Generating a condensed representation for association rules

Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, Lotfi Lakhal

Generating a Condensed Representation for Association Rules

Nicolas Pasquier (nicolas.pasquier@unice.fr)
I3S (CNRS UMR 6070) - Université de Nice-Sophia Antipolis, 06903 Sophia Antipolis, France

Rafik Taouil (taouil@univ-tours.fr)
LI - Université Francois Rabelais de Tours, 3 place Jean Jaurès, 41000 Blois, France

Yves Bastide (yves.bastide@irisa.fr)
IRISA - INRIA Rennes, campus universitaire de Beaulieu, 35042 Rennes, France

Gerd Stumme (stumme@uni-kassel.de)
Fachbereich Mathematik/Informatik, Universität Kassel, 34121 Kassel, Germany

Lotfi Lakhal (lotfi.lakhal@lim.univ-mrs.fr)
LIM (CNRS FRE 2246) - Université de la Méditerranée, case 901, 13288 Marseille, France

Abstract. Association rule extraction from operational datasets often produces several tens of thousands, and even millions, of association rules. Moreover, many of these rules are redundant and thus useless. Using a semantic based on the closure of the Galois connection, we define a condensed representation for association rules. This representation is characterized by frequent closed itemsets and their generators. It contains the non-redundant association rules having minimal antecedent and maximal consequent, called min-max association rules. We think that these rules are the most relevant since they are the most general non-redundant association rules. Furthermore, this representation is a basis, i.e., a generating set for all association rules, their supports and their confidences, and all of them can be retrieved needless accessing the data. We introduce algorithms for extracting this basis and for reconstructing all association rules. Results of experiments carried out on real datasets show the usefulness of this approach. In order to generate this basis when an algorithm for extracting frequent itemsets – such as APRIORI for instance – is used, we also present an algorithm for deriving frequent closed itemsets and their generators from frequent itemsets without using the dataset.

Keywords: Data mining, Galois closure operator, frequent closed itemsets, generators, min-max association rules, basis for association rules, condensed representation.

1. Introduction

The purpose of association rule extraction, introduced in (Agrawal et al., 1993), is to discover significant relations between binary attributes, called *items*, in large datasets. An example of association rule extracted from a dataset of supermarket sales is: 'cereals $\wedge$ sugar $\rightarrow$ milk (*support*=7%, *confidence*=67%)'. This rule states that customers who buy cereals and sugar also tend to buy milk. The support measure defines the range of the rule, i.e., the proportion of customers who bought the three items among all customers. The confidence measure defines the precision of the rule, i.e., the proportion of customers who bought milk among those who bought cereals and sugar. Only rules with support and confidence above some minimal support and confidence thresholds, defined by the analyst according to the application, are extracted.
Classical approaches for mining association rules operate in two phases:

1. Extracting *frequent itemsets* and their support from the dataset. Frequent itemsets are sets of items contained in a proportion of objects above the minimum support threshold.

2. Generating association rules from frequent itemsets and supports. Only rules with confidence above the minimum confidence threshold are generated.

The first phase is the most computationally intensive, since the number of potential frequent itemsets is exponential in the size of the set of items and several dataset scans, very expensive in execution times, are required to count their supports. Classical approaches can be classified into three main trends. Approaches in the first trend are based on the *levelwise* extraction of frequent itemsets (Agrawal and Srikant, 1994; Mannila et al., 1994). That is a breadth-first exploration of the search space where all potential frequent itemsets of a given size are considered simultaneously (Mannila and Toivonen, 1997). These approaches are efficient for mining association rules from weakly correlated data, such as market basket data, but performances drastically decrease when data are dense or correlated, such as statistical data for instance. Approaches in the second trend are based on the extraction of maximal[1] frequent itemsets (Bayardo, 1998; Lin and Kedem, 1998; Zaki et al., 1997) to improve the efficiency. Once all maximal frequent itemsets are extracted, all frequent itemsets are derived and their support are counted in the dataset. In the third trend, approaches are based on the extraction of *frequent closed itemsets* (Pasquier et al., 1998; Zaki and Ogihara, 1998) defined using the Galois closure operator. These approaches first extract all frequent closed itemsets and then, both frequent itemsets and their support are derived from them, without dataset access. In the case of dense or correlated data, there are much fewer frequent closed itemsets than frequent itemsets and thus, these approaches improve the extraction efficiency compared to approaches in the first trend. Compared to approaches in the second trend, approches based on frequent closed itemsets can be more efficient in the case of correlated data due to the cost of generating all subsets of the maximal frequent itemsets and counting their support in the dataset.

Another major research topic in data mining is the problem of relevance and usefulness of extracted association rules. This problem is related to the number of extracted rules – that is most often very large – and to the important proportion of redundant rules, i.e. rules bringing the same information, among them. This problem becomes crucial when data are dense or correlated, such as statistical data, telecommunication data or nominative market basket data (Bayardo and al., 2000; Brin and al., 1997; Siverstein et al., 1998). For instance, using a census dataset sample constituted of 10,000 objects, each one containing values of 73 binary attributes, more than 2,000,000 association rules with support and confidence above 90% were extracted. The analyst is then confronted with the following problems: How to handle such a list of association rules ? Is it possible to reduce its size without losing information ? Moreover, the inspection of extracted association rules shown that redundant rules represent the majority of them. Their suppression will thus considerably reduce the number of rules to be handled by the analyst. In the previous example, this

[1] All maximal and minimal sets considered are defined according to the inclusion relation.

suppression reduced the number of rules to a few thousands. In addition, redundant rules can be misleading as discussed in example 1. Thus, the following question arises: How to reduce extracted association rules to a smaller list containing only non-redundant association rules ?

Example 1. To illustrate the problem of redundant association rules, we present nine rules extracted from the MUSHROOMS dataset describing characteristics of 8 416 mushrooms (Blake and Merz, 1998) in table I. These rules have identical supports and confidences, of 51% and 54% respectively, and the item "free gills" in the antecedent.

Table I. Redundant association rules.

1) free gills → edible	6) free gills, partial veil → edible, white veil
2) free gills → edible, partial veil	7) free gills, white veil → edible
3) free gills → edible, white veil	8) free gills, white veil → edible, partial veil
4) free gills → edible, partial veil, white veil	9) free gills, partial veil, white veil → edible
5) free gills, partial veil → edible	

Obviously, rules 1 to 3 and 5 to 9 do not add any information to rule 4 since all these rules have identical supports and confidences. We thus say that these rules are redundant compared to rule 4, the most relevant from the analyst's point of view for it summarizes the nine rules. This rule has a *minimal antecedent* (left-hand side) and a *maximal consequent* (right-hand side) among the nine rules. Moreover, examining only one of these eight rules, say for instance rule 9, the analyst will believe that a mushroom has 54% chances to be edible if it has free gills and a partial white veil. As a matter of fact, it has 54% chances to be edible *and* have a partial white veil if it has free gills. Redundant rules can therefore be misleading and cause misinterpretations of the results. We believe that extracting only rule 4 will improve the result relevance.

In the rest of the paper, we differentiate *exact association rules*, noted $l \Rightarrow l'$, that have a 100% confidence, and *approximate association rules*, noted $l \rightarrow l'$, that have a confidence lower than 100%. Exact association rules are valid for all objects in the dataset whereas approximate association rules are valid for a proportion of objects equal to their confidence.

1.1. Related Work

Approaches addressing this issue can be classified into three main trends. Approaches in the first trend provide mechanisms for filtering extracted association rules. In the two other trends, approaches "extend" the definition of association rules in order not to extract "similar" ones.
Approaches in the first trend allow the analyst to define some templates (Baralis and Psaila, 1997; Klemettinen and al., 1994), boolean operators (Bayardo and al., 2000; Ng et al., 1998; Srikant et al., 1997) or SQL-like operators (Meo et al., 1998) in order to select rules according to his/her preferences. In (Bayardo and al., 2000),

boolean operators are coupled with further measures of "usefulness" of the rules. By selecting a subset of all extracted association rules, these approaches reduce the number of rules to handle during the visualization, but redundancies are not suppressed.

In the second trend, some approaches use a taxonomy of items to extract generalized association rules (Han and Fu, 1999; Srikant and Agrawal, 1995), i.e., association rules between sets of items that belong to different levels of the taxonomy. Some approaches use statistical measures, such as Pearson's correlation or χ^2 test for instance, instead of the confidence to determine the precision of the rule (Brin and al., 1997; Morimoto et al., 1998; Siverstein et al., 1998). Other approaches in this trend allow to extract only rules with maximal antecedents among those with the same supports and the same consequents (Srikant and Agrawal, 1996; Toivonen et al., 1995). That is, a rule r will be pruned if another rule r' has the same consequent and an antecedent that is a superset of the one of r. In example 1, rules 4, 6, 8 and 9 have maximal antecedents and will be extracted. Finally, the approach proposed in (Bayardo and Agrawal, 1999) identifies optimal rules according to several interestingness metrics (confidence, conviction, lift, Laplace, gain, etc.) and a partial order on the rules.

Approaches in the third trend make use of the closure of the Galois connection to extract *bases*, or *reduced covers*, for association rules. Informally, a basis is a non-redundant set that is minimal according to some mathematical property and from which all association rules are deducible, with support and confidence, without accessing the dataset. These bases are adaptations of the Duquenne-Guigues basis for global implications (Duquenne and Guigues, 1986; Ganter and Wille, 1999) and the Luxenburger basis for partial implications (Luxenburger, 1991). They were introduced in Formal Concept Analysis and their adaptation to the association rule framework is studied in (Pasquier et al., 1999c; Taouil et al., 2000; Zaki, 2000). In the Duquenne-Guigues basis for exact association rules, antecedents of rules are *frequent pseudo-closed itemsets* and consequents are frequent closed itemsets. In the Luxenburger basis for approximate association rules, both antecedents and consequents are frequent closed itemsets: We select approximate rules with both a maximal antecedent and a maximal consequent among rules having identical supports and confidences. In example 1, rule 9 will be the only one extracted. The union of the Duquenne-Guigues and the Luxenburger bases is a basis for all association rules. This basis is minimal with respect to the number of rules and, since for most data types there are much fewer frequent closed and pseudo-closed itemsets than there are frequent itemsets, it is very small. However, it does not contain non-redundant rules with minimal antecedent and maximal consequent.

In previous works about the pruning of redundant implication rules (functional dependencies), such as the canonical and the minimum covers definitions (Beeri and Bernstein, 1979; Maier, 1980), redundant rules are defined according to an inference system based on Armstrong axioms (Armstrong, 1974). However, these results cannot be directly applied to the association rule framework since redundant association rules cannot be defined according to this system: Supports and confidences are important information that must be considered to characterize redundant rules. Such an inference system for association rules does not exist to our knowledge.

The idea behind non-redundant association rules as defined hereafter is to identify the most relevant rules, each one bringing the same information as several others.

1.2. Contribution

Our goal is to improve association rules relevance and usefulness by extracting as few rules as possible without losing information. To achieve this, we propose to generate a *condensed representation* (Mannila and Toivonen, 1996) by maximizing the information brought by each rule. As pointed out in example 1, we believe that the most relevant association rules are the most general[2] non-redundant rules: Those with minimal antecedent and maximal consequent. Extracting such rules will improve the result usefulness, while reducing its size. Therefore, in the following:

- We define non-redundant association rules with minimal antecedent and maximal consequent, called *min-max association rules*. These rules are defined using the semantic for association rule extraction based on the Galois closure. Their antecedents and consequents are characterized by frequent closed itemsets and their *generators* (Pasquier et al., 1998).
- We show that the min-max association rules constitute a basis, called *min-max basis for association rules*. All association rules can be deduced by generating all the sub-rules of the min-max association rules, considering their supports and confidences.
- We propose efficient algorithms to generate the min-max basis from frequent closed itemsets and their generators, such as extracted by the Close (Pasquier et al., 1998; Pasquier et al., 1999b) and the A-Close (Pasquier et al., 1999a) algorithms. We also introduce algorithms to reconstruct all association rules, or a part of them, from this basis without having to access the data.
- We present the Close$^+$ algorithm that identifies frequent closed itemsets, their generators and their supports among frequent itemsets and their supports. This algorithm is simple and efficient since it does not require any dataset access. It enables the generation of the min-max basis when an algorithm for extracting all frequent itemsets, such as Apriori (Agrawal and Srikant, 1994) for instance, is used.

Extracting min-max association rules minimizes as much as possible the number of rules while keeping the same information in the result: Only the most general non-overlapping association rules are extracted and therefore redundant rules are pruned. Since for many real datasets redundant rules represent the majority of extracted rules, the reduction will be almost always significant. This reduction will be considerable in the case of dense or correlated data for which the total number of rules is very large and most are redundant (Bayardo and Agrawal, 1999; Brin and al., 1997; Siverstein et al., 1998).

[2] We say that a rule $r : a \rightarrow c$ is more general than a rule $r' : a' \rightarrow c'$ if they have identical supports and confidences, the antecedent a of r is a subset of a' and the consequent c of r is a superset of c'. r' is then called a sub-rule of r, and r a super-rule of r'.

With the min-max basis, the analyst is presented a set of rules covering all the attributes of the dataset: All of the data-space is characterized by the min-max rules, overcoming an important deficiency of most reduction methods where large sub-spaces of the data-space may be poorly characterized or even entirely uncharacterized (Bayardo and Agrawal, 1999). This property helps insuring that rules "surprising" for the analyst, that are important information (Piatetsky and Matheus, 1994; Silberschatz and Tuzhilin, 1996), will be present. Moreover, the min-max basis does not represent any information loss for the analyst: all information brought by the set of all association rules is brought by the min-max basis. This approach does not suffer of the problem of information loss – from the analyst's point of view – that is an important drawback in association rule reduction methods (Liu and al., 1999). If the analyst so wishes, it is also possible to efficiently deduce all other association rules, with supports and confidences, from the min-max basis alone.

1.3. Organization

In section 2, we recall the semantic for association rules based on the Galois connection and the Close algorithm for extracting frequent closed itemsets and generators. We also present the Close$^+$ algorithm for efficiently deriving frequent closed itemsets, their generators and their supports from frequent itemsets and their supports. Min-max association rules and the min-max basis for association rules are defined in section 3. Algorithms for generating this basis are also presented. In section 4, we present simple methods and algorithms for deriving all association rules from the min-max basis. Results of experiments conducted to evaluate the usefulness of this approach are given in section 5 and section 6 concludes the paper.

2. Semantic for association rules based on the Galois connection

The association rule extraction is performed from a data mining context[3], that is a triplet $\mathcal{D} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, where $\mathcal{O}$ and $\mathcal{I}$ are finite sets of objects and items respectively, and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$ is a binary relation. Each couple $(o, i) \in \mathcal{R}$ denotes the fact that the object $o \in \mathcal{O}$ is related to the item $i \in \mathcal{I}$. An itemset l is a set of items $l \subseteq \mathcal{I}, l \neq \varnothing$.

Example 2. A data mining context $\mathcal{D}$ constituted of six objects, each one identified by its *OID*, and five items is represented in table II. This context is used as support for the examples in the rest of the paper.

The Galois connection of a finite binary relation (Ganter and Wille, 1999) is a couple of applications (ϕ, ψ). ϕ associates with a set of objects $O \subseteq \mathcal{O}$ the items related to all objects $o \in O$ and ψ associates with an itemset $l \subseteq \mathcal{I}$ the objects related to all items $i \in l$. When an object o is related to all items $i \in l$, we say that o *contains* l. We denote *minsupp* and *minconf* the minimal support and confidence thresholds.

Definition 1. (Frequent itemsets) The support of an itemset l is the proportion of objects in the context containing l: $supp(l) = |\psi(l)| \,/\, |\mathcal{O}|$. l is a frequent itemset if $supp(l) \geq minsupp$.

[3] We will use *context* and *dataset* interchangeably in the sequel.

Table II. Data mining context $\mathcal{D}$.

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

Definition 2. (Association rules) An association rule r is an implication between two frequent itemsets $l_1, l_2 \subseteq \mathcal{I}$ with the form $l_1 \rightarrow (l_2 \backslash l_1)$ where $l_1 \subset l_2$. The support and confidence of r are defined by: $supp(r) = supp(l_2)$, $conf(r) = supp(l_2) \,/\, supp(l_1)$.

The closure operator $\gamma = \phi \circ \psi$ associates with an itemset l the maximal set of items common to all the objects containing l: The closure of an itemset is equal to the intersection of all the objects containing it. Using this closure operator, we define the frequent closed itemsets.

Definition 3. (Frequent closed itemsets) A frequent itemset $l \subseteq \mathcal{I}$ is a frequent closed itemset iff $\gamma(l) = l$. The minimal closed itemset containing an itemset l is its closure $\gamma(l)$.

The set of frequent closed itemsets and their supports is a minimal non-redundant generating set for all frequent itemsets and their supports, and thus for all association rules, their supports and their confidences. This theorem relies on the properties that the support of a frequent itemset is equal to the support of its closure and that maximal frequent itemsets are maximal frequent closed itemsets (Pasquier et al., 1998). In order to improve the efficiency of frequent closed itemset extraction, the Close and A-Close algorithms compute generators of frequent closed itemsets.

Definition 4. (Generators) An itemset $g \subseteq \mathcal{I}$ is a generator of a closed itemset l iff $\gamma(g) = l$ and $\nexists g' \subseteq \mathcal{I}$ with $g' \subset g$ such that $\gamma(g') = l$. A generator of cardinality k is a k-generator.

Generators are the minimal itemsets to consider for discovering frequent closed itemsets, by computing their closures. Based on the following lemma, Close and A-Close perform a breadth-first search for generators in a levelwise manner.

Lemma 1. All subsets $s \subseteq \mathcal{I}$ of a generator $g \subseteq \mathcal{I}$ are also generators. The closure of s is a closed subset of the closure of g: $\gamma(s) \subset \gamma(g)$.

Proof. See (Pasquier et al., 1999b).

2.1. Extracting frequent closed itemsets and generators with Close

The Close algorithm is an iterative algorithm for extracting generators and frequent closed itemsets in a levelwise manner. During an iteration k, a list of candidate k-generators is considered; their closures and their supports are computed from the dataset and infrequent generators are discarded. Frequent generators are then used to construct candidate (k+1)-generators. The closures of frequent generators are the frequent closed itemsets and the support of a generator is also the support of its closure.
During the k^{th} iteration, a set FC_k is considered. Each element of this set consists of three information: a k-generator, its closure and their support. The algorithm first initializes the candidate 1-generators in FC_1 with the list of 1-itemsets and then carries out some iterations. During each iteration k:

1. Closures of all candidate k-generators and their supports are computed: The number of objects containing a generator determines its support and their intersection generates its closure. Each object is considered once and this phase requires only one scan of the dataset.

2. Infrequent k-generators, i.e., generators with support lower than *minsupp*, are removed from FC_k.

3. The set of candidate (k+1)-generators is constructed by joining the frequent k-generators in FC_k as follows.

   a) Two k-generators in FC_k that have the same first k–1 items are joined to create a candidate (k+1)-generator. For instance, the 3-generators {ABC} and {ABD} will be joined in order to create the candidate 4-generator {ABCD}.
   b) Candidate (k+1)-generators that are infrequent or non-minimal are removed. One of the k-subsets of such a generator is either infrequent or non-minimal and thus does not belong to the set of frequent k-generators in FC_k.
   c) The third phase removes (k+1)-generators which closures were already computed. Such a generator g is easily identified as it is included in the closure of a frequent k-generator g' in FC_k: We have $g' \subset g \subseteq \gamma(g')$.

The algorithm stops when no new candidate generator can be created. Then, each set FC_k stores the frequent k-generators, their closures and their supports.

Example 3. Figure 1 shows the execution of the Close algorithm on the context $\mathcal{D}$ for *minsupp* = 2/6. The set FC_1 is initialized with the list of all 1-itemsets. The algorithm computes supports and closures of the 1-generators in FC_1 and infrequent ones are discarded. Then, joining the frequent generators in FC_1, six new candidate 2-generators are created: {AB}, {AC}, {AE}, {BC}, {BE} and {CE} in FC_2. The 2-generators {AC} and {BE} are removed form FC_2 because we have {AC} $\subseteq$ γ({A}) and {BE} $\subseteq$ γ({B}). The algorithm determines supports and closures of the remaining 2-generators in FC_2 and suppresses infrequent ones. Then, the candidate 3-generator {ABE} is created by joining the frequent generators in FC_2 but is removed because the 2-generator {BE} $\subset$ {ABE} is not in FC_2 and the algorithm stops.

Scan $\mathcal{D}$ $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}	{AC}	3/6
{B}	{BE}	5/6
{C}	{C}	5/6
{D}	{ACD}	1/6
{E}	{BE}	5/6

Pruning infrequent itemsets $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}	{AC}	3/6
{B}	{BE}	5/6
{C}	{C}	5/6
{E}	{BE}	5/6

Scan $\mathcal{D}$ $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}	{ABCE}	2/6
{AE}	{ABCE}	2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

Pruning infrequent itemsets $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}	{ABCE}	2/6
{AE}	{ABCE}	2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

Figure 1. Extracting frequent closed itemsets in the context $\mathcal{D}$ with Close.

The A-Close algorithm improves the efficiency of the extraction in case of weakly correlated data. It does not compute closures of candidate generators during the iterations, but during an ultimate scan carried out after the end of these iterations if necessary. Experimental results show that Close and A-Close are particularly efficient for mining association rules from dense or correlated data. On such data, Close outperforms A-Close, and they both outperform algorithms for extracting frequent itemsets and maximal frequent itemsets. In that case, algorithms for extracting maximal frequent itemsets suffer from the cost of the frequent itemset supports computation that requires accessing the dataset. On the contrary, for weakly correlated data, algorithms for extracting maximal frequent itemsets are the most efficient and algorithms for extracting frequent itemsets, as well as A-Close, outperform Close.
The ChARM (Zaki and Hsiao, 1999) and Closet (Pei et al., 2000) algorithms extract frequent closed itemsets. However, none of these algorithm extract generators and can be used to generate the min-max basis for association rules. The Pascal (Bastide and al., 2000) algorithm is an optimization of Apriori based on *inference counting* and equivalence classes defined according to itemset supports. It can easily be extended to generate the min-max basis since generators and closed itemsets are respectively bottom and top patterns of an equivalence class.

2.2. Deriving frequent closed itemsets and generators from frequent itemsets

The Close$^+$ algorithm identifies frequent closed itemsets and generators among frequent itemsets without accessing the dataset. It enables the efficient generation of the min-max basis when an algorithm for extracting frequent itemsets is used. Such an algorithm gives as result the sets F_k, each set F_k containing all frequent k-itemsets, with k varying from 1 to μ (the size of the longest maximal frequent itemsets). The frequent closed itemsets and generators are identified among frequent itemsets using propositions 1 and 2 that are derived from the property that an itemset's support is equal to its closure's support. The completeness of the approach

is insured by the property that maximal frequent itemsets are maximal frequent closed itemsets (Pasquier et al., 1998).

Proposition 1. The support of a generator is smaller than the supports of all its subsets.

Proof. Let g be a k-generator and s a $(k-1)$-subsets of g. We then have $s \subset g \Rightarrow \psi(s) \supseteq \psi(g)$. If $\psi(s) = \psi(g)$ then $\gamma(s) = \gamma(g)$ and g is not a generator: It is not a minimal itemset whose closure is $\gamma(g)$. It follows that $\psi(s) \supset \psi(g) \Rightarrow supp(g) > supp(s)$.

Proposition 2. The support of a closed itemset is greater than the supports of all its supersets.

Proof. Let l be a closed k-itemset and s a superset of l. We then have $l \subset s \Rightarrow \psi(l) \supseteq \psi(s)$. If $\psi(l) = \psi(s)$ then $\gamma(l) = \gamma(s) \Rightarrow l = \gamma(s) \Rightarrow s \subseteq l$ (absurd). It follows that $\psi(l) \supset \psi(s) \Rightarrow supp(l) > supp(s)$.

The pseudo-code of the CLOSE$^+$ algorithm is given in figure 2. It examines successively all frequent itemsets in each set F_k, with k varying from 1 to μ. It generates the sets FC_m, $1 \leq m \leq \nu$, where ν is the size of the longest generators, containing the m-generators, their closures and their supports. It first determines if a frequent k-itemset is a generator by examining all its $(k-1)$-subsets' supports; it then determine if it is a closed itemset by examining all its $(k+1)$-supersets' supports and if so, identifies its generators by examining all its subsets' supports. The boolean variables *isclosed* and *isgenerator* are used to determine if an itemset l is a closed itemset or is a generator.
At the beginning of the k^{th} iteration (steps 1 to 21), the set FC_k is empty (step 2). In steps 3 to 20, frequent itemsets in F_k are considered successively. If an itemset l has the same support as one of its $(k-1)$-subset l' in F_{k-1} (steps 5 to 7), then l is not a generator (step 6). Otherwise, l and its support are inserted in FC_k (step 8). Then, we test if l has the same support as one of its $(k+1)$-superset l" in F_{k+1} (steps 10 to 12). If so, we have $l' \subseteq \gamma(l)$ and then $l \neq \gamma(l)$: l is not closed (step 11). Otherwise, l is a frequent closed itemset and we determine the generators of l (steps 13 to 19) as follows. For each generator g of size n, with $1 \leq n \leq k$, that is a subset of l (steps 14 to 18), if the supports of g and l are equal then g is a generator of l and l is inserted in FC_n as the closure of g (step 16). Thus, at the end of the algorithm, each set FC_k contains all frequent k-generators, their closures and their supports.

Correctness. The correctness of the computation of sets FC_k for $1 \leq k \leq \mu$ relies on propositions 1 and 2. Using the first one, we determine if a frequent k-itemset l is a generator of a closed itemset by comparing its support and the supports of the frequent $(k-1)$-itemsets included in l. The second proposition enables to determine if a frequent k-itemset l is closed by comparing its support and the supports of the frequent $(k+1)$-itemsets in which l is included. Since a generator has the same support as its closure, the determination of the generators of a closed itemset is correct.

Input : sets F_k of frequent k-itemsets
Output : sets FC_k of frequent k-generators, with closure and support

```
1)  for k = 1 to μ do
2)      FC_k ← ∅
3)      forall itemsets l ∈ F_k do
4)          isgenerator ← true
5)          forall subsets l' ∈ F_{k-1} of l do
6)              if (l'.supp = l.supp) then isgenerator ← false
7)          end
8)          if (isgenerator = true) then insert l in FC_k.generators with l.supp
9)          isclosed ← true
10)         forall supersets l'' ∈ F_{k+1} of l do
11)             if (l''.supp = l.supp) then isclosed ← false
12)         end
13)         if (isclosed = true) then do
14)             for n = k to 0 step −1 do
15)                 forall subsets g ∈ FC_n.generators of l do
16)                     if (g.supp = l.supp) then insert l in g.closure
17)                 end
18)             end
19)         end
20)     end
21) end
22) return ⋃ FC_k
```

Figure 2. CLOSE$^+$ algorithm for deriving frequent closed itemsets and generators.

Example 4. Figure 3 shows the execution of the CLOSE$^+$ algorithm using the sets F_1 to F_4 of frequent itemsets extracted from the context $\mathcal{D}$ with $minsupp = 2/6$. All frequent 1-itemsets are frequent 1-generators since none of their subsets is a frequent itemset: The empty set is not considered as a frequent itemset. The 1-itemset {C} is also its own closure since all its supersets in F_2 have a smaller support. In F_2, the 2-itemsets {AC} and {BE} are not generators since they have the same support as itemsets {A} and, {B} and {E} respectively. These two itemsets are closed since their support is lower than those of all their supersets in F_3; {AC} is the closure of {A} and {BE} is the closure of {B} and {E}. No frequent 3-itemset in F_3 is a generator and {BCE}, that has the same support as {BC} and {CE} and a greater support than {ABCE} in FC_4, is the closure of {BC} and {CE} in FC_2. Finally, the 4-itemset {ABCE} is closed since it is a maximal frequent itemset is the closure of {AB} and {AE}, and is inserted in FC_2.

Remark. As a simple optimization, the algorithm can stop testing if frequent k-itemsets are generators after the first iteration n during which no frequent n-itemset examined is a generator. In example 4, the algorithm will not test if 4-itemsets in F_4 are generators since no 3-itemset is a generator (FC_3 is empty at the end of the third iteration).

F_1

Itemset	Supp
{A}	3/6
{B}	5/6
{C}	5/6
{E}	5/6

Generators of size 1 $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}		3/6
{B}		5/6
{C}		5/6
{E}		5/6

Closures of size 1 $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}		3/6
{B}		5/6
{C}	{C}	5/6
{E}		5/6

F_2

Itemset	Supp
{AB}	2/6
{AC}	3/6
{AE}	2/6
{BC}	4/6
{BE}	5/6
{CE}	4/6

Generators of size 2 $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}		2/6
{AE}		2/6
{BC}		4/6
{CE}		4/6

Closures of size 2 $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}	{AC}	3/6
{B}	{BE}	5/6
{C}	{C}	5/6
{E}	{BE}	5/6

F_3

Itemset	Supp
{ABC}	2/6
{ABE}	2/6
{ACE}	2/6
{BCE}	4/6

Generators of size 3 $\longrightarrow$

FC_3

Generator	Closed itemset	Supp

Closures of size 3 $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}		2/6
{AE}		2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

F_4

Itemset	Supp
{ABCE}	2/6

Generators of size 4 $\longrightarrow$

FC_4

Generator	Closed itemset	Supp

Closures of size 4 $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}	{ABCE}	2/6
{AE}	{ABCE}	2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

Figure 3. Deriving frequent closed itemsets and generators with CLOSE$^+$.

3. Min-max basis for association rules

We first define min-max association rules: The most general non-redundant association rules according to their semantic. Informally, an association rule is redundant if it brings the same information or less information than is brought by another rule of same support and confidence. Then, the min-max association rules are the non-redundant association rules having minimal antecedent and maximal consequent: r is a min-max association rule if no other association rule r' has the same support and confidence, an antecedent that is a subset of the antecedent of r and a consequent that is a superset of the consequent of r.

Definition 5. (Min-max association rules) Let $\mathcal{AR}$ be the set of association rules extracted. An association rule $r: l_1 \rightarrow l_2 \in \mathcal{AR}$ is a min-max association rule iff $\nexists\, r'\colon l'_1 \rightarrow l'_2 \in \mathcal{AR}$ with $supp(r') = supp(r)$, $conf(r') = conf(r)$, $l'_1 \subseteq l_1$ and $l_2 \subseteq l'_2$.

Based on this definition, we characterize exact and approximate min-max association rules that constitute respectively the *min-max exact basis* and the *min-max approximate basis* in the two following sections.

3.1. Exact min-max association rules

First, notice that exact association rules, with the form $r: l_1 \Rightarrow (l_2 \setminus l_1)$, are rules between two frequent itemsets $l_1 \subset l_2$ having the same closure: $\gamma(l_1) = \gamma(l_2)$. Since $conf(r) = 1$ we have $supp(l_1) = supp(l_2)$, and as $l_1 \subset l_2$ we see that $\gamma(l_1) = \gamma(l_2)$. We define min-max association rules among these exact rules.
Let g be the generator of $\gamma(l_1) = \gamma(l_2)$ such that $g \subseteq l_1$. Since g is minimal, we have $g \subseteq l_1 \subset l_2 \subseteq \gamma(l_2)$. Furthermore, all itemsets in the interval $[g, \gamma(l_2)]$, defined by inclusion[4], have the same closure $\gamma(l_2)$ and thus the same support. The min-max association rule among all rules with the form $r: l_1 \Rightarrow (l_2 \setminus l_1)$ with $l_1, l_2 \in [g, \gamma(l_2)]$ is the rule $g \Rightarrow (\gamma(l_2) \setminus g)$. This rule has a minimal antecedent, g, and a maximal consequent, $\gamma(l_2)$, among all these rules that have the same support.
We generalize this definition to all generators of the frequent closed itemset $\gamma(l_2)$. Let $Gen_{\gamma(l_2)}$ be the set of these generators. All exact min-max association rules constructed with $\gamma(l_2)$ are rules with the form $g \Rightarrow (\gamma(l_2) \setminus g)$ with $g \in Gen_{\gamma(l_2)}$. The extension of this property to all frequent closed itemsets defines the min-max exact basis containing all exact min-max association rules characterized in definition 5.

Definition 6. (Min-max exact basis) Let *Closed* be the set of frequent closed itemsets extracted from the context and, for each frequent closed itemset f, let's denote Gen_f the set of generators of f. The min-max exact basis is:

$$MinMaxExact = \{r\colon g \Rightarrow (f \setminus g) \mid f \in Closed \ \wedge \ g \in Gen_f \ \wedge \ g \neq f\}.$$

The condition $g \neq f$ discards rules with the form $g \Rightarrow \varnothing$; it is equivalent to the condition $l_1 \subset l_2$ in the definition of association rules. We state in the following proposition that the min-max exact basis does not lead to information loss.

[4] The interval $[l_1, l_2]$ contains all the supersets of l_1 that are subsets of l_2.

The pseudo-code of the algorithm for constructing the min-max exact basis using frequent closed itemsets and their generators is presented in figure 4. Each element of a set FC_k contains three fields: a k-generator *generator*, its closure *closure* and their support *supp*. The algorithm returns the set *MinMaxExact* containing the exact min-max rules.

Input : sets FC_k
Output : set *MinMaxExact*

1) $MinMaxExact \leftarrow \varnothing$
2) **for** $k = 1$ **to** ν **do**
3) **forall** k-generator $g \in FC_k$ **do**
4) **if** $(g \neq g.closure)$
5) **then insert** $\{r: g \Rightarrow (g.closure \setminus g), g.supp\}$ **in** *MinMaxExact*
6) **end**
7) **end**
8) **return** *MinMaxExact*

Figure 4. Algorithm for generating the min-max exact basis.

First, *MinMaxExact* is initialized with the empty set (step 1). Then, each set FC_k is examined in increasing order of k values (steps 2 to 7). For each k-generator $g \in FC_k$ of the frequent closed itemset $\gamma(g)$ (steps 3 to 6), if g is different from its closure $\gamma(g)$ (step 4), the rule $r: g \Rightarrow (\gamma(g) \setminus g)$, which support is equal to the support of g and $\gamma(g)$, is inserted into *MinMaxExact* (step 5). Finally, the algorithm returns the set *MinMaxExact* containing all exact min-max association rules between generators and their closures (step 8).

Example 5. The min-max exact basis extracted from context $\mathcal{D}$ for $minsupp = 2/6$ is presented in table III. It contains seven rules whereas the set of all exact association rules, presented in table IV, contains fourteen rules.

Table III. Min-max exact basis extracted from $\mathcal{D}$.

Generator	Closure	Exact rule	Supp
{A}	{AC}	A ⇒ C	3/6
{B}	{BE}	B ⇒ E	5/6
{C}	{C}		
{E}	{BE}	E ⇒ B	5/6
{AB}	{ABCE}	AB ⇒ CE	2/6
{AE}	{ABCE}	AE ⇒ BC	2/6
{BC}	{BCE}	BC ⇒ E	4/6
{CE}	{BCE}	CE ⇒ B	4/6

Table IV. Exact association rules extracted from $\mathcal{D}$.

Exact rule	Supp	Exact rule	Supp
A ⇒ C	3/6	BC ⇒ E	4/6
B ⇒ E	5/6	CE ⇒ B	4/6
E ⇒ B	5/6	AB ⇒ CE	2/6
AB ⇒ C	2/6	AE ⇒ BC	2/6
AB ⇒ E	2/6	ABC ⇒ E	2/6
AE ⇒ B	2/6	ABE ⇒ C	2/6
AE ⇒ C	2/6	ACE ⇒ B	2/6

Proposition 3. (i) All exact association rules and their supports can be deduced from the min-max exact basis. *(ii)* All rules in the min-max exact basis are min-max association rules.

Proof. (i) Let $r : l_1 \Rightarrow (l_2 \setminus l_1)$ be an exact association rule between two frequent itemsets with $l_1 \subset l_2$. Since $conf(r) = 1$, we have $supp(l_1) = supp(l_2)$ and as an itemset's support is equal to its closure's support, we deduce that $supp(\gamma(l_1)) = supp(\gamma(l_2))$ which implies that $\gamma(l_1) = \gamma(l_2) = f$. The itemset f is a frequent closed itemset $f \in FC$ and, obviously, there exists a rule $r' : g \Rightarrow (f \setminus g) \in MinMaxExact$ such that g is a generator of f with $g \subseteq l_1$ and $g \subset l_2$. We show now that the rule r and its support can be deduced from the rule r' and its support. Since $g \subseteq l_1 \subset l_2 \subseteq f$, rule r's antecedent and consequent can be derived from those of rule r'. From $\gamma(l_1) = \gamma(l_2) = f$, we deduce that $supp(r) = supp(l_2) = supp(\gamma(l_2)) = supp(f) = supp(r')$.
(ii) Let $r : g \Rightarrow (f \setminus g) \in MinMaxExact$. According to definition 6, we have $g \in Gen_f$ and $f \in Closed$. We demonstrate that there is no other rule $r' : l'_1 \Rightarrow (l'_2 \setminus l'_1) \in MinMaxExact$, such as $supp(r') = supp(r)$, $conf(r') = conf(r)$, $l'_1 \subseteq g$ and $f \subseteq l'_2$. If $l'_1 \subset g$ then, according to definition 4, we have $\gamma(l'_1) \subset \gamma(g) = f \Longrightarrow l_1 \notin Gen_f$ and then $r' \notin MinMaxExact$. If $f \subset l'_2$ and according to definition 3, we have $f = \gamma(f) = \gamma(g) \subset l'_2 = \gamma(l'_2)$. From definition 4 we deduce $g \notin Gen_{l'_2}$ and we conclude that $r' \notin MinMaxExact$.

3.2. Approximate min-max association rules

Approximate association rules, with the form $r : l_1 \rightarrow (l_2 \setminus l_1)$, are rules between two frequent itemsets $l_1 \subset l_2$ such that $\gamma(l_1) \subset \gamma(l_2)$. Since $conf(r) < 1$ we have $supp(l_1) > supp(l_2)$ and we deduce that $\gamma(l_1) \subset \gamma(l_2)$.
We deduce the definition of approximate min-max association rules. Let g_1 be a generator of the frequent closed itemset f_1 and g_2 be a generator of the frequent closed itemset f_2 such that $f_1 \subset g_2 \subseteq l_2 \subseteq f_2$. All rules with the form $r : l_1 \rightarrow (l_2 \setminus l_1)$ where $l_1 \in [g_1, f_1]$ and $l_2 \in [g_2, f_2]$ have the same confidence and the same support since g_1, l_1 and f_1 have the same support as well as g_2, l_2 and f_2. We then deduce that the min-max association rule among all these rules is $g_1 \rightarrow (f_2 \setminus g_1)$. Indeed, g_1 is the minimal itemset in $[g_1, f_1]$ and f_2 is the maximal itemset in $[g_2, f_2]$.

The generalization of this property to all couples of frequent itemsets l_1 and l_2 such that $l_1 \subset l_2$ and $supp(l_1) \neq supp(l_2)$ defines the min-max approximate basis containing all approximate min-max association rules characterized in definition 5.

Definition 7. (Min-max approximate basis) We denote *Gen* the set of generators of the frequent closed itemsets in *Closed*. The min-max approximate basis is:

$$MinMaxApprox = \{r : g \rightarrow (f \setminus g) \mid f \in Closed \ \wedge \ g \in Gen \ \wedge \ \gamma(g) \subset f\}.$$

The pseudo code of the algorithm for generating the set *MinMaxApprox* of approximate min-max rules using frequent closed itemsets and their generators is presented in figure 5.

Input : sets FC_k, confidence threshold *minconf*
Output : set *MinMaxApprox*

```
1)  MinMaxApprox ← ∅
2)  for k = 1 to ν − 1 do
3)      forall k-generator g ∈ FC_k do
4)          forall frequent closed itemset f ∈ F_{j>k} | f ⊃ g.closure do
5)              if (f.supp/g.supp ≥ minconf)
6)              then insert {r: g → (f \ g), f.supp/g.supp, f.supp} in MinMaxApprox
7)          end
8)      end
9)  end
10) return MinMaxApprox
```

Figure 5. Algorithm for generating the min-max approximate basis.

The algorithm examines the sets FC_k in increasing order of k values (steps 2 to 9). For each k-generator $g \in FC_k$ (steps 3 to 8), it considers all closed supersets f of the closure of g (steps 4 to 7). It computes the confidence of the rule $r : g \rightarrow (f \setminus g)$ (step 5) and inserts r in *MinMaxReduc* if it is above the *minconf* threshold (step 6).

Example 6. The min-max approximate basis extracted from context $\mathcal{D}$ for *minsupp* = 2/6 and *minconf* = 2/5 is presented in table V. It contains ten rules whereas the set of all approximate association rules, presented in table VI, contains thirty-six rules.

Proposition 4. (i) All approximate association rules can be deduced, with their supports and confidences, from the min-max approximate basis. *(ii)* All rules in the min-max approximate basis are min-max association rules.

Proof. (i) Let $r : l_1 \rightarrow (l_2 \setminus l_1)$ be an association rule between two frequent itemsets with $l_1 \subset l_2$. Since $conf(r) < 1$ we also have $\gamma(l_1) \subset \gamma(l_2)$. For any frequent itemsets l_1 and l_2, there is a generator g_1 such that $g_1 \subset l_1 \subseteq \gamma(l_1) = \gamma(g_1)$ and a generator g_2 such that $g_2 \subset l_2 \subseteq \gamma(l_2) = \gamma(g_2)$. Since $l_1 \subset l_2$, we have $l_1 \subseteq \gamma(g_1) \subset l_2 \subseteq \gamma(g_2)$ and the rule $r' : g_1 \rightarrow (\gamma(g_2) \setminus g_1)$ is in the min-max approximate basis. We show

Table V. Min-max approximate basis extracted from $\mathcal{D}$.

Generator	Closure	Closed superset	Approximate rule	Supp	Conf
{A}	{AC}	{ABCE}	A → BCE	2/6	2/3
{B}	{BE}	{BCE}	B → CE	4/6	4/5
{B}	{BE}	{ABCE}	B → ACE	2/6	2/5
{C}	{C}	{AC}	C → A	3/6	3/5
{C}	{C}	{BCE}	C → BE	4/6	4/5
{C}	{C}	{ABCE}	C → ABE	2/6	2/5
{E}	{BE}	{BCE}	E → BC	4/6	4/5
{E}	{BE}	{ABCE}	E → ABC	2/6	2/5
{AB}	{ABCE}				
{AE}	{ABCE}				
{BC}	{BCE}	{ABCE}	BC → AE	2/6	2/4
{CE}	{BCE}	{ABCE}	CE → AB	2/6	2/4

Table VI. Approximate association rules extracted from $\mathcal{D}$.

Approximate rule	Supp	Conf	Approximate rule	Supp	Conf	Approximate rule	Supp	Conf
BCE → A	2/6	2/4	B → ACE	2/6	2/5	B → CE	4/6	4/5
AC → BE	2/6	2/3	C → ABE	2/6	2/5	C → BE	4/6	4/5
BC → AE	2/6	2/4	E → ABC	2/6	2/5	E → BC	4/6	4/5
BE → AC	2/6	2/5	A → BC	2/6	2/3	A → B	2/6	2/3
CE → AB	2/6	2/4	B → AC	2/6	2/5	B → A	2/6	2/5
AC → B	2/6	2/3	C → AB	2/6	2/5	C → A	3/6	3/5
BC → A	2/6	2/4	A → BE	2/6	2/3	A → E	2/6	2/3
BE → A	2/6	2/5	B → AE	2/6	2/5	E → A	2/6	2/5
AC → E	2/6	2/3	E → AB	2/6	2/5	B → C	4/6	4/5
CE → A	2/6	2/4	A → CE	2/6	2/3	C → B	4/6	4/5
BE → C	4/6	4/5	C → AE	2/6	2/5	C → E	4/6	4/5
A → BCE	2/6	2/3	E → AC	2/6	2/5	E → C	4/6	4/5

that the rule r, its support and its confidence can be deduced from the rule r', its support and its confidence. Since $g_1 \subset l_1 \subseteq \gamma(g_1) \subset g_2 \subset l_2 \subseteq \gamma(g_2)$, the antecedent and the consequent of r can be rebuilt starting from the rule r'. Moreover, we have $\gamma(l_2) = \gamma(g_2)$ and thus $supp(r) = supp(l_2) = supp(\gamma(g_2)) = supp(r')$. Since $g_1 \subset l_1 \subseteq \gamma(g_1)$, we have $supp(g_1) = supp(l_1)$ and we thus deduce that: $conf(r) = supp(l_1) \,/\, supp(l_2) = supp(g_1) \,/\, supp(\gamma(g_2)) = conf(r')$.

(ii) Let $r : g \Rightarrow (f \setminus g) \in \mathit{MinMaxExact}$. According to definition 7, we have $f \in \mathit{Closed}$, $g \in \mathit{Gen}_{f'}$ and $f' \subset f$. We demonstrate that there is no other rule $r' : l'_1 \Rightarrow (l'_2 \setminus l'_1) \in \mathit{MinMaxApprox}$, such as $supp(r') = supp(r)$, $conf(r') = conf(r)$, $l'_1 \subseteq g$ and $f \subseteq l'_2$. If $l'_1 \subset g$ then, according to definition 4, we have $\gamma(l'_1) \subset \gamma(g) = f'$ and then $l_1 \notin \mathit{Gen}_{f'}$. We deduce that $supp(l'_1) > supp(g)$ and then $conf(r') < conf(r)$. If $f \subset l'_2$ then, according to definition 3, we have $f = \gamma(f) \subset l'_2 = \gamma(l'_2)$. We deduce that $supp(f) > supp(l'_2)$ and we conclude that $conf(r) > conf(r')$.

3.3. Non-transitive approximate min-max association rules

We can further reduce the number of approximate association rules extracted without losing the ability to deduce all approximate association rules, with support and confidence, by removing *transitive min-max association rules.*
A min-max association rules $g \rightarrow (f \setminus g)$ with $\gamma(g) \subset f$ is transitive if it exists a frequent closed itemset f' such that $\gamma(g) \subset f' \subset f$. Let g' be the generator of f' such that $\gamma(g) \subset g' \subseteq f' \subset f$. Then, we have the two following approximate min-max association rules: $g \rightarrow (f' \setminus g)$ and $g' \rightarrow (f \setminus g')$. The rule $g \rightarrow (f \setminus g)$ is the transitive composition of the two previous rules; its support is equal to the second rule's support and its confidence is equal to the product of their confidences.
We generalize this characterization to all triplets consisting of a generators g, its closure f and a closed superset f' of f to define the *non-transitive min-max approximate basis*, that is the transitive reduction of the min-max approximate basis. Let's denote $l_1 \lessdot l_2$ when an itemset l_1 is an immediate predecessor of an itemset l_2, i.e. $\nexists l_3$ such that $l_1 \subset l_3 \subset l_2$. The non-transitive min-max approximate rules are of the form $g \rightarrow (f \setminus g)$ where f is a frequent closed itemset and g a frequent generator such that $\gamma(g)$ is an immediate predecessor of f.

Definition 8. (Non-transitive min-max approximate basis) The non-transitive min-max approximate basis is:

$$\mathit{MinMaxReduc} = \{r : g \rightarrow (f \setminus g) \mid f \in \mathit{Closed} \ \wedge \ g \in \mathit{Gen} \ \wedge \ \gamma(g) \lessdot f\}.$$

Remark. This transitive reduction decreases the number of approximate rules extracted, by selecting the most precise rules, i.e. whith highest confidences, since transitive rules have lower confidences than non-transitive rules.

The algorithm presented in figure 6 constructs the set *MinMaxReduc* of non-transitive approximate min-max rules using frequent closed itemsets and their generators. For each generator g, it determines all frequent closed itemsets f that are immediate successors of the closure of g and then, it generates all rules between g and f that have a sufficient confidence.
First, *MinMaxReduc* is initialized with the empty set (step 1) and sets FC_k are successively examined in increasing order of k values (steps 2 to 19). For each k-generator $g \in FC_k$ (steps 3 to 18), the set ImSucc_g of immediate successors of g closure is initialized with the empty set (step 4). The sets S_j of frequent closed j-supersets of $\gamma(g)$ for $|\gamma(g)| < j \leq \mu$ are constructed (steps 5 to 7). Then, sets S_j are considered successively in ascending order of j values (steps 8 to 17). For each itemset $f \in S_j$ that is not a superset of an immediate successor of $\gamma(g)$ in ImSucc_g (step 10), f is inserted in ImSucc_g (step 11) and the confidence of the rule $r : g \rightarrow (f \setminus g)$ is computed (step 12). If the confidence of r is above *minconf*, the rule r is inserted in *MinMaxReduc* (steps 13 and 14). When all the generators of size lower than $\nu - 1$ have been considered, the algorithm returns the set *MinMaxReduc* (step 20).

Example 7. The non-redundant min-max approximate basis extracted from context $\mathcal{D}$ for *minsupp* $= 2/6$ and *minconf* $= 2/5$ is presented in table VII. It contains

Input : sets FC_k, confidence threshold *minconf*
Output : set *MinMaxReduc*

```
1)  MinMaxReduc ← ∅
2)  for k = 1 to ν − 1 do
3)      forall k-generator g ∈ FC_k do
4)          ImSucc_g ← ∅
5)          for j = |g.closure| to μ do
6)              S_j ← {f ∈ FC.closure | f ⊃ g.closure ∧ |f| = j}
7)          end
8)          for j = |g.closure| to μ do
9)              forall frequent closed itemset f ∈ S_j do
10)                 if (∄s ∈ ImSucc_g | s ⊂ f) then do
11)                     insert f in ImSucc_g
12)                     conf ← f.supp/g.supp
13)                     if (conf ≥ minconf)
14)                     then insert {r: g → (f \ g), conf, f.supp} in MinMaxReduc
15)                 end
16)             end
17)         end
18)     end
19) end
20) return MinMaxReduc
```

Figure 6. Algorithm for generating the non-transitive min-max approximate basis.

only seven rules, that is three rules less than the approximate min-max basis. These three rules are B → ACE, C → BE and E → ABC that have minimal support and confidence measures among the ten rules of the approximate min-max basis.

Table VII. Non-transitive min-max approximate basis extracted from $\mathcal{D}$.

Generator	Closure	Closed superset	Approximate rule	Supp	Conf
{A}	{AC}	{ABCE}	A → BCE	2/6	2/3
{B}	{BE}	{BCE}	B → CE	4/6	4/5
{B}	{BE}	{ABCE}			
{C}	{C}	{AC}	C → A	3/6	3/5
{C}	{C}	{BCE}	C → BE	4/6	4/5
{C}	{C}	{ABCE}			
{E}	{BE}	{BCE}	E → BC	4/6	4/5
{E}	{BE}	{ABCE}			
{AB}	{ABCE}				
{AE}	{ABCE}				
{BC}	{BCE}	{ABCE}	BC → AE	2/6	2/4
{CE}	{BCE}	{ABCE}	CE → AB	2/6	2/4

Proposition 5. All approximate association rules, with support and confidence, can be deduced from the non-transitive min-max approximate basis.

First, we show that all approximate min-max association rules can be derived from the non-transitive min-max approximate association rules. Then, from proposition 4 we conclude that all approximate association rules can also be deduced.

Proof. Let $r: g_1 \rightarrow (f_n \setminus g_1)$ be an approximate min-max association rule between a generator g_1 whose closure is f_1 and a frequent closed superset f_n of f_1. If $f_1 \lessdot f_n$ then r is non-transitive: $r \in \mathit{MinMaxReduc}$. If $f_1 \not\lessdot f_n$ then r is transitive and there is a sequence f_1, f_2, ..., f_n of frequent closed itemsets such that $g_1 \subseteq f_1 \lessdot f_2 \lessdot \ldots \lessdot f_n$ with $n \geq 3$. Each f_i has at least one generator g_i such that $\gamma(g_i) = f_i$ and since $f_1 \lessdot f_2 \lessdot \ldots \lessdot f_n$, there is a sequence of rules $r_i: g_i \rightarrow (f_{i+1} \setminus g_i)$ for $i \in [1, n-1]$ that are non-transitive min-max rules. The antecedent of r is the antecedent g_1 of the first rule r_1 of the sequence. The consequent of r is $(f_n \setminus g_1) = (((f_n \setminus g_{n-1}) \cup g_{n-1}) \setminus g_1)$, i.e. the union of rule r_{n-1}'s antecedent and consequent minus rule r_1's antecedent. We now show that support and confidence of r can be deduced of those of rules r_i. We have $\mathit{supp}(r) = \mathit{supp}(g_1 \cup (f_n \setminus g_1)) = \mathit{supp}(f_n) = \mathit{supp}(g_{n-1} \cup (f_n \setminus g_{n-1})) = \mathit{supp}(r_{n-1})$. The support of r is equal to the support of the last rule r_{n-1} of the sequence. We also have: $\mathit{conf}(r) = \mathit{supp}(f_n)/\mathit{supp}(g_1) = \mathit{supp}(f_n)/\mathit{supp}(g_{n-1}) \times \mathit{supp}(g_{n-1})/\mathit{supp}(g_1)$ $= \mathit{supp}(f_n)/\mathit{supp}(g_{n-1}) \times \mathit{supp}(f_{n-1})/\mathit{supp}(g_{n-2}) \times \ldots \times \mathit{supp}(f_2)/\mathit{supp}(g_1) =$ $\mathit{conf}(r_{n-1}) \times \mathit{conf}(r_{n-2}) \times \ldots \times \mathit{conf}(r_1)$. The confidence of r is equal to the product of the confidences of the rules r_i for $i = 1$ to $n - 1$.

4. Deriving association rules from the min-max bases

We introduce in this section simple techniques and algorithms to reconstruct all exact association rules, all approximate association rules and all transitive approximate min-max association rules from the min-max bases.

4.1. Deriving exact association rules

The graph-oriented representation of the exact and the exact min-max association rules extracted from context $\mathcal{D}$ for $\mathit{minsupp} = 2/6$ and $\mathit{minconf} = 2/5$ are given in figure 7 and 8 respectively.
Each vertex v_l represents a frequent itemset l that is a subset of the maximal frequent itemset {ABCE}. Each edge between two vertices v_a and v_c represents the exact association rule $a \Rightarrow c \setminus a$. A *closed interval* is a sub-graph containing all vertices representing itemsets of the intervals $[g_i, f]$ where each g_i is a generator of the frequent closed itemset f. Since all itemsets in a closed interval have the same support, all rules in this interval also have the same support.
In the graph representation, deriving all exact rules means adding all possible edges between two vertices of the same closed interval. Each edge in figure 8 between two vertices v_g and v_f represents a rule between a generator g and its closure f. Then,

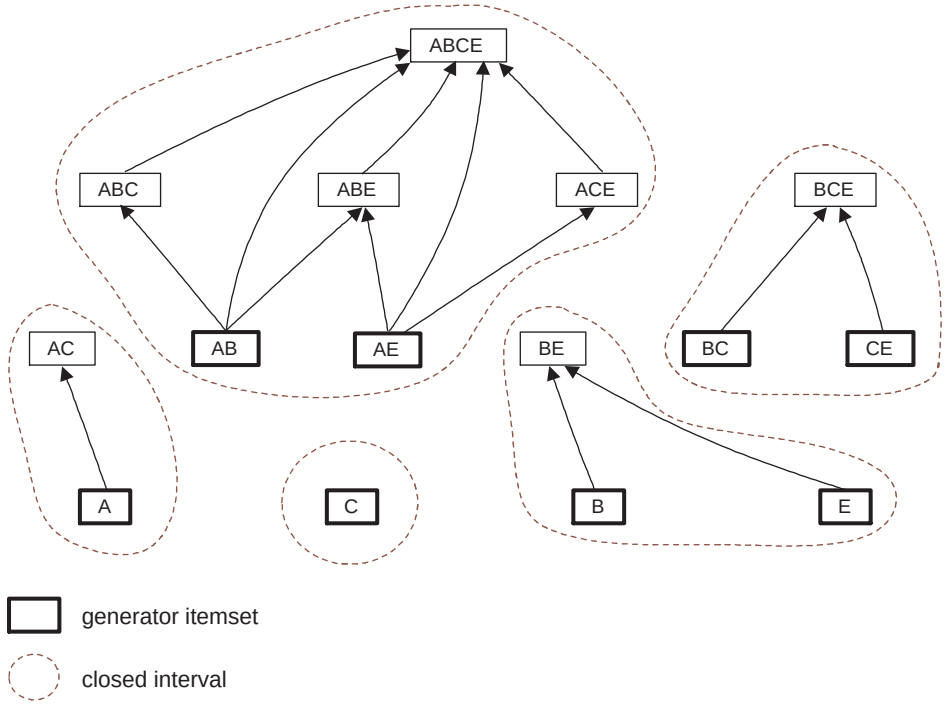

Figure 7. Exact association rules extracted from $\mathcal{D}$.

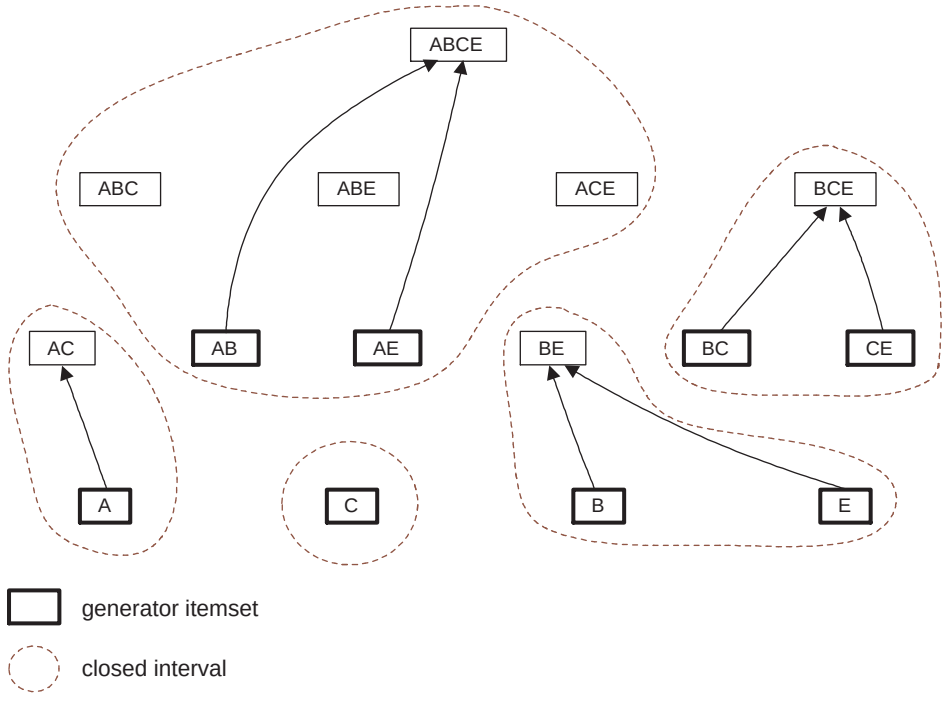

Figure 8. Exact min-max association rules extracted from $\mathcal{D}$.

we add all edges between two vertices, one representing a superset of g and the other a subset of f.

The algorithm receives the set *MinMaxExact* of exact min-max rules as input and it returns the set *AllExact* containing all exact association rules. Its pseudo-code is presented in figure 9. It considers all exact min-max rules $r_1 : a_1 \Rightarrow c_1$ with $|c_1| > 1$ (steps 2 to 8). For all subset c_2 of c_1 (steps 3 to 7), it generates all rules with the form $r_2 : a_1 \Rightarrow c_2$ and $r_3 : a_1 \cup c_2 \Rightarrow c_1 \setminus c_2$ (steps 4 and 6). These rules have the same support as r_1. Since rule r_3 can be generated several times, the algorithm first tests if it has not already been inserted in *AllExact* (step 5).

Input : set *MinMaxExact*
Output : set *AllExact*

1) *AllExact* $\leftarrow \varnothing$
2) **forall** rule $\{r_1 : a_1 \Rightarrow c_1, r_1.supp\} \in$ *MinMaxExact* **with** $|c_1| > 1$ **do**
3) **forall** subset $c_2 \subset c_1$ **do**
4) **insert** $\{r_2 : a_1 \Rightarrow c_2, r_1.supp\}$ **in** *AllExact*
5) **if** $\{r_3 : a_1 \cup c_2 \Rightarrow c_1 \setminus c_2, r_1.supp\} \notin$ *AllExact*
6) **then insert** r_3 **in** *AllExact*
7) **end**
8) **end**
9) **return** *AllExact*

Figure 9. Algorithm for reconstructing all exact association rules.

Example 8. Consider rule AB ⇒ CE represented in figure 4 by the edge between vertices {AB} and {ABCE}. From this rule we deduce rules AB ⇒ C, AB ⇒ E, ABC ⇒ E and ABE ⇒ C and from rule AE ⇒ BC, we deduce rules AE ⇒ B, AE ⇒ C, ABE ⇒ C and ACE ⇒ B. All these rules have the same support.

Remark. For constructing all exact rules using sets FC_k of generators and frequent closed itemsets, we consider each generator g and its closure f. We generate all rules $r : g \Rightarrow l \setminus g$ and $r : l \Rightarrow f \setminus l$ for $l \in [g, f[$. For instance, from the generator {AB} and its closure {ABCE}, we generate rules AB ⇒ CE, AB ⇒ C, AB ⇒ E, ABC ⇒ E and ABE ⇒ C. Their support is equal to the support of g and f, i.e. the support of {AB} and {ABCE}.

4.2. Deriving approximate association rules

Figures 10 and 11 depict the graph-oriented representations of the approximate and the approximate min-max association rules extracted from context $\mathcal{D}$ for *minsupp* = 2/6 and *minconf* = 2/5. Each edge between two vertices v_a and v_c represents the approximate rule $a \rightarrow c \setminus a$.
In figure 11, each edge between two vertices v_g and v_f represents the min-max approximate rule $g \rightarrow f \setminus g$ where g is a generator and f a frequent closed superset of g. That is to say an edge between a minimal vertex of a closed interval and the maximal vertex of another closed interval above the first one. For instance, the edge between vertices containing {A} and {ABCE} represents the rule A → BCE.

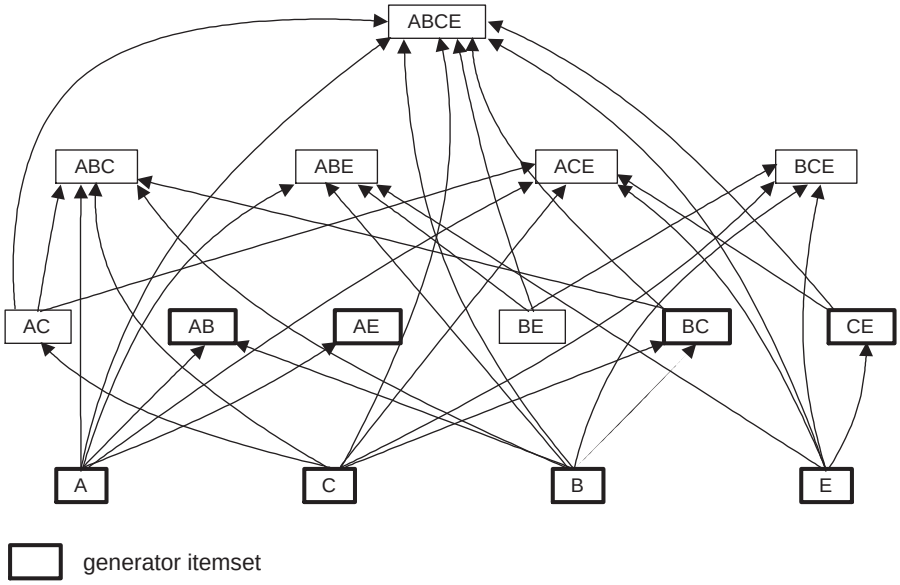

Figure 10. Approximate association rules extracted from $\mathcal{D}$.

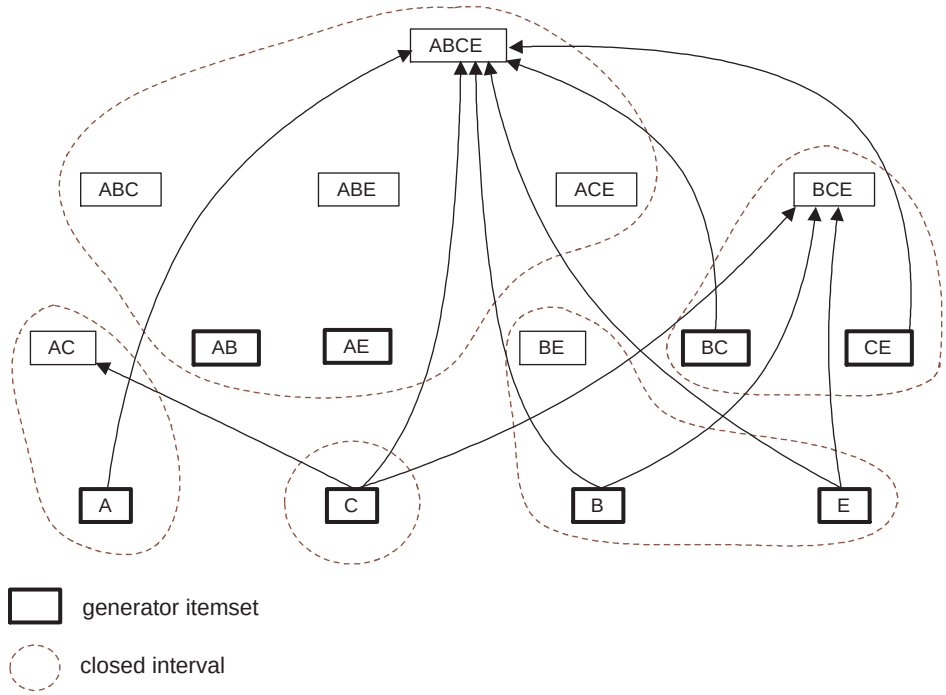

Figure 11. Approximate min-max association rules extracted from $\mathcal{D}$.

To derive all approximate rules, when there is an edge between two vertices of two closed intervals we create all possible edges between each vertex of the first interval and each vertex of the second interval. All these rules have the same support and confidence. In figure 11 for instance, we add all edges between vertices of the closed

interval {{A},{AC}} and the closed interval {{AB}, {AE}, {ABC}, {ABE}, {ACE}, {ABCE}}. These rules have the same support and confidence as rule A → BCE.
A simple and efficient method to derive all approximate rules is to proceed in two phases. First, we generate all rules with the form $g_1 \rightarrow l_i \setminus g_1$ between a generator g_1 and all its frequent supersets $l_i \in [g_i, f_i]$ where g_i is a generator of f_i and $g_1 \subset g_i$. Second, we "extend" these rules by replacing their antecedent by all itemsets $l_1 \in [g_1, f_1]$ where f_1 is the closure of g_1.
The input of the algorithm are the sets *MinMaxApprox* and *MinMaxExact* of approximate and exact min-max rules. Its result is the set *AllApprox* containing all approximate rules. Its pseudo-code is presented in figure 12.

Input : set *MinMaxApprox*, set *MinMaxExact*
Output : set *AllApprox*

1) *AllApprox* ← *MinMaxApprox*
2) **for** $i = 2$ **to** $\mu - 1$ **do**
3) **forall** rule $\{r_1 : a_1 \rightarrow c_1, r_1.supp, r_1.conf\} \in$ *MinMaxApprox* **with** $|c_1| = i$ **do**
4) **forall** subset $c_2 \subset c_1$ **do**
5) **if** $(\{r_2 : a_1 \rightarrow c_2, r_2.supp, r_2.conf\} \notin$ *AllApprox*)
6) **and** $(\{r_3 : a_1 \Rightarrow c_2, r_3.supp\} \notin$ *MinMaxExact*)
7) **then insert** $\{r_2 : a_1 \rightarrow c_2, r_1.supp, r_1.conf\}$ **in** *AllApprox*
8) **end**
9) **end**
10) **end**
11) **forall** rule $\{r_1 : a_1 \rightarrow c_1, r_1.supp, r_1.conf\} \in$ *AllApprox* **do**
12) **forall** rule $(\{r_2 : a_1 \Rightarrow c_2, r_2.supp\} \in$ *MinMaxExact*) **do**
13) **forall** subset $c_3 \subseteq c_2$ **do**
14) **insert** $\{r_3 : a_1 \cup c_3 \rightarrow c_1 \setminus c_3, r_1.supp, r_1.conf\}$ **in** *AllApprox*
15) **end**
16) **end**
17) **end**
18) **return** *AllApprox*

Figure 12. Algorithm for reconstructing approximate min-max association rules.

In the first phase (steps 2 to 10), it considers min-max approximate rules $a_1 \rightarrow c_1$ with $|c_1| > 1$ in increasing order of their consequent's size (steps 3 to 9). For each min-max rule $a_1 \rightarrow c_1$, all rules with the form $a_1 \rightarrow c_2$ with $c_2 \subset c_1$ are generated if they were not previously generated and there is no exact rule $a_1 \Rightarrow c_2$ (steps 4 to 8). All these rules have the same support and confidence. In the second phase(steps 11 to 17), it considers all approximate rules $a_1 \rightarrow c_1$ and for each min-max exact rule $a_1 \Rightarrow c_2$ (steps 12 to 16), it generates all rules with the form $a_1 \cup c_3 \rightarrow c_1 \setminus c_3$ for all subset c_3 of c_2 (steps 13 to 15).

Example 9. Considering rule A → BCE in figure 11, we deduce rules A → B, A → E, A → BC, A → BE, A → CE. Rule A → C is not generated since A ⇒ C is an exact rule, i.e. {A} and {AC} belong to the same closed interval. Then, since we have

A $\Rightarrow$ C, extending all rules with A as antecedent we obtain rules AC $\rightarrow$ B, AC $\rightarrow$ E, AC $\rightarrow$ BE.

In order to generate all approximate rules using sets FC_k of generators and frequent closed itemsets, we consider each couple of intervals $\{[g_1, f_1], [g_2, f_2]\}$ with $\gamma(g_1) = f_1$ and $\gamma(g_2) = f_2$ such that $g_1 \subset g_2$. We generate all rules $r\colon l_1 \rightarrow l_2 \setminus l_1$ for $l_1 \in [g_1, f_1]$ and $l_2 \in [g_2, f_2]$. The support of these rules is $supp(f_2)$ and their confidence is $supp(f_2)/supp(f_1)$. For instance, from the generator {B} and its closure {BE} and the generator {BC} and its closure {BCE}, we generate the rules B $\rightarrow$ C, B $\rightarrow$ CE and BE $\rightarrow$ C.

4.3. Deriving transitive approximate min-max association rules

The graph-oriented representation of the non-transitive approximate min-max association rules extracted from context $\mathcal{D}$ for $minsupp = 2/6$ and $minconf = 2/5$ is given in figure 13.

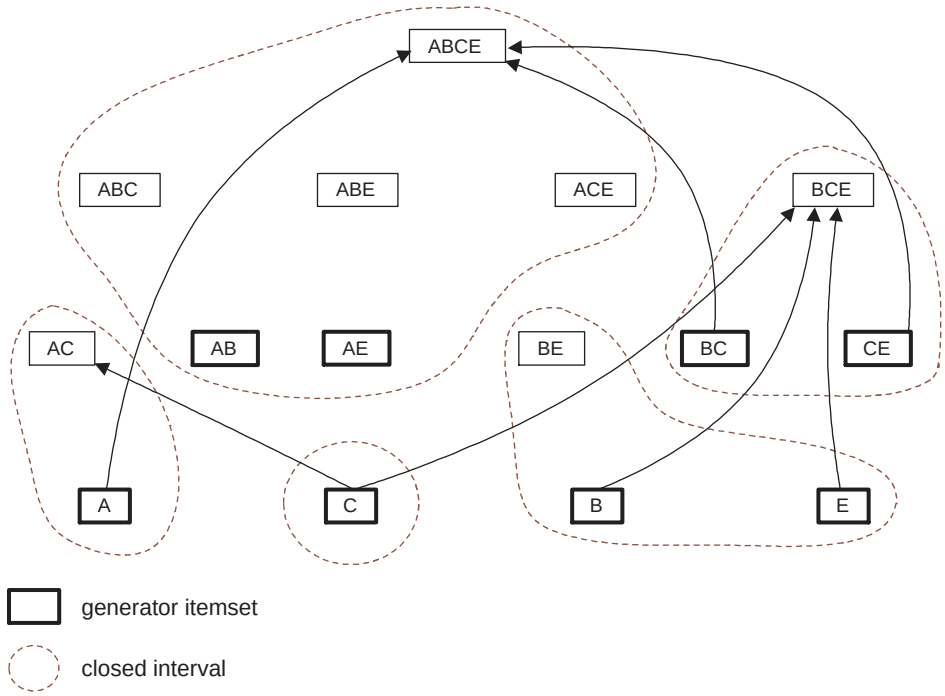

Figure 13. Non-transitive approximate min-max association rules extracted from $\mathcal{D}$.

Each edge between two vertices v_g and v_f represents the non-transitive approximate rule $g \rightarrow f \setminus g$ where g is a generator and f a frequent closed immediate successor of the closure of g. That is an edge between a minimal vertex of a closed interval and the maximal vertex of an immediately above closed interval.
An edge in figure 11 represents a transitive rule if it is an edge between a minimal vertex of a closed interval and the maximal vertex of another closed interval that is not immediately above the first one: There is a closed interval "intermediate" between these two intervals. For instance, the rule C $\rightarrow$ ABE between the closed

intervals {{C}} and {{AB},{AE},{ABC},{ABE},{ACE}, {ABCE}} is transitive since we have rules C → A and A → BCE and the closed interval {{A},{AC}} is intermediate, i.e., {C} ⊂ {AC} ⊂ {ABCE}. The confidence of C → ABE is equal to the product of rules C → A and A → BCE confidences.
In order to derive all transitive rules, we first add all rules that are compositions of two non-transitive rules, we then derive from them rules that are compositions of three non-transitive rules and so on until no new rule can be derived. The three transitive min-max rules reconstructed are C → ABE, B → ACE and E → ABC. They are all compositions of two non-transitive rules, that have the form $g_i \rightarrow f_j \setminus g_i$ with $g_i \subseteq \gamma(g_i) = f_i \Lessdot f_j$, represented in figure 7.
The algorithm presented in figure 14 generates the set *MinMaxApprox* of approximate min-max rules using the set *MinMaxReduc* of non-transitive approximate min-max rules and the *minconf* threshold as its input.

Input : set *MinMaxReduc*, confidence threshold *minconf*
Output : set *MinMaxApprox*

1) $Test \leftarrow MinMaxReduc$
2) $MinMaxTrans \leftarrow \varnothing$
3) **while** ($Test \neq \varnothing$) **do**
4) **forall** rule $\{r_1 : a_1 \rightarrow c_1, r_1.supp, r_1.conf\} \in Test$ **do**
5) **forall** rule $\{r_2 : a_2 \rightarrow c_2, r_2.supp, r_2.conf\} \in MinMaxReduc$
6) **with** $a_2 \subset a_1 \cup c_1 \subset a_2 \cup c_2$ **do**
7) **if** $(r_1.conf \times r_2.conf \geq minconf)$
8) **and** $(\{r_3 : a_1 \rightarrow (a_2 \cup c_2) \setminus a_1\} \notin MinMaxTrans)$ **then**
9) $MinMaxTrans \leftarrow MinMaxTrans \cup \{r_3, r_2.supp, r_1.conf \times r_2.conf\}$
10) $Test \leftarrow Test \cup \{r_3, r_2.supp, r_1.conf \times r_2.conf\}$
11) **end**
12) **end**
13) $Test \leftarrow Test \setminus \{r_1\}$
14) **end**
15) **end**
16) **return** $MinMaxApprox \leftarrow MinMaxReduc \cup MinMaxTrans$

Figure 14. Algorithm for reconstructing transitive approximate min-max association rules.

The approach is incremental: We iteratively add new transitive min-max rules until no new rule has been created (steps 3 to 15). During each iteration, the *Test* set contains all rules examined to generate new transitive rules and the algorithm stops when *Test* is empty. This set is initialized with all non-transitive rules (step 1) and all rules r_1 it contains, that have the form $g_i \rightarrow f_j \setminus g_i$, are successively examined (steps 4 to 14). For each non-transitive rule r_2 in *MinMaxReduc* with the form $g_j \rightarrow f_m \setminus g_j$ such that $g_j \subset f_j \subset f_m$ (steps 5 and 6 to 12), the transitive rule r_3 with the form $g_i \rightarrow f_m \setminus g_i$ is generated in *MinMaxTrans* and *Test* (steps 9 and 10) if its confidence is sufficient and it is not already present in *MinMaxTrans* (steps 7

and 8). Then, rule r_1 is removed from *Test* (step 13) since it is not needed anymore: Only transitive rules generates from r_1 will be examined in the following iterations.

Example 10. The transitive rule B $\rightarrow$ ACE is derived from rules B $\rightarrow$ CE and BC $\rightarrow$ AE whose antecedent {BC} is a subset of {B} $\cup$ {CE} = {BCE}, and {BCE} is itself a subset of {BC} $\cup$ {AE} = {ABCE}. The rule E $\rightarrow$ ABC is derived from E $\rightarrow$ BC and CE $\rightarrow$ AB. The rule C $\rightarrow$ ABE can be derived from rules C $\rightarrow$ A and A $\rightarrow$ BCE, or from rules C $\rightarrow$ BE and BC $\rightarrow$ AE or CE $\rightarrow$ AB.

5. Experimental results

We used the four following datasets during these experiments: T10I4D100K[5] is a synthetic dataset built according to sales data properties. It contains 100,000 objects with an average object size of 10 items and an average size of potential maximal frequent itemsets of 4 items. The MUSHROOMS dataset describes 23 characteristics (attributes) of 8,416 mushrooms (objects): Each object is related to 23 items and we have 127 items on the whole. The C20D10K and C73D10K (Hettich and Bay, 1999) datasets are samples of the 1990 census in Kansas, each containing 10,000 objects corresponding to the first 10,000 listed people. Each object is described by 20 attributes (20 items by objects and 386 items on the whole) in C20D10K and 73 attributes (73 items by objects and 2,178 items on the whole) in C73D10K.
Running times of the generation of all association rules and of the min-max bases are not shown since they are insignificant compared to execution times of the itemset extraction. Indeed, no dataset scan is required for this phase and all computations take place in main memory. As a data-point, the largest running time obtained was 46.27 seconds for the generation of the 2,053,936 approximate association rules for C73D10K on a Pentium II at 333MHz with 256MB of main memory.

Number of exact association rules extracted. The total number of exact association rules and the number of min-max exact association rules are presented in table VIII. No exact association rule is extracted from T10I4D100K since, for this *minsupp* value, all frequent itemsets are frequent closed itemsets. Thus, they are themselves their own unique generator and consequently, there is no exact association rule $l_1 \Rightarrow (l_2 \setminus l_1)$ between two frequent itemsets $l_1 \subset l_2$ having identical closures $\gamma(l_1) = \gamma(l_2)$. The three other datasets are made up of correlated data, and the total number of exact rules is important, making it difficult to discover interesting information. For these datasets, the min-max exact basis reduces the number of rules by a factor varying from 13 to 50. Since there is no information loss, it brings a complete summary of relevant information that is easier to exploit for the analyst.

Number of approximate association rules extracted. The total number of approximate association rules and the number of approximate and non-transitive approximate min-max rules are presented in table IX. The number of approximate

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

Table VIII. Number of exact association rules extracted.

Dataset	minsupp	Exact rules	Min-max basis
T10I4D100K	0.5%	0	0
MUSHROOMS	30%	7,476	543
C20D10K	50%	2,277	457
C73D10K	90%	52,035	1,369

rules is very significant for the four datasets, up to more than 2,000,000. Reducing this number is thus essential in order to make it usable by the analyst. For T10I4D100K, all frequent itemsets are both closed and their own generators and the approximate min-max basis is identical to the set of all rules. The non-transitive basis represents a reduction by a factor of 5 approximately of the number of rules. For the three other datasets, the total number of approximate rules is much more important than for the synthetic dataset since they contain dense and correlated data: The number of frequent itemsets is much more important and thus, it is the same for the number of approximate rules. However, the fraction of frequent itemsets that are closed is small and the bases reduce considerably the number of rules, by a factor of varying from 10 to 50 for the approximate min-max basis and, from 40 to 500 for the non-transitive basis.

Table IX. Number of approximate association rules extracted.

Dataset (minsupp)	minconf	Approximate rules	Approximate min-max basis	Non-transitive min-max basis
T10I4D100K	70%	20,419	20,419	4,004
(0.5%)	30%	22,952	22,952	4,519
MUSHROOMS	70%	37,671	2,961	1,221
(30%)	30%	71,412	6,571	1,578
C20D10K	70%	89,601	10,116	1,957
(50%)	30%	116,791	13,634	1,957
C73D10K	90%	2,053,896	43,171	5,718
(90%)	80%	2,053,936	43,175	5,718

Examining rules generated in the min-max approximate basis and its transitive reduction for the MUSHROOMS dataset, we verified that rule 4 of example 1 in section 1 is the only one generated among the nine rules. Indeed, the itemsets {free gills} and {free gills, edible, partial veil, white veil} are frequent closed itemsets and the first is an immediate predecessor of the second. Moreover, they are the only frequent closed itemsets in the interval [$\varnothing$, {free gills, edible, partial veil, white veil}] and the frequent closed itemset {free gills} is itself its own unique generator. Thus, rule 4 is the only min-max approximate rule among the nine rules and is non-transitive.

6. Conclusion

The problem of association rules relevance occurs for most operational datasets. This problem is related to the huge number of rules generated and the presence of many redundancies. The approach proposed in this paper consists in generating bases for association rules that minimize as much as possible the number of extracted rules while bringing the same information to the end-user. Using a semantic based on the Galois connection, we first characterized min-max association rules as the non-redundant rules with minimal antecedent and maximal consequent. Each min-max rule summarizes several other rules, suggesting that these rules are the most relevant from the analyst's point of view. From this characterization, we defined the min-max basis for exact association rules, the min-max basis for approximate association rules and its transitive reduction – which we believe is more useful for the analyst as it retains only the most precise rules. The union of the former and one of the latter of these bases constitutes a min-max basis for association rules that is a generating set for all association rules, their supports and their confidences.
We presented algorithms for generating these bases from the frequent closed itemsets and their generators, such as extracted by the Close and A-Close algorithms. When all frequent itemsets have been mined, the Close$^+$ algorithm identifies frequent closed itemsets and their generators among frequent itemsets. We also introduced simple methods and algorithms to derive all exact rules, all approximate rules and all transitive approximate min-max rules from the bases. None of these algorithms requires accessing the dataset and their execution times are thus insignificant compared to the running times of the frequent itemsets, or the frequent closed itemsets, extraction.
Experimental results conducted on both synthetic and operational datasets show that the extraction of these bases considerably reduces the number of rules, particularly in the case of dense or correlated data. The result is easier to browse and since redundant – and often misleading – rules are suppressed, its usefulness is improved. Moreover, all of the data-space is characterized by the min-max rules and this approach does not suffer from poorly characterized or uncharacterized sub-spaces of the data-space, an important weakness of many reduction methods. Another interesting feature of this approach is the possibility to construct a graph-oriented representation of the min-max bases that is easily understandable for the end-user. It provides a natural, simple and clear graphical representation of association rules covering all the data-space and from which the deduction of all other rules is direct.
An interesting perspective of future work is the definition of an inference system for association rules equivalent to the Armstrong axioms for implications. As pointed out in section 1.1, up to now no complete and sound inference system that takes supports and confidences into consideration has been proposed. Another attractive perspective of future work is the introduction of the min-max bases in the data analysis and the Formal Concept Analysis domains. Indeed, the min-max association rule definition is valid within the global and partial implication rule frameworks. Hence, the definitions of the min-max bases for exact and approximate association rules are also valid for global and partial implication rules respectively. Since these bases represent no information loss and are constituted of the most relevant rules

from the analyst's point of view, we believe that studying their impact in these domains is also an interesting perpective.

References

Agrawal R., Imielinski T. and Swami A. Mining association rules between sets of items in large databases. *Proceedings of the SIGMOD conference*, pp 207–216, May 1993.

Agrawal R. and Srikant R. Fast algorithms for mining association rules in large databases. *Proceedings of the VLDB conference*, pp 478–499, September 1994.

Armstrong W. W. Dependency structures of data base relationships. *Proceedings of the IFIP congress*, pp 580–583, August 1974.

Baralis E. and Psaila G. Designing templates for mining association rules. *Journal of Intelligent Information Systems*, 9(1):7–32, L. Kerschberg, Z. Ras and M. Zemankova editors, Kluwer Academic Publishers, August 1997.

Bastide Y., Taouil Y., Pasquier N., Stumme G. and Lakhal L. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, U. Fayyad, S. Sarawagi and P. Bradley editors, ACM Computer Press, December 2000.

Bayardo R. J. Efficiently mining long patterns from databases. *Proceedings of the SIGMOD conference*, pp 85–93, June 1998.

Bayardo R. J. and R. Agrawal. Mining the most interesting rules. *Proceedings of the KDD conference*, pp 145–154, August 1999.

Bayardo R. J., Agrawal R. and Gunopulos D.. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4(2/3):217–240, S. Chaudhuri editor, Kluwer Academic Publishers, July 2000.

Beeri C. and Bernstein P. A. Computational problems related to the design of normal form relational schemas. *Transactions on Database Systems*, 4(1):30–59, March 1979.

Blake C. L. and Merz C. J. *UCI Machine Learning databases Repository.* University of California, Irvine, Department of Information and Computer Science, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Brin S., Motwani R., Ullman J. D. and Tsur S. Dynamic itemset counting and implication rules for market basket data. *Proceedings of the SIGMOD conference*, pp 255–264, May 1997.

Brin S., Motwani R and Silverstein C. Beyond market baskets: Generalizing association rules to correlation. *Proceedings of the SIGMOD conference*, pp 265–276, May 1997.

Duquenne V. and Guigues J.-L. Famille minimale d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 24(95):5–18, 1986.

Ganter B. and Wille R. *Formal Concept Analysis: Mathematical foundations.* Springer-Verlag, 1999.

Han J. and Fu Y. Mining multiple-level association rules in large databases. *Transactions on Knowledge and Data Engineering*, 11(5):798–804, P.-S. Yu editor, IEEE Computer Science, September/October 1999.

Hettich S. and Bay S. D. *UCI Knowledge Discovery in Databases Archive.* University of California, Irvine, Department of Information and Computer Science, 1999. `http://kdd.ics.uci.edu`.

Klemettinen M., Mannila H., Ronkainen P., Toivonen H. and Verkamo A. I. Finding interesting rules from large sets of discovered association rules. *Proceedings of the CIKM conference*, pp 401–407, November 1994.

Lin D. and Kedem Z. M. Pincer-Search: A new algorithm for discovering the maximum frequent set. *Proceedings of the EDBT conference*, pp 105–119, March 1998.

Liu B., Hsu W. and Ma Y. Pruning and summarizing the discovered association rules. *Proceedings of the KDD conference*, pp 125–134, August 1999.

Luxenburger M. Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 29(113):35–55, 1991.

Maier D. Minimum covers in relational database model. *Journal of the ACM*, 27(4):664–674, ACM Computer Press, October 1980.

Mannila H., Toivonen H. and Verkamo A. I. Efficient algorithms for discovering association rules. *Proceedings of the AAAI workshop on Knowledge Discovery in Databases*, pp 181–192, July 1994.

Mannila H. and Toivonen H. Multiple uses of frequent sets and condensed representations. *Proceedings of the KDD conference*, pp 189–194, August 1996.

Mannila H. and Toivonen H. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, U. Fayyad editor, Kluwer Academic Publishers, September 1997.

Meo R., Psaila G. and Ceri S. An Extension to SQL for mining association rules. *Data Mining and Knowledge Discovery*, 2(2):195–224, U. Fayyad editor, Kluwer Academic Publishers, June 1998.

Morimoto Y., Fukuda T., Matsuzawa H., Tokuyama T. and Yoda K. Algorithms for mining association rules for binary segmentations of huge categorical databases. *Proceedings of the VLDB conference*, pp 380–391, August 1998.

Ng R. T., Lakshmanan V. S., Han J. and Pang A. Exploratory mining and pruning optimizations of constrained association rules. *Proceedings of the SIGMOD conference*, pp 13–24, June 1998.

Pasquier N., Bastide Y., Taouil R. and Lakhal L. Pruning closed itemset lattices for association rules. *Proceedings of the BDA conference*, pp 177–196, Octobre 1998.

Pasquier N., Bastide Y., Taouil R. and Lakhal L. Discovering frequent closed itemsets for association rules. *Proceedings of the ICDT conference*, LNCS 1540, pp 398–416, January 1999.

Pasquier N., Bastide Y., Taouil R. and Lakhal L. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, M. Jarke, D. Shasha editors, Elsevier Science, March 1999.

Pasquier N., Bastide Y., Taouil R. and Lakhal L. Closed set based discovery of small covers for association rules. *Proceedings of the BDA conference*, pp 361–381, Octobre 1999.

Pei J., Han J. and Mao R. CLOSET: An efficient algorithm for mining frequent closed itemsets. *Proceedings of the DMKD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp 21–30, May 2000.

Piatetsky-Shapiro G. and Matheus C. J. The interestingness of deviations. *Proceedings of the AAAI workshop on Knowledge Discovery in Databases*, pp 25–36, July 1994.

Silberschatz A. and Tuzhilin A. What makes patterns interesting in knowledge discovery systems. *Transactions on Knowledge and Data Engineering*, 8(6):970–974, IEEE Computer Science, December 1996.

Silverstein C., Brin S. and Motwani R. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, U. Fayyad editor, Kluwer Academic Publishers, January 1998.

Srikant R. and Agrawal R. Mining generalized association rules. *Proceedings of the VLDB conference*, pp 407–419, September 1995.

Srikant R. and Agrawal R. Mining quantitative association rules in large relational tables. *Proceedings of the SIGMOD conference*, pp 1–12, June 1996.

Srikant R., Vu Q. and Agrawal R. Mining association rules with item constraints. *Proceedings of the KDD conference*, pp 67–73, August 1997.

Taouil R., Pasquier N., Bastide Y. and Lakhal L. Mining bases for association rules using closed sets. *Proceedings of the ICDE conference*, p 307, Febuary 2000.

Toivonen H., Klemettinen M., Ronkainen P., Hätönen K. and Mannila H.. Pruning and grouping discovered association rules. *Proceedings of the ECML Workshop*, pp 47–52, April 1995.

Zaki M. J., Parthasarathy S., Ogihara M. and Li W. New algorithms for fast discovery of association rules. *Proceedings of the KDD conference*, pp 283–286, August 1997.

Zaki M. J. and Ogihara M. Theoretical foundations of association rules. *Proceedings of the DMKD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp 7:1–7:8, June 1998.

Zaki M. J. and Hsiao C.-J. CHARM: An efficient algorithm for closed association rule mining. *Technical Report 99–10*, October 1999.

Zaki M. J. Generating non-redundant association rules. *Proceedings of the KDD conference*, pp 34–43, August 2000.