CONCISE: An Algorithm for Mining Positive and Negative Non-Redundant Association Rules

Bemarisika Parfait

Laboratoire de Mathmatiques et Informatique de l'ENSET

Totohasina André

Laboratoire de Mathématiques et Informatique de l'ENSET, Université d'Antsiranana, Madagascar

Corresponding author: Bemarisika Parfait Email: bemarisikap7@yahoo.fr

One challenge problem in association rules mining is the huge size of the extracted rule set many of which are uninteresting and redundant. In this paper, we propose an efficient algorithm Concise for generating all non-redundant positive and negative association rules. We first introduce an algorithm CMG (Closed, Maximal and Generators) for mining all frequent closed, maximal and their generators itemsets from large transaction databases. We then define four new bases representing non-redundant association rules from these frequent itemsets. We prove that these bases significantly reduce the number of extracted rules. We show the efficiency of our algorithm by computational experiments compared with existing algorithms.

Keywords: Association rules, Interesting rules, Non-Redundant rules

1 Introduction and Motivations

Positive and negative association rules (PNAR) mining is one of fundamental problems in data mining. Let X and Y be two disjoints itemsets of database , an association rule $X \rightarrow Y$ states that a significant proportion in this database containing items in the premise (or antecedent) X also contain items in the consequent (or conclusion) Y. This rule can indicate the positive relations between

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 13–34. Computing & Intelligent Systems, SCRS, India.
https://doi.org/10.52458/978-93-91842-08-6-2

different items, is called a positive association rule (PAR) in . The association rules at other three forms $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$ and $\overline{X} \rightarrow \overline{Y}$, which can indicate the negative relations between items in database, are called negative association rules (NAR).

One big problem in association rules mining is the huge number of association rules generated many of which are uninteresting and redundant. An association rule is said to be uninteresting if its premise and its consequent are negatively dependent or statistically independent (even close to independence). It is said to be non-redundant association rules (or informative rules) if its premise (resp. consequent) is minimal (resp. maximal). Many approaches [1], [2], [3], [4], [5] based on traditional measure confidence [6], has been developed for reducing the size of the extracted rule set. However, no method to prune uninteresting association rules (UAR) has been found in the literature. Indeed, this classic measure confidence is not efficient to prune uninteresting rules. In addition, these approaches are insufficient, because they consider only the positive association rules, and this, with less selective pair support-confidence. Therefore, discovering NAR, which can be interest to several domains [7], [8, 9], [10], [11] such as Artificial Intelligence, Machine Learning, Data Mining, Big Data, Visualization, etc, is much more less developed than PAR due to the significant problem complexity caused by high computational cost and huge search space in calculating NAR candidates.

In this article, we propose a Concise algorithm to extract the set of non-redundant positive and negative association rules. Indeed, 1) we define CMG algorithm. 2) We propose a new model for sectioning significant rules. 3) We propose an efficient strategy to prune the UARs using M_{GK} [12]. 4) We propose a new method to prune the redundant rules.

The rest of this paper is organized as follows. Section 2 introduces preliminaries. The Concise algorithm is described in Section 3. Section 4 shows the results of computational experiments. Section 5 concludes this paper.

2 Basic Concepts

A database (cf. Table 1) is a triplet $= (\mathcal{T}, \mathcal{I}, \mathcal{R})$. and are finite sets of transactions and items respectively. $\subseteq \times$ is a binary relation between and . Let $X \subseteq$, $\overline{X} = \{t \in \mathcal{T} | \exists i \in X : (i,t) \notin\}$ is a complementary set of X (i.e. $\backslash X$). A subset $X \subseteq$ with $k = |X|$ is called k-itemset, where $|\ell|$ denotes the cardinality of ℓ. For example, A (resp. AB) is a 1-itemset (resp. 2-itemset). We write 2 (resp. 2) to denote the set of all subsets of (resp. the set of all subsets of). For $I \subseteq$ and $T \subseteq$, we define

Table 1: Database

TID	Items
1	ACD
2	BCE
3	ABCE
4	BE
5	ABCE
6	BCE

the two functions ϕ and ψ. Given $I \in 2$ and $T \in 2$: $2 \rightarrow 2, \phi(I) = I' = \{t \in |it, \forall i \in I\}$ and $2 \rightarrow 2, \psi(T) = T' = \{i \in |it, \forall t \in T\}$. The couple (ϕ, ψ) is a Galois connection [13] between the partial orders $(2, \subseteq)$ and $(2, \subseteq)$. $\gamma = \psi o \phi$ and $\tilde{\gamma} = \phi o \psi$ are Galois closure operators. Given $X \subseteq$, X is closed iff $X = \gamma(X)$; its support is given $supp(X) = P(X') = \frac{|\phi(X)|}{||}$ where P is a discrete probability. It is given as $supp(X) = supp(\gamma(X))$. If $X \subseteq Y$, then $\phi(X) \supseteq \phi(Y)$. Let $minsup \in]0, 1]$, X is frequent if $supp(X) \geqslant minsup$. We define the set of all frequent itemset as $= \{X \subseteq |supp(X) \geqslant minsup\}$. X and Y are said to be equivalent, denoted by $X \cong Y$, iff $\gamma(X) = \gamma(Y)$. The set of itemsets that are equivalent to X is defined as $[X] = \{Y \subseteq \mid X \cong Y\}$.

Let $minsup \in]0, 1]$, we define the set of all frequent closeds (Pasquier et al. [3,4]) as :

$$\mathcal{FC} = \{\in \mid = \gamma(),\ supp() \geqslant minsup\} \tag{2.1}$$

G is said to be generator of closed iff $\gamma(G) =$ and $\nexists g \subseteq$ with $g \subseteq G$ such that $\gamma(g) =$. Let $minsup \in]0, 1]$, the set of all frequent generators (Pasquier et al. [3,4]) is defined as :

$$= \{G \in [] | \in \mathcal{FC},\ \nexists g \subset G,\ supp(G) \geqslant minsup\} \tag{2.2}$$

Let $\mathcal{FC}$ be the set of all closed. We define $\mathcal{FM}$ the set of all maximal frequent [14], [15], [16] as:

$$\mathcal{FM} = \{\in \mathcal{FC} \mid \nexists\tilde{} \supset, \tilde{} \in \mathcal{FC}\} \tag{2.3}$$

The support, confidence (Agrawal and Srikant [6]) and M_{GK} [12] of an association rule $X \rightarrow Y$ are respectively defined as : $supp(X \cup Y) = \frac{|\phi(X \cup Y)|}{||}$,

$conf(X \to Y) = P(Y'|X')$ and

$$M_{GK}(X \to Y) = \begin{cases} \frac{P(Y'|X')-P(Y')}{1-P(Y')}, \text{if } P(Y'|X') > P(Y') \\ \frac{P(Y'|X')-P(Y')}{P(Y')}, \text{if } P(Y'|X') \leqslant P(Y') \end{cases} \qquad (2.4)$$

M_{GK} varies in $[-1, +1]$. It can be negative if X and Y are negatively dependent. Otherwise, $M_{GK}(X \to Y) > 0$ and quantifies the degree of positive dependence between these patterns. It equals -1 in repulsion limit between X and Y, passes to 0 at independence between X and Y ($P(Y'|X') = P(Y')$), and runs to 1 at logical implication between X and Y (i.e. $X' \subset Y'$).

3 Concise Algorithm

Concise algorithm will divided into two steps: (*i*) Mining all frequent closed, frequent maximal and their generators, and infrequent minimal itemsets, (*ii*) Generating all valid positive and negative rules from these frequent sets. Certain proofs of the Properties are omitted, for lack of space.

3.1 Mining all frequent itemsets

Our strategy for mining frequent itemsets will be synthesized in the main algorithm, called CMG (Algorithme 1).

This algorithm takes as input a database and a minimum support *minsup*. It returns the frequent closed, maximal and their generators using two sub-procedures (lines 17 and 18). CMG algorithm is a level-wise procedure for searching space. First, it finds all frequent itemsets using the EOMF algorithm [17] (line 2). It then verifies, for each k-frequent itemset ($k \geq 2$), if it is a closed by examining the supports of all its subsets of size $k-1$. Two boolean variables *key* and *closed* are then used to identify whether an itemset is a generator itemset or a closed. If $_k$ is empty and $_{k-1}$ is nonempty, the elements of $_{k-1}$ are closed and the *key* is a generator (lines 16 and 18). Conversely, if $_k$ is nonempty and $_{k-1}$ is empty, all itemsets in $_k$ are generators, and no extra step is needed as all itemsets are initially marked as generators.

An itemset q is identified as a generator during steps 8-15. If the support of q is the same as that of one of its subsets having length $k-1$ in $_{k-1}$, then q is not a generator, and conversely, it is not a closed. In steps 16 and 20, all the closed itemsets of length $k-1$ are added to the set $\mathcal{FC}_{k-1}$. Step 25 discovers the set of maximum length closures. In steps 17, 21 and 26, the GenMaximal procedure is

Algorithm 1 CMG

```
Require: Database , minimum support threshold $minsup \in ]0, 1]$.
Ensure: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Supp \rangle$ // Frequent closed, maximal and their generators/supports.
$\mathcal{FCMG} \leftarrow \emptyset$; $\mathcal{FC} \leftarrow \emptyset$; $\mathcal{FM} \leftarrow \emptyset$; $\mathcal{FG} \leftarrow \emptyset$; $\mathcal{FCMG}.Support \leftarrow 0$;
$\mathcal{F} \leftarrow EOMF(, minsupp)$ // $= \{_1, _2, \ldots, _\ell\}$, where $\ell$ is the size of largest frequent itemset.
for (each itemset $h \in \mathcal{F}_1$) do
    $h.key \leftarrow true$; $h.closed \leftarrow true$;
end for
for all ($k \leftarrow 2; k \leq \ell; k++$) do
    if ($_k \neq \emptyset$) then
        for (each itemset $h \in \mathcal{F}_k$) do
            $h.key \leftarrow true$; $h.closed \leftarrow true$;
            for all (subset $\tilde{h} \in \mathcal{F}_{k-1}$ of $h$) do
                if ($supp(\tilde{h}) == supp(h)$) then
                    $h.key \leftarrow false$; $\tilde{h}.closed \leftarrow false$;
                end if
            end for
        end for
        $\mathcal{FC}_{k-1} \leftarrow \{h \in_{k-1} | h.closed = true\}$;
        GenMaximal($\mathcal{FC}_{k-1}, _k$);
        GenGenerators($\mathcal{FC}_{k-1}, _k$);
    else
        $\mathcal{FC}_{k-1} \leftarrow \{h \in_{k-1} | h.closed = true\}$;
        GenMaximal($\mathcal{FC}_{k-1}$);
        GenGenerators($\mathcal{FC}_{k-1}$);
    end if
end for
$\mathcal{FC}_k \leftarrow_k$;
GenMaximal($\mathcal{FC}_k$);
GenGenerators($\mathcal{FC}_k$);
$\mathcal{FCMG} \leftarrow \bigcup_{j=1}^{k} \{\mathcal{FCMG}_j.Generator, \mathcal{FCMG}_j.Closed, \mathcal{FCMG}_j.Maximal, \mathcal{FCMG}_j.Support\}$;
```

called to generate the maximal itemsets. It takes the set $\mathcal{FC}_k$ as input. For each closed itemset h, it verifies if there is no other closed $\tilde{h}$ containing h such that $\tilde{h}$ is maximal itemset (line 2 of the GenMaximal procedure). If this is the case, the closed itemset h is maximal, then added also in the set of maximal $\mathcal{FM}$. At

Algorithm 2 Procedure GenMaximal($\mathcal{FC}_k$)

```
Require: $\mathcal{FC}_k$ // Frequent closed itemset of size $k$.
Ensure: Assign the maximal to each closed itemsets of $\mathcal{FC}_k$.
1: for (each itemset $h \in \mathcal{FC}_k$) do
2:     if ($\nexists \tilde{h} \supset h | \tilde{h}.maximal = true$) then
3:         $\mathcal{FM} \leftarrow \mathcal{FM} \cup \{h\}$;
4:     end if
5: end for
```

steps 18, 22 and 27, the GenGenerators procedure is called in order to update the global list of generators and to assign these generators to the respective sets of closed (or maximal) itemsets. It takes the set $\mathcal{FC}_k$ as input. For each closed c, its proper subsets in the global set of $\mathcal{FG}$ generators are then removed and added to the generators of c (steps 1-5 of the GenGenerators procedure). This procedure updates the global set of generators $\mathcal{FG}$ by the itemsets, which are not closed but are generators before the starting of the next iteration. If the set of generators of a given closed itemset is empty, it then indicates that the closed itemset is the generator of itself (so it is the only pattern in its equivalence class). Figure 1 illustrates the running of the CMG algorithm with the database of Table 1 and a minimum support $minsup = 2/6$.

Algorithm 3 Procedure GenGenerators($\mathcal{FC}_k$)

```
Require: $\mathcal{FC}_k$ // Frequent closed itemset of size $k$.
Ensure: Assign the generators to each closed itemsets of $\mathcal{FC}_k$.
1: for (each itemset $c \in \mathcal{FC}_k$) do
2:     for all (subset $\tilde{c} \in \mathcal{FG}$ of $c$) do
3:         add $\tilde{c}$ in $c$.generator;
4:     end for
5: end for
6: $\mathcal{FG} \leftarrow \mathcal{FG} \cup \{h \in_k |h.generator = true \wedge h.closed = false \wedge h.maximal = false\}$;
```

5*Scan →

$\mathcal{FCMG}_1$

Generator	Closed	Maximal	Supp
A	AC	-	3/6
B	BE	-	5/6
C	C	-	5/6
D	ACD	-	1/6
E	BE	-	5/6

5*Prune infreq. →

$\mathcal{FCMG}_1$

Generator	Closed	Maximal	Supp
A	AC	-	3/6
B	BE	-	5/6
C	C	-	5/6
E	BE	-	5/6

4*Scan →

$\mathcal{FCMG}_2$

Generator	Closed	Maximal	Supp
AB	ABCE	ABCE	2/6
AE	ABCE	ABCE	2/6
BC	BCE	-	4/6
CE	BCE	-	4/6

4*Prune infreq. →

$\mathcal{FCMG}_2$

Generator	Closed	Maximal	Supp
AB	ABCE	ABCE	2/6
AE	ABCE	ABCE	2/6
BC	BCE	-	4/6
CE	BCE	-	4/6

Figure 1: List of frequent closed, maximal and their generators with $minsup = 2/6$

3.2 Generating valid association rules

We first present our model for selecting a valid association rule in database . It is thus a question of observing some transactions in containing X and not Y without having the general tendency to have Y when X is present contested. The aim is to control the degree of implication of $X \rightarrow Y$. To do this, we adopt a model similar to the one proposed in [18] but with another measure M_{GK}. This consists to quantify the number of counterexamples $n_{X\wedge\overline{Y}}$ under the hypothesis H_0 of independence between X and Y. Let us note by mgk the value estimated by M_{GK}, and by Φ the distribution function of the standardized normal distribution $(0,1)$. Such a rule $X \rightarrow Y$ is statistically valid [19] at risk $\alpha \in]0,1]$ if and only if :

$$p_{XY} = mgk(X,Y) = 1 - \Phi(-mgk(X,\overline{Y})) \geq 1 - \alpha \tag{2.5}$$

We now present our strategy to eliminate uninteresting association rules (UARs). We recall that an association $X \rightarrow Y$ is said to be uninteresting rule if Y is independent on X (i.e. $P(Y'|X') = P(Y')$) `or` if Y is negatively dependent on X (i.e. $P(Y'|X') < P(Y')$). Notice that the classic support-confidence [6] can return the all UARs. For this, we use M_{GK}. Table 2 below illustrates not only the discriminating power of M_{GK} but also the limits of support-confidence. The information given in this Table 2 can be used to evaluate the association $A \rightarrow B$ and `tea` $\rightarrow$ `coffee`. For the (A,B), we have $supp(A \cup B) = 0.72$ and $conf(A \rightarrow B) = 0.9$. For the (tea,coffee), we have $supp(\text{tea}\cup\text{coffee}) = 0.2$ and $conf(\text{tea} \rightarrow \text{coffee}) = 0.8$. The support and the confidence are considered fairly high for both rules. However, $P(B'|A') = P(B') = 0.9 \Rightarrow M_{GK}(A \rightarrow B) = 0$ means that B is independent on A. This proves that $A \rightarrow B$ is not pertinent rule. We also get $conf(\text{tea} \rightarrow \text{coffee}) = 0.8 < 0.9 = supp(\text{coffee}) \Rightarrow M_{GK}(\text{tea} \rightarrow \text{coffee}) < 0$ implies that `tea` disfavors `coffee`, this proves that `tea` $\rightarrow$ `coffee` is an UAR. As a result that, by using the M_{GK} measure, the UARs are systematically eliminated.

Table 2: Contingency table for (A,B) and (tea, coffee)

	A	$\neg A$	$\sum$		coffee	¬coffee	$\sum$
B	72	18	90	tea	20	5	25
$\neg B$	8	2	10	¬tea	70	5	75
$\sum$	80	20	100	$\sum$	90	10	100

Very often, the response time of mining rules is not always better when the database is dense. For this, we adopt a pruning *reduce-rules-space* procedure [20, 21]. The challenge of this procedure is to reduce the number of rules without loss of information. Given $\mathscr{C} = \{X \rightarrow Y, Y \rightarrow X, \overline{X} \rightarrow \overline{Y}, \overline{Y} \rightarrow \overline{X}, X \rightarrow \overline{Y}, \overline{X} \rightarrow Y, \overline{Y} \rightarrow X, Y \rightarrow \overline{X}\}$ a set of global candidates, our method consists to divide $\mathscr{C}$ in possible subsets according to the dependence between X and Y. We then seek, for each subset, a member subset which can call on orginal members. Indeed, we demonstrated that $M_{GK}(X \rightarrow Y) = -M_{GK}(X \rightarrow \overline{Y}), \forall X, Y \subseteq$ [20]. As result, if $X \rightarrow Y$ is interesting, then $X \rightarrow \overline{Y}$ cannot be interesting. This divides $\mathscr{C}$ into 2 disjoint subsets $\mathscr{C}_1 = \{X \rightarrow Y, Y \rightarrow X, \overline{X} \rightarrow \overline{Y}, \overline{Y} \rightarrow \overline{X}\}$ and $\mathscr{C}_2 = \{X \rightarrow \overline{Y}, \overline{X} \rightarrow$

$Y, \overline{Y} \to X, Y \to \overline{X}\}$. In $\mathscr{C}_1$, we demonstrated that $M_{GK}(X \to Y) = M_{GK}(\overline{Y} \to \overline{X})$, $M_{GK}(Y \to X) = M_{GK}(\overline{X} \to \overline{Y})$, and $M_{GK}(X \to Y) \leq M_{GK}(Y \to X)$, $\forall X, Y \subseteq$ such that $\forall X \subseteq Y$. From these 3 relations, we can deduce that $Y \to X$, $\overline{X} \to \overline{Y}$ and $\overline{Y} \to \overline{X}$ can be derived from $X \to Y$. In $\mathscr{C}_2$, we established $M_{GK}(X \to \overline{Y}) = M_{GK}(Y \to \overline{X})$, $M_{GK}(\overline{X} \to Y) = M_{GK}(\overline{Y} \to X)$ and $M_{GK}(X \to \overline{Y}) \leq M_{GK}(\overline{X} \to Y)$, $\forall X, Y \subseteq$. From these 3 relations, we can deduce that $\overline{X} \to Y$, $\overline{Y} \to X$ and $Y \to \overline{X}$ are derivable from $X \to \overline{Y}$. This notes that two rules $X \to Y$ and $X \to \overline{Y}$ are sufficient to study the global set $\mathscr{C}$ of rules. This gives 100(8-2)/8=75% reductions of space complexities.

We now present our method for pruning redundant rules. A rule $r_1 : X_1 \to Y_1$ is said to be redundant if there exists a rule $r_2 : X_2 \to Y_2$, where $X_1 \supset X_2, Y_1 \subset Y_2$ such that $supp(r_1) = supp(r_2)$ and $M_{GK}(r_1) = M_{GK}(r_2)$. We then exploit the concept of minimal base of rules. Let R be a set of rules, and a basis of R. is said to be minimal if there is no subset $' \subset$ such that $'$ is a basis of R. Corresponding to popular approaches [22], [4], [5], we define 4 bases (Definitions 3.2, 3.2, 3.2) based on the two rules $X \to Y$ and $X \to \overline{Y}$ as retained in the previous paragraph. Then, we show that these bases are non-redundant (Theorems 3.2, 3.2, 3.2, 3.2).

[CBE^+ Basis] Let $\mathcal{FC}$ be the set of frequent closed itemsets. For each $\in \mathcal{FC}$, let be the set of minimal generators of , we have:

$$CBE^+ = \{G \to \backslash G \mid G \in, \in \mathcal{FC},\ G \neq\} \tag{2.6}$$

(i) All valid positive exact rules can be derived from to CBE^+ basis. (ii) All rules in CBE^+ are non-redudant exact rules.

(i) Let $r_1 : X_1 \to Y_1 \backslash X_1$ be the exact positive rule (i.e. $M_{GK}(r_1) = 1$) between two frequents X_1 and Y_1 such that $X_1 \subset Y_1$. Let be a frequent closed itemset (i.e. $\in \mathcal{FC}$). Since $M_{GK}(r_1) = 1$, we have $supp(X_1) = supp(Y_1)$. From $supp(X_1) = supp(Y_1)$, we derived that $supp(\gamma(X_1)) = supp(\gamma(Y_1)) \Rightarrow \gamma(X_1) = \gamma(Y_1) =$. Obviously, there exists a rule $r_2 : G \to \backslash G \in CBE^+$ such that G is a generator of for which $G \subseteq X_1$ and $G \subseteq Y_1$. We show that the rule r_1 and its supports can be derived from the rule r_2 and its supports. From $\gamma(X_1) = \gamma(Y_1) =$ and $\gamma(G) =$, we then have $supp(r_1) = supp(\gamma(X_1)) = supp(\gamma(Y_1)) = supp() = supp(r_2)$, and deduce that $M_{GK}(r_1) = M_{GK}(r_2)$. This explains that r_1 can be derived from r_2, and is a redundant rule of r_2, so it's pruned in CBE^+ base.

(ii) Let $r_2 : G \to \backslash G \in CBE^+$, we then have $G \in$ and $\in \mathcal{FC}$. We demonstrate that there is no other rule $r_3 : X_3 \to Y_3 \backslash X_3 \in CBE^+$ such as $supp(r_3) = supp(r_2)$, $M_{GK}(r_3) = M_{GK}(r_2)$, $X_3 \subseteq G$ and $\subseteq Y_3$. If $X_3 \subseteq G$, we then have $\gamma(X_3) \subseteq \gamma(G) =$. We deduce that $X_3 \notin \Rightarrow r_3 \notin CBE^+$. If $\subseteq Y_3$, we then have $= \gamma() = \gamma(G) \subset Y_3 =$

$\gamma(Y_3) \Rightarrow G \notin_{Y_3}$. In other words, r_2 is non-redundant. This proves that CBE^+ is a non-redundant base.

[CBA^+ Basis] Let $\mathcal{FC}$ be the set of frequent closed. For each $\in \mathcal{FC}$, let be the set of generators of . Given α, $0 < \alpha \leqslant 1$, we have:

$$CBA^+ = \{G \rightarrow \backslash G | (G,) \in_{\gamma(G)} \times \mathcal{FC}, \gamma(G) \subset, P('|G') > P('), p_G \geqslant 1 - \alpha\} \tag{2.7}$$

(i) All valid positive approximate rules can be derived from the rules of CBA^+. (ii) All association rules in the CBA^+ basis are non-redundant approximate rules.

(*i*) Let $r_1 : X_1 \rightarrow Y_1 \backslash X_1 \in CBA^+$ such that $X_1 \subset Y_1$. For any X_1 and Y_1, there is a generator G_1 such that $G_1 \subset X_1 \subseteq \gamma(X_1) = \gamma(G_1)$ and a generator G_2 such that $G_2 \subset Y_1 \subseteq \gamma(Y_1) = \gamma(G_2)$. Since $X_1 \subset Y_1$, we have $X_1 \subseteq \gamma(G_1) \subset Y_1 \subseteq \gamma(G_2)$ and the rule $r_2 : G_1 \rightarrow (\gamma(G_2)) \backslash G_1 \in CBA^+$. We show that r_1 can be derived from r_2. Since $G_1 \subset X_1 \subseteq \gamma(X_1) = \gamma(G_1)$ and $G_2 \subset Y_1 \subseteq \gamma(Y_1) = \gamma(G_2)$, we have $supp(G_1) = supp(X_1)$ and $supp(G_2) = supp(Y_1) = supp(\gamma(G_2))$. This gives that $supp(r_1) = supp(r_2)$ and $M_{GK}(r_1) = M_{GK}(r_2)$, in other words, r_1 can be derived from r_2 and therefore, r_1 is a redundant rule of r_2.

(*ii*) Let $r_2 : G \rightarrow \backslash G \in CBA^+$, we then have $\in \mathcal{FC}$ and $G \in$. We demonstrate that there is no other rule $r_3 : X_3 \rightarrow Y_3 \backslash X_3 \in CBA^+$ such as $supp(r_3) = supp(r_2)$, $M_{GK}(r_3) = M_{GK}(r_2)$, $X_3 \subseteq G$ and $\subseteq Y_3$. If $X_3 \subseteq G$, we then have $\gamma(X_3) \subset \gamma(G) \Longrightarrow X_3 \notin_C$. If $\subseteq Y_3$, we then have $= \gamma() \subset Y_3 = \gamma(Y_3)$. As result, $G \notin_{Y_3} \Rightarrow r_3 \notin CBA^+$, in other words, r_2 is a non-redundant rule. This proves that CBA^+ is a non-redundant base.

[CBE^- Basis] Let $\mathcal{FM}$ be the set of frequent maximal, and *minsup* $\in]0, 1]$. For each $\in \mathcal{FM}$, let be the set of frequent generators of , we have :

$$CBE^- = \{G \rightarrow \overline{y} \,|\, G \in, \in \mathcal{FM},\ y \notin \wedge supp(y) < minsup\} \tag{2.8}$$

(i) All valid negative exact rules can be derived from the rules of the CBE^- basis. (ii) All association rules in the CBE^- basis are non-redudant negative exact rules.

(*i*) Let $r_1 : X_1 \rightarrow \overline{Y}_1 \backslash X_1 \in CBE^-$ such that $X_1 \subset \overline{Y_1} \subseteq$ where $\in \mathcal{FM}$. Since $M_{GK}(r_1) = 1$, we have $X_1 \cong \overline{Y}_1 \Rightarrow supp(X_1) = supp(\overline{Y_1})$. Since $supp(X_1) = supp(\overline{Y_1})$, we have $supp(\gamma(X_1 \cup \overline{Y}_1)) = supp(\gamma(X_1)) = supp(\gamma(\overline{Y_1})) \Rightarrow \gamma(X_1 \cup \overline{Y}_1) = \gamma(X_1) = \gamma(\overline{Y_1}) = (a)$. Obviously, $\exists r_2 : G \rightarrow \overline{y} \backslash G \in CBE^-$ such that $G \in$ for which $G \subseteq X_1$ and $G \subseteq \overline{Y}_1$, and thus $G \subseteq \overline{y}$ (by Definition 3.2). We show that the rule r_1 can be derived from r_2. Since $r_2 : G \rightarrow \overline{y} \backslash G \in CBE^-$, we have $supp(G \cup \overline{y}) = supp(G)$. From $supp(G \cup \overline{y}) = supp(G)$, we have $supp(\gamma(G \cup \overline{y})) = supp(\gamma(G)) =$

$supp(\gamma(\overline{y})) \Rightarrow \gamma(G \cup \overline{y}) = \gamma(G) = \gamma(\overline{y}) =$ (a'). From relations (a) and (a'), we have $\gamma(G \cup \overline{y}) = \gamma(X_1 \cup \overline{Y_1}) \Leftrightarrow supp(r_1) = supp(r_2)$. Since $G \subseteq X_1 \subset \overline{Y_1} \subset \overline{y} \subseteq \gamma(G) =$, we have $supp(G) = supp(X_1) = supp(\overline{Y_1}) = supp(\overline{y}) = supp() \Rightarrow M_{GK}(r_1) = M_{GK}(r_2)$. These results explain that r_1 can be derived from r_2, and is a redundant rule w.r.t r_2.

(*ii*) Let $r_2 : G \rightarrow \overline{y} \backslash G \in CBE^-$, where $G \in$ and y is an infrequent 1-itemset (i.e. $y \notin$). We demonstrate that there is no other rule $r_3 : X_3 \rightarrow \overline{Y_3} \backslash X_3 \in CBE^-$ such as $supp(r_3) = supp(r_2)$, $M_{GK}(r_3) = M_{GK}(r_2)$, $X_3 \subseteq G$ and $\overline{y} \subseteq \overline{Y_3}$. If $X_3 \subseteq G$, we then have $\gamma(X_3) \subseteq \gamma(G) \subset \gamma(\overline{y}) =$. We deduce that $X_3 \notin$ and conclude that $r_3 \notin CBE^-$. If $\overline{y} \subseteq \overline{Y_3}$, we then have $\gamma(G) \subset \gamma(\overline{y}) \subseteq \gamma(\overline{Y_3}) =$. We deduce that $G \notin_{\overline{Y}_3}$ and conclude that $r_3 \notin CBE^-$. This implies that r_2 is a non-redundant rule, and proves that CBE^- is a non-redundant base.

[CBA^- Basis] For each $(, C) \in \mathcal{FC} \times \mathcal{FC}$, let (resp. $_C$) be the set of generators of closed (resp. C) such that $\subsetneq C$ and $P(C'|') < P(C')$. Given α, $0 < \alpha \leqslant 1$, we have :

$$CBA^- = \{G \rightarrow \overline{g} | (G, g) \in \times_C, \subsetneq C, P(\overline{g'}|G') > P(\overline{g'}), p_{G\overline{g}} \geqslant 1 - \alpha\} \tag{2.9}$$

(i) All valid negative approximate association rules can be derived from the rules of CBA^-. (ii) All association rules in the CBA^- are non-redudant negative approximate rules.

(*i*) Let $r_1 : X_1 \rightarrow \overline{Y_1} \backslash X_1 \in CBA^-$ with $X_1 \subset \overline{Y_1}$. For any frequent X_1 and Y_1, there is a generator G_1 such that $G_1 \subset X_1 \subseteq \gamma(X_1) = \gamma(G_1)$ and a generator G_2 such that $G_2 \subset Y_1 \subseteq \gamma(Y_1) = \gamma(G_2)$. Since $X_1 \subset \overline{Y_1}$, we have $X_1 \subseteq \gamma(G_1) \subset \overline{Y_1} \subset \overline{G_2} \subseteq \gamma(\overline{Y_1}) = \gamma(\overline{G_2})$. Obviously, $\exists r_2 : G_1 \rightarrow \overline{G_2} \backslash G_1 \in CBA^-$ such that $\gamma(G) \subsetneq \gamma(g)$ (by Definition 3.2). We show that r_1 can be derived from r_2. From $G_1 \subset X_1 \subseteq \gamma(G_1)$ and $G_2 \subset Y_1 \subseteq \gamma(G_2)$, we then have $G_1 \cong X_1$ and $\overline{Y_1} \cong \overline{G_2} \Rightarrow supp(X_1 \cup \overline{Y_1}) = supp(G_1 \cup \overline{G_2})$ and $M_{GK}(X_1 \rightarrow \overline{Y_1}) = M_{GK}(G_1 \rightarrow \overline{G_2})$. This explains that r_1 can be derived from r_2, and is a redundant rule.

(*ii*) Let $r_2 : G \rightarrow \overline{g} \backslash G \in CBA^-$, i.e. $G \in_C$ and $g \in$ such that $\gamma(G) \subsetneq \gamma(g)$ (i.e. $C \subsetneq$). We demonstrate that there is no other rule $r_3 : X_3 \rightarrow \overline{Y_3} \backslash X_3 \in CBA^-$ such that $supp(r_3) = supp(r_2)$, $M_{GK}(r_3) = M_{GK}(r_2)$, $X_3 \subset G$ and $\overline{Y_3} \supset \overline{g}$. If $X_3 \subset G$, we then have $\gamma(X_3) \subset \gamma(G) = C \Rightarrow X_3 \notin_C$. Since $X_3 \subset G$, we have $supp(X_3) > supp(G) \Rightarrow M_{GK}(r_3) < M_{GK}(r_2)$. If $\overline{g} \subset \overline{Y_3}$, we then have $supp(\overline{g}) > supp(\overline{Y_3}) \Rightarrow M_{GK}(r_2) > M_{GK}(r_3)$. This means that r_2 is a non-redundant rule, and proves that CBE^- is a non-redundant base.

The generation of these bases is done with a main procedure called CBNRR

(Concise base of non-redundant rules). This main CBNRR procedure (Algorithm 4)

Algorithm 4 CBNRR, Concise base of non-redundant rules

```
Require: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Support \rangle$.
Ensure: $\mathcal{CB}$, Concise base of non-redundant rules.
$\mathcal{CBE}^+(\mathcal{FCMG})$;
$\mathcal{CBA}^+(\mathcal{FCMG})$;
$\mathcal{CBE}^-(\mathcal{FCMG})$;
$\mathcal{CBA}^-(\mathcal{FCMG})$;
```

takes as input the set $\mathcal{FCMG}$, and returns the minimal set of rules (called base) by calling four secondary procedures $\mathcal{CBE}^+$ (Algo.5), $\mathcal{CBA}^+$ (Algo.6), $\mathcal{CBE}^-$ (Algo.7) and $\mathcal{CBA}^-$ (Algo.8). This choice of decomposition of the algorithms is motivated by the parallelization of these four procedures during the implementation to have simultaneously the four bases of the non-redundant association rules defined above.

The procedure $\mathcal{CBE}^+$ (Algorithm 5) takes as input a set $\mathcal{FCMG}$, and returns as output the exact positive association rule base. The $\mathcal{CBE}^+$ procedure is initialized to empty (line 1). Then, each element of $\mathcal{FCMG}$ is examined in order of increasing k (lines 2-8). For each k-generator $G \in \mathcal{FCMG}$, it verifies if G is not a unique element in its equivalence class (line 4). If this is true, then $G \rightarrow \backslash G$ is a valid exact rule, and added to the list $\mathcal{CBE}^+$ (line 5). Finally, the algorithm 5 returns the set $\mathcal{CBE}^+$ containing the list of exact positive rules between generators and their closures (line 9).

Algorithm 5 Procedure $\mathcal{CBE}^+$

```
Require: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Support \rangle$.
Ensure: $\mathcal{CBE}^+$, Concise base of exact positives rules.
1: $\mathcal{CBE}^+ = \emptyset$;
2: for ($k = 1$ to $\ell$, where $\ell$ is the size of largest frequent itemset in $\mathcal{FCMG}$) do
3:     for all ($k$-generator $G$ of $\mathcal{FCMG}_k.Generator$) do
4:         if ($G \neq \gamma(G)$) then
5:             $\mathcal{CBE}^+ \leftarrow \mathcal{CBE}^+ \cup \{G \rightarrow \gamma(G) \backslash G, G.supp\}$;    /* $\mathcal{CBE}^+$ Basis */
6:         end if
7:     end for
8: end for
```

The $\mathcal{CBA}^+$ procedure (Algorithm 6) takes as input a set $\mathcal{FCMG}$, and returns as output the exact positive association rule base. First, the procedure $\mathcal{CBA}^+$ is initialized to empty (line 1). It then examines the $\mathcal{FCMG}$ in order of increasing k (lines 2-10). For each k-generator $G \in \mathcal{FCMG}_k.Generator$, it considers a closed

Algorithm 6 Procedure $\mathcal{CBA}^+$

```
Require: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Support \rangle$, a real $\alpha \in ]0, 1]$.
Ensure: $\mathcal{CBA}^+$, Concise base of approximate positive rules.
 1: $\mathcal{CBA}^+ = \emptyset$;
 2: for ($k = 1$ to $\ell$, where $\ell$ is the size of largest frequent itemsets in $\mathcal{FCMG}$) do
 3:     for all ($k$-generator $G \in \mathcal{FCMG}_k.Generator$) do
 4:         for all (frequent closed itemset $\in \mathcal{FCMG}_{j>k}$ | $\supset \gamma(G)$) do
 5:             if ($mgk(G, ) \geq 1 - \alpha$) then
 6:                 $\mathcal{CBA}^+ \leftarrow \mathcal{CBA}^+ \cup \{r : G \rightarrow \backslash G, r.mgk, r.supp\}$;        /* $\mathcal{CBA}^+$ Basis */
 7:             end if
 8:         end for
 9:     end for
10: end for
11: return $\mathcal{CBA}^+$
```

containing the $\gamma(G)$ closure of G (lines 4-9). Then, it verifies if the pair $(G,)$ is valid (i.e. $mgk(G,) \geq 1 - \alpha, \ \forall\alpha \in]0, 1]$) (line 5). If this is true, then $G \rightarrow \backslash G$ is a valid approximate association rule, and is added to the list $\mathcal{CBA}^+$ (line 6).

The $\mathcal{CBE}^-$ procedure (Algorithm 7) takes as input a set $\mathcal{FCMG}$, and returns as output the exact negative rule base. It is initialized to empty (line 1). Then, it examines the set $\mathcal{FCMG}$ in ascending order of k (lines 2-8). For each k-generator $g \in \mathcal{FCMG}_k.Generator$ of a maximal itemset h, it verifies if there is any infrequent 1-itemset that is not part of the maximal itemset h. If so, then the rule $g \rightarrow \overline{z} \backslash g$ is an exact negative rule, and added to the list $\mathcal{CBE}^-$ (line 5).

Algorithm 7 Procedure $\mathcal{CBE}^-$

```
Require: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Support \rangle$, $minsup$.
Ensure: $\mathcal{CBE}^-$, A concise base of exact negative rules.
 1: $\mathcal{CBE}^- = \emptyset$;
 2: for ($k = 1$ to $\ell$, where $\ell$ is the size of largest frequent itemsets in $\mathcal{FCMG}$) do
 3:     for (each $k$-maximal $h \in \mathcal{FCMG}.Maximal$) do
 4:         for (each $k$-generator $g \in \mathcal{FCMG}.Generator$ of $h$) do
 5:             if ($\exists$ 1-itemset $z \notin h$ | $supp(z) < minsup$) then
 6:                 $\mathcal{CBE}^- \leftarrow \mathcal{CBE}^- \cup \{r : g \rightarrow \overline{z} \backslash g, r.supp\}$;        /* $\mathcal{CBE}^-$ Basis */
 7:             end if
 8:         end for
 9:     end for
10: end for
```

The $\mathcal{CBA}^-$ procedure (Algorithm 8) takes as input a set $\mathcal{FCMG}$, and a risk α such that $0 < \alpha \leq 1$. It returns as output the approximate negative association rule base. The procedure $\mathcal{CBA}^-$ is initialized to empty (line 1). It then examines

Algorithm 8 Procedure $\mathcal{CBA}^-$

```
Require: $\mathcal{FCMG} = \langle Closed, Maximal, Generator, Support\rangle$, and a real $\alpha \in ]0, 1]$.
Ensure: $\mathcal{CBA}^-$, Concise base of approximate negative rules.
 1: $\mathcal{CBA}^- = \emptyset$;
 2: for ($k = 1$ to $\ell$, where $\ell$ is the size of largest frequent itemset in $\mathcal{FCMG}$) do
 3:     for (each $k$-generator $G \in \mathcal{FCMG}_k.Generator$) do
 4:         for (each other $k$-generator $g \in \mathcal{FCMG}_{j\neq k}.Generator \mid \gamma(G) \subsetneq \gamma(g) \wedge P(\gamma(g)'|\gamma(G)') < P(\gamma(g)')$) do
 5:             if ($mgk(G, \overline{g}) \geqslant 1 - \alpha$) then
 6:                 $\mathcal{CBA}^- \leftarrow \mathcal{CBA}^- \cup \{G \to \overline{g}\backslash G\}$;        /* Base $\mathcal{CBA}^-$ */
 7:             end if
 8:         end for
 9:     end for
10: end for
```

the set $\mathcal{FCMG}$ according to the increasing order of k (lines 2-10). For each k-generator $G \in \mathcal{FCMG}_k.Generator$, and each other k-generator $g \in \mathcal{FCMG}_{j\neq k}.Generator$ such that $G.closure$ and $g.closure$ are incomparable and negatively dependent, the $\mathcal{CBA}^-$ procedure verifies if the pair $(G, \overline{g})$ is significant (line 5). If it is valid, then the rule $G \to \overline{g}\backslash G$ is an approximate negative rule, and added in the list $\mathcal{CBE}^-$ (line 6).

We now present simple algorithms for reconstructing all exact and approximate positive/negative rules. In order to develop these algorithms, we introduce a Proposition 3.2 for deriving all valid rules.

Soient $X_1 \to Y\backslash X_1$ et $X_2 \to Y\backslash X_2$ deux régles quelconques telles que $X_1 \subseteq X_2 \subseteq Y$, on a $M_{GK}(X_1 \to Y\backslash X_1) \leqslant M_{GK}(X_2 \to Y\backslash X_2)$, i.e. $mgk(X_1, Y) \leqslant mgk(X_2, Y)$ [20].

It follows that M_{GK} is antimonotonic according to the inclusion$_\subseteq$: the more attributes are passed from left to right, the more M_{GK} decreases. For example, for frequent 4-itemset $ABCD$, we have : $M_{GK}(ABC \to D) \geqslant M_{GK}(AB \to CD) \geqslant M_{GK}(A \to BCD) \Leftrightarrow mgk(ABC, D) \geqslant mgk(AB, CD) \geqslant mgk(A, BCD)$ means that if $A \to BCD$ is valid, then $ABC \to D$ and $AB \to CD$ are also valid. In other words, if r_1 is valid, then r_2 is also valid. So, r_2 can be derived by r_1. On the other hand, as we pointed out above, it is immediate that for all itemsets X and Y such that $X \subset Y$, we have $M_{GK}(X \to Y) < M_{GK}(Y \to X)$ [20]. Now, for all X and Y, we have $M_{GK}(Y \to X) = M_{GK}(\overline{X} \to \overline{Y})$ [20]. Therefore, $M_{GK}(X \to Y) < M_{GK}(\overline{X} \to \overline{Y})$, i.e. $mgk(X, Y) < mgk(\overline{X}, \overline{Y})$, $\forall X \subset Y$ means that if $X \to Y$ is valid, then $\overline{X} \to \overline{Y}$ is also valid. This notes that $\overline{X} \to \overline{Y}$ can be derived from $X \to Y$.

We firts present an algorithm 9 that derives all exact positive and negative rules. It takes as input $\mathcal{CBE}^+$, and returns as output the set $AllExact^{+-}$ of all

Algorithm 9 Deriving All Exact Positives and Negatives Rules

```
Require: $\mathcal{CBE}^+$.
Ensure: $AllExact^{+-}$, All exact positives and negatives rules.
 1: $AllExact^{+-} = \emptyset$;
 2: for all ($\{r_1 : X_1 \to Y_1, r_1.supp\} \in \mathcal{CBE}^+ | \, |Y_1| > 1$) do
 3:     for all (subset $y_1 \subset Y_1$) do
 4:         $AllExact^{+-} \leftarrow AllExact^{+-} \cup \{r_2 : X_1 \to y_1, r_3 : \overline{X_1} \to \overline{y}_1, r_1.supp\}$;
 5:         if ($\{r_4 : X_1 \cup y_1 \to Y_1 \backslash y_1, r_5 : \overline{X_1} \cup \overline{y}_1 \to \overline{Y_1} \backslash \overline{y}_1, r_1.supp\} \notin AllExact^{+-}$) then
 6:             $AllExact^{+-} \leftarrow AllExact^{+-} \cup \{r_4, r_5, r_1.supp\}$;
 7:         end if
 8:     end for
 9: end for
```

exact positive and negative rules. Indeed, for each $\{r_1 : X_1 \to Y_1\} \in \mathcal{CBE}^+$ with $|Y_1| > 1$ (lines 2-9) and each subset $y_1 \subset Y_1$ (lines 3-8), it generates all rules of the form $r_2 : X_1 \to y_1, r_3 : \overline{X}_1 \to \overline{y}_1, r_4 : X_1 \cup y_1 \to Y_1 \backslash y_1$ and $r_5 : \overline{X}_1 \cup \overline{y}_1 \to \overline{Y}_1 \backslash \overline{y}_1$ (lines 4 and 6). These rules have the same support as r_1 (cf. Theorem 3.2).

The algorithm 10 is built with the same optimizations properties as the Proposition 3.2. It takes as input the $\mathcal{CBA}^+$ database of approximate rules, and returns the $AllApprox^-$ set of all approximate positive rules of a database. It proceeds in

Algorithm 10 Deriving All Approximate Positives Rules

```
Require: $\mathcal{CBA}^+$.
Ensure: $AllApprox^+$, All approximate positive rules.
 1: $AllApprox^+ = \mathcal{CBA}^+$;
 2: for all ($\{r_1 : X_1 \to Y_1, r_1.supp\} \in \mathcal{CBA}^+ | \, |Y_1| > 1$) do
 3:     for all (subset $y_1 \subset Y_1$) do
 4:         if ($\{r_2 : X_1 \to y_1, r_2.supp, r_2.mgk\} \notin AllApprox^+$) then
 5:             $AllApprox^+ \leftarrow AllApprox^+ \cup \{r_2 : X_1 \to y_1, r_3 : \overline{X_1} \to \overline{y}_1, r_1.supp, r_1.mgk\}$;
 6:         end if
 7:     end for
 8: end for
 9: for all ($\{r_1 : X_1 \to Y_1, r_1.supp, r_1.mgk\} \in AllApprox^+$) do
10:     for all (subset $y_1 \subset Y_1$) do
11:         $AllApprox^+ \leftarrow AllApprox^+ \cup \{r_2 : X_1 \cup y_1 \to Y_1 \backslash y_1, r_3 : \overline{X_1} \cup \overline{y}_1 \to$
            $\overline{Y_1} \backslash \overline{y}_1, r_1.supp, r_1.mgk\}$;
12:     end for
13: end for
```

two phases. In the first phase (lines 2-10), it considers the approximate positive rules $X_1 \to Y_1$ with $|Y_1| > 1$ in the increasing order of their consequent size (lines

3-8). For each rule $X_1 \rightarrow Y_1$, all rules of the form $X_1 \rightarrow y_1$, with $y_1 \subset Y_1$, are generated if they have not been generated previously (line 5). These rules have the same support and *mgk* (cf. Theorem 3.2). In the second phase (lines 11-15), for each rule $X_1 \rightarrow Y_1$, it generates all rules of the form $X_1 \cup y_1 \rightarrow Y_1 \backslash y_1$ and $\overline{X}_1 \cup \overline{y}_1 \rightarrow \overline{Y}_1 \backslash \overline{y}_1$, $\forall y_1 \subset Y_1$. These rules have the same support as r_1.

The algorithm 11 takes as input the base $\mathcal{CBE}^-$, and returns as output the set $AllExact^-$ of all exact negative association rules. For each exact negative rule $X_1 \rightarrow \overline{Y_1}$ of $\mathcal{CBE}^-$ with $|\overline{Y_1}| > 1$ (lines 2-9) and each subset y_1 of Y_1 (lines 3-8), the algorithm 11 generates all exact negative rules of the form $X_1 \rightarrow \overline{y}_1$ and $\overline{X}_1 \rightarrow y_1$ (line 4). Thanks to the Propositions 3.2, any rules of the form $X_1 \cup y_1 \rightarrow \overline{Y_1} \backslash y_1$ and $\overline{X}_1 \cup \overline{y}_1 \rightarrow Y_1 \backslash \overline{y}_1$ with $y_1 \subset Y_1$ are generated (line 6) if they have not been generated previously (line 5). These rules have the same support as r_1 (cf. Theorem 3.2).

Algorithm 11 Deriving All Exact Negatives Rules

```
Require: $\mathcal{CBE}^-$.
Ensure: $AllExact^-$, All exact negative rules.
1: $AllExact^- = \emptyset$;
2: for all ($\{r_1 : X_1 \rightarrow \overline{Y}_1, r_1.supp\} \in \mathcal{CBE}^- \mid |\overline{Y}_1| > 1$) do
3:     for all (subset $\overline{y}_1 \subset \overline{Y}_1$) do
4:         $AllExact^- \leftarrow AllExact^- \cup \{r_2 : X_1 \rightarrow \overline{y}_1, r_3 : \overline{X_1} \rightarrow y_1, r_1.supp\}$;
5:         if ($\{r_4 : X_1 \cup y_1 \rightarrow \overline{Y}_1 \backslash y_1, r_5 : \overline{X}_1 \cup \overline{y}_1 \rightarrow Y_1 \backslash \overline{y}_1, r_1.supp\} \notin AllExact^-$) then
6:             $AllExact^- \leftarrow AllExact^- \cup \{r_4, r_5, r_1.supp\}$;
7:         end if
8:     end for
9: end for
```

The algorithm 12 takes as input the base $\mathcal{CBA}^-$, and returns the set $AllApprox^-$ of all approximate negative rules. It proceeds in two steps. In the first step (lines 2-8), it considers the rules $G \rightarrow \overline{g}$ with $|\overline{g}| > 1$ in the increasing order of their consequent (lines 3-9). For each $G \rightarrow \overline{g}$, all rules of the form $G \rightarrow \overline{g}_1$ and $\overline{G} \rightarrow g_1$, with $\overline{g}_1 \subset \overline{g}$, are generated if they have not been generated previously (line 5). These rules have the same support and *mgk*. In the 2nd step (lines 11-15), it considers all the rules $G \rightarrow \overline{g}$, and generates all rules of the form $G \cup g_1 \rightarrow \overline{g}_1 \backslash g_1$ and $\overline{G} \cup \overline{g}_1 \rightarrow \overline{g} \backslash \overline{g}_1$, $\forall \overline{g}_1 \subset \overline{g}$.

Let $X \rightarrow Y$ and $X \rightarrow \overline{Y}$ two principal rules generated, $m = ||$ the number of all attributs in database , and l the size of anatecedent rule. The worst case time complexity of principal procedure (Algorithm 4) is $(|\mathcal{FCMG}|^3(3^m + 2^{m+l} - 2^m - 2^l - 2m))$.

The lines 1-2 can produce different rules of the form $X \rightarrow Y$ which can cal-

Algorithm 12 Deriving All Approximate Negatives Rules

```
Require: $\mathcal{CBA}^-$.
Ensure: $AllApprox^-$, All approximate negative rules.
$AllApprox^- = \emptyset$;
for ($i = 1$ to $s - 1$, where $s$ is the size of largest frequent generator itemset) do
    for all ($\{r_1 : G \to \overline{g}, r_1.supp\} \in \mathcal{CBA}^- \mid |g| > i$) do
        for all (subset $\overline{g}_1 \subset \overline{g}$) do
            if ($\{r_2 : G \to \overline{g}_1, r_2.supp, r_2.mgk\} \notin AllApprox^-$) then
                $AllApprox^- \leftarrow AllApprox^- \cup \{r_2 : G \to \overline{g}_1, r_3 : \overline{G} \to g_1, r_1.supp, r_1.mgk\}$;
            end if
        end for
    end for
end for
for all ($\{r_1 : G \to \overline{g}, r_1.supp, r_1.mgk\} \in AllApprox^-$) do
    if ($\exists g_1 \in \gamma(g_1) \mid \gamma(g_1) \subsetneq \gamma(g) \wedge |g_1| < |g|$) then
        $AllApprox^- \leftarrow AllApprox^- \cup \{r_2 : G \cup g_1 \to \overline{g} \backslash g_1, r_3 : \overline{G} \cup \overline{g}_1 \to$
        $\overline{g} \backslash \overline{g}_1, r_1.supp, r_1.mgk\}$;
    end if
end for
```

culate as follows. An antecedent X, can be selected from in $\binom{m}{l}$ ways. Since $l = 1, \ldots, m-1$, the number of all possible ways to select an antecedent is $\sum_{l=0}^{m-1} \binom{m}{l} = 2^m - 2$. When the consequent contains a set of attributes, the idea of redundancy is also enlarged. Now rule $X \to Y \backslash X$ can be redundant with respect to $Q \to Z$ such that $Q \subseteq X$, $Z \supseteq Y \backslash X$ (i.e. $|Q| < |X|$, $|Y \backslash X| < |Z| < ||$). If $|Q| \leq |X|$, Q can be selected in $2^l - 1$ different ways, and if $|Q| < |X|$, it can be selected in $2^l - 2$ different ways. Similarly, if $|Z| \leq ||$, Z can be selected in $2^m - 1$ different ways, and if $|Z| < ||$, it can be selected in $2^m - 2$ different ways. Since Q and Z are separate, all combinations of Q and Z are possible. The number of all rules, such that $|Q| < |X|$ and $|Z| \leq ||$ is given $(2^l - 2)(2^m - 1) = 2^{m+l} - 2.2^m - 2^l + 2$. In addition, there are $2^m - 2$ rules, $|Q| = |X|$ and $|Z| < ||$. These make different rules :

$$2^{m+l} - 2.2^m - 2^l + 2 + 2^m - 2 = 2^{m+l} - 2^m - 2^l$$

Thus, the all lines 1-2 (Algorithm 9 lines 2-8 and Algorithm 10 lines 2-10) take at most

$$(|\mathcal{FCMG}|^3(2^{m+l} - 2^m - 2^l)) \tag{2.10}$$

The lines 3-4 can produce different rules of the form $X \to \overline{Y}$, and calculates as follows. Let $W = X \cup \overline{Y}$. If $|W| = i$, W can be selected from in $\binom{m}{i}$ different ways. For each W, there are:

$$\begin{aligned} m\text{-itemsets} &\Rightarrow \binom{m}{m}2^{m+1} \\ (m-1)\text{-itemsets} &\Rightarrow \binom{m}{m-1}2^{m} \\ &\vdots \\ (2)\text{-itemsets} &\Rightarrow \binom{m}{2}2^{2+1} \end{aligned}$$

In sum, we have

$$\begin{aligned} \sum_{i=2}^{m}\binom{m}{i}(2^{i+1}) &= 2\sum_{i=2}^{m}\binom{m}{i}2^{i} \\ &= 2\left[\sum_{i=0}^{m}\binom{m}{i}2^{i} - (1+2m)\right] \end{aligned}$$

From the binomial theorem $\sum_{i=0}^{m}\binom{m}{i}x^{i} = (1+x)^{m}$, we can derive :

$$\sum_{i=2}^{m}\binom{m}{i}2^{i+1} = 2(3^{m} - 2m - 1)$$

Thus, the all lines 3-4 (Algorithm 11 lines 2-10 and Algorithm 12 lines 2-10) take at most

$$(|\mathcal{FCMG}|^{3}(3^{m} - 2m)) \qquad (2.11)$$

Therefore, the total complexity ((2.10)+(2.11)), $\forall m, l$, is $(|\mathcal{FCMG}|^{3}(3^{m} + 2^{m+l} - 2^{m} - 2^{l} - 2m))$.

4 Experimental Evaluation

We evaluate Concise with two comparable baseline approaches: Pasquier's approach and Feno's approach. All algorithms are implemented in R, on a PC Core i3-2350M with 4CPUs. The Table 3 summarizes for each database, the number of

Table 3: Data characteristics

Database	‖	‖	⌈‖	ρ
T10I4D100K	100 000	1 000	10	1%
T20I6D100K	100 000	1 000	20	2%
C20D10K	10 000	386	20	5.18%
Mushrooms	8 416	119	23	19.33%

transactions ||, the number of items ||, the average size of transactions ⌈|, and the density ρ of the database which is given by ⌈|/||. The choice of databases is then motivated by the variety of their ||, || and ρ. Some databases like Mushrooms [1] and C20D10K (cf. footnote 1) are very dense (with $\rho = 19.33\%$ and $\rho = 5.18\%$ respectively), other databases like T10I4D100K[2] and T20I6D100K (cf. footnote 2) are sparse (with $\rho = 1\%$ and $\rho = 2\%$ respectively). For this, consider $\alpha = 5\%$ for Concise and Feno's approach, and minimal confidence $minconf = 80\%$ for Pasquier's approach. E^+ (resp. A^+) indicates a positive exact (resp. approximate) rules. E^- (resp. A^-) denotes a negative exact (resp. approximate) rules. We also denote by ”-” a subset which could not generated. Table 4 reports, for each algorithm, the number of extracted rules by varying the *minsup*. We observe that no negative rules are generated by Pasquier. For each algorithm, no E^+ and A^- are generated on T10I4D100K when $minsup \leq 30\%$. The reason is that all frequent are closed itemsets. On other databases, Feno's approach represents a smaller number w.r.t. Concise and Pasquier's approach. The explanation is that Feno is based on concept of pseudo-closed [14] which returns a reduced number of frequent itemsets and thus, it is the same for number of rules generated. However, Feno's approach is not informative. Whereas Concise and Pasquier's approach generate the more informative non-redundant association rules.

Table 4: Number of all valide positive and negative association rules

		Pasquier's approach				Feno's approach				Concise			
Dataset	*minsup*	$\lvert E^+\rvert$	$\lvert A^+\rvert$	$\lvert E^-\rvert$	$\lvert A^-\rvert$	$\lvert E^+\rvert$	$\lvert E^-\rvert$	$\lvert A^+\rvert$	$\lvert A^-\rvert$	$\lvert E^+\rvert$	$\lvert E^-\rvert$	$\lvert A^+\rvert$	$\lvert A^-\rvert$
3*T10I4D100K	10%	0	11625	-	-	0	0	10555	1256	0	0	725	52
	20%	0	8545	-	-	0	0	6656	1058	0	0	545	34
	30%	0	3555	-	-	0	0	2785	954	0	0	355	25
3*T20I6D100K	10%	115	71324	-	-	95	98	51899	3897	115	103	1804	56
	20%	76	57336	-	-	66	91	35560	2705	76	95	1403	38
	30%	58	45684	-	-	43	63	21784	1887	58	63	1175	27
3*C20D10K	10%	1125	33950	-	-	975	255	28588	11705	1125	285	1856	182
	20%	997	23821	-	-	657	135	19582	8789	997	185	1453	123
	30%	967	18899	-	-	567	98	11581	4800	967	101	1221	97
3*Mushrooms	10%	958	4465	-	-	758	289	3850	3887	958	304	1540	89
	20%	663	3354	-	-	554	178	2144	2845	663	198	1100	78
	30%	543	2961	-	-	444	109	1140	1987	543	115	998	39

[1] http://kdd.ics.uci.edu/

[2] http://www.almaden.ibm.com/cs/quest/syndata.html

On dense databases (C20D10K and Mushrooms), Concise algorithm is more selective than Pasquier's approach and Feno's approach for all *minsup*. Indeed, on C20D10K and $minsup = 1\%$, Pasquier's approach (resp. Feno) contains 33950 (resp. 28588) positive approximate rules as showed in Table 4, while the Concise contains 1856 positive approximate rules; this gives the reduction ratio 94.5% and 93.51% respectively. In this case, 32094 (resp. 26732) positive approximate rules can be deduced either from the Pasquier (resp. Feno) or from the Concise algorithm. The main reason is associated to the different techniques to prune both UARs and redundant association rules.

We present in the following the execution times of Concise compared to those existing. However, this comparison is still very difficult, for several reasons. First, Feno is not comparable to Concise, because it ignores the central step for mining frequent itemsets. Pasquier could not generate the negative rules. Then, we partialy compare Concise w.r.t. Pasquier. The results will be represented in Fig. 2 by varying the *minsup* at fixed $\alpha = 0.05$ and $minconf = 0.6$. On sparse databases (T10I4D100K and T20I6D100K), Concise algorithm and Pasquier's approach are almost identical for positive exact rules E^+ for all *minsup* (cf. Fig. 2a and 2b). On approximate rules A^+, it is very obvious that Concise algorithm is better than Pasquier (cf. Figure 2a, 2b). The explanation is that all frequent are closed itemsets, that complicates the task of Pasquier who performs more operations than Concise algorithm for counting frequent closed itemsets in database.

On dense databases (C20D10K and Mushrooms), Concise algorithm leads to significant average time compared to Pasquier for all *minsup* (cf. Figure 2c and Figure 2d). The main reason is associated to the technique for pruning search space of valid rules. Thanks to the different optimizations as developed, Concise algorithm can reduce considerable amount the execution time for all minimum support threshold *minsup*, it is not the case for Pasquier's approach. The latter obtains the least performance. This is mainly due to the lack of techniques for pruning the search space for valid association rules. This obviously affects its execution time. However, Pasquier's approach joins Concise algorithm for the E^+ execution times, when *minsup* is 20% to 30%.

5 Conclusion

In this work, we have presented and evaluated the Concise algorithm for mining non-redundant positive and negative association rules in a database. We have proved theoretically and confirmed by experience that this algorithm allows to remove a large number of redundant rules. Here are the possible perspectives

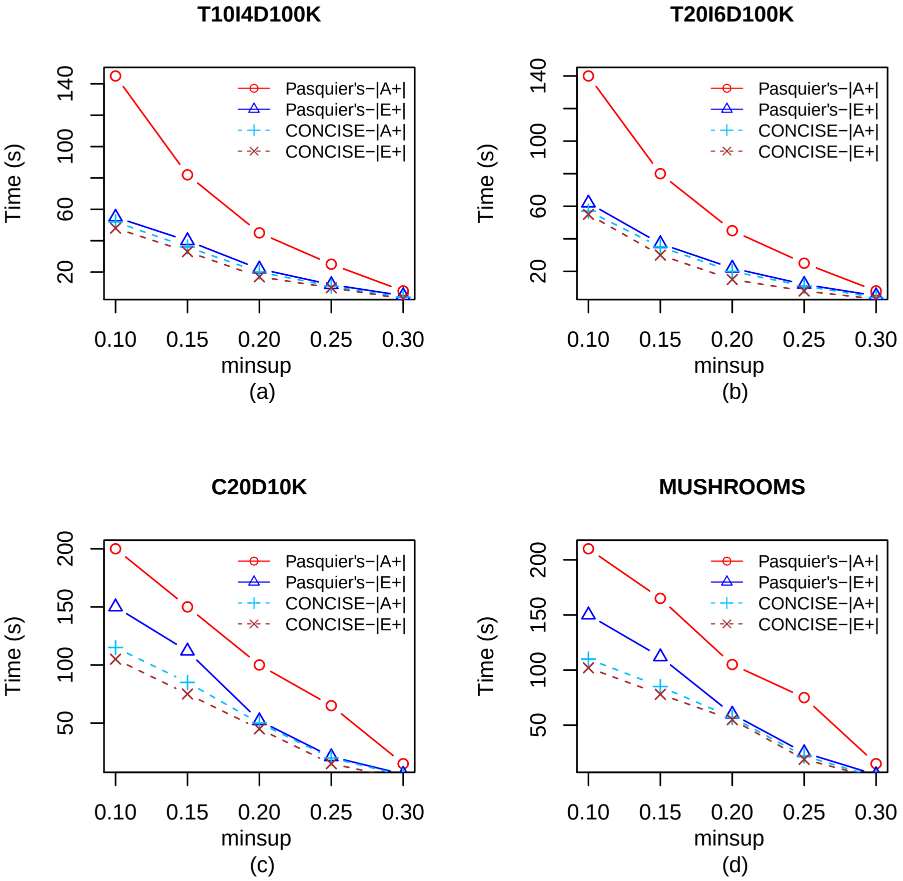

Figure 2: Response times by varying *minsup* at fixed $\alpha = 0.05$ and $minconf = 0.6$

related to this work: Elaboration of an implicative graph from these association rules collected; Construction of a hierarchical tree implicative this same set of association rules.

Bibliography

[1] Duong, H.V. & Truong, T.C. (2015). An efficient method for mining association rules based on minimum single constraints. Vietnam Journal Computer Sciences, pp. 67–83.

[2] Fournier-Viger, P. & Tseng, V.S. (2012). Mining Top-K Non-redundant Association Rules. *In L. Chen et al. (Eds.)*, pp. 31–40, Springer-Verlag.

[3] Pasquier, N., Taouil, R., Bastide, Y., Stumme, G. & Lakhal, L. (2005). Generating a condensed representation for association rules. *In J. of Intell. Info. Syst.*, pp. 29–60.

[4] Pasquier, N. (2009). Frequent Closed Itemsets Based Condensed Representations for Association Rules. *In Knowledge Extraction*, pp. 248–273.

[5] Xu, Y., Li, Y. & Shaw, G. (2011). Reliable representations for association rules. *In Data and Knowledge Engineering*, pp. 555–575.

[6] Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *In Proceedings of 20th VLDB Conference*, pp. 487–499.

[7] Cao, L., Dong, X. & Zheng, Z. (2016). E-NSP: Efficient negative sequential pattern mining. *In Artificial Intelligence*, pp. 156–182.

[8] Dong, X., Hao, H., Zhao, L. & Xu, T. (2018). An efficient method for pruning redundant negative and positive association rules. *In NEUCOM.*

[9] Dong, X., Yongshun, G. & Cao, L. (2018). F-NSP$^+$: A Fast Negative Sequential Patterns Mining Method with Self-adaption Data Storage Strategy. *In Pattern Recognit*, pp. 13–27.

[10] Rodriguez-Jimenez, J.M., Cordero,P., Encisoa, M. & Mora, A. (2014). Negative attributes and implications in Formal Concept Analysis. *In Procedia Computer Science*, pp. 758–765.

[11] Xu, T., Li, T. & Dong, X. (2018). Efficient High Utility Negative Sequential Patterns Mining in Smart Campusy. *In IEEE Access*, pp. 23839–23846.

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 13–34. Computing & Intelligent Systems, SCRS, India.
https://doi.org/10.52458/978-93-91842-08-6-2

[12] Totohasina, A. & Ralambondrainy, H. (2005). ION, A pertinent new measure for mining information from many types of data. *In SITIS.*

[13] Ganter, B. & Wille, R. (1999). Formal concept analysis: Mathematical foundations. *In Springer.*

[14] Mannila, M. & Toivonen, H. (1997). Levelwise Search and Borders of Theories in Knowledge Discovery. *In Data Mining Knowledge Discovery*, pp. 241–258.

[15] Durand, N. & Quafafou, M. (2013). Approximation de bordures de motifs frquents par le calcul de traverses minimales approchées d'hypergraphes. *Conférence Francophone sur l'Apprentissage Automatique (CAp 2013).*

[16] Liu, G., Li, J., Wong, L. & Hsu, W. (2014). Positive Borders or Negative Borders: How to Make Lossless Generator Based Representations Concise. *SIAM*, pp. 469–472.

[17] Bemarisika, P. & Totohasina, A. (2016). EOMF: Un algorithme d'extraction optimisée des motifs fréquents. *AAFD & SFC'2016*, pp. 198–203.

[18] Gras, R., Régnier, J-C., Marinica, C. & Guillet, F. (2013). L'analyse statistique implicative, Méthode exploratoire et confirmatoire é la recherche de causalités. *In Cépadués Ed.*, 11–40.

[19] Bemarisika, P. & Totohasina, A. (2021). Generating a Condensed Representation for Positive and Negative Association Rules. *Business Information Systems (BIS)*, pp. 175–186.

[20] Bemarisika, P. & Totohasina, A. (2019). An Informative Base of Positive and Negative Association Rules on Big Data. *IEEE Intnal Conf. on Big Data*, pp. 2428–2437.

[21] Bemarisika, P. & Totohasina, A. (2020). An Efficient Method for Mining Informative Association Rules in Knowledge Extraction. *A. Holzinger et al. (Eds.): CD-MAKE 2020*, pp. 227–247.

[22] Feno, D.R., Diatta, J. & Totohasina, A. (2006). Galois Lattices and Based for M_{GK}-valid Association Rules. *In Ben Yahia et al. (Eds.)*, pp. 186–197.