Nonredundant Generalized Rules and Their Impact in Classification

François Rioult, Bruno Zanuttini, Bruno Crémilleux

Nonredundant Generalized Rules and Their Impact in Classification

François Rioult, Bruno Zanuttini, and Bruno Crémilleux

Abstract. Association rules are commonly used in classification based on associations. These rules are made of conjunctions of attributes in the premise and a class attribute in conclusion. In this chapter, we are interested in understanding the impact of generalized association rules in classification processes. For that purpose, we investigate the use of generalized association rules, *i.e.*, rules in which the conclusion is a disjunction of attributes. We propose a method which directly mines nonredundant generalized association rules, possibly with exceptions, by using the recent developments in condensed representations of pattern mining and hypergraph transversals computing. Then we study the impact of using such rules instead of classical ones for classification purposes. To that aim, we view generalized rules as rules with negations in the premise and possibly concluding on a negative class attribute. To study the impact of such rules, we feed the standard CMAR method with these rules and we compare the results with the use of classical ones.

1 Introduction

Supervised classification is the well-known task of predicting classes. In such an approach, classifiers are built from a set of labelled data and then are used to predict the class of new objects. There are a lot of classification methods and recently, many contributions based on association rules [23, 22, 5, 33] have been proposed. The quality of the results is high (85% classification score in average on the UCI benchmarks), and it becomes difficult to do better with the same classifier on a lot of data sets. This explains the development of many original works for optimizing the process, and why recent propositions are more and more complex.

François Rioult · Bruno Zanuttini · Bruno Crémilleux
GREYC, CNRS - UMR 6072, Université de Caen Basse-Normandie
F-14032 Caen cedex, France
e-mail: FirstName.LastName@unicaen.fr

Except for few works [3, 6], these methods use *classical* association rules, *i.e.*, rules of the form $X \rightarrow c$ where X is a set of attributes and c is a class value. Such a rule is used for prediction: when the conjunction of attributes X is satisfied, then the class value c is predicted. Basically, current approaches mine all classification rules satisfying some interestingness measures (e.g., frequency, confidence), then classifiers are designed by selecting some of these candidate classification rules. Surprisingly, there are very few attempts for handling the natural generalization of association rules to rules with a disjunction in the conclusion $X \rightarrow \vee Y$. We will see in Section 5 that, for classification purposes, these rules can be viewed as rules with negations of attributes in the premise (*i.e.*, rules of the form $X \wedge \overline{Y} \rightarrow c$) and possibly in the conclusion (rules $X \wedge \overline{Y} \rightarrow \overline{c}$). In the first case, the class c is prescribed for the objects containing X but no attribute of Y, whereas in the second case this class is excluded. In this context, a particular case of rules excluding classes is proposed in [3]: rules are built from patterns evaluated as interesting w.r.t. classes according to a statistical measure.

Such a lack on using generalized association rules may be explained by the hard task of mining these rules. In this chapter, we propose a complete and correct algorithm mining all frequent exact generalized association rules with a minimal premise and conclusion, by using the recent developments in condensed representations of patterns and hypergraph transversals computing. Generalized rules offer an extended semantics as compared to classical rules, and it is natural to expect that they improve classification based on associations. Studying this question is the goal of this chapter. Our aim is to evaluate the impact of disjunction in classification based on association rules. For that purpose, we rewrite generalized rules into rules with negations and we investigate the impact of using such rules instead of classical ones in the classification process. This task is performed by using the CMAR classification method [22] because it is considered as a reference in the area of associative classification. We compare the results obtained with CMAR when using either classical or generalized association rules. Since only the set of rules is modified, this enables us to estimate the impact of disjunction.

The contribution of the chapter is twofold. First, we propose a method to mine the whole set of frequent exact generalized association rules with minimal premise and conclusion, possibly with exceptions. The property of minimality is a key point in building classifiers: it is well-known with classical rules [7] and we will see that it is also the case for generalized association rules. Second, we study the extended semantics induced by disjunction and its impact on classification based on associations. We use the CMAR technique as a reference, adapting it for using generalized rules. We insist that our goal is not to propose the "best" classification method, but to study the interest of enhancing classification methods based on association rules with generalized rules. We evaluate this impact on numerous UCI benchmarks, and our results show that the use of disjunction is not so simple.

This chapter is organized as follows: we first recall the definition of classical association rules, then we sketch the supervised classification methods using them, and introduce generalized association rules (Section 3). Then we present a correct and complete method for extracting the set of nonredundant generalized rules

(Section 4). We show how they can be used in supervised classification, and we present our experiments with CMAR enhanced with generalized rules (Section 5). Finally, based on these experiments, we discuss the impact of generalized rules (Section 6).

2 Association Rules and Classification

This section provides the background on association rules and classification based on associations (AR-based classification) which is required for this chapter.

2.1 Database and Patterns

A *database* is defined by a *boolean context* $r = (\mathscr{A}, \mathscr{O}, \mathscr{R})$, where $\mathscr{O}$ is a set of objects, $\mathscr{A}$ a set of boolean attributes and $\mathscr{R}$ a binary relation between the objects and the attributes. Table 1 gives an example of database. An object is also a subset of $\mathscr{A}$. For instance, the object $o_1 = \{a_1, a_3, a_5\}$ is noted $a_1a_3a_5$. For supervised classification, each object is labelled by a single-valued class attribute c belonging to $\mathscr{C} = \{c_1, \ldots, c_n\}$, a subset of the attribute set. In our running example (cf. Table 1), the class labels are a_1 and a_2, thus also noted c_1 and c_2.

Table 1 Example of boolean context r

	attributes						
objects	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_1	×		×		×		
o_2		×	×		×		
o_3	×		×		×		
o_4	×			×		×	
o_5		×	×			×	
o_6		×	×			×	
o_7	×			×			×
o_8		×		×			×
classes	c_1	c_2					

A *pattern* X is a subset of $\mathscr{A}$, its *support* in r is the set of objects containing X (we note $supp(X) = \{o \in \mathscr{O} \mid X \subseteq o\}$) and its *frequency* is $\mathscr{F}(X) = |supp(X)|$ *i.e.*, the number of objects in its support. In Table 1, $supp(a_2a_3) = o_2o_5o_6$ and its frequency is 3.

2.2 Association Rules

Let X and Y be two nonempty and disjoint patterns. A *classical* association rule [2] is an expression $X \rightarrow Y$, where X is called the *premise* and Y is called the *conclusion*.

In such a rule, both X and Y are interpreted conjunctively, that is, the rule is read $(x_1 \wedge \cdots \wedge x_k) \rightarrow (y_1 \wedge \cdots \wedge y_d)$ where $X = \{x_1, \ldots, x_k\}$ and $Y = \{y_1, \ldots, y_d\}$, and it says that (up to its confidence) the objects which support X also support Y. The quality of such an association rule is evaluated by interestingness measures (computed relatively to a database) such as *frequency* ($\mathscr{F}(X \rightarrow Y) = \mathscr{F}(X \cup Y)$) and *confidence* ($conf(X \rightarrow Y) = \mathscr{F}(X \cup Y)/\mathscr{F}(X)$). Intuitively, the frequency of a rule is the number of objects for which it fires and is true, and its confidence is the conditional probability that it is true for an object which supports X. In our example, the rule $a_1a_3 \rightarrow a_5$ has a frequency of 2 and its confidence is 1 (such a rule is said an *exact* association rule), whereas $a_2a_3 \rightarrow a_6$ has a frequency of 2 and its confidence is 2/3.

2.3 Classification Based on Association Rules

Supervised classification aims at building a classifier from a *training set* in order to predict the class attribute of unseen examples.

In supervised classification, one usually has to cope with two main issues. Overfitting is the most well-known one. A classifier is said to *overfit* the training set when it uses rules too specific to this set. In this case, the rules are very efficient on the training set but are not able to generalize the prediction process to unseen examples. In the area of association rules, restricting to nonredundant rules with high enough frequency limits this problem [7]. The second issue is unbalanced class populations [17]. It happens when some classes contain many more objects than some others. In this case, classifiers often tend to focus on the prediction of the prevailing class(es). To cope with this difficulty, constraints on the coverage of the training set are introduced [22]. For instance, with association rules, a rule is kept for classification only if it classifies at least one object which is not covered by the rules already selected.

In associative classification, each rule may contribute to the decision. Since rules with different conclusions can be triggered by a same object, a vote schema is usually used. Typically, rules are weighted by measures such as χ^2, which evaluates the correlation between the premise and the conclusion. This technique is used for instance by the CMAR method [22]. As there are a lot of measures on rules (e.g., frequency, confidence, conviction, χ^2) [27] and these measures may lead to different behaviors, this one difficulty is to choose the right measure.

2.4 State of the Art of AR-Based Classification

The CBA method [23] is the pioneer proposal. This method uses the classical interestingness measures of frequency and confidence: rules are first ranked according to their confidence, then according to their frequency. Several other methods improve this rule selection process. CMAR [22] weights rules with a normalized χ^2 between the premise and the conclusion, then selects multiple rules by learning set coverage. As already said, CMAR is often considered as a reference method in associative

classification. L3 [4] extends the cover principle by using the excluded rules when no selected rule is triggered. L3m [5] improves L3 by using multiple rules. Classical methods (*i.e.* CBA, CMAR) avoid redundancy by selecting the rules with the best confidence. If two rules have the same confidence, then the one with the highest frequency is kept. Then, if their frequencies are the same, the rule having the shortest premise is selected[1]. These operations are performed by filtering the output of an association rule mining algorithm.

As far as we know, in the context of data mining, there has been no study on using rules with a disjunctive conclusion for classification purposes, as we do here. There have been some works, though, about the use of rules with negative attributes, which is related to disjunction (see Section 4.1). For instance, [3] proposes a heuristic technique for mining some specific kind of rules with negations. The technique extracts rules of the form $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$, or $\overline{X} \rightarrow \overline{Y}$ when the correlation of $X \rightarrow Y$ is negative. This technique does not ensure that the complete set of rules is mined, but this issue has been solved recently in [29]. In [6], an enumerative approach is used to build the rules with negations by adding attributes to the premise.

Current tendencies focus on selecting the useful rules more precisely, so that the expert can access them [35]. Optimization processes also allow classifiers to learn how to choose the useful rules [35]. Since the cover principle depends on the order of the objects, HARMONY [30] proposes an instance-centric alternative and optimizations for large datasets. Finally, since association rules mining is costly, associative classification can benefit of nonredundant rule extraction [25, 36]. A first contribution in this direction is given in [10].

Comparing the different approaches is complex for several reasons. On the one hand, all prototypes are not available, and experiments are not always reproducible. On the other hand, the articles compare the performances to those of universal classifiers, such as C4.5, Foil, Ripper or CPAR, but not always to other contributions using association rules. In practice, the comparisons are often made according to the official scores given in the reference articles.

3 Generalized Association Rules

3.1 Definitions

In this chapter, we are interested in the generalization of association rules allowing disjunction in the conclusion. From a logical point of view, classical association rules can be seen as Horn rules/clauses of the form $X \rightarrow y$, while generalized association rules can be seen as generalized rules/clauses of the form $X \rightarrow (y_1 \vee \cdots \vee y_d)$. So, due to the well-known fact that any propositional formula is logically equivalent to a conjunction of clauses, any association, possibly involving negative attributes and complex Boolean combinations in the premise and the conclusion, can

[1] This order is referred to as CSA (Confidence, Support, size of Antecedent) in [31].

be represented as a set of generalized association rules[2]. So, from a logical point of view, generalized association rules are fully expressive, contrary to classical rules. Of course, we will not use this expressive power to overfit the data in the training set, but instead we will select rules among this bigger set of rules.

As we shall see in more details in Section 5.1, a positive attribute in the conclusion is logically equivalent to the negation of this attribute in premise: we have $(X \rightarrow y \vee Y) \equiv (X \wedge \overline{y} \rightarrow \vee Y)$, and dually $(X \wedge x \rightarrow \vee Y) \equiv (X \rightarrow \overline{x} \vee Y)$. In this section, though, we stick to the definition with disjunction, since both visions differ when it comes to defining support and confidence. We first defined *generalized* rules (they are named *disjunctive* rules in [11]).

Definition 1 (generalized association rule). *Let X, Y be two nonempty and disjoint patterns. A* generalized association rule *based on $Z = X \cup Y$ is an expression $X \rightarrow \vee Y$. Its* depth *is the number of attributes in Y. Given a database, the* frequency *of $X \rightarrow \vee Y$ is $\mathscr{F}(X)$, and it is* exact *if every object which supports X also contains at least one attribute of Y.*

Example 1. In Table 1, each object containing a_6 contains a_2 or a_4, so the rule $a_6 \rightarrow a_2 \vee a_4$ is exact.

Observe in Definition 1 that the frequency of a generalized rule is not defined as for classical rules. Indeed, for a generalized rule $X \rightarrow \vee Y$, it would make no sense to reason on the frequency of $X \cup Y$, because different objects can support different subsets of Y. Note however that when Y contains only one attribute, the rule $X \rightarrow \vee Y$ is equivalent to the classical rule $X \rightarrow Y$ in the sense that one is exact in a database if and only if the other also is, and their frequencies are the same.

We now define nonexact generalized rules. In the context of classical rules, it is well-known that exact rules are not sufficient for classification purposes, in particular because they tend to overfit the training data and do not allow to cope with noise [7]. There are two main ways of defining nonexact rules. The first one is to consider rules with *confidence* less than 1, that is, with some probability of being false on an example, where the probability is fixed for the whole set of rules. The second way, which we follow here, is to allow a fixed number of *exceptions* to the rules. So, the difference between both approaches is as follows. If the confidence is fixed, then the higher the frequency of a rule, the greater the number of exceptions to it allowed. Dually, if the number of exceptions is fixed, the higher the frequency of a rule, the greater the confidence required for it.

Definition 2 (δ-approximative association rule). *Let X, Y be two nonempty and disjoint patterns. The generalized association rule $X \rightarrow \vee Y$ is said to be* δ-approximative *if $|\{o \in O \mid X \subseteq o \wedge o \cap Y = \emptyset\}| \leq \delta$.*

Example 2. On the toy example (Table 1), the rule $a_6 \rightarrow a_2$ is 1-approximative : each object supporting a_6 supports also a_2, except o_4.

[2] Observe that the transformation of an arbitrary association into a set of general rules may yield an exponential increase in size.

We will discuss the impact of considering exceptions instead of confidence in Section 5.2 and 6.3. Observe however that in both definitions, an exact generalized rule already summarizes nonexact classical rules. For example (Table 1), the generalized rule $a_2a_3 \rightarrow a_5 \vee a_6$ is exact (with frequency 3), and it summarizes the two classical rules $a_2a_3 \rightarrow a_5$ (frequency 1 and confidence 1/3) and $a_2a_3 \rightarrow a_6$ (frequency 2 and confidence 2/3). So, in some sense, using disjunction in the conclusion already involves some considerations of nonexactness.

We wish to emphasize here that, though a generalized rule, say, $X \rightarrow y_1 \vee \cdots \vee y_d$, can be seen as the rule with negation, say, $X \wedge \overline{y_1} \wedge \cdots \wedge \overline{y_{d-1}} \rightarrow y_d$, this is true only from a logical point of view. Indeed, in general the frequency of both rules will be different. For our running example, the rule $a_3 \rightarrow \vee a_2 \vee a_5$ is exact and its frequency is 5, but the frequency of the corresponding rules with negation $a_3 \wedge \overline{a_5} \rightarrow a_2$ is only 3. So our approach focuses on supports and exactness defined on the presence of attributes, while rules with negations consider the absence of an attribute at the same level as the presence of one.

3.2 Nonredundant Generalized Rules

We now define several notions of nonredundancy for generalized rules. Such notions (minimal, free premise) have been defined for classical rules and have proven to be useful, especially because focusing on nonredundant rules drastically reduces the number of rules (to be extracted and stored) without impacting classification [10]. Our definitions are rather straightforward generalizations of those for classical rules.

Definition 3 (minimal conclusion). *A generalized association rule $X \rightarrow \vee Y$ is said to have a* minimal conclusion *in a database r if there is no rule $X \rightarrow \vee Y'$ with $Y' \subset Y$ and admitting the same number of exceptions as $X \rightarrow \vee Y$ in r.*[3]

Definition 4 (minimal premise). *A generalized association rule $X \rightarrow \vee Y$ is said to have a* minimal premise *in a database r if there is no rule $X' \rightarrow \vee Y$ with $X' \subset X$ and $\mathscr{F}_r(X') = \mathscr{F}_r(X)$.*[4]

As in the classical case, it turns out that minimal premises of rules coincide with free patterns (defined below). We will use this property for reusing well-known extraction techniques in Section 4.3, using the important fact that the notion of freeness is independent from the rule under consideration. Recall that a (classical) rule is said to be *based on X* if it is of the form $X' \rightarrow X''$ with $X' \cup X'' = X$. Intuitively, a pattern is free in a database if its frequency is less than that of any of its subsets.

Definition 5 (free/key pattern [11, 25]). *A pattern X is said to be* free *in a database r if there is no classical rule which is based on X and exact in r.*

[3] Observe that by definition of frequency for generalized rules, $X \rightarrow \vee Y$ and $X \rightarrow \vee Y'$ always have the same frequency.

[4] Observe that by definition, if there was such a rule, then the objects supporting X and X' would be the same and so, rules $X \rightarrow \vee Y$ and $X' \rightarrow \vee Y$ would have the same number of exceptions.

Proposition 1. *An exact generalized association rule $X \rightarrow \vee Y$ in a database r has a minimal premise if and only if X is free in r.*

Proof. The proof works as in the classical case. If $X_1 \rightarrow X_2$ is an exact rule based on X in r, then every object supporting X_1 also supports $X_1 \cup X_2 = X$. So $\mathscr{F}_r(X_1) = \mathscr{F}_r(X)$ while $X_1 \subsetneq X$, and $X \rightarrow \vee Y$ does not have a minimal premise.

Conversely, if $X \rightarrow \vee Y$ does not have a minimal premise, then let $X' \rightarrow \vee Y$ be a rule such that $X' \subset X$ and $\mathscr{F}_r(X') = \mathscr{F}_r(X)$. From $\mathscr{F}_r(X') = \mathscr{F}_r(X)$ we get that every support for X' is one for X, and so that the classical rule $X' \rightarrow X \setminus X'$ is exact in r. So X is not free in r.

Based on these well-defined notions of nonredundancy, we can now define the notion of *irredundant generalized rule*. From the definitions above it follows that for every generalized rule $X \rightarrow \vee Y$ in r, there is an irredundant generalized rule of the form $X' \rightarrow \vee Y'$ with $X' \subset X$, $Y \subset Y'$, and the same frequency and number as exceptions as $X \rightarrow \vee Y$ in r. So, irredundant generalized rules cover all generalized rules, just as is the case for classical rules.

Definition 6 (irredundant generalized rule). *A generalized rule $X \rightarrow \vee Y$ is said to be* irredundant *in a database r if it has a minimal premise and a minimal conclusion in r.*

4 Mining Generalized Rules

In this section, we give our proposal for mining generalized (frequent, irredundant) rules from a database r. We first survey existing approaches for tasks related to this one, then give the necessary background on hypergraph transversals, and finally give our algorithm.

4.1 Existing Approaches

Mining generalized rules under a frequency constraint is a hard task. A naive approach is to add the negation of each attribute as a new attribute to the database, then compute classical rules. This leads to rules with negation (e.g., $X \wedge \overline{Y} \rightarrow y$) which in turn can be seen as disjunctive rules ($X \rightarrow y \vee Y$). Nevertheless, with this approach the frequency computed is that of $X \wedge \overline{Y}$, which is different in general from that of $X \rightarrow y \vee Y$ (defined to be the frequency of X). Moreover, this approach leads to very dense databases, which are intractable for extraction of classical rules [9].

In the same vein, one can find a lot of contributions about mining rules with negation: [12, 28] restrict the conclusion to one attribute, [3] focuses on classical rules $X \rightarrow Y$ and considers rules of the form $X \rightarrow \bigwedge \overline{Y}$ (the conclusion is a conjunction of negative attributes), $\overline{X} \rightarrow Y$, or $\overline{X} \rightarrow \bigwedge \overline{Y}$, depending on the correlation between X and Y in the database. [34] uses the same heuristic, but the rules are computed

using a taxonomy of equivalent attributes. Some other approaches are restricted to a conjunction of negated attributes in the conclusion: [32] uses an anti-monotonic measure for pruning the search space, [13, 29] introduce one by one the negated attributes which are not present in the premise, and [14] uses two frequency bounds for frequent and infrequent patterns.

To the best of our knowledge, the only approach mining generalized frequent patterns (and, more generally, arbitrary Boolean expressions) is given in [38], but it does not focus on generalized rules nor on classification. This work mines nonredundant patterns with the help of a closure operator related to the maximality of a pattern for the support equivalence class. Our approach is different because our notion of nonredundancy is rule-oriented, and we do not need only exact expressions. Nevertheless, we also use minimal transversals for computing nonredundant conclusions [26].

In the field of logic, efficient algorithms are available, but they are not adequate for classification based on association rules. For example, [37] gives a quadratic algorithm for mining a set of generalized rules with minimal premises and conclusions, from which any other rule follows by the standard resolution rule. But the resulting set gives a covering of all (exact) associations in the dataset, and so exactly fits the dataset, which is not relevant for prediction purposes.

4.2 Hypergraph Transversals

We now present hypergraph transversals (also known as *hitting sets*). Briefly speaking, a hypergraph is a set of patterns, and a transversal is a pattern intersecting with each element of the hypergraph. Computing minimal hypergraph transversals has been widely studied [15] and is related to many other problems in the field of data mining [18]. We use here algorithms which compute minimal transversals for computing the (nonredundant) conclusions of generalized association rules (see Section 4.3).

A hypergraph $\mathscr{H} = (V, \mathscr{E})$ is a couple of vertices and hyperedges. This notion generalizes the notion of a graph, because hyperedges are subsets of V (instead of pairs in a graph). We define hypergraphs starting from a Boolean context.

Definition 7 (hypergraph). *Let $r = (\mathscr{A}, \mathscr{O}, \mathscr{R})$ be a Boolean context. Then r defines the hypergraph $\mathscr{H}_r$ whose vertices are the attributes $\mathscr{A}$ of r and whose hyperedges are the patterns corresponding to the objects of r.*

Conversely, a Boolean context $r_{\mathscr{H}}$ can be defined starting from a hypergraph $\mathscr{H}$.

Definition 8 (transversal). *Let $\mathscr{H} = (V, \mathscr{E})$ be a hypergraph. A set of vertices $Y \subseteq V$ is a* transversal *of $\mathscr{H}$ if for all $E \in \mathscr{E}$, $E \cap Y \neq \emptyset$ holds. It is a* minimal *tranversal of $\mathscr{H}$ if moreover, no $Y' \subset Y$ is a transversal of $\mathscr{H}$. It is a* minimal δ-approximative *transversal of $\mathscr{H}$ if moreover $|\{E \in \mathscr{E} \mid Y \cap E = \emptyset\}| \leq \delta$, and for no $Y' \subset Y$ we have $|\{E \in \mathscr{E} \mid Y' \cap E = \emptyset\}| = |\{E \in \mathscr{E} \mid Y \cap E = \emptyset\}|$.*

Example 3. The hypergraph corresponding to our example is $\mathscr{H} = (V, \mathscr{E})$ with $V = \{a_1 \dots a_7\}$ and $\mathscr{E} = \{a_1a_3a_5, a_2a_3a_5, \dots a_2a_4a_7\}$. $a_1 \dots a_7$ is a trivial transversal of $\mathscr{H}$, but it is not minimal. $a_2a_4a_5$ is a minimal one, and a_2a_4 is 2-approximative.

The problem of mining the minimal transversals of a hypergrah is equivalent to the MOnotone Normal Form Equivalence Test (MONET), and as such it is at the core of many practical applications in logic, graph theory, data mining, *etc.* [19]. Its algorithmical aspects are very interesting, because it is a good candidate as a problem separating **P** from **NP**.

The main algorithm for computing the transversals of a hypergraph are detailed in [19], where it is also shown that none of them is output-polynomial. Fredman-Kachyian's algorithm [16] has the best theoretical time bound ($n^{o(\log n)}$), but practical experiments make it hard to choose the best algorithm overall. In this paper, we use our proposal from [20], called MTMINER. This algorithm is original in the way that it mines the tranversals, namely in a levelwise manner. This is an essential aspect which we need for mining special kinds of tranversals: minimal of course, but also with bounded length or with exceptions.

4.3 Mining Nonredundant Generalized Rules

We now present our method for mining nonredundant generalized association rules in a database r. To that aim, we first establish the link between these rules and the transversals of the hypergraph defined from r, and then give our algorithm.

Definition 9 (pattern restriction). *Let $r = (\mathscr{A}, \mathscr{O}, \mathscr{R})$ be a boolean context. Let X be a pattern. Then the* restriction of r to X*, written $r[X]$, is the database $(\mathscr{A}', \mathscr{O}', \mathscr{R}')$ where $\mathscr{A}' = \mathscr{A} \setminus X$, $\mathscr{O}' = supp_r(X)$, and $\mathscr{R}'$ is the restriction of $\mathscr{R}$ to $\mathscr{A}' \times \mathscr{O}'$.*

Informally, the restriction of r to X is the multiset of all objects described by the pattern X' such that the pattern $X \cup X'$ describes an object in r.

Proposition 2. *Let r be a database and X be a pattern. The patterns Y, such that $X \rightarrow \vee Y$ is an exact (resp. δ-approximative) generalized association rule in r, are exactly the* transversals *(resp. δ-approximative transversals) of the hypergraph defined from $r[X]$.*

Proof. Assume $X \rightarrow \vee Y$ is exact in r. Then by definition of exactness, every object in r which supports X also contains at least one attribute of Y. From Definition 1, X and Y are disjoint. It follows that every object in $r[X]$ contains at least one attribute in Y. This is equivalent to say that the hypergraph $r[X]$ has Y as a transversal. The converse is shown similarly, and so is the case of δ-approximative rules.

Using Propositions 1 and 2, we derive the following corollary, which is at the core of our method for mining irredundant generalized association rules.

Corollary 1. *Extracting the exact (resp. δ-approximative) irredundant generalized association rules with frequency at least γ in a database r amounts to:*

1. *extract all patterns X which are free and with frequency at least γ in r,*
2. *for each such X, extract the minimal transversals (resp. δ-approximative minimal transversals) Y of the restriction $r[X]$,*
3. *for each such couple (X,Y), generate the generalized association rule $X \rightarrow \vee Y$.*

Using the results above, the $l\delta$-`miner` algorithm (depicted as Algorithm 1) mines the nonredundant frequent generalized classification rules. Since we will use such rules for classification purposes, the algorithm restricts to the rules which contain a class attribute in premise or in conclusion.

Algorithm 1. $l\delta$-`miner`- Mining nonredundant generalized classification rules.

Data: a boolean context r, a minimum frequency threshold γ, and a maximum number of exceptions δ ($\delta = 0$ for exact rules).
Result: the set $\mathscr{RULES}$ of all generalized classification rules which are nonredundant, with frequency at least γ, and with at most δ exceptions in r.

```
1  RULES ← ∅;
2  foreach pattern X frequent and free in r do
3      if ∃c ∈ C | c ∈ X then
4          foreach Y minimal δ-approximative transversal of H_{r[X]} do
5              add X → ∨Y to RULES;
6          end
7      else
8          foreach Y minimal δ-approximative transversal of H_{r[X]} containing
           c ∈ C do
9              add X → ∨Y to RULES;
10         end
11     end
12 end
```

We now give details for mining the free patterns and the minimal transversals, as required by Algorithm 1.

Since freeness and frequency constraints are anti-monotonic, we use a levelwise approach for mining the premises of the rules. This strategy allows for efficient pruning conditions [24]. The method is depicted as Algorithm 2, where *apriori_gen* is the classical procedure for producing next-level candidates from the current-level frequent and free patterns [1]. In a nutshell, *apriori_gen* generates $X \cup \{a,b\}$ from $X \cup \{a\}$ and $X \cup \{b\}$ if all subsets of $X \cup \{a,b\}$ are frequent and free. As compared to the classical `apriori`-like frequent pattern mining algorithm [1], Algorithm 2 only adds the freeness constraint at Step 3.

Algorithm 2. Mining free patterns.

Data: a boolean context $r = (\mathscr{A}, \mathscr{O}, \mathscr{R})$, a minimum frequency threshold γ.
Result: the set $\mathscr{F}ree$ of all patterns which are free and with frequency at least γ in r.

$k \leftarrow 1$;
$\mathscr{C}and_1 \leftarrow \{\{a\} \mid a \in \mathscr{A}\}$;
repeat
  $\mathscr{F}ree_k \leftarrow \{X \in \mathscr{C}and_k \mid \mathscr{F}(X) \geq \gamma \wedge (\forall X' \subsetneq X, \mathscr{F}(X) < \mathscr{F}(X'))\}$;
  $k++$;
  $\mathscr{C}and_k \leftarrow apriori\_gen(\mathscr{F}ree_k)$;
until $\mathscr{F}_k = \emptyset$;
return $\mathscr{F}ree = \bigcup_{i=1..k} \mathscr{F}ree_i$;

Mining the minimal transversals (Algorithm 3) uses the same search algorithm, but with the *anti-frequency* instead of the frequency constraint. Anti-frequency is defined by $\overline{\mathscr{F}}(X) = |\{o \in \mathscr{O} \mid X \cap o = \emptyset\}|$. Details and proofs for Algorithm 3 can be found in [20]. Let us simply give the intuition that the anti-frequency constraint is anti-monotonic, and that when it is null the corresponding pattern is a tranversal. Generators, whose anti-frequency is not strictly less than that of any of their subsets, are not minimal and are pruned.

Algorithm 3. Mining minimal transversals.

Data: a hypergraph $\mathscr{H} = (V, \mathscr{E})$ and a maximum number of exceptions δ ($\delta = 0$ for usual transversals).
Result: the set $\mathscr{M}inTr$ of δ-approximative minimal transversals of $\mathscr{H}$.

$k \leftarrow 1$;
$r_{\mathscr{H}} = (a, \mathscr{O}, \mathscr{R}) \leftarrow$ the Boolean context deduced from $\mathscr{H}$;
$\mathscr{C}and_1 \leftarrow \{\{a\} \mid a \in \mathscr{A}\}$;
repeat
  $\mathscr{M}inTr_k \leftarrow \{X \in \mathscr{C}and_k \mid \overline{\mathscr{F}}(X) \leq \delta\}$;
  $k++$;
  $\mathscr{C}and_k \leftarrow apriori\_gen(\{X \in \mathscr{C}and_{k-1} \mid$
    $\overline{\mathscr{F}}(X) \leq \delta \wedge (\forall X' \subsetneq X, \overline{\mathscr{F}}(X) < \overline{\mathscr{F}}(X'))\}$;
until $\mathscr{M}inTr_k = \emptyset$;
return $\mathscr{M}inTr = \bigcup_{i=1..k} \mathscr{M}inTr_i$;

When compared to the classical minimal transversals algorithms, the interest of Algorithm 3 lies in the fact that it can easily provide minimal transversals with constraints. The version presented on Figure 3 integrates the constraint on the number of exceptions, but we also used this algorithm with a constraint on the length of minimal transversals/depth of generalized rules for our experiments (see Section 5.2). Indeed, since the algorithm browses the search space in a levelwise manner (where the current level is the length of the transversals generated), it can be straightforwardly restricted to a maximum length.

5 Using Generalized Rules in a Classification Process

This section presents our experiments and results about the impact of using generalized rules (instead of classical ones) in a classification process, such rules being provided by Algorithm 1. We first show how a generalized rule can be seen as a rule prescribing or excluding a class value, possibly with negations in the premise and/or the conclusion. It enables us to handle experiments with the CMAR method for AR-based classification.

5.1 From Generalized Rules to Classification Rules

The link from generalized rules to classification rules is done via the following definition and proposition.

Definition 10 (classification rule). *Let $o \in \mathcal{O}$ be an object, X, Y be two nonempty and disjoint patterns, and let $c \notin (X \cup Y)$ be a class value. The generalized rule $X \cup \{c\} \rightarrow \vee Y$ is said to* exclude *the class c for o if $X \subseteq o$ and $o \cap Y = \emptyset$. The generalized rule $X \rightarrow \vee (Y \cup \{c\})$ is said to* prescribe *the class c for o if $X \subseteq o$ and $o \cap Y = \emptyset$.*

Say that a rule with negations of the form $X \wedge \overline{Y} \rightarrow c$ (resp. $X \wedge \overline{Y} \rightarrow \overline{c}$) is exact in r if all objects in r which support all attributes of X but none of Y also support c (resp. do not support c), and similarly when exceptions are allowed. Then we have the following.

Proposition 3. *Let X, Y be two nonempty and disjoint patterns, and let $c \notin (X \cup Y)$ be a class attribute. The rule $X \wedge \overline{Y} \rightarrow c$ (resp. $X \wedge \overline{Y} \rightarrow \overline{c}$) is exact in a database r if and only if the rule $X \rightarrow \vee (Y \cup \{c\})$ (resp. $(X \cup \{c\}) \rightarrow \vee Y$) is exact in r. The same holds for δ-approximative rules.*

Proof. We consider δ-approximative rules, since exact rules are a particular case of them. The rule $X \wedge \overline{Y} \rightarrow c$ has less than δ exceptions in r if and only if (by definition) we have $|\{o \in r \mid X \subseteq o, o \cap Y = \emptyset, c \notin o\}| \leq \delta$. This is exactly the same as $|\{o \in r \mid X \subseteq o, X \cap (Y \cup \{c\}) = \emptyset\}| \leq \delta$, which is the definition of $X \rightarrow \bigvee (Y \cup \{c\})$ being δ-approximative in r.

Now to the case of $X \wedge \overline{Y} \rightarrow \overline{c}$. This rule has less than δ exceptions in r if and only if we have $|\{o \in r \mid X \subseteq o, o \cap Y = \emptyset, c \in o\}| \leq \delta$, that is, $|\{o \in r \mid X \cup \{c\} \subseteq o, o \cap Y = \emptyset\}| \leq \delta$, that is, $(X \cup \{c\}) \rightarrow \vee Y$ is δ-approximative in r.

5.2 Using Generalized Rules in CMAR

We now present how we have adapted CMAR [22] in order to handle nonredundant generalized rules, seen as rules with negations prescribing or excluding a class. By only changing the type of the used rules, we can understand the impact of nonredundant generalized rules in classification. We use CMAR because this method is considered as a reference in the area of associative classification. Its performances are great, and it is technically very complete: nonredundant rules, χ^2 selection, cover principle and multiple rules vote.

For a new object, some rules prescribe or exclude one or several classes, and each rule contributes positively or negatively with its measure. In CMAR, a weighted χ^2 quantifies the correlation between the premise X and the conclusion c (in the case of a classical rule $X \rightarrow c$). This χ^2 score is obtained from the contingency table between X and c.

Now consider the case of a generalized rule $X \wedge \overline{Y} \rightarrow c$. To tackle such a rule, we use a natural adaptation with a contingency table between $X \wedge \overline{Y}$ and c. A rule with negations, prescribing c, will be interesting if c is predominantly present with X. In order to measure this interest, we compute a *local* χ^2 on the only objects containing X, between $\overline{Y}$ and c. This follows the same intuition as that for the frequencies of generalized rules, which in our approach is computed on the presence of attributes (see the end of Section 3.1).

After rules have been selected by training set coverage, the vote schema is straightforward: if an object matches the premise of a rule prescribing (resp. excluding) a class c, then this class receives a positive (resp. negative) contribution from the χ^2. The final decision is to prescribe the class with the highest sum of contributions.

Avoiding over-fitting. Generalized rules turn out to be uninteresting for classification if their conclusion is too large: this leads to overfitting. So, in practice we only extract generalized rules of depth limited to 1 (classical rules), 2, or 3 attributes. Experiments show that larger conclusions are useless. Moreover, since the complexity of Algorithm 1 lies in computing the minimal transversals, restricting the depth of rules (the length of transversals) also lightens the computation.

Allowing exceptions in classical rules. We compare the results of CMAR with generalized rules to those with classical rules. Classical association rules are usually mined under frequency and high confidence constraints. Although we use the frequency threshold, we select (classical) rules with a small number of exceptions, rather than with high confidence. More precisely, let $\delta \geq 0$ be a user-specified number of tolerated exceptions (e.g. $0, 1, 2, 3$ in our experiments): we select only the classical rules $X \rightarrow Y$ such that $\mathscr{F}(X) - \mathscr{F}(X \cup Y) \leq \delta$ (their confidence is over $1 - \frac{\delta}{\mathscr{F}(X)}$, and thus high). Moreover, this kind of rules maximizes a large variety of interest measures, obviously the confidence, but also the lift, Laplace, Jaccard, conviction, *etc.*[21]. The confidence is not defined for generalized rules, so we use a fixed number of exceptions instead. Section 6.3 shows that the adaptation of the confidence framework to the use of exceptions is running.

5.3 Results

Our classification method is named $l\delta$-`miner`: l is the depth of the rules and δ is the number of authorized exceptions. The performances have been evaluated on UCI benchmarks [8], the minimum frequency threshold is set to 1, 2 or 5%, δ goes from 0 to 3 and the depth l varies from 1 to 3. The $l\delta$-`miner` parameters reported in Table 2 are those who gave the best score.

This table shows the classification scores after a 10-cross-validation on several benchmarks for the reference methods C4.5, CBA et CMAR and gives the results for $l\delta$-`miner` and the corresponding parameters. The columns tell the characteristics of the datasets and the $l\delta$-`miner` parameters for the best score over the experiments:

dataset :	the name of the dataset	type :	the type of the rules: + for prescribing rules, - for excluding rules, = for the union of both
cl :	the number of classes	l :	the conclusion length
obj :	the number of objects	δ :	the maximum number of tolerated exceptions
attr :	the number of attributes	γ :	the minimum frequency threshold (in %)
c45 :	the C4.5 score [22]		
cba :	the CBA score [23]		
cmar :	the CMAR score [22]		
$l\delta$-`miner`:	the score of our method		

The last column named `time` tells how much time it took for the whole cross-validation process. This time gives an indication about the hardness of the task, which mainly lies in rule mining.

Table 2 Classification scores

	characteristics			reference methods			$l\delta$-	parameters				time
dataset	cl	obj	attr	c45	cba	cmar	`miner`	type	l	δ	γ	(sec.)
anneal	6	898	73	94.8	97.9	97.3	93.5	=	2	0	1	200
austral	2	69	55	84.7	84.9	86.1	87.5	+	1	1	2	65
auto	5	202	137	80.1	78.3	78.1	81.6	=	1	0	1	1064
breast	2	699	26	95.0	96.3	96.4	95.4	+	2	3	1	10
cleve	2^5	303	43	78.2	82.8	82.2	83.8	+	1	2	5	11
crx	2	690	59	84.9	84.7	84.9	86.5	+	1	1	1	189
german	2	1000	76	72.3	73.4	74.9	74.5	+	1	1	1	1930
glass	6^6	214	34	68.7	73.9	70.1	67.7	=	3	3	5	30
heart	2	270	38	80.8	81.9	82.2	84.3	+	2	2	5	83
horse	2	368	75	82.6	82.1	82.6	83.7	+	1	0	5	23
hypo	2	3163	47	99.2	98.9	98.4	95.1	+	2	2	5	66201
iono	2	351	100	90.0	92.3	91.5	93.2	=	1	2	2	1885
iris	3	150	15	95.3	94.7	94.0	95.8	+	2	1	5	7
lymph	4	148	63	73.5	77.8	83.1	86.2	=	3	0	5	400
pima	2	768	26	75.5	72.9	75.1	74.3	+	2	2	1	12
sonar	2	208	234	70.2	77.5	79.4	81.4	=	2	2	5	197091
tic-tac	2	958	29	99.4	99.6	99.2	100.0	+	1	0	1	32
vehicle	4	846	58	72.6	68.7	68.8	68.5	-	2	0	2	921
wine	3	178	45	92.7	95.0	95.0	95.3	+	3	2	1	802
zoo	7	101	43	92.2	96.8	97.1	97.8	=	3	0	5	237
average				84.14	85.52	85.82	86.3					

This table shows that our adaptation $l\delta$-`miner` of CMAR is operational and reaches similar performances. Sometimes, results can be even better than CMAR, because synergy between prescribing and excluding rules is powerful. Moreover, negations in premise (when $l \geq 2$) also improve some scores.

Let us recall that our purpose does not concern the proposition of the best classification method, but aims at studying the interest of enhancing an AR-based classification methods with rules containing negations. As our method is effective, we can use it to measure the impact of the negations: the next section details this impact regarding the different type of rules.

5.4 Computational Aspects

Figure 1 details two computational aspects of mining generalized rules. The results come from the dataset *anneal* but the experiments on other benchmarks lead to the same shapes.

- the extraction time (top part) is exponential with the minimum support, and increasing the depth l of the rules induces an exponential increase.
- the number of rules has the same behavior: it is exponential according to the depth.

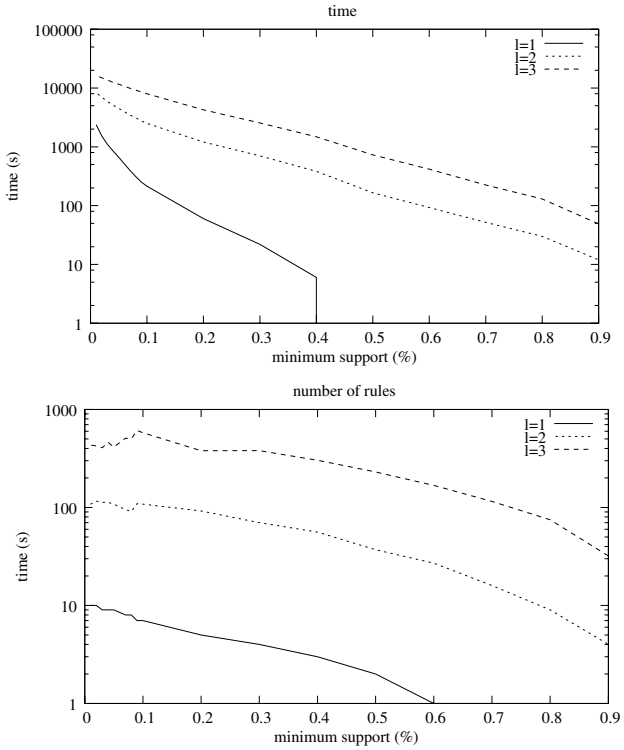

Fig. 1 Computational aspects of the depth of the rules. The top charts shows the execution time, the bottom one shows the number of rules, for a depth of $l = 1..3$.

We did not represent the impact of the number of exceptions on the running time and the number of rules, because the number of exceptions polynomially makes the running time longer and the number of rules bigger.

These experiments confirm the intuition that the higher the depth of the rule, the huger the number of rules and the computation time. This fact can improve the traditional classification model, as shown in the next section.

6 Discussion about the Semantics of the Generalized Rules

The contribution of the negations for the supervised classification is discussed through three axis:

1. What is the interest of class excluding rules? These rules stem from the generalized rules containing a class attribute in premise.
2. Is the use of rules with negations in premise advantageous? Varying the depth of the generalized rules allows to study the contribution of the negations in premise.
3. Are the δ-approximative rules useful and better than exact rules ? We compare the results obtained with various values of δ.

6.1 Contribution of Rules Excluding Classes

Figure 2 compares the scores for the rules prescribing the classes (noted $l\delta$-miner +), the rules excluding the classes ($l\delta$-miner $-$) or the rules combining both types ($l\delta$-miner =). This figure plots the gap between the best scores for each method and the reference score of $l\delta$-miner (Table 2). This kind of presentation allows to better show the contribution of each type of rule.

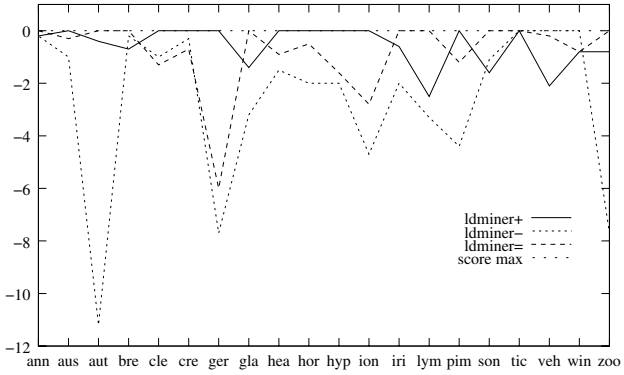

Fig. 2 Contribution of class excluding rules

We first notice that the three types of rules lead to similar results, but classical rules prescribing classes are the best in average. This confirms the intuition that the useful knowledge simultaneously lies in rules prescribing or excluding classes. The

gap between the results can be explained by the lack of optimization dedicated to the method using rules excluding classes, and refers to our perspectives. In particular, minimum support should be adapted.

The rules excluding classes give lower scores, but sometimes slightly improve the classical rules: breast (0.5%), sonar (0.5%), vehicle (2.1%) and wine (0.8%). In these cases, combining both types of rules may lead to better scores.

We can conclude that focusing on classical rules prescribing classes is enough, because their expressive power enhances the available knowledge that they are able to translate. This knowledge is similar as the one obtained by involving excluding classes. On a computation point of view, let us remark that rules excluding classes are not harder to mine than rules prescribing classes. If they improve the process, one should not worth ignore them. The classical rule mining algorithm has indeed not to be updated, the method should just focus on the rules containing a class attribute in premise.

6.2 Impact of Negation in Premise

Figure 3 shows the impact of the conclusion length for the generalized rules, *i.e.* the number of negations in the premise. It plots the gap between the best scores for $l = 1,2,3$, $\delta = 0$ and the reference score of Table 2. $l = 1$ corresponds to the classical *exact* association rules, and when $l = 2$ (resp. 3) there are one (resp. two) negation in the premise.

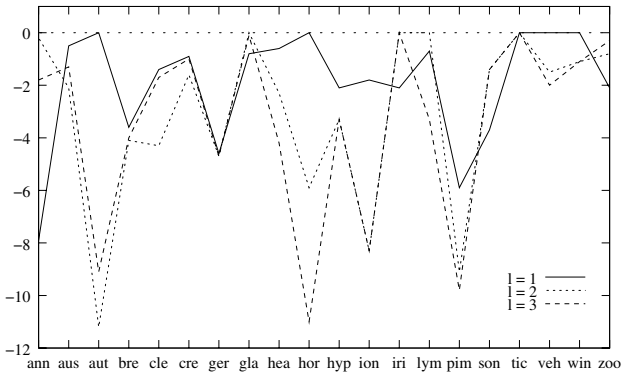

Fig. 3 Impact of the conclusion length for the generalized rules

We can see that the classical exact rules ($l = 1$) often give the best score, compared to the rules with $l = 2$ or $l = 3$. This score does not always reach the reference score, because the latter can be obtained by allowing some exceptions in the rule ($\delta = 1,2,3$, see Table 2 for details). It shows that the exact rules are not sufficient for the best scores.

On some datasets, one or two negations in the premise allow to improve the results of classical rules and the contribution is more significative than with a negation in conclusion: anneal (7.7%), glass (0.8%), iris (2.1%), sonar (2.3%) and zoo (1.8%).

In some cases, the expressive power of the rules with negations in premise is better than without negation. It then allows to find rules that do not fit to the classical rules. Moreover, there are much more generalized rules (or rules with negations in premise) than classical rules. This can be useful when the dataset is small (few objects and attributes).

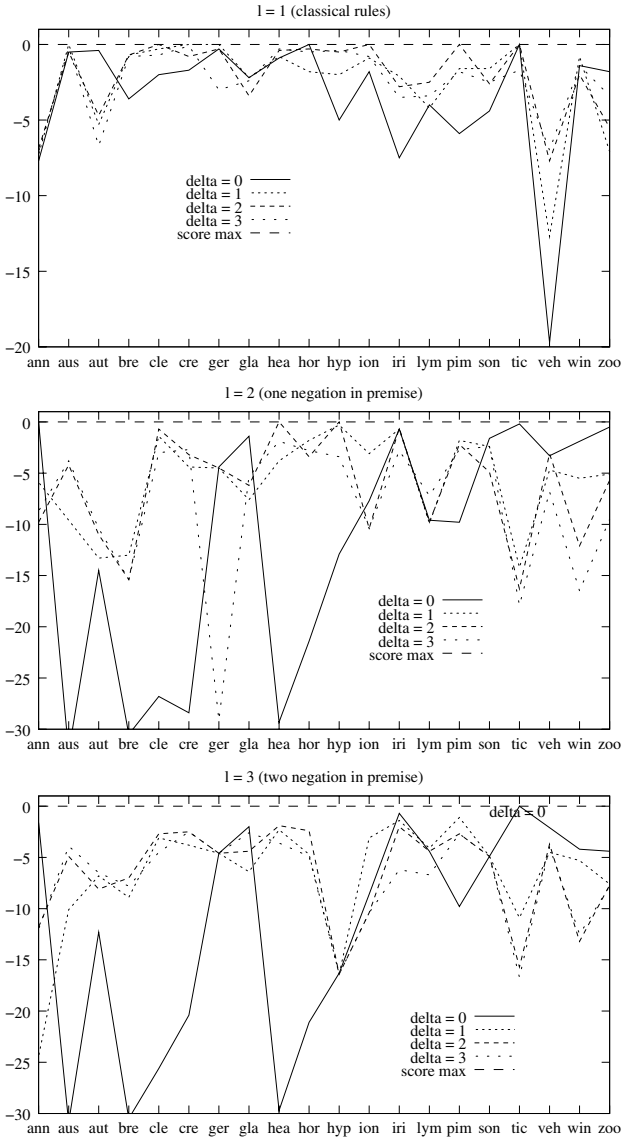

Fig. 4 Impact of δ-approximative generalized rules

In the other cases, the presence of negations lowers the scores, probably because of over-fitting: these rules are too strict and they are not enough generalized. It seems difficult to improve the performances with larger conclusions, because such rules become always more specific. Moreover, the computation time grows quickly with the conclusion length. We conclude that exact rules with negations in premise are hard to enhance but sometimes significantly improve the method.

6.3 Impact of the Exceptions for Generalized Rules

We end this discussion with studying the impact of approximative rules compared to exact rules. Figure 4 shows on three charts for $l = 1$, $l = 2$ and $l = 3$ the maximum scores of prescribing rules with δ varying from 0 to 3. The plain line gives the best score for the exact rules.

Except for some datasets (glass, tic-tac-toe, wine and zoo), allowing exceptions in the rules gives better scores than exact rules. This corresponds to the intuition that exact rules over-fit the data and a good classifier needs approximative rules. In classical approaches, approximative rules are obtained with a confidence threshold, below 1. For generalized association rules, confidence has no really sense, but our experiments show that using δ-approximative rules can replace the confidence framework.

7 Conclusion and Perspectives

In this paper, we have studied the impact of generalized association rules in classification processes, *i.e.* rules containing negations in premise and prescribing or excluding classes. For that purpose, we have proposed an algorithm to mine the whole set of frequent exact generalized association rules with minimal premise and conclusion, possibly with exceptions. This algorithm takes benefit from recent notions coming from condensed representations of patterns and hypergraph transversals area. We have provided a method to evaluate the impact of such rules and proposed a classifier $l\delta$-`miner` using rules prescribing and/or excluding classes.

The contribution of generalized association rules in classification is shy: they can improve the scores on a few benchmarks, but classical rules are often sufficient. Using generalized rules for classification is then sensitive. The class excluding rules are easy to mine and do not need any modification for the rule mining algorithm; they can give a slight improvement. The generalized rules with more than one attribute in the conclusion are hard to mine, but can reveal to be very useful when the dataset is small.

Our perspectives focus on two points: first, the optimization of the classification process for rules excluding classes, so that it can fit to the scores with rules prescribing classes. Second, we would like to further investigate rules with negations in premise on large datasets. For that purpose, the exception threshold has to be relative to the number of objects (for example $1‰$) or a fraction of the frequency threshold leading to a value close to the notion of confidence. In this chapter, we used fixed values (1..3) that are not well-suited for thousands of objects.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining pp. 307–328 (1996)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Intl. Conference on Very Large Data Bases (VLDB 1994), Santiago de Chile, Chile, pp. 487–499 (1994)
3. Antonie, M.L., Zaïane, O.: An associative classifier based on positive and negative rules. In: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2004), Paris, France (2004)
4. Baralis, E., Garza, P.: A lazy approach to pruning classification rules. In: IEEE International Conference on Data Mining, ICDM 02 Maebashi City, Japan (2002)
5. Baralis, E., Garza, P.: Majority classification by means of association rules. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 35–46. Springer, Heidelberg (2003)
6. Baralis, E., Garza, P.: Associative text categorization exploiting negated words. In: SAC 2006: Proceedings of the 2006 ACM symposium on Applied computing, pp. 530–535. ACM, New York (2006), http://doi.acm.org/10.1145/1141277.1141402
7. Bayardo, R.J.: The hows, whys, and whens of constraints in itemset and rule discovery. In: Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.) Constraint-Based Mining and Inductive Databases. LNCS (LNAI), vol. 3848, pp. 1–13. Springer, Heidelberg (2006)
8. Blake, C., Merz, C.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
9. Boulicaut, J.F., Bykowski, A., Jeudy, B.: Towards the tractable discovery of association rules with negations. In: Fourth Int. Conference on Flexible Query Answering Systems FQAS 2000, pp. 425–434 (2000)
10. Bouzouita, I., Elloumi, S.: Integrated generic association rule based classifier. In: DEXA 2007: Proceedings of the 18th International Conference on Database and Expert Systems Applications (DEXA 2007), pp. 514–518. IEEE Computer Society, Washington (2007), http://dx.doi.org/10.1109/DEXA.2007.90
11. Calders, T., Goethals, B.: Minimal k-free representations of frequent sets. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 71–82. Springer, Heidelberg (2003)
12. Chan, K.C.C., ho Au, W.: An effective algorithm for mining interesting quantitative association rules. In: Proc. of the 12th ACM Symp. on Applied Computing (SAC 1997), pp. 88–90. ACM Press, New York (1997)
13. Cornells, C., Yan, P., Zhang, X., Chen, G.: Mining positive and negative association rules from large databases. In: IEEE Conference on Cybernetics and Intelligent Systems, pp. 1–6 (2006)
14. Dong, X., Niu, Z., Shi, X., Zhang, X., Zhu, D.: Mining both positive and negative association rules from frequent and infrequent itemsets. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 122–133. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-73871-8_13
15. Eiter, T., Gottlob, G.: Identifying the minimal transversals of a hypergraph and related problems. SIAM Journal on Computing 24(6), 1278–1304 (1995)
16. Fredman, M., Kachiyan, L.: On the complexity of dualization of monotone disjunctive normal forms. Journal of Algorithms 21(2), 618–628 (1996)

17. Gu, L., Li, J., He, H., Williams, G.J., Hawkins, S., Kelman, C.: Association rule discovery with unbalanced class distributions. In: Australian Conference on Artificial Intelligence, pp. 221–232 (2003)
18. Gunopulos, D., Mannila, H., Khardon, R., Toivonen, H.: Data mining, hypergraph transversals, and machine learning. In: ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1997), Tucson, USA (1997)
19. Hagen, M.: Algorithmic and computational complexity issues of monet. Ph.D. thesis, Firedrich-Schiller-University Jena, Germany (2008)
20. Hébert, C., Bretto, A., Crémilleux, B.: A data mining formalization to improve hypergraph transversal computation. Fundamenta Informaticae 80(4), 415–433 (2007)
21. Hébert, C., Crémilleux, B.: A unified view of objective interestingness measures. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 533–547. Springer, Heidelberg (2007)
22. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: IEEE International Conference on Data Mining (ICDM 2001), San Jose, USA (2001)
23. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rules mining. In: International Conference on Knowledge Discovery and Data Mining (KDD 1998), New York, USA, pp. 80–86 (1998)
24. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997), citeseer.nj.nec.com/mannila97levelwise.html
25. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. Journal Intelligent Information Systems (JIIS) 24(1), 29–60 (2005), http://www.kde.cs.uni-kassel.de/stumme/papers/2005/pasquier2005generating.pdf
26. Rioult, F., Crémilleux, B.: Mining correct properties in incomplete databases. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, Springer, Heidelberg (2007)
27. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26, 2002, pp. 32–41 (2002)
28. Thiruvady, D.R., Webb, G.: Mining negative rules using GRD. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 161–165. Springer, Heidelberg (2004)
29. Wang, H., Zhang, X., Chen, G.: Mining a complete set of both positive and negative association rules from large databases. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 777–784. Springer, Heidelberg (2008)
30. Wang, J., Karypis, G.: On mining instance-centric classification rules. IEEE Trans. Knowl. Data Eng. 18(11), 1497–1511 (2006)
31. Wang, Y., Xin, Q., Coenen, F.: Hybrid rule ordering in classification association rule mining. To appear in Transactions on Machine Learning and Data Mining in Pattern Recognition (2008)
32. Wu, X., Zhang, C., Zhang, S.: Efficient mining of both positive and negative association rules. ACM Trans. Inf. Syst. 22(3), 381–405 (2004), http://doi.acm.org/10.1145/1010614.1010616

33. Yin, X., Han, J.: Cpar: Classification based on predictive association rules. In: Proceedings of the 2003 SIAM Int. Conf. on Data Mining (SDM 2003). San Fransisco, CA (2003)
34. Yuan, X., Buckles, B.P., Yuan, Z., Zhang, J.: Mining negative association rules. In: ISCC 2002: Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02), p. 623. IEEE Computer Society Press, Washington (2002)
35. Zaïane, O.R., Antonie, M.-L.: On pruning and tuning rules for associative classifiers. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3683, pp. 966–973. Springer, Heidelberg (2005)
36. Zaki, M.: Generating non-redundant association rules. In: ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, USA, pp. 34–43 (2000)
37. Zanuttini, B., Hébrard, J.J.: A unified framework for structure identification. Information Processing Letters 81(6), 335–339 (2002)
38. Zhao, L., Zaki, M.J., Ramakrishnan, N., Blosom, N.: A framework for mining arbitrary boolean expression. In: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (KDD 2006), pp. 827–832 (2006)