Application of expertise acquisition in savings plans

Philippe Lenca

Application of expertise acquisition in savings plans

Philippe Lenca

Département Intelligence Artificielle et Systèmes Cognitifs, Télécom Bretagne
Technopôle de Brest Iroise, BP 832 - 29285 Brest Cedex, France
E-mail: Philippe.Lenca@enst-bretagne.fr

Abstract

In this paper we propose to develop a method for learning strategies established by a decision maker for a feature task of the selection/elimination of objects described by several attributes (parameters ...). This method is based upon a cognitive model of the decision maker stemming from psychological research. We will explain its mathematical and algorithmical consequences. We will present an algorithm (and a way of coding the objects) which extracts rules used by the decision maker. The complexity and efficiency of the algorithm will be discussed. It will be illustrated with a concrete example and real data in the last section.

Keywords: Knowledge & Expertise Acquisition, Machine Learning.

1 Introduction

We assume that a decision maker uses expertise and stable strategies to evaluate objects of a world O (for instance: situations or alternatives, etc) for a specific task (categorization, selection/elimination ...) within his/her domain of expertise. In this paper, we will consider a specific task undertaken by the decision maker: linearly ordered categorization; especially a selection/elimination task that defines the categorization problem with 2 separate clusters (the world of objects O is clustered in 2 separated sets: the "accepted" objects and the "rejected" objects). This task is executed on objects of the world O which are relevant for the decision maker and supports his/her knowledge. An object is described by a set of attributes and also has a natural representation in a multi-attribute space. We want to learn rules (conjunction/disjunction on the attributes) which explain the decision maker's choices. These strategies are assumed to be stored in his/her long-term memory and may be rather complex; they have been constructed by his/her experience ... First, we will explain the psychological model. We will also explain what are our constraints and show the mathematical and algorithmical consequences of the model. In the last part, we will show an application for extracting the individual rules of choice bet savings plans.

2 Psychological model

We use the Moving Basis Heuristic model, which stems from decision and judgment psychological research (see Barthélemy and Mullet [3] [4]). This model is based upon a multi-attribute representation space describing the set of objects O with several attributes (which can take several values). The model assumes that the decision maker shows rationality (for situations in his/her usual domain of expertise) in the way that something is optimized. But this **rationality is bounded** (see Simon [18]) by: his/her cognitive abilities (short-term memory and computing capacity) and his/her satisfaction (pleasure, risk ...) in performing the task.

This bounded rationality constrains the expert to search among aspects (attribute values) for a short subcollection: a **dominance structure** (see Montgomery [14], Montgomery and Svenson [15] and Svenson [19]). This dominance structure is limited but large enough to achieve decisions (the data are processed in the short-term memory).

This model involes three cognitive principles:

(1) **parsimony**: the decision maker manipulates a short subset of aspects which is a dominance structure due to his/her short-term memory capacity (storage capacity: there is no intermediate storage in the long-term memory, and computational abilities), Aschenbrenner and Kasubek [2], Johnson and Payne [10],

(2) **reliability/warrantability**: the chosen subcollection of aspects has to be large enough for individual or social justification, Adelbratt *et al.* [1], De Hoog and Van de Wittenboer [7], Montgomery [14], Ranyard *et al.* [17],

(3) **decidability/flexibility**: the decision maker must effect a choice by appropriate changes in dominance structure election until a decision is

taken (he/she has to achieve decision quickly in almost all cases), Huber [9], Montgomery [14] and Svenson [19].

The model also assumes a **monotonicity property** (threshold hypothesis): when an object is selected (respectively rejected), then every object that has better (respectively worse) values on all its attributes is selected (respectively rejected): the decision maker uses thresholds on aspects in order to take a decision.

Note that this principle, which implies that the attributes and categories must be linearly ordered, is close to the parsimony principle (an object is selected if each aspect of the current dominance structure used by the expert is over a minimal threshold on focused attributes) and implies that strategies are defined by a sequence of threshold objects, so we can use the same representation space for coding objects or strategies.

3 Mathematical modeling

3.1 Some notions about ordered sets

Let E be a set.

Partial order: a partial order on E is a binary relation $\leq_E$ that fulfils, $\forall\ x, y, z \in E$, reflexivity ($x \leq_E x$), antisymmetry (if $x \leq_E y$ and $y \leq_E x$ then $x = y$) and transitivity (if $x \leq_E y$ and $y \leq_E z$ then $x \leq_E z$) properties. If there no confusion possible we will now just write $x \leq y$ instead of $x \leq_E y$.

Partially ordered set or poset: a set structured with a partial order.

Chain: an order on E that satisfies a fourth property called completeness ($\forall\ x, y \in E$, either $x \leq y$ or $y \leq x$) is called a chain. This means that in a chain, or a linear order, we can always compare 2 elements.

Direct product of posets: let p sets $P_1, ..., P_p$, $P = P_1 \times ... \times P_p$ is their direct product and an element x of P is a point with p coordinates $(x_1, ..., x_p)$ where $x_i \in P_i$. If each P_i is linearly ordered then $(P, \geq)$ is a partially ordered set where $\geq$ is the direct product order defined by: $\forall\ x, y \in P, x \geq y$ iff $\forall\ i \in [1..p]\ x_i \geq_{P_i} y_i$.

Antichain: a subset A of a poset E is an antichain of E, if $\forall\ x, y \in A$, $x \leq y$ iff $x = y$ (2 distinct elements of A are not comparable).

Covering relation : in E, x covers y and is noted $x \succ y$ if $\forall\ z \in E$, $y < x$ and if $y \leq z < x$ then $y = z$ (x is the smallest element greater than y).

We can see an example of a direct product of two linear orders in figure 1.

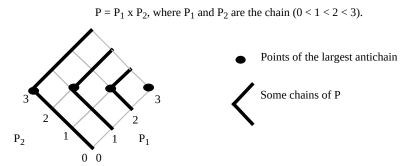

Figure 1: Direct product of 2 posets

3.2 Consequences of the decision maker's preferences

Each object o of O is described by p attributes X_i and is partially ordered by the decision maker's preferences. Thus O has a "natural" representation in a multi-attribute space P with p dimensions.

These attributes are linearly ordered on a domain value $P_i = \{0 \leq 1 \leq ... \leq c_i\}$ (ie. each attribute can take $c_i + 1$ different values). Note that the P_i are linearly ordered to conserve the order between objects in O due to the expert's preferences on each attribute. They are in fact a way of coding the "real" domain values taken by X_i; X_i can take different values (discrete values or intervals) in a domain V_i.

So, P is the direct product of the p linear orders P_i and an object o of O is represented by $x = (x_1, ..., x_p) \in P$, where x_i is the value taken by o on the attribute X_i in the domain P_i.

The objects must be assigned to the category C_1 (selected objects) or to C_2 (rejected objects); thus there is no "no choice". The monotonicity principle is translated in P with the following "propagation" rules:

$$\forall x \in P, \text{ if } x \in C_1 \text{ then } \forall y \in P \mid y \geq x \Rightarrow y \in C_1,$$
$$\forall x \in P, \text{ if } x \in C_2 \text{ then } \forall y \in P \mid y \leq x \Rightarrow y \in C_2.$$

Read this rule: "if an object o of O (represented by x in P) is accepted by the decision maker then all objects o' (represented by x' in P) that are better than o are accepted too; if o is rejected then all objects that are worst than o are rejected too". So $C_1 \geq C_2$ (the categories are ordered) and P is split into 2 covering and disconnected parts (there is no $(x, y) \in P^2$ with $x \in C_1$, $y \in C_2$ and $y \geq x$).

We can now see that to find the strategies used by the expert is equivalent to computing the set of minimal elements of C_1. Each minimal object summarizes a minimal threshold set on attributes that is equivalent to a dominance structure. This is equivalent to seeking an antichain (Pichon *et al.* [16]) of P. In general, if there are c linearly ordered categories we have to find $c - 1$ antichains in P.

This problem is similar of the learning concept problem in version space (see Mitchell [13]).

3.3 Modeling consequences

To find an antichain of P: if x is in C_1, y is in C_2 (x, y points of P) and x covers y, then x represents a rule of selection. Then, for all $(x, y) \in P^2 \mid x \in C_1$, $y \in C_2$ and $x \succ y$ we have to calculate the minimum x's to compute the antichain.

Combinatorial: insofar as we create all the possible combinations between attributes the number of points n of P ($n = \prod_{i=1}^{p}(c_i + 1)$) can be large. This will be considered for the number of points to compute for the learning algorithm. This suggests that we will not do an exhaustive search of the antichain (especially in the case of interactive knowledge extraction). We can use a greedy algorithm which does not go back and does not "allow" the contradictions (we always propose a point of P which can be assigned to C_1 or C_2).

Cognitive monsters: with all the possible combinations we can create "cognitive monsters" in P. These "monsters" are points of P which do not have any real exitence in O. So they can pertub the decision maker's judgment. Once more, we have to solve this problem algorithmically.

Coding of points and strategies: we assume that strategies are defined by a sequence of threshold objects so we can use the same representation space P for coding objects of O and strategies (ie. an antichain of P). To present the results we will use polynomial representation (see Barthelemy and Mullet [5]). A choice polynomial A is a formal expression: $A = M_1 + M_2 + ... + M_r$, where the M_i (called monoms) are formal expressions on attributes.

A monom has the form: $M_i = X_{i_1}^{a_1/c_{i_1}} ... X_{i_k}^{a_k/c_{i_k}}$, where X_{i_j} are attributes and the exponents a_k are values on the scales of each X_{i_j} (recalled by the c_{i_j}). If we code the P_i by the ordered list $(0 \leq ... \leq c_i)$, we will only use exponents greater than 0, because an a_k equal to 0 means that the worst value on X_{i_k} is accepted in the monom concerned. Products in monoms must be read as "and" (conjunction on attributes and values on these attributes) and "+" in polynoms read as "or".

The number of monoms is the number of rules that explain the decision maker's strategies. The length of the monoms will show us how many attributes are used for (or explain) the decisions. Examining the polynom A will also show compensation phenomena, the level of exigency of the decision maker ...

4 Learning algorithm

We have shown that we only need a general algorithm to search for an antichain in a direct product of chains. First, we consider below some situations for the learning algorithm to deal with.

4.1 Interactive vs. automatic acquisition of the decision maker's expertise

We will make the following separation: an **interactive extraction** which needs the decision maker's collaboration (cooperation and time...) and an **automatic extraction** of which the decision maker is unaware.

In the first case, we need an efficient algorithm for two reasons essentially. We want to minimize the number of objects proposed to the decision maker (length of questionnaire) because it is important to save the expert's time and cognitive efforts. So, we must be able to supervise the questionnaire and the objects successively presented to the decision maker must not be chosen with a random law.

We also require a short time between two questions; so we must find a way of coding P that allows us quickly to search for the objects we have to propose to the decision maker; do not forget that the only way we have of comparing the objects is to compare them attribute by attribute (so if there are a lot of objects to compare it can take a long time before deciding which object is "good" for proposing to decision maker for evaluation). We can already see it costs a lot to add an attribute to describe the objects (because this is equivalent to adding a constraint for comparison between objects). It is also important to have a user-friendly interface to communicate with the decision maker.

In the second case, the program just observes the decision maker during his/her work and monitors his/her actions (when an object is presented to the decision maker the algorithm loops until the object is selected or rejected). The environment is his/her normal environment and we do not need to develop a user-friendly interface and we do not care about time (if the decision maker has not changed his/her strategies during this time for different reasons; for instance the world has changed, he/she has increased his knowledge!). In any case, if we detect this phenomenon, we have to start a new learning phase for the program. This is close to machine learning and we just have to be sure that the different situations (in this case the objects will constitute the sample test) that appear to the decision maker will be great enough in quantity and quality to extract the strategies used.

4.2 The decision maker's contradictions

Now, we must discriminate between two different ways of learning, especially if we use more than two categories and if we are in the interactive situation: extraction with contradictions (if one contradiction is detected then ask the expert which answer he/she wants to change) or extraction with no contradiction (at each step we must only propose to the decision maker the categories allowed).

Below, we will consider a non-contradictory and interactive algorithm which can be seen in figure 2.

4.3 Greedy Algorithm

Begin Search for an Antichain
Data: P (algorithmic representation of O)
$E_0 = P$ (E_i represents uncategorized objects so $E_0 = P$)
$i = 0$ (number of proposed objects)
$C_1 = \mathbb{O}$ (at the beginning nothing is categorized)
$C_2 = \mathbb{O}$
begin loop
$i = i + 1$
  choose an uncategorized object $x_i \in E_{i-1}$
  read the expert answer (select or reject)
  if x_i is selected **then**
    $C_1 = C_1 + \{x \in E_{i-1} \mid x \geq x_i\}$ (monotonicity)
    $E_i = E_{i-1} - \{x \in E_{i-1} \mid x \geq x_i\}$ (elimination)
  endif
  if x_i is rejected **then**
    $C_2 = C_2 + \{x \in E_{i-1} \mid x \leq x_i\}$ (monotonicity)
    $E_i = E_{i-1} - \{x \in E_{i-1} \mid x \leq x_i\}$ (elimination)
  endif
loop until $E_i = \mathbb{O}$
return C_1 minimals (an antichain of P)
End Search an Antichain

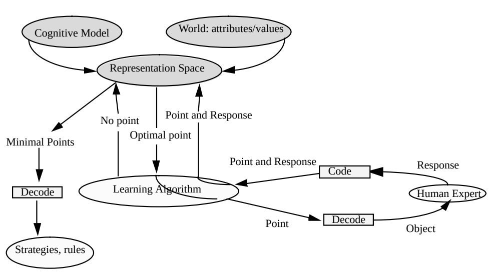

Figure 2: Interactive questionnaire

4.4 Complexity of the search

We must consider two things: the number of objects presented to the decision maker and the time between two objects proposed.

This algorithm must present the shortest questionnaire to the decision maker (minimize the number of proposed objects). We can show that when doing a dichotomic search on the chains of P which is partitioned into a minimum number of chains (see Pichon *et al.* [16] and figure 1 where the minimum number of chains is 4), the complexity of the algorithm is bounded by $\alpha \times \log_2 h$ (where α is the width of P, ie. the magnitude of a largest antichain of P and h his height, ie. the magnitude of the largest chain equal to the sum of the c_i). This is the case when we are looking for the largest antichain of P. This complexity, is fortunately, theoretical and we have much better empirical results. In fact, it considers that every chain of P has the same length (in most cases this is not true) and that the propagation is carried out in only one chain (the chain which the point/question belongs to) at each step (in fact it is effected on P). But we are not searching for the largest antichain of P; this will be incompatible with the parcimony principle because it will say that there are a lot of rules with all the attributes. Furthermore, in a direct product of chains, points of the same level (a level c is the set $\{x = (x_1, ..., x_p) \in P \mid \sum_i^p x_i = c\}$) form some antichains of P and if the decision maker's antichain is a level of P this means we can explain his/her choices with the utility model (see Von Neumann *et al.* [20]) where all the attributes have the same weight. In that way, he/she does not use strategies on the attributes and we could say that he/she is not a "good" decision maker; there is no experience on the attributes and they are all consider the same. Note that the width of P is the maximum number of rules we can find to explain the decision maker's choices (number of monoms in the polynom of choices).

The width of P can be evaluated by Anderson' approximation (Leclerc [11]):

$$\alpha \approx \sqrt{\frac{6}{\pi}} \frac{\prod c_i + 1}{\sqrt{\sum c_i(c_i+2)}}$$

The consequences of a high number of attributes: it is easy to verify that it costs a lot to add an attribute because of the numerator's product. This is naturally understandable; when we add an attribute we add a constraint of comparability on the objects so that the performance of the elimination procedure is reduced. It also increases the time between two propositions because there are many more points in P to be computed.

It is easy to see that the performances of the algorithm (for the number of objects proposed) depends on the choice of x_i because it conditions the elimination procedure. We have chosen to eliminate the maximum of objects at each step, so x_i is chosen by the following optimization rule:

$$\max_{x_i, x_i \in E_{i-1}} [\ \min_{x_i}\ (\mid B\ (x_i) \mid, \mid W\ (x_i) \mid\ )\ ],$$

where $B(x_i) = \{x \in E_{i-1} \mid x \geq x_i\}$ is the set of points better than x_i, and $W(x_i) = \{x \in E_{i-1} \mid x \leq x_i\}$ is the set of points worst than x_i.

This rule guaranties the minimum regret: we minimize the decision maker's "worst" answer. We do not have the complexity of this algorithm, but we have observed, experimentally, that we need 2α objects to find the largest antichain. This experimental complexity has been deduced from "reasonable" direct products of linear orders; from small ones to $P_1 \times P_2 \times P_3 \times P_4 \times P_5$, where each $P_i = \{0 \leq 1 \leq 2 \leq 3 \leq 4\}$.

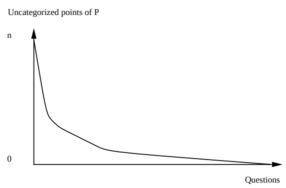

Figure 3: Progress of the algorithm

The progress of the algorithm is represented in figure 3. At the beginning it is possible to eliminate a lot of points at each step. But this decreases rapidly because the order structure of P is quickly broken (the subsets of comparable points become smaller and smaller). So we might wonder whether it is possible to stop the learning phase early and when This will considerably reduce the number of objects proposed but the quality of the learning rules too.

4.5 The way of coding P

Because we want a short time between two questions, we must have a way to decide quickly which uncategorized objects must be presented to the decision maker's judgment. The rule used to optimize the length of the questionnaire can take a long time especially in two cases: first if the number of objects is very great and second if we do not use an appropiate data structure which allows us easily to find the points we are searching for (for each uncategorized point we have to calculate its comparables) with no exhaustive search in P. There are different ways of coding P (see Guillet *et al.* [8], Lenca [12], Pichon *et al.* [16], and Wang [21]). Here, we expose another way of codifying the points of P which enable them to be localized directly.

First, to save memory, an object o described by p attributes (which are themselves integers in P_i) is coded by only one integer in P. In this coding system, each numeral of the integer represents the level on the corresponding attribute X_i. For instance, if the objects of the world are described by the three attributes, X_1, X_2, X_3 which take respectively their values in $V_1 = \{bad, middle, good\}$, $V_2 = \{small, middle, large\}$ and $V_3 = \{expensive, middle, cheap\}$, then the three linear orders P_i coding the attributes are $\{1 \leq 2 \leq 3\}$ and the object (bad, middle, cheap) is represented in P by 321. So if 2147483647 is the maximum integer we can use in machine, this allows us to represent objects with 9 attributes and at most 9 different values per attribute (coding starts at 1), which could be sufficient.

Then the representation space P is easy to generate with a development in base 10 starting at the minimum point 11..1: $x = (x_1, x_2, ..., x_p)$ is coded in P by the integer $x_p x_{p-1} ... x_1$:

$$x_p x_{p-1} ... x_1 = \sum_{i=1}^{p} x_i \times 10^{i-1}.$$

The points of P are stored in the natural order; for the previous example, $P[1] = 111$, $P[2] = 112$, $P[3] = 113$, ..., $P[26] = 233$, $P[27] = 333$.

We can directly generate the comparables of any point (with the same development starting at this point) and it is easy to verify that when we have a point we can immediately calculate its position in P and look whether it is categorized or not.

4.6 Advantages and difficulties of this methodology in learning the decision maker's strategies

We only deduce strategies from the observation of the decision maker's behavior through his/her decision results. There is no verbalization and introspective effort (language biases, interpreting, omission due to what is left unsaid, contradictions ...). The generation of the shortest questionnaire also economizes the decision maker's time and combined efforts.

The decision maker uses his/her own subjective scale. With this method we do not need to know it.

When we deal with real problems it is sometimes difficult to choose the attributes; they must be relevant and linearly ordered. It is also not easy to decide correlations between them, especially if there are a lot (and in this case we must consider the combinatorial problems). The cognitive monsters are, as we have said before, to be taken into account for two reasons: first, they perturb the decision maker and second they increase the number of objects proposed unduly. We must distinguish two categories of monsters: first, the very good and the very bad ones, which are in the "top" and "bottom" of P and secondly the middle ones. The first category does not need special work because they will not be considered by the algorithm (this is due to the way of seaching for the antichain). But, the second ones, are for the same reason dangerous because they can often respect the rule of choice of a question. They exist in most cases because of correlation between attributes and require (if we want!) special attention.

In light of the previous remarks, it is now easy to know what the "very good" problems are for this methodology.

5 Illustration in savings plans

The Moving Basis Heuristic is now use in different domains, for example, to know how students are searching for paid employement (see Barthélemy *et al.* [6]), in industry (see Guillet *et al.* [8]), in schools (see Wang [21]) and in banking (see Lenca [12]).

We illustrate below an application of the model in banking. It is useful to know and understand how people (who are considered as experimented-subjects when thinking about their money, "they know what they want even if they can not always say it") accept or refuse a savings plan. It is useful for the bank to be able to propose exactly what its customers want to have in this complex world of savings plans (see Lenca [12]); the bank consultant can use this methodoly for extracting the rules of choices of his/her clientèle (personally) and then advise exactly the corresponding savings plans. This will be a definite advantage (there is a strong competition between banks); with their money, customers do not like to be be very bably advised!

Subject

The subject is a twenty-years old male who works for the French administration. He knows the world of savings plans very well and has contracted 3 savings plans at his bank. He has agreed to be "used" several times for this experience and he will be followed during one year to see his evolution. He has to select or refuse the savings plan we propose to him.

Description of the saving product world O

The world O of savings plans is described by 4 attributes: fiscal system (noted F), expected profit (noted E), minimum guaranteed rate (noted T) and availability of investment (noted D). They could take respectively 4 ($c_1 = 3$), 3 ($c_2 = 2$), 3 ($c_3 = 2$) and 5 ($c_4 = 4$) different values, so there are $4 \times 3 \times 3 \times 5 = 180$ possible products represented in P. These four attributes have been chosen because they are nearly always present for all savings plans (in France) so they are significant and important in most cases for people who evaluate a savings plan. The attributes and their different values were presented to the subject. He was informed that he had to evaluate the savings plans using this description.

But, within the 180 points created there are "cognitive monsters". They can perturb the subject when they are presented to his judgement. But, it could also be of interest to see if they are recognised by the subject. So in this experience the "monsters" were not eliminated. Note that if they are eliminated, the number of objects proposed will diecrease (in this space P there are more than 25 "monsters").

We need only one piece of information about the subject: we have to know which policy level he supports in order to order the attribute F. The other attributes are assumed to be naturally well ordered.

Description of P

$P = F \times E \times T \times D$, where
F = {40% taxes$\leq$ 20% taxes $\leq$ declaration $\leq$ tax free},
E = {nearly good $\leq$ good $\leq$ very good },
T = {none $\leq$ 4,5% $\leq$ 6%},
D = {8 years $\leq$ 5 years $\leq$ 2 years $\leq$ 3 months $\leq$ free}.
An attribute X_i with v_i values is coded by $\{1 \leq 2 \leq ... \leq v_i\}$. This is due to the way of coding P in the program. So, with polynomial representation, we will use exponents only if they are greater than 1.

Complexity of the search

For this space P, $h = 11$ and $\alpha = 31$ so $\alpha \times \log_2 h = 107$ (59% of points of P). So we know that we will have at most 107 objects to propose. But, $2\alpha = 62$, which is much better and reassuring.

Detailled information given to the subject

The following details form part of the French legislation (for savings plans) so, some of them could be completely incomprehensible for the foreigner: "You will pay tax on F if F is not *"tax free"*, if you do not respect the attribute D or if you cross threshold for transfer", "T is strictly guaranteed whateter E is", "D can be the time you must respect (because of F) or a prescribed delay", "E represents experts's knowledge of savings plans, state of rates, sate of the market ...".

Questionnaire

Learning phase
At the first object proposed, the question he asked was: "There is no charge, no threshold for the scheme, do I need to pay money into my account regularly?". He was trying to see if these attributes were present or not. This confirms that he is an experimented subject for savings plans; he knows that these attributes are important or could constrain him a lot. He was informed that he must consider himself to be "free" and not to consider them anymore.

38 objects (21% of P, remember that the "monsters" were not eliminated; some of them were also

presented to him and he recognised them immediately) were sufficient to learn his strategies of choice. It takes about 10 minutes (9 minutes for the subject: he needs about 14 seconds to evaluate a savings plan and give his answer). This generates 8 rules.

Verification phase

Half a hour later, the 38 objects were presented to him a second time. He makes four mistakes, which is not a lot. But two of them were points which are in the antichain. These points are at limit. For the first one ($F^{2/4}E^{1/3}T^{2/3}D^{3/5}$), he makes no disccuss and we do not know why he makes this "mistake" because it is a good scheme for him. In fact, it seems that $E^{1/3}$ was not enough against $F^{2/4}D^{3/5}$ because in France it is possible to have savings plans like $F^{4/4}T^{2/3}D^{5/5}$ but with nothing more to expect; he does not want to wait and risk paying taxes for just a *nearly good* expected profit (attribute E). For the second one ($F^{4/4}E^{3/3}T^{1/3}D^{1/5}$), he said it was a good scheme which necessitates devoting a certain lenth of time to follow its evolution (there is no minimum guaranteed rate), and he accepted it during the learning phase. But he does not want to do that; he wants to be "free" and he refuses it later (maybe because he has to justify himself).

Then the height rules are proposed to the subject. He confirms all of them. In particuliar the two points which were contradictary just before (objects which are rules are necessary in the questionnaire): this confirms that he is a little undecided on these points.

Results and discussion

They are 8 rules and if we note A the antichain:

$$A = D^{4/5} + E^{2/3}T^{3/3} + E^{3/3}T^{2/3} + F^{2/4}T^{2/3}D^{3/5} + F^{2/4}T^{3/3} + F^{3/4}E^{2/3}T^{2/3}D^{2/5} + F^{4/4}T^{2/3} + F^{4/4}E^{3/3}$$

Remmenber (for instance) that $D^{4/5}$ represents $F^{1/4}E^{1/3}T^{1/3}D^{4/5}$ and must be read "if the attribute availability D is better or equal of *three months* then the savings plan is accepted".

This can be reduced to,

$$A = T^{2/3}E^{2/3} \times (T + E + D^2F^3) + F^{2/4}T^{2/3} \times (T + D^3) + D^{4/5} + F^{4/4}E^{3/3}$$

We can say a lot about these results but we will consider only some important points. It is fairly obvious that: this man does not want to take any risk with his money. $T^{2/3}$ is nearly always present except in $F^{4/4}E^{3/3}$ where he is very demanding for the other attributes (and do not forget it comes from a "mistake" made by the subject). Anyway, in this monom, we can see that he takes a minimum of risks because $F^{4/4}$ allows him to leave the savings plan when he wants; this rule is coming because the "cognitive monsters" have not been eliminated. We must notice that in fact $T^{2/3}$ is in $D^{4/5}$. The subject justifies himself: if a savings plan is made for three months it could not be a risky one because man cannot consider a risky product for a short time (this man is not a short term speculator).

He is ready to have long-term savings plans if there is a good final rate. We must consider T, E and F (which does not have to be considered if T is respected because in this case F will be *"tax free"*); they condition the final and net rate. We can see this in A: when he "loses" a level in T or E he wants to "win" two or three levels for D.

The two monoms $E^{2/3}T^{3/3}$ and $E^{3/3}T^{2/3}$ show the compensatory principle between E and T. We can see this phenomenon in other monoms too; the examination of the antichain A shows conjonction/disjonction on the attributes and the threshold values which explain his choices.

The monoms are short which means that few attributes are necessary to explain his choices when they have good values (parsimony principle and threshold hypothesis). There is an exception, $F^{3/4}E^{2/3}T^{2/3}D^{2/5}$, which is an average product so all the attributes have to be taken into account.

Conclusion and future developments

The Moving Basis heuristic model has been now experimented in different situations. We have found efficient algorithms to allow us to use this method in real problems. The results are encouraging but there is still a lot of work to do: reduce time between objects proposed to the decision maker if we want to describe the world O with a lot of attributes (more than five) and also the number of objects we propose; develop a user-friendly interface that allows a novice (novice with knowledge extraction problem) to use this methodoloy. We could also be interested in searching for something else other antichains; this will be necessary if the decision-maker does not use ordered categories...

To reduce the length of the questionnaire

One of the most important things to do to apply this methodology in an interactive way is to minimize the number of objects proposed. For this purpose we can look for different ways. The most obvious one seems to eliminate the "cognitive monsters" (if any exist). If we have special knowledge of the domain of application and where the antichain is located (for example we are almost sure it is near the middle of P) then we must use this knowledge for searching for the antichain (the optimisation rule is better if we know nothing about the antichain). We can also wonderabout the opportunity of using a probabilistic model to know when we can stop the learning algorithm (while being able to calculate the quality of the results).

Future utilisations in banking

We can use this method to know exactly what people want to have concerning savings plans but also to learn how credit is accorded to customers; in this case, the decision maker is the bank consultant and extraction of his/her knowledge will be useful for constructing the expert-system in credit... The knowledge extracted could be useful for novice consultants.

References

[1] T. Adelbratt and H. Montgomery. Attractiveness of decision rules. *Acta Psychologica*, (45):177–185, 1980.

[2] K.M. Aschenbrenner and Kasubek W. Challenging the custing syndrome : Multi-attribute evaluation of cortison drugs. *Organizational Behaviour and Human Performances*, (22):216–234, 1978.

[3] J.P. Barthélemy and E. Mullet. Choice basis : A model for multi-attribute preference. *British journal of Mathematical and Statistical Psychology*, 39:106–124, 1986.

[4] J.P. Barthélemy and E. Mullet. A polynomial model for expert categorical judgment. *Progress in mathematical psychology*, 1:419–437, 1987.

[5] J.P. Barthélemy and E. Mullet. A model of selection by aspects. *Acta Psychologica*, 79:1–19, 1992.

[6] J.P Barthémy and Mullet E. Expert individual decision: the moving basis heuristics. Technical Report RR-94005-IASC, Télécom Bretagne, BP 832, 29285 Brest Cedex, France, April 1994.

[7] R. de Hoog and G. van der Wittenboer. Decision justification, information structure and the choice of decision rules. In Berndt Brehmer, Helmut Jungermann, Peter Lourens, and Guje Sevón, editors, *New directions in research on decision making*. North-Holland, 1986.

[8] F. Guillet and G. Coppin. Industrial process control : experimental design and expertise acquisition. In *First Conference on 'Cognitive Sciences and Industry'*, Centre de Recherche Public - Centre Universitaire du Luxembourg, June 1994.

[9] O. Huber. Decision making as a problem solving process. In Berndt Brehmer, Helmut Jungermann, Peter Lourens, and Guje Sevón, editors, *New directions in research on decision making*. North-Holland, 1986.

[10] E. J. Johnson and J. W. Payne. Effort and accuracy in choice. *Managment science*, 31:295–314, 1985.

[11] B. Leclerc. Sur le nombre d'éléments des niveaux des produits de chaînes et des treillis permutoèdres. *Mathematiques Informatique et Sciences Humaines*, 28e annèe(112):37–48, 1990.

[12] P. Lenca. Extraction et analyse de processus de jugement et de décision - application au domaine de l'épargne. Technical Report RR-94008-IASC, Télécom Bretagne, BP 832 - 29285 Brest Cedex, April 1994.

[13] T. M Mitchell. Generalization as search. *Artificial Intelligence*, (18):203–206, 1982.

[14] H. Montgomery. Decision rules and the search for a dominance structure : toward a process model of decision-making. In P. C. Humphrey, O. Svenson, and A. VARI, editors, *Analyzing and Aiding Decision Process*, pages 343–369. Amsterdam North-Holland, 1983.

[15] H. Montgomery and O. Svenson. On decision rules and information processing strategies for choices among multiattribute alternatives. *Scand. J. Psychol*, 17:283–291, 1976.

[16] E. Pichon, P. Lenca, F. Guillet, and J. W. Wang. Un algorithme de partition d'un produit direct d'ordres totaux en un nombre minimum de chaînes. *Mathématiques Informatique et Sciences Humaines*, 32e année(125):5–15, 1994.

[17] R. Ranyard and R. Crozier. Reasons given for risky judgment and choice : a comparison of three tasks. In P. C. Humphrey, O. Svenson, and A. Vari, editors, *Analyzing and Aiding Decision Process*. Amsterdam North-Holland, 1983.

[18] H.A. Simon. *Model of Thought*. Yale University Press, 1979.

[19] O. Svenson. Process description of decision making. *Organizational behavior and human performance*, 23:86–112, 1979.

[20] J Von Neumann and O Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[21] J.W. Wang. *Système cognitif d'extraction automatique de règles de décision*. PhD thesis, Telecom Paris - Département Informatique, 46, rue Barrault - 75634 PARIS Cedex 13 - France, 1994.