - mass$						$\times$		
$diabetes - pedigree$							$\times$	
age								$\times$

Table 4.13: The values of K_x for the diabetes domain

	K_e	K_t
$times - pregnant$	0	0
$plasma - glucose$	0.5	0.5
$diastolic - blood$	0.25	0.008
$triceps - skin$	0	0.008
$serum - insulin$	1	1
$body - mass$	0	0.017
$diabetes - pedigree$	0	0
age	0	0



(a) $\leq_m$ (b) $\leq_c$

Figure 4.1: The partial orders $\leq_x$ for the diabetes domain

For type I error criteria, we represent the knowledge using a cost function Ie_h depicted table 4.14.

Table 4.14: The values of Ie_h for the diabetes domain

	0	1
Ie_h	0.8	1

For type II error criteria we also use cost functions which are detailed in table 4.15.

4.3.2 Results and analysis

For each test performed in the diabetes domain we have used the constants $\delta = 0.15$ and $\epsilon = 0.2$ for our algorithm.

Table 4.15: The values of IIe_x for the diabetes domain

	0	1
IIe_e	0.4	1
IIe_t	0	0
IIe_h	1	0.4
IIe_c	0	1

General test

The results of the general test for the diabetes domain are shown in tables 4.16, 4.17, 4.18 and 4.19 where our approach $MEDBK$ is compared to the information gain approach IG in terms of *Accuracy*, AC_A and AC_D. For each approach, the measures which are being minimized (or maximized in the case of accuracy) are remarked in bold.

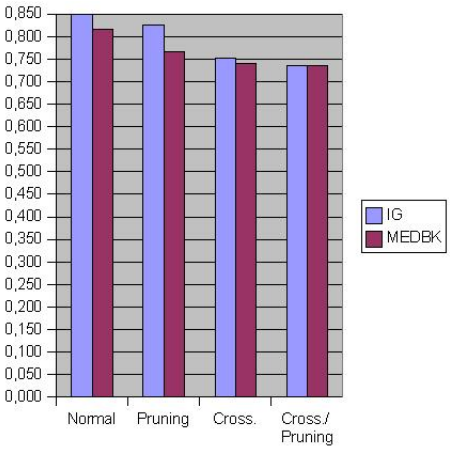
Table 4.16: General test for the diabetes domain

	Accuracy	AC_A	AC_D
IG	**0.850**	1.592	0.130
$MEDBK$	0.816	**1.140**	**0.136**

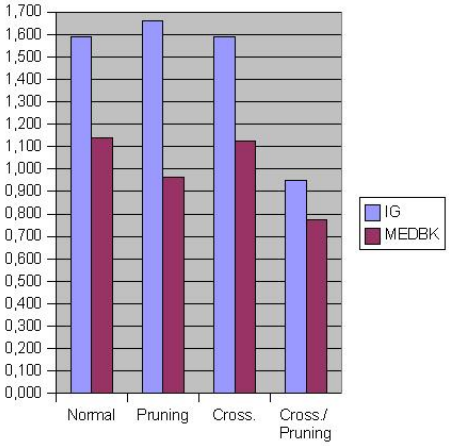
Table 4.17: General test (pruning) for the diabetes domain

	Accuracy	AC_A	AC_D
IG	**0.826**	1.662	0.140
$MEDBK$	0.767	**0.962**	**0.183**

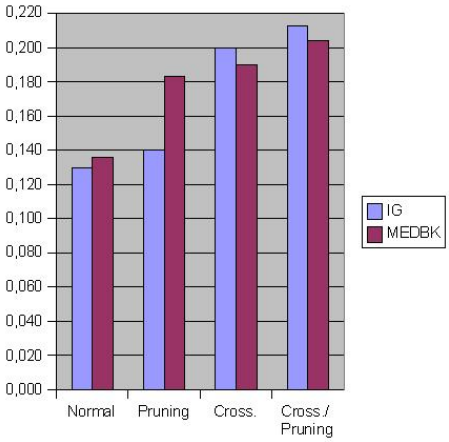
As far as accuracy is concerned, we observe that both approaches obtain similar results (see figure 4.2(a)). Although $MEDBK$ is not trying to minimize directly the accuracy its results are never more than 0.06 worse than IG. This is because the medical cost of a final decision according to our methodology is strongly related to the accuracy (see equation 3.21). The same way, the results obtained by IG in terms of AC_D are very good (see figure 4.2(c)). Once again, this is because the strong relation between accuracy and AC_D. Nevertheless, when comparing $MEDBK$ to IG we can find cases where IG is better in accuracy but worse in AC_D. For example, this is the case of the test without pruning and with crossvalidation. It could



(a) Accuracy



(b) AC_A



(c) AC_D

Figure 4.2: Results of the general test on the diabetes domain

Table 4.18: General test (crossvalidation) for the diabetes domain

	Accuracy	AC_A	AC_D
IG	**0.753**	1.593	0.200
$MEDBK$	0.740	**1.123**	**0.190**

Table 4.19: General test (crossvalidation/pruning) for the diabetes domain

	Accuracy	AC_A	AC_D
IG	**0.735**	1.682	0.213
$MEDBK$	0.737	**0.920**	**0.204**

seem (according to accuracy) that IG has obtained a better decision tree. But we have to take in account that the misclassification errors that IG has done are worst (according to AC_D). Therefore, in this cases, IG has made more correct decisions but the errors that it has commited can have worse consequences.

The results of AC_A are clearly favorable to $MEDBK$ (see figure 4.2(b)). This means that the order chosen by IG to select attributes in the decision process is not as logical medically as the order chosen by $MEDBK$. IG generates simple and efficient decision trees which are good in terms of accuracy (and usually in terms of AC_D too) but which are more incomprehensible by physicians than the decision trees generated with $MEDBK$.

Finally we observe the effect that the procedures of pruning and crossvalidation take to the results. Pruning provides generally smaller decision trees and usually improve the measure of AC_A because less attributes are needed to make a final decision. Nevertheless, in this domain, the results in AC_A with and without pruning are very similar. In the case of crossvalidation we evaluate the robustness of our decision structures. The decision tree is used to make decisions on patients which have not been used in the generation of the decision tree but results do not deteriorate drastically.

In figure 4.3, the decision tree obtained with pruning is depicted. If observe the order of selection of attribute we will see that *age* is the first one and $body - mass$ is the second one. According to the medical adherence partial order (which is the most relevant criterion in the selection of attributes) *age* is the most prioritary and $body - mass$ is the second one. According to the rest of criteria they also have low costs. Therefore, the order followed by our tree in the selection of attributes is very similar to the order that a

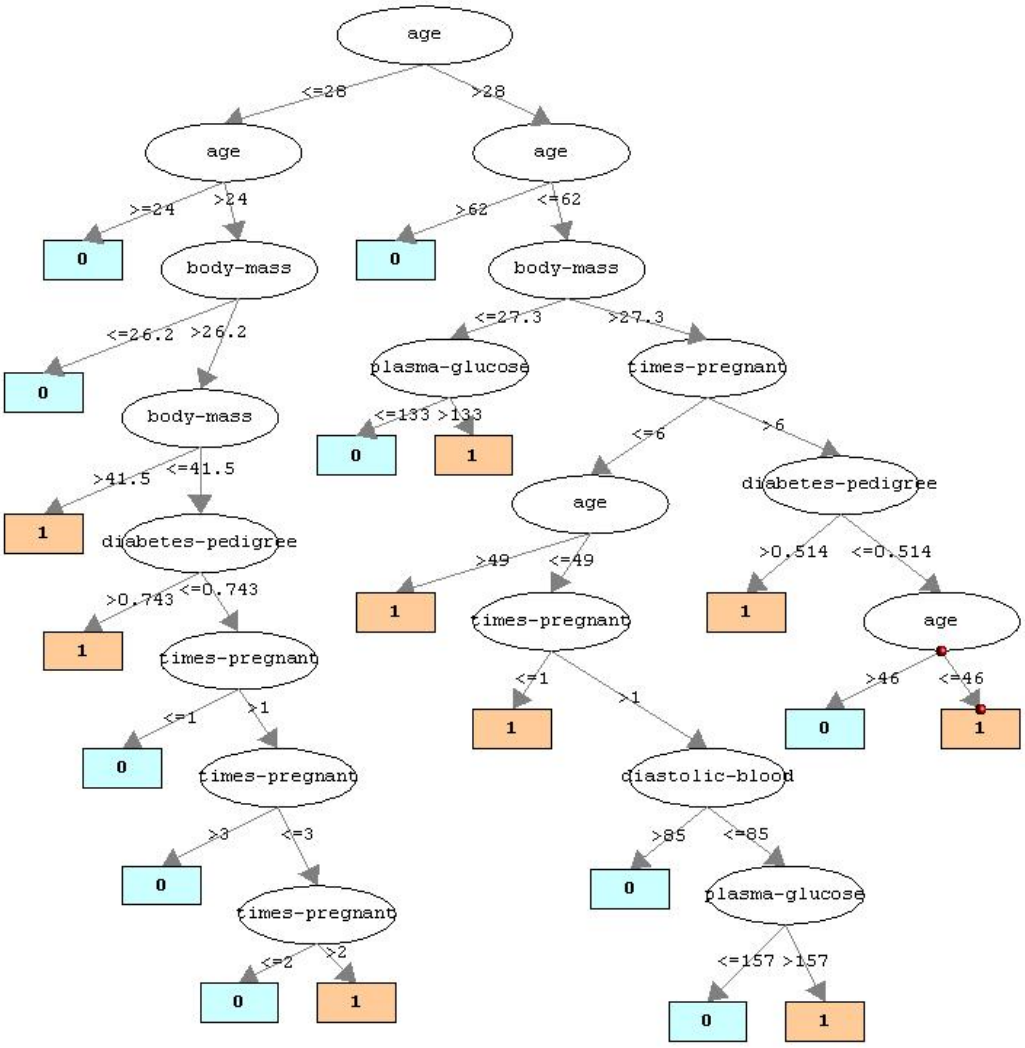


Figure 4.3: Decision tree obtained for the diabetes domain (with pruning)

physician would follow.

Medical aspects test

The results of the medical aspects test for the diabetes domain are shown in tables 4.20 and 4.21 where the versions of our approach ECO, MED and ACC and the information gain approach IG are compared to each other in terms of *Accuracy*, AC_A^{ECO}, AC_D^{ECO}, AC_A^{MED}, AC_D^{MED}, AC_A^{ACC} and AC_D^{ACC}. For each approach, the measures that have been minimized with a stronger priority (or maximized in the case of accuracy) are remarked in bold.

Table 4.20: Medical aspects test for the diabetes domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.850**	0.594	0.029	2.255	0.134	1.479	0.054
ECO	0.783	**0.223**	**0.050**	1.351	0.181	0.798	0.086
MED	0.837	0.449	0.042	**1.675**	**0.126**	1.282	0.070
ACC	0.813	0.282	0.032	1.708	0.177	**0.858**	**0.062**

Table 4.21: Medical aspects test (pruning) for the diabetes domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.826**	0.551	0.041	2.348	0.142	1.445	0.071
ECO	0.767	**0.176**	**0.049**	1.191	0.202	0.688	0.088
MED	0.776	0.315	0.061	**1.444**	**0.168**	0.983	0.099
ACC	0.763	0.135	0.047	1.464	0.210	**0.721**	**0.086**

In this test IG is the best as far as accuracy is concerned although all the approaches obtain similar results. The versions of our approach have not always been the best in the measure that they were trying to improve. Often, the minimum value for a certain measure is achieved by a version which gave more priority to other criteria. We can observe that each version V is usually one of the versions that obtains better results in terms of AC_A^V and AC_D^V but not always the best one. This is caused because, in spite of the fact that a certain version gives more weight to some criteria, it will not necessarily induce decision trees which are bad at the other criteria. In fact, some criteria are usually related to each other for some attributes or decisions. For example, the attribute *age* has a low cost in each criteria (it has no economic cost, it is instantaneous, it is neither risked nor uncomfortable and it is also a prioritary attribute according to medical adherence).

It is important to notice that, like happened in the general test, IG obtains clearly bad results in terms of AC_A^x. In 100% of the cases it is the worst approach in AC_A^x.

4.4 Heart disease domain

4.4.1 The dataset and the background knowledge

The aim of this domain is to detect the presence of heart disease in the patient. The attributes used are described in table 4.22.

Table 4.22: Decription of the attributes of the heart disease domain

	Description	*Values*
age	Age in years	integer
sex	Sex	1=male, 0=female
cp	Chest pain type	1=typical angina, 2=atypical angina, 3=non-anginal, 4=asymptomatic
trestbps	Resting blood pressure	integer
chol	Serum cholestoral in mg/dl	integer
fbs	Fasting blood sugar > 120 mg/dl?	1=true, 0=false
restecg	Resting electrocard. results	0=normal, 1=ST-T wave, 2=left vent. hypertrophy
thalach	Maximum heart rate achieved	integer
exang	Exercise induced angina	1=yes, 0=no
oldpeak	ST depression induced by exercise relative to rest	real
slope	Slope of the peak exercise ST segment	1=upsloping, 2=flat, 3=downsloping
ca	Number of major vessels	integer between 0 and 3
thal	Thal	3=normal, 6=fixed defect, 7=reversable defect

The possible final decisions are described in table 4.23.

Table 4.23: Decription of the decisions of the heart disease domain

	Description
0	$< 50\%$ diameter narrowing (angiographic disease status)
1	$> 50\%$ diameter narrowing (angiographic disease status)

Table 4.24: Tests needed for each attribute in the diabetes domain

	α	β	γ	δ	ϵ	ζ	η	θ
age	×							
sex		×						
cp			×					
$trestbps$				×				
$chol$					×			
fbs					×			
$restecg$						×		
$thalach$							×	
$exang$							×	
$oldpeak$							×	
$slope$							×	
ca								×
$thal$								×

The grouping of the attributes in tests is shown in table 4.24.

For attribute criteria, the final values for the cost functions K_e and K_t are detailed in table 4.25 and the final partial orders $\leq_m$, leq_h and $\leq_c$ are depicted in figure 4.4. In this case the partial orders leq_h and $\leq_c$ are identic.

For type I error criteria, we represent the knowledge using a cost function Ie_h depicted table 4.26.

For type II error criteria, the values for the cost function IIe_t are detailed in table 4.27.

Finally, the partial orders $\leq_{IIe-e}$, $\leq_{IIe-h}$ and $\leq_{IIe-c}$ for type II error criteria are identic. They are depicted in figure 4.5.

4.4.2 Results and analysis

For each test performed in the diabetes domain we have used the constants $\delta = 0.1$ and $\epsilon = 0.2$ for our algorithm.

General test

The results of the general test for the heart disease domain are shown in tables 4.28, 4.29, 4.30 and 4.31 where our approach $MEDBK$ is compared to the information gain approach IG in terms of $Accuracy$, AC_A and AC_D.

Table 4.25: The values of K_x for the heart disease domain

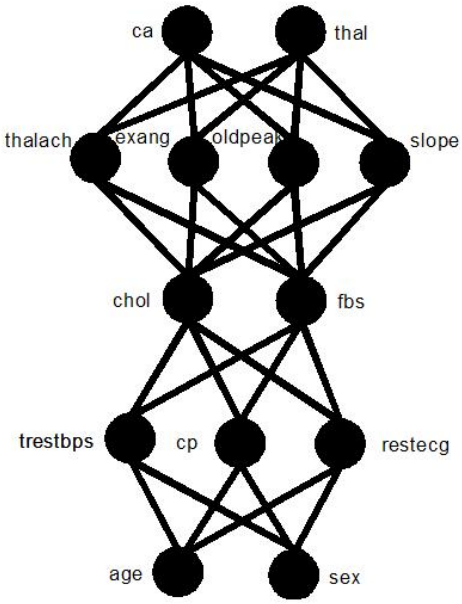
	K_e	K_t
age	0	0
sex	0	0
cp	0	0
$trestbps$	0	0.008
$chol$	0.063	1
fbs	0.063	1
$restecg$	0.041	0.042
$thalach$	0.083	0.25
$exang$	0.083	0.25
$oldpeak$	0.083	0.25
$slope$	0.083	0.25
ca	1	0.5
$thal$	1	0.5

Table 4.26: The values of Ie_h for the heart disease domain

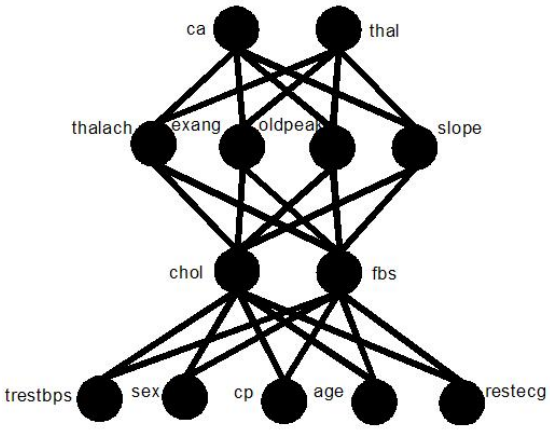
	0	1
Ie_h	0.25	1

Table 4.27: The values of IIe_x for the post-operative domain

	0	1
IIe_t	0	0



(a) $\leq_m$



(b) $\leq_h$ / $\leq_c$

Figure 4.4: The partial orders $\leq_x$ for the heart disease domain

Figure 4.5: The partial order $\leq_{IIe-e}$ / $\leq_{IIe-h}$ / $\leq_{IIe-c}$ for the heart disease domain

For each approach, the measures which are being minimized (or maximized in the case of accuracy) are remarked in bold.

Table 4.28: General test for the heart disease domain

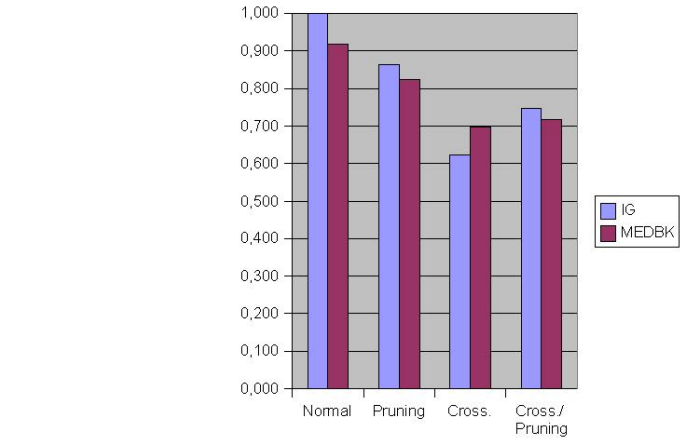
	Accuracy	AC_A	AC_D
IG	**1.000**	2.474	0.000
$MEDBK$	0.917	**0.357**	**0.046**

Table 4.29: General test (pruning) for the heart disease domain

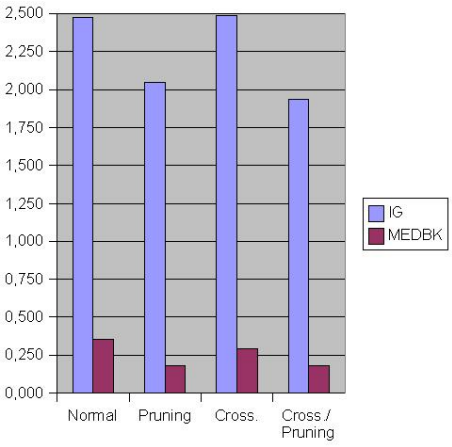
	Accuracy	AC_A	AC_D
IG	**0.865**	2.047	0.071
$MEDBK$	0.825	**0.181**	**0.095**

Like in the diabetes domain, the difference between both approaches in terms of accuracy is minimal (see figure 4.6(a)). Moreover, when making crossvalidation without pruning, $MEDBK$ increases the results of IG. In this domain, the results of AC_D are more correlated to the results of accuracy. The misclassification errors commited by IG are not extremely bad and, so, there are not cases where this approach is higher in accuracy but also higher in AC_D (see figure 4.6(c)).

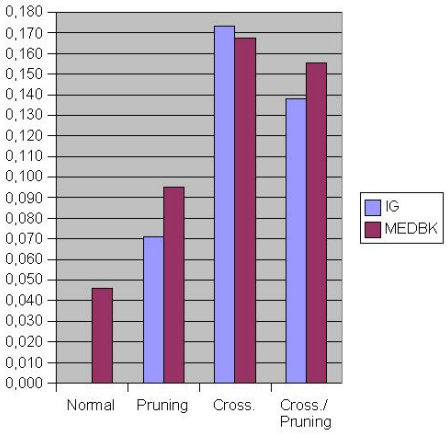
Where the difference can be more appreciated is analyzing the results of AC_A (see figure 4.6(b)). Although IG is able to make correct final decisions, the decision process followed is not easily comprehensible by physicians. $MEDBK$ can always reduce the cost of attributes in more than 1.7.



(a) Accuracy



(b) AC_A



(c) AC_D

Figure 4.6: Results of the general test on the heart disease domain

Table 4.30: General test (crossvalidation) for the heart disease domain

	Accuracy	AC_A	AC_D
IG	**0.624**	2.490	0.173
$MEDBK$	0.697	**0.293**	**0.168**

Table 4.31: General test (crossvalidation/pruning) for the heart disease domain

	Accuracy	AC_A	AC_D
IG	**0.748**	1.938	0.138
$MEDBK$	0.718	**0.179**	**0.155**

In the heart disease domain, the decision trees obtained with $MEDBK$ without pruning are more robust than the IG ones. The accuracy obtained by IG falls considerably when making the crossvalidation procedure. The decision trees obtained with this approach seem to overfit the dataset used in the generation and are not so good when dealing with unseen cases.

Figure 4.7 shows the decision tree obtained with pruning for the heart disease domain. The order of selection is very adequated to the medical background knowledge. For example, all the decisions are done without performing the test for *ca* and *thal* which is expensive, long, risked and uncomfortable.

Medical aspects test

The results of the medical aspects test for the diabetes domain are shown in tables 4.32 and 4.33 where the versions of our approach ECO, MED and ACC and the information gain approach IG are compared to each other in terms of *Accuracy*, AC_A^{ECO}, AC_D^{ECO}, AC_A^{MED}, AC_D^{MED}, AC_A^{ACC} and AC_D^{ACC}. For each approach, the measures that have been minimized with a stronger priority (or maximized in the case of accuracy) are remarked in bold.

As IG obtained an accuracy of 1.0 in the decision tree without pruning it is impossible that better results can be obtained in terms of accuracy and AC_D^x. When pruning is applied, the difference in accuracy and AC_D^x is reduced and even equalled in AC_D^{ECO}. On the contrary, the fact that IG does not consider background knowledge about the order of selecting attributes causes that once again it obtains the worst results in terms AC_A^x.

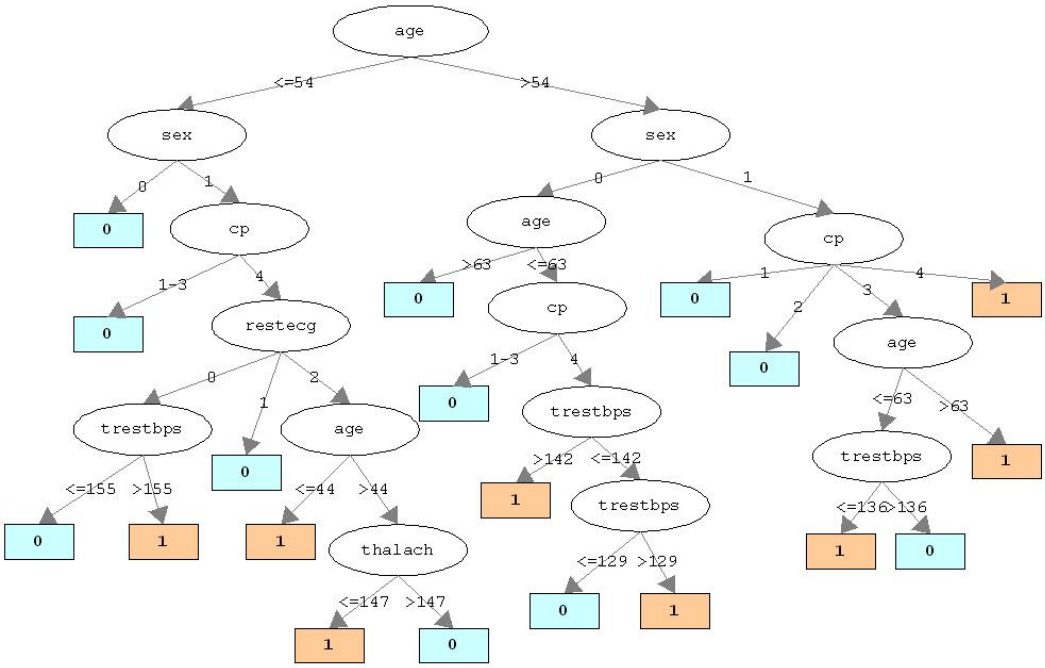


Figure 4.7: Decision tree obtained for the heart disease domain (with pruning)

Table 4.32: Medical aspects test for the heart disease domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**1.000**	2.034	0.000	2.870	0.000	2.366	0.000
ECO	0.914	**0.080**	**0.014**	0.521	0.048	0.208	0.028
MED	0.917	0.117	0.014	**0.471**	**0.046**	0.150	0.028
ACC	0.917	0.117	0.014	0.471	0.046	**0.150**	**0.028**

Table 4.33: Medical aspects test (pruning) for the heart disease domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.865**	1.652	0.021	2.355	0.071	1.749	0.031
ECO	0.825	**0.014**	**0.021**	0.247	0.950	0.035	0.306
MED	0.825	0.014	0.021	**0.247**	**0.950**	0.035	0.306
ACC	0.825	0.014	0.021	0.247	0.950	**0.035**	**0.306**

In this domain, one of the facts that is more evident is the similiary among ECO, MED and ACC. The partial orders of health risk and comfortability of the attributes are identical and very similar to the partial order of medical adherence (see figure 4.4). Therefore the decision trees obtained by MED and ACC are the same. The economy criteria are also similar to the other criteria. When pruning is applied the decision trees of ECO, MED and ACC are equal. Without pruning the results of ECO in AC_A^{ECO} (and MED/ACC in AC_A^{MED}/AC_A^{ACC}) are slightly better.

4.5 Post-operative domain

4.5.1 The dataset and the background knowledge

In this domain we deal with the decision problem of determining where the patients in a post-operative recovery area should be sent to next. The attributes used are described in table 4.34.

Table 4.34: Decription of the attributes of the post-operative domain

	Description	*Values*
$L-CORE$	Patient's internal temperature	high, mid, low
$L-SURF$	Patient's surface temperature	high, mid, low
$L-O2$	Oxygen saturation	excellent, good, fair, poor
$L-BP$	Last measurement of blood pressure	high, mid, low
$SURF-STBL$	Stability of patient's surface temperature	stable, mod-stable, unstable
$CORE-STBL$	Stability of patient's core temperature	stable, mod-stable, unstable
$BP-STBL$	Stability of patient's blood pressure	stable, mod-stable, unstable
$COMFORT$	Patient's perceived comfort at discharge	integer between 0 and 20

The possible final decisions are described in table 4.35.

Table 4.35: Decription of the decisions of the post-operative domain

	Description
I	Patient sent to Intensive Care Unit
S	Patient prepared to go home
A	Patient sent to general hospital floor

According to the background knowledge obtained from the physicians, the attributes are grouped in the six tests of table 4.36.

Table 4.36: Tests needed for each attribute in the post-operative domain

	α	β	γ	δ	ϵ	ζ
$L-CORE$	$\times$					
$L-SURF$	$\times$					
$L-O2$		$\times$				
$L-BP$			$\times$			
$SURF-STBL$				$\times$		
$CORE-STBL$				$\times$		
$BP-STBL$					$\times$	
$COMFORT$						$\times$

For attribute criteria, the final values for the cost functions K_e and K_t are detailed in table 4.37 and the final partial orders $\leq_m$ and $\leq_c$ are depicted in figure 4.8. As in the diabetes domain, the partial order $\leq_h$ is not shown because in this domain there not seem to be tests more risked than others.

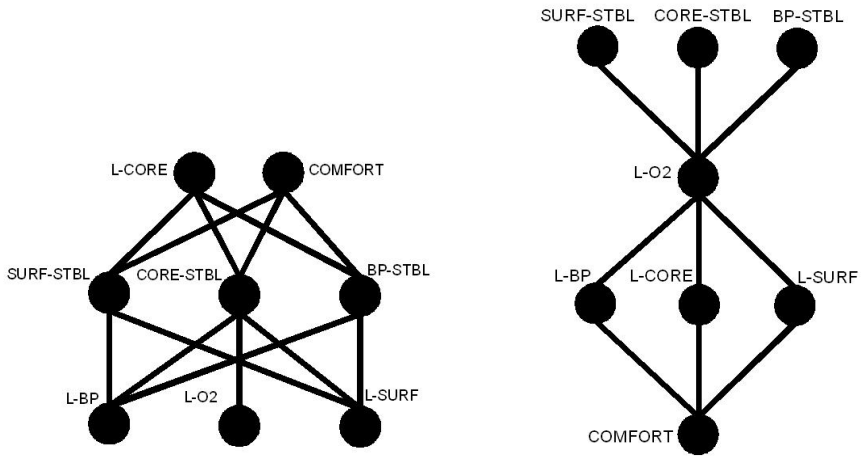
Table 4.37: The values of K_x for the post-operative domain

	K_e	K_t
$L-CORE$	0	0.02
$L-SURF$	0	0.02
$L-O2$	1	0
$L-BP$	1	0.03
$SURF-STBL$	0	1
$CORE-STBL$	0	1
$BP-STBL$	0	1
$COMFORT$	0	0

For type I error criteria, the final partial order $\leq_{Ie-h}$ is depicted in figure 4.9.

For type II error criteria, the values for the cost functions IIe_e and IIe_t are detailed in table 4.38.

Finally, the partial orders $\leq_{IIe-h}$ and $\leq_{IIe-c}$ corresponding to type II error criteria are depicted in figure 4.10.



(a) $\leq_m$ (b) $\leq_c$

Figure 4.8: The partial orders $\leq_x$ for the post-operative domain

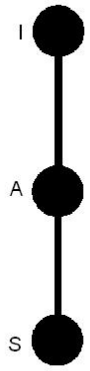
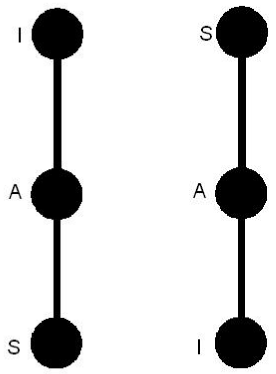


Figure 4.9: The partial order $\leq_{Ie-h}$ for the post-operative domain

Table 4.38: The values of IIe_x for the post-operative domain

	I	S	A
IIe_e	0.14	1	0.29
IIe_t	1	0	0.33



(a) $\leq_{IIe-h}$ (b) $\leq_{IIe-c}$

Figure 4.10: The partial orders $\leq_{IIe-x}$ for the post-operative domain

4.5.2 Results and analysis

For each test performed in the diabetes domain we have used the constants $\delta = 0.01$ and $\epsilon = 0.2$ for our algorithm.

General test

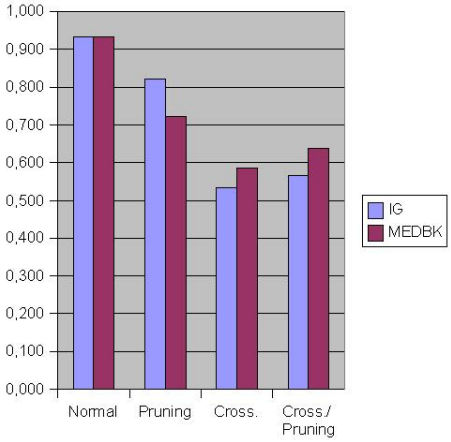
The results of the general test for the post-operative domain are shown in tables 4.39, 4.40, 4.41 and 4.42 where our approach $MEDBK$ is compared to the information gain approach IG in terms of *Accuracy*, AC_A and AC_D. For each approach, the measures which are being minimized (or maximized in the case of accuracy) are remarked in bold.

Table 4.39: General test for the post-operative domain

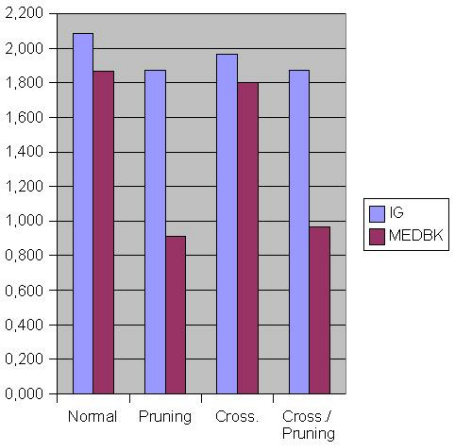
	Accuracy	AC_A	AC_D
IG	**0.933**	2.084	0.039
$MEDBK$	0.933	**1.867**	**0.017**

This domain is another example of the great similarity in accuracy of both IG and $MEDBK$ (see figure 4.11(a)). $MEDBK$ is better in two cases and in the other two, IG is better in one and they are equal in the other.

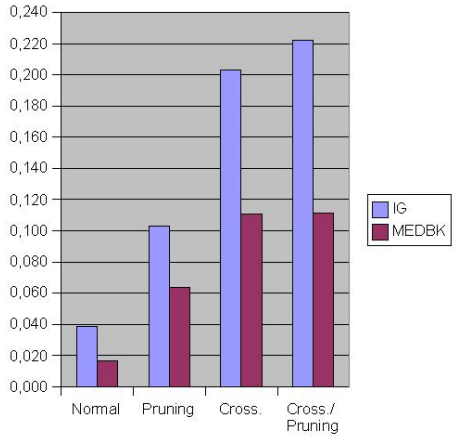
As far as costs are concerned, $MEDBK$ is definitively the best approach in the post-operative domain. The values of AC_A are minimal for $MEDBK$



(a) Accuracy



(b) AC_A



(c) AC_D

Figure 4.11: Results of the general test on the post-operative domain

Table 4.40: General test (pruning) for the post-operative domain

	Accuracy	AC_A	AC_D
IG	**0.822**	1.872	0.103
$MEDBK$	0.722	**0.911**	**0.064**

Table 4.41: General test (crossvalidation) for the post-operative domain

	Accuracy	AC_A	AC_D
IG	**0.533**	1.968	0.203
$MEDBK$	0.587	**1.800**	**0.111**

in each case with a difference from IG that ranges from 0.168 to 0.961 (see figure 4.11(b)). The difference is not as substantial as in other domains but it is considerable.

The same way, $MEDBK$ minimizes the AC_D in each case (see figure 4.11(c)). Thus, although IG is better than $MEDBK$ in accuracy in one case and equal in another case, the errors of misclassification commited are more critical.

The decision trees obtained from the post-operative domain are not very good at dealing with new data. Both IG and $MEDBK$ deteriorate their results in crossvalidation (IG in 0.4 and 0.255 and $MEDBK$ in 0.346 and 0.083). In this domain, the procedure of pruning reduces the costs of AC_A in almost 1.0 because they are smaller.

The decision tree obtained with pruning for the post-operative domain is depicted in figure 4.12. The order of selection of attributes is correct according to the background knowledge.

Table 4.42: General test (crossvalidation/pruning) for the post-operative domain

	Accuracy	AC_A	AC_D
IG	**0.567**	1.871	0.222
$MEDBK$	0.639	**0.965**	**0.111**

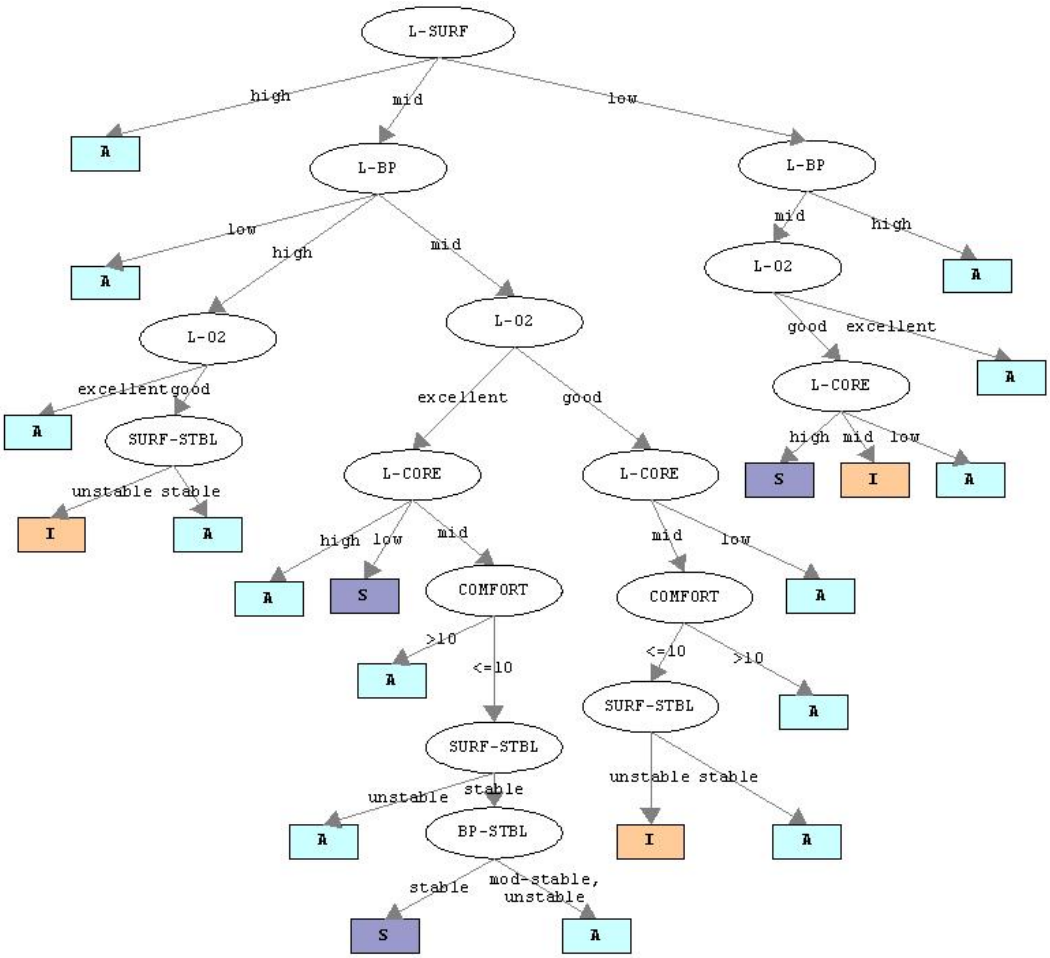


Figure 4.12: Decision tree obtained for the post-operative domain (with pruning)

Medical aspects test

The results of the medical aspects test for the post-operative domain are shown in tables 4.43 and 4.44 where the versions of our approach ECO, MED and ACC and the information gain approach IG are compared to each other in terms of *Accuracy*, AC_A^{ECO}, AC_D^{ECO}, AC_A^{MED}, AC_D^{MED}, AC_A^{ACC} and AC_D^{ACC}. For each approach, the measures that have been minimized with a stronger priority (or maximized in the case of accuracy) are remarked in bold.

Table 4.43: Medical aspects test for the post-operative domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.933**	1.333	0.010	2.331	0.039	2.528	0.016
ECO	0.933	**0.677**	**0.010**	3.532	0.017	2.352	0.017
MED	0.922	1.186	0.015	**1.446**	**0.017**	2.628	0.021
ACC	0.933	1.523	0.011	3.231	0.017	**1.924**	**0.017**

Table 4.44: Medical aspects test (pruning) for the post-operative domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.822**	1.263	0.032	2.064	0.103	2.319	0.044
ECO	0.744	**0.245**	**0.045**	2.671	0.058	1.290	0.066
MED	0.756	1.294	0.053	**0.783**	**0.053**	1.726	0.070
ACC	0.767	0.655	0.042	2.726	0.072	**1.333**	**0.061**

The post-operative domain is where the versions (ECO, MED and ACC) of our methodology are more different to each other. In 79% of the cases, the approach which minimizes (maximizes in the case of accuracy) each measure is the one that is actually giving more importance to the respective medical aspect (i.e., the cells remarked in bold are usually the minimum (maximimum in the case of accuracy) of their respective columns). This is because in this domain the different criteria are not as correlated as in others. For example, the order of the attributes according to the medical adherence is not very similar to the order determined by the rest of attribute criteria. The attribute $COMFORT$ is a question that the patient has to answer. It is a very prioritary attribute with respect to some criteria like economic cost or response time because it requires a cheap and instantaneous test. But, according to the knowledge of the physicians, it is not a prioritary decision to make this question. In the results we can appreciate these differences. When

we use the ECO version we always minimize the AC_A^{ECO} while obtaining the worst results in AC_A^{MED}. The same way, the MED version obtains the best results in AC_A^{MED} but the maximum values of AC_A^{ECO}.

4.6 Thyroid domain

4.6.1 The dataset and the background knowledge

In this domain the patients are classified according to their thyroid functioning into three classes: normal (not hypothyroid), hyperfunction or subnormal functioning. The attributes used are described in table 4.45.

Table 4.45: Decription of the attributes of the thyroid domain

	Description	*Values*
age	Age normalized	real
sex	Sex	1=male, 0=female
on_thyroxine	On thyroxine	0=false, 1=true
query_on_thyroxine	Query on thyroxine	0=false, 1=true
on_antithyroid_medication	On antithyroid medication	0=false, 1=true
sick	Sick	0=false, 1=true
pregnant	Pregnant	0=false, 1=true
thyroid_surgery	Thyroid surgery	0=false, 1=true
*I*131*_treatment*	I131 treatment	0=false, 1=true
query_hypothyroid	Query hypothyroid	0=false, 1=true
query_hyperthyroid	Query hyperthyroid	0=false, 1=true
lithium	Lithium	0=false, 1=true
goitre	Goitre	0=false, 1=true
tumor	Tumor	0=false, 1=true
hypopituitary	Hypopituitary	0=false, 1=true
psych	Psychological symptoms	0=false, 1=true
TSH	Thyroid-stimulating hormone	real
$T3$	Triiodothyronin	real
$TT4$	Total thyroxine	real
$T4U$	T4 uptake	real
FTI	Free thyroxine index	real

The possible final decisions are described in table 4.46.

Only a group of 5 attributes are obtained with the same test (see table 4.47).

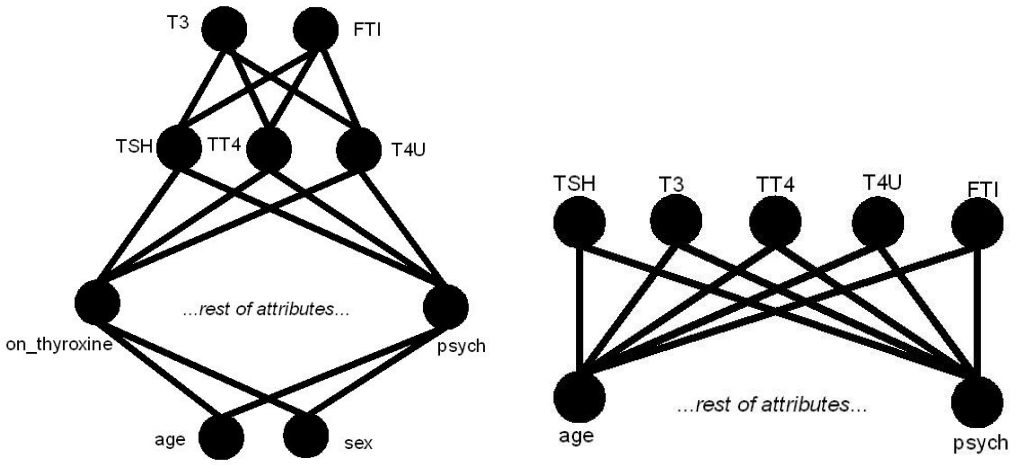
Table 4.46: Decription of the decisions of the heart disease domain

	Description
1	normal (not hypothyroid)
2	hyperfunction
3	subnormal functioning

Table 4.47: Tests needed for each attribute in the diabetes domain

	α	β	γ	δ	ϵ	ζ	η	θ	ι	κ	λ	μ	ν	ξ	o	π	ρ
age	×																
sex		×															
on_thyroxine			×														
query_on_thyroxine				×													
on_antithyroid_medication					×												
sick						×											
pregnant							×										
thyroid_surgery								×									
*I*131*_treatment*									×								
query_hypothyroid										×							
query_hyperthyroid											×						
lithium												×					
goitre													×				
tumor														×			
hypopituitary															×		
psych																×	
TSH																	×
*T*3																	×
*TT*4																	×
T4U																	×
FTI																	×

For attribute criteria we have only used partial orders. The partial orders $\leq_e$, $\leq_t$ and $\leq_c$ are the same and they are depicted in figure 4.13 beside the partial order $\leq_m$. For space reasons we have not included all the attributes in the figure. We do not show the partial order $\leq_h$ because there are no risked tests.



(a) $\leq_m$ (b) $\leq_e$ / $\leq_t$ / $\leq_c$

Figure 4.13: The partial orders $\leq_x$ for the thyroid domain

For type I error criteria, we represent the knowledge using a cost function Ie_h depicted table 4.48.

Table 4.48: The values of Ie_h for the thyroid domain

	1	2	3
Ie_h	1	1	1

For type II error criteria we only use cost functions (see table 4.49).

4.6.2 Results and analysis

For each test performed in the thyroid domain we have used the constants $\delta = 0.07$ and $\epsilon = 0.2$ for our algorithm.

Table 4.49: The values of IIe_x for the thyroid domain

	1	2	3
IIe_e	0	1	1
IIe_t	1	1	1
IIe_h	1	1	1
IIe_c	0.5	1	1

General test

The results of the general test for the thyroid domain are shown in tables 4.50, 4.51, 4.52 and 4.53 where our approach $MEDBK$ is compared to the information gain approach IG in terms of *Accuracy*, AC_A and AC_D. For each approach, the measures which are being minimized (or maximized in the case of accuracy) are remarked in bold.

Table 4.50: General test for the thyroid domain

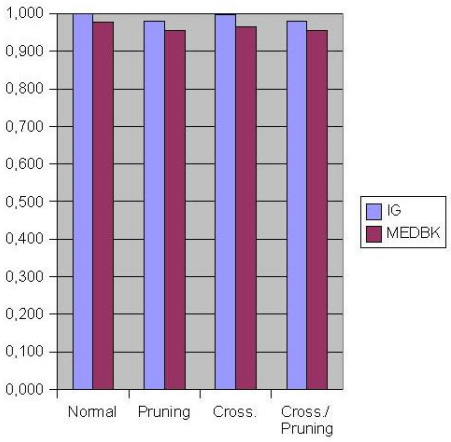
	Accuracy	AC_A	AC_D
IG	**0.999**	1.096	0.000
$MEDBK$	0.977	**1.548**	**0.023**

Table 4.51: General test (pruning) for the thyroid domain

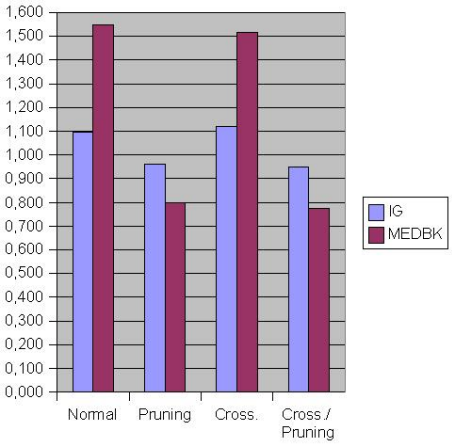
	Accuracy	AC_A	AC_D
IG	**0.979**	0.961	0.021
$MEDBK$	0.955	**0.798**	**0.045**

This domain is somewhat special because it is extremely unbalanced. Concretely, 92.5% of the patients are classified as having a thyroid subnormal functioning. Therefore, all of the cases obtain an accuracy greater than this value (see figure 4.6(a)). In terms of accuracy both approaches obtain similar results (less than 0.032 of difference).

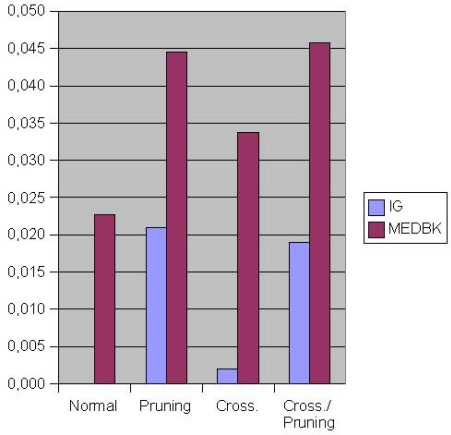
Considering the costs of the misclassification errors (AC_D), we are always dealing with decision trees that commit a very few errors, thus, they are minimal (always under 0.046) (see figure 4.6(b)). In this domain, although IG does not consider background knowledge in the generation of the decision



(a) Accuracy



(b) AC_A



(c) AC_D

Figure 4.14: Results of the general test on the thyroid domain

Table 4.52: General test (crossvalidation) for the thyroid domain

	Accuracy	AC_A	AC_D
IG	**0.998**	1.120	0.002
$MEDBK$	0.966	**1.517**	**0.034**

Table 4.53: General test (crossvalidation/pruning) for the thyroid domain

	Accuracy	AC_A	AC_D
IG	**0.981**	0.951	0.019
$MEDBK$	0.954	**0.776**	**0.046**

trees it achieves better results for AC_D than $MEDBK$. This is caused because of the great values of accuracy and the fact that there are not specially critical misclassification errors.

At the level of AC_A our approach is the best when pruning is applied (see figure 4.6(c)). When the decision trees are not pruned, IG obtains better results. In cases like these, where IG is better at AC_A than our methodology, the explanation is that we probably are falling in a local minimum. We are generating decision tree with greedy algorithms (i.e., algortihms that choose the best option in each node in accordance to some criterion) and sometimes selecting a seemingly better attribute in a certain node can lead to a worse final decision tree.

Because of the fact that the great majority of the dataset belong to a certain class, the crossvalidation does not change substantially the conditions on which the decision tree is generated and evaluated. Thus, very similar results are obtained.

In figure 4.15, we can observe the decision tree obtained with pruning. The order specified by the background knowledge is once again respected causing a more logical decision process. The more prioritary attributes like *age*, *sex*, etc. are used soon and hence only a subset of patients are required to perform the most expensive test that provides the values of TSH.

Medical aspects test

The results of the medical aspects test for the thyroid domain are shown in tables 4.54 and 4.55 where the versions of our approach ECO, MED and ACC and the information gain approach IG are compared to each other in

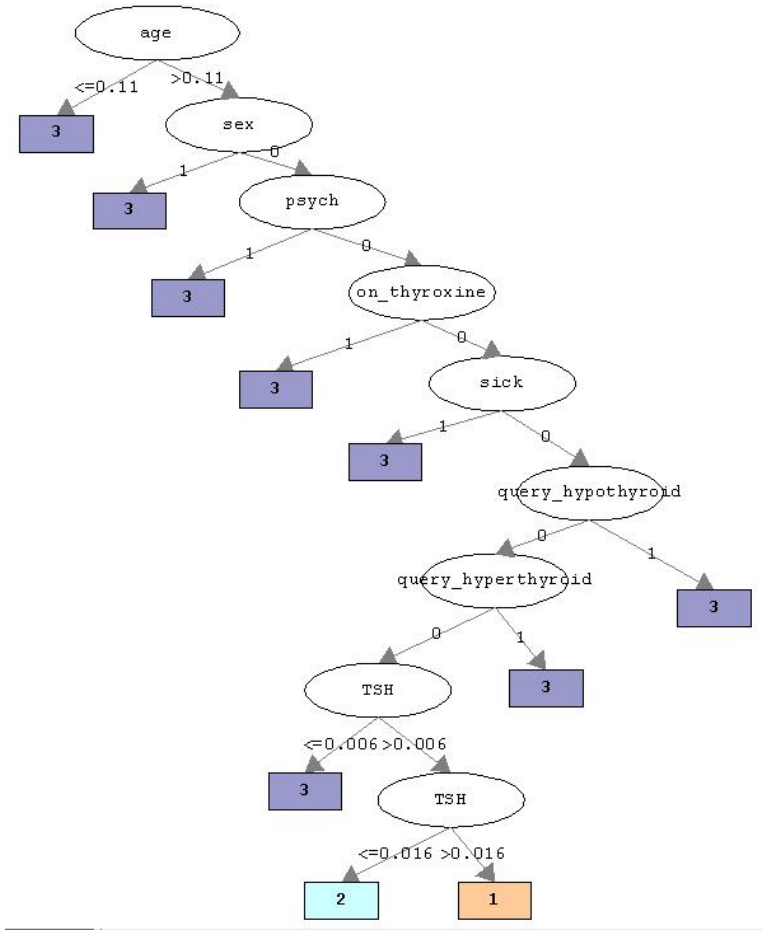


Figure 4.15: Decision tree obtained for the thyroid domain (with pruning)

terms of *Accuracy*, AC_A^{ECO}, AC_D^{ECO}, AC_A^{MED}, AC_D^{MED}, AC_A^{ACC} and AC_D^{ACC}. For each approach, the measures that have been minimized with a stronger priority (or maximized in the case of accuracy) are remarked in bold.

Table 4.54: Medical aspects test for the thyroid domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.999**	1.000	0.000	0.946	0.000	1.173	0.000
ECO	0.977	**0.504**	**0.011**	3.240	0.023	1.869	0.015
MED	0.977	0.504	0.011	**3.240**	**0.023**	1.869	0.015
ACC	0.977	0.504	0.011	3.240	0.023	**1.869**	**0.015**

Table 4.55: Medical aspects test (pruning) for the thyroid domain

	Accuracy	AC_A^{ECO}	AC_D^{ECO}	AC_A^{MED}	AC_D^{MED}	AC_A^{ACC}	AC_D^{ACC}
IG	**0.979**	1.000	0.018	0.782	0.021	0.957	0.013
ECO	0.955	**0.471**	**0.020**	1.603	0.045	1.172	0.027
MED	0.955	0.471	0.020	**1.603**	**0.045**	1.172	0.027
ACC	0.955	0.471	0.020	1.603	0.045	**1.172**	**0.027**

The thyoid domain is where we deal with more correlated criteria. In fact, the three decision trees created with ECO, MED and ACC are equal in both cases. We observed in the previous section that there are a lot of attributes with a very high priority and 5 attributes with a higher cost. Moreover, these 5 attributes are obtained with the same test so once the test is performed all of them reduce their cost. This leads to a very equalled priority for all the attributes. With decisions happens the same because there are not specially critical decisions.

The decision trees generated by IG are better in almost all the aspects. Although IG is only considering the information gain it is not necesarily maximizing the costs. Usually this fact causes that IG is worse than $MEDBK$ in AC_A but, in the thyroid domain, the decision trees obtained are good at the level of costs.

Chapter 5

Conclusions and future work

We have developed a model for including the background knowledge from the physicians in the automatic generation of medical decision structures.

We observed that the current approaches used to generate decision trees are not successful when are applied in complex domains like medicine. The information gain approach is exclusively centered in obtaining decision structures that are simple and take correct decisions but which are usually medically incomprehensible. Moreover, it does not consider differences among the possible misclassification errors and this fact, in medicine, can lead to critical errors. Several approaches have been done that partially solve these problems including a cost function to minimize or a partial order to take in account.

We have identified drawbacks from each of the approaches and defined a model that tries to solve them. Our model includes several kinds of important medical criteria which are applied when selecting which attribute to obtain from the patient and when making a final decision. Moreover the physician is allowed to give different priority and relevance to each of the criteria.

After testing the model ($MEDBK$) with real data from the domains of diabetes, heart disease, post-operative and thyroid we can conclude that:

- The results obtained in accuracy have always been very similar than the ones of a approach only based on information gain (IG)(always a difference lower than 0.1).

- Usually the results in AC_D (average cost of the decisions) are similar for IG and $MEDBK$ (the maximum difference is 0.161 and it is favorable to $MEDBK$). In some cases where IG approach obtained a better

accuracy than $MEDBK$, its AC_D was greater. This is because we are considering that each misclassification error has a different degree of relevance according to several medical criteria, while IG does not. Therefore, in these cases, although IG makes more correct decisions, the bad decisions made are more critical medically.

- The results on AC_A are clearly better for $MEDBK$ (always better for each domain except for thyroid which is only better for 50% of the cases). This fact means that the decision trees generated with our approach follow a more medically coherent decision process.

- The behaviour of $MEDBK$ in the procedures of crossvalidation and pruning is similar than with IG. With the crossvalidation the accuracy is deteriorated but for the diabetes and the thyoid domains the differences are not substantial. In the heart disease and post-operative domain $MEDBK$ seems to be more able to deal with new data because, although IG is more accurate without crossvalidation, our methodology is more accurate with crossvalidation. The pruning produces more brief and generalized trees which always improve the AC_A and usually obtain a slightly worse results in *accuracy* and AC_D.

- The greedy algorithms like $MEDBK$ and IG have the drawback that they can fall into a local minima. This happens, for example, in the unpruned decision trees of the thyroid domain where $MEDBK$ obtains worse results than IG in AC_A. The same way in 41.67% of the cases $MEDBK$ has equal or better results than IG in accuracy. Therefore, in spite of the fact that a greedy algorithm is minimizing in each step a certain criterion it can not assure that it will obtain the global minimum on this criterion. Minimizing the accuracy (the medical costs) does not necessarily mean maximizing the medical costs (the accuracy).

Moreover, we have also tested 3 versions of our methodology oriented to give more priority to the economy (ECO), the medical aspects (MED) and the acceptability of the patient (ACC) respectively with the IG approach and we have observed that:

- There are usually medical criteria which are correlated. In the domains of heart disease and thyroid we have obtained for the 75% of the cases the same decision tree for each methodology.

- In the post-operative and diabetes domain the criteria are less correlated and, so, the version which minimizes the economy / medical / acceptability costs is in 66.67% the version which is giving more priority to economy / medical / acceptability criteria. This does not happen for the rest of cases because, the fact that a version is minimizing an aspect does not always mean that is maximizing another aspect.

- Usually, IG obtains worse results for AC_A^{ECO}, AC_A^{MED} and AC_A^{ACC} than any of our versions. In the diabetes and heart disease domains, this happens the 100% of the cases.

There are lots of aspects of the methodology which can be reconsidered and even improved in the future. We may identify new medical criteria which are also involved in the decision process.

Another important point is the combination of criteria. This is a complex scope where we deal with different knowledge structures that have different priorities and relevances and this multi-criteria decision problem can be solved by several procedures which might be more appropiated.

Another interesting point of research is to solve the problem of the local minima. An option would be to work with algorithms of generation of decision trees which were not greedy and that could assure a global minimum.

Although we have performed lots of tests on several domains, we work with a model which is highly parametrized and thus, we still need to perform more tests over the same and new domains in order to analyze its behaviour.

Our intention is to include this model in the automatic generation of SDA* clinical algorithms [29]. This is a domain which involves the generation of decision trees which need to be medically coherent both at the level of the final decisions and at the level of the sequential process of decision.

Finally, within the scope of this thesis, we have also introduced different mathematical operations over partial orders (sections 2.5.3, 2.5.4, 2.5.7 and 2.5.8) which, as far as we are aware, have not been defined before and which can be used in several domains where partial orders are needed.

Bibliography

[1] Bohada, J.A., Riaño, D., Real, F.: Induction of partial orders to predict patient evolutions in medicine *AIME 2007, LNAI 4594, pp. 489-499. 2007.*

[2] Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees. *California: Wadsworth. 1984.*

[3] Chai, X., Deng, L., Yang, Q., Ling, C.X.: Test-Cost Sensitive Nayïve Bayesian Classification. *Proceedings 4th IEEE Intl Conf. Data Mining. 2004.*

[4] Fawcett, T., Provost, F.J. Activity monitoring: Noticing interesting changes in behavior. *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, KDD-99. 1999.*

[5] Fawcett, T., Provost, F.J.: Adaptive fraud detection. *Data Mining and Knowledge Discovery, 1 (3). 1997.*

[6] Fawcett, T., Provost, F.J.: Combining data mining and machine learning for effective user profiling. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD-96, pp. 8-13. 1996.*

[7] Freitas, A., Costa-Pereira, A., Brazdil, P.: Cost-sensitive decision trees applied to medical data *DaWaK 2007, LNCS 4654, pp.303-312. 2007.*

[8] Hermans, J., Habbema, J.D.F., Van der Burght, A.T.: Cases of doubt in allocation problems, k populations. *Bulletin of the International Statistics Institute, 45, 523-529. 1974.*

[9] Holland, J. H.: Adaptation in Natural and Artificial Systems, *University of Michigan Press, Ann Arbor. 1975*

[10] Insall, M., Loy, J., Weisstein, E.W.: MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/`

[11] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review *ACM Computer Surveys 31(3): 264-320. 1999.*

[12] James, M.: Classification Algorithms *Wiley-Interscience. 1986.*

[13] Kuperman, G., Gardner, R., Pryor, T.: HELP: A dynamic hospital information system. *New York: Springer Verlag. 1991.*

[14] Ling, C.X., Sheng, V.S., Yang, Q.: Test Strategies for Cost-Sensitive Decision Trees. *IEEE transaction on knowledge and data engineering, vol. 18, no. 8, August. 2006.*

[15] Ling, C.X., Yang, Q., Wang, J., Zhang. S.: Decision Trees with Minimal Costs. *Proceedings 21st Intl Conf. Machine Learning. 2004.*

[16] López-Vallverdú, J.A., Riaño, D., Collado, A.: Increasing acceptability of decision trees with domain attributes partial orders. *Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems, Maribor, Slovenia. 2007.*

[17] MacQueen, J.: Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Volume I, Statistics. Edited by Lucien M. Le Cam and Jerzy Neyman. University of California Press. 1967.*

[18] Michie, D.: On machine intelligence (2nd ed.). *Chichester, UK: Ellis Horwood. 1986.*

[19] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning databases
`http://www.ics.uci.edu/~mlearn/MLRepository.html`

[20] Nuñez, M.: Economic induction: A case of study. *Proceedings of the Third European Working Session on Learning, EWSL-88, pp. 139-145. California: Morgan Kaufmann. 1988.*

[21] Nuñez, M.: The use of background knowledge in decision tree induction. *Machine Learning, 6, 231-250. 1991.*

[22] Patil, G.P., Taillie, C.: Multiple indicators, partially ordered sets, and linear extensions: Multi-criterion ranking and prioritization. *Environmental and Ecological Statistics 11, 199-228, 2004.*

[23] Peek, N., Verduijn, M., Sjoe-Sjoe, W. T., Rosseel, P., de Jonge, E., de Mol, B.: ProCarSur: A System for Dynamic Prognostic Reasoning in Cardiac Surgery. *Artificial Intelligence in Medicine, 2007, 4594, 336-340. 2007.*

[24] Pemmaraju, S., Skiena S.: Computational Discrete Mathematics. *Cambridge University Press, 2003.*

[25] Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I.: Decision trees: an overview and their use in medicine. *J Med Syst. 26(5):445-63. 2002.*

[26] Quinlan, J.R.: C4.5: Programs for Machine Learning. *Morgan Kaufmann, San Mateo, CA., USA. 1993.*

[27] Quinlan, J.R.: C5.0 Online Tutorial. `http://www.rulequest.com`. *2003.*

[28] Quinlan, J.R.: Induction of decision trees. *Machine Learning, 1(1):81-106. 1986.*

[29] Riaño, D., López-Vallverdú, J.A., Tu, S.: Mining Hospital Data to Learn SDA* Clinical Algorithms *LNCS 4924, pp.46-61. 2008.*

[30] Shannon, C., Weaver, W.: The mathematical theory of communication. *University of Illinois Press, Urbana, IL, USA. 1948.*

[31] Shortliffe, E.: Computer-Based Medical Consultations: MYCIN. *Artificial Intelligence Series. New York: Elsevier, 1976.*

[32] Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning, 13, 7-33. 1993.*

[33] Tan, M.: Cost-sensitive reinforcement learning for adaptive classification and control. *Proceedings of the Ninth National Conference on Artificial Intelligence, 774-780. San Jose, CA: AAAI Press. 1991.*

[34] Tan, M.: Learning a cost-sensitive internal representation for reinforcement learning. *Proceedings of the Eighth International Workshop on Machine Learning, 358-362. Evanston, IL: Morgan Kaufmann. 1991.*

[35] Turney, P.D.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research, 2, 369-409. 1995.*

[36] Turney, P.D.: Types of cost in inductive concept learning. *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning, Stanford University, California. 2000.*

[37] Warner, H., Haug, P., Bouhaddou, O., et al.: ILIAD as an expert consultant to teach differential diagnosis. *Proceedings of the Twelfth Annual Symposium on Computer Applications in Medical Care, New York: IEEE Computer Society Press, 1987, 371-377. 1987.*

[38] Xu, R., Wunsch, D.: Survey of clustering algorithms *IEEE transactions on Neural Networks 16(3):645-678, 2005.*