New Structures for Conditional Probability Tables.

Conference Paper · January 2002

Source: DBLP

Juan A. Fernández del Pozo
Universidad Politécnica de Madrid

Concha Bielza
Universidad Politécnica de Madrid

27 Congreso Nacional de Estadística e Investigación Operativa
Lleida, 8–11 de abril de 2003

New Structures for Conditional Probability Tables

J.A. Fernández del Pozo[1], and C. Bielza[2],

Decision Analysis Group, Dept. of Artificial Intelligence,
Technical University of Madrid, Boadilla del Monte, 28660 Madrid, Spain
[1]E-mail: jafernandez@fi.upm.es
[2]E-mail: mcbielza@fi.upm.es

ABSTRACT

The aim of this paper is a new approach to structure and implement conditional probability tables. These tables represent the uncertainty on probabilistic graphical models, like Bayesian Networks and Influence Diagrams. We rewrite the tables as multidimensional matrices and these matrices as lists of generalised propositions from the original tables. This procedure allows us to obtain a framework to store and manage any kind of table with discrete explanatory attributes and a response variable. It is proposed to exploit the granularity on data and mine knowledge in high dimensional tables. Also, we deal with other related issues in probabilistic graphical models like probabilistic model assignment (elicitation and validation) and graphical representation of high dimensional data sets.

Keywords conditional probability tables, data structures, information storage and retrieval, reasoning under uncertainty, knowledge representation models
AMS Classification: (68P05), (68P20), (68T37), (68T30).

1 Introduction

Bayesian Networks (BN) and Influence Diagrams (ID) are useful tools to represent and evaluate decision-making problems under uncertainty and make inferences. A BN is a Directed Acyclic Graph (DAG) whose nodes are the concepts of the problem and its directed edges represent conditional probabilistic influence, quantified by means of conditional probability tables (CPT) among nodes. Probability propagation allows us to make inferences about some nodes given some evidence [Pearl, J., (1988)]. An ID is a DAG, with CPT's, like BN, and Utility Tables as main data structures. Its evaluation allows us to achieve optimal decision policies for every scenario represented on the model [Shachter, R.D., (1986)].

Dealing with these models on Artificial Intelligence and Decision Analysis implies management of very large tables of decision, probability and utility. In this paper we focus our attention on CPT's. Both formalisations need elicit [Jensen, F.V., (2001)] and/or learn [Buntine, W., (1996)] CPT's for knowledge representation, inference processing and decision-making. Also, their storage and management in an efficient way are required. Note the CPT (exponential) complexity according to the attribute set size in any context or parent set.
This complexity may be reduced efficiently by exploiting independence among variables, as BN's and ID's representations and inference algorithms do. However, when a CPT shows certain regularities not captured within BN or ID structures, the CPT complexity may be reduced further more. These regularities are independencies held only in certain contexts, the so-called *context-specific independence* [Boutilier, C. et al., (1996)]. Most literature has dealt with it by proposing extensions to the usual structures that capture these independencies and support efficient algorithms. See, e.g. [Heckerman, D., (1990)]; [Boutilier, C. et al., (1996)]; [Cano, A. et al., (2000)] within the BN framework, and [Smith, J. E. et al., (1993)]; [Shenoy, P.P., (2000)] within the ID framework. In general, most suggestions are based on tree-structured CPT's.
In [Fdez del Pozo, J.A. et al., (2001)] we introduced the *KBM2L* framework to store and manage any kind of table with discrete explanatory attributes and a response variable. The table is structured as a special list, see Section 2. Now, in this work, that framework accommodates CPT's showing different viewpoints depending on whether probability parameters play the role of a response variable or an explanatory attribute, see Section 3 and some illustrative examples in Section 4.
Our aim is to help decision analysts and experts to develop complex PGM's. Compact representations of a distribution while simultaneously discovering relevant variables within a context, allow to support probability elicitation, and to validate distributions, both during and after the model construction, as part of the sensitivity analysis and diagnosis tasks.

2 Overview of *KBM2L*

In [Fdez del Pozo, J.A. et al., (2001)], we introduced *KBM2L* structures to store and manage general tables as multidimensional matrices (MM). It stands for a **K**nowledge **B**ase **MM** transformed into a **L**ist. Every multidimensional table has an attribute set, that we call *schema* and a defined order for the elements of the schema, that we call *base*. The table content is called *response* or data. Then, these are arranged on a list like a computer allocates the data on its memory. Equal consecutive entries are joined later forming each element of the list called *item*. The *KBM2L* framework makes a list of items that represents the knowledge from multidimensional data tables. The data sequence, in memory, depends on the base. Some of these bases are optimal in the sense that the coalescence of cases is maximum

and the list has minimum length.
Given a fixed base with a fixed domain order, the tables may be considered as MM's. The content of the table stored in the cell with coordinates $\overrightarrow{c} = (c_0, c_1, ..., c_n)$ will be assigned to the position $MM[c_0, c_1, ..., c_n]$. The MM values will be stored successively in a computer, where only the memory address of the first one is known.
Thus, the values may be ordered by means of the application $f : R^{n+1} \rightarrow R$

$$f(c_0, c_1, ..., c_n) = \sum_{j=0}^{n}(c_j * \prod_{i=j+1}^{n} D_i) = q \quad (1)$$

which provides *offset* q of a value with respect to the first element of the table. For the i-th attribute ($i = 0, 1, ..., n$), D_i is the cardinal of its domain and $\prod_{i=j+1}^{n} D_j$ is called its *weight* w_i. $\overrightarrow{w}$ is the vector of weights $(w_0, w_1, ...w_n)$ with $w_n = 1$. The weights are the coefficients that multiply the coordinates in (1). Therefore, we can use this relationship to access the values.
While in [Fdez del Pozo, J.A. et al., (2001)], we focused our attention on the tables of optimal decisions, i.e. the ID evaluation outputs, in this paper our tables of interest will be CPT's. For every CPT, we have a set of involved attributes and a sequence of cases induced by the base.
This framework of CPT's fits very well the *KBM2L* structures, where the content tries to represent all combinatorial configurations and the coordinates or indices represent the conditional context or scenario. High dimensional and huge CPT's show regularities and sparse parameters values (perhaps a lot of zeros).
The basic element of the list is the item. An item is made up of adjacent cells for which the table stores the same information. The items represent *grains* of knowledge or sets of cases with the same data. If the content of the table presents some level of granularity, we can store only one value for each group of cases. If we use another order for the attributes, we have the same knowledge but we change the granularity and, hence, the memory requirements to store the final list of items. An important objective is to get a base that minimises the number of items, bringing up the grains of knowledge.
Note that the indices and the offsets play the same role in the information arrangement process. The notation $\langle$*offsets or indices, data*$|$ we will use later, reflects two ideas. Firstly, the offsets in (1) or indices of the items are strictly increasing and, secondly, it summarises a set of adjacent cells with the same data and different to the data of the next item. We define a constant that represents unknown data, *unKB*. Through the examples, we will use label -1 instead of *unKB*. Index components are denoted by $0, 1, \ldots$ as computer memory coordinates. Data are denoted by $0, 1, \ldots$ as enumerated domain values but the order usually is not relevant. The initial list has always only one item representing the absence of knowledge: $\langle$*MaxOffset,unKB*$|$ or $\langle$*MaxIndex,unKB*$|$. It is incrementally completed with new cases by employing a set of item management rules introduced in [Fdez del Pozo, J.A. et al., (2002)].

We extend this notation to CPT's to emphasize which is the conditional attribute from the original table and which are the probability parameters. For the conditional attribute we use brackets (x), and for probability parameters as explanatory variables, square brackets $[p]$.

Spectrum charts: allow us to show the unidimensional memory layout and the coalescence of cases, see Figure 1 below. The spectrum is not a histogram; it is an image of the relative position and size of the information according to the *KBM2L* list. It can represent cases (or items) depending on the relative size of items on any base. It is a visualisation tool for (high) multidimensional data. Every color block represents one item. The base change transposes the blocks and joins all adjacent blocks with the same color, the response. We can see graphically the sensitivity of the data faced with the context and explore patterns on data. We denote the base as a CPT attribute permutation, using natural numbers {0,1,2,...} as the attribute names. The base defines the attribute order that determines the case allocation in memory (case sequence in physical memory) and the coalescence of cases.

3 CPT's using *KBM2L*

In this paper, a CPT is a set of propositional clauses that measure the uncertainty of a random variable X in a context $\bar{Y} = \bar{y}$ of a decision model. The usual notation is $P(X = x|\bar{Y} = \bar{y}) = p$, where X is a discrete variable, $\bar{Y}$ is a vector of discrete variables and $p \in [0.0, 1.0]$ is a real number that represents the probability of $X = x$ given $\bar{Y} = \bar{y}$, one context or state of the model.
Note that the CPT content, i.e. the probability values, is a real data type and, as we said above, *KBM2L* is useful for discrete attributes and response, with a few domain values. Therefore, in the following subsections, we propose two possible ways of representing this kind of tables using a *KBM2L* storage of: probability p and a logical value $(\{true, false\})$ regarding the conditional probability declaration. When implementing the later, we consider a discretisation of the probability real value to avoid infinite weights w_i and to make easier the item coalescence. The probability discretisation is fixed by the expert as a resolution parameter useful for the elicitation process, and it is refined later via sensitivity analysis.
Despite our representations manage with any attribute cardinality, it is convenient to use only binary attributes. That is carried out as in other frameworks (e.g logistic regression) by increasing the number of variables, but also by introducing consistency constraints on the *KBM2L* to control which are the allowed new configurations that reflect the old ones.
For example, suppose an attribute A with domain $\{0, 1, 2\}$. We can break it down into binary attributes in different ways. One way is: A_0 with domain $\{0, 1 \vee 2\}$ and A_1 with domain $\{1, 0 \vee 2\}$. But we need to introduce some consistency constraints: if e.g. $A_0 = 0$ and $A_1 = 1$, then the list must store *impossible* (syntatic constraint),

labelled as *coKB*. Another possibility is to define A_0 with domain $\{0, \bar{0}\}$, A_1 with domain $\{1, \bar{1}\}$, and A_2 with domain $\{2, \bar{2}\}$. To break down attributes amounts to increasing the representation dimension, see Table 1.

Table 1: Correct binary codifications of A

A_0	A_1	A	A_0	A_1	A_2	A
0	1	*coKB*	0	1	2	*coKB*
0	$0 \vee 2$	0	0	1	$\bar{2}$	*coKB*
$1 \vee 2$	1	1	0	$\bar{1}$	2	*coKB*
$1 \vee 2$	$0 \vee 2$	2	0	$\bar{1}$	$\bar{2}$	0
			$\bar{0}$	1	2	*coKB*
			$\bar{0}$	1	$\bar{2}$	1
			$\bar{0}$	$\bar{1}$	2	2
			$\bar{0}$	$\bar{1}$	$\bar{2}$	*coKB*

Some attribute binary representation like A_0 with domain $\{0 \vee 1, \overline{0 \vee 1}\}$ and A_1 with domain $\{2, \bar{2}\}$ is incomplete, and some like A_0 with domain $\{0 \vee 1, 1 \vee 2\}$ and A_1 with domain $\{0 \vee 2, 1 \vee 2\}$ is indefinite. These schemas do not have the original semantics. It is not possible to retrieve the original schema as a part of the new schema, see Table 2.

Table 2: Wrong binary codifications of A

A_0	A_1	A	A_0	A_1	A
$0 \vee 1$	2	*coKB*	$0 \vee 1$	$0 \vee 2$	0
$0 \vee 1$	$\bar{2}$	$0 \vee 1$	$0 \vee 1$	$1 \vee 2$	1
$\overline{0 \vee 1}$	2	2	$1 \vee 2$	$0 \vee 2$	2
$\overline{0 \vee 1}$	$\bar{2}$	*coKB*	$1 \vee 2$	$1 \vee 2$	$1 \vee 2$

With *coKB*, we try not only representing syntactic constraints but also impossible domain configurations, see Section 4. Note that a base change on a binary artificial schema is equivalent to a permutation of the original attribute domain values. Thus, we can achieve more coalescence on the list at the expense of more complexity on the schema.

3.1 Probability p as Response

The most natural way of managing a CPT is to build an index with both the conditioning attributes $\bar{Y}$ and the conditional attribute X, and to build a (continuous) response with p, the probability parameter. We define a formal item as $\langle (x)\bar{y}, p|$, see Section 2. With binary domains $\{0,1\}$ the empty list is $\langle(1)\bar{1}, unKB|$. $(0)\bar{0}$ is *MinIndex* and $(1)\bar{1}$ is *MaxIndex*. Label *unKB* means no knowledge about p in the item list.

All index dimensions have finite domains. But it is difficult to coalesce the continuous response p on the *KBM2L*. One possibility is to discretise the p values. Thus, if X is the absence or presence of a pathology, a p-discretisation as *null / low / medium / high / maximum* probability of suffering that pathology would allow to compact the CPT and derive diagnoses. This is typically suggested by a human expert according to the coarseness level desired. In this case, we trade accuracy to allow more compact representation and easier knowledge elicitation. Another possibility is to discretise the p values in a dynamic way, i.e. different discretisations for certain values of $\bar{y}$. [Cano, A. et al., (2000)] propose the average of similar propositions. Yet, we fix a maximum *KBM2L* size and try to smooth the spectrum on the optimal base.

3.2 Logical Value as Response

This other *KBM2L* definition has a discrete response. The response is coded as three possible values $\{-1,0,1\}$. -1 has the usual meaning of *unKB*, 0 is *false* and 1 is *true*. This option has the following item formalisation: $\langle[p](x)\bar{y}$,{*false*,*true*}|. The empty list is $\langle[1.0](1)\bar{1}, unKB|$.

Now we need a discretisation of the probability parameters to compute the offset of the items. For the definition of the previous and posterior index of a given index we need a certain level of discretisation K: 10, 100, 1000,... values for p to denote the "next" value of a continuous parameter p. I.e., $[\lfloor p*K\rfloor-1](1)\bar{1}$ is considered as the previous index of $[\lfloor p*K\rfloor](0)\bar{0}$. If we fix the probability resolution for discretisation on 100, the empty list will be $\langle[100](1)\bar{1}$,*unKB*|.

We claim that this is the most suitable *KBM2L* representation for CPT's when there is imprecision on the p values, see Section 4. On the other hand, the first definition in Section 3.1 is rather intuitive for precise p values although it may yield low coalescence of cases unless a coarse discretisation is made.

3.3 Item Interpretation

Let *fixed* be the subset of attributes whose values are constant for all the cases of an item, [Fdez del Pozo, J.A. et al., (2001)]. Suppose the whole *KBM2L* with any *unKB* items.

Probability p as Response Let $\langle\bar{y}_1(x)\bar{y}_2, p|$ be a generic item. Suppose X is binary, If $x \in$ *fixed* then exist two items $\langle\bar{y}_1(0)\bar{y}_2, p|$, $\langle\bar{y}_1(1)\bar{y}_2, 1-p|$ or $\langle\bar{y}_1(1)\bar{y}_2, p|$, $\langle\bar{y}_1(0)\bar{y}_2, 1-p|$. If $x \notin$ *fixed* then $p = 0.5$. The attribute subset not contained in *fixed* on one item represents all scenarios where the measure of uncertainty is p.

Here $KBM2L$ items are $\langle (x)\bar{y}, p|$ representing the quantitative dependency among attributes on an explicit way.
This representation avoids the common expression for two different situations: uniform distribution (p is equal $\forall x$ for some contexts $\bar{y}$) and lack of knowledge. Uniformity is dealt with $((0)\bar{y}, 0.5)$ and $((1)\bar{y}, 0.5)$, if x is a binary variable. On the other hand, the lack of knowledge is $((0)\bar{y}, unKB)$ and $((1)\bar{y}, unKB)$. In general, the complementary of $((0)\bar{y}, p)$ is $((1)\bar{y}, 1-p)$.

Logical Value as Response Let $\langle \bar{y}_1(x)\bar{y}_2[p]\bar{y}_3, \mathit{true}|$ or $\langle \bar{y}_1[p]\bar{y}_2(x)\bar{y}_3, \mathit{true}|$ be generic items. If $p \in \mathit{fixed}$ then this item has similar interpretation as above. If $p \notin \mathit{fixed}$ the item represents a probability interval. When the response is *false*, it represents semantic constraints, while *coKB* represents syntactic constraints, see Tables 1 and 2.
This item represents the logical probabilistic knowledge: $\langle [p](x)y, \{\mathit{false},\ \mathit{true}\}| = \langle [p](x)y, \{0, 1\}|$. The response is domain-independent, in the sense that it is always a logical value $\in \{\mathit{false},\ \mathit{true}\}$.
Other situations are represented as follows. Complementaries are $((0)\bar{y}[p], 1)$ and $((1)\bar{y}[1.0-p], 1)$, uniformity is $((0)\bar{y}[0.5], 1)$ and $((1)\bar{y}[0.5], 1)$ and ignorance or *unKB* is $((x)\bar{y}[p], unKB)$.

3.4 Applications

Support for CPT Elicitation: the list and its spectrum are useful and complementary methods to support probability assignment of precise parameters and decide the probability parameter imprecision, like probability intervals. Experts can declare general rules and constraints regarding the CPT and analyse the list on different bases.
Next, we show some examples of how different expert statements can be encoded. We specify the base with $\{x, y\}$ as a subscript. For instance, the expert can say *when* Y_j *is 0,* X *is* 1 *with probability* 0.95 which is written as $((1), \ldots, 0_j, \ldots, 0.95)$ (a generic precise case). On the base $\{X, Y_j, \ldots\}$, it induces: $\ldots, \langle (0)1_j, 1, 1, \ldots, 1, *|$, $\langle (1)0_j, 1, 1, \ldots, 1.95|$,... Also, *$X$ may be 0 or 1 but 2 is not possible if Y_j is 1* is $((2), \ldots, 1_j, \ldots, coKB)$ (a constrained case). And also, the expert can say *I think the probability of this type of situations, $x\bar{y}$, involves a probability close to 0.25, but I do not know exactly* (an imprecise case). Following the first alternative, Section 3.1, it amounts to having several cases, e.g., $((x)\bar{y}, 0.24)_{\{x,y\}}$, $((x)\bar{y}, 0.25)_{\{x,y\}}$, $((x)\bar{y}, 0.26)_{\{x,y\}}$, and the complementary cases. But this defines a multivalued response $(0.24 \vee 0.25 \vee 026)$ or an interval for imprecise probabilities $([0.24, 0.26])$. We think that this representation is not suitable because it produces an irregular discretisation (non-homogeneous width intervals) and a wrong response domain (non-exclusive response values). On the other hand, following the second alternative, Section 3.2, we build an item like $\ldots$, $\langle (x)\bar{y}[0.23], \mathit{false}|_{\{x,y,p\}}$, $\langle (x)\bar{y}[0.26], \mathit{true}|_{\{x,y,p\}}$, $\langle (x)\bar{y}[0.27], \mathit{false}|_{\{x,y,p\}}$, ... Then, we would insert it into the corresponding list, that

in general is expressed in a different base and we would search for the optimal list, with minimum storage space.
We encode the rules and constraints, represent the spectrum and browse the items, its bounds, fixed (relevant) parts,... [Fdez del Pozo, J.A. et al., (2001)]. The list helps to gather all expert knowledge in a consistent way and gives us clues about which information is needed for building the CTP. The optimal list allows us to synthesize the information to a small set of relevant and short queries for the expert to improve the CPT model. The analyst can formulate questions to the expert about regions with lack of knowledge of the *KBM2L* representation space and check for inconsistencies.

Support for CPT Validation: the list summarises a large parameter set. We propose this structure for validation by the expert or by means of a data sample over the list. Two kinds of strategies are proposed: first, we try to study large items and the attribute role (fixed or variable part) [Fdez del Pozo, J.A. et al., (2001)], and second, we try to analyse tiny items with similar indices but different response. Large items involve a few fixed attributes and therefore, the questions formulated to the expert may be simple. Tiny items involve many fixed attributes and therefore, by comparing two of them with only slight differences in their fixed part, the questions formulated would be easily derived from these differences.
The validation process is similar to the elicitation process but we suppose that the CPT is given and we try to extract the general rules for the experts from the optimal list. They accept or not the results. The Elicitation and Validation are reciprocal procedures in the construction of a knowledge-based system.

Support for CPT Knowledge Mining: on large CPT's, *KBM2L* is useful to discover general patterns, the main components of the probabilistic relationships. We can try to explore several rules or descriptions about the problem and show the agreement with the CPT that was built using data or model evaluation. The ID evaluation produces very large CPT's, posterior distributions, that can be analysed with this tool for explanations and sensitivity analysis. On high dimensions we could see the performance of this technique although here it will not be shown.
Finally, we think that *KBM2L*'s are not useful for inference and probability propagation, but we have not studied this issue in depth yet.

4 Examples

We show here *KBM2L* examples of the two proposed CPT representations, see Table 3 for a summary.
Suppose the following simple CPT for X given Y: $P(x_0|y_0) = a$, $P(x_1|y_0) = 1.0 - a$, $P(x_0|y_1) = b$, $P(x_1|y_1) = 1.0 - b$, with variables or attributes: $X : \{x_0, x_1\}$; Y :

Table 3: Summary of CPT representations using *KBM2L* structures with binary domains and a 100-discretisation

response	$\rightarrow p$
item list	empty list
$\langle (x)\bar{y}, p|$	$\langle (1)\bar{1}, unKB|$
response	$\rightarrow \{false, true\}$
item list	empty list
$\langle [p](x)\bar{y}, \{0,1\}|$	$\langle [100](1)\bar{1}, unKB|$

$\{y_0, y_1\}$. We use the ordinal domain values $\{0, 1\}$ instead of $\{x_0, x_1\}$ and $\{y_0, y_1\}$ ($x_0 = 0$, $x_1 = 1$, $y_0 = 0$ and $y_1 = 1$).
For exposition clarity, it holds: $0 < a < b < 0.5$, but any $a, b \in [0.0, 1.0]$ are possible.

4.1 Discretisation and Imprecision

The discretisation of probabilities with 100 values implies codifications for $a, b, 1.0 - a, 1.0 - b$ as follows: $A = \lfloor a * 100 \rfloor$, $B = \lfloor b * 100 \rfloor$, $\bar{B} = \lfloor (1.0 - b) * 100 \rfloor$, $\bar{A} = \lfloor (1.0 - a) * 100 \rfloor$.
We take into account two possibilities:

1. The parameters are precise, $a = 0.15$ and $b = 0.45$. Then, $A = 15$, $B = 45$, $\bar{B} = 55$, $\bar{A} = 85$.

2. The parameters are unknown, but we know the value bounds and then $a \in [0.12, 0.16]$ and $b \in [0.40, 0.45]$. Thus, $A = \{12, ..., 16\}$ and $B = \{40, ..., 45\}$.

4.2 Probability p as Response

The discretised *KBM2L* is $(\langle (0)0, A| \; \langle (0)1, B| \; \langle (1)0, \bar{A}| \; \langle (1)1, \bar{B}|)_{\{x,y\}}$, i.e.

$$(\langle (0)0, 15| \; \langle (0)1, 45| \; \langle (1)0, 85| \; \langle (1)1, 55|)_{\{x,y\}}.$$

All item size are equal to 1 and the base is the very best.

4.3 Logical Value as Response

Remember the notation and let us denote the item size, number of cases, as an exponent: $\langle [probability](conditional\ attribute)context\ attributes, logical\ value|^{item\ size}$.
The simple *KBM2L* for X is:

$$(\langle [14](1)1, 0|^{60} \; \langle [15](0)0, 1|^{1} \; \langle [45](0)0, 0|^{120} \; \langle [45](0)1, 1|^{1} \; \langle [55](1)0, 0|^{41} \; \langle [55](1)1, 1|^{1} \; \langle [85](0)1, 0|^{118} \; \langle [85](1)0, 1|^{1} \; \langle [100](1)1, 0|^{61})_{\{p,x,y\}},$$

i.e. the precise case.
The complex and imprecise *KBM2L* is:

$$(\langle[11](1)1,0|^{48}\ \langle[12](0)0,1|^{1}\ \langle[12](1)1,0|^{3}\ \langle[13](0)0,1|^{1}$$
$$\langle[13](1)1,0|^{3}\ \langle[14](0)0,1|^{1}\ \langle[14](1)1,0|^{3}\ \langle[15](0)0,1|^{1}$$
$$\langle[15](1)1,0|^{3}\ \langle[16](0)0,1|^{1}\ \langle[40](0)0,0|^{96}\ \langle[40](0)1,1|^{1}$$
$$\langle[41](0)0,0|^{3}\ \langle[41](0)1,1|^{1}\ \langle[42](0)0,0|^{3}\ \langle[42](0)1,1|^{1}$$
$$\langle[43](0)0,0|^{3}\ \langle[43](0)1,1|^{1}\ \langle[44](0)0,0|^{3}\ \langle[44](0)1,1|^{1}$$
$$\langle[45](0)0,0|^{3}\ \langle[45](0)1,1|^{1}\ \langle[55](1)0,0|^{41}\ \langle[55](1)1,1|^{1}$$
$$\langle[56](1)0,0|^{3}\ \langle[56](1)1,1|^{1}\ \langle[57](1)0,0|^{3}\ \langle[57](1)1,1|^{1}$$
$$\langle[58](1)0,0|^{3}\ \langle[58](1)1,1|^{1}\ \langle[59](1)0,0|^{3}\ \langle[59](1)1,1|^{1}$$
$$\langle[60](1)0,0|^{3}\ \langle[60](1)1,1|^{1}\ \langle[84](0)1,0|^{94}\ \langle[84](1)0,1|^{1}$$
$$\langle[85](0)1,0|^{3}\ \langle[86](0)1,1|^{1}\ \langle[86](1)1,0|^{3}\ \langle[87](0)1,1|^{1}$$
$$\langle[87](1)1,0|^{3}\ \langle[88](1)0,1|^{1}\ \langle[100](1)1,0|^{49})_{\{p,x,y\}}.$$

The base change $\{p,x,y\} \to \{y,x,p\}$ for this list produces an improvement of $43 \to 15$ items:

$$(\langle 0(0)[11],0|^{12}\ \langle 0(0)[16],1|^{5}\ \langle 0(0)[85],0|^{69}\ \langle 0(0)[88],1|^{3}$$
$$\langle 0(1)[83],0|^{96}\ \langle 0(1)[88],1|^{5}\ \langle 1(0)[39],0|^{52}\ \langle 1(0)[45],1|^{6}$$
$$\langle 1(0)[85],0|^{40}\ \langle 1(0)[88],1|^{3}\ \langle 1(1)[54],0|^{67}\ \langle 1(1)[60],1|^{6}$$
$$\langle 1(1)[84],0|^{24}\ \langle 1(1)[85],1|^{1}\ \langle 1(1)[100],0|^{15})_{\{y,x,p\}}.$$

Figure 1 shows the spectrum chart of both lists.

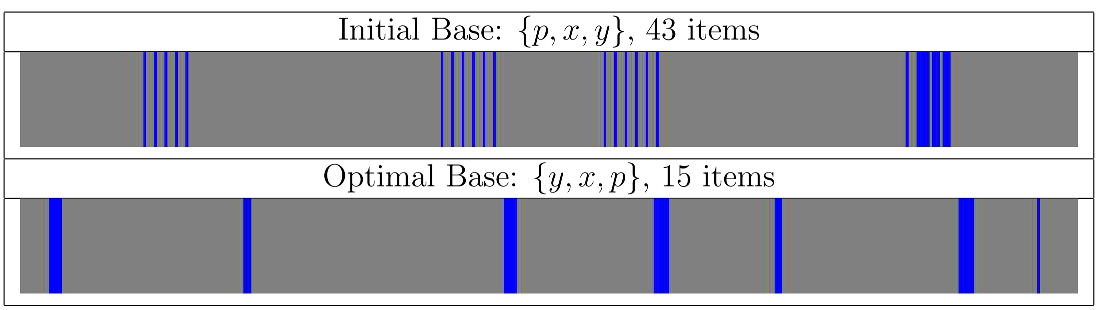

Figure 1: *KBM2L* spectrums

Because $|A| = 5$ and $|B| = 6$, this structure represents $5 * 6 = 30$ CPT's like that with $a = 0.14, b = 0.40$, or that with $a = 0.13, b = 0.44$. We store implicitly $0.83 < P(x_1|y_0) \leq 0.88$ in $\langle 0(1)[88],1|$ and $P(x_0|y_0) \geq 0.11$ in $\langle 0(0)[11],0|$. We can increase the resolution easily because p has a weight equal to 1; the *KBM2L* length does not change.

A Real Example. Finally, we show a high dimensional example adapted from [Bielza, C. et al., (2001)] for this last representation. In this model we need to elicit

many big CPT's about the neonatal jaundice, a common clinical problem. Our example is a CPT for hemoglobin concentration (X=CHgb: low(0)/normal(1)/high(2)) for a patient, conditioned to several (8) pathologies (Pth absent(0)/present(1)) and the critical patient age (Age No(0)/Yes(1)). Therefore, let $\bar{Y}$ be a context vector with 9 binary components. Let us take a 100-discretisation resolution for p. The empty list is $\langle[100](1)\bar{1}, unKB|$ with $155,136$ cases. The CPT has $3 * 2^9 = 1,536$ parameters, but only 1024 parameters are free. In [Bielza, C. et al., (2001)] the elicitation problem was partly solved by means of a *noisy OR-gate* [Pearl, J., (1988)] and all probability parameters were precise. But often it is difficult the noisy OR-gate hypotheses to be held (independence between causes, some variables are not strictly causes, the domains are not binary,...) and we try to elicit the general CPT: $P(CHgb|Age, Pth_1, \ldots, Pth_8)$.
Suppose the information is null (empty list) or some prior knowledge like:

$$P(0|000000000) = 0.0,\ P(1|000000000) = 1.0,$$
$$P(0|111111111) = 1.0,\ P(1|111111111) = 0.0.$$

The initial list is shown in Table 4, and one fragment of the optimal list in Table 5.

Table 4: *KBM2L* example, high dimensional CPT, initial list

Item	p	X	$\bar{Y}$	$\{0,1\}$
0	$\langle[0]$	(0)	000000000,	$1\|^{1}$
1	$\langle[0]$	(1)	111111110,	$-1\|^{1,022}$
2	$\langle[0]$	(2)	000000000,	$1\|^{2}$
3	$\langle[0]$	(2)	111111110,	$-1\|^{510}$
4	$\langle[0]$	(2)	111111111,	$1\|^{1}$
5	$\langle[100]$	(0)	111111110,	$-1\|^{152,575}$
6	$\langle[100]$	(1)	000000000,	$1\|^{2}$
7	$\langle[100]$	(2)	111111111,	$-1\|^{1,023}$

With the initial list shown above, the analyst asks the expert (or makes queries to the data base) the proposition set with probability equal to $1.0, 0.90, \ldots, 0.0$. The expert declares which are the contexts with $p = 100, 90, \ldots, 10, 0$. This process can be repeated several times over the whole list or over the *unKB* items, restricted contexts. The main idea is to fill the list with knowledge. Also, the expert may declare domain rules, constraints or modulating attributes. For example, the following rules:

1. *when three or more pathologies are present, $CHgb$ is pathologically low with probability 1.0*,
2. *no more than four pathologies are present on one patient, and $Pth1$ and $Pth8$ never are jointly present,...*

3. *when one pathology is present, Age increments the probability of low about 10%*,

The *KBM2L* must be consistent [Fdez del Pozo, J.A. et al., (2002)], while the information (parameters, rules or constraints) is introduced into the structure, see Figure 2. Here the spectrums do not show the whole list because there are very large unknown items. The building and optimisation process may be run in parallel. The final result is the optimised CPT, see Figure 2. Such system helps the analyst to formulate questions to the expert. Both of them have a chart of the CPT that shows the relationship among the attributes.

5 Conclusion

Central issues concerning this paper are:

Knowledge Representation. In short, we have tried to develop an extended representation of *KBM2L* focused on CPT's. This structure can represent the probabilistic relationships on PGM's and it is difficult to build it when it has a high dimension. Two equivalent approaches have been proposed –probability and logical values– playing the role of the content of the table. We have described the syntax and semantics, computational storage and operations of coalascence on big CPT's. Also, we have used the *KBM2L* to support imprecise elicitation of probabilities with intervals.

Applications. Some applications have been provided to support the Elicitation of Probabilities, Knowledge Validation and Knowledge Mining. The three applications are close from the point of view outlined in this paper.
We have proposed an Elicitation Protocol, see the final example for big CPT's. The proposal takes into account Domain Constraints and Expert Rules. It allows to mix Expert Knowledge and Data Bases. It makes Maintenance of Consistency (if $\bar{y}$ is not possible within $X|\bar{y}$, then $\langle [p_j](x_i)\bar{y}, coKB| \; \forall i, j)$. Actually, we are performing Sensitivity Analysis of attributes on optimal bases, according to the attribute weights (these reveal the attribute importance on the CPT).

Open Issues. Some suggestions for future work are:

1. Improvement of elicitation and validation tasks. Now we are working even on other $KBM2L$ representation, with the table indexed by the probability parameter and the context. Thus, the response is the conditional attribute.

2. Development of algorithms for probability inference on BN's and evaluation on ID's, that exploit the optimal storage.

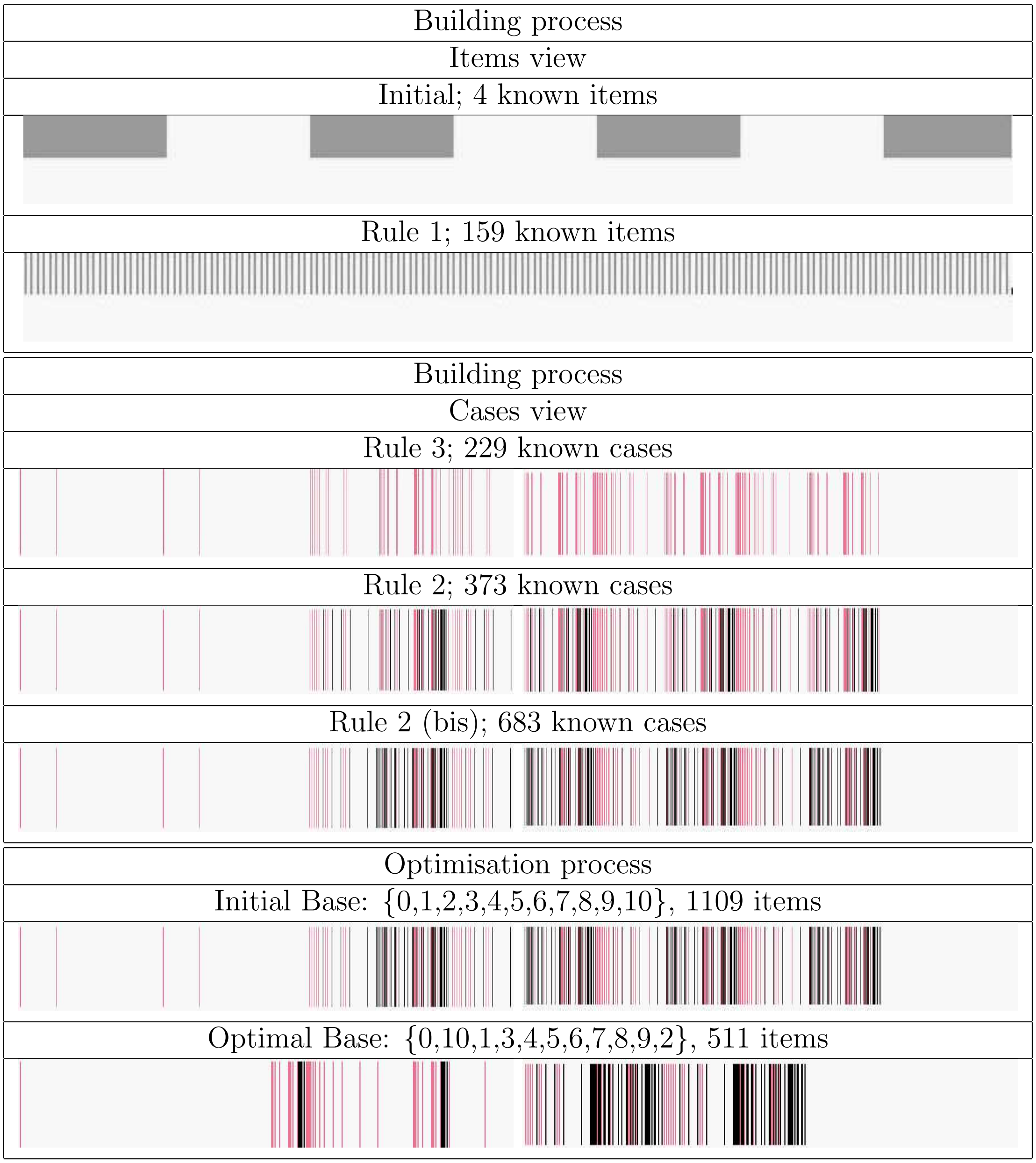

Figure 2: High dimensional CPT

3. Generalisation of the structure to multidimensional offsets and spectrum charts. The application $f : R^{n+1} \rightarrow R^k$ uses several indices and offsets. This multidimensional *KBM2L* implies a complex definition of items as a generalisation of (1), the previous definition of adjacent cases.

Acknowledgments. Research supported by Ministry of Science and Technology, Project DPI2001-3731.

Table 5: *KBM2L* example, high dimensional CPT, optimal list

Item	p	Y_{10}	X	$\{Y_3, Y_4, Y_5, Y_6, Y_7, Y_8, Y_9, Y_2\}$	$\{0, 1\}$
0	$\langle[0]$	0	(0)	00000001,	$1\|^{2}$
1	$\langle[0]$	0	(1)	11000001,	$-1\|^{448}$
2	$\langle[0]$	0	(1)	11000101,	$1\|^{4}$
3	$\langle[0]$	0	(1)	11000111,	$-1\|^{2}$
4	$\langle[0]$	0	(1)	11001001,	$1\|^{2}$
5	$\langle[0]$	0	(1)	11001111,	$-1\|^{6}$
6	$\langle[0]$	0	(1)	11010001,	$1\|^{2}$
7	$\langle[0]$	0	(1)	11011111,	$-1\|^{14}$
8	$\langle[0]$	0	(1)	11100101,	$1\|^{6}$
9	$\langle[0]$	0	(1)	11100111,	$-1\|^{2}$
10	$\langle[0]$	0	(1)	11101001,	$1\|^{2}$
11	$\langle[0]$	0	(1)	11101111,	$-1\|^{6}$
...	...	...	...	...	...
184	$\langle[0]$	1	(2)	11111010,	$0\|^{3}$
185	$\langle[0]$	1	(2)	11111101,	$-1\|^{3}$
186	$\langle[0]$	1	(2)	11111111,	$0\|^{2}$
187	$\langle[1]$	0	(1)	10000000,	$-1\|^{385}$
188	$\langle[1]$	0	(1)	10000001,	$1\|^{1}$
189	$\langle[1]$	0	(1)	11100000,	$-1\|^{63}$
190	$\langle[1]$	0	(1)	11100001,	$1\|^{1}$
191	$\langle[3]$	0	(1)	01000000,	$-1\|^{2943}$
192	$\langle[3]$	0	(1)	01000001,	$1\|^{1}$
193	$\langle[3]$	0	(1)	01100000,	$-1\|^{31}$
194	$\langle[3]$	0	(1)	01100001,	$1\|^{1}$
...	...	...	...	...	...
253	$\langle[30]$	1	(1)	00000001,	$-1\|^{763}$
254	$\langle[30]$	1	(1)	00000010,	$1\|^{1}$
255	$\langle[50]$	0	(0)	11110001,	$-1\|^{29935}$
256	$\langle[50]$	0	(0)	11110011,	$0\|^{2}$
257	$\langle[50]$	0	(0)	11110101,	$-1\|^{2}$
258	$\langle[50]$	0	(0)	11110111,	$0\|^{2}$
259	$\langle[50]$	0	(0)	11111011,	$-1\|^{4}$
...	...	...	...	...	...
506	$\langle[100]$	1	(0)	11110111,	$-1\|^{4}$
507	$\langle[100]$	1	(0)	11111001,	$0\|^{2}$
508	$\langle[100]$	1	(0)	11111101,	$-1\|^{4}$
509	$\langle[100]$	1	(0)	11111111,	$0\|^{2}$
510	$\langle[100]$	1	(2)	11111111,	$-1\|^{512}$

Bibliography

[Bielza, C. et al., (2001)] Bielza, C., Gómez, M., Ríos-Insua, S., Fernández del Pozo, J.A. (2000) Structural, Elicitation and Computational Issues Faced when Solving Complex Decision Making Problems with Influence Diagrams, *Computers & Operations Research* 27(7-8), 725-740.

[Boutilier, C. et al., (1996)] Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D., (1996) Context-Specific Independence in Bayesian Networks, *12th Annual Conference on Uncertainty in Artificial Intelligence*, Portland, Oregon, 115-123.

[Buntine, W., (1996)] Buntine, W., (1996) A guide to the literature on learning probabilistic networks from data, *IEEE Trans. on Knowledge and Data Engineering*, 8(2), 195-210

[Cano, A. et al., (2000)] Cano, A., Moral, S., Salmerón, A., (2000) Penniless propagation in join trees, *International Journal of Intelligent Systems*, 15(11), 1027-1059.

[Fdez del Pozo, J.A. et al., (2001)] Fdez del Pozo, J.A., Bielza, C., Gómez, M., (2001) Knowledge Organisation in a Neonatal Jaundice Decision Support System, *Medical Data Analysis*, Lecture Notes in Computer Sciences, Crespo, J. and Maojo, V. and Martín, F., Springer, Berlin, 2199, 88-94.

[Fdez del Pozo, J.A. et al., (2002)] Fdez del Pozo, J.A., Bielza, C., Gómez, M., (2003) Heuristics for Multivariate Knowledge Synthesis, *European Journal of Operational Research, to appear.*

[Heckerman, D., (1990)] Heckerman, D., (1990) Probabilistic Similarity Networks, *PhD Thesis*, Stanford University, CA.

[Jensen, F.V., (2001)] Jensen, F.V., (2001) Graphical Models as Languages for Computer Assisted Diagnosis and Decision Making, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Lecture Notes in Artificial Intelligence, Benferhat, S. and Besnard, P., Springer, Berlin, 2143, 1-15.

[Pearl, J., (1988)] Pearl, J., (1988) Probabilistic Reasoning in Intelligent Systems, *Morgan Kaufmann*, San Mateo, CA.

[Shachter, R.D., (1986)] Shachter, R.D., (1986) Evaluating Influence Diagrams, *Operations Research*, 34(6), 871-882.

[Shenoy, P.P., (2000)] Shenoy, P.P., (2000) Valuation Network Representation and Solution of Asymmetric Decision Problems, *European Journal of Operational Research*, 121(3), 579-608.

[Smith, J. E. et al., (1993)] Smith, J. E., Holtzman, S., Matheson, J. E., (1993) Structuring Conditional Relationships in Influence Diagrams, *Operations Research*, 41(2), 280-297.