Combining explicitness and classifying performance via MIDOVA lossless representation for qualitative datasets

Martine Cadot
Université de Nancy/LORIA,
Département Informatique
Nancy - France
Martine.Cadot@loria.fr

Alain Lelu
Université de Franche-Comté/LASELDI
LORIA
Nancy - France
Alain.Lelu@univ-fcomte.fr

***Abstract*—Basically, MIDOVA (Multidimensional Interaction Differential of Variability) lists the relevant combinations of K boolean variables in a datatable, giving rise to an appropriate expansion of the original set of variables, and well-fitted to a number of data mining tasks. MIDOVA takes into account the presence as well as the absence of items. The building of level-k itemsets starting from level-k-1 ones relies on the concept of *residue*, which entails the potential of an itemset to create higher-order non-trivial associations – unlike Apriori method, bound to count the sole presence of itemsets and exposed to the combinatorial explosion. We assess the value of our representation by presenting an application to three well-known classification tasks: the resulting success proves that our objective of extracting the relevant interactions hidden in the data, and only these ones, has been hit.**

Keywords-symbolic discrimination; variable interaction; machine learning; classification; non-linear discrimination; user comprehensibility; feature construction; feature selection; itemset extraction

I. INTRODUCTION

We introduce here a novel representation of an observation × variable datatable with binary modalities. This representation aims at enlightening the interactions between variables hidden in the data. It takes into account the negative as well as positive modalities of the variables, and eliminates redundancy, since it lists the only itemsets necessary and sufficient for reconstituting the data.

In order to assess the validity and usability of such a representation, we present an application of it to classification tasks on a few well-known test datasets. It must be clear that the main subject of this paper is *not* another classification method, which should be systematically assessed against the best existing ones on a broad variety of datasets: our success on a small number of classification tasks is nothing but a sufficient clue for arguing that the proposed representation is useful in many data mining applications, whether supervised (e.g., classification) or not (e.g., data-driven modeling).

We depart here from the two main research lines in data mining:

- the statistical line relies on correlations, covariances or contingency tables for assessing the two-by-two relations between variables, and generally ignores the interactions of rank three or more.

- the symbolic line (*knowledge discovery*) is bound to extract *frequent* itemsets or association rules in sparse datatables, which restrains it in practice to enlighten the sole interactions between positive modalities of the variables, and results in an excessive accumulation of logical conjunctions needing empirical frequency thresholds.

In contrast, our method 1) takes into account the 2^N facets of each order-N relation, including negative modalities when necessary, 2) self-limits the number and nature of the extracted relations to the ones necessary and sufficient for reconstituting the whole datatable, up to a permutation of the observations.

The assessment of a new data representation is far from being trivial: it is widely agreed that unsupervised data representations are difficult to assess. In our case an extra difficulty lies in the fact that few people use and interpret order-3 and higher interactions discovered in an unsupervised framework. Using a supervised one is thus a way for us to circumvent this limitation. This is why we have decided to feed a classic and well-known machine learning method (*Naïve Bayes*) with extra data issued from our MIDOVA representation of the datatable as is now exposed.

The illustrative objective of this paper is to present a proof-of-concept solution for a problem of ever-growing importance for software industry: in the framework of discrimination tasks, when facing massive and mostly qualitative data as it becomes common now, there is a growing need to explicit the reasons underlying a given discrimination, i.e., to uncover the variables or combination of variables intervening in the discrimination function.

The core principle of Support Vector Machines is that classification performance is increased by *increasing the dimensionality of the representation space* in which the discrimination task has to be performed, and not by reducing it. This paradox is explained by the concept of interaction: two (or more) variables may not have the same effect as each one separately. In particular the combination of two (or more) variables may impact the target variable, while none of these variables would do that individually. This principle inspired us for setting up an enlarged data-space in classification problems involving binary variables, for the time being: our aim is to detect *all* the order-k interactions, so as to select those involving the target variable. We will show below that the number of combinations to consider is tractable: 1) this number is intrinsically limited by the

number of described entities (or "cases", "individuals", "observations") in the data, 2) at the k-th level, it is far below the number of theoretical combinations – as k increases, it decreases abruptly after its initial growth phase. Our solution has been briefly exposed yet in DBKDA 2010 [1]. As subtle combinatorics considerations are to be developed, we will take more pages here, and thoroughly examine all the facets of our solution considering a single toy example. Let us turn now to our main topic.

For supervised classification tasks, the main criterion is the overall classification performance, i.e., best generalizing power. In this regard, it is widely admitted that kernel Support Vector Machines [2] have outperformed their competitors. But in many application areas, the explicitness of the discrimination function is also a major criterion: kernel SVMs are blackboxes not akin to "explain" their decisions. The "kernel trick" is a powerful by-pass for computation costs, but at the expense of turning the decision process blind. This is a big concern in domains with life-threatening issues, such as medical or military ones. It is impossible to let a machine take (or suggest) such decisions of vital importance without explaining why, i.e., without clarifying what variables or combination of variables are involved in each specific decision-making.

In this respect, many other classification methods, whether linear as Naïve Bayes [3] or Fisher discrimination [2], or non-linear, such as Rule-based Classifiers [4] or Learning Classifier Systems [5], are *explicit*: they display (or may display) the (possibly weighted) list of variables or combinations of them leading to their classification decision.

Our objective is to meet both criteria of explicitness *and* performance, restricted here to the case of Boolean variables (i.e., qualitative variables with two modalities, True and False, noted by a and $\bar{a}$ for the variable **a**). The solution we propose consists of expanding the original variables with Boolean conjunctions of these ones. This idea has yet been worked out [6] in the framework of the Apriori method for extracting frequent itemsets [7]. Let us recall that an "itemset" as defined in Apriori is an unordered list of variables, its "support" is the number of co-occurrences of these variables in the dataset, and if the support exceeds a given threshold, the itemset is said "frequent". In this paper, we will deepen the presentation of our MIDOVA method, first described in French in [8], then in English in DBKDA 2010 [1]. As it is an unsupervised information extraction method, we will assess its performance applying it to supervised machine learning tasks, and comparing its results to the best published ones. The excellent classification performance obtained is mainly due to the conjunction of its original features, among which: it takes into account negative modalities as well as positive ones, it replaces the straightforward support criterion of Apriori-like methods by our "residue" criterion, detailed below.

First we will present the MIDOVA representation of a datatable: its context and motivation, its general and core principles, and an overview of the algorithm and its pseudo-code expression. Then we will detail the whole MIDOVA process on a toy dataset. At last the experimental section will present the application of MIDOVA expansion to the most basic discrimination problem: 2 classes and qualitative data, taken out of the UCI repository. Conclusions will be drawn, as well as perspectives.

II. MIDOVA Representation

After some generalities on MIDOVA, the reader will be presented a small example illustrating the essential concepts of component of an itemset, degree of freedom, support, residue and gain. We will insist on the concept of variation interval and illustrate it by Venn diagrams. We will confirm that MIDOVA uncovers the two clusters we had placed in our dataset. At the end we will develop some quantitative assessments of MIDOVA, especially in comparison to Apriori.

A. Context of the MIDOVA representation

We consider the case of a relation R between two sets, a set **S** of n individuals (or instances) $\mathbf{s}_i$ ($1 \leq i \leq n$), and a set **V** of p binary variables (or variables) $\mathbf{v}_j$ ($1 \leq j \leq p$). For example the individuals are patients in a hospital, and the (dichotomous) variables are the symptoms they experience, or not. Or the individuals are clients of a supermarket, and the variables are the products they are akin to buy, or not.

A first way to represent these data is as a list of lists: the lists of symptoms experienced by each patient, or the lists of commodities bought by each client, i.e., a set of sales slips. The second representation is more appropriate for reasoning: a Boolean matrix with n rows and p columns, and value 1 at the crossing of row i and column j if the individual experiences the symptom, or else the value 0. It is well known that these two representations are equivalent, the first one being a generally compact form of the second (as it gets rid of the zero values). The datasets illustrating our method are formatted as Boolean matrices.

We will propose a third representation, which is a set of itemsets. This representation is rooted in the extension of the statistical concept of contingency table between two variables a and b, to the concept of "contingency hyper-table" between more than two variables. The classic contingency table includes values in the four cells describing how the total number n of individuals distributes respectively along the four crossings of a and b (i.e., the counts of individuals who simultaneously satisfy a *and* b, a *and* $\bar{b}$, $\bar{a}$ *and* b, $\bar{a}$ *and* $\bar{b}$). Note that the first count is the support of itemset ab. In the same way, the three-way contingency table (between three variables) will be a cube with eight cells, and the appending of one more variable doubles the number of cells where the individuals distribute. As the sum of these cells is n, their content is smaller and smaller, and empty cells are more and more common. We call *components* of our k-itemsets (itemsets of length k, i.e with k variables) the cells of the k-way contingency table. Note that our definition of an itemset is more general than the one generally accepted, in that it includes all the cells of the contingency table, and not the sole "True, True, ...True" one. Note also that we have started from the 2-way contingency table, which is the most common one, by increasing the dimension, but for the sake of generality we also define the 1-way contingency tables for the 1-itemsets, i.e., with only one

variable and the empty itemset of 2^V, which we will admit being satisfied for all the individuals.

B. Principles for building the MIDOVA representation

The core principle of MIDOVA is based on the properties of contingency tables, which can be extended to hyper-tables. They are expressed in terms of marginal totals and degrees of freedom.

1) Marginal totals and degrees of freedom

The sums of all the counts corresponding to a variable, e.g., a, in a k-way contingency table are the exact counts of the (k-1)-way contingency tables with all variables except a. For example, in Figure 2, Row 2, the first 2-way contingency table corresponds to variables a and b, with four counts (3 individuals who verify a and b, 4 who verify a but not b, 2 who verify b and not a, 6 who verify nor a nor b). In its right-hand column, marginal totals are the two counts of a (7 individuals who verify a, 8 who do not verify a), the same as those written in the 1-way table of a (Figure 2, Row 1). In the bottom row of this 2-way table, the marginal totals are the two counts of b (5, 10) also written in the 1-way table b. In other words, the marginal sums are forcing the counts in the cells of the table, restraining the "freedom" of their contents.

The concept of "degrees of freedom" embeds the number of cells whose content may be fixed independently from the others – within limits we will express below. In our case of Boolean variables, the degree of freedom of each hyper-table is equal to one because all counts can be written as an algebraic expression of x (where x is the count of some fixed cell),and of the marginal values (see Figure 1 and Figure 2, and details of calculations in Section C).

2) MIDOVA indicators

In the previous paragraph, we have seen that the values of the cells of a K-way contingency table (i.e., the components of a K-itemset), are all linked to the value x of a single cell (i.e., to a single component) and to marginal values (i.e., to components of its sub-itemsets) by means of simple algebraic expressions. At step k-1, the x value is unknown, but it is possible to obtain its *variation interval*, i.e., its different values. For example, for itemset ab in Figure 1, the variation interval of x is [2 ; 7], and for itemset abc in Figure 2, the variation interval of x is [1 ; 2] (for details of calculations, see Section C; for an interpretation of the variation interval see Section F 4).

Three useful properties of itemsets are ensuing (their proofs are in [8].)

- The variation intervals of the all components of a k-itemset are entirely determined by the components of its (k-1)-itemsets.
- The components of a k-itemset all have the same amplitude noted L, "liberty".
- L is a non-increasing function of k.

We define two parameters: 1) Mr, the "MIDOVA-residue", equal to $2^{k-1}e$, where e is the gap (i.e., absolute difference) between the support and the closest bound of its variation interval, 2) Mg, the "MIDOVA-gain", which is proportional to the difference between the support s and the center c of its variation interval, $Mg=2^{k-1}(s-c)$. Let us recall that the support is the content of the "True True...True" cell corresponding to the case of values of variables all equal to 1.

Mr is a non-negative integer, which cannot exceed the value n/2 where n is the total number of subjects When the residue Mr of a k-itemset is zero, it remains no more liberties (L=0) for the k'-itemsets (where k'>k) including the same variables. This frozen situation stops their computation. This k-itemset is interesting in that it corresponds to an exact relation between all its variables. Its sub-itemsets may be also interesting: they correspond to relations between fewer variables, thus more general, but these relations are not exact, and have a number of counter-examples which increases with Mr.

Mg is an integer that takes values between –n/2 and n/2. If the gain Mg of a k-itemset is zero, the relation between the k variables can be rigorously deduced from the lower-level relations between k-1 variables. In the opposite case, the greater the absolute value of Mg, the larger the unexpectedness in this relation, and the more interesting it is. If its value is positive, the appended variable increases the relation between the previous variables, and decreases it otherwise.

C. Illustration of the MIDOVA principles

At the left of Figure 1, we have represented the 0-way contingency table corresponding to the 0-itemset, with 0 variable, always true, and the 1-way contingency tables respectively corresponding to the 1-itemsets a (true for 7 subjects) and b (true for 5 subjects), with the total n in the marginal cell. The 2-way contingency table corresponding to the 2-itemset ab is displayed at the right-hand part of the Figure 1. The components of itemset a are in the marginal column and the component of itemset b in the marginal row, and among the four cells of the table, only one cell can be affected independently of the other cells. We have chosen the cell corresponding of the number of individuals who satisfy a but not b, and the unknown number x is written in this cell.

n
15

a	$\bar{a}$	Tot.
7	8	15

b	$\bar{b}$	Tot.
5	10	15

	b	$\bar{b}$	Tot.
a	7-x	x	7
$\bar{a}$	8-(10-x)	10-x	8
Tot.	5	10	15

Figure 1. Expression of the 4 components of itemset ab given a, b and the total n

The other counts of the table are relative to x and the four marginal sums, their algebraic expression can be easily derived. For example, as the sum of the two cells in the first line is 7 (in blue, in the marginal column of the 2-way table, and in the first column of the 1-way table of 1-itemset a, which indicates that 7 subjects verify a), the number of subjects who verify a and b is 7-x. In the same way, as Column 2 contains two cells, which total is 10 (in red, in the marginal row of the 2-way table, and in the second column of the 1-way table of 1-itemset b), and as the cell in the first

row contains x, the second cell contains 10-x. And the value of the last cell derives by difference between the total of Row 2 (8, bold and blue) and the content (10-x) of the other cell of Row 2.

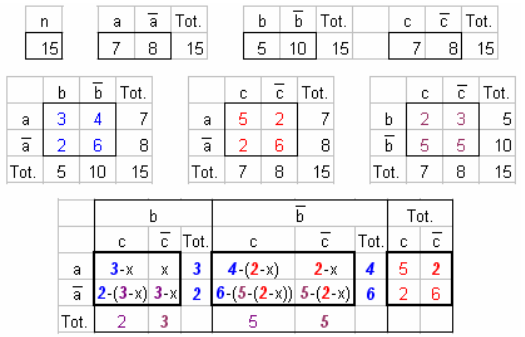

Figure 2. Expression of the 8 components of itemset abc given ab, ac, bc, a, b, c and n (a=v2, b=v4, c=v3 from Table 1)

.The value of x can vary between 2 and 7 because the four counts of the 2-way table (7-x, x, x-2 and 10-x) must be non negative. It follows that the liberty of the ab itemset is L=7-2=5. In Figure 2, we have the same items a and b, the ab itemset has been fixed (x=4), and a new item c has been added, with ac and bc itemsets, which are all known through their tables. The abc itemset is unknown, but the eight counts of the 3-way table in the third row depend on the value of x and on the marginal sums. It may be observed that the value of x is between 1 (as x-1, which is the count of individuals satisfying b and c but not a, cannot be negative) and 2 (as 2-x, which is the count of individuals satisfying a but neither b nor c, may not be negative), and so is L=2-1=1 for the abc itemset. For the sequence of itemsets a, ab, abc, the corresponding sequence of the L values of liberty is 15, 5, 1.

The computation of Mr and Mg for the abc itemset is developed in Section F.4: in the case of x=1, the three variables a, b, c are equivalent to the variables v2, v4, v3 of the Table I

D. Representation of a MIDOVA sequence

When one knows the count of a unique cell per each contingency table (there are 2^p such tables), it is enough for reconstructing the whole relation R. To this special role we will assign by convention the cell where all variables are set to 1, corresponding to the support of the itemset. For example in Figure 2 the relation between a, b, c is wholly defined, if x=1, by: Ø(15), a(7), b(5), c(7), ab(3), ac(5), bc(2), abc(2).

Generally (and fortunately), not all itemsets appear in the MIDOVA representation. As soon as an itemset is frozen (Mr=0), no following itemset is set up (i.e if abc is frozen, no abcd, abce, abcde, …, itemset will ensue). When the grand total n distributes among the 2^k cells of a k-way contingency table and $n<2^k$, then one cell at least is empty, and no more-than-k-dimensional table will be created. The maximum order of a component of R will be $\log_2(n)+1$.

E. MIDOVA algorithm

The MIDOVA algorithm is a levelwise algorithm, derived from the Apriori one [7]. The major difference between our algorithm and the Apriori one is our criterion of positive Mr allowing to enlarge a k-itemset into a k+1-itemset, instead of a support exceeding a threshold. In our algorithm, we have added (through the function "conditions") the possibility of saving the only itemsets with a support, and/or a gain and/or a residue greater than given thresholds. The description of MIDOVA algorithm comprises 3 parts: the variables, the functions and the main procedure.

1) Variables

- *V* : list of the variable headings in lexicographic order
- *M* : boolean matrix of the relation R
- *L0* : empty list
- *e* : element of a list
- *it1* : k-association, i.e., list of k variable headings in lexicographic order with their 2^k components (number of individuals in each cell)
- *it2* : the same as it1, with lenqth k+1
- *it0* : empty association
- *k* : lenqth of the associations at the current step
- *I* : global list of the Midova-oriented representation, accompanied for each association with its heading and other measures computes from its components
- *L1* : list of associations built at step k
- *L2* : sub-list of associations saved for the k+1 step
- *Mr* : Midova residue
- *Possible_follow_up* is a boolean variable, true if it is possible to extract in L2 an association built upon association it1

2) Functions

conditions(it) : boolean function ; true if association it checks up the conditions specified by the user, for example thresholds of support, gain and/or residue, computed starting from the components of the association.

append(I, it) adds the it association to I, keeping just the heading and the sole part of information issued from the components wanted by the application designer, e.g., support, gain, residue.

parc_L2(it, k, j) boolean function ; true if the heading of the j^{th} association in the L2 list next to it has the same (k-1) first elements as the it heading.

create_it(M, i) builds an association starting from its heading and its list of components computed from M.

succ(L2, it1, j) identifies the successor of it1 located j slots further in the L2 list.

extract(it2) extracts the (k-1)-subassociations in the heading of it2 except the two fisrt ; e.g : extract("abdf") yields the list ["adf", "bdf"]

rech_co_L2(M, it1, k, j) tests the grouping of the k- association it1 with the k-association it3 positioned j slots further in the L2 list. It yields the it2 heading by appending the last variable of it3 heading to the list of variables of the it1 heading. It checks then whether any sub-association included in it2 exists in L2. If not, the computed association is void. Else it creates the it2 association appending to its heading the values of its 2^k components computed from M.

Pseudo-code for rech_co_L2 function

rech_co_L2(M, it1, k, j)

```
it3=succ(L2, it1, j)
it2=concat(intit(it2),intit(it3)[k])
succeed=True
list_it←extract(it2)
it3=succ(L2, it3,1)
for it in liste_it do
      while not(it3=it0) and (it3<it) do
            it3=succ(L2, it3,1)
      end
      if not(it=it4) then
            succeed=False
            return it0
      end
end
it2←create_it(M, it2)
return it2
```

3) Main procedure

Pseudo-code for generating the MIDOVA representation

```
#Initialisation :
K ← 1 ; I ← L0 ; L1 ← L0 ; L2 ← L0
for e in V do
      it ← create_it(M, e)
      L1 ← L1 + it
end
for e in L1 do
      If conditions(e) then append(I, e) end
      If Mr(e)>0 then L2 ← L2 + e end
end

#further steps
while not ( L2 = L0) do
      k ← k+1, L1 ← L0
      for it1 in L2 do
        j←1
        Possible_follow_up ← parc_L2(it1, k, j)
        while Possible_follow_up do
              it2 ←rech_co_L2(M, it1, k, j)
              if  not(it2=it0) then L1 ← L1 + it2 End
              j←j+1
              Possible_follow_up ← parc_L2(it1, k, j)
        end
      end
      L2 ← L0
      for e in L1 do
        If conditions(e) then append (I, e) end
        If Mr(e)>0 then L2 ← L2 + e end
      end
end
```

F. Running MIDOVA on a toy dataset

We illustrate here the whole MIDOVA operation line on a small example of dataset, extending from the raw matrix representation, to the interpretation of the final MIDOVA results. For the sake of clarity, we take a simpler example than the partial one presented in the above subsection. Table 1 displays our artificial dataset, showing the Boolean values of 15 subjects s1, s2, …, s15 for 10 variables v1, v2,…, v10. One may read, for example, that for the subject s2, 5 variables upon 10, i.e., v1, v2, v3, v4 and v9, are true; or that the variable v6 is true for 4 subjects upon 15, i.e., s10, s12, s14 and s15.

TABLE I. BOOLEAN TABLE OF 15 SUBJECTS (E.G.,.PATIENTS)AND 10 VARIABLES (E.G., SYMPTOMS)

	v1	**v2**	**v3**	**v4**	**v5**	**v6**	**v7**	**v8**	**v9**	**v10**
s1	1	1	1	1	1	0	0	0	0	0
s2	1	1	1	1	0	0	0	0	1	0
s3	1	1	1	0	1	0	0	0	0	0
s4	1	1	0	1	1	0	0	0	0	0
s5	1	1	0	0	1	0	1	0	0	0
s6	1	1	1	0	0	0	0	0	0	0
s7	1	0	1	0	1	0	0	0	0	0
s8	1	1	1	0	0	0	0	1	0	0
s9	1	0	0	1	1	0	0	0	0	0
s10	1	0	0	0	0	1	1	0	0	0
s11	1	0	1	0	0	0	0	1	1	0
s12	1	0	0	0	0	1	1	0	0	0
s13	1	0	0	1	0	0	1	1	1	0
s14	1	0	0	0	0	1	1	0	1	0
s15	1	0	0	0	0	1	1	1	1	0

1) Scrutinizing a few 2-itemsets

The link between two variables is akin to be more or less pronounced, spanning from a complete opposition to a complete similarity through a complete unrelatedness. We exemplify below these three situations of respectively: 1) opposition, or contradiction, 2) linkage, or attraction 3) independence, or lack of connectedness.

The itemset A={v4 ; v6}

- includes two variables, thus its *length* k is 2.
- As the variable v4 is true for the five subjects s1, s2, s4, s9 and s13, it appears that no subject satisfying v6 belongs to this list, thus the *support* of A is zero.
- Its support could have been 1, 2, 3 or 4, but not further, depending on the possible number or common subjects. The *variation interval* of the support of A is therefore [0 ; 4], with a lower bound b_{inf}=0 and an upper bound b_{sup}=4.
- As seen above, its *residue* is a function of the gap (absolute value of the difference) between its actual support and the closest bound; in this case, $Mr=2^{k-1}(s-b_{inf})$ and its value is zero since $s= b_{inf} =0$.
- Its *gain* is a function of the difference between its actual support and the center of the variation interval (c=2), $Mg=2^{k-1}(s-c)$, i.e., -4.

This itemset is interesting in that it sheds light on an opposition between v4 and v6: the subjects who satisfy v4 do not satisfy v6, and vice-versa. This is precisely the type of knowledge we try to mine out of the data. But its super-itemsets {v1, v4, v6}, {v2, v4, v6}, …, {v10, v4, v6} are uninteresting: no need to return to Table 1 to conclude that their supports are zero, this conclusion proceeds from the zero support of A. We call A a *frozen itemset* for it does not generate super-itemsets carrying an extra knowledge, out of its own contribution. The zero value of the residue points at this frozen situation.

The itemset B={v6, v7} is true for 4 subjects, those who satisfy v6; v7 is true for these 4 subjects and two other ones, s5 and s13. Its support is 4 and its variation interval is [0; 4] as above. As the support equals the lower bound, the residue

Mr of B appears to be zero and the gain Mg is 4. B bears interesting information, i.e., all the subjects for which v6 is true satisfy also v7. This can be also expressed by the *association rule* v6→v7 whose *confidence* is 100%. Its super-itemsets are uninteresting: no need to return to Table 1 for delineating the subjects who satisfy the 3-itemset {v6; v7; v9}, they are exactly those satisfying {v6; v9}. The itemset B is frozen, as pointed out by its zero residue.

The itemset C={v6, v9} is true for 2 subjects, s14 and s15, sharing both variables, while v6 and v9 are respectively true for 4 and 5 subjects. The variation interval of this itemset is again [0; 4], but the support s=2 of C is now central in the interval, which corresponds to a residue of value 4 and a gain of 0. This zero gain shows that the relation between v6 and v9 is uninteresting. Conversely, as Mr is greater than zero, this itemset is not frozen, and one has to consider its derived and possibly interesting super-itemsets.

2) MIDOVA selection of the k-itemsets

The MIDOVA algorithm works level-wise, which means that once established, the level-k itemsets are combined for deriving the level-(k+1) ones. A level-(k+1) itemset needs k+1 level-k itemsets. In this way the 3-itemset {v2, v3, v4} derives from the three 2-itemsets {v2, v3}, {v2, v4} and {v3, v4}.

- Level 1: the 1-itemsets, made of a single variable, are generated ; there are ten of them.
- Level 2: the 1-itemsets are combined by twos for creating the 2-itemsets. For generating the sole knowledge-carrying 2-itemsets, the 1-itemsets with non-zero residues are selected, which eliminates the two variables v1, true for all the subjects, and v10, true for none of them (their residues are zero, and their respective gains are 7.5 and -7.5). In this way 28 2-itemsets are to be processed instead of the 45 ones when keeping v1 and v10. Considering these 28 itemsets, 8 are frozen (Mr=0), among which 7 have zero support and gains between -6 and -4, and one has support and gain values of 4. They won't contribute to build the upper levels, but as their gain values are important, they are left apart for the final interpretation step. The 20 remaining 2-itemsets are kept for building the next levels. Four of them have a zero gain, the others spread from -4 to 3.
- Level 3: The 20 2-itemsets with Mr≠0 are combined by threes for building the 3-itemsets, yielding 22 of them, among which only three have a non-zero residue; therefore, this sole number prevents building any 4-itemsets. Their gain values are zero. Among the remaining 19 with a zero residue, two only have a non-zero gain, i.e., {v2, v3, v4} with a support of 2 and a gain value of 2, and {v2, v3, v5} with a support of 2 and a gain value of -2.

In this way we have built the wholeness of the k-itemsets akin to provide pieces of information about the dataset. Throughout three steps, 10, 28 and 22 itemsets respectively have been considered, summing up to 60 (see Annex). Among these ones, those taken into account for building the ones at the next k+1 level amounts to respectively 8, 20 and 3; those with non-zero gain have been 10, 24 and 2, establishing in this way a total of 26 interesting relations between variables (out of the trivial 1-itemsets).

3) Differences between MIDOVA algorithm and Apriori-like ones

In Table II, the wholeness of the 1023 k-itemsets (k>0) potentially built starting from the data are split up and counted in reference to the zero value, or not, of their residues and gains. They yield from the MIDOVA process parameterized without any residue threshold instead of the threshold value 1 assigned above. There are then as many k-itemsets as combinations of variables, i.e., $1024=2^{10}$. The first itemset is the void one, which includes no variable, and is true for 15 subjects – it has been taken out from the itemset list as a trivial and uninteresting one. The last one is the « full » itemset which includes all the variables, but is true for no subject. The 26 interesting itemsets are emphasized in boldface, so as to point out the efficiency of our algorithm compared to a brute force one who would examine the 1023 itemsets. It is noticeable that, in the scope of a fair comparison between the Apriori algorithm and ours, Apriori should be considered as such: its efficiency follows from the sole principle of applying a threshold to the supports, an option we have discarded in order to retain the itemsets with zero supports, as these ones mostly points to interesting opposition relations. Moreover, 7 itemsets out of the selected 26 ones have a zero support.

TABLE II. HOW THE 1023 ITEMSETS EXHAUSTIVELY ISSUED FROM TABLE 1 DISTRIBUTE AS REGARDS TO THE VALUE ZERO OR NOT OF THEIR MR AND MG INDICES.

		Number of k-itemsets										
	k	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Mr=0	Mg=0		17	115	210	252	210	120	45	10	1	980
	Mg≠0	2	**8**	**2**								12
	Total	2	25	117	210	252	210	120	45	10	1	992
Mr>0	Mg=0		4	3								7
	Mg≠0	8	**16**									24
	Total	8	20	3								31
Total		10	45	120	210	252	210	120	45	10	1	1023

Going on with the subject line of enlightening the differences between our principles and Apriori's, we have shown in Figures 3 an 4 how do interesting itemsets rise or not in both methods, starting from the exhaustive 1023 itemsets:

- The 623 firsts, lexically ranked for each increasing value of k (1<k<6: 45 2-itemsets, 120 3-itemsets, 210 4-itemsets, 252 5-itemsets).
- The remaining ones (k>=6) have zero-valued supports, residues and gains.

In both figures, the x axis indicates the rank of the itemset according to the above-mentioned ordering, and the dotted line visually recalls the number k.

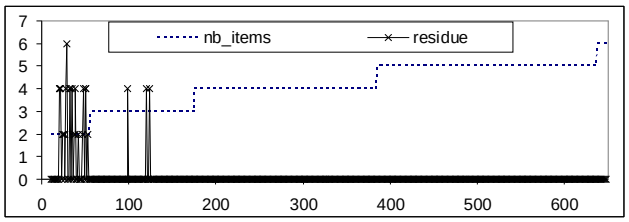

Figure 3. Residues of all k-itemsets of Table I.

In Figure 3, the y axis specifically stands for the MIDOVA residues, in order to enlighten how soon our algorithm locates the « no-future » value of k, thus how soon the algorithm is stopped.

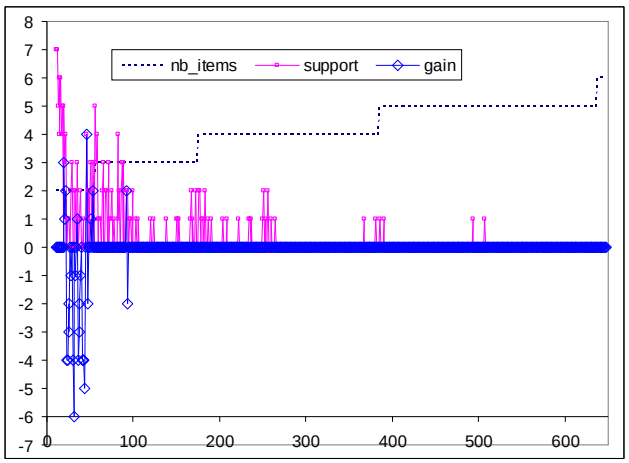

Figure 4. Interesting k-itemsets of Table 1, mined with MIDOVA or Apriori (with threshold 0)

In Figure 4, the y axis stands for the support and gains of the successive itemsets. It visually jumps out that 1) gain and support measure totally different phenomena, 2) MIDOVA does detect the strong gain values that clearly stands out as grouped for the small values of k. These two features explain both the good performance of MIDOVA, able to detect the wholeness of the interesting itemsets, and only these ones, including those with zero support, and its relative efficiency, compared to Apriori parameterized with a zero support threshold.

More precisely, one may read in Table III that among the 627-528=99 k-itemsets extracted by MIDOVA or Apriori with zero threshold, only 19 of them (17 2-itemsets and 2 3-itemsets) are simultaneously selected by the two methods – and the number of common items would still decrease using Apriori with a threshold. This confirms that the 26 inter-variable relations mined out from the Table I are different by nature from those issued from the Apriori family algorithms.

TABLE III. HOW THE 627 K-ITEMSETS (1<K<6) ISSUED FROM TABLE 1 DISTRIBUTE, ACCORDING TO ZERO OR NON-ZERO VALUES OF SUPPORTS AND GAINS

supp>s	**|gain|>g0**	**2**	**3**	**4**	**5**	**Total**
support=0	Mg=0	9	82	189	248	528
	Mg≠0	7				7
support≠0	Mg=0	12	36	21	4	73
	Mg≠0	**17**	**2**			19
	Total	45	120	210	252	627

As we have already interpreted the meaning of three examples of 2-itemsets, we now interpret the only two interesting relations between 3 variables.

4) Interpreting the 3-itemsets with non-zero gain

Let us examine first the itemset D={v2, v3, v4}. In Figure 5 a Venn diagram shows how the 15 subjects distribute among the three variables v2, v3 and v4. In this ensemblist layout, the subjects are splitted in 8 parts according to their values of the three considered variables. For example s7 and s11 are in the segment of v3 exterior to v2 and v4, as their values for v3 are 1, whereas they are zero for v2 and v4. In the same way the s1 and s2 subjects lie in the central part common to the three variables, as their values are 1 for all of them. Thus the support of the itemset D is 2.

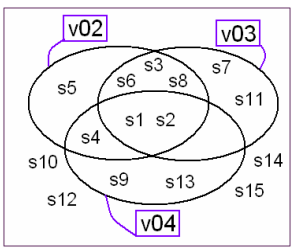

Figure 5. Venn diagram of the {v2, v3, v4} itemset.

The variation interval of the support can be found by trying to modify the support of D without modifying the supports of its component 2-itemsets. This can be done by moving the subjects from one area to another one: the only possible configuration is shown in Figure 6.

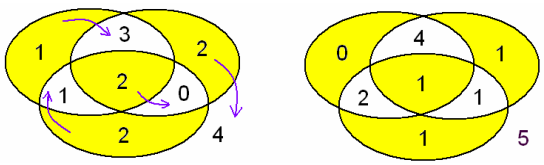

Figure 6. Transfer of 4 subjects of the {v2, v3, v4} itemset: the movements are drawn in the left-hand figure, resulting in the right-hand configuration.

In the left part of Figure 6, blue arrows indicate the authorized movements of the subjects (starting from a yellow area, where the number of negative variables is even, to a white one). As far as each support of the 2- and 1-itemsets

includes as much yellow areas as white ones (one each for the 2-itemsets and two for the 1-itemsets), their values are kept unchanged. For example the sub-itemset {v2, v3}, which included at the left a white area with 3 subjects and a yellow one with 2, now includes at the right a white area of 4 and a yellow one of 1, while its support stays the same (5=3+2=4+1) – obeying a kind of conservation principle, so to speak. The sub-itemset {v2}, which included two white areas with 1 and 3 subjects, and two yellow ones with 1 and 2 subjects, now includes two white ones with 2 and 4, and two yellow ones with with 0 and 1, whereas its support keeps constant (7=(1+3)+(1+2)=(2+4)+(0+1)). Implementing these changes of the variation interval can be done by just modifying the four values in the datatable. This can be done in different ways, one of which is displayed in Figure 7. In the left-hand part of the table the values of Figure 1 are reproduced, and the values to be changed for decreasing the support of D from 2 to 1 are highlighted in yellow; at the right-hand part of the table, these values have been modified.

Figure 6 shows the only possibility for the variation of the D itemset subjected to the stability conditions of the supports of its sub-itemsets. It follows that the variation interval of its support is [1; 2], of center 1.5 and gain $Mg=2^{k-1}(s-c)= 2^2(2-1.5)=2$, which points out a tight relation between v2, v3 and v4, relatively to the three underlying 2-relations, i.e., (v2 and v3, v2 and v4, v3 and v4).

	v02	v03	v04		v02	v03	v04
s01	1	1	1		0	1	1
s02	1	1	1		1	1	1
s03	1	1	0		1	1	0
s04	1	0	1		1	0	1
s05	1	0	0		1	1	0
s06	1	1	0		1	1	0
s07	0	1	0		0	0	0
s08	1	1	0		1	1	0
s09	0	0	1		1	0	1
s10	0	0	0		0	0	0
s11	0	1	0		0	1	0
s12	0	0	0		0	0	0
s13	0	0	1		0	0	1
s14	0	0	0		0	0	0
s15	0	0	0		0	0	0

Figure 7. An example of modification of 4 subjects of the {v2, v3, v4} itemset for implementing the changes in Figure 6.

The second 3-itemset with non-zero gain is E={v2, v3, v5}, which Venn diagram is shown in Figure 8.

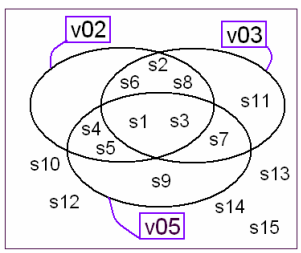

Figure 8. Venn diagram of the {v2, v3, v5} itemset.

In the same way Figure 9 shows the only way to move subjects in order to modify the support of E without modifying the supports of its component 2-itemsets, which yields a variation interval [2; 3] for the support, and thus a gain of -2 expressing a loosening of the link between the 3 variables v2, v3 and v5.

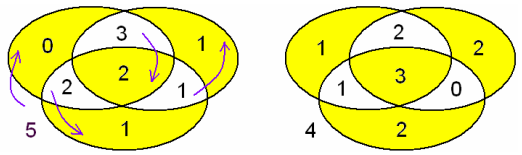

Figure 9. Transfer of 4 subjects of the {v2, v3, v5} itemset: the movements are drawn in the left-hand figure, resulting in the right-hand configuration.

5) Global interpretation

The reader may probably have observed that our example data had a special structure: apart from v1, always true, and v2, never true, the variables v2 to v5 mainly characterize the subjects s1 to s9, and so do the variables v6 to v9 for the subjects s10 to s15. This fact explains why the k-itemsets with non-zero gain extracted by MIDOVA are quasi-exclusively of order 2, because the dominant structure in the table is an opposition between two subject clusters. In this framework, the 2-itemsets that link the variables characterizing one cluster are mainly endowed with a high positive gain, whereas those linking two variables from different clusters tend to have highly negative gains, while being mostly characterized by a zero support. We can conclude that MIDOVA has uncovered the knowledge that had been incorporated in the data, i.e., the existence of two contrasting clusters, a bit blurred with some noise.

G. Performance assessment of MiDOVA vs. Apriori

TABLE IV. NUMBER OF ITEMSETS AS A FONCTION OF 1) THEIR LENGTH, 2) THE ALGORITHM (WBC DATASET, FROM UCI REPOSITORY)

		MIDOVA with all variables			**MIDOVA with class: benign**		
k	***Apriori***	***Mr=0***	***Mr>0***	***Total***	***Mr=0***	***Mr>0***	***Total***
2	4095	1652	2443	4095	23	66	89
3	121485	23750	7931	31681	1119	368	1487
4	2672670	12134	1174	13308	762	83	845
5	46504458	186	0	186	18	0	18
6	666563898	0	0	0	0	0	0
7	8,09 E+9						
…	…						
Tot.	2,48 E+27	37722	11548	49270	1922	517	2439

As an example, we have performed MIDOVA and Apriori on the UCI test-data Wisconsin Breast Cancer (WBC) [9; 10]. This test set comprises ten categorical variables, among which the target variable with the Benign /

Malignant modalities, 8 with ten modalities and 1 with 9 modalities; all these were translated into 91 dichotomous variables, observed for 699 individuals. In Table IV the length of itemsets is in the first column; in the second and third ones are displayed the number of itemsets obtained with Apriori and MIDOVA respectively, with threshold 0, separating those with zero residue, the others, and those who include the target variable with value = Benign.

The Apriori algorithm yields 4095 2-itemsets (see Table IV), which are all the possible combinations of these 91 variables 2 by 2; MIDOVA too because none of these 91 variables could be eliminated, by default of zero residue value. However more than a third of these 4095 itemsets (1652) have a zero residue, and are thus eliminated from the list of itemsets grounding the building of 3-itemsets. At next step (k=3) the number of itemsets grows significantly (31 681, ie. 7.7 times more than at step 2), but much lesser than with Apriori (29.7 times more than at step 2). Among the 3-itemsets generated by MIDOVA, a large proportion has a zero residue (23 750, i.e., more than 70%). From step 3 to step 4 the number of itemsets is multiplied by 22 with Apriori, 0.42 with MIDOVA (see Figure 10). At step 5, 186 itemsets are kept by MIDOVA, all of them with a zero residue, which explains that no itemset of length>5 exists, at the same time when the number of itemsets keeps growing exponentially in Apriori. At last, 1922 itemsets have been kept as candidate variables ("expansion" of the original ones) for predicting the benign target modality.

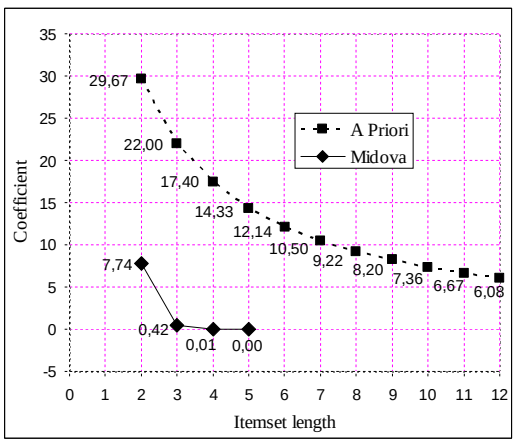

Figure 10. Multipliers for the number of itemsets – an Apriori and MIDOVA comparison (WBC dataset, from UCI repository)

In this way, only a minor part of the 49 270 components of the MIDOVA decomposition is used for the classification task, i.e., the most relevant itemsets in this respect: the ones with Mr=0, Mg≠0 and including the label variable C, thus expressing the tightest relationship with this variable. We will take advantage of this observation in next section III.

H. Effectiveness of MIDOVA

When no threshold support is applied, the maximum complexity at level 2 is O(Ns, Nv^2), where Ns and Nv are the number of subjects and variables respectively. The complexity at the further levels depends heavily on the presence or not of interactions, and their distribution along the successive levels. For example, a structure of simple clusters, i.e., "blocks", mainly results in 2-itemsets, with few higher-order ones. It is the case in our toy example in subsection F, where the only two order-three itemsets may be considered as random noise.

We are aware that many enhancements of the Apriori algorithm have been published since [7]. However these variants do not change, to our knowledge, the basic principles at work, and could also be applied for increasing the efficiency of MIDOVA – this prospect is one of our current interests.

III. NAÏVE BAYES CLASSIFICATION OF THE EXPANDED DATATABLE

We address now the application of MIDOVA to the classification problem. For the sake of simplicity we will tackle the 2-class problem. The class variable C is thus a binary one, and its modality is known for each observation of the learning set. As MIDOVA gives a complete view of how any variable is related to the other ones, C included, we have applied it to the learning set and we have selected the only subset of k-itemsets involving C with 0-valued residue, and extracted their 0-valued components (corresponding to the cells of count 0 in the k-way contingency table). Each component results in a new variable, product of the values 0 or 1 of its variables (except C). We call "MIDOVA expansion" the set of components and "MIDOVA-expanded datatable" the datatable with the new variables.

For the sake of simplicity again we will use the most basic classification approach, i.e., the Naïve Bayes method This approach poses the hypothesis of independent variables, i.e., the log-odd for a data-vector to belong to class C is the sum of the contributions from priors and separate contributions from each of the variables.

Which translates, in our specific case of two classes C and ¬C and binary components of data-vectors $\mathbf{x}=\{x_i\}$:

- For each variable i the contribution s_i writes:
  $$s_i = \log(P(x_i|C)) - \log(P(x_i|\neg C))$$
- For a new data-vector
  $$\text{Evidence}(\mathbf{x}) = \log(P(C)) - \log(P(\neg C)) + \langle \mathbf{x},\mathbf{s} \rangle \quad (1)$$

  where P(.) is a probability, and <.,.> is a dot product.

Our parameter tuning heuristics for optimizing the generalization accuracy criterion, i.e., error percentage, (or F-score variant if necessary) is as follows:

- 0 – We start from the MIDOVA-expanded datatable, whose number of variables depends on our threshold parameters for the gain Mg and residue Mr indices, generally Mg>0, Mr=0.
- 1 – A first pass on the training set yields the ordered list of variables most contributing to the classification, sorted by decreasing s_i unsigned values.

- 2 – A 5-fold (or 6-fold) cross-validation on the training set yields the "optimal cut" Iopt for the number of relevant variables.
- 3 – A last pass on the whole training set, with parameter *Iopt*, yields the optimal value for the evidence threshold *Eopt*.
- 4 – The test set is then classified with formula (1) and parameters *Iopt* and *Eopt*.

IV. EXPERIMENTS

To our knowledge, public access test sets fitting to our requirements of qualitative datasets - binary or categorical – with two classes are uncommon. We present here a benchmarking of our classification method on three UCI repository dataset, All records containing unknown values have been removed. [9, 10, 11]: Tic-Tac-Toe, Wisconsin Breast Cancer, and Monks-2 [12] (known to be the most difficult of the three Monks problems).

As our aim in this paper consists in assessing the soundness of a novel data decomposition, and not in presenting a competitive learning algorithm, we have not tried to assess our MIDOVA application on the many other public access test sets with a k-class output variable (k>2),or with numerical attributes to discretize. This way we avoid 1) further uncertainties in the comparisons due to the discretization steps, and 2) reprogramming other reference, or highly successful, methods, in our present proof-of-concept phase.

A. Tic-Tac-Toe

We have encoded the nine 3-category nominal variables (empty / cross / circle) into 27 binary variables, plus 2 binary variables for the class variable ("win/loose"). 638 instances are in the train set, 320 in the test set. The cross begins the game, and has to play when the given configuration instance appears.

The MIDOVA expansion yields 102 components, each including C or ¬C (i.e., non C), with M*g*>0 and M*r*=0.

After reordering these variables, the Naïve Bayes parameter tuning, with a 6-fold cross-validation, keeps the *Iopt*=32 most relevant ones, with threshold *Eopt*=.8716, resulting in the maximum, but yet attained, accuracy of 100% on the test set.

Note that a variant with 5-fold cross-validation results in 2 test errors (Accuracy=99.37%), and another one with *Iopt*=100 results in 3 errors (Accuracy=99.06%).

Our method allows us to interpret the ordered list of the relevant itemsets: for example, the 4 top ones, with a prominent gain index of 168, encode the four diagonal patterns (O,X,O) and (X,O,X) associated with "loose" (the latter configuration is included in 68 instances, 42 "loose" and 26 "win"); the next 6 ones, with a gain of 144, encodes the trivial cases of three circles aligned in a row or a column, also associated with "loose", and so on…

B. Wisconsin Breast cancer

This dataset consists in 683 patients (train set: 455; test set: 228) described along 9 ordinal scales. Eight of the scales have ten values, and one has nine ones.

For the sake of not losing the orderliness information, we have encoded each of the nine variables as follows: the i^{th} value is encoded by i "1" and 10-i "0" (for example, V1-3 results in {1 1 1 0 0 0 0 0 0 0}).

The MIDOVA expansion on these 89 binary variables yields 1283 components, each including C or ¬C, with Mg>0 and Mr=0.

After reordering these variables, the Naïve Bayes parameter tuning, with a 5-fold cross-validation, keeps the Iopt=130 most relevant ones, with threshold Eopt=.3189, resulting in the maximum, not yet attained by explicit methods to the best of our knowledge, accuracy of 98.24% on the test set (4 errors). The recent reference [13] reports a 99.63% accuracy using a blind method (Artificial Metaplasticity Multilayer Perceptron)

Note that a variant with a standard binary coding scheme results in 5 errors (Accuracy=.9781).

Like any rule-based method, ours allows a medical expert to interpret the ordered list of the relevant itemsets, which top elements are:

Malignant←V2.5,V4.2	Malignant←V3.5,V6.4,V7.4
Malignant←V2.5,V7.4	Malignant←V3.5,V7.5
Malignant←V2.5,V4.3	……
Malignant←V6.4 ,V4.2	Benign←V2.5,V3.4
Malignant←V2.6,V4.2	……
Malignant←V6.8,V7.4	Benign←V1.4,V6.4,V7.4
Malignant←V2.5,V7.5	……

C. Monks-2

Monks2 is the harder of the three Monks problems: the solution cannot be described simply as a conjunction of disjunctions, it needs a method for pulling the concept of "exact number (n) of variables with value 1 amongst m ones" out of sample data.

We have encoded the six 3 nominal variables into 19 binary variables, plus 2 binary variables for the class variable ("two features/else"). 169 instances are in the train set, 432 in the test set.

The MIDOVA expansion yields 99 components, each including C or ¬C, with Mg>0 and Mr=0.

After reordering these variables, the Naïve Bayes parameter tuning, with a 5-fold cross-validation, keeps all of the Iopt=99 of them, with threshold Eopt=1.6994, resulting in the honorable accuracy of 71.5% on the test set: in the review [12], 9 symbolic learning techniques upon 24 result in a clearly better score. We are aware of only one SVM method [14] resulting in a better score (85.3%)..

Our method is clearly adapted to detecting classification rules expressed as conjunctions of disjunctions, not to more sophisticated hypotheses. But our experience is that this ability is enough for most of the real-life problems in the domain of supervised learning.

V. RELATED METHODS

Since the very beginning of this paper, we have continuously compared our method to Apriori: let us recall that our main objective here is to expose a novel representation of a 0/1 database, made of "salient" itemsets, close to the representation issued from Apriori, but with a very different definition of "salient".

First, we will summarize the main similarities:

- Both are levelwise methods, starting from order-1 itemsets for building order-2, order-3, …ones.

- Both aim at extracting the "local" information embedded in the interactions between two and more variables, resulting in a representation far from the global "datacloud" scheme of most of the data analysis methods.

- Both use anti-monotone properties: as regards to Apriori, the support of an itemset never exceeds the support of its subsets; concerning MIDOVA, the residue of an itemset never exceeds the residue of its subsets.

Then, the main differences are as follows:

-An implicit hypothesis needed by Apriori for giving rise to tractable computations is that each instance has a description of a "pick-any" type: it consists of a small number of items picked among a large number of potential ones, as it is the case for "market basket" data or language data – hence itemsets never include *absent* items in real-life applications. On the contrary MIDOVA is well-fit for any type of boolean data, whether pick-any or not, for it takes symmetrically into account both presence or absence of an item.

- The general principle of both methods are different: Apriori operates by counting the occurrences of combinations of the items, while MIDOVA condenses the 2^K facets of the huge K-way contingency table implicitly defined by any N × K boolean datatable into a list of its essential facets, where essential means "necessary and sufficient for rebuilding the datatable".

- Both aim at minimizing the number of extracted itemsets, but Apriori uses frequency thresholds, while MIDOVA uses other criteria, preserving the cases where interesting itemsets may be unfrequent, if not absent (it is the case of the XOR function, and more generally of situations of exclusiveness – in a medical context, to characterize health may be as important as characterizing illness; see the *benign/malignant* example above).

- As a stopping criterion, Apriori uses support thresholds, while MIDOVA uses "residue", a rigorous measure of the association potential of a considered itemset.

As regard to the illustrative part of our paper, which concerns the classification problem, we will review a few families of methods close to our classification scheme:

- The principles of the Association Rule-based Classifiers are as follows: 1) they start, as we do, from a Boolean matrix including the target variable, 2) they mine all the high-confidence association rules with a single variable in the right part – we use our novel MIDOVA process instead, 3) they filter the only rules implying the target variable, as we do, 4) they implement a rule-ordering strategy, generally based on support and confidence, instead of our Naïve Bayes-based expansion/selection process. The Large Bayes method of [6] is a salient reference in the 90's. The references [15] and [4] report maximal accuracy rates of respectively 93.95% and 95.1% on WBC data, 92.6% and 98.2% on Tic-Tac-Toe, and no results on the Monks problems. Harmony [16] has taken over in the 2000's. It uses an instance-centric rule generation framework where the ordering of local rule lists is based on confidence, entropy or correlation criteria. At the end of the process and for each class, these lists are merged and sorted by the chosen criterion: when an unknown test instance is presented, the sums of the criterion for the k first relevant rules in each class are computed and compared, determining then the presumed class label. The recent reference [17] reports a 95.85% Harmony score for WBC data, and 97.98 for TTT. It presents a general scheme close to ours: the authors first create new features (based on frequency criteria, unlike ours) for expanding the data, then they use classic learning methods, among which Naïve Bayes, for the classification task. Their results with and without their "Feature Creation" (FC) expansion are reported in the recapitulative Table V.

Learning Classifier Systems (LCS) [5] are not as close to our method as it could seem at first glance: these *incremental* data-streaming algorithms use genetic optimization for the selection of best-fitted classification rules. This problem being harder than our batch-processing objective, no surprise that a 95,5% accuracy has been reported on WBC data [18].

TABLE V. REPORTED ACCURACIES FOR WISCONSIN BREAST CANCER, TICTACTOE AND MONKS2 DATASETS ([a]: REPORTED IN [17])

Method	**WBC**	**TTT**	**Monks2**
Naïve Bayes	97.88 [a]	68.47 [a]	67.0
Naïve Bayes + FC	96.59 [a]	79.72 [a]	n.a.
Harmony	95.85 [a]	97.98 [a]	n.a.
(CBA (Classification Based on Associations)	93.95	92.6	n.a.
GARC (Gain based Association Rule Classification)	94.8	***100.0***	n.a.
LCS (Learning Classifier System)	95.5	n.a.	n.a.
Naïve Bayes + MIDOVA	***98.24***	***100.0***	***71.5***
Maximum reported performance with blind methods	99.58	100.0	85.3

VI. CONCLUSIONS AND PERSPECTIVES

We have presented in this text a novel representation scheme for qualitative data sets: a list of frequent and infrequent itemsets condensing all the noticeable, non-trivial information embedded in the interactions between Boolean variables. We have shown that this list is far less prone to the combinatorial explosion than the one resulting from the Apriori algorithm and that the maximum order of the interesting itemsets is limited to $\log_2 N+1$, N being the number of instances in the database.

For proving the quality of this representation, we have decided to put it into practice in a supervised framework, in which a quantitative assessment is possible, specifically in the framework of the two-class discrimination problem. For this purpose, we have selected the subset of itemsets related to the class variable, resulting in an expanded datatable. We have chosen the Naïve Bayes classification method for providing the explicit discrimination criterion we wished and assess the quality of our data expansion.

The results on three open-access test datasets are satisfactory:we have proven on two test datasets (WBC and Tic-Tac-Toe) that, as a matter of accuracy performance, our derived classification method could compete with SVMs, while reaching the same human readability of the results as Learning Classifier Systems or Classification Association Rules. Honorable results were obtained on the Monks2 problem, which is an artificial test bench for general artificial intelligence.

Apart from developing non-supervised applications of our representation method, such as data-driven modeling, our middle-term prospects are many in the machine learning domain:

- Classify more than two classes and include splits on numerical variables, which could multiply our possible test benches, and outline more precisely the qualities and limits of our approach; we already know that on the Monks-2 dataset, our performance is good, but not excellent.
- Scale-up the implementation of our algorithm, for tackling real-life problems.
- Use another selection and classification method as Naïve Bayes, if necessary.
- Increase our theoretical understanding of the method, and bridge the gap with statistical learning approaches.

REFERENCES

[1] Cadot M. and Lelu A. 2010. Optimized Representation for Classifying Qualitative Data. DBKDA 2010, pp. 241-246

[2] Guermeur Y. 2002. Combining discriminant models with new multi-class SVMs. Pattern Analysis and Applications (PAA), Vol. 5, N. 2, pp. 168-179.

[3] Naïm P., Wuillemin P. H., Leray P., Pourret O. and Becker A. 2007. Réseaux bayésiens, Collection Algorithmes, Eyrolles, Paris.

[4] Chen, G., Liu, H., Yu, L., Wei, Q., and Zhang, X. 2006. A new approach to classification based on association rule mining. Decis. Support Syst. 42, 2 (Nov. 2006), pp. 674-689. (DOI= http://dx.doi.org/10.1016/j.dss.2005.03.005, 2012-06-01)

[5] Bull L. (Editor), Bernada-Mansilla Ester (Editor), John Holmes (Editor), , 2008. Learning Classifier Systems In Data Mining, Springer

[6] Meretakis D. and Wuthrich B. 1999. Classification as mining and use of labeled itemsets. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD-99).

[7] Agrawal R. and Srikant. R. 1994. Fast algorithms for mining association rules in large databases, Research Report RJ 9839, IBM Almaden Research Center, San Jose, California.

[8] Cadot, M. 2006. Extraire et valider les relations complexes en sciences humaines : statistiques, itemsets et règles d'association. Ph. D. thesis, Université de Franche-Comté (DOI= http://www.loria.fr/~cadot/cadot_these_2006.pdf , 2012-06-01)

[9] **UC Irvine Machine Learning Repository** (http://archive.ics.uci.edu/ml/,2012-06-01).

[10] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 , 2012-06-01.

[11] http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame, 2012-06-01.

[12] Thrun S. and al. 1991. The MONK's Problems: A Performance Comparison of Different Learning Algorithms. S. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R.S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang. (DOI= http://robots.stanford.edu/papers/thrun.MONK.ps.gz, 2012-06-01)

[13] Marcano-Cedeño A., Quintanilla-Domínguez J., Andina D., 2010. Breast cancer classification applying artificial metaplasticity algorithm, Neurocomputing, Volume 74, Issue 8, pp. 1243-1250.

[14] Rüping S. 2001. Incremental learning with support vector machines, in Proceedings of the 2001 IEEE International Conference of Data Mining.

[15] Rajanish Dass. 2008. Classification Using Association Rules, IIMA Working Papers 2008-01-05, Indian Institute of Management Ahmedabad, Research and Publication Department. Downloadable in RePec: (DOI= http://ideas.repec.org/p/iim/iimawp/wp02079.html, 2012-06-26)

[16] Wang J. and Karypis G. 2005. HARMONY: Efficiently Mining the Best Rules for Classification. SIAM International Conference on Data Mining, pp. 205-216

[17] Gay D., Selmaoui-Folcher N., Boulicaut J.F. 2012. Application-independent feature construction based on almost-closedness properties, Knowledge and Information Systems, Volume 30, Issue 1, pp.87-111, Springer.

[18] Wilson, S. W. 2002. Compact rulesets from XCSI, in Advances in learning classifier systems: Fourth international workshop, IWLCS 2001. (LNAI 2321), P. L. Lanzi, W. Stolzmann, and S. W. Wilson, Eds. Springer-Verlag, pp. 196–208.

ANNEX

For the sake of comparability with other methods: K-itemsets extracted by MIDOVA out of the data in Table 1

k: #_items	k-itemset	Support	Gain	Residue	Frozen ?	Interesting ?	Interaction sign
1	{v01}	15	7.5	0	*		
1	{v02}	7	-0.5	7			
1	{v03}	7	-0.5	7			
1	{v04}	5	-2.5	5			
1	{v05}	6	-1.5	6			
1	{v06}	4	-3.5	4			
1	{v07}	6	-1.5	6			
1	{v08}	4	-3.5	4			
1	{v09}	5	-2.5	5			
1	{v10}	0	-7.5	0	*		
2	{v02; v03}	5	3	4		*	+
2	{v02; v04}	3	1	4		*	+
2	{v02; v05}	4	2	4		*	+
2	{v02; v06}	0	-4	0	*	*	-
2	{v02; v07}	1	-4	2		*	-
2	{v02; v08}	1	-2	2		*	-
2	{v02; v09}	1	-3	2		*	-
2	{v03; v04}	2	-1	4		*	-
2	{v03; v05}	3	0	6			
2	{v03; v06}	0	-4	0	*	*	-
2	{v03; v07}	0	-6	0	*	*	-
2	{v03; v08}	2	0	4			
2	{v03; v09}	2	-1	4		*	-
2	{v04; v05}	3	1	4		*	+
2	{v04; v06}	0	-4	0	*	*	-
2	{v04; v07}	1	-3	2		*	-
2	{v04; v08}	1	-2	2		*	-
2	{v04; v09}	2	-1	4		*	-
2	{v05; v06}	0	-4	0	*	*	-
2	{v05; v07}	1	-4	2		*	-
2	{v05; v08}	0	-4	0	*	*	-
2	{v05; v09}	0	-5	0	*	*	-
2	{v06; v07}	4	4	0	*	*	+
2	{v06; v08}	1	-2	2		*	-
2	{v06; v09}	2	0	4			
2	{v07; v08}	2	0	4			
2	{v07; v09}	3	1	4		*	+
2	{v08; v09}	3	2	2		*	+
3	{v02; v03; v04}	2	2	0	*	*	+
3	{v02; v03; v05}	2	-2	0	*	*	-
3	{v02; v03; v08}	1	0	0	*		
3	{v02; v03; v09}	1	0	0	*		
3	{v02; v04; v05}	2	0	4			

3	{v02; v04; v07}	0	0	0	*
3	{v02; v04; v08}	0	0	0	*
3	{v02; v04; v09}	1	0	0	*
3	{v02; v05; v07}	1	0	0	*
3	{v02; v07; v08}	0	0	0	*
3	{v02; v07; v09}	0	0	0	*
3	{v02; v08; v09}	0	0	0	*
3	{v03; v04; v05}	1	0	4	
3	{v03; v04; v08}	0	0	0	*
3	{v03; v04; v09}	1	0	4	
3	{v03; v08; v09}	1	0	0	*
3	{v04; v05; v07}	0	0	0	*
3	{v04; v07; v08}	1	0	0	*
3	{v04; v07; v09}	1	0	0	*
3	{v04; v08; v09}	1	0	0	*
3	{v06; v08; v09}	1	0	0	*
3	{v07; v08; v09}	2	0	0	*