Generating an informative cover for association rules

Laurentiu Cristofor and Dan Simovici*

January 12, 2002

Abstract

One of the main issues in mining association rules is that large numbers of rules can be generated from databases, thus making the results hard to examinate by a human analyst. Previous approaches have focused on introducing various measures of interestingness that could be used either to generate only interesting rules or to prune rules that are non-interesting. Pasquier et al. have discussed the generation of Guigues-Duquenne-Luxenburger bases (GDL bases) defined as sets of association rules that are non-redundant with respect to a set of inference rules. Using a similar approach, we introduce a new rule of inference and define the notion of association rules cover as a minimal set of rules that are non-redundant with respect to this new rule of inference.

We present an algorithm for generating such a cover. Our experimental results (obtained using both synthetic and real data sets) show that our covers are smaller than the GDL bases and they are computed in time that is comparable to the classic Apriori algorithm for generating rules. Interestingly, while the number of rules increases when the minimal confidence is lowered, the size of covers tends to decrease unlike the GDL bases. The covers that we introduced can be regarded as good summarizations of the sets of association rules.

Keywords: association rule, inference rule, cover, cover rule, coverage relation

1 Introduction

The discovery of association rules in databases is one of the most important problems in data mining and the focus of many publications since its introduction in [AIS93]. In that paper the problem was divided into two subproblems:

1. finding all frequent itemsets

*University of Massachusetts at Boston, Boston MA 02125, USA, [laur,dsim]@cs.umb.edu

2. generating all association rules using the frequent itemsets

Finding all frequent itemsets is a computationally intensive problem and the papers dealing with it have proposed increasingly efficient algorithms: Apriori [AS94b], Partition [SON95], Dynamic Itemset Counting [BMUT97], A-Close [PBTL99b], Close [PBTL99c], ChARM [ZH99], Closure [CCS00], Tree-projection [AAP00], [AAP01], FP-growth [HPY00], CLOSET [PHM00], and Partial Total Trees [CGL01]. The main issue in the generation of association rules is that in practice the number of association rules generated is too large and can easily overwhelm a human analyst. To address this problem several methods were proposed. Measures of interestingness can be used to assess the interestingness of a rule and thus generate only interesting rules or prune non-interesting ones. Various such interestingness measures were discussed in [BA99], [LHM99], and [JS01]; for other related research we refer the reader to the excellent survey of [HH99]. [SLRS99] presented pruning rules for removing semantically redundant association rules. [PT98] and [PT00] introduced the concept of rules that are unexpected with respect to prior beliefs and analyzed the generation of such rules. [PBTL99a], [BPT$^+$00], and [Zak00] gave various definitions of the concept of redundant rules. Out of these redundancy definitions, the one from [PBTL99a] is of special interest to us since in this paper the redundancy of an association rule is determined by whether or not the rule can be infered from other association rules through the use of a number of inference rules.

Our paper continues the line of research from [PBTL99a] by introducing a new rule of inference for association rules and by defining the concept of a cover of the association rules as a minimal set of rules that are non-redundant with respect to this new inference rule. Our concept of cover is different from the rule cover concept defined in [TKR$^+$95], which referred to the pruning of sets of association rules with identical consequent and was not related to the use of inference rules.

We consider that the use of inference rules in the definition of redundant association rules is an important approach because it offers a great potential for minimizing the number of rules generated and also because it provides a general mechanism for computer assisted exploration of the set of association rules. The inference rule that we introduce in this paper materializes such potential due to its interesting properties and simplicity.

The remainder of the paper is structured as follows: Section 2 presents known bases and rules of inference for association rules, and introduces new inference rules. Section 3 defines the concept of a cover for association rules, discusses properties of covers, and introduces an algorithm for cover generation. Section 4 presents our experimental results and Section 5 concludes with a review of the presented facts.

2 Rules of inference for association rules

2.1 Association rule mining

We begin with some basic definitions and notations. Let $\tau = (T, H, \rho)$ be a binary table, where T is the name of the table, $H = A_1 \dots A_n$ is the heading of the table, and $\mathsf{aDom}(A_1) = \dots = \mathsf{aDom}(A_n) = \{0, 1\}$ where 1 indicates the presence of an attribute in a tuple and 0 indicates its absence. We will also refer to the attributes $A_1 \dots A_n$ as items. To simplify notation we write XY instead of $X \cup Y$, and A_1A_2 instead of $\{A_1, A_2\}$.

Definition 2.1 An *itemset* I is a set of items $I \subseteq H$. The *support* of an itemset I, denoted by $\mathtt{supp}(I)$, is given by the fraction of tuples in the table that contain the items of I: $\frac{|\{t \in \rho | t[I] = \mathbf{1}_I\}|}{|\rho|}$. $\mathtt{supp}(\emptyset)$ is defined to be 1.

The *closure* of an itemset I is denoted by $\mathtt{cl}(I)$ and consists of all items that appear in all tuples that contain I: $\{A_i \in H | \text{if } t[I] = \mathbf{1}_I \text{ then } t[A_i] = 1\}$ An itemset is called *closed* if it is equal to its closure.

An *association rule* $X \to Y$ consists of two disjoint non-empty itemsets X and Y, called *antecedent* and respectively *consequent* and denoted by $\mathsf{antc}(r)$ and $\mathsf{cons}(r)$, respectively. The *attribute set* of a rule r is the set $\mathsf{attr}(r) = \mathsf{antc}(r) \cup \mathsf{cons}(r)$.

The *support* of the association rule is the support of $\mathsf{attr}(r)$. The *confidence* of the association rule is the ratio:

$$\mathtt{conf}(r) = \frac{\mathtt{supp}(\mathsf{attr}(r))}{\mathtt{supp}(\mathsf{antc}(r))}.$$

An association rule with confidence 1 is called an *exact* association rule. An association rule that is not exact is called an *approximative* association rule (see [PBTL99a]). □

An exact association is denoted by $X \Rightarrow Y$. Not all authors require the antecedent and consequent of a rule to be disjoint, so this requirement should be ignored in the case of results we quote from other papers.

The problem of mining association rules is the following: given a table τ, a minimum support value $\mathsf{minsupp}$, and a minimum confidence value $\mathsf{minconf}$, we are required to generate all association rules with support greater or equal to $\mathsf{minsupp}$ and confidence greater or equal to $\mathsf{minconf}$.

Definition 2.2 An association rule is said to be *valid* (see [PBTL99c]) if its support and confidence are greater or equal to the $\mathsf{minsupp}$ and respectively $\mathsf{minconf}$ values. □

2.2 Inference rules and bases for association rules

An important issue in association rule mining is that many times the number of association rules generated is overwhelming for the user of the mining system.

One approach to this issue is to only generate the association rules that are non-redundant in the sense that they cannot be deduced from other rules by using certain rules of inference. A minimal set of such association rules was called *basis* in [PBTL99a]. To avoid confusion, we mention here that the single word rule will only be used in the sense of association rules and will never be used to refer to an inference rule.

Based on the results of [GD86] and [Lux91], [PBTL99a] introduced the *Guigues-Duquenne basis* for exact association rules and the *Luxenburger basis* for approximative association rules, which together form a basis for the valid association rules.

The Guigues-Duquenne basis is a minimal set of exact association rules from which the complete set of exact association rules can be deduced using the following two inference rules:

1. $X \Rightarrow Y, W \Rightarrow Z \vdash XW \Rightarrow YZ$
2. $X \Rightarrow Y, Y \Rightarrow Z \vdash X \Rightarrow Z$

The Guigues-Duquenne basis does not allow us to infer the support of the rules and in fact, by ignoring the support values, the first inference rule can lead to association rules that have inferior support compared to the rules used in its generation. Note also that the rules of inference used here are similar to the Armstrong rules for functional dependencies [ST95]. It is easy to verify that the first rule mentioned above is exchangeable for the Armstrong augmentation rule $X \Rightarrow Y \vdash XZ \Rightarrow YZ$, that is, given one of them we can infer the other.

In [SCC00] we introduced purity dependencies as a generalization of functional dependencies and proved that these new types of dependencies satisfy properties similar to the Armstrong rules. Given that the Armstrong rules for functional dependencies transcribe well to exact association rules, one might expect that the properties of the generalizations of functional dependencies also hold for the approximative association rules. However, this is not the case, since an augmented rule can fail to achieve both minimum support and minimum confidence, and from the two transitivity properties that we proved in [SCC00], only one of them is holding in the case of association rules, as the next theorem shows.

Theorem 2.3 *Let $X \rightarrow Y$ and $Y \Rightarrow Z$ be two valid association rules. Then $X \rightarrow Z$ is a valid association rule.*

Proof. We have $\mathtt{supp}(Y) = \mathtt{supp}(YZ)$, so $\mathtt{supp}(XY) = \mathtt{supp}(XYZ)$ and $\mathtt{supp}(XYZ) \leq \mathtt{supp}(XZ)$, which implies that the rule $X \rightarrow Z$ has minimum support. The confidence of this rule is $\frac{\mathtt{supp}(XZ)}{\mathtt{supp}(X)} \geq \frac{\mathtt{supp}(XY)}{\mathtt{supp}(X)}$ based on the observation we made before. Therefore our rule also has minimum confidence, so it is valid. ∎

The Luxenburger basis is a minimal set of approximative rules from which the complete set of approximative rules can be deduced using the two properties introduced in [Lux91]:

1. the association rule $X \rightarrow Y$ has the same support and confidence as the rule $\mathtt{cl}(X) \rightarrow \mathtt{cl}(Y)$

2. for any three closed itemsets I_1, I_2, and I_3, such that $I_1 \subseteq I_2 \subseteq I_3$, the confidence of the rule $I_1 \rightarrow I_3$ is equal to the product of the confidences of the rules $I_1 \rightarrow I_2$ and $I_2 \rightarrow I_3$, and its support is equal to the support of the rule $I_2 \rightarrow I_3$

The first property indicates that we only need to generate association rules whose antecedent and consequent are closed itemsets, while the second property allows us to eliminate transitive rules between closed itemsets (see also [Zak00]). Both of these properties can be regarded as new types of inference rules and they permit the inference of both the support and confidence of the resulting rules.

Together, the Guigues-Duquenne basis and the Luxenburger basis, provide a minimal basis for association rules, which we will denote as the GD-L basis. Starting from this basis it is however possible to infer rules that are not valid, as was noticed in the presentation of the Guigues-Duquenne basis.

Both [BPT+00] and [Zak00] have introduced other bases based on new definitions of redundancy that had a lesser emphasis on the use of inference rules, and emphasized more the format of the association rules, starting from the idea of presenting the user with the most informative rules. As a result, these new bases are larger than the GD-L basis.

2.3 A new inference rule for association rules

We now introduce a rule of inference for association rules that does not take into account their support and confidence, these being easily computed in a data mining application by using the results obtained during the generation of all frequent itemsets. The reason for this approach is that the inference of association rules with exact support and confidence is not a fundamental requirement since it is hard to imagine the user trying to infer manually other rules than those presented to him. On the other hand, the data mining application already has all the information regarding the support of the frequent itemsets and this information can be used to compute the exact support and confidence of any rule that we know is valid.

Theorem 2.4 *Let r, r' be two association rules such that* $\mathsf{attr}(r') \subseteq \mathsf{attr}(r)$ *and* $\mathsf{supp}(\mathsf{antc}(r')) \leq \mathsf{supp}(\mathsf{antc}(r))$. *Then,*

$$\mathsf{supp}(r') \geq \mathsf{supp}(r) \text{ and } \mathsf{conf}(r') \geq \mathsf{conf}(r).$$

Proof. Since $\mathsf{attr}(r') \subseteq \mathsf{attr}(r)$, it follows that $\mathsf{supp}(\mathsf{attr}(r')) \geq \mathsf{supp}(\mathsf{attr}(r))$, so the rule r' has greater support. Also, for the confidence of this rule we can write

$$\mathsf{conf}(r') = \frac{\mathsf{supp}(r')}{\mathsf{supp}(\mathsf{antc}(r'))} \geq \frac{\mathsf{supp}(r)}{\mathsf{supp}(\mathsf{antc}(r))} = \mathsf{conf}(r)$$

because $\mathsf{supp}(\mathsf{antc}(r')) \leq \mathsf{supp}(\mathsf{antc}(r))$. ∎

This justifies the introduction of the inference rule:

$$\frac{r, \mathsf{attr}(r') \subseteq \mathsf{attr}(r), \mathsf{supp}(\mathsf{antc}(r)) \leq \mathsf{supp}(\mathsf{antc}(r'))}{r'}.$$

Note that this inference rule is sound since by applying it to a valid association rule we can only derive other valid association rules.

Since this inference rule requires knowledge of the support of the rules and of the support of the antecedents of the association rules, it cannot be used as a purely syntactical inference rule. This is because, although $\mathsf{supp}(\mathsf{antc}(r))$ can be obtained by computing the ratio of the support and confidence of r, we need the additional information about $\mathsf{supp}(\mathsf{antc}(r'))$ in order to use the inference rule. Examples of purely syntactical rules are the Guigues-Duquenne rules and the rule of Theorem 2.3, since they only rely on knowledge of association rules characteristics (antecedent, consequent, support, and confidence). In this respect, our rule is similar to the Luxenburger rules which can be applied only to association rules whose antecedents and consequents are known to be closed itemsets.

The new inference rule is very powerful because starting from a valid rule it can allow us to infer many other valid rules, as shown in the next example.

Example 2.5 Consider the table from Figure 1 that was used in [PBTL99a]. We will examine this table assuming $\mathsf{minsupp} = \frac{2}{5}$ and $\mathsf{minconf} = \frac{1}{2}$

TID	A	B	C	D	E
1	1	0	1	1	0
2	0	1	1	0	1
3	1	1	1	0	1
4	0	1	0	0	1
5	1	1	1	0	1

Figure 1: A sample binary table

TID stands for tuple identifier. Note that the rule $B \rightarrow ACE$ is valid and that B, along with C and E, is the most frequent item in the table, so by using Theorem 2.4, we can also infer the valid rules: $E \rightarrow BCE$, $B \rightarrow E$, $E \rightarrow B$, $AC \rightarrow BE$, to enumerate just a few. The exact support and confidence of these rules can be computed based on the information we have about the frequent itemsets. □

Definition 2.6 If for two association rules r_1 and r_2 it is possible to infer r_2 from r_1 using Theorem 2.4, then we say that rule r_2 is *covered* by rule r_1 (and that r_1 *covers* r_2), and we write $r_1 \prec r_2$. The *coverage relation* $\prec$ consists of all ordered pairs of rules (r_1, r_2), such that $r_1 \prec r_2$. □

Because of Definition 2.6, we will also refer to the property of Theorem 2.4 as the *coverage rule*.

Definition 2.7 Two association rules, r_1 and r_2, are called *equipotent* if $r_1 \prec r_2$ and $r_2 \prec r_1$. □

The next theorem shows that equipotent rules must have the same attribute sets and that the support of their antecedents must be equal.

Theorem 2.8 *Let r_1, r_2 be two association rules. Then r_1 and r_2 are equipotent if and only if* $\mathsf{attr}(r_1) = \mathsf{attr}(r_2)$ *and* $\mathsf{supp}(\mathsf{antc}(r_2)) = \mathsf{supp}(\mathsf{antc}(r_2))$.

Proof. This statement follows immediately from Definition 2.7. ∎

Example 2.9 Consider again the table from Figure 1.

The rules $B \to ACE$, $E \to BCE$, and $BE \to CE$ are all equipotent since they have the same attribute set, and since $\mathsf{supp}(B) = \mathsf{supp}(E) = \mathsf{supp}(BE) = \frac{4}{5}$. □

The following theorem shows that the coverage relation is a preorder on the set of association rules.

Theorem 2.10 *The coverage relation $\prec$ is reflexive and transitive but is not anti-symmetric.*

Proof. Reflexivity results from Theorem 2.4 and the lack of antisymmetry follows from Definition 2.7 and Theorem 2.8.

To prove the transitivity property let $r_1 = X_1 \to Y_1$, $r_2 = X_2 \to Y_2$, and $r_3 = X_3 \to Y_3$, be three rules such that $r_1 \prec r_2$ and $r_2 \prec r_3$. This means that $\mathsf{supp}(X_2) \leq \mathsf{supp}(X_1)$ and $\mathsf{supp}(X_3) \leq \mathsf{supp}(X_2)$, which imply that $\mathsf{supp}(X_3) \leq \mathsf{supp}(X_1)$. Also we have that $X_2Y_2 \subseteq X_1Y_1$ and $X_3Y_3 \subseteq X_2Y_2$, which imply that $X_3Y_3 \subseteq X_1Y_1$. Therefore, since the conditions of Theorem 2.4 are met, it results that $r_1 \prec r_3$. ∎

Equipotent rules are important because they are interchangeable from the point of view of the coverage relation.

Corollary 2.11 *If r_1 and r_2 are equipotent and $r_1 \prec r_3$ for some r_3, then also $r_2 \prec r_3$.*

Proof. Since r_1 and r_2 are equipotent we have $r_2 \prec r_1$ so, from the hypothesis and the transitivity of $\prec$, we also obtain $r_2 \prec r_3$. ∎

3 Covers for association rules

Based on Theorem 2.4, we define the notion of a cover of the set of valid association rules:

Definition 3.1 Let $\mathcal{R}$ be the set of all valid association rules extracted from a table τ. A *cover of* $\mathcal{R}$ is a minimal set $\mathcal{C} \subseteq \mathcal{R}$, such that any rule from $\mathcal{R} - \mathcal{C}$ is covered by a rule in $\mathcal{C}$. A rule belonging to $\mathcal{C}$ is called a $\mathcal{C}$-*cover rule*. □

Note that $\mathcal{R}$ does not have to have an unique cover, since there can be several covers due to the existence of equipotent rules.

Proposition 3.2 *Given a cover $\mathcal{C}$ of $\mathcal{R}$, such that one of its rules r_1 is equipotent with a rule r_2, $(\mathcal{C} - \{r_1\}) \cup \{r_2\}$ is another cover.*

Proof. The proof is immediate from Corollary 2.11. ∎

Example 3.3 Consider again the table from Example 2.5.

If we mine the table using $\mathsf{minsupp} = \frac{2}{5}$ and $\mathsf{minconf} = \frac{1}{2}$, then we obtain 50 association rules which we omit listing here. The GD-L basis contains 6 rules: $\{A \Rightarrow C,\ B \Rightarrow E,\ E \Rightarrow B,\ AC \rightarrow BE,\ BE \rightarrow C,\ C \rightarrow A\}$. For this table there is only one possible cover that contains the rule: $B \rightarrow ACE$, and it can be easily verified that the rules in the GD-L basis are all covered by this rule. □

The next theorem expresses some important properties of cover rules.

Theorem 3.4 *Let $\mathcal{C}$ be a cover of a set of association rules $\mathcal{R}$ extracted from a table τ. The following statements hold:*

1. *If $r_1, r_2 \in \mathcal{C}$, then $\mathsf{attr}(r_1) \neq \mathsf{attr}(r_2)$.*
2. *If r is a $\mathcal{C}$-cover rule, then for any $r' \in \mathcal{R}$ such that $\mathsf{attr}(r) = \mathsf{attr}(r')$, we have*
$$\mathtt{supp}(r) \leq \mathtt{supp}(r') \text{ and } \mathtt{conf}(r) \leq \mathtt{conf}(r').$$
3. *If r is a $\mathcal{C}$-cover rule, then there is no $r' \in \mathcal{R}$ such that $\mathsf{antc}(r) = \mathsf{antc}(r')$ and $\mathsf{cons}(r) \subset \mathsf{cons}(r')$.*

Proof.

1. If two $\mathcal{C}$-cover rules have the same attribute set, then one of them will be covered by the other one and therefore does not belong to the cover.
2. This is immediate from the fact that Theorem 2.4 only allows for the inference of rules with greater or equal values of support and confidence.
3. Suppose that r is the $\mathcal{C}$-cover rule $I \rightarrow X$ and $r' \in \mathcal{R}$ has the form $I \rightarrow Y$ such that $X \subset Y$. Note that $r' \prec r$, and thus, $r' \notin \mathcal{C}$. Therefore, there is a $\mathcal{C}$-cover rule r_1 such that $r_1 \prec r'$. Note that $r_1 \neq r$, since otherwise, r, r' would be equipotent, which is impossible because $IX \subset IY$. Thus, we would have $r_1 \prec r$, which contradicts our hypothesis.

∎

Out of the set of possible covers, some of them are more informative than others:

Definition 3.5 An *informative cover* is a cover where for each cover rule r there is no equipotent rule r' such that $\mathsf{antc}(r') \subset \mathsf{antc}(r)$. □

Theorem 3.6 *In an informative cover, a rule whose attribute set is I has an antecedent of smaller or equal size than that of any other valid rule that has the same attribute set.*

Proof. Let $X \rightarrow I-X$ be a rule in an informative cover and suppose there exists a valid rule $Y \rightarrow I-Y$ with $Y \subset X$. If this happens, then the rule $X \rightarrow I-X$ is covered by $Y \rightarrow I-Y$, for if $Y \subset X$ we have $\texttt{supp}(Y) \geq \texttt{supp}(X)$. The rule $Y \rightarrow I-Y$ cannot be a cover rule itself based on the first property from Theorem 3.4, so it must be covered by some rule r, and by transitivity of the coverage relation it follows that $X \rightarrow I-X$ is covered by r. Since $X \rightarrow I-X$ is a cover rule, this can happen only if $r = X \rightarrow I-X$ is equipotent to $I \rightarrow I-Y$ and this contradicts Definition 3.5. Thus, the rule $Y \rightarrow I-Y$ with $Y \subset X$ cannot exist. ∎

An informative cover is interesting because it can be smaller than the bases mentioned in Section 2, it allows the inference of all valid rules and only the inference of valid rules, and, for given attribute sets, it contains rules with minimum antecedent and maximum consequent (which was a criteria used to judge the informativeness of a rule in [BPT+00]).

The reverse of this is the fact that cover rules have smaller support and confidence, which happens because the coverage rule only leads to rules with higher or equal support and confidence.

A cover summarizes the set of valid rules similarly to the way in which the frequent itemsets are summarized by knowledge of the maximal frequent itemsets [Bay98]. A cover can also be used to simplify the presentation of association rules to users: initially only cover rules could be shown to a user, then the user could select a cover rule r and retrieve a subset[1] of all rules covered by r, and then the process could be repeated. In this manner, the user could guide his search for association rules without being overwhelmed by their number. A similar type of rule exploration has been proposed in [LHM99], in the context of the so called *direction setting rules*.

3.1 Generating an informative cover for association rules

The following pseudocode describes an algorithm for generating an informative cover for the set of valid association rules.

Algorithm 3.7 (**CoverRules**) Algorithm for generating an informative cover for the valid association rules.

Let R be a queue that will contain frequent itemsets and let C be the set of association rules in which we will place the cover rules.

1 Initialize R by enqueuing into it all maximal frequent itemsets, in decreasing order of their size. C is $\emptyset$.

[1] For example, this subset could consist of the rules covered by r that have the same size and antecedent size, and of the covered rules of size inferior by 1 that have minimal antecedent size.

2 If R is empty, then output C and exit; else extract an itemset I from R.

3 For all strict subsets I_i of I, with $i = 1 \ldots 2^{|I|} - 2$, sorted primarily by their support values and secondarily by their cardinality, do:

3.1 If the rule $I_i \rightarrow I - I_i$ is valid, then add it to C if it is not covered by a rule already in C. Go to step 2.

3.2 If $i = 1$ and $|I| > 2$, then add to R each subset of I that has size $|I|-1$ and that is not already included in an itemset from R. Continue step 3.

∎

Algorithm **CoverRules** starts from the set of maximal frequent itemsets and examines them in decreasing order of their cardinalities (steps 1-2). For each such itemset I, we search for a subset S having maximum support, such that $S \rightarrow I - S$ is a valid association rule (step 3). Such a rule is a candidate cover rule and once found, the search stops and the rule is added to the set of cover rules C if it is not covered by one of the rules of C (step 3.1). While we examine each subset S of I, we may encounter some subsets such that they cannot be used as an antecedent of a rule based on the items of I. For these subsets we will have to verify whether they can be antecedents of rules based on subsets of I. This is why, in step 3.2 of the algorithm, we add to R all the subsets of I. However, those subsets that are already included in an itemset of R do not need to be added. Step 3.2 needs to be performed only once, so we perform it if the first subset examined in step 3 cannot be used as an antecedent. The collection R is a queue because we first want to examine the maximal frequent itemsets and then their subsets, added in step 3.2, in decreasing order of their size. We examine these itemsets in decreasing order of their size because an association rule whose attribute set is larger cannot be covered by an association rule whose attribute set is smaller. This ensures that a cover rule added to C cannot be covered by another cover rule that we may discover later. However, each time that we intend to add a rule to C, we still need to check whether that rule can be covered by one of the rules already in C.

The strategy of examining first the maximal frequent itemsets and then their subsets in decreasing order of their size guarantees that the set of rules that we generate is minimal. Step 3.2 guarantees that from set C we can infer all valid rules. Together, these ensure that the resulting set C is a minimal set of rules from which all valid rules can be infered, and thus is a cover. The cover is informative because in step 3, for subsets with same support, we examine first those with smaller cardinality.

There are several optimizations that can be made to Algorithm 3.7. The examination of the subsets from step 3 of **CoverRules** can be done in stages, at each stage we can examine subsets of equal cardinality, and we can reduce the number of subsets that need to be examined during the next stage based on the results obtained in the current stage. For example, if during the examination of the subsets of size k we discover one with high support that can be used to

generate a possible cover rule, then in the next stages, when examining subsets of cardinality greater than k, we will ignore all those subsets whose support is less than the support of this antecedent. The search from step 3 should also be done using binary search. For the sake of clarity, the description of Algorithm 3.7 was intentionally kept simple and omitted the details of such optimizations.

4 Experimental results

We implemented the optimized version of **CoverRules** in Java and ran it on several databases. In a first experiment we ran the algorithm on the Mushroom database from [BM98]. We present in Figure 2 the results that we obtained, as well as the results obtained for this database by [PBTL99a].

Mushroom database (minsupp = 30%)			
minconf	Valid rules	Cover	GD-L Basis
90%	20399	238	382
70%	45147	176	453
50%	64179	159	479
30%	78888	78	493

Figure 2: Results for Mushroom database

From these results it can be observed that the cover contains less rules than the GD-L basis. Another interesting result is the fact that the size of the cover decreases as minconf is lowered (see also Figure 3). This may seem surprising at first but is due to the fact that, as minconf is lowered, more valid rules exist, the redundancy of these rules is greater, and thus they can be summarized better. In fact, for minconf = 30%, the size of the cover is identical to the number of maximal frequent itemsets existing in the mushroom database (for minsupp = 30% there are 78 such maximal frequent itemsets), and this happens because all rules that can be generated using subsets of a maximal frequent itemset are valid. Notice that in this case the cover size is one order of magnitude smaller than the size of the GD-L basis, and three orders of magnitude smaller than the total number of valid association rules.

For minconf = 30% all the cover rules have as antecedent the item `veil-type = partial`. Interestingly, this attribute is common to all the mushrooms described in the database, so its support value is 1 — the maximum possible support value. By looking at a cover rule separately, the fact that the rule has as antecedent the most frequent item in the database might make us think that the rule is trivial. However, knowing that this is a cover rule, the fact that its antecedent is the most frequent item takes new meaning since it implies that any association rule that we can build from the items of the cover rule will be a valid rule. Usually, the most frequent items in a database are known to the users of the database, so a cover rule having such an item for antecedent can be easily interpreted, even without the help of the computer.

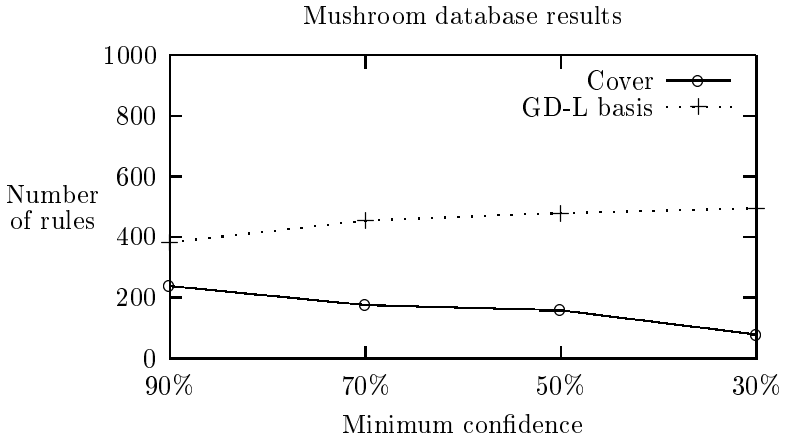

Figure 3: Graphical plot of the results obtained on the Mushroom database

In the case of the Mushroom database, the **CoverRules** algorithm is about as fast as the Apriori **ap_genrules** procedure for generating all valid rules, which was described in [AS94a]. Both algorithms finished their processing in a couple of seconds, so we do not include their detailed time results here.

We also tested our algorithm on synthetically generated data. For this purpose we used our own implementation of the synthetic data generator described in [AS94b]. We generated database SPARSE with 100000 transactions of average size 10, having 100 items, and containing 300 patterns of average size 5. This is a sparse database that we mined for minsupp $= 5\%$, thus discovering 207 maximal frequent itemsets. For all our experiments on this database, the times taken by **CoverRules** and **ap_genrules** were well below 1 second so we omit them again. The number of association rules discovered and the corresponding cover size are presented in Figure 4.

We can observe that in this experiment the cover size increases initially as minconf decreases. This happens because in this experiment the database is sparse, so the redundancy is poor and rules that are discovered when the confidence threshold is lowered do not necessarily allow the inference of rules with higher confidence. For minconf $= 10\%$ we obtain all valid rules and lowering the confidence threshold further does not bring any new rules. In fact, the 194 cover rules correspond to the maximal large itemsets that have cardinality greater than 1, since there are 13 such maximal frequent itemsets of size 1.

For our final experiment we generated a dense synthetic database, which we will call DENSE, with 100000 transactions of average size 15, having 100 items, and containing 100 patterns of average size 10. Our strategy for obtaining dense synthetic databases consists of choosing fewer and longer patterns. We mined this database for minsupp $= 5\%$ and we obtained 3182 maximal frequent itemsets. For this experiment the times taken by the **CoverRules** and **ap_genrules**

SPARSE database (minsupp = 5%)		
minconf	Valid rules	Cover
90%	3	2
80%	19	13
70%	42	25
60%	87	55
50%	186	124
40%	321	196
30%	455	240
20%	658	257
10%	880	194
5%	880	194
1%	880	194

Figure 4: Results for SPARSE database

algorithms became noticeable and we include them in Figure 5.

We can see from these results that again, for this dense database, the cover size generally tends to decrease as we lower the confidence threshold. All valid rules are discovered for confidence 5%, so lowering minconf further does not result in more rules. There is only one maximum frequent itemset of size 1, which accounts for the difference between the number of maximal frequent itemsets and the cover size obtained in this case. The time taken by the rule generation algorithms is more significative and allows us to notice that **CoverRules**'s performance tends to improve with the lowering of the confidence threshold, while **ap_genrules** tends to take more time as minconf is decreased. **ap_genrules** runs initially faster than **CoverRules** which performs better for lower values of minconf. These results do not include however the time necessary to output the association rules generated. The space requirements of **ap_genrules** are more significant than those of **CoverRules**, and in some experiments we had to increase the memory available to the Java Virtual Machine so that **ap_genrules** would not run out of memory. This is why, for this experiment, we ran our Java programs on a Sun Ultra-10 with 512MB RAM that did not have a JIT compiler, which is another important reason why the times taken by the algorithms became more noticeable than in the previous experiments.

In general, as expected, the performance of **CoverRules**, as well as that of **ap_genrules**, slows down when the databases are more dense, and when the number of maximal frequent itemsets increases. The algorithms performance vary differently with the change of minconf. For dense databases the size of the cover is one-two orders of magnitude smaller than the number of valid association rules and shows the tendency of getting smaller as the redundancy in the generated association rules increases.

DENSE database (minsupp = 5%)				
minconf	Valid rules	**ap_genrules** Time(seconds)	Cover	**CoverRules** Time(seconds)
90%	87722	9	8875	215
80%	344001	30	9375	236
70%	511191	46	9020	220
60%	574554	49	7878	178
50%	603861	50	6483	130
40%	630706	52	6506	133
30%	656724	51	5496	104
20%	682076	53	5674	99
10%	703373	52	3416	41
5%	703924	52	3181	37
1%	703924	52	3181	37

Figure 5: Results for DENSE database

5 Conclusions

We presented new inference rules for association rules and introduced the concepts of cover and informative cover for the set of valid association rules. We discussed the properties of these covers and their relation to the previous research done on the topic of bases for association rules. Finally, we presented algorithm **CoverRules** for the generation of an informative cover, and the results of our experiments confirmed the validity of our approach.

References

[AAP00] Ramesh C. Agarwal, Charu C. Aggarwal, and V.V.V. Prasad. Depth first generation of long patterns. In *Proceedings of the 6th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 108–118, 2000.

[AAP01] Ramesh C. Agarwal, Charu C. Aggarwal, and V.V.V. Prasad. A tree projection algorithm for generation of frequent itemsets. *Journal of Parallel and Distributed Computing, Special Issue on High Performance Data Mining*, 61(3):350–371, 2001.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[AS94a] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. RJ 9839, IBM Almaden Research Center, Almaden, California, 1994.

[AS94b] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases*, pages 487–499, 1994.

[BA99] Roberto J. Bayardo and Rakesh Agrawal. Mining the most interesting rules. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154, 1999.

[Bay98] Roberto J. Bayardo. Efficiently mining long patterns from databases. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 85–93, 1998.

[BM98] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[BMUT97] Sergey Brin, Rajeev Motwani, Jeffrey Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 255–264, 1997.

[BPT+00] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Proceedings of the First International Conference on Computational Logic*, pages 972–986, 2000.

[CCS00] Dana Cristofor, Laurentiu Cristofor, and Dan A. Simovici. Galois connections and data mining. *Journal of Universal Computer Science*, 6(1):60–73, 2000.

[CGL01] Frans Coenen, Graham Goulbourne, and Paul H. Leng. Computing association rules using partial totals. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 54–66, 2001.

[GD86] J.L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 24(95):5–18, 1986.

[HH99] Robert J. Hilderman and Howard J. Hamilton. Knowledge discovery and interestingness measures: A survey. Technical Report CS 99-04, University of Regina, Regina, Saskatchewan, Canada, 1999.

[HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[JS01] Szymon Jaroszewicz and Dan A. Simovici. A general measure of rule interestingness. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 253–265, 2001.

[LHM99] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125–134, 1999.

[Lux91] Michael Luxenburger. Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 29(113):35–55, 1991.

[PBTL99a] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Closed set based discovery of small covers for association rules. In *Proceedings of the 15th Conference on Advanced Databases*, pages 361–381, 1999.

[PBTL99b] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory*, pages 398–416, 1999.

[PBTL99c] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[PHM00] Jian Pei, Jiawei Han, and Runying Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[PT98] Balaji Padmanabhan and Alexander Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 94–100, 1998.

[PT00] Balaji Padmanabhan and Alexander Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 54–63, 2000.

[SCC00] Dan A. Simovici, Dana Cristofor, and Laurentiu Cristofor. Mining for purity dependencies in databases. TR 00-2, University of Massachusetts at Boston, Boston, Massachusetts, 2000.

[SLRS99] Devavrat Shah, Laks V. S. Lakshmanan, Krithi Ramamritham, and S. Sudarshan. Interestingness and pruning of mined patterns. In *1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1999.

[SON95] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *Proceedings of the 21st International Conference on Very Large Databases*, pages 432–444, 1995.

[ST95] Dan A. Simovici and Richard L. Tenney. *Relational Database Systems.* Academic Press, New York, 1995.

[TKR+95] Hannu Toivonen, Mika Klemettinen, Pirjo Ronkainen, Kimmo Hätönen, and Heikki Mannila. Pruning and grouping discovered association rules. In *Proceedings of the European Conference on Machine Learning Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, pages 47–52, 1995.

[Zak00] Mohammed J. Zaki. Generating non-redundant association rules. In *Proceedings of the 6th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34–43, 2000.

[ZH99] Mohammed J. Zaki and Ching-Jui Hsiao. Charm: An efficient algorithm for closed association rule mining. Technical Report 99-10, Renssaeler Polytechnic Institute, Troy, New York, 1999.