Optimisation of pattern mining : a new method founded on database transposition

François Rioult   Bruno Crémilleux

Université de Caen - GREYC - CNRS UMR 6072
Campus Côte de Nacre
F-14032 Caen Cedex, France

{Francois.Rioult, Bruno.Cremilleux}@info.unicaen.fr

Abstract

Looking in a database for interesting patterns of attributes (the columns) or groups of objects (lines) that verify some properties is a classic task in data mining, today well mastered. Nevertheless, some difficult contexts such as the data provided by gene analysis remain intractable, because of a disproportionate number of attributes, compared to the number of objects. In these conditions, it is naturally tempting to transpose the matrix of data to carry out more efficiently the pattern mining. This article exposes this new method and shows its interest but also the difficulties to solve so that this approach is fruitful. With using the Galois connection, the extraction achieved in the transposed base allows to infer results on the initial data. We detail the contributions of this practice on data containing a big number of attributes, such as data of genome, as well as its possible application to the mining under monotonous constraint and the obtaining of the totality of the closed patterns.

Keywords :

Closed patterns mining, concept lattice, Galois connection, levelwise algorithm.

1 Introduction

The search for interesting patterns in a database is today a well mastered task. Since the creation of the first algorithm APRIORI [1], techniques have refined themselves and we now use other tools such as the condensed representations and closed patterns [17, 2]. We do no more worry about the treatment of databases whose classic configuration includes a big number of lines (several millions) compared to the number of columns (a few thousand) The extracted patterns are constituted of the attributes corresponding to columns.

On the other hand, some specific data context coming from biology and genes sequencing still did not find any algorithm allowing to efficiently extract the knowledge. It is the case for databases including few lines and big quantity of columns (at the time of genes sequencing, lines are often the achieved experiences and columns genes expressions). In this situation, it is tempting to apply the data mining techniques to the transposed data matrix. The new database has now compatible measurements with a pattern mining on numerous lines (genes) and few columns (experiences). Unfortunately, the results are relative to patterns of experiences, what fascinates biologists little. The difficulty of the problem is reduced, but other problems emerge : interpretation of results, conversion of mining parameters, etc.

We intend in this article to study a new mining method that take fully advantage of geometric features of the database. We study the extraction of information from the transposed database, then use the Galois connection to infer results in the initial database. This connection defines concepts as a unique association between an attributes pattern and objects that contain them, or while transposing, between a group of objects and the attributes they share. When the format is favorable (few objects, a lot of attributes), the transposition of matrix allows to work on data whose exploration will be facilitated by a reversing tendencies lines / column.

We think that this method is new, even though it integrates classic tools of the learning community (patterns mining, Galois connection, concepts lattice). Its novelty resides in solutions that it provides to problems up to here inaccessible, and this by an clever combination of conventional methods in a particular data-mining context.

We start with describing the theoretical foundations of pattern research, which satisfy a given property in the database, and in particular the Galois connection. Then we define the transposition of a patterns mining and show how to exploit this connection to infer results on the original base. The last section is dedi-

	Attributes									
Obj.	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
o_1	1	1	1	1	0	1	1	0	0	0
o_2	1	1	1	1	0	0	0	0	1	1
o_3	1	1	1	1	0	0	0	0	1	1
o_4	0	0	0	0	1	1	1	1	1	1
o_5	1	0	1	0	1	1	1	1	0	0

TAB. 1 – Running example `re` of a database

cated to the survey of possible developments for this method : mining the totality of closed patterns, monotonous constraint and common border processing, and applications on biologic data.

2 Pattern mining

In this section are recalled the theoretical foundations that allow us to develop the method of pattern mining founded on the transposition. We describe the patterns lattice and the specialisation relation that link them, according to Mitchell's context [19]. Then we detail the contribution of anti-monotonous constraint in the search algorithms and specify borders of theory that allows to compare mining, in the framework described by Mannila and Toivonen [18]. We finish with the Galois connection, necessary to the understanding of our method.

2.1 Patterns lattice

Table 1 presents a database denoted `re` for five objects of survey (in lines) and indicates for each the absence or presence of 10 binary attributes (in columns). One can notice that the group of objects o_1, o_2, o_3 shares the same attributes a_1, a_2, a_3, a_4, and that o_4 and o_5 share a_5, a_6, a_7, a_8, etc. We will use more precisely the term of attributes pattern or objects pattern rather than of group.

Let $\mathcal{O}$ be a set of objects and $\mathcal{A}$ a set of attributes. On our example, $\mathcal{O} = \{o_1, o_2, \ldots o_5\}$ and $\mathcal{A} = \{a_1, a_2, \ldots a_{10}\}$. Data to explore are represented by the matrix of the binary relation $R \subset \mathcal{O} \times \mathcal{A}$ between every object and every attribute (cf. Table 1). Thus, $(o_i, a_j) \in R$ means that the object i has the attribute j. A database *bd*, also named a *context*, is a triplet $(\mathcal{O}, \mathcal{A}, R)$ associating two sets of objects and attributes with the help of a binary relation.

The set $\mathcal{A}$ of attributes allows to construct the language $\mathcal{L}_\mathcal{A} = 2^\mathcal{A}$ of patterns on $\mathcal{A}$. We then look among $\mathcal{L}_\mathcal{A}$ for patterns that verify in *bd* a property or predicate q. For example, we want to know the frequent attributes patterns (when the number of objects containing this pattern, called *support*, pass a certain threshold). On our running example, attributes patterns $\{a_1, a_2, a_3, a_4\}$ and $\{a_9, a_{10}\}$ are present at least three times, on the contrary $\{a_2, a_8\}$ is never present.

The set $\mathcal{L}_\mathcal{A}$ of attributes patterns is naturally represented by a lattice (cf. Figure 1). At the top, patterns of length 1 : $\{A_1\} \ldots \{a_{10}\}$. Next level, patterns of length 2 : $\{A_1, a_2\}$, $\{a_1, a_3\}$, $\{a_1, a_4\}$, $\ldots\{a_1, a_{10}\}$, $\{a_2, a_3\}$, $\ldots\{a_9, a_{10}\}$. Then patterns of length 3, etc. The before last level contains the ten patterns of length 9 and the last contains the unique pattern of length 10.

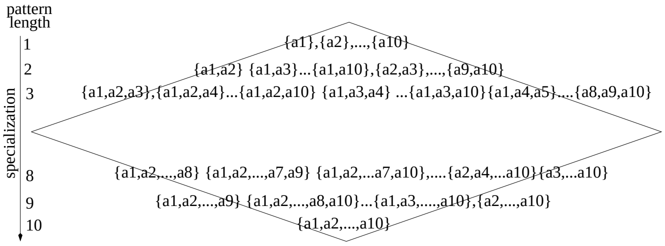
pattern length
specialization
1
2
3
8
9
10
{a1},{a2},...,{a10}
{a1,a2} {a1,a3}...{a1,a10},{a2,a3},...,{a9,a10}
{a1,a2,a3},{a1,a2,a4}...{a1,a2,a10} {a1,a3,a4} ...{a1,a3,a10}{a1,a4,a5}....{a8,a9,a10}
{a1,a2,...,a8} {a1,a2,...,a7,a9} {a1,a2,...a7,a10},.....{a2,a4,...a10}{a3,...a10}
{a1,a2,...,a9} {a1,a2,...,a8,a10}...{a1,a3,....,a10},{a2,...,a10}
{a1,a2,...,a10}

FIG. 1 – Pattern lattice

The lattice is oriented from the top (patterns of minimal length, the singletons) downwards to the bottom (the longest possible patterns) according to a *specialisation* relation (Mitchell's context [19]), that provides a method for their production, starting with the singletons. A pattern X of attributes of $\mathcal{A}$, is more specific than another pattern Y if $Y \subset X$. The lattice can then be generated methodically, level by level, shortest to longest patterns, with the help of a progressive specialisation by addition of attribute. For example, from $Y \cup \{a\}$ and $Y \cup \{b\}$, one produces by fusion $X = Y \cup \{a, b\}$. It only remains to browse the lattice, searching for interesting patterns.

2.2 Mining under anti-monotonous constraint

We try to calculate $Th(bd, \mathcal{L}_\mathcal{A}, q)$, the theory or set of patterns belonging to $\mathcal{L}_\mathcal{A}$ that satisfy a predicate q (we indifferently will write *constraint*). A very classic choice for q is : is the pattern *frequent* in the database *bd*? (does its support pass a fixed threshold γ?), is it *rare*? (the opposite), but one will also wonder if the pattern contains a certain fixed pattern, if it is free (ie. does not contain any association), closed, etc.

Some predicates have better properties than others in order to search in the lattice, and in particular those that respect the specialisation relation, or dual relation of generalisation. Generating the space to browse, it maintains a link between the patterns that guides the search for the predicate.

With regard to the support constraints, the specialisation of a rare pattern can only be rare. It is indeed a *monotone* constraint, that is preserved by the specialisation relation. Symmetrically, an *anti-monotonous* constraint (such as be frequent) is preserved by *generalisation* : q_{am} is anti-monotonous if $(q_{am}(X) \wedge Y \subset X) \Rightarrow q_{am}(Y)$. It means that when a pattern of the lattice verifies q_{am}, then all those that are higher in the lattice verify also q_{am}. On the contrary, when a pattern does not verify q_{am}, its lower specialisations neither.

The anti-monotonous constraints provide two pruning criterions :

Definition 1 (Criterion 1) *If a pattern X does not verify q_{am}, its specialisations can not verify q_{am} : it is useless to examine them and the lattice can be pruned under X.*

Definition 2 (Criterion 2) *If the candidate X to the verification of q_{am} contains a pattern that does not verify the predicate, then this candidate must be rejected : the lattice is pruned above X.*

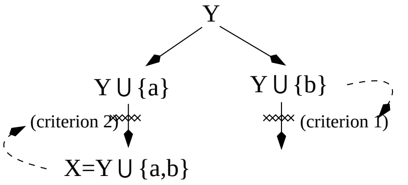

FIG. 2 – Pruning criterions

Figure 2, the two generators $Y \cup \{a\}$ and $Y \cup \{b\}$ are represented, as well as their fusion, $X = Y \cup \{a, b\}$. If one detects that $Y \cup \{b\}$ does not verify the constraint, then the branch coming from $Y \cup \{b\}$ is cut (criterion 1). Besides, X contains an incorrect pattern $Y \cup \{b\}$: it does not agree either (criterion 2).

The levelwise algorithm by [18], popularised in the particular case of the anti-monotonous constraint of frequency, is as follows : from the singletons (level 1), produce every pattern for a level with the help of patterns from the previous level, under condition that it verifies the two pruning criterions of the anti-monotonous constraint.

2.3 Galois connection

Up to here, we essentially spoke of attribute patterns. By the way of the relation R of the database, these patterns are connected to the objects patterns. On $bd = (\mathcal{O}, \mathcal{A}, R)$, one defines that for operators f and g of Galois connection between a pattern X of attributes from $\mathcal{A}$ and a pattern T of objects from $\mathcal{O}$:

Definition 3 (Galois connection) $f(T) = \{a \in \mathcal{A} \mid \forall o \in T, (o, a) \in R\}$ *and* $g(X) = \{o \in \mathcal{O} \mid \forall a \in X, (o, a) \in R\}$. *$f$ represents the set of all common attributes to a group of objects t (one says intention) and g the set of objects sharing the same attributes X (extension). The couple (f, g) defines the Galois connection [5] between $\mathcal{A}$ and $\mathcal{O}$, $h = f \circ g$ and $h' = g \circ f$ are the Galois closure operators.*

So, if we know how to represent the lattice of attributes patterns, we can represent the lattice of objects patterns. The patterns of the two lattices are connected by operators f and g. Concepts symbolise the link and associate two closed patterns :

Definition 4 (Closed patterns) *A pattern X of attributes is closed iff $h(X) = X$. A pattern T of objects is closed iff $h'(T) = T$. A concept (X, T) associates two closed patterns X of attributes and T of objects, such that $X = f(T)$ (or $T = g(X)$).*

In the context of the data mining, the closed patterns are well known, among others because their properties are multiple : they allow to efficiently calculate supports [2], to determine the minimal sets of association rules [20, 23, 16], encourage the clustering [10], etc. We use them here for the link that they define between objects and attributes. In facts, one notes that operators f and g preserve the property of closing (if X is closed, $g(X)$ as well, in the same way for T and $f(T)$). These invariance properties by closure operators provide a strong bi-directional link between a pattern of attributes and a pattern of objects : the Galois connection. Besides, this invariance assures that a closed pattern will be connected to another closed pattern. Knowing a closed attributes pattern, it is possible to pass to a closed objects pattern and vice versa. This link will be used Section 4, applied on the complete collection of the closed patterns of a type (attributes or objects) to infer closed patterns of the other type.

The theory relative to a predicate in a database can be restrict from attribute patterns to concepts :

Definition 5 (Concepts theory) *The concepts theory relatively to a data base bd, the associated language $\mathcal{L}$, and a constraint q, noted $Th_{\mathcal{C}}(bd, \mathcal{L}, q)$, is the set of concepts (X, T) such that X belongs to $Th(bd, \mathcal{L}_{\mathcal{A}}, q)$.*

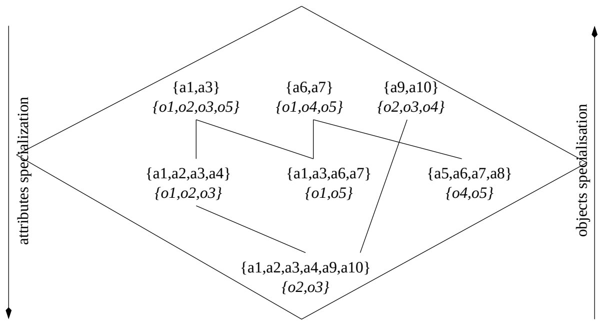

FIG. 3 – Galois lattice

We represented Figure 3 the same lattice that Figure 1, restricted to closed patterns. This lattice is ordinarily called concepts lattice [22]. Every element is a concept, labelled both by an attributes pattern and an objects pattern. One remark that the specialisation relation of attribute, that downwards orients the lattice, is now accompanied by the dual specialisation relation, on objects this time, but in inverse sense. Indeed, the connection reverses the inclusion direction : if $X \subset T$ then $g(X) \supseteq g(Y)$.

3 New mining method by way of transposition

We propose in this section a new mining method, that takes advantage of the Galois connection between the attributes patterns and objects patterns. We give definitions of the database and predicate transposition and finish by expressing the central result of complementarity between a pattern mining and its transposed.

3.1 Database transposition

With the Galois connection, the double lattice represents at the same time attributes patterns and objects patterns, with both different sense of specialisation (cf. Figure 3). It is therefore possible to achieve two minings :

- on attributes, with beginning at the top of the lattice and according to the specialisation relation of attributes,
- on objects, with beginning at the bottom of the lattice and according to the relation specialisation of objects.

If $bd = (\mathcal{O}, \mathcal{A}, R)$, the first mining computes $Th(bd, \mathcal{L}_\mathcal{A}, q)$ where $\mathcal{L}_\mathcal{A}$ is the language of attribute patterns. The second mining computes the theory relative to the *transposed* database of bd on the language of objects patterns :

Definition 6 (Transposed database) *Let $bd = (\mathcal{O}, \mathcal{A}, R)$ a database. The transposed database is ${}^t bd = (\mathcal{A}, \mathcal{O}, {}^t R)$ where $(a, o) \in {}^t R \iff (o, a) \in R$.*

3.2 Transposition of predicate

If the database transposition is relatively natural, it is not in the same way for the predicate that constraints the search. In the case of the frequency constraint, the dual notion for attributes patterns support is the length of the corresponding objects pattern. For example Figure 3, the pattern $\{a_9, a_{10}\}$ is present in objects o_2, o_3, o_4 : its support is 3, the length of $\{o_2, o_3, o_4\}$. We specify this notion by using the Galois connection to go from attributes formalism to objects formalism :

Definition 7 (Transposed predicate) *Let q be a predicate on the language $\mathcal{L}_\mathcal{A}$. The transposed predicate ${}^t q$ is defined on $\mathcal{L}_\mathcal{O}$ by (f is the Galois operator) :*

$$\forall T \in \mathcal{L}_\mathcal{O},\ {}^t q(T) \iff q(f(T))$$

For example, if q indicates that the attributes pattern has a support above the threshold γ, the transposed predicate ${}^t q$ will indicate that the objects pattern is longer than γ.

Relatively to the attributes specialisation, ${}^t q$ will be monotonous (resp. anti-monotonous) if q is monotonous (resp. anti-monotonous). The objects specialisation however follows the inverse sense of attributes ; so q is anti-monotonous w.r.t. attributes, ${}^t q$ is monotonous w.r.t. objects and it is necessary to take its negation, in order to recover an anti-monotonous predicate that can guide our search. We get the following property :

Property 1 *If q is anti-monotonous w.r.t. the attributes specialisation, then $\neg {}^t q$ is also anti-monotonous w.r.t. the objects specialisation.*

Proof : The Galois connection reverses the sense of inclusion : q, anti-monotonous w.r.t. attributes provides a transposed predicate ${}^t q$, that is monotonous w.r.t. objects : its negation is anti-monotonous.

To illustrate this property, let us come back to the frequency constraint on attributes patterns. One will for example have interest for patterns which are at least present three times. The transposed predicate is a constraint on the length of objects patterns, that requires therefore objects patterns to be longer than three. This new constraint is monotonous relatively to the objects specialisation (the specialisations of an objects pattern having more than three objects contains also more than three objects). The negation of the transposed constraint, that requires patterns to contain less than three objects, is therefore anti-monotonous.

3.3 Transposed Mining

We now have an operation of database transposition data and a new anti-monotonous predicate relatively to the specialisation of objects : we are able to apply the classic levelwise algorithm. Nevertheless, getting the new predicate requires to transpose the original predicate but especially to deny it in order to guarantee its anti-monotonicity. The new mining will then produce the complementary of the theory provided by the original mining.

Definition 8 (Transposed theory of concepts) *On ${}^t bd$ with the constraint $\neg {}^t q$, the levelwise algorithm levels will extract the theory of concepts $Th_\mathcal{C}({}^t bd, \mathcal{L}, \neg {}^t q)$, transposed of $Th_\mathcal{C}(bd, \mathcal{L}, q)$.*

The property immediately follows :

Property 2 (Complementarity of theories) *Relatively to the entirety of concepts, the theory of concepts $Th_\mathcal{C}(bd, \mathcal{L}, q)$ and its transposed $Th_\mathcal{C}({}^t bd, \mathcal{L}, \neg {}^t q)$ are complementary.*

Proof : On attributes or patterns, q and its transposed constraint ${}^tq$ are equivalent (their theories contain the same concepts). The theories relative to q and ${}^tq$ are therefore identical : relatively to q and $\neg{}^tq$ they are complementary.

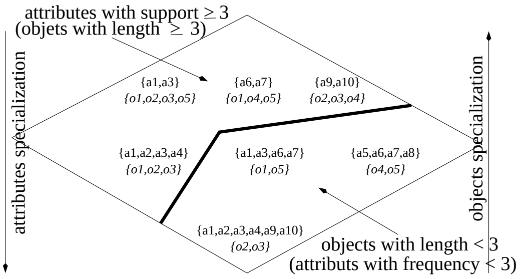

FIG. 4 – Complementarity of theories

On our running example, closed patterns of minimum support 3 are $\{a_1, a_3\}$, $\{a_1, a_2, a_3, a_4\}$, $\{a_6, a_7\}$ and $\{a_9, a_{10}\}$. With transposing and denying this constraint, one will search for objects patterns of maximal length 3 : one finds patterns $\{o_1, o_5\}$, $\{o_4, o_5\}$ and $\{o_2, o_3\}$. One verifies that the two sets of concepts are complementary (see Figure 4).

This property of complementarity justifies our new method of mining with working on the transposed base, because it is possible, while transposing, to get the complementary of the initial searched theory, and therefore to get easily the theory itself. One can also take advantage of the duality of borders, exposed at the end of the section 2, because the border of the theory of the original base coincides, thanks to complementarity, with the border of the transposed theory. The efficient [11] algorithms of direct mining of the border [19, 12, 4] can be used, and the negative border in the transposed base provides directly the positive border in the original base. The different aspects of this method are discussed in the section following.

4 Applications

In this section, we detail the use of the transposed mining, and present several examples of application. The first interesting operation consists in extracting entirely the closed objects patterns, to deduct for a null cost the closed attributes patterns. Interesting for situations where the normal mining fails due to the width of the database, this approach is validated on the biologic data. Then, we show the improvement of performances relatively to the pruning criterion 2, and we finish with a method of mining under monotonous constraint.

4.1 Entirely mining closed patterns

An algorithm calculating the set of closed patterns provides attributes patterns, and the Galois connection deducts corresponding closed objects patterns. Reciprocally, the transposed mining provides closed objects patterns, which the Galois connection translates to closed attributes patterns.

	`re`		^{t}`re`	
long.	success	failures	success	failures
1	32	13	9	1
2	24	22	4	4
Total	56	35	13	5
free	38		14	
closed	10			

TAB. 2 – Success/failures of the pruning criterion 2 on running example `re`

It is therefore possible to get the same set of concepts with extracting in the base or its transposed. The choice of one or the other method will be guided by measurements of the database : between the number of attributes or objects, one chooses the smallest, the one that will generate the less vast lattice.

On our running example, the smallest dimension is the one of objects : 5 patterns that generate $2^5 = 32$ patterns. Among these 32 patterns, only 10 are closed. But the original database contains 10 attributes or $2^{10} = 1024$ patterns, and it has always only 10 closed patterns. It is clear that it is more efficient to extract them with choosing the smallest dimension. These data are given in the summary table 2.

Different techniques exist to get the set of closed patterns. We use the *free* [7] or *key* patterns [3] to generate closed patterns (a free pattern is a pattern on which there is no association rule, cf. [6]). Besides, the freeness constraint is anti-monotonous, what is not the case of the closure constraint, and therefore provides a pruning criterion for the levelwise algorithm. But one closed pattern can be generated by several free patterns. On our example, the mining of free patterns provide 38 patterns that generate the 10 closed patterns, whereas the mining in transposed contents itself with 14 free patterns. The gain is immediate.

We also applied our method to the database `sain8`, coming from an experience achieved to the unit INRA/INSERM U449 [21]. It relates the transcriptome analysis of human muscular biopsies, before and after three hours of euglycemic-hyperinsulinemic clamp. The resulting expression matrix contains 6 lines and 1065 columns. In this disposition of the data, the mining provides in some minutes 667 831 free patterns but only 41 closed patterns. In the transposed database, the mining only takes some hundredth of seconds and satisfies itself with 42 free patterns, for the same 41 closed patterns.

This example is quite symptomatic of the efficiency of

	sain8		^{t}sain8	
long.	success	failures	success	failures
1	777	0	6	0
2	172 548	128 928	15	0
3	2 315 383	4 713 114	16	4
4	2 965 726	9 371 325	6	9
5	0	1 544 485	0	2
Total	5 454 434	15 757 852	43	15
free	667 831		42	
closed	41			

TAB. 3 – Success/failures of the pruning criterion 2 on sain8

the transposed mining. For the biologic contexts and their pathological measurements, this improvement is spectacular.

More generally, the use of the transposition to get the totality of closed patterns is justified in all contexts where the number of examples is weak compared to the number of describers. This situation forbids often the use of classic algorithms, historically conceived to treat databases including numerous lines. But in medicine and in biology, experiences are expensive and it is necessary to be content with a small number of situations. Let us notice that even though the mining of text handles more examples, it can be embarrassed by too numerous attributes. The transposed mining of concepts offers an opportunity to palliate a classic problem of measurements. Finally, as condensed representation of patterns, the collection of concepts allow multiple uses : calculation of supports, association rules, classification rules, clustering, etc.

4.2 Pruning criterion

Table 3 presents for sain8 and its transposed the number of patterns by level of the algorithm that satisfied the two pruning criterions and that should be examined after, and the number of failures to criterion 2. At every level, candidates are generated. Those that successfully pass the test will rejoin the free patterns after query of the base, others lead to a failure because of one not satisfying q sub-patterns.

The mining in ^{t}sain8 is exceptionally efficient : it produces less patterns to test in the base (37 against 5 453 657) and it leads to refuse infinitely less : 13 against 14 213 367.

In the biologic contexts, this argument is especially relevant. A mining on few lines but numerous columns will fail for lack of memory or time resource. Even though there are few lines, the number of failures to criteria 2 degrades performances and too numerous candidates are generated, for a very weak output.

All in all, the complexity result [18] specify that mining algorithms will examine at least as many patterns that there are in the theory and its border. This number of predicate tests is crucial, particularly in the huge volumes of data, where querying the database is slow. The lower bound on database request is therefore precious : it measures data access savings, directly consecutive to the pruning criterions. If the first criterion is easy to verify, the second can reveal to be the bottleneck of the algorithm [13], and this since the first levels.

4.3 Mining under monotonous constraints

The anti-monotonous constraints provide an important class of constraint that allows minings. But there are also very useful others, and of course monotonous constraints ! For example, looking for rare patterns in a database [9, 15] uses a monotonous constraint. The mining under any constraint is self a domain of research [14] and algorithms extracting simultaneously under the two types exist [8]. A method can consist in breaking out the constraint in conjunction and disjunction of monotonous, anti-monotonous constraints. We must therefore have methods for handling the basic two types : monotonous and anti-monotonous.

If q is monotonous, it forbids the classic algorithms, founded on anti-monotonous constraints, which prune the search space. But with the property of complementarity, $Th(bd, \mathcal{L}, q)$ and $Th({}^tbd, \mathcal{L}, \neg^t q)$ provide the theory for q and its complementary. According to the proprety 1, $\neg^t q$ is also monotonous, therefore $Th({}^tbd, \mathcal{L}, \neg^t q)$ is not calculable with levelwise algorithm either. On the other hand, ${}^tq$ is anti-monotonous for the specialisation relation of objects, and the following property holds :

Property 3 *$Th_{\mathcal{C}}(bd, \mathcal{L}, q)$, the concepts theory relative to a monotonous predicate q is :*

$$Th_{\mathcal{C}}({}^tbd, \mathcal{L}, {}^t q)$$

According to a monotonous constraint, the mining of patterns can therefore be led with simply using the levelwise algorithm, with the transposed constraint from attributes to objects, in the transposed database. For example, if one looks for the rare patterns, which support is lower than 3, it is well a monotonous constraint. Its transposition requires that objects patterns are shorter than 3. It is an anti-monotonous constraint, relatively to objects specialisation. The mining on the transposed database is feasible with the levelwise algorithm and provides the searched concepts.

5 Conclusion - Perspectives

We explained that the mining of theory relative to a constraint in a database is an arduous task in very large volumes of data (many attributes), a classic case in biologic datas. We proposed then a new method of mining, founded on the exploitation of the data base and the constraint transpositions. Then we have shown the utility of this technique to get more easily all closed patterns, and save the cost of the algorithm with minimising the space to to browse and failures of pruning criterias. At the end we have proposed a new process of mining under monotonous constraint.

Perspectives of application are numerous, precisely in the domains of genome biology. It remains however to complete the tools that can exploit this method. In particular, the depth-first mining algorithms that directly provide the common border can be adapted to take advantage of this property.
In short, it will be very useful to study constraint transposition. If the translation from support constraint to objects length is easy, it is not such natural for the freeness constraint (no association present in the pattern) for example.

Acknowledgements. Authors thank Sophie Rome and Hubert Vidal (INRA-INSERM U449) for supplying the data and Jeremy Besson (LIRIS CNRS FRE 2672) for pre-treatment. François Rioult is financed by the IRM unit of Caen CHU, the Manche Committee of the League against Cancer and the Regional Council of Bass-Normandy.

Références

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. inkeri Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, 1996.

[2] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. *DOOD'00*, pages 972–986, 2000.

[3] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Pascal : un algorithme d'extraction des motifs fréquents. *Technique et science informatiques. Vol. 21 - n° 1*, pages 65–95, 2002.

[4] R. Bayardo. Efficiently mining long patterns from databases. *ACM-SIGMOD*, pages 85–93, 1998.

[5] G. Birkhoff. Lattices theory. *American Mathematical Society, vol. 25*, 1967.

[6] J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. *PAKDD 2000*, 2000.

[7] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *PKDD'00*, pages 75–85, 2000.

[8] C. Bucila, J. Gehrke, D. Kifer, and W. White. Dualminer : A dual-pruning algorithm for itemsets with constraints. *SIGKDD'02*, 2002.

[9] G. Dong and J. Li. Efficient mining of emerging patterns : discovering trends and differences. *SIGKDD'99*, pages 43–52, 1999.

[10] N. Durand and B. Crémilleux. Ecclat : a new approach of clusters discovery in categorical data. *ES'02*, pages 177–190, December 2002.

[11] M.L. Fredman and L. Kachiyan. On the complexity of dualization of monotone disjunctive normal forms. In *J. Algorithms (21)*, pages 618–628, 1996.

[12] D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *PODS 1997*, pages 209–216, 1997.

[13] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD'00*, pages 1–12, 2000.

[14] B. Jeudy. Optimisation de requêtes inductives : application à l'extraction sous contrainte de règles d'association. *Ph.D. Thesis, INSA of Lyon*, 2002.

[15] S. Kramer L. De Raedt. The levelwise version space algorithm and its application to molecular fragment finding. *IJCAI'01*, pages 853–862, 2001.

[16] Viet Phan Luong. Reasoning on association rules. *BDA'01*, 2001.

[17] 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.

[18] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3) :241–258, 1997.

[19] T. Mitchell. Generalization as search. *Artificial Intelligence, vol. 18, n 2 p. 203-226*, 1980.

[20] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Closed set based discovery of small covers for association rules. In *BDA'99*, pages 361–381, 1999.

[21] S. Rome, K. Clément, R. Rabasa-Lhoret, E. Loizon, C. Poitou, G. S. Barsh, J.-P. Riou, M. Laville, and H. Vidal. Microarray profiling of human skeletal muscle reveals that insulin regulates 800 genes during an hyperinsulinemic clamp. *Journal of Biological Chemistry*, 2003.

[22] R. Wille. Concept lattices and conceptual knowledge systems. *Computer mathematic applied, 23(6-9) :493-515*, 1992.

[23] Mohammed J. Zaki. Generating non-redundant association rules. *SIGKDD'00, Boston*, pages 34–43, 2000.