An Efficient Approach for Extraction Positive and Negative Association Rules from Big Data

Bemarisika Parfait, Ramanantsoa Harrimann, Totohasina André

An Efficient Approach for Extraction Positive and Negative Association Rules from Big Data

Bemarisika Parfait[1,2], Ramanantsoa Harrimann[1], and Totohasina André[1]

[1] Laboratoire de Mathématiques et d'Informatique, ENSET, Université d'Antsiranana, Madagascar
bemarisikap7@yahoo.fr, ramana_riri@yahoo.fr, andre.totohasina@gmail.com
[2] Laboratoire d'Informatique et de Mathématiques, EA2525, Université de La Réunion, France

Abstract. Mining association rules is an significant research area in Knowledge Extraction. Although the negative association rules have notable advantages, but they are less explored in comparaison with the positive association rules. In this paper, we propose a new approach allowing the mining of positive and negative rules. We define an efficient method of support counting, called *reduction-access-database*. Moreover, all the frequent itemsets can be obtained in a single scan over the whole database. As for the generating of interesting association rules, we introduce a new efficient technique, called *reduction-rules-space*. Therefore, only half of the candidate rules have to be studied. Some experiments will be conducted into such reference databases to complete our study

Keywords: Big Data · Extraction Association Rules · Reduction-access-database · Reduction-rules-space.

1 Introduction and Motivations

Since Agrawal's work [1], the extraction of association rules has been on of the most popular techniques for in Knowledge Extraction. An association rule is an implication of the form "if `Condition` then `Result`". Association rules may be used for store layout, target marketing, organize promotions of the supermarket, etc. In the literature, there exist two types of association rules: positive and negative rules. An association rule is said to be positive when it considers the presence of variables. It is negative when it considers the absence of these same variables. Although the negative rules have obvious advantages [6,10], they remain less explored in comparaison with positive rules. One of the major disadvantages lies in their difficult extraction, this type increases the exponential costs. Besides, the current approaches [10,11,15,16,18] are limited on the Apriori's data structure and support-confidence pair. While, this data structure imposes the repetitive accesses over the database, which can be costly. In addition, the support-confidence pair is questionable: (*i*) finding frequent itemsets is very complex in large databases and/or for low minimum support threshold; (*ii*) the number of rules that can be reduced nevertheless remains high that

many prove uninteresting. In order to exceed these notable limits, we propose an efficient approach for mining positive and negative association rules using a new pair, *support-M_{GK}*. We introduce a new economical technique of support counting, called *reduction-access-database*, based on the new data structure MatrixSupport and generator concepts. Therefore, a simple pass allows us to extract all the frequent itemsets over the whole database. As for association rules generating, we introduce an efficient method, called *reduction-rules-space*, partitioning the search space rules. Therefore, only half of the candidate rules are to study. Bazed on these optimizations, we also propose Erapn algorithm, less consumer in memory. We present the experimental evaluation conducted with databases from the literature by showing performances compared to semantically close approach such that RAPN algorithm [14] and Wu's algorithm [19].

The rest of this paper is organized as follows. Section 2 introduces the formal concepts. Section 3 details our approach. Section 4 summarizes our experimental results. Section 5 reviews the related work. A conclusion is given in Section 6.

2 Preliminaries concepts

This section describes association rules terminology (Subsection 2.1) and limits of the support-confidence pair (Subsection 2.2).

2.1 Association rules and Terminology

A transactional context is a triple $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$, where $\mathcal{T}, \mathcal{I}$ and $\mathcal{R}$ are finite and not empty sets. An element of $\mathcal{I}$ is called item (or attribute). A set of items, called an itemset. An element of $\mathcal{T}$ is called transaction (or object) represented by a `TID-Transaction IDentifier`, and $\mathcal{R}$ is a binary relationship between $\mathcal{T}$ and $\mathcal{I}$. So, $|\mathcal{T}|$ and $|\mathcal{I}|$ denotes the total number of transactions and items respectively. The table below represents an example. Given $X, Y \subseteq \mathcal{I}$,

Table 1: Example of the transactional context $\mathcal{B}$

TID	Items	Positive and negative items	Equivalent binary
1	ACD	A$\neg B$CD$\neg E$	10110
2	BCE	$\neg A$BC$\neg D$E	01101
3	ABCE	ABC$\neg D$E	11101
4	BE	$\neg A$B$\neg C\neg D$E	01001
5	ABCE	ABC$\neg D$E	11101
6	BCE	$\neg A$BC$\neg D$E	01101

$\neg X = \overline{X} = \mathcal{I} \backslash X = \{t \in \mathcal{T} | \exists i \in X : (i,t) \notin \mathcal{R}\}$ is called the logical negation of X. For example, with the table 1, we have $AB = \{3,5\}$, so $\overline{AB} = \{1,2,4,6\}$. A k-itemset is an itemset of length k. We will use the correspondances (Galois connections [12]) $g(\mathcal{I}) = \{t \in \mathcal{T} | \forall i \in \mathcal{I}, i\mathcal{R}t\}$ and $f(\mathcal{T}) = \{i \in \mathcal{I} | \forall t \in \mathcal{T}, i\mathcal{R}t\}$.

The function g is antimonotony: for all $X, Y \subseteq \mathcal{I}$, if $X \subseteq Y$ then $g(Y) \subseteq g(X)$. It is clear that if $X \subseteq Y$ then $supp(X) \geq supp(Y)$. The applications $\gamma = fog$ and $\gamma' = gof$ are Galois closure operators. An itemset X is closed if $X = \gamma(X)$.

A positive rule is an implication of the form $X \rightarrow Y$. It is called negative rule which consider the absence of the item, i.e., $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$ and $\overline{X} \rightarrow \overline{Y}$, where $X \cap Y = \emptyset$. X is called the premise and Y the conclusion. To determine an association rule interesting, two measures are used, support and confidence [1]. The support of X is the number of transactions that contain X, defined as $supp(X) = \frac{|\{t \in \mathcal{T} | X \subseteq t\}|}{|\mathcal{T}|} = \frac{|g(X)|}{|\mathcal{T}|}$. Denoting by P the intuitive probability measure defined on $(\mathcal{T}, \mathcal{P}(\mathcal{T}))$ by $P(Z) = \frac{|Z|}{|\mathcal{T}|}$ for $Z \subseteq \mathcal{T}$, the support of X can be written in terms of P as $supp(X) = P(X)$. The item X is said to be frequent if its support exceeds a minimum support threshold value, $minsup \in [0, 1]$, i.e. $supp(X) \geq minsup$. The support and confidence of $X \rightarrow Y$ are defined as $supp(X \cup Y) = \frac{|g(X \cup Y)|}{|\mathcal{T}|} = \frac{|g(X) \cap g(Y)|}{|\mathcal{T}|}$ and $conf(X \rightarrow Y) = P(Y|X) = \frac{supp(X \cup Y)}{supp(X)}$, respectively. Thereafter, we will omit the sign union and sometimes write XY instead of $X \cup Y$. According to Morgan, we obtain, for all $X, Y \subseteq \mathcal{I}$, $supp(\overline{X}) = 1 - supp(X)$, $supp(X\overline{Y}) = supp(X) - supp(XY)$, $supp(\overline{X}Y) = supp(Y) - supp(XY)$ and $supp(\overline{X}\ \overline{Y}) = 1 - supp(X) - supp(Y) + supp(XY)$.

2.2 Limit of support-confidence pair

Despite its notable contribution, this pair support-confidence easily selects uninteresting association rules (independence stochastic between two itemsets (For all $X, Y \subseteq \mathcal{I}, P(Y|X) = P(Y)$), or dependence negative ($P(Y|X) < P(Y)$)). The examples in table 2 illustrate this. In this case, the first four columns give the

Table 2: Limit of the couple support-confidence

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

characteristics of the purchase of products A and B, the last four indicate those of the purchase of coffee and tea. We obtain $supp(A \cup B) = 0.72$ and $P(B|A) = 0.9$. These reasonably high values lead us to believe that the persons buying A also buy B. However, we find that the confidence is equal to the probability of the conclusion regardless of the premise (*i.e.* $P(B|A) = P(B)$), it is a stochastic independence between A and B. The rule $A \rightarrow B$ that seemed interesting is therefore misleading. On the other hand, we obtain $supp(\text{tea} \cup \text{coffee}) = 0.2$, which assumes that tea favors coffee. However, the share of people buying coffee regardless of whether they also buy tea is higher, it is a negative dependence

between tea and coffee. The rule tea → coffee that seemed interesting is therefore misleading. That's why the support-confidence couple sometimes extracts uninteresting rules. The use of other more effective measures is imperative.

3 Mining Positive and Negative Association Rules

In this section, we introduce our approach for mining positive and negative association rules. It describes in a double problematic: finding frequent itemsets and generating potential valid association rules based on the previously extracted frequent itemsets. The first problem is often complex (in the worst case, it reaches $2^{|\mathcal{I}|}$) and dramatic when one considers the negative items. With the small database from Table 1, we have 1024 different items instead of 32 positive. The second problem is also complex (for an m-itemset, we have $5^m - 2(3^m) + 1$ instead of $3^m - 2^{m+1} + 1$). From Table 1, we have 2640 different rules instead of 180 classical rules. In these dimensions, it is necessary to select only a part. In [5,8], we have initiated the solution, which will be refined in Subsections 3.1 (*reduction-access-database* method) and 3.2 (*reduction-rules-space* method).

3.1 Mining frequent itemsets: Reduction-Access-Database

This is based on two steps: finding (in a single scans) frequent 1 and 2-itemsets, and frequent k-itemsets ($k \geq 3$). After that first step, frequent 2-itemsets are used to generate candidate 3-itemsets. The process continues until no more candidate can be generated. Given a *minsup*, finding the set of frequent itemsets $\mathcal{F}$, defined:

$$\mathcal{F} = \{X \subseteq \mathcal{I} | X \neq \emptyset \wedge supp(X) \geq minsup\}. \quad (1)$$

As noted, mining frequent itemets is very complex. The worst case concerns the small itemsets (1 and 2-itemsets). To answer this, we develop a new data structure MatrixSupport. The following Table 3 describes its formalism on the small database from Table 1. It is a projection of database $\mathcal{B}$ in relation to its attributes. The idea is to acquire data as the structure develops and store

Table 3: Formalism of the MatriceSupport in dataset $\mathcal{B}$

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

Scan the database $\mathcal{B}$

MatriceSupport					
$i \backslash j$	A	B	C	D	E
A	3	2	3	1	2
B	-	5	4	0	5
C	-	-	5	1	4
D	-	-	-	1	0
E	-	-	-	-	5

it. To each attribute corresponds a cell of the matrix to which we associate the absolute support, noted $|v_{ij}|$, expressing the number of times the item v_j

appears with the item υ_i, where i (resp. j) denotes the i-th line (resp. j-th column) of table. This is then used to identify the relative support, defined by:

$$supp(\upsilon_{ij}) = |\upsilon_{ij}|/|\mathcal{B}|. \quad (2)$$

For example, in Table 3, $supp(\upsilon_{11}) = supp(A) = |\upsilon_{11}|/6 = 3/6$, $supp(\upsilon_{12}) = supp(AB) = |\upsilon_{12}|/6 = 2/6$ and $supp(\upsilon_{23}) = supp(BC) = |\upsilon_{23}|/6 = 4/6$. For this technique, the supports of the small itemsets are retrievable in a simple pass over the whole dataset $\mathcal{B}$. As we mentioned, the generation of candidate k-itemsets is obtained from the frequent $(k-1)$-itemsets. To this end, the support of the candidate will be calculated as follows using the generator concept. An itemset X is a generator if it's minimal (of set inclusion) in its equivalence class. Its equivalence class is given by $[X] = \{X' \subseteq \mathcal{I} | \gamma(X') = \gamma(X)\}$. Note that the computational cost of closures is very exponential. However, the following lemma exploits the monotony of support upon set inclusion.

Lemma 1. *$\forall X, Y \in \mathcal{I}$, if $X \subseteq Y$ and $supp(X) = supp(Y)$, then $\gamma(X) = \gamma(Y)$.*

This Lemma 1 indicate that an itemset X is generator if it has no proper subset with the same support. For example, from the table 3, $supp(AB) = supp(ABC) = supp(ABE) = supp(ABCE) = 2/6$, we have $\gamma(AB) = \gamma(ABC) = \gamma(ABE) = \gamma(ABCE)$. Because, AB is minimal, then it is generator. If the candidate is a not generator, it will be calculated using the following proposition 1.

Proposition 1. *For all X non generator, $supp(X) = min\{supp(X') | X' \subset X\}$.*

Proof. Let $\mathcal{I}$ be a set items. Let X and X_1 be two itemsets on $\mathcal{I}$ such that $X_1 \subseteq X$. Due to the monotonicity of support, we have $supp(X) \leq supp(X_1)$. In addition (by assumption), X is not generator, it exists $X' \subseteq X$ on $\mathcal{I}$ such that $supp(X') = supp(X)$. However, $supp(X_1)$ is minimal in $\mathcal{I}$, so $supp(X_1) < supp(X')$. Finally, $supp(X) = supp(X_1) = min\{supp(X') | X' \subset X\}$. □

The support of a non generator k size is exactly the smallest support of its $(k-1)$-subsets. For example, from the table 3, we have $supp(AC) = supp(A) = 3/6$, therefore AC and its superset ABC are not generators itemsets. However, the superset of its subset ABC is then obtained by $supp(ABC) = min\{2/6, 3/6, 4/6\} = 2/6$. The following properties generalizes this observation.

Property 1. Given $X \subseteq \mathcal{I}$, if X is a generator, then $\forall Y \subseteq X$, Y is a generator, whereas if X is not a generator, $\forall Z \supseteq X$, Z is not a generator.

Theorem 1. *Any subset of a generator itemset must also be a generator. Any superset of a nongenerator itemset must also be nongenerator.*

Proof. Let X and Z be two itemsets on $\mathcal{I}$ satisfy $X \subseteq Z$. It exists an itemset $Y \subseteq \mathcal{I}$ $(Y \neq \emptyset)$, disjoint of X such that $Z = X \cup Y$. If X is assumed to be a non generator itemset, then it admits a proper subset T $(T \neq \emptyset)$ that is equivalent to it $T \subseteq X$ and $T \approx X$, giving $T \cup Y \approx X \cup Y$. By hypothesis, $X \cap Y = \emptyset$, so $T \cup Y \subseteq X \cup Y$, The itemset Z is equivalent to a proper subset $T \cup Y$, so it is not generator. The contrapose gives the result. □

This theorem is central in search space of frequent itemsets, no pass is done if a candidate is not generator. Only the generator are generated from database.

3.2 Generating Association Rules: Reduction-Rules-Space

The most common framework in the association rules generation is the support-confidence pair. As we already mentioned (see Subsection 2.2), this pair allow the pruning of many associations that are discovered in data, there are cases when many uninteresting may be produced. As such, we use the new pair support-M_{GK}. The next paragraph introduces the new measure, M_{GK} [13,17,19].

Given $X, Y \subseteq \mathcal{I}$, such that $X \cap Y = \emptyset$, M_{GK} of $X \to Y$ is defined by:

$$M_{GK}(X \to Y) = \begin{cases} \frac{P(Y|X)-P(Y)}{1-P(Y)}, \text{ if } X \text{ favors } Y,\ P(Y) \neq 1 \\ \frac{P(Y|X)-P(Y)}{P(Y)}, \text{ if } X \text{ disfavors } Y,\ P(Y) \neq 0. \end{cases} \quad (3)$$

In equation 3, X favors (resp. disfavors) Y indicate $P(Y|X) > P(Y)$ (resp. $P(Y|X) < P(Y)$). In our approach, an association rule $X \to Y$ is positive exact if $M_{GK}(X \to Y) = 1$, else, it is positive approximate rule. The following definition defines the interesting and uninteresting rules.

Definition 1. *Given $X, Y \subseteq \mathcal{I}$, an association rule $X \to Y$ is interesting if $P(Y|X) > P(Y)$, it's not interesting if $P(Y|X) \leq P(Y)$.*

The range of values for M_{GK} varies in $[-1, 1]$. Two zones are present: *attractive zone* and *repulsive zone*. The first is a zone that ranges from independence ($P(Y|X) = P(Y)$) to logical implication ($P(Y|X) = 1$). The second is a zone that ranges from incompatibility ($P(Y|X) < P(Y)$) to independence. If $M_{GK}(X \to Y) = 1$, then X and Y are strongly correlated, which denotes the logical implication between X and Y. Moreover, the rule $X \to Y$ is exact. Similarly, if $M_{GK}(X \to Y) = -1$, then X and Y are incompatible. This corresponds to the repulsion limit between X and Y. If $M_{GK}(X \to Y) = 0$, then X and Y are stochastically independant, moreover, the rule $X \to Y$ is not interesting. If $-1 \leq M_{GK}(X \to Y) < 0$, Y is negatively dependent on X. Similarly, if $0 < M_{GK}(X \to Y) \leq 1$, then Y is positively dependent on X.

Let $minsup \in [0, 1]$ and $minmgk \in [0, 1]$ be two minimum thresholds of support and M_{GK}, respectively. The rule $X \to Y$ is said to be valid according to our approach if its support $supp(X \cup Y)$ is frequent and $M_{GK}(X \to Y) \geq minmgk$. The set of all valid association rules from $\mathcal{B}$ is denoted $\mathcal{E}_{RAPN}$, formally:

$$\mathcal{E}_{RAPN} = \{X, Y \in \mathcal{I} | supp(X \cup Y) \geq minsup \ \& \ M_{GK}(X \to Y) \geq minmgk\}. \quad (4)$$

For the sake of comprehension, we apply this model on a same example in Table 1. The minimum support (resp. $minmgk$) is equal to 0.1 (resp. 0.8). Because, $M_{GK}(A \to B) = 0 < 0.8$, then A and B are stochastically independant, the association rule $A \to B$ is invalid. Moreover, it is not added in $\mathcal{E}_{RAPN}$. But, $supp(\overline{A} \cup B) = supp(B) - supp(A \cup B) = 0.90 - 0.72 = 0.18 > 0.1$ and $M_{GK}(\overline{A} \to B) = 0.88 > 0.8$, the rule $\overline{A} \to B$ is valid, it added in $\mathcal{E}_{RAPN}$. On the other hand, one has $M_{GK}(\text{tea} \to \text{coffee}) < 0$, coffee is negatively dependant on tea. This is a situation we should consider the negative association rules.

In the following paragraph, we present our strategies for elimination of uninteresting association rules from $\mathcal{B}$. We show that only half candidates are to study by using the new technique, *reduction-rules-space*. Indeed, we are interested in partitioning the search space as shown in the following proposition 2.

Proposition 2. *For all* $X, Y \in \mathcal{I}$, (1) X *fav* $Y \Leftrightarrow Y$ *fav* $X \Leftrightarrow \overline{X}$ *fav* $\overline{Y} \Leftrightarrow \overline{Y}$ *fav* $\overline{X}$. (2) X *disfav* $Y \Leftrightarrow X$ *fav* $\overline{Y} \Leftrightarrow \overline{Y}$ *fav* $X \Leftrightarrow Y$ *fav* $\overline{X} \Leftrightarrow \overline{X}$ *fav* Y.

Proof. Let X and Y be items of $\mathcal{I}$. We first prove, (a) X favors $Y \Leftrightarrow Y$ favors X, (b) X favors $Y \Leftrightarrow \overline{X}$ favors $\overline{Y}$ and (c) X favors $Y \Leftrightarrow \overline{Y}$ favors $\overline{X}$. In second time, (a) X disfavors $Y \Leftrightarrow X$ favors $\overline{Y}$, (b) X disfavors $Y \Leftrightarrow \overline{Y}$ favors X, (c) X disfavors $Y \Leftrightarrow Y$ favors $\overline{X}$ and (d) X disfavors $Y \Leftrightarrow \overline{X}$ favors Y.

$1(a)$ X favors $Y \Leftrightarrow P(Y|X) > P(Y) \Leftrightarrow \frac{supp(X \cup Y)}{supp(X)} > supp(Y) \Leftrightarrow \frac{supp(X \cup Y)}{supp(Y)} > supp(X) \Leftrightarrow P(X|Y) > P(X) \Leftrightarrow Y$ favors X. (b) X favors $Y \Leftrightarrow supp(X \cup Y) > supp(X)supp(Y) \Leftrightarrow 1 - supp(X) - supp(Y) + supp(X \vee Y) > 1 - supp(X) - supp(Y) + supp(X)supp(Y) \Leftrightarrow 1 - supp(X \wedge Y) > (1 - supp(X))(1 - supp(Y)) \Leftrightarrow supp(\overline{X \wedge Y}) > supp(\overline{X})supp(\overline{Y}) \Leftrightarrow \frac{supp(\overline{X} \vee \overline{Y})}{supp(\overline{X})} > supp(\overline{Y}) \Leftrightarrow P(\overline{Y}|\overline{X}) > P(\overline{Y}) \Leftrightarrow \overline{X}$ favors $\overline{Y}$. (c) X favors $Y \Leftrightarrow supp(\overline{X} \vee \overline{Y}) > supp(\overline{X})supp(\overline{Y})$ implies $\frac{supp(\overline{X} \vee \overline{Y})}{supp(\overline{Y})} > supp(\overline{X}) \Leftrightarrow P(\overline{X}|\overline{Y}) > P(\overline{X}) \Leftrightarrow \overline{Y}$ favors $\overline{X}$.

$2(a)$ X disfavors $Y \Leftrightarrow P(Y|X) < P(Y) \Leftrightarrow 1 - P(Y|X) > 1 - P(Y) \Leftrightarrow P(\overline{Y}|X) > P(\overline{Y}) \Leftrightarrow X$ favors $\overline{Y}$. (b) X disfavors $Y \Leftrightarrow P(\overline{Y}|X) > P(\overline{Y}) \Leftrightarrow \frac{supp(X \cup \overline{Y})}{supp(\overline{Y})} > supp(X) \Leftrightarrow P(X|\overline{Y}) > P(X) \Leftrightarrow \overline{Y}$ favors X. (c) X disfavors $Y \Leftrightarrow Y$ disfavors $X \Leftrightarrow P(X|Y) < P(X) \Leftrightarrow P(\overline{X}|Y) > P(\overline{X}) \Leftrightarrow Y$ favors $\overline{X}$. (d) X disfavors $Y \Leftrightarrow P(\overline{X}|Y) > P(\overline{X}) \Leftrightarrow P(Y|\overline{X}) > P(Y) \Leftrightarrow \overline{X}$ favors Y. □

This is if X favors Y $(P(Y|X) > P(Y))$, then only $X \rightarrow Y$, $Y \rightarrow X$, $\overline{X} \rightarrow \overline{Y}$ and $\overline{Y} \rightarrow \overline{X}$ are to be studied. If X disfavors Y $(P(Y|X) < P(Y))$, then only $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$, $\overline{Y} \rightarrow X$ and $Y \rightarrow \overline{X}$ are to be studied. That's why our method studies only half of the candidates. The following proposition describes the independence between a pair of two variables.

Proposition 3. *Given* X *and* Y *two itemsets of* $\mathcal{I}$, *if* (X, Y) *is a stochastically independent pair, so are pairs* $(\overline{X}, Y)$, $(X, \overline{Y})$, $(\overline{X}, \overline{Y})$.

Proof. $P(\overline{X})P(Y) - P(\overline{X} \wedge Y) = (1 - P(X))P(Y) - (P(Y) - P(X \cap Y)) = P(X \wedge Y) - P(X)P(Y)$. So, if $P(X \wedge Y) = P(X)P(Y)$, then $P(\overline{X})P(Y) = P(\overline{X} \wedge Y)$. Since X and Y play symmetric roles, we have the same result for $(X, \overline{Y})$, then replacing Y with $\overline{Y}$, for $(\overline{X}, \overline{Y})$. □

This proposition 3 is ideal, no association rule can be interesting if (X, Y) is stochastically independent. We continue our analysis by studying the candidate rules over the attractive class. To do this, we introduce the proposition 4 in order to pruning certain positive and negative association rules.

Proposition 4. *For all* X *and* Y *of* $\mathcal{I}$ *such that* X *favors* Y *and* $X \subseteq Y$, *we have* (1) $M_{GK}(X \rightarrow Y) \leq M_{GK}(Y \rightarrow X)$, (2) $M_{GK}(X \rightarrow Y) = M_{GK}(\overline{Y} \rightarrow \overline{X})$, (3) $M_{GK}(Y \rightarrow X) = M_{GK}(\overline{X} \rightarrow \overline{Y})$, (4) $M_{GK}(X \rightarrow Y) \leq M_{GK}(\overline{X} \rightarrow \overline{Y})$.

Proof. Let X and Y be items of $\mathcal{I}$. (1) According to proposition 2(1), X favors $Y \Leftrightarrow Y$ favors X, it gives $M_{GK}(Y \to X) = \frac{P(X|Y)-P(X)}{1-P(X)} = \frac{P(X)[P(Y|X)-P(Y)]}{P(\overline{X})P(Y)} = \frac{P(X)P(\overline{Y})}{P(\overline{X})P(Y)} \frac{P(Y|X)-P(Y)}{1-P(Y)} = \frac{P(X)}{P(\overline{X})} \frac{P(\overline{Y})}{P(Y)} M_{GK}(X \to Y)$. Because X favors Y and $X \subseteq Y$, we have $P(X) \geq P(Y) \Leftrightarrow P(\overline{X}) \leq P(\overline{Y}) \Leftrightarrow P(X)P(\overline{Y}) \geq P(\overline{X})P(Y)$, where $M_{GK}(X \to Y) \leq M_{GK}(Y \to X)$. (2) $M_{GK}(X \to Y) = \frac{P(Y|X)-P(Y)}{1-P(Y)} = \frac{-P(X|\overline{Y})+P(X)}{P(X)} = \frac{P(\overline{X}|\overline{Y})-P(\overline{X})}{1-P(\overline{X})} = M_{GK}(\overline{Y} \to \overline{X})$. From this property, M_{GK} is implicative. So, the property (3) is immediate, it derives from this implicative character of the M_{GK}. The property remains to be shown (4). Indeed, according to proposition 2(2), we have $M_{GK}(X \to Y) \leq M_{GK}(Y \to X) = M_{GK}(\overline{X} \to \overline{Y})$, which gives us $M_{GK}(X \to Y) \leq M_{GK}(\overline{X} \to \overline{Y})$. □

In this proposition 4, the properties (1), (2), (3) and (4) guarantee that if $X \to Y$ is valid, then $Y \to X$, $\overline{X} \to \overline{Y}$ and $\overline{Y} \to \overline{X}$ will also be the same because M_{GK} of the rule $X \to Y$ is less than or equal to those of $Y \to X$, $\overline{X} \to \overline{Y}$ and $\overline{Y} \to \overline{X}$. The set of valid rules of the class is thus derived from the only rule $X \to Y$. This will significantly limit the research space. The following proposition 5 is introduced to loosen certain rules of the repulsive class.

Proposition 5. *For all X and Y of $\mathcal{I}$, such that X disfavors Y and $X \subseteq Y$, we have* (1) $M_{GK}(X \to \overline{Y}) = M_{GK}(Y \to \overline{X})$, (2) $M_{GK}(\overline{X} \to Y) = M_{GK}(\overline{Y} \to X)$, *and* (3) $M_{GK}(X \to \overline{Y}) \leq M_{GK}(\overline{X} \to Y)$.

Proof. Let X and Y of $\mathcal{I}$. According to proposition 2(2), we have X disfavors $Y \Leftrightarrow X$ favors $\overline{Y} \Leftrightarrow \overline{Y}$ favors $X \Leftrightarrow Y$ favors $\overline{X} \Leftrightarrow \overline{X}$ favors Y. Thus, due to the implicitive character of M_{GK}, the properties (1) and (2) are then immediate. It remains to show (3). As X favors $\overline{Y}$, we get $M_{GK}(X \to \overline{Y}) = \frac{P(\overline{Y}|X)-P(\overline{Y})}{1-P(\overline{Y})} = \frac{P(\overline{X})P(\overline{Y})}{P(X)P(Y)} \frac{P(Y|\overline{X})-P(Y)}{1-P(Y)} = \frac{P(\overline{X}P(\overline{Y})}{P(X)P(Y)} M_{GK}(\overline{X} \to Y)$. By hypothesis, X disfavors Y and $X \subseteq Y$, we have $P(X) \geq P(Y) \Leftrightarrow P(\overline{X}) \leq P(\overline{Y})$ implie $P(\overline{X})P(\overline{Y}) \leq P(X)P(Y)$, finally $M_{GK}(X \to \overline{Y}) \leq M_{GK}(\overline{X} \to Y)$. □

The properties (1), (2) and (3) of this proposition 5 indicate that if $X \to \overline{Y}$ is valid, then $\overline{X} \to Y$, $Y \to \overline{X}$ and $\overline{Y} \to X$ will be valid, because this M_{GK} is less than or equal to those of $\overline{X} \to Y$, $Y \to \overline{X}$ and $\overline{Y} \to X$. Only $X \to \overline{Y}$ will make it possible to deduce the interest of the class.

Proposition 6. *For all X and Y of $\mathcal{I}$, $M_{GK}(X \to \overline{Y}) = -M_{GK}(X \to Y)$.*

Proof. We show in two cases: (1) X favors Y and (2) X disvaors Y. (1) X favors $Y \Leftrightarrow X$ favors $\overline{\overline{Y}} \Leftrightarrow X$ disfavors $\overline{Y}$ implies $M_{GK}(X \to \overline{Y}) = \frac{P(\overline{Y}|X)-P(\overline{Y}))}{P(\overline{Y})} = -\frac{P(Y|X)-P(Y)}{1-P(Y)} = -M_{GK}(X \to Y)$. (2) X disfavors $Y \Leftrightarrow X$ favors $\overline{Y}$, this gives $M_{GK}(X \to \overline{Y}) = \frac{P(\overline{Y}|X)-P(\overline{Y}))}{1-P(\overline{Y})} = -\frac{P(Y|X)-P(Y)}{P(Y)} = -M_{GK}(X \to Y)$. □

The next result of the following proposition makes it possible to characterize the exact negative association rules according to support-M_{GK} pair.

Proposition 7. *Let X, Y and Z be three itemsets disjoint 2 to 2, if $XZ \rightarrow Y$ (resp. $XZ \rightarrow \overline{Y}$) is an exact rule, so is rule $X \rightarrow Y$ (resp. $X \rightarrow \overline{Y}$).*

Proof. $M_{GK}(XZ \rightarrow Y) = 1 \Leftrightarrow \frac{P(Y|XZ)-P(Y)}{1-P(Y)} = 1 \Leftrightarrow \frac{supp((X\cup Z)\cup Y)}{supp(X\cup Z)} = 1$. Since X, Y and Z are disjoint 2 to 2, $\frac{supp(X\cup Y)}{supp(X)} = 1 \Leftrightarrow \frac{P(Y|X)-P(Y)}{1-P(Y)} = 1 \Leftrightarrow M_{GK}(X \rightarrow Y) = 1$. Replacing Y with $\overline{Y}$ for $XZ \rightarrow \overline{Y}$ and $X \rightarrow \overline{Y}$. □

The corollary 1 is the consequence of the proposition 7.

Corollary 1. *Let X and Y be two itemsets on $\mathcal{I}$, for all $Z \subseteq \mathcal{I}$ such that $Z \subset X$, if $M_{GK}(X \rightarrow \overline{Y}) = 1$, then $M_{GK}(Z \rightarrow \overline{Y}) = 1$.*

Proof. For all Z, such that $Z \subset X$, we have $supp(X) > 0$. Therefore, by proposition 7, we have $M_{GK}(Z \rightarrow \overline{Y}) = 1$. □

Proposition 8. *Let X, Y, T and Z be four itemsets of $\mathcal{I}$, such that X favors Y and Z favors T, and $X \cap Y = Z \cap T = \emptyset$, and $X \subset Z \subseteq \gamma(X)$, and $Y \subset T \subseteq \gamma(Y)$. Then, $supp(X \cup Y) = supp(Z \cup T)$ and $M_{GK}(X \rightarrow Y) = M_{GK}(Z \rightarrow T)$.*

Proof. $\forall X, Y, T, Z \subseteq \mathcal{I}$, $supp(X\cup Y) = \frac{|g(X\cup Y)|}{|\mathcal{T}|} = \frac{|g(X)\cap g(Y)|}{|\mathcal{T}|}$ and $supp(Z\cup T) = \frac{|g(X\cup T)|}{|\mathcal{T}|} = \frac{|g(Z)\cap g(T)|}{|\mathcal{T}|}$. Because $X \subset Z \subseteq \gamma(X)$ and $Y \subset T \subseteq \gamma(Y)$, we have $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$. It causes $g(X) = g(Z)$ and $g(Y) = g(T)$ implies $supp(X \cup Y) = supp(Z \cup T)$. As $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$, we have $P(Y|X) = P(T|Z) \Leftrightarrow P(Y|X) - P(Y) = P(T|Z) - P(Y) \Leftrightarrow \frac{P(Y|X)-P(Y)}{1-P(Y)} = \frac{P(T|Z)-P(T)}{1-P(T)} \Leftrightarrow M_{GK}(X \rightarrow Y) = M_{GK}(Z \rightarrow T)$. □

Proposition 9. *Let $X, Y, T, Z \subseteq \mathcal{I}$, such that X disfavors Y and Z disfavors T, and $X \cap Y = Z \cap T = \emptyset$, and $X \subset Z \subseteq \gamma(X)$, and $Y \subset T \subseteq \gamma(Y)$. Then, $supp(X \cup Y) = supp(Z \cup T)$ and $M_{GK}(X \rightarrow \overline{Y}) = M_{GK}(Z \rightarrow \overline{T})$.*

Proof. $\forall X, Y, T, Z \subseteq \mathcal{I}$, $supp(X \cup Y) = \frac{|g(X\cup Y)|}{|\mathcal{T}|}$ and $supp(Z \cup T) = \frac{|g(Z\cup T)|}{|\mathcal{T}|}$. Because $X \subset Z \subseteq \gamma(X)$ and $Y \subset T \subseteq \gamma(Y)$, we have $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$. It causes $g(X) = g(Z)$ and $g(Y) = g(T)$ implies $supp(X \cup Y) = supp(Z \cup T)$. Because $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$, we have $P(Y|X) = P(T|Z) \Leftrightarrow P(\overline{Y}|X) = P(\overline{T}|Z) \Leftrightarrow P(\overline{Y}|X) - P(\overline{Y}) = P(\overline{T}|Z) - P(\overline{Y}) \Leftrightarrow \frac{P(\overline{Y}|X)-P(\overline{Y})}{1-P(\overline{Y})} = \frac{P(\overline{T}|Z)-P(\overline{T})}{1-P(\overline{T})} \Leftrightarrow M_{GK}(X \rightarrow \overline{Y}) = M_{GK}(Z \rightarrow \overline{T})$. □

Proposition 10. *Let $X, Y, T, Z \subseteq \mathcal{I}$, such that X disfavors Y and Z disfavors T, and $X \cap Y = Z \cap T = \emptyset$, and $X \subset Z \subseteq \gamma(X)$, and $Y \subset T \subseteq \gamma(Y)$. Then, $supp(X \cup Y) = supp(Z \cup T)$ and $M_{GK}(\overline{X} \rightarrow Y) = M_{GK}(\overline{Z} \rightarrow T)$.*

Proof. $\forall X, Y, T, Z \subseteq \mathcal{I}$, $supp(X \cup Y) = \frac{|g(X\cup Y)|}{|\mathcal{T}|}$ and $supp(Z \cup T) = \frac{|g(Z\cup T)|}{|\mathcal{T}|}$. Because $X \subset Z \subseteq \gamma(X)$ and $Y \subset T \subseteq \gamma(Y)$, we have $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$. It causes $g(X) = g(Z)$ and $g(Y) = g(T)$ implies $supp(X \cup Y) = supp(Z \cup T)$. Because $supp(X) = supp(Z)$ and $supp(Y) = supp(T)$, we have $P(Y|X) = P(T|Z) \Leftrightarrow P(Y|\overline{X}) = P(T|\overline{Z}) \Leftrightarrow P(Y|\overline{X}) - P(T) = P(T|\overline{Z}) - P(T) \Leftrightarrow \frac{P(Y|\overline{X})-P(Y)}{1-P(Y)} = \frac{P(T|\overline{Z})-P(T)}{1-P(T)} \Leftrightarrow M_{GK}(\overline{X} \rightarrow Y) = M_{GK}(\overline{Z} \rightarrow T)$. □

The following Subsection summarizes these different optimizations via the algorithm 1 and the algorithm 3.

3.3 Our Algorithm

As we mentioned, our approach describes in a double problematic: mining frequent itemsts (algorithm 1) and generation of potential valid positive and negative association rules (algorithm 3). The algorithm 1 takes as argument a context $\mathcal{B}$, a *minsup*. It returns a set $\mathcal{F}$ of frequent itemsets, where $\mathcal{C}_k$ denotes the set of candidate k-itemsets, and $\mathcal{CGM}_k$ the set of generator k-itemsets. The database

Algorithm 1 EOMF: Frequent Itemset Mining

Require: A dataset $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$, a minimum support *minsup*.
Ensure: All frequent itemsets $\mathcal{F}$.
1: MATRICESUPPORT ← `Scan`($\mathcal{B}$); //Scan dataset $\mathcal{B}$
2: $\mathcal{F}_1 \leftarrow \{c_1 \in$ MATRICESUPPORT$|supp(c_1) \geq minsup\}$; //Generate 1-itemsets
3: $\mathcal{F}_2 \leftarrow \{c_2 \in$ MATRICESUPPORT$|supp(c_2) \geq minsup\}$; //Generate 2-itemsets
4: **for** $(k = 3; \mathcal{F}_{k-1} \neq \emptyset; k++)$ **do**
5: $C_k \leftarrow$ EOMF-GEN$(\mathcal{F}_{k-1})$; //New candidate (see algorithm 2)
6: **for all** (transaction $t \in \mathcal{T}$) **do**
7: $C_t \leftarrow$ subset(C_k, t) or $C_t = \{c \in C_k | c \subseteq t\}$ //Select candidate in t
8: **for all** (candidate $c \in C_t$) **do**
9: **if** $(c \in \mathcal{CGM}_k)$ **then**
10: $supp(c) = |\{t \in \mathcal{T} | c \subseteq t\}|/|\mathcal{T}|$;
11: **else**
12: $supp(c) = min\{supp(c') | c' \subset c\}$;
13: **end if**
14: $supp(c)++$;
15: **end for**
16: **end for**
17: $\mathcal{F}_k \leftarrow \{c \in C_k | supp(c) \geq minsup\}$; //Generate frequent itemsets
18: **end for**
19: **return** $\mathcal{F} = \bigcup_k \mathcal{F}_k$

$\mathcal{B}$ is built in line 1. Next, $\mathcal{F}_1$ and $\mathcal{F}_2$ are generated in a single pass (algo. 1 lines 2 and 3). The EOMF-GEN function (algorithm 2) is called to generate candidates

Algorithm 2 EOMF-GEN Procedure

Require: A set $\mathcal{F}_{k-1}$ of frequent $(k-1)$-itemset
Ensure: A set C_k of candidate k-itemset
1: $C_k \leftarrow \emptyset$
2: **for all** itemset $p \in \mathcal{F}_{k-1}$ **do**
3: **for all** itemset $q \in \mathcal{F}_{k-1}$ **do**
4: **if** $(p[1] = q[1], \ldots, p[k-2] = q[k-2], p[k-1] < q[k-1])$ **then**
5: $c \leftarrow p \cup q(k-1)$; //Generate candidate
6: **end if**
7: **for all** ($(k-1)$-subset s of c) **do**
8: **if** $(s \in \mathcal{F}_{k-1})$ **then**
9: $C_k \leftarrow C_k \cup \{c\}$;
10: **end if**
11: **end for**
12: **end for**
13: **end for**
14: **return** C_k

(algo. 1 line 5). It takes as argument $\mathcal{F}_{k-1}$, and returns a superset C_k. The initialization of C_k to the empty set is done in line 1 (algo. 2). A join between

the elements of $\mathcal{F}_{k-1}$ is then made (algo. 2 lines 2 to 6). Indeed, two p and q items of $\mathcal{F}_{k-1}$ form a c if, and only if they contain common $(k-2)$-itemsets. For example, joining ABC and ABD gives $ABCD$. However, joining ABC and CDE does not work because they do not contain common 2-itemsets. Once C_k has been established, it researches among the elements of C_k. If this is the case, it calculates the support in two cases (algo. 1 lines 9 to 13): if c is generator, an access to the database is made to know its support (algo. 1 line 10), otherwise, it is derived from its subsets without going through the database (algo. 1 line 12). The support is then increased (algo. 1 line 14). And, only frequent itemsets are retained in $\mathcal{F}_k$ (algo. 1 line 17). For the sake of comprehension, we apply this algorithm 1 on a small database $\mathcal{B}$, shown in Table 1. The minimum support is equal to 2/6, where Gen. designates a generator itemset. Results are shown in Fig. 1. After reading the dataset $\mathcal{B}$, D is not frequent, its support is smaller than

Scan Database $\longrightarrow$

MatriceSupport					
$i \backslash j$	A	B	C	D	E
A	3	2	3	1	2
B	-	5	4	0	5
C	-	-	5	1	4
D	-	-	-	1	0
E	-	-	-	-	5

Generate $\mathcal{F}_1$ and $\mathcal{F}_2$ $\longrightarrow$

C_1	Gen	$supp$	$\mathcal{F}_1$
A	yes	3/6	A
B	yes	5/6	B
C	yes	5/6	C
D	yes	1/6	-
E	yes	5/6	E

C_2	Gen	$supp$	$\mathcal{F}_2$
AB	yes	2/6	AB
AC	no	3/6	AC
AE	yes	2/6	AE
BC	yes	4/6	BC
BE	no	5/6	BE
CE	yes	4/6	CE

Generate C_3 $\longrightarrow$

C_3	Gen	$supp$
ABC	no	min(2/6,3/6,4/6)=2/6
ABE	no	min(2/6,2/6,5/6)=2/6
ACE	no	min(3/6,2/6,4/6)=2/6
BCE	no	min(4/6,5/6,4/6)=4/6

Generate $\mathcal{F}_3$ $\longrightarrow$

$\mathcal{F}_3$	Gen	$supp$
ABC	no	2/6
ABE	no	2/6
ACE	no	2/6
BCE	no	4/6

Generate C_4 $\longrightarrow$

C_4	Gen	$supp$
ABCE	no	$min(\frac{2}{6}, \frac{2}{6}, \frac{2}{6}, \frac{4}{6}) = 2/6$

Generate $\mathcal{F}_4$ $\longrightarrow$

$\mathcal{F}_4$	Gen	$supp$
ABCE	no	2/6

Fig. 1: Example of the Algorithm 1, $minsup = 2/6$

the $minsup$. It is pruned from the next step. The other elements are kept to generate C_2. These elements are frequent, its gives F_2. Then, C_3 is generated. We have $supp(AC) = supp(A)$ and $supp(BE) = supp(B) = supp(E)$, AC and BE are not generators. No candidate of C_3 is then generator, i.e. no access over the $\mathcal{B}$. Also in the last step, the support of $ABCE$ is equal to ABC (or ABE, or ACE). From this example, our approach does it in a single pass to the database, this is not the case for the existing ones, they do it in 4 passes.

The following algorithm 3 embodies the different optimizations we have defined in above Subsection 3.2. The algorithm 3 takes as argument a set $\mathcal{F}$, thresholds $minsup$ and $minmgk$, and returns a set $\mathcal{E}_{RAPN}$. It is initialized by the empty set in line 1. Next, for each itemset of $\mathcal{F}$ set, the set $\mathcal{A}$ is gererated (line 3). For each subset X_{k-1} of $\mathcal{A}$ (line 4), the algorithm proceeds in two recursive steps. The first consists in generating attractive class rules using the single

Algorithm 3 Association Rules Generation

Require: A set $\mathcal{F}$ of frequent itemsets, a $minsup$ and $minmgk$.
Ensure: A set $\mathcal{E}_{RAPN}$ of valid positive and negative rules.
1: $\mathcal{E}_{RAPN} = \emptyset$;
2: **for all** (k-itemset X_k of $\mathcal{F}$, $k \geq 2$) **do**
3: $\mathcal{A} = \{(k-1)$-itemset $\mid X_{k-1} \subset X_k\}$
4: **for all** ($X_{k-1} \in \mathcal{A}$) **do**
5: $X = X_{k-1}$; $Y = X_k \backslash X_{k-1}$;
6: **if** ($P(Y|X) > P(Y)$) **then**
7: $supp(X \cup Y) = \frac{|g(X \cup Y)|}{|\mathcal{T}|}$; $M_{GK}(X \rightarrow Y) = \frac{P(Y|X)-P(Y)}{1-P(Y)}$;
8: **if** ($supp(X \cup Y) \geq minsup$ & $M_{GK}(X \rightarrow Y) \geq minmgk$) **then**
9: $\mathcal{E}_{RAPN} \leftarrow \mathcal{E}_{RAPN} \cup \{X \rightarrow Y, Y \rightarrow X, \overline{Y} \rightarrow \overline{X}, \overline{X} \rightarrow \overline{Y}\}$;
10: **end if**
11: **else**
12: $supp(X \cup \overline{Y}) = \frac{|g(X \cup \overline{Y})|}{|\mathcal{T}|}$; $M_{GK}(X \rightarrow \overline{Y}) = \frac{P(\overline{Y}|X)-P(\overline{Y})}{1-P(\overline{Y})}$;
13: **if** ($supp(X \cup \overline{Y}) \geq minsup$ & $M_{GK}(X \rightarrow \overline{Y}) \geq minmgk$) **then**
14: $\mathcal{E}_{RAPN} \leftarrow \mathcal{E}_{RAPN} \cup \{X \rightarrow \overline{Y}, Y \rightarrow \overline{X}, \overline{Y} \rightarrow \overline{X}, \overline{X} \rightarrow Y\}$;
15: **end if**
16: **end if**
17: **end for**
18: **end for**
19: **return** $\mathcal{E}_{RAPN}$

rule $X \rightarrow Y$ (algo. 3 lines 6 to 11). Indeed, if $supp(X \rightarrow Y) \geq minsup$ and $M_{GK}(X \rightarrow Y) \geq minmgk$, then the $\mathcal{E}_{RAPN}$ set is updated by adding $X \rightarrow Y$, $Y \rightarrow X$, $\overline{Y} \rightarrow \overline{X}$ and $\overline{X} \rightarrow \overline{Y}$ (algo. 3 line 9). The second step consists in generating repulsive class rules by studying only $X \rightarrow \overline{Y}$ (algo. 3 lines 11 to 16). The algorithm 3 is updated by adding $X \rightarrow \overline{Y}$, $Y \rightarrow \overline{X}$, $\overline{Y} \rightarrow X$ and $\overline{X} \rightarrow Y$ (algo. 3 line 14). ERAPN returns the $\mathcal{E}_{RAPN}$ set (algo. 3 line 19).

Example illustrate of algorithm 3. Indeed, we consider the frequent itemset $ABC \subseteq \mathcal{F}$ (cf. Fig. 1). We will study a total $|\mathcal{E}_{RAPN}(ABC)| = 72$ rules, where 12 positive rules and 60 negative rules. First, we start to study the positive rules. There are 6 possible rules: $A \rightarrow BC$, $B \rightarrow AC$, $C \rightarrow AB$, $AB \rightarrow C$, $AC \rightarrow B$ and $BC \rightarrow A$. Since ABC is frequent, then its subests A, B and C are also frequent, which gives the other candidates $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$, $B \rightarrow A$, $C \rightarrow A$ and $C \rightarrow B$. Indeed, we will study first $A \rightarrow B$, $A \rightarrow C$ and $B \rightarrow C$. Given $minsup = 0.1$ and $minmgk = 0.6$. Results are shown in Table 4 below. Because $M_{GK}(B \rightarrow C) = M_{GK}(B \rightarrow A) = M_{GK}(C \rightarrow A) = -0.2 < 0.6$ and

Table 4: Generation of positive association rules, $minsup = 0.1$ and $minmgk = 0.6$

$X \rightarrow Y$	$P(X)$	$P(Y)$	$supp(X \cup Y)$	$P(Y\|X) - P(Y)$	$1 - P(Y)$	$M_{GK}(X \rightarrow Y)$
$A \rightarrow B$	0.50	0.83	0.33	0.17	-0.17	-1
$A \rightarrow C$	0.50	0.83	0.50	0.17	0.17	1
$B \rightarrow C$	0.83	0.83	0.67	0.17	-0.03	-0.2
$B \rightarrow A$	0.83	0.50	0.33	0.50	-0.10	-0.2
$C \rightarrow A$	0.83	0.50	0.50	0.50	0.10	0.2
$C \rightarrow B$	0.83	0.83	0.67	0.17	-0.03	-0.2

$M_{GK}(C \to B) = 0.2 < 0.6$, then $B \to C$, $B \to A$, $C \to A$ and $C \to B$ are not valid. So, by Proposition 7 and 8, $A \to BC$, $BC \to A$, $B \to AC$ and $C \to AB$ are also invalid. Since $M_{GK}(A \to B) = -1 < 0$, then $A \to B$ is invalid.

Here, we derive the valid positive and negative tradional. Because, $supp(A\overline{B}) = supp(A) - supp(AB) = 0.17 > 0.1$, and, by proposition 6, $M_{GK}(A \to \overline{B}) = -M_{GK}(A \to B) = 1 > 0.6$. Therefore, $A \to \overline{B}$ is exact negative rule. Since $supp(AC) = 0.50 > 0.1$ and $M_{GK}(A \to C) = 1 > 0.6$, $A \to C$ is exact positive rule. Because, $A \to C$ is exact, by Proposition 7, $AB \to C$ is also exact rule. Because $A \to \overline{B}$ is exact negative, by Proposition 9, $AC \to \overline{B}$ is also exact negative. Because $\overline{A} \to B$ is exact negative, by Corollary 1, $\overline{A}B \to B$ is also exact negative. Because $\overline{B} \to A$ is exact negative, by Proposition 10, $\overline{B} \to AC$ is also exact negative. Therefore, because $\overline{B} \to AC$ is exact negative, by Proposition 7, $\overline{B} \to C$ is also exact negative. Results are shown in following Table 5. In this

Table 5: Potential valid association rules according to support-M_{GK}

$X \to Y$	$P(X)$	$P(Y)$	$supp(X \cup Y)$	$P(Y\|X) - P(Y)$	$1 - P(Y)$	$M_{GK}(X \to Y)$
$A \to C$	0.50	0.83	0.50	0.17	0.17	1
$AB \to C$	0.33	0.83	0.33	0.17	0.17	1
$A \to \overline{B}$	0.50	0.83	0.33	0.17	0.17	1
$AC \to \overline{B}$	0.50	0.83	0.33	0.17	0.17	1
$\overline{A} \to B$	0.50	0.83	0.50	0.17	0.17	1
$\overline{AC} \to B$	0.50	0.83	0.50	0.17	0.17	1
$\overline{B} \to A$	0.17	0.50	0.17	0.50	0.50	1
$\overline{B} \to C$	0.17	0.83	0.17	0.17	0.17	1
$\overline{B} \to AC$	0.17	0.50	0.17	0.50	0.50	1

small example, our approach restores nine valid positive and negative association rules in total, including two positve rules of type $X \to Y$, two negative rules of type $X \to \overline{Y}$ and five negative rules of type $\overline{X} \to Y$.

Complexity of ERAPN algorithm There are three lenses: *average*, *best* and *worst case*. The first model evaluates the average time, which proves to be very difficult and leaves the framework of this work. The second model estimates the minimal time, which also leaves the framework of this work. We are interested in the last one, because we want to evaluate the costs of calls of the most expensive operations. In what follows, we present the study of the complexity of our ERAPN algorithm. This is calculated for each of the two constituting steps: frequent itemsets mining and finding positive and negative association rules.

Complexity of frequent itemsets mining (Algorithm 1): The algorithm 1 takes as input the transaction context $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$. Let $n = |\mathcal{T}|$ and $m = |\mathcal{I}|$. There is worst case if the candidate are generators (i.e. $2^{\mathcal{I}}$). The time complexity of support counting for 1 and 2-itemsets is $\mathcal{O}(m \times n)$ (line 1). The instructions for

lines 2-3 are $\mathcal{O}(2)$. The cost of finding longest frequent itemsets (i.e. all itemsets of sizes ≥ 3) (lines 4-16) is equal to the sum of the following costs. EOMF-GEN: there are $(2^m - m - 1)$ candidates to generate. Thus, the cost of this procedure is $\mathcal{O}(2^m - m)$ (lines 2-13 in the algorithm 2). The cost of support counting of longest candidates is $\mathcal{O}(n(2^m - m))$ (lines 6-16). The time complexity of space frequent itemsets is $\mathcal{O}(2^m - m)$ (line 17). The global complexity of this algorithm 1 is therefore $\mathcal{O}(mn + 2^m - m + n(2^m - m) + 2^m - m) = \mathcal{O}(n2^m)$.

Complexity of rule generation (Algorithm 3): The algorithm takes as input a set of frequent itemsets $\mathcal{F}$, which is obtained from a context $\mathcal{B}$. Its global complexity is linear in $|\mathcal{F}|$, which takes $\mathcal{O}(2^{-1}|\mathcal{F}|(5^m - 2(3^m)))$. This complexity is obtained by the following instructions. The "for" loop (line 2), which runs through all of the $\mathcal{F}$ itemsets, is done in $\mathcal{O}(|\mathcal{F}|)$ at worst. The second "for" loop (line 4) is $\mathcal{O}(|\mathcal{A}|/2)$ at worst, because only half of the candidate rules that are traversed in our approach to test their eligibility (instructions 6 to 16). It is carried out in two identical tests (lines 8 and 13). For each of the tests, the possible number of rules generated, at a m-itemset, is equal to $2^{2m} - 2^{m+1}$. Which gives $C_m^{m-1}(2^{2(m-1)} - 2^m)$ for a $(m-1)$-itemset, $C_m^{m-2}(2^{2(m-2)} - 2^{(m-1)})$ for a $(m-2)$-itemset, and so an. In sum, we have $\mathcal{O}(|\mathcal{A}|) = \sum_{k=2}^{m} C_m^k(2^{2k} - 2^{k+1}) = \sum_{k=2}^{m} C_m^k 4^k - 2\sum_{k=2}^{m} C_m^k 2^k = \left[\sum_{k=0}^{m} C_m^k 4^k - (1+4m)\right] - 2\left[\sum_{k=0}^{m} C_m^k 2^k - (1+2m)\right]$. Now, for all x of $\mathbb{R}$, $\sum_{k=0}^{m} C_m^k x^k = (1+x)^m$, so $\mathcal{O}(|\mathcal{A}|) = \mathcal{O}(5^m - 2(3^m) + 1) = \mathcal{O}(5^m - 2(3^m))$. Finally, the overall time complexity is $\mathcal{O}(|\mathcal{F}||\mathcal{A}|/2) = \mathcal{O}(2^{-1}|\mathcal{F}|(5^m - 2(3^m)))$.

In the worst case, the total complexity of the ERAPN algorithm is of the order of $\mathcal{O}(2^{-1}|\mathcal{F}|(5^m - 2(3^m))) + \mathcal{O}(n2^m) = \mathcal{O}(2^{-1}|\mathcal{F}|(5^m - 2(3^m)) + n2^m)$.

4 Experimental results

This section presents the experimental study conducted in order to evaluate the performances of our algorithm. The latter is implemented in R and tested on PC Core i3 and 4GB of RAM running under Windows system. We compare the results with those of Wu and RAPN, conducted out on four databases from UCI, such as `Adult, German, Income` and `Iris`. For each algorithm, we have chosen the same thresholds to avoid biasing the results. The following table 6 reports the characteristics of datasets, and the number of positive and negative rules by varying the minimum thresholds *minsup* and *minmgk*. Indeed, the first three columns indicate the data characteristics in question, the last fiveteen columns present the different results, where the column labelled "++" corresponds to the type $X \to Y$, column "-+" to $\overline{X} \to Y$, column "+-" to $X \to \overline{Y}$, and "- -" to $X \to \overline{Y}$. The behaviour of algorithms varies according to data characteristics. The large database is much more time-consuming to run. In other words, the number of rules increases as thresholds decrease. Except for dense databases (`Adult` and `German`) and relatively low thresholds ($minsup = 1\%$ et $minmgk = 60\%$), the number of rules (see Table 6) in Wu is 100581 and 89378 for RAPN. They are relatively large, due to the strong contribution of positive rules of type $X \to Y$

Table 6: Caracteristics of Datasets and Results extracts

Database	$\|\mathcal{T}\|$	$\|\mathcal{I}\|$	$minsup$	$minmgk$	Wu					RAPN					ERAPN				
					+ +	- +	+ -	- -	$\sum$	+ +	- +	+ -	- -	$\sum$	+ +	- +	+ -	- -	$\sum$
Adult	48842	115	1%	60%	97956	625	1215	785	100581	87800	542	615	421	89378	27500	422	510	352	28784
			2%	70%	55925	453	852	556	57786	53950	323	503	344	55120	25536	354	385	225	26500
			3%	80%	38750	345	412	310	39817	22033	156	254	145	22588	18523	124	154	124	18925
German	1000	71	1%	60%	51478	456	1148	555	53637	41235	401	565	412	42613	26456	340	380	245	27421
			2%	70%	39683	352	744	456	41235	38555	234	425	384	39598	18800	220	203	156	19379
			3%	80%	9835	144	545	321	10845	18500	75	325	232	19132	12157	95	85	55	15392
Income	6876	50	1%	60%	2800	227	527	254	3808	2130	350	385	286	3151	1552	95	103	84	1834
			2%	70%	2200	127	327	213	2867	2054	214	330	185	2783	1433	55	63	65	1616
			3%	80%	1325	87	252	121	1785	1212	65	156	45	1478	923	35	30	17	1005
Iris	150	15	1%	60%	2437	159	196	160	2952	1954	10	60	24	2048	1500	150	165	124	1939
			2%	70%	2000	159	145	120	2424	1323	10	55	20	1407	1122	75	85	65	1347
			3%	80%	1200	159	59	45	1463	1056	25	30	-	1111	965	14	22	18	1019

and negative rules of type $X \to \overline{Y}$ (see table 6), than for ERAPN (28784 rules). The rules of type $\overline{X} \to Y$ and $\overline{X} \to \overline{Y}$ remain reasonable for each algorithm. On less dense databases (`Income` and `Iris`), these algorithm gives the reasonable number of rules. Note that RAPN, for Iris data, does not extract the type $\overline{X} \to \overline{Y}$ (see Table 6) for $minsup$ (resp. $minmgk$) over 3% (resp. 80%). Figure 2 below shows the response times by varying the $minupp$ and keeping $minmgk = 60\%$. They also increase when thresholds are lowered. The execution time of ERAPN

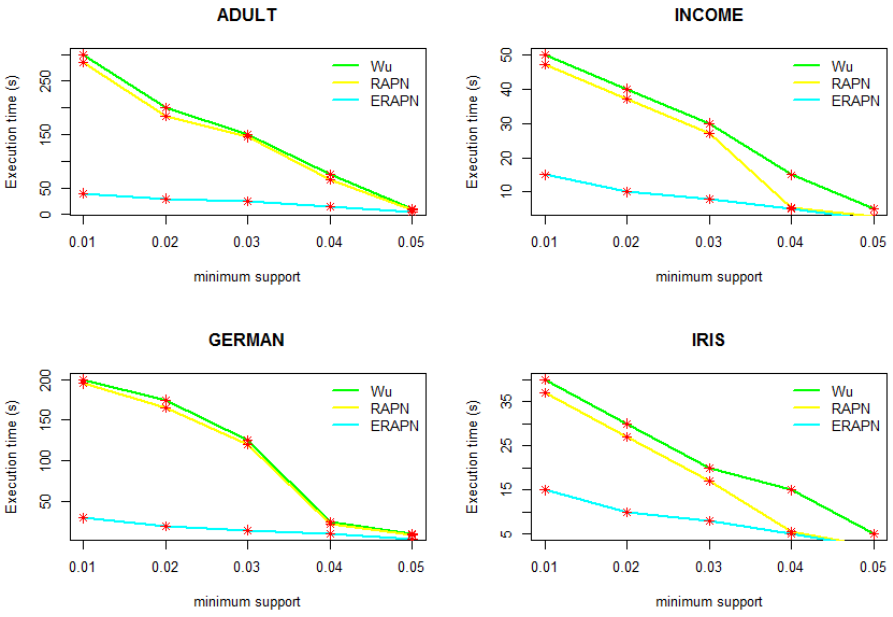

Fig. 2: Performances for each algorithm according to $minsup$

is faster than that of Wu and RAPN. ERAPN gained 7 more times the best response in the worst cases. These different performances can be explained as follows. RAPN and Wu are limited on classical data structure, which requires repetitive access over the whole database. Wu has the lowest performance. One of the main reasons lies in the pruning technique. In this case, the *interest* measure does not have effective properties for frequent itemsets mining. In addition, the search space of valid association rules can be covered exhaustively. To this, our algorithm introduces the different optimizations. Therefore, the all frequent itemsets can be traversed only once. Moreover, the search space of rules is only half full. In all cases, our model remains the most selective and concise.

5 Related Work

Association rules mining is an active topic in Big Data. Apriori algorithm [2] is the first model that deals with this topic. On the other hand, it scans database multiple times as long as large frequent itemsets are generated. Apriori TID algorithm [2] generates candidate itemset before database is scanned with the help of Apriori-Gen function. Database is scanned only first time to count support, rather than scanning database it scans candidate itemsets. Despite their notable contributions, Apriori and Apriori TID [2] algorithms are limited on a single type of classical (or positive) association rules. The negative association rules has not been studied. To this, outreach works has been proposed.

Brin et al. [10] propose a model generating negative association rules by using the χ^2 measure. It is a first time in the literature the notion of negative relationships. The statistical chi-square is used to verify the independence between two variables. It's also used to determine the nature of the relationship, and a correlation metric. Although effective, the model suffers the problem of space memory due to the chi-square χ^2 was used. In [15], the authors present an approach to mine strong negative rules. They combine positive frequent itemsets with domain knowledge in the form of a taxonomy to mine negative association rules. However, as mentioned in many works [3,14], their approach is hard to generalized since it is domain dependant and requires a predefined taxonomy. Boulicaut et al. [9] present an approach using contraints to generate the association of the form $X \wedge Y \rightarrow \overline{Z}$ or $\overline{X} \wedge Y \rightarrow Z$ with negations using closed itemsets. Despite its notable contribution, this method is limited of this form. Wu et al. [19] propose an approach for generating both positive and negative association rules. They add on top of the support-confidence framework other two measures, called *interest* and *CPIR* for a better pruning of the frequent itemset and frequent association rules, respectively. One of the key problems lies in pruning: no optimized techniques are used, and the search space can be exhaustively explored due to the measure *interest* was used. In [3], the authors propose an approach for mining positive and negative association rules. They add on top of the support-confidence framework another measure, called *Correlation coefficient*. Nevertheless, it requires to challenging problem of finding the frequent association rules, their strategy for search space is not optimized, which can be

costly. In [16], the authors propose a new algorithm SRM (substitution rules mining) for mining only negative association of the type $X \rightarrow \overline{Y}$. Although effective, SRM algorithm is limited of this only type. In [11], the authors propose the PNAR algorithm. Although obtaining notable contributions, PNAR suffers the high volume of results, due to support-confidence pair wase used. Wilhelmiina proposes the Kingfisher algorithm [18] using the Fisher test. A notable limitation of this model lies in the computation of *p-value* imposing exhaustive passes over the whole database, which gives the high computational time. Guillaume and Papon [14] propose RAPN algorithm based on support-confidence pair and other measure, M_G (M_{GK} [13] modified). Although effective, RAPN suffers relatively the high computational cost on the search space frequent itemsets.

Note that the major handicap of these works stems mainly from the computational costs for frequent itemsets mining (repetitive passes over the whole database) and association rules mining (exhaustive passes over the search space).

Recently, we proposed a new algorithm, EOMF [5], allowing the extraction of frequent itemsets. Therefore, a single pass over the database will extract all frequent itemsets, which significantly reduces the costs of calculation. As for association rules mining, we introduced in [7,8] a new approach allowing the extraction of positive and negative association rules using a new pair, support-M_{GK}. As a result, only half of all candidate rules are studied, which also reduces the search space significantly. In this paper, we combine our works [5,7,8]. Ameliorations have been made, especially in terms of accuracy and simplicity. In [5], the path of frequent itemsets space has been quite heavy: the non-generator itemsets are implicitly taken twice for each calculation step, which can be costly. This gap has been corrected in the current work. We introduced a new strategy of the search space via a notable property (cf. property 1) exploiting the monotony concepts of the generator itemsets, which consequently reduces the cost. Therefore, improvements have been added in algorithm 1 (lines 4 to 16), which makes the approach robust. In [7,8], we used the parameter $vc_\alpha(r) = \sqrt{\frac{1}{n}\frac{n-n_X}{n_X}\frac{n_Y}{n-n_Y}\chi^2(\alpha)}$, to prune association rules. Nevertheless, this parameter presents a notable limit. It requires the exhaustive paths over the whole database to know its values for each candidate association rule, i.e. for a m-itemset, computable in $\mathcal{O}(2^m)$ on its contingency table, it gives $4C_m^k 2^k$, tradionally cost, for all k. This parameter is not very selective, its sometimes eliminates the interesting rules (or robust), but considers the unintersing rules (far from the logical implication), because, of its critical value. In this paper, we try to close this limit using a simple parameter, $minmgk \in [0, 1]$, that does not require access to the database. In addition, we introduced effective properties for the search space (cf. propositions 7 to 10).

6 Conclusion

In this paper, we have studied the problems of positive and negative association rules for Big Data. Further optimizations have been defined. Experiments conducted on reference databases, compared to RAPN and Wu algorithms, have

emphasized the efficiency of our approach. A study on the extraction of disjunctions/conjunctions association rules has not been initially developed, which gives leads to explore from a methodological and algorithmic point of view.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In Proc. of ACM SIGMOD, 207–216 (1993).
2. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In Proceedings of 20th VLDB Conference, Santiago Chile, 487–499 (1994).
3. Antonie, M.-L., Zaïane, O. R.: Mining Positive and Negative Association Rules: An Approach for Confined Rules. In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD, 27–38 (2004).
4. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: PASCAL: un algorithme d'extraction des motifs fréquents. Tech. et Sces. Info., 65–95 (2002).
5. Bemarisika, P., Totohasina, A.: Eomf, un algorithme d'extraction optimisée des motifs fréquents. In Proc. of AAFD & SFC, Marrakech Maroc, 198–203 (2016).
6. Bemarisika, P.: Extraction de règles d'association selon le couple support-M_{GK}: Graphes implicatifs et Application en didactique des mathématiques. Université d'Antananarivo, Madagascar, (2016).
7. Bemarisika, P., Totohasina, A.: Optimisation de l'extraction des règles d'association positives et négatives. In Actes des 24èmes Rencontres de la Société Francophone de Classification, Lyon 1, France, 25–28, (2017).
8. Bemarisika, P., Totohasina, A.: Optimized Mining of Potential Positive and Negative Association Rules. In International Conference on Big Data Analytics and Knowledge Discovery, Lyon 2, France, 424–432, (2017).
9. Boulicaut, J.-F., Bykowski, A., Jeud, B.: Towards the tractable discovery of association rules with negations. Conference on FQAS'00, 425–434 (2000).
10. Brin, S., Motwani, R., Silverstein, C.: Bayond market baskets: Generalizing association rules to correlation. In Proc. of the ACM SIGMOD, 265–276 (1997).
11. Cornelis, C., Yan, P., Zhang, X., Chen, G.: Mining Positive and Negative Association Rules from Large Databases. In Proceedings of the IEEE, 613-618 (2006).
12. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations. Springer Verlag, (1999).
13. Guillaume, S.: Traitement de données volumineuses: Mesure et algorithmes d'extraction de règles d'association. Ph.D. thesis, Université de Nantes, (2000).
14. Guillaume, S., Papon, P.-A.: Extraction optimisée de règles d'association positives et négatives (RAPN). In Actes de la 13e Conf. Int. Franco. EGC, 157–168 (2013).
15. Savasere, A., Omiecinski, E., Navathe, S.: Mining for strong negative associations in a large database of customer transactions. In Proc. of ICDE, 494–502 (1998).
16. Teng, W.-G., Ming-Jyh, H., Ming-Syan, C.: A statistical framework for mining substitution rules. In Knowl. Inf. Syst., Volume (7), 158–178 (2005).
17. Totohasina, A., Ralambondrainy H.: ION, A pertinent new measure for mining information from many types of data. In IEEE, SITIS, 202–207 (2005).
18. Wilhelmiina: Kingfisher: An efficient algorithm for searching for both positive and negative dependence rules with statistical significance measures. In Knowl. Inf. Syst.,383–414 (2012).
19. Wu, X., Zhang, C., S. Zhang, S.: Efficient mining of both positive and negative association rules. In ACM Transactions on information Systems 3, 381–405 (2004).