Condensed Representations in Presence of Missing Values

François Rioult and Bruno Crémilleux

GREYC, CNRS - UMR 6072, Université de Caen
F-14032 Caen Cédex France
{Francois.Rioult, Bruno.Cremilleux}@info.unicaen.fr

Abstract. Missing values are an old problem that is very common in real data bases. We describe the damages caused by missing values on condensed representations of patterns extracted from large data bases. This is important because condensed representations are very useful to increase the efficiency of the extraction and enable new uses of frequent patterns (e.g., rules with minimal body, clustering, classification). We show that, unfortunately, such condensed representations are unreliable in presence of missing values. We present a method of treatment of missing values for condensed representations based on δ-free or closed patterns, which are the most common condensed representations. This method provides an adequate condensed representation of these patterns. We show the soundness of our approach, both on a formal point of view and experimentally. Experiments are performed with our prototype MVminer (for Missing Values miner), which computes the collection of appropriate δ-free patterns.

1 Introduction

Context. Missing values in data bases are an old problem that always arises in presence of real data, for instance, in the medical domain. With regard to opinion polls, it is rare that interviewees take the pain to fill entirely the questionnaire. It is a strong problem because many analysis methods (e.g., classification, regression, clustering) are not able to cope with missing values. The deletion of examples containing missing values can lead to biased data analysis. Elementary techniques (e.g., use of the mean, the most common value, default value) are not more satisfactory, because they exaggerate correlations [7]. At last some treatments are devoted to specific databases [8], but it is difficult to apply them in the general case and it is clear that there is not a single, independent and standard method to deal with missing values. We will see that this problem occurs also on condensed representations.

Condensed representations of frequent patterns provide useful syntheses of large data sets [4,12], highlighting the correlations embedded in data. There is a twofold advantages of such an approach. First, it allows to improve efficiency of algorithms for usual tasks such as association rules [1]. Even if this technique is, today, well mastered, the use of condensed representations enables to

M.R. Berthold et al. (Eds.): IDA 2003, LNCS 2810, pp. 578–588, 2003.

achieve extraction of rules in contexts where usual Apriori-like algorithms fail [3,12]. Second, condensed representations enable multiple uses of frequent patterns [9,6,16] (e.g., strong rules, informative rules or rules with minimal body, non-redundant rules, clustering, classification) which is a key point in many practical applications. These uses are today required by experts on data who know that the whole value embedded in their data can only be acquired by the new developments of such analysis methods.

Motivations. In real data bases, users have to cope with missing values and we are going to see that, unfortunately, condensed representations are no more valid in presence of missing values. This is the starting point of our work. Let us give an example: the left part of Table 1 (called r or the reference table) provides an example of a transactional database composed of 7 transactions (each one identified by its Tid) and 5 items denoted $A \dots E$. For instance, in medical area (e.g., Hodgkin's disease), the item A denotes an item which means `‘‘Bsymptoms’’`[1] `= present`, the item B means `mediastinum = enlarged`, and so on. This table is used as the running example throughout the paper. The right part of Table 1 provides the condensed representation based on 0-free sets with an absolute support threshold of 2. For each 0-free, we indicate its closure. These notions are explained in Section 2.

Table 1. Running example of a database without missing values (r)

r

Tid	Items
1	A D
2	C E
3	A B C D E
4	A D
5	A B D E
6	A B D E
7	A B C D E

0-free set	closure	0-free set	closure
A	$\{D\}$	AC	$\{B, D, E\}$
B	$\{A, D, E\}$	AE	$\{B, D\}$
C	$\{E\}$	BC	$\{A, D, E\}$
D	$\{A\}$	CD	$\{A, B, E\}$
E		DE	$\{A, B\}$

For instance, from this condensed representation, we can extract rules such $AE \Rightarrow BD$ with a support of 4 and a confidence of 100%. Now, let us suppose that some parameters used in the definition of `‘‘Bsymptoms’’` are not been caught, so we do not know for instance whether `‘‘Bsymptoms’’ = present` is true or not. Then, missing values appear and we use the character `’-’` before an item to note such a situation. The left part of Table 2 (called $\tilde{r}$) includes 5 missing values.

How to deal with missing values? Elementary methods remove data with missing values but it may lead to a biased data set from which extracted information is unreliable. Let us see that in our example. The right part of Table 2

[1] Bsymptoms are features of lymphoma, and include fever, drenching night sweats, weight loss more than 10% of body mass in previous 6 months

Table 2. Running example of a database with missing values ($\tilde{r}$)

$\tilde{r}$

Tid	Items				
1	A			D	
2			C		E
3	A	B	C	D	E
4	$-A$			D	
5	A	B		$-D$	E
6	$-A$	$-B$		D	E
7	A	B	C	D	$-E$

0-free set	closure	0-free set	closure	0-free set	closure
A		AD		ABC	$\{E\}$
B		AE	$\{D\}$	ABD	$\{E\}$
C	$\{E\}$	BC	$\{E\}$	ABE	$\{D\}$
D		BD	$\{E\}$	ACD	$\{E\}$
E		BE	$\{D\}$	BCD	$\{E\}$
AB		CD	$\{E\}$	$ABCD$	$\{E\}$
AC	$\{E\}$	DE	$\{A, B\}$		

depicts the condensed representation on $\tilde{r}$ achieved with this strategy. This condensed representation contains new patterns which are not present in the condensed representation of reference (extracted on r). We will qualify (see Section 2.3) such patterns of *pollution*. We also note that a lot of items have disappeared from the almost-closures. Consequently, rules such as $AE \Rightarrow BD$ are no longer found. Clearly, missing values are responsible for the removal of relationships and the invention of new groundless associations. Experiments in Section 4 show that this result is general and it is clear that usual condensed representations cannot safely be used in presence of missing values. The aim of this paper is to propose a solution to solve this open problem.

Contributions. The contribution of this paper is twofold. First, we describe the damages caused by missing values in condensed representations. Second, we propose a method of treatment of missing values for condensed representations based on δ-free or closed patterns (which are the most common condensed representations). This method avoids these damages. We show the soundness of the obtained condensed representations, both on a formal point of view and experimentally. We think that this task is important in data mining owing to the multiple uses of condensed representations.

Organization of the paper. This paper is organized as follows: Section 2 briefly reviews the condensed representations of δ-free patterns and we set out in Section 2.3 the effects of missing values on these representations. In Section 3 we describe the corrections (and we give a formal result) that we propose so as to treat missing values. Section 4 presents experiments both on benchmarks and real world data (medical database on Hodgkin's disease).

2 Condensed Representations

We give the necessary material on condensed representations which is required for the rest of the paper and we present the negative effects of missing values.

2.1 Patterns Discovery

Let us consider a transactional database: a database r is a set of transactions t composed of items. In Table 1 $r = \{t_1, \ldots, t_7\}$ where $t_1 = AD$ (note that we use a string notation for a set of items, e.g., AD for $\{A, D\}$), $t_2 = CE$, etc. Let Z be a *pattern* (i.e. a set of items), an association rule based on Z is an expression $X \Rightarrow Y$ with $X \subset Z$ and $Y = Z \backslash X$.

The support of X with respect to a set of transactions r is the number of transactions of r that contain X, i.e. $supp(X, r) = |\{t \in r \mid X \subseteq t\}|$. We note r_X the subset of transactions of r containing X, we have $supp(X, r) = |r_X|$. If r is clear from the context, we will use $supp(X)$ for $supp(X, r)$. The confidence of $X \Rightarrow Y$ is the proportion of transactions containing X that also contain Y [1], i.e. $conf(X \Rightarrow Y) = supp(X \cup Y)/supp(X)$.

2.2 δ-Free Patterns

Due to the space limitation, we give here only the key intuition of δ-free patterns and the background required to understand the effects of missing values. We start by defining δ-strong rules.

Definition 1 (δ-strong rule) *A δ-strong rule is an association rule of the form $X \Rightarrow Y$ that admits a maximum of δ exceptions [13,3].*

The confidence of such a rule is at least equal to $1 - (\delta/supp(X))$.

Definition 2 (δ-free pattern) *A pattern Z is called δ-free if there is no δ-strong rule $X \Rightarrow Y$ (with $X \subset Z$ and $Y = Z \backslash X$) that holds.*

The case $\delta = 0$ (corresponding to 0-free patterns) is important: no rule with confidence equal to 1 holds between proper subsets of Z. For instance, DE is a 0-free pattern because all rules constructed from proper subsets of DE have at least one exception. If $\delta = 1$, DE is not a 1-free set owing to the rule $E \Rightarrow D$ which has only one exception. From a technical perspective, δ-strong rules can be built from δ-free patterns that constitute their left-hand sides [4]. δ-free patterns are related to the concept of almost-closure:

Definition 3 (almost-closure) *Let δ be an integer. $AC(X, r)$, the almost-closure of X in r, gathers patterns Y so that :*

$$supp(X, r) - supp(X \cup Y, r) \leq \delta \tag{1}$$

Note that if $X \Rightarrow Y$ is a δ-strong rule in r, items of Y belong to $AC(X, r)$. In other words, when an item belongs to $AC(X, r)$, it means that it is present in all the transactions that contain X with a number of exceptions bounded by δ. Following our example given in Table 1, $D \in AC(E, r)$ with $\delta = 1$ (there is only one exception, transaction t_4). δ-freeness satisfies a relevant property (anti-monotonous constraint) and we get tractable extractions for practical mining tasks that are not feasible with Apriori-like algorithms [4].

A collection of frequent δ-free patterns is a condensed representation of the collection of frequent patterns. If $\delta = 0$, one can compute the support of every frequent pattern. In this case, the almost-closure corresponds to the special case of the *closure*. Such closed patterns have relevant properties to compute informative [2] or non-redundant [16] rules, or achieve clustering [6]. If $\delta > 0$, one can approximate the support of every frequent pattern X with a bounded error: in [3], it is shown that the error is very low in practice. δ-free patterns verify suitable properties to build δ-strong rules with a high value of confidence [3] and rules characterizing classes [5].

2.3 Effects of the Missing Values on the Condensed Representations Based on δ-Free Patterns

Missing values produce effects on δ-free patterns and items of almost-closures. Let us suppose that an item A belongs to the almost-closure of a δ-free X. It means that A is always present with X except a number of exceptions lower than δ. If missing values on A occur, this number of exceptions can only increase and can become higher than δ: then, A comes out of the almost-closure and the $X \cup \{A\}$ pattern becomes free (see Figure 1). In our example, with $\delta = 0$, B and D belong to the closure of AE on r whereas B comes out of this closure on $\tilde{r}$ owing to the missing value on transaction t_6. Moreover, AB and BE become free and such created free patterns are qualified of *pollution*.

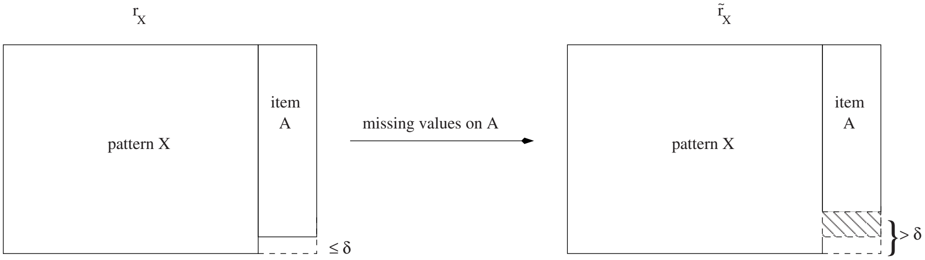


Fig. 1. Missing values on items of almost-closures

3 Corrections of the Missing Values

We present here our corrections on support and almost-closure to deal with missing values. We start by giving the notion of disabled data which is a key step in our corrections.

3.1 Disabled Data

In presence of missing values, support values decrease [14]. For instance, in our example, $supp(DE, r) = 4$ but $supp(DE, \tilde{r}) = 2$. In fact, to compute properly the support of an itemset in $\tilde{r}$, it is necessary to distinguish the transactions of $\tilde{r}$ having at least one missing value among the items of X. These transactions will be temporarily disabled to compute $supp(X, \tilde{r})$ because they do not enable to take a decision to this support. It is shown that this approach allows to retrieve relevant values of support [14].

Definition 4 (disabled data) *A transaction t of $\tilde{r}$ is disabled for X if t contains at least one missing value among the items of X. We note $Dis(X, \tilde{r})$ the transactions of $\tilde{r}$ disabled for X.*

With this approach, the whole data base is not used to evaluate *a* pattern X (owing to the missing values on X) but, as data are only temporarily disabled, finally the whole data base is used to evaluate *all* patterns.

3.2 Correction of the Effect of Missing Values

In case of missing values, we have to take into account, with respect to X, the disabled data of a candidate pattern Y to test if $Y \subseteq AC(X, \tilde{r})$. We propose to redefine the almost-closure of X as follows:

Definition 5 *$AC(X, \tilde{r})$, the almost-closure of X in $\tilde{r}$, gathers patterns Y so that :*

$$supp(X, \tilde{r}) - supp(X \cup Y, \tilde{r}) \leq \delta + Dis(Y, \tilde{r}_X) \tag{2}$$

Of course, this definition makes only sense if there remains at least one transaction in $\tilde{r}$ containing both X and Y.

Let us note that if there is no missing value, $Dis(Y, \tilde{r}_X) = 0$ and $r = \tilde{r}$ and we recognize the usual definition of the almost-closure (see Definition 3). This new definition is fully compatible with the usual one when there are no missing values. Inequality 2 can be seen as a generalization of Inequality 1 (Section 2.2). Inequality 2 can be interpreted as a local relaxation of the constraint on δ (i.e. δ is adjusted for each Y) according to the number of missing values on Y in $\tilde{r}_X$. This definition leads to the important following property:

Property 1 *Definition 5 of the almost-closure is sound in presence of missing values, i.e. $Y \subset AC(X, r) \Rightarrow Y \subset AC(X, \tilde{r})$. Then, by using this definition, the effect of missing values (loss of items from almost-closures and pollution of δ-free patterns (cf. Section 2.3)) is corrected.*

The two following lemmas help to prove this property.

Lemma 1 $supp(X, \tilde{r}) = supp(X, r \backslash Dis(X, \tilde{r})) = supp(X, r) - |Dis(X, \tilde{r})|$

Proof The claim follows directly from Definition 4.

Lemma 2 $supp(X \cup Y, \tilde{r}) = supp(X \cup Y, r) - |Dis(X, \tilde{r})| - |Dis(Y, \tilde{r}_X)|$

Proof The claim follows directly from Lemma 1 and the fact that $|Dis(X \cup Y, \tilde{r})| = |Dis(X, \tilde{r})| + |Dis(Y, \tilde{r}_X)|$.

It is now easy to prove Property 1.
Proof: The effect arises when Y is in $AC(X, r)$ but no more in $AC(X, \tilde{r})$. Consider $Y \subset AC(X, r)$, then $supp(X, r) - supp(X \cup Y, r) \leq \delta$. Adding the terms $|Dis(X, \tilde{r})|$ and $|Dis(Y, \tilde{r}_X)|$ to this inequality, we get:

$$\begin{aligned} &supp(X, r) - |Dis(X, \tilde{r})| - (supp(X \cup Y, r) - |Dis(X, \tilde{r})| - |Dis(Y, \tilde{r}_X)|) \\ &\quad \leq \delta + |Dis(Y, \tilde{r}_X)| \end{aligned}$$

Lemmas 1 and 2 enable to write $supp(X, \tilde{r}) - supp(X \cup Y, \tilde{r}) \leq \delta + Dis(Y, \tilde{r}_X)$ and we recognize Inequality 2. Thus $Y \subset AC(X, r)$ implies $Y \subset AC(X, \tilde{r})$: Definition 5 is sound in presence of missing values and Property 1 holds.

This property assures to perfectly recover items in the almost-closures in the presence of missing values otherwise they may come out. In our example, this method fully recovers on $\tilde{r}$ the condensed representation of reference (given in the right part of Table 1).

4 Experiments and Results

The purpose of this section is to compare condensed representations achieved by our corrections versus condensed representations obtained without corrections. Experiments are performed with our prototype MVminer which extracts δ-free patterns with their almost-closures according to our corrections (MVminer can be seen as an instance of the level-wise search algorithms presented in [10]).

4.1 The Framework of the Experiments

The data base without missing values is the reference data base and it is denoted r. Let us call *reference condensed representation* the condensed representation of reference performed on r. Then, some missing values are randomly introduced in r and the data base with missing values is noted $\tilde{r}$. We are going to discuss of the condensed representation of $\tilde{r}$ obtained by the elementary method (i.e. ignore the missing values) and the one achieved with our corrections (i.e. running MVminer on $\tilde{r}$) compared to the reference condensed representation. For the following, our method with corrections is simply called MVminer and the elementary is called "the usual method". Used measures are: pollution of δ-free patterns and recovery of items of almost-closures, differences on the values of support between the usual method and MVminer, pollution on 0-free patterns according to the rate on missing values.

Experiments have been done both on benchmarks (usual data bases used in data mining and coming from the University of Irvine [11]) and on a real database provided by the European Organization of Research and Treatment of Cancer (EORTC). All the main results achieved are similar and, due to the lack of space, we choose to present here only results on EORTC data base (the results on most data bases are available in [15]). EORTC has the advantage to be a real data base on a problem for which physicians have a great interest. This data base gathers 576 people suffering from Hodgkin's disease, a gland cancer. There are 26 multi-variate attributes, which bring 75 highly correlated binary items.

The protocol of experimentation suggests the use of three parameters: the rate of artificially introduced missing values, the minimum support used in the extraction and the value of δ. Actually, experiments show that only the variation of δ produces changes of tendency [15]. So, we present, below, the results on experiments with 10% of missing values per attribute (randomly introduced), a minimum support of 20%. δ varies from 0 to 20 transactions (i.e. 0 to 3.5%).

4.2 Correction on Almost-Closures

The phenomena described in this section are linked to the effect of missing values, previously described Section 2.3. Figure 2 (on the left) depicts the proportion of δ-free patterns obtained in $\tilde{r}$ by the usual method and *not* belonging to the reference condensed representation (i.e. pollution of δ-free patterns). The usual method undergoes a pollution of 50% as soon as the first values of δ (1.5% or 8 transactions). It means that there are a lot of meaningless patterns which are, obviously, impossible to distinguish from the true patterns. Such a result shows a great interest of MVminer: we note there is no pollution with MVminer for any δ value. It is clear that this result was expected owing to the property given in Section 3.2.

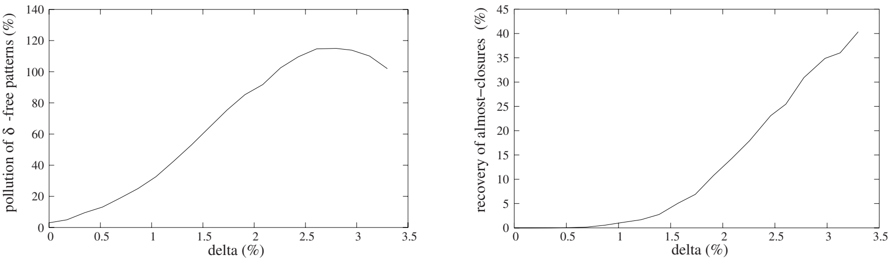
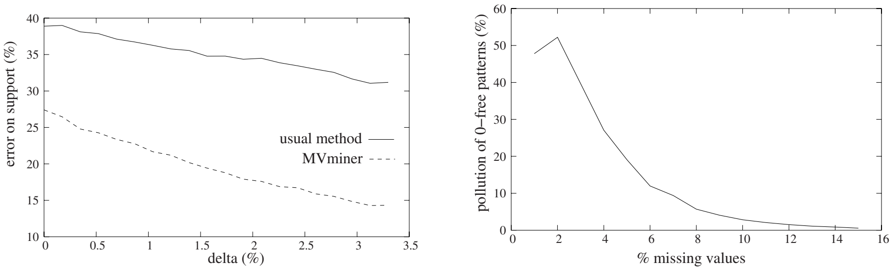


Fig. 2. Drawbacks of the usual method

Figure 2 (on the left) indicates that the level of pollution with the usual method stays low when $\delta = 0$ or is very small and then we can imagine using this method. Figure 2 (on the right) deletes this hope: items of almost-closures are not recovered. Even with high values of δ, only half of the items of almost-closures are found. With corrections, MVminer fully recovers all items of almost-closures. This shows that it is impossible to trust condensed representations given by the usual method even with a low value of δ and thus highlights again the advantages of MVminer.

4.3 Support of δ-Free Patterns and Recovery of 0-Free Patterns According to the Rate of Missing Values

Figure 3 (on the left) compares errors on values of the supports of δ-free patterns between the usual method and MVMINER. Error is about twice as low as with MVMINER. Such an improvement may be precious to compute measures of interestingness on rules.



Error on support: usual method versus MVMINER

Pollution of 0-free (closed) patterns (usual method)

Fig. 3.

Let us consider now the pollution brought by the usual method on the condensed representation made up of 0-free patterns (a very often used condensed representation) according to the rate of missing values. This pollution is the proportion of obtained patterns and not belonging to the reference condensed representation. Missing values were introduced on each item according to the percentage indicated on Figure 3. This figure (on the right) shows a high value of pollution as soon as the first missing values are introduced (50% of meaningless patterns). With a lot of missing values, pollution decreases because few patterns are recovered but the condensed representation is groundless. Let us recall that there is no pollution on patterns with MVMINER (cf. Section 4.2).

5 Conclusion and Future Work

We have presented the damages due to missing values on condensed representations based on δ-free patterns and closed patterns which are the most common condensed representations. Without processing, such condensed representations are unreliable in presence of missing values which prevent the multiple uses of extracted patterns. Our analysis clarifies the effects of missing values on δ-free patterns and their almost-closures. We have proposed corrections and a sound new definition of the almost-closure which is a generalization of the usual one and fully compatible with this one. These corrections deal with missing values and recover initial patterns with their almost-closures. The corrections never introduce pollution on patterns. As they check the property of anti-monotonicity, these corrections are effective even in the case of huge, dense and/or highly correlated data. Experiments on real world data and benchmarks show that it is impossible to use trustingly condensed representations without corrections (e.g. high level of pollution on δ-free patterns even with low rates of missing values) and confirm the relevance of the corrections.

A straightforward further work is to use these condensed representations suitable for missing values to determine reliable rules to predict missing values. Such condensed representations have good properties for this task (see for instance [2]). Our collaborations in medicine provide excellent applications.

Acknowledgements. The authors wish to thank Dr M. Henry-Amar (Centre François Baclesse, France) for providing data on Hodgkin's disease and many valuable comments.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *Proc. of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
2. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. *Proceedings of the 6th International Conference on Deductive and Object Databases (DOOD'00)*, pages 972–986, 2000.
3. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *Principles of Data Mining and Knowledge Discovery (PKDD'00)*, pages 75–85, 2000.
4. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets : a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, 7(1):5–22, 2003. Kluwer Academics Publishers.
5. B. Crémilleux and J.F. Boulicaut. Simplest rules characterizing classes generated by delta-free sets. *proceedings of the 22nd Annual International Conference of the Specialist Group on Artificial Intelligence (ES'02), Springer, Cambridge, United-Kingdom*, December 2002.
6. N. Durand and B. Crémilleux. Ecclat : a new approach of clusters discovery in categorical data. *22nd Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 177–190, December 2002.
7. J. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. *International Conference on Rough Sets and Current Trends in Computing, Banff, Canada*, 2000.
8. S. Jami, X. Liu, and G. Loizou. Learning from an incomplete and uncertain data set: The identification of variant haemoglobins. *The Workshop on Intelligent Data Analysis in Medicine and Pharmacology, European Conference on Artificial Intelligence, Brighton, 23th-28th August*, 1998.
9. H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *Knowledge Discovery and Data Mining*, pages 189–194, 1996.
10. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

11. Set of databases for the data mining. Provided by the university of californy, irvine. *http://www.ics.uci.edu/˜mlearn/MLRepository.html*.
12. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.
13. G. Piatetsky-Shapiro. Discovery, analysis and presentation of strong rules. *Knowledge Discovery in Databases*, pages 229–248, 1991.
14. A. Ragel and B. Crémilleux. Treatment of missing values for association rules. *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'98), Melbourne, Australia*, pages 258–270, April 1998.
15. F. Rioult. Représentation condensée pour les bases de données adéquate aux valeurs manquantes. *Technical Report, 60 pages, GREYC, Université de Caen*, 2002.
16. Mohammed J. Zaki. Generating non-redundant association rules. *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston*, pages 34–43, 2000.