Associative Classification with Statistically Significant Positive and Negative Rules

Jundong Li
Department of Computing Science
University of Alberta
Edmonton, Canada
jundong1@ualberta.ca

Osmar R. Zaïane
Department of Computing Science
University of Alberta
Edmonton, Canada
zaiane@ualberta.ca

ABSTRACT

Rule-based classifier has shown its popularity in building many decision support systems such as medical diagnosis and financial fraud detection. One major advantage is that the models are human understandable and can be edited. Associative classifiers, as an extension of rule-based classifiers, use association rules to associate attributes with class labels. A delicate issue of associative classifiers is the need for subtle thresholds: minimum support and minimum confidence. Without prior knowledge, it could be difficult to choose the proper thresholds, and the discovered rules within the support-confidence framework are not statistically significant, i.e., inclusion of noisy rules and exclusion of valuable rules. Besides, most associative classifiers proposed so far, are built with only positive association rules. Negative rules, however, are also able to provide valuable information to discriminate between classes. To solve the above mentioned problems, we propose a novel associative classifier which is built upon both positive and negative classification association rules that show statistically significant dependencies. Experimental results on real-world datasets show that our method achieves competitive or even better performance than well-known rule-based and associative classifiers in terms of both classification accuracy and computational efficiency.

Categories and Subject Descriptors

D.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

Associative Classification; Negative Rules; Statistical Significance

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806524.

1. INTRODUCTION

The task of mining association rules over market basket analysis was introduced in [3] which aims to find associations between items or itemsets in a transaction database. The data is typically retail sales in the form of customer transactions, but can be any data if it is transformed to transactions like image data or text data. The problem is formally defined as follows: assume a transaction dataset $\mathcal{D}$ consists of a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$, each transaction T is a set of items such that $T \subseteq \mathcal{I}$ and is associated with a unique identifier. A transaction T is said to contain X, i.e., a set of items in $\mathcal{I}$, if $X \subseteq T$. Then an association rule is an implication of the form "$X \rightarrow Y$", where $X \subseteq \mathcal{I}$, $Y \subseteq \mathcal{I}$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has a *support* s in the transaction set $\mathcal{D}$ if the percentage of transactions in $\mathcal{D}$ that contain X and Y is s. It is also said that the rule $X \rightarrow Y$ holds in the transaction dataset $\mathcal{D}$ with *confidence* c if the percentage of transactions in $\mathcal{D}$ that contain X also contain Y is c. The problem of discovering all association rules from a set of transactions $\mathcal{D}$ consists of generating association rules that have a *support* and *confidence* greater than given thresholds. These discovered rules are called *strong rules*.

Classification is another common task in data mining and machine learning [15, 19, 26, 27]. A classifier is a system that is able to assign an unlabeled object to one or more predefined classes. Usually, it is created by building a learning model on the training data whose class labels are known in advance, then its ability to discriminate between classes is evaluated on the test data. Associative classification [25] is a classification method that integrates association rule mining and classification together. To build an associative classifier, classification association rules (CARs) with consequent as class labels are first mined by association rule mining techniques. Afterwards, some noisy CARs are pruned to improve the prediction performance, the remaining CARs form the actual associative classifier and these CARs are able to predict classes for unlabeled test data.

Existing associative classification methods mine the training dataset mostly in an Apriori-like way [4] or through a FP-growth approach [22], both of which are based on the support and confidence paradigm. However, appropriate support and confidence thresholds are not easy to determine. Besides, according to Webb [36, 37], traditional association rule mining methods have the risk of false discoveries such that the antecedent part and consequent part of some rules are not strongly correlated. In the worst cases, we may find all spurious rules while miss all rules with strong correlation. On the other hand, most associative classifiers proposed so

far use only positive CARs (rules of the form $X \rightarrow C$) in the classification process. In addition to positive CARs, negative CARs are also able to provide valuable information to discriminate between classes. A negative CAR is in one of the following form: $X \rightarrow \neg C$ or $\neg X \rightarrow C$ (where X and $\neg X$ indicate the presence and absence of itemset X, respectively). Let us consider the following example:

Example 1. For a binary classification problem, assume we discover some positive and negative CARs, such as: $X \rightarrow c_1$, $Y \rightarrow c_2$, $XY \rightarrow \neg c_1$. Now we have a new unlabeled instance XY, how to classify it? When only positive CARs $X \rightarrow c_1$ and $Y \rightarrow c_2$ are considered, both c_1 and c_2 are possible as there is a matching rule for either c_1 or c_2. But the classification task is easier since we have another negative CAR $XY \rightarrow \neg c_1$ which reinforces the decision in favor of class c_2.

In this paper, we propose a novel associative classifier to tackle the above mentioned issues. Following traditional associative classification methods, the proposed associative classifier consists of three steps: rule generation, rule pruning and rule classification.

The main contributions of this paper are as follows:

1. A novel associative classifier is proposed, it achieves a competitive or even better performance compared with some well-known rule-based classifiers in terms of both classification accuracy and computational efficiency.

2. By extending the Kingfisher algorithm in [20], we are able to find the complete set of both positive and negative CARs that show statistically significant dependencies efficiently.

3. We develop a novel rule pruning strategy to prune noisy positive and negative CARs simultaneously without jeopardizing the classification accuracy.

4. We present and compare different rule classification methods to investigate how to make a prediction with multiple matching positive and negative CARs.

The reminder of the paper is organized as follows. The overview of related work on associative classifier and negative rule mining is given in Section 2. Section 3 describes the proposed associative classifier in three steps. Section 4 contains the experimental results and the statistical analysis. Section 5 concludes the paper and discusses some possible future work.

2. RELATED WORK

In this section, we review some related work on associative classification and negative rule mining.

2.1 Associative Classification

The first reference to use association rules as CARs is credited to [12], while the first associative classifier, CBA, was introduced by Liu et al. [25]. The main steps in building an associative classifier are as follows:

- Modeling the data into the transaction dataset $\mathcal{D}$ in which the numerical attributes are transformed to discrete attributes.

- Generating the set of CARs from the transaction dataset $\mathcal{D}$. The CARs are in the form of $X \rightarrow C$ where X is a conjunction of attributes and C is a class label. The CARs are usually generated by pushing the constraint in the association rule mining process to generate association rules that always have as consequent a class label given minimum support and minimum confidence thresholds.

- Pruning the discovered CARs by some rule pruning strategies. The previous rule generation phase usually generates an overwhelming number of CARs including many noisy CARs and it is very important to prune these rules to make the classifier more effective and efficient. The phase is employed to choose a best subset of CARs and weed out those rules that may introduce errors or are overfitting in the classification stage.

- Classifying a new unlabeled object to a predefined class. At this level a system that can make a prediction for a new object is built. The challenge here is how to rank and make use of the set of rules from the previous phase to give a good prediction.

CBA [25] mines the complete set of CARs through an apriori-like approach, in addition, it ignores rules by "pessimistic error rate" as C4.5 [26]. In the rule pruning phase, CBA adopts a strategy called "database coverage". Database coverage consists of going over all the rules ranked by their confidence values and evaluating them against the training instances. Whenever a rule applies correctly on some instances, the rule is marked and the instances are eliminated until all training instances are covered. Finally, unmarked rules are simply pruned. New instances are classified by a matching rule with the highest ranking.

Motivated by the idea of CBA, many improvements have been proposed to build more accurate associative classifiers. CMAR [24] maintains a CR-tree to compactly store and retrieve rules, the CARs are discovered by a FP-growth approach. In addition to the database coverage method, CMAR also prunes lower ranked and more specific rules. The rule R_1: $P \rightarrow C$ with confidence $conf_1$ is a lower ranked and more specific rule w.r.t rule R_2: $P' \rightarrow C$ with confidence $conf_2$ if $P' \subsetneq P$ and $conf_1 \leq conf_2$. For a new unlabeled instance, CMAR makes a prediction based on multiple matching rules with a weighted chi-square measure.

In the classification phase, ARC [6] takes all rules that apply within a confidence range, but instead, calculates the average confidence for each set of rules grouped by class labels in the consequent, and selects the class label of the group with the highest confidence average.

There are some other variants of associative classifiers: Harmony [35] is an example which directly mines CARs. It adopts an instance-centric approach to find the highest confidence rule for each training instance and builds the classification model from the union of these rules. It shows to be more effective and scalable than other associative classifiers.

2SARC [9] is a two-stage classification model that is able to automatically learn to select rules for classification. First, an associative classifier is learned by standard techniques. Second, multiple predefined features are computed on the associative classifier, then they act as input to a neural network to achieve a more accurate classification model.

CCCS [10] uses a new measure, "Complement Class Support" (CCS) to mine positively correlated CARs for the im-

balanced classification problem. It forces the CCS measure to be monotonic, thus the complete set of CARs are discovered by a row enumeration algorithm. An associative classifier is then built upon these positively correlated CARs.

SPAR-CCC [33] is another associative classifier designed for imbalanced data. It also integrates a new measure, "Class Correlation Ratio" (CCR) into the rule mining phase, the classifier works comparably on balanced datasets and outperforms other associative classifiers on imbalanced datasets.

ARC-PAN [7] is the first associative classifier that uses both positive and negative CARs. It proposes to add Pearson's correlation coefficient on the basis of support and confidence framework to mine positively and negatively correlated CARs. The ability of negative CARs is demonstrated by their usage in the classification phase.

2.2 Negative Rule Mining

A negative association between two positive itemsets X, Y are rules of the following forms: $\neg X \rightarrow Y$ and $X \rightarrow \neg Y$, where $\neg X$ and $\neg Y$ indicate the absence of itemsets X and Y in the transaction dataset $\mathcal{D}$, respectively. Mining association rules from a transaction dataset that contains information about both present and absent itemsets is computationally expensive, traditional association rule mining algorithms cannot be directly applied. This is the reason why new algorithms are needed to efficiently mine association rules with negative itemsets. Here we survey algorithms that efficiently mine some variety of negative association rules from data.

Brin et al. [13] mention for the first time the notion of negative relationships in the literature. They propose to use chi-square test between two itemsets. The statistical test verifies the independence between two itemsets. To determine the nature (positive or negative) of the relationship, a correlation metric is used.

Aggarwal and Yu [1, 2] introduce a new method for finding interesting itemsets in data. Their method is based on mining strongly collective itemsets. The collective strength of an itemset I is defined as $C(I) = \frac{1-v(I)}{1-E[v(I)]} \times \frac{E[v(I)]}{v(I)}$, where $v(I)$ is the violation rate of an itemset I, i.e., the fraction of violations over the entire set of transactions and $E[v(i)]$ is its expected value. An itemset I is in a violation of a transaction if only a subset of its itemsets appears in that transaction. The collective strength ranges from 0 to ∞, where a value of 0 means that the items are perfectly negatively correlated and a value of ∞ means that the items are perfectly positively correlated.

In [29], the authors present a new idea to mine strong negative rules. They combine positive frequent itemsets with domain knowledge in the form of taxonomy to mine negative associations. The idea is to reduce the search space by constraining the search to positive patterns that pass the minimum support threshold. When all the positive itemsets are discovered, candidate negative itemsets are considered based on the used taxonomy.

Wu et al. [38] derive another algorithm for generating both positive and negative association rules. The negative associations discovered are of the following forms: $\neg X \rightarrow Y$, $X \rightarrow \neg Y$ and $\neg X \rightarrow \neg Y$. They add another measure called "mininterest" on top of the support-confidence framework for a better pruning of the frequent itemsets, which is also used to assess the dependency between two itemsets.

The SRM algorithm [30, 31], discovers a subset of negative associations. The authors develop an algorithm to discover negative associations of the type $X \rightarrow \neg Y$. These association rules can be used to discover which items are substitutes for others in market basket analysis.

Antonie and Zaïane [8] propose an algorithm to mine strong positive and negative association rules based on the Pearson's correlation coefficient. In their algorithm, itemset and rule generation are combined and the relevant rules are generated on-the-fly while analyzing the correlations within each candidate itemset.

In [32], the authors extend an existing algorithm for association rule mining, i.e., GRD (generalized rule discovery), to include negative items in the discovered rules. The algorithm discovers top-K positive and negative rules.

Cornelis et al. [16] propose a new Apriori-based algorithm (PNAR) that exploits the upward closure property of negative association rules. With this upward closure property, valid positive and negative association rules can be discovered efficiently. Wang et al. [34] give a more intuitive way to express the validity of both positive and negative association rules, the mining process is very similar to PNAR.

MINR [23] is a method that uses Fisher's exact test to identify itemsets that do not occur together by chance, i.e., with a statistically significant probability. An itemset with a support greater than the positive chance threshold is considered for positive rule generation, while an itemset with a support less than the negative chance threshold is considered for negative rule generation.

Kingfisher [20, 21] is developed to discover both positive and negative dependency rules. The dependency rule can be formulated on the basis of association rule and the statistical dependency of a rule can be calculated by Fisher's exact test. In order to reduce the search space, the author introduces a branch-and-bound search method with three lower bounds for the measure of p_F-value. Another two pruning strategies (pruning by minimality and pruning by principles of Lapis philosophorum) are also included to speed up the search.

A more detailed review of negative rule mining can be referred to [5].

3. PROPOSED ASSOCIATIVE CLASSIFIER

In this section, we introduce details about the proposed associative classifier in three steps: rule generation, rule pruning and rule classification. Before talking about the detailed steps, we introduce some notations and definitions used in this paper.

3.1 Notations and Definitions

Assume $\mathcal{D}$ is a transformed transaction dataset with a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ and a set of class labels $\mathcal{C} = \{c_1, c_2, ..., c_n\}$. Each transaction T is associated with a set of items X and a class label c_k, where $X \subseteq \mathcal{I}$ and $c_k \in C$. A CAR is in the form of $X \rightarrow c_k$ or $X \rightarrow \neg c_k$, and it is considered dependent if $P(X, c_k) \neq P(X)P(c_k)$ or $P(X, \neg c_k) \neq P(X)P(\neg c_k)$. Since we intend to find statistically significant positive and negative CARs, we take Fisher's exact test [20, 23, 33] as a significance measure. The dependency of the CAR $X \rightarrow c_k$ or $X \rightarrow \neg c_k$ is considered statistically significant at level α, if the probability of observing equal or stronger dependency in the dataset under a null hypothesis model, is not greater than α. X and c_k (or $\neg c_k$) are independent in the null hypothesis. The

probability p, i.e., p-value is:

$$p_F(X \to c_k) = \sum_{i=0}^{min\{\sigma(X,\neg c_k)\sigma(\neg X,c_k)\}} \frac{\binom{\sigma(X)}{\sigma(X,c_k)+i}\binom{\sigma(\neg X)}{\sigma(\neg X,\neg c_k)+i}}{\binom{|\mathcal{D}|}{\sigma(c_k)}}$$

$$p_F(X \to \neg c_k) = \sum_{i=0}^{min\{\sigma(X,\neg c_k)\sigma(\neg X,c_k)\}} \frac{\binom{\sigma(X)}{\sigma(X,\neg c_k)+i}\binom{\sigma(\neg X)}{\sigma(\neg X,c_k)+i}}{\binom{|\mathcal{D}|}{\sigma(c_k)}},$$

where the consequent class label c_k can either be present or absent. $\sigma(*)$ denotes the support count of $*$, $*$ can be the conjunction of any itemsets, either being present or absent, for example, it can be X, Xc_k, $\neg Xc_k$, $\neg X\neg c_k$, etc. The significance level α is usually set to be 0.05.

The dependency of the positive CAR $X \to c_k$ or negative CAR $X \to \neg c_k$ is statistically significant, if $p_F(X \to c_k) \leq \alpha$ or $p_F(X \to \neg c_k) \leq \alpha$. In the field of rule mining, an important task is to mine non-redundant rules. Rules are considered redundant when they do not add new information to the remaining rules. Without the non-redundancy property taken into consideration, the number of discovered rules is usually too large for people to read and interpret. In order to reduce the number of rules and to make the classification model more readable, only non-redundant CARs are considered. Following [20], we define non-redundant CARs and minimal CARs as follows:

Definition 1. Non-redundant CARs
The CAR $X \to c_k$ or $X \to \neg c_k$ is considered as non-redundant, if there does not exist any CARs in the form of $Y \to c_k$ or $Y \to \neg c_k$ such that:

$$p_F(Y \to c_k) < p_F(X \to c_k)$$

or

$$p_F(Y \to \neg c_k) < p_F(X \to \neg c_k),$$

where $Y \subsetneq X$.

Definition 2. Minimal CARs
The CAR $X \to c_k$ or $X \to \neg c_k$ is considered as minimal, if and only if $X \to c_k$ or $X \to \neg c_k$ is non-redundant, and, there does not exist any CARs in the form of $Z \to c_k$ or $Z \to \neg c_k$ such that:

$$p_F(Z \to c_k) < p_F(X \to c_k)$$

or

$$p_F(Z \to \neg c_k) < p_F(X \to \neg c_k),$$

where $X \subsetneq Z$.

3.2 Proposed Associative Classifier

3.2.1 Positive and Negative CARs generation

In traditional association rule mining algorithms, the measure of *support* is usually used to prune non-frequent patterns or rules since it has a downward closure property, but it is not the case for the p-value, making it impossible to be used as a monotonic property for some effective pruning. Recently, Kingfisher [20] was proposed to find the complete set of positive and negative rules that show statistically significant dependencies. However, it was designed for the discovery of general rules, not specifically for CARs. Therefore, adaption of the Kingfisher algorithm to enable the discovery of only CARs is necessary as it can reduce the number of discovered rules. To find statistically significant positive and negative CARs, we extend the Kingfisher algorithm by pushing the rule constraint in the rule generation phase.

First, two theorems in [20] are given:

Theorem 1. In a transaction database $\mathcal{D}$, assume $\mathcal{R}$ is the set of all items, for any item $A \in \mathcal{R}$ and $X \subseteq \mathcal{R}\backslash A$, it has $p_F(X \to A) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$ and $p_F(X \to \neg A) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$; if $\sigma(A) \leq \frac{|\mathcal{D}|}{2}$, then for any $B \in \mathcal{R}$, $X \subseteq \mathcal{R}\backslash\{A, B\}$, it has $p_F(XA \to B) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$ and $p_F(XA \to \neg B) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$. Therefore, there exists a threshold $\gamma \leq 0.5$, when $\sigma(A) < \gamma|\mathcal{D}|$, the item A cannot appear in any statistically significant rules.

Theorem 2. In a transaction database $\mathcal{D}$, assume $\mathcal{R}$ is the set of all items, for any item $A \in \mathcal{R}$, $X \subseteq \mathcal{R}\backslash A$ and $Q \subseteq \mathcal{R}\backslash\{X, A\}$, if $\sigma(X) \leq \sigma(A)$ or $\sigma(X) \leq \sigma(\neg A)$ holds, then it has $p_F(XQ \to A) \geq \frac{\sigma(\neg X)!\sigma(A)!}{|\mathcal{D}|!(\sigma(A)-\sigma(X))!}$ or $p_F(XQ \to \neg A) \geq \frac{\sigma(\neg X)!\sigma(\neg A)!}{|\mathcal{D}|!(\sigma(\neg A)-\sigma(X))!}$, respectively.

Given these two theorems, we derive three corollaries to generate positive and negative statistically significant CARs.

COROLLARY 1. *There exists a threshold $\gamma \leq 0.5$ such that the item $I \in \mathcal{I}$ is impossible to be in any statistically significant CARs if its support is smaller than $\gamma|\mathcal{D}|$.*

PROOF. Corollary 1 is a special case of Theorem 1 when $I \in \mathcal{I}$. First we assume that I can be in the consequent part of the rule, then according to Theorem 1, we can find a threshold $\gamma \leq 0.5$ such that when $\sigma(I) < \gamma|\mathcal{D}|$, I cannot appear in any statistically significant rules. Since we only intend to find CARs where item I can only be in the antecedent part, if the condition $\sigma(I) < \gamma|\mathcal{D}|$ holds, item I can cannot appear in any statistically significant CARs. □

Some impossible items are pruned before further analysis by Corollary 1. It is assumed that s items ($s \leq m$) are left. The remaining s items are reordered and renamed in an ascending order by their support count, i.e., $\mathcal{I}_{rest} = \{i_1, i_2, ..., i_s\}$, where $\sigma(i_1) \leq \sigma(i_2) \leq ... \leq \sigma(i_s)$. Then in order to traverse the whole search space, an enumeration tree is built over $\mathcal{I}_{rest}$. For each node in the tree, the antecedent part is a combination of items in the power set of $\mathcal{I}_{rest}$. The consequent part is $2n$ possible class labels (either positive or negative). Therefore, for each node in the enumeration tree, we have to check all $2n$ possible CARs to see if they are statistically significant.

COROLLARY 2. *For any $X \subseteq \mathcal{I}_{rest}$, $Q \subseteq (\mathcal{I}_{rest}\backslash X)$, if $\sigma(X) \leq \sigma(c_k)$ or $\sigma(X) \leq \sigma(\neg c_k)$ holds, we can get $p_F(XQ \to c_k) \geq \frac{\sigma(\neg X)!\sigma(c_k)!}{|\mathcal{D}|!(\sigma(c_k)-\sigma(X))!}$ or $p_F(XQ \to \neg c_k) \geq \frac{\sigma(\neg X)!\sigma(\neg c_k)!}{|\mathcal{D}|!(\sigma(\neg c_k)-\sigma(X))!}$, respectively.*

PROOF. Corollary 2 can be considered as a special case of Theorem 2 when either c_k or $\neg c_k$ is the consequent part of a rule. □

According to Corollary 2, the lowest value of $p_F(XQ \to c_k)$ and $p_F(XQ \to \neg c_k)$ provide the lower bounds for $p_F(X \to c_k)$ and $p_F(X \to \neg c_k)$, respectively. Therefore, if the lower bound exceeds α, the corresponding CAR $X \to c_k$ or $X \to \neg c_k$ is not statistically significant and can be directly pruned. Otherwise, the CAR $X \to c_k$ or $X \to \neg c_k$ is considered as ***PSS***, i.e., "Potentially Statistically Significant".

Definition 3. The CAR $X \to c_k$ or $X \to \neg c_k$ is defined as ***PSS***, i.e., "Potentially Statistically Significant", if it meets either of the following conditions: (1) $\sigma(X) \leq \sigma(c_k)$ or $\sigma(X) \leq \sigma(\neg c_k)$ holds, and the lower bound $\frac{\sigma(\neg X)!\sigma(c_k)!}{|\mathcal{D}|!(\sigma(c_k)-\sigma(X))!}$ or $\frac{\sigma(\neg X)!\sigma(\neg c_k)!}{|\mathcal{D}|!(\sigma(\neg c_k)-\sigma(X))!}$ is smaller than or equal to α, respectively; (2) $\sigma(X) > \sigma(c_k)$ or $\sigma(X) > \sigma(\neg c_k)$ holds.

If a CAR is ***PSS***, we need to calculate the exact p-value to see if it is indeed statistically significant.

COROLLARY 3. *If CAR $X \to c_k$ or $X \to \neg c_k$ is PSS, then any of its parent rule $Y \to c_k$ or $Y \to \neg c_k$ is also PSS, where $Y \subsetneq X$ and $|Y| = |X| - 1$.*

PROOF. There are two situations making $X \to c_k$ being PSS. The first situation is when $\sigma(X) > \sigma(c_k)$, since $Y \subsetneq X$, thus $\sigma(Y) > \sigma(X) > \sigma(c_k)$, and it is easy to see the parent rule $Y \to c_k$ is also PSS. The second situation is when $X \to c_k$ but $lowerbound(p_F(XQ \to c_k)) < \alpha$, where $Q \subseteq (\mathcal{I}_{rest} \backslash X)$. Now let $XQ = Y(X \backslash Y)Q = YR$, because $(X \backslash Y) \subseteq (\mathcal{I}_{rest} \backslash Y)$ and $Q \subseteq (\mathcal{I}_{rest} \backslash X) \subseteq (\mathcal{I}_{rest} \backslash Y)$, thus $R = (X \backslash Y)Q \subseteq (\mathcal{I}_{rest} \backslash Y)$ and therefore, there must exists $R \subseteq (\mathcal{I}_{rest} \backslash Y)$ making $lowerbound(p_F(YQ \to c_k)) < \alpha$, i.e., rule $Y \to c_k$ is PSS. The proof is similar for the negative CAR $X \to \neg c_k$. □

With these three corollaries, the whole search problem can be summarized as follows: We first use Corollary 1 to prune impossible items, sort and rename the remaining items in an ascending order by their support. Next, all candidate CARs with only one antecedent item are listed. We then use Corollary 2 to check if they are *PSS*, non-*PSS* candidate CARs can be pruned directly without further analysis. *PSS* CARs are further checked to see if they are indeed statistically significant. From *PSS* 1-itemset CARs, we generate candidate *PSS* 2-itemset CARs by Corollary 3. The process repeats until no *PSS* CARs are generated at a certain level. It also needs to be mentioned that in the searching process, the minimality of the CARs is considered, if the CAR is marked as minimal, we stop the expansion from this CAR because all of its children CARs are impossible to get a lower p-value. In fact, checking minimality for a CAR is a hard task, because we have to consider its whole subtree. We use a well-proven result from [20] that if $P(c_k|X) = 1$ or $P(\neg c_k|X) = 1$, the corresponding CAR $X \to c_k$ or $X \to \neg c_k$ is minimal. In other words, the property of minimality can be detected by calculating the conditional probability of c_k or $\neg c_k$ given X. Therefore, for a certain CAR, we do not need to check all its children CARs in its subtree to see if it is minimal anymore. The rule generation process is presented in Algorithm 1.

Data: Transaction Dataset $\mathcal{D}$, set of antecedent items $\mathcal{I}$, class labels C, significance level $\alpha = 0.05$.
Result: Statistically significant positive and negative CAR sets $\mathcal{R}_{pos}$ and $\mathcal{R}_{neg}$.

```
Prune impossible antecedent items I with Corollary 1;
I_rest: the reoredered and renamed antecedent item set;
Create root node and level-1 nodes;
Set l = 1;
while l ≤ |I_rest| do
    for each candidate l-set CAR r do
        if all parent rules of r are PSS and not minimal then
            if p_F(r) ≤ α then
                if r is non-redundant then
                    if r is minimal then
                        r.minimal = true;
                    end
                    if r.class is positive then
                        R_pos.add(r);
                    else
                        R_neg.add(r);
                    end
                end
            end
        else
            prune CAR r and all its decedent rules from
            the enumeration tree;
        end
    end
    l = l + 1;
end
```

Algorithm 1: Statistically significant positive and negative CARs generation.

3.2.2 Rule Pruning

In the rule generation phase, redundant (lower ranked and more specific) CARs have been removed, but the number of discovered statistically significant CARs could still be very large. The disadvantages of a large number of CARs are two folds: first, noisy CARs may be included, they may jeopardize the classification performance; second, a classifier with a small number of rules is important since it allows domain experts to tune a classifier by editing rules if necessary.

The most widely used rule pruning strategy is database coverage [25], however, the database coverage heuristic can only be applied to positive CARs and negative CARs in the form of $\neg X \to c_k$. Through our rule generation phase, the discovered CARs are all in the form of $X \to \neg c_k$, to reduce the number of negative CARs in this type, we propose a novel rule pruning strategy to prune noisy positive and negative CARs simultaneously.

We first scan through the set of discovered negative CARs. For each negative CAR $X \to \neg c_k$, if it misclassifies at least one training instance, in other words, if we find an instance t in the training dataset such that $X \subseteq t.antecedent$ and $c_k = t.class$, the negative CAR $X \to \neg c_k$ is pruned, otherwise, it is kept for the following classification phase.

For the positive CARs $X \to c_k$, we first rank them by their confidence values, then use the database coverage method to select a subset of high quality.

Here a problem arises, in some datasets, the number of remaining negative CARs may be much larger than the number of remaining positive CARs. In the extreme case if only negative CARs $X \to \neg c_k$ are left, it is still hard to make a prediction for a new instance, for example, for instance XY, the only information obtained is that class label c_k is not correct. Therefore, we still wish the positive CARs dominate the classification decision phase, while taking negative CARs as a complement to help positive CARs. Considering this, we adjust the number of negative CARs, make it at most as large as the number of positive CARs. To be more specific, let n_{neg} and n_{pos} denote the number of pruned positive and negative CARs, respectively, if $n_{neg} > n_{pos}$, only the first n_{pos} negative CARs and all positive CARs are cho-

Data: Set of positive and negative CARs $\mathcal{R}_{pos}$, $\mathcal{R}_{neg}$ from rule generation phase.
Result: Pruned CARs set $\mathcal{R}_{newpos}$ and $\mathcal{R}_{newneg}$.
Ranking $\mathcal{R}_{pos}$ and $\mathcal{R}_{neg}$ according to confidence values;
$\mathcal{R}_{newpos} = \emptyset$, $\mathcal{R}_{newneg} = \emptyset$;
// 1. Negative CARs pruning
for *each CAR r in* $\mathcal{R}_{neg}$ **do**
  for *each training instance t in training dataset* $\mathcal{D}$ **do**
    if $r.antecedent \subseteq t.antecedent$ *and* $r.class = t.class$ **then**
      $\mathcal{R}_{neg}.remove(r)$;
      break;
    end
  end
end
Assign $\mathcal{R}_{neg}$ to $\mathcal{R}_{newneg}$;
// 2. Positive CARs pruning
for *each CAR r in* $\mathcal{R}_{positive}$ **do**
  for *each training instance t in training dataset* $\mathcal{D}$ **do**
    if $r.antecedent \subseteq t.antecedent$ *and* $r.class = t.class$ **then**
      $\mathcal{R}_{newpos}.add(r)$;
      remove instances covered by r in $\mathcal{D}$;
    end
  end
end
// 3. Negative CARs set adjustment
if $|\mathcal{R}_{newneg}| > |\mathcal{R}_{newpos}|$ **then**
  $\mathcal{R}_{newneg}$ = first $|\mathcal{R}_{newpos}|$ rules in $\mathcal{R}_{newneg}$;
end

Algorithm 2: Positive and negative CARs pruning.

sen as the actual classifier. The whole process is illustrated in Algorithm 2.

3.2.3 Classification

The set of statistically significant positive and negative CARs left from the previous rule pruning phase represents the actual associative classifier. Given a new unlabeled object, the classification process searches for the set of CARs that are relevant to this object, and makes the prediction according to the label information of all these relevant rules. Here we discuss how to make predictions for new objects based on the set of rules in the classifier. There are two types of CARs in our classifier: positive CARs in the form of $X \rightarrow c_k$ and negative CARs in the form of $X \rightarrow \neg c_k$. These two types of CARs are both considered in our classification phase. A simple way to classify a new object is to select the matching rule with the highest confidence value and assign its label to the new object. This is the strategy adopted in CBA [25]. But in this way, the negative rule $X \rightarrow \neg c_k$ does not make any sense if it has the highest confidence since it does not allow the labeling. In this way, only positive CARs influence the classification decision. Therefore, we propose to divide all matching rules into groups according to their class labels. The groups are ordered either by average confidence values or sum of confidence values. Then the group with the highest average or sum of confidence values will be assigned to the new object. These three possible classification methods are denoted as BEST, AVE and SUM, representing classifying by the best matching rule, by the average and by the sum of confidence values, respectively.

Data: A new instance o to be classified. Set of positive CARs $\mathcal{R}_{newpos}$ and negative CARs $\mathcal{R}_{newneg}$ from rule pruning phase.
Result: Class label of the new instance o.
$T_{pos} = \emptyset$; // set of positive rules matching o
$T_{neg} = \emptyset$; // set of negative rules matching o
for *each CAR r in* $\mathcal{R}_{newpos}$ **do**
  if $r.antecedent \subseteq o.antecedent$ **then**
    T_{pos}.add(r);
  end
end
for *each CAR r in* $\mathcal{R}_{newneg}$ **do**
  if $r.antecedent \subseteq o.antecedent$ **then**
    $r.confidence = -r.confidence$;
    T_{neg}.add(r);
  end
end
Divide T into n subsets by class labels: $T_1, T_2, ..., T_n$;
// 1. Classification method BEST
for *each subset* $T_1, T_2, ..., T_n$ **do**
  find the CAR with the highest confidence value
end
Assign its class label to o;
// 2. Classification method AVE
for *each subset* $T_1, T_2, ..., T_n$ **do**
  average confidence value of matching CARs in each class
end
Assign the class with the highest average of confidence value to o;
// 3. Classification method SUM
for *each subset* $T_1, T_2, ..., T_n$ **do**
  sum up the confidence values of matching CARs in each class
end
Assign the class with the highest sum of confidence value to o;

Algorithm 3: Three methods to classify a new instance.

The detailed descriptions of these classification methods are presented in Algorithm 3.

It is obvious that the confidence values of positive CARs $X \rightarrow c_k$ are added to the class c_i in the classification phase. However, the negative CARs $X \rightarrow \neg c_k$ is treated differently, we choose to subtract its confidence value from the total confidence of the corresponding class c_k.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed associative classifier on 20 datasets from the UCI Machine Learning Repository [11]. In these datasets, the numerical attributes have been discretized by the author of [14], the discretization strategy is different from [24, 25], thus the classification performance may be slightly different from the results reported before. All the experimental results are reported as an average over 10-fold cross validation. To have a fair comparison with some other methods, we also list the classification performance of two rule-based classifier: C4.5 [26] and FOIL [27]; two associative classifier on positive CARs: CBA [25] and CMAR [24]; an associative classifier on both positive and negative CARs: ARC-PAN [7]; a hybrid of rule-based and associative classifier: CPAR [39]; Naïve Bayes [28]

Table 1: Classification accuracy on 20 UCI datasets by different classifiers.

Dataset	C4.5	FOIL	CBA	CMAR	CPAR	PAN	NB	SVM	BEST	rules+		rules+-	
										SUM	AVE	SUM	AVE
adult	78.8	84.6	84.2	81.3	77.3	83.1	82.3	75.8	80.8	83.2	76.1	83.2	76.1
anneal	76.7	98.8	94.5	90.7	95.1	86.9	86.3	85.0	94.2	88.0	94.3	91.9	94.5
breast	91.5	89.3	94.1	89.9	93.0	89.4	95.9	95.7	90.0	91.3	90.0	91.3	90.0
cylBands	69.1	74.1	76.1	76.5	70.0	42.2	72.2	76.6	66.1	72.0	65.4	72.2	70.4
flare	82.1	83.8	84.2	84.3	63.9	83.4	71.0	73.8	83.9	83.2	84.0	83.2	84.1
glass	65.9	66.5	68.4	71.1	64.9	48.5	48.6	68.6	70.6	72.0	65.9	72.4	69.2
heart	61.5	55.2	57.8	56.2	53.8	58.8	53.4	55.4	51.5	56.8	51.8	57.4	56.4
hepatitis	84.1	77.8	42.2	79.6	75.5	39.8	84.5	79.3	85.8	83.2	83.2	83.9	82.6
horseColic	70.9	83.4	78.8	82.3	81.2	81.5	77.9	72.5	70.7	76.7	73.6	75.8	71.5
ionosphere	84.6	86.6	32.5	91.5	88.9	83.7	82.6	87.7	80.1	85.0	74.6	82.3	78.6
iris	91.3	94.0	93.3	94.0	94.7	94.6	94.6	94.6	94.7	94.7	95.3	94.7	95.3
led7	73.9	60.5	73.1	73.2	71.3	59.5	73.1	73.6	70.9	73.3	56.6	73.3	56.7
letRecog	50.4	50.0	32.5	28.3	58.2	26.9	64.1	65.2	54.2	61.8	40.6	64.9	54.5
mushroom	92.8	99.5	46.7	100.0	98.5	98.9	95.8	99.8	100.0	100.0	100.0	100.0	100.0
pageBlocks	92.0	92.4	90.9	90.1	92.5	89.9	90.8	91.2	90.7	91.1	90.6	91.2	90.8
penDigits	70.5	84.1	92.3	87.4	80.5	79.5	85.7	86.7	87.4	90.3	75.7	91.3	85.7
pima	71.7	71.9	74.6	74.4	74.0	74.2	75.3	74.0	75.1	67.7	72.9	68.0	72.8
soybean	60.3	88.0	89.2	88.1	83.1	81.8	92.9	88.7	90.3	89.6	90.6	90.5	90.9
wine	75.8	88.2	49.6	92.7	88.2	89.3	96.0	94.9	84.3	88.2	80.9	91.6	89.9
zoo	91.0	93.1	40.7	93.0	94.1	86.1	93.0	92.2	93.1	93.1	93.1	94.1	94.1
Average	76.7	81.1	69.8	81.3	79.9	73.9	80.8	81.6	80.7	82.1	77.8	**82.7**	80.2

and SVM [17]. The parameters of all these classifiers follow the default settings as the original papers.

4.1 Classification Performance

4.1.1 Accuracy

First, we show the classification accuracy by different methods on 20 UCI datasets. The experimental results are shown in Table 1. **Columns 2-9** list the classification accuracies of these compared methods. In **Column 10**, classification accuracy is determined by the best matching rule. **Columns 11-12** show the performance with only positive CARs on two classification methods, SUM and AVE, while **Columns 13-14** list the classification results when negative CARs are also considered.

As can be observed, rules+- with SUM gets the best overall classification performance (82.7%) and wins 4 out of 20 datasets, followed by rules+ with SUM (82.1%). Both of their average classification accuracy outperform the others.

4.1.2 Different rule classification strategies

Then we compare three different classification strategies, SUM, BEST as well as AVE. The classification method SUM is better than the other two classification methods BEST and AVE by winning around 2%-4% average classification accuracy, no matter with only positive CARs or with both positive and negative CARs. When the other two classification methods BEST and AVE are used, the associative classifier still performs comparably to other well-known rule-based and associative classifiers on average of 20 datasets.

4.1.3 Effect of negative CARs

To validate the effect of negative CARs in the associative classifier, we compare rules+- with SUM and AVE to their corresponding alternatives: rules+ with SUM and AVE. The average classification accuracy is higher when negative CARs are included. We also compare the count of wins and losses of rules+- when it is measured against rules+ on SUM and AVE classification methods, when the negative CARs are integrated, both of them win their positive alternatives by 12 times and only loses 2 and 3 times, respectively. It demonstrates the power of negative CARs. They indeed help us get more reliable and more accurate classification results on most datasets.

4.1.4 Effect of rule pruning

In the rule pruning phase, due to the absence of negative rule pruning strategies in the literature, we propose a novel rule pruning method to prune positive and negative CARs simultaneously. We compare the classification performance of three different scenarios: prune both positive and negative CARs, prune only positive CARs and without rule pruning. The comparison is performed on SUM classification methods. In Table 2, **Columns 2-4** show the classification results of these three scenarios on SUM classification method. The average accuracy of **Column 2** (prune both positive and negative CARs) is the highest and it wins 18 out of 20 datasets. Therefore, the proposed rule pruning method not only reduces the number of CARs in the classifier, but also improves the classification performance compared to the associative classifier pruning only positive CARs and the associative classifier without rule pruning phase.

4.2 Statistical Analysis

From Table 1, we can conclude that the associative classifier built with positive and negative CARs is an as good or even better associative classifier compared with other well-known classifiers and the associative classifier with only positive CARs; the SUM classification method is better than BEST and AVE. In Table 2, we show the proposed rule

Table 2: Comparison of rule pruning strategies.

Dataset	rules+- with SUM		
	prune +-	prune +	w/o prune
adult	83.2	82.3	81.9
anneal	91.9	65.7	86.7
breast	91.3	87.0	81.0
cylBands	72.2	63.7	63.7
flare	83.2	78.0	76.6
glass	72.4	60.7	69.6
heart	57.4	60.1	59.7
hepatitis	83.9	82.6	81.3
horseColic	75.8	73.1	72.0
ionosphere	82.3	75.7	75.5
iris	94.7	94.0	94.7
led7	73.3	74.2	73.9
letRecog	64.9	55.1	52.6
mushroom	100.0	97.9	97.7
pageBlocks	91.2	90.4	89.8
penDigits	91.3	83.8	86.8
pima	68.0	65.6	65.1
soybean	90.5	78.2	61.6
wine	91.6	91.6	91.6
zoo	94.1	77.2	94.1
Average	**82.7**	76.9	77.8

pruning strategy is effective. These conclusions are obtained mainly by measuring average classification accuracies and winning times. Although they give us some intuition about the lead of a certain classifier, a certain rule pruning or a classification strategy, the conclusion is not forceful since the dominance is unsurpassed over all 20 datasets.

To better validate the conclusions we get, we use Demsar's [18] method, conducting a set of non-parametric statistical tests to compare different classifiers over multiple datasets. In the first step, Friedman test is applied to measure if there is a significant difference between different classification models on Table 1. We first rank different classifiers on each dataset separately, r_i^j denotes the rank of the j-th of k classifiers on i-th of N datasets. Then the average rank of j-th classifier is computed as:

$$R_j = \frac{1}{N}\sum_i r_i^j$$

In the null hypothesis, the average ranks of different classifiers are equivalent, and the Friedman statistic is:

$$\chi_F^2 = \frac{12N}{k(k+1)}(\sum_j R_j^2 - \frac{k(k+1)^2}{4})$$

with $k-1$ degrees of freedom, when $N > 10$ and $k > 5$. If the Friedman statistic exceeds a critical value, the null hypothesis is rejected and we conduct post-hoc tests to make pairwise comparisons between different classifiers, otherwise, there is no statistically significant differences among the k classifiers over these N datasets.

The Friedman statistics of 13 classifiers from Table 1 exceeds the critical value, so we continue to use Wilcoxon signed-ranks test to compare the differences between different classifiers pairwisely. In Wilcoxon signed-ranks test, suppose d_i denotes the classification accuracy difference on the i-th of N datasets. We then rank the difference d_i according to their absolute values, if ties occur, average ranks are assigned. Next, the sum of ranks R^+, R^- are calculated on datasets in which the second classifier outperforms the first classifier and the first classifier outperforms the second classifier, respectively:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$

Let T be the smaller value of these two sums, when $N \geq 20$, Wilcoxon W statistic tends to form a normal distribution, then we can use z-value to evaluate the null hypothesis that there is no statistical difference between these two classifiers. The z-score is:

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}$$

If $z < -1.96$ then the corresponding p-value is smaller than 0.05, therefore, the null hypothesis is rejected.

A series of Wilcoxon signed-ranks test from Table 1 and Table 2 are listed in Table 3. It shows the count of wins, losses, ties and corresponding p-value for pairwise post-hoc comparisons. **Rows 2-9** show the comparisons of our associative classifier rules+- (SUM) with the other 8 well established classifiers. Our associative classifier always wins more than half of the 20 datasets, but the only strong conclusion we draw is that our method is significantly better than C4.5 and ARC-PAN. ARC-PAN is an associative classifier most similar to our method which also uses the negative CARs, however, it fails to consider the statistical dependency of the discovered CARs. The statistically significant difference between our method and ARC-PAN is very appealing. It shows the power of introducing statistical dependency in the associative classification problem. **Rows 10-11** show SUM method is significantly better than BEST and AVE. Through **Rows 12-13**, we can find that when the negative CARs are included, the associative classifier is significantly better than that with only positive CARs. **Rows 14-16** indicate the effect of the proposed rule pruning strategy, the difference between pruning only positive CARs and without pruning is not statistically significant although pruning only positive rules wins 14 times. But when we also prune negative CARs, the classification performance is greatly improved, the p-value is very small. Therefore, by applying the proposed rule pruning strategy, we get a much better classifier with higher accuracy and fewer rules.

4.3 Running Time Comparison

To test the efficiency of the proposed associative classifier, we compare the running time of rules+- (SUM) with three associative classifiers, CBA, CMAR and ARC-PAN. We only report the results on 6 datasets due to space limit. The results are shown in Table 4. As can be seen from the table, the proposed classifier is faster than the three contender associative classifiers in many cases.

5. CONCLUSION AND FUTURE WORK

In this paper, we introduce a novel associative classifier which is built on statistically significant positive and neg-

Table 3: Statistical analysis of Table 1 and Table 2; (*) indicates a statistically significant difference.

row ID	comparisons	wins	losses	ties	p-value
2	rules+-(SUM) vs. C4.5 *	13	7	0	0.011
3	rules+-(SUM) vs. FOIL	12	8	0	0.33
4	rules+-(SUM) vs. CBA	11	9	0	0.24
5	rules+-(SUM) vs. CMAR	12	7	1	0.42
6	rules+-(SUM) vs. CPAR	12	6	2	0.058
7	rules+-(SUM) vs. PAN *	15	5	0	0.014
8	rules+-(SUM) vs. NB	12	7	1	0.32
9	rules+-(SUM) vs. SVM	12	7	1	0.31
10	rules+-(SUM vs. BEST) *	14	4	2	0.028
11	rules+-(SUM vs. AVE) *	13	5	2	0.021
12	SUM(rules+- vs. rules+) *	12	2	6	0.033
13	AVE(rules+- vs. rules+) *	13	4	3	0.011
14	SUM(prune+- vs. prune+) *	17	2	1	0.001
15	SUM(prune+- vs. w/o) *	15	2	3	0.001
16	SUM(prune+ vs. w/o)	14	4	2	0.528

Table 4: Comparison of running time.

Dataset	CBA	CMAR	ARC-PAN	rule+-(SUM)
adult	583.7s	221.6s	561.2s	**177.0**s
cylBands	19.7s	5.9s	74.1s	**3.8**s
ionosphere	57s	95.6s	243.9s	**36.2**s
letRecog	5.9s	**5.2**s	129.2s	10.7s
penDigits	88.6s	257.2s	301.5s	**49.8**s
wine	25.4s	23.2s	98.0s	**11.2**s

ative CARs. The proposed associative classifier consists of three steps: rule generation, rule pruning and rule classification. In the first phase, we extend the Kingfisher algorithm by pushing the rule constraint in the rule generation phase to enable the discovery of statistically significant positive and negative CARs. After the rule generation step, there are still many noisy CARs which may jeopardize the classification phase or overfit the model, therefore we propose a novel rule pruning strategy to prune both positive and negative CARs simultaneously. At the last step, we present and compare different rule classification methods to ensure the correct prediction of unlabeled data.

The experimental results are very encouraging. Our associative classifier achieves a comparably good or even better classification result when measured against other classifiers. Meanwhile, it is also computational efficient. By integrating negative CARs in the classifier, the classification performance indeed improves compared to the classifiers built with only positive CARs. We also propose a novel rule pruning strategy to prune positive and negative CARs simultaneously. The pruning not only reduces the number of CARs, but also greatly improves the performance of the classification. Three different rule classification methods are presented and compared, the SUM method works best, it indicates that to classify a new instance, we should sum up the confidence values of all matching rules and make a class label prediction based on the summations from different classes. A two step statistical test are used to validate all these conclusions.

In the future, we aim to further improve the Kingfisher algorithm to enable the discovery of positive and negative CARs more efficiently. Another challenging task is to find more effective rule pruning strategies post rule generation to further improve the classification performance.

6. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proc. of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 18–24, 1998.

[2] C. C. Aggarwal and P. S. Yu. Mining associations with the collective strength approach. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):863–873, 2001.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data*, pages 207–216, 1993.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Intl. Conf. on Very Large Data Bases*, pages 487–499, 1994.

[5] L. Antonie, J. Li, and O. Zaiane. Negative association rules. In *Frequent Pattern Mining*, pages 135–145. Springer, 2014.

[6] M.-L. Antonie and O. R. Zaïane. Text document categorization by term association. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 19–26, 2002.

[7] M.-L. Antonie and O. R. Zaïane. An associative classifier based on positive and negative rules. In *Proc. of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 64–69, 2004.

[8] M.-L. Antonie and O. R. Zaïane. Mining positive and negative association rules: an approach for confined rules. In *Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 27–38, 2004.

[9] M.-L. Antonie, O. R. Zaïane, and R. C. Holte. Learning to use a learned model: A two-stage approach to classification. In *Proc. of the 6th IEEE Intl. Conf. on Data Mining*, pages 33–42, 2006.

[10] B. Arunasalam and S. Chawla. Cccs: a top-down associative classifier for imbalanced class distribution. In *Proc. of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 517–522, 2006.

[11] K. Bache and M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[12] R. J. Bayardo Jr. Brute-force mining of high-confidence classification rules. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 123–126, 1997.

[13] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proc. of the 1997 ACM SIGMOD Intl. Conf. on Management of Data*, pages 265–276, 1997.

[14] F. Coenen. The LUCS-KDD software library. `http://cgi.csc.liv.ac.uk/~frans/KDD/Software/`, 2004.

[15] W. W. Cohen. Fast effective rule induction. In *Proc. of the 12th Intl. Conf. on Machine Learning*, pages 115–123, 1995.

[16] C. Cornelis, P. Yan, X. Zhang, and G. Chen. Mining positive and negative association rules from large databases. In *Proc. of the 2006 IEEE Conf. on Cybernetics and Intelligent Systems*, pages 1–6, 2006.

[17] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[18] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[19] R. O. Duda, P. E. Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[20] W. Hämäläinen. Efficient discovery of the top-k optimal dependency rules with fisher's exact test of significance. In *Proc. of the 10th IEEE Intl. Conf. on Data Mining*, pages 196–205, 2010.

[21] W. Hämäläinen. Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and Information Systems*, 32(2):383–414, 2012.

[22] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data*, pages 1–12, 2000.

[23] Y. S. Koh and R. Pears. Efficiently finding negative association rules without support threshold. In *Proc. of the 20th Australian Joint Conf. on Artificial Intelligence*, pages 710–714, 2007.

[24] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 369–376, 2001.

[25] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 80–86, 1998.

[26] J. R. Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[27] J. R. Quinlan and R. M. Cameron-Jones. Foil: A midterm report. In *Proc. of the 1993 European Conf. on Machine Learning*, pages 3–20, 1993.

[28] S. Russell and P. Norvig. Artificial intelligence: a modern approach. 1995.

[29] A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proc. of the 14th Intl. Conf. on Data Engineering*, pages 494–502, 1998.

[30] W.-G. Teng, M.-J. Hsieh, and M.-S. Chen. On the mining of substitution rules for statistically dependent items. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 442–449, 2002.

[31] W.-G. Teng, M.-J. Hsieh, and M.-S. Chen. A statistical framework for mining substitution rules. *Knowledge and Information Systems*, 7(2):158–178, 2005.

[32] D. R. Thiruvady and G. I. Webb. Mining negative rules using grd. In *Proc. of the 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 161–165, 2004.

[33] F. Verhein and S. Chawla. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proc. of the 7th IEEE Intl. Conf. on Data Mining*, pages 679–684, 2007.

[34] H. Wang, X. Zhang, and G. Chen. Mining a complete set of both positive and negative association rules from large databases. In *Proc. of the 12th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining*, pages 777–784, 2008.

[35] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *Proc. of the 2005 SIAM Intl. Conf. on Data Mining*, pages 205–216, 2005.

[36] G. I. Webb. Discovering significant rules. In *Proc. of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data mining*, pages 434–443, 2006.

[37] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[38] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information System*, 22(3):381–405, 2004.

[39] X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *Proc. of the 3rd SIAM Intl. Conf. on Data Mining*, pages 331–335, 2003.