Exploiting Statistically Significant Dependent Rules for Associative Classification

Jundong Li[*]

Computer Science and Engineering
Arizona State University, USA
jundongl@asu.edu

Osmar R. Zaiane

Department of Computing Science
University of Alberta, Canada
zaiane@ualberta.ca

Abstract

Established associative classification algorithms have shown to be very effective in handling categorical data such as text data. The learned model is a set of rules that are easy to understand and can be edited. However, they still suffer from the following limitations: first, they mostly use the support-confidence framework to mine classification association rules which require the setting of some confounding parameters; second, the lack of statistical dependency in the used framework may lead to the omission of many interesting rules and the detection of meaningless rules; third, the rule generation process usually generates a sheer number of rules which puts in question the interpretability and readability of the learned associative classification model.

In this paper, we propose a novel associative classifier, SigDirect, to address the above problems. In particular, we use Fisher's exact test as a significance measure to directly mine classification association rules by some effective pruning strategies. Without any threshold settings like minimum support and minimum confidence, SigDirect is able to find non-redundant classification association rules which express a statistically significant dependency between a set of antecedent items and a consequent class label. To further reduce the number of noisy rules, we present an instance-centric rule pruning strategy to find a subset of rules of high quality. At last, we propose and investigate various rule classification strategies to achieve a more accurate classification model. Experimental results

[*] The work was done when the author was at University of Alberta

on real-world datasets show that SigDirect achieves better performance in terms of classification accuracy when measured with state-of-the-art rule based and associative classifiers. Furthermore, the number of rules generated by SigDirect is orders of magnitude smaller than the number of rules found by other associative classifiers, which is very appealing in practice.

Keywords. Associative Classification; Rules; Statistical Significance

1 Introduction

The concept of association rule mining was first introduced by Agrawal et al. [1] and extensively studied in the past two decades [2, 19, 3]. Association rules describe correlation between items in a transaction database. Assume a transaction database $\mathcal{D}$ consists of a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$, then an association rule is an implication of the form "$X \to Y(support, confidence)$", where X and Y are disjoint subsets of $\mathcal{I}$. The support indicates the probability that X and Y appear together in the transaction database. The strength of the rule is measured by the confidence, which is the conditional probability of Y given X. The problem of discovering association rules consists of generating rules that have a support and a confidence value higher than given thresholds. It has a variety of applications ranging from market basket analysis [18], web link analysis [28] to spatial colocation pattern discovery [22, 20].

Classification is another canonical task in the data mining and machine learning community. Given a set of attributes for an object, a classifier tries to assign the object to one or more pre-defined classes. A typical classification method consists of two steps: firstly, it builds a model on the training dataset whose attributes and class labels are known in advance; then the ability of the model to correctly classify objects in the test dataset is evaluated.

Recent studies on associative classification integrates association rule mining and classification together [25, 23, 4, 9]. These associative classifiers have proven to achieve competitive classification accuracies as decision trees [27], rule inductions [26, 13], naïve-bayes [15] as well as some probabilistic methods [24]. Besides, instead of taking a greedy algorithm as most rule-based classifiers, associative classification directly mines the complete set of rules to avoid missing any important ones. Another advantage of associative classification is that each individual rule in the model is human readable. To classify an object, associative classifiers first adopt association rule mining techniques to mine classification association rules (CARs) with given support-confidence thresholds and constrain the consequent of the rule to be a class label. Then a subset of CARs after pruning are selected to form the classifier, the selection is usually made by utilizing the database coverage heuristic [25]. Finally, once the classifier is built, it chooses one or more matching CARs to make predictions on the test dataset.

The existing associative classification methods mine the complete set of CARs mostly in an apriori-like fashion [2] or through a FP-growth way [19]. Although the rule generation process might be slightly different, all of them use

Table 1: An illustrative example of *type 1* error and *type 2* error.

Items	Class Label	Frequency
x	c_1	4400
y	c_2	80
z	c_2	5480
x, y	c_1	20
y, z	c_2	15
x, y, z	c_3	5

the support-confidence framework to find CARs for classification. However, it is difficult to determine the appropriate support and confidence thresholds for each dataset without any prior knowledge. Furthermore, traditional association rule mining methods are based on frequency to prune infrequent patterns. The strength of a rule is decided afterwards with its confidence value. Therefore, CARs cannot capture the actual statistical dependency between attributes and corresponding class. In the worst case, it may only find spurious CARs while leaving statistically significant dependent CARs undiscovered. These two types of scenarios are called *type 1* error and *type 2* error. Table 1 shows an example of these two types of errors.

Example 1 *A transaction database is shown in Table 1. Let* $\text{min\_support} = 1\%$ *and* $\text{min\_confidence} = 50\%$*. On one hand, through an Apriori-like or a FP-growth method, we generate some CARs. The CAR: $y \rightarrow c_2$ is among them because its support is 1% and confidence is around 79% which meets the support-confidence thresholds. Although the confidence value is high, there is a very weak dependency between y and c_2, because the support of c_2 is much higher than the support of y. In other words, y might happen to appear together with c_2, and in fact, they are more likely to be independent of each other. This is a typical example of type 1 error. On the other hand, it misses an strong CAR: $(x, y, z) \rightarrow c_3$. The CAR is not found because it has a very low support, which is 0.05%, but the confidence value is 100%. Besides, c_3 and itemsets (x, y, z) always co-occur which demonstrates that it is a CAR with strong dependency and the missing of this CAR is considered as an example of type 2 error.*

To avoid missing any strong CARs, most associative classifiers maintain a small minimum support threshold, but it is still possible to encounter the *type 2* error and at the same time it introduces a new problem: association rule mining methods end up generating a huge number of CARs making them impossible to be manually edited and even defeating the readability of the classification model. From another perspective, some post-processing strategies have been proposed to alleviate the *type 1* error [23, 7, 11, 32], but the discovered CARs are still not statistically significant and are more or less confronted with *type 1* error and *type 2* error. In addition, even though that we could find statistically significant CARs, it is still not clear how to reduce noisy CARs and how to make use of multiple informative CARs for a final classification model.

Therefore, in this paper, we propose a novel associative classifier, SigDirect (**S**tatistically S**IG**nificant **D**ependent Class**I**fication Association **R**ul**E**s for **C**lassifica**T**ion). The main contributions of this work are as follows:

- We propose a novel associative classifier, SigDirect. It achieves a competitive or even better classification performance as the state-of-the-art rule based and associative classifiers while generates an order of magnitude less of classification association rules.

- By pushing the rule constraint in the Kingfisher algorithm, we are able to find the complete set of CARs that show statistically significant dependencies efficiently.

- To reduce noisy CARs, we propose an instance-centric rule pruning strategy to find a globally optimal CAR for each instance in the training dataset without jeopardizing the classification accuracy.

- In the classification phase, we propose and investigate different rule classification methods to study how to make a label prediction with multiple matching CARs.

The remainder of the paper is organized as follows: Section 2 gives a view of related work on associative classifiers. In Section 3 we introduce the general steps to build the proposed associative classifier SigDirect. Section 4 presents the experimental results. We summarize our work in Section 5.

2 Related Work

This section first gives a brief introduction about associative classification and then introduces some popular associative classifiers.

The first reference to using association rules as CARs is credited to [10], while the first classifier using these CARs, CBA, was introduced in [25] and later improved in CMAR [23], ARC-AC and ARC-BC [4]. The idea is very straightforward. Given a training dataset modeled with transactions where each transaction contains all features of an object in addition to the class label of the object, we can constrain the mining process to generate association rules that always have as consequent a class label. In other words, the problem consists of finding the subset of strong association rules of the form $X \rightarrow C$, where C is a class label and X is a conjunction of features. After modeling the dataset into transactions, there are three steps in building an associative classifier:

- Rule Generation: In this phase, a mining algorithm is used to find classification association rules (CARs) of the form $set\_of\_items \rightarrow class\_label$ given the minimum support and minimum confidence thresholds. CBA and ARC use an apriori-like [2] fashion to mine the complete set of CARs, while CMAR utilizes the FP-growth [19] method for CARs generation. Both methods push the rule consequent constraint in the rule generation process.

- Rule Pruning: The rule generation process usually generates a large number of CARs, especially when the minimum support threshold is very low. Pruning techniques are used to discover the best subset of CARs that can cover the training dataset, meanwhile, they weed out noisy CARs that may mislead or overfit the classification model. Database coverage [25] is the most widely used pruning method which checks the extent of exposure of CARs in the training dataset.

- Classification: In this phase, the model is able to make a class label prediction for a new unlabeled object. How to utilize the set of left CARs to make a correct prediction is a challenging problem. CBA [25] classifies an object using the matching CAR with the highest confidence. However, making a prediction with a single CAR may lead to poor results. CMAR [23] adopts a chi-square weighting scheme to make a prediction, while ARC [4] predicts new objects using the average confidence of the selected matching CARs within a confidence margin.

There are some other variants for associative classification: Harmony [30] is an example which directly mines CARs, it directly finds the highest confident 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 [5] is a two-stage classification model that is able to automatically learn to select the rules for classification. In the first stage, an associative classifier is learned by standard techniques. Second, multiple predefined features are computed on the associative classifier, they act as input to a neural network model to weigh different features of the associative classifier to achieve a more accurate classification model. CCCS [7] uses a new measure, "Complement Class Support" (CCS) to mine positively correlated CARs to tackle the imbalanced classification problem. It forces the measure of CCS 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 [29] is another associative classifier designed for imbalanced data. It integrates another new measure, "Class Correlation Ratio" (CCR) into the statistically significant rules, the classifier works comparably on balanced dataset and outperforms other associative classifiers on imbalanced dataset. ARC-PAN [6] is the first associative classifier that uses both positive and negative CARs. It proposes to add Pearsons correlation coefficient on the basis of the support-confidence framework to mine positively and negatively correlated CARs. The ability of negative CARs have been demonstrated by their usage in the classification phase. Li and Zaiane [21] proposed to leverage both positive and negative CARs that show statistically significant dependencies for classification and the proposed classifier achieves competitive and even better performance compared with other rule based and associative classifiers.

3 Proposed Method

This section introduces the proposed associative classifier, SigDirect. Similar as most existing associative classifiers, SigDirect consists of three phases: 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 Basic Notations and Definitions

Definition 1 Dependency of a CAR
Let $\mathcal{D}$ be a transaction database, it consists of a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ and a set of class labels $C = \{c_1, c_2, ..., c_n\}$. Each transaction T in $\mathcal{D}$ is associated with a set of items X and a particular class label c_k, where $X \subseteq \mathcal{I}$ and $c_k \in C$. A CAR is in the form of $X \to c_k$, the antecedent part and the consequent class label of the CAR is dependent if and only if $P(X, c_k) \neq P(X)P(c_k)$, where $P(X)$ denotes the probability of itemset X.

Definition 2 Fisher's exact test
The dependency of the CAR $X \to c_k$ is considered to be statistically significant at level α, if the probability p of observing equal or stronger dependency in a dataset complying with a null hypothesis is not greater than α. In the null hypothesis, X and c_k are assumed to be independent of each other. The probability p, i.e., p-value, can be calculated by Fisher's exact test [16, 17]:

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

where $\sigma(X)$ denotes the support count of X. The significance level α is usually set to be 0.05.

Definition 3 Confidence
The confidence of the CAR $X \to c_k$ is:

$$conf(X \to c_k) = \frac{\sigma(X, c_k)}{\sigma(X)}$$

Definition 4 Parent and Child CAR
Let the CAR $X \to c_k$ be as before. The CAR $Y \to c_k$ is considered as its parent CAR if $Y \subsetneq X$ and $|Y| = |X| - 1$. Meanwhile, $X \to c_k$ is considered as the child CAR of the rule $Y \to c_k$.

Definition 5 Non-redundant CARs
The CAR $X \to c_k$ is non-redundant, if there does not exist any CARs in the form of $Y \to c_k$ such that $Y \subsetneq X$ and $p_F(Y \to c_k) < p_F(X \to c_k)$.

Definition 6 Minimality
The CAR $X \to c_k$ is minimal, if and only if $X \to c_k$ is non-redundant, and, there does not exist any CARs in the form of $Z \to c_k$ such that $X \subsetneq Z$ and $p_F(Z \to c_k) < p_F(X \to c_k)$.

3.2 Rule Generation

To find the relevant rules for classification, SigDirect first needs to generate the complete set of statistically significant dependent CARs. It means to find rules in the form of $X \to c_k$ which has a relevant small p_F-value, i.e., $p_F(X \to c_k) \leq \alpha$. Since the p_F-value is not a monotonic property, it is impossible for us to do some pruning as apriori-like algorithms. One possible solution is to enumerate the whole search space. However, the size of the whole search space is $|\mathcal{P}(\mathcal{I})|.|C|$, where $\mathcal{P}(\mathcal{I})$ is the power set of $\mathcal{I}$, it grows exponentially with the size of antecedent items. Recently, Kingfisher [16, 17] was proposed to find the complete set of rules that show statistically significant dependencies. Still, 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 CARs, we extend the Kingfisher algorithm by pushing the rule constraint in the rule generation phase. First, two theorems in [16, 17] are given as follows:

Theorem 1 *[16, 17] 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 $p_F(X \to A) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$; if $\sigma(A) \leq \frac{|\mathcal{D}|}{2}$, then for any $B \in \mathcal{R}$, $X \subseteq \mathcal{R} \backslash \{A, B\}$, it has $p_F(XA \to B) \geq \frac{\sigma(A)!\sigma(\neg A)!}{|\mathcal{D}|!}$. 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 *[16, 17] 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 $\sigma(X) \leq \sigma(A)$ holds, then it has $p_F(XQ \to A) \geq \frac{\sigma(\neg X)!\sigma(A)!}{|\mathcal{D}|!(\sigma(A)-\sigma(X))!}$.*

Given these two theorems, we derive three corollaries that enable us to generate 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 count 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 $\mathcal{I}_{rest}$ (Figure 1). Since

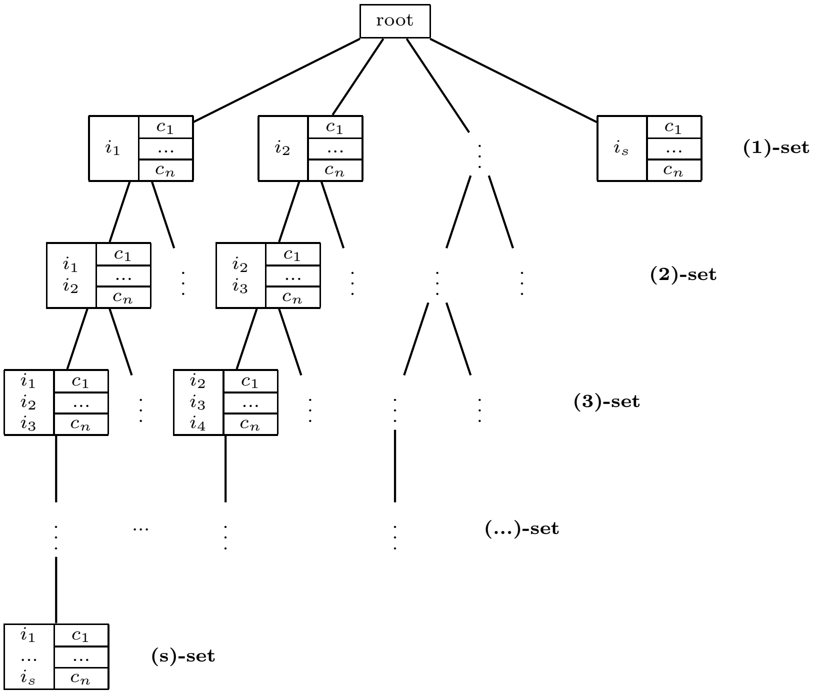

Figure 1: Enumeration of the whole search space.

the enumeration tree lists the whole search space, for each node in the enumeration tree, we check all the n possible CARs $X \to c_k(X \subseteq \mathcal{I}_{rest}, k \in \{1, ..., n\})$ to see if they are statistically significant, where X denotes the antecedent itemsets in the corresponding node, as illustrated in Figure 1.

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

Proof: Corollary 2 can be considered as a special case of Theorem 2 when c_k is the consequent part of a rule. □

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

Definition 7 *The CAR $X \to c_k$ is defined as* **PSS**, *i.e., "Potentially Statistically Significant", if it meets either of the following conditions: (1) $\sigma(X) \leq \sigma(c_k)$ holds, and the lower bound $\frac{\sigma(\neg X)!\sigma(c_k)!}{|\mathcal{D}|!(\sigma(c_k)-\sigma(X))!}$ is smaller than or equal to α; (2) $\sigma(X) > \sigma(c_k)$ 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 $X \rightarrow c_k$ is PSS, then any of its parent rule $Y \rightarrow c_k$ is also PSS, where $Y \subsetneq X$ and $|Y| = |X| - 1$.*

Proof: There are two situations making $X \rightarrow c_k$ 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 $Y \rightarrow c_k$ is also PSS. The second situation is when $X \rightarrow c_k$ but $lowerbound(p_F(XQ \rightarrow c_k)) < \alpha$, 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 $lowerbound(p_F(YQ \rightarrow c_k)) < \alpha$, i.e., rule $Y \rightarrow c_k$ is PSS. □

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 difficult, because we have to consider its whole subtree. We use a well-proven result from [16, 17] that if $P(c_k|X) = 1$, the corresponding CAR $X \rightarrow c_k$ is minimal. In other words, the property of minimality can be detected by calculating the conditional probability of 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.

3.3 Rule Pruning

In the rule generation phase, we have taken the non-redundancy property into consideration. However, the number of statistically significant dependent CARs could still be very large. One possible disadvantage of a large number of CARs is that it could contain some noisy information which may mislead the classification process. Another drawback is that a large number of CARs will make the classification process slower. This could be a problem in applications where fast responses are required. Moreover, in classification applications where evidence checking is required, rule-based models are an advantage but a large number of rules is a significant drawback and defeats the purpose. In order to reduce the number of CARs in the classification phase, many associative classifiers take a sequential database coverage paradigm. However, the final set of CARs may

```
Data: Transaction database $\mathcal{D}$, set of antecedent itemset $\mathcal{I}$, class set $C$, significance
      level $\alpha = 0.05$.
Result: Statistically significant dependent classification association rules set $\mathcal{R}$.
Prune impossible antecedent items with Corollary1 1;
$\mathcal{I}_{rest}$: the arranged and renamed antecedent itemset;
Create root node and level-1 nodes, set $l = 1$;
while $l \leq |\mathcal{I}_{rest}|$ do
    for each candidate rule $r$ in level $l$ do
        if all parent CARs of $r$ are PSS and not minimal then
            if $p_F(r) \leq \alpha$ then
                if $r$ is non-redundant then
                    if $r$ is minimal then
                        $r$.minimal = true;
                        $\mathcal{R}$.add($r$);
                    else
                        $\mathcal{R}$.add($r$);
                    end
                end
            end
        else
            prune CAR $r$ and all its children CARs from the enumeration tree;
        end
    end
    $l = l + 1$;
end
```

Algorithm 1: Rule Generation Phase.

```
Data: Set of statistically significant dependent CARs $\mathcal{R}$ found in the rule generation
      phase, transaction database $\mathcal{D}$.
Result: A subset of CARs $\mathcal{R}_{new}$ for the classification process.
for each instance $t$ in the transaction database $\mathcal{D}$ do
    Scan the set of CARs in $\mathcal{R}$ to find the matching CAR $r$, i.e.,
    ($r.antecedent \subseteq t.antecedent$ and $r.class = t.class$) with the highest confidence
    value;
    if $r \notin \mathcal{R}_{new}$ then
        $\mathcal{R}_{new}$.add($r$);
        $r$.count = 1;
    else
        $r$.count += 1;
    end
end
```

Algorithm 2: Rule Pruning Phase.

not be the globally best CARs for some instances in the training dataset. In order to reduce the number of CARs and to find the globally best CARs for all training instances, we propose an instance-centric rule pruning approach to select the best CAR for each instance in the training dataset, the best CAR is defined as the matching CAR with the highest confidence value. Each candidate CAR may be selected by multiple training instances, therefore, each candidate CAR is associated with an attribute "count", it records how many times the CAR is selected in the pruning process. The detailed algorithm is shown in Algorithm 2.

3.4 Classifying New Instances

After the rule pruning phase, the subset of the most statistically significant dependent CARs form the actual classifier. In this phase, we utilize the built classifier to make new predictions. Given a new instance without a class label, the classification process searches the subset of CARs matching the new instance to make a class label prediction. This subsection discusses three approaches that we take to label new instances.

A simple solution is to select the matching CAR in $\mathcal{R}_{new}$ with the highest confidence value or the lowest p_F-value and assign its label to the new instance. Another alternative is to divide all matching CARs into groups according to their class labels. The groups are then ordered according to the average confidence value or average p_F-value. The class that has the highest average confidence value or the lowest average p_F-value will be assigned to the new instance. However, these two classification heuristics are often biased to minority classes. To solve this problem, an intuitive way is to calculate the total confidence value or total p_F-value instead of the average values. But p_F-value is different from confidence, the lower the value, the better the CAR is, therefore, simply sum up the p_F-value is not a solution. Therefore, we transform the p_F-value to its log scale, subsequent steps are on a log-transformed value.

Then, we propose three different heuristics, denoted as S1, S2 and S3, to consider the sum of $ln(p_F)$, sum of confidence and sum of $ln(p_F)$.confidence of matching CARs in each class, respectively:

- S1: Calculate the sum of $ln(p_F)$ of matching CARs in each class, the class label of the new instance is determined by the class of the lowest value.
- S2: Calculate the sum of confidence of matching CARs in each class, the class label of the new instance is determined by the class of the highest value.
- S3: Calculate the sum of $ln(p_F)$.confidence of matching CARs in each class, the class label of the new instance is determined by the class of the lowest value.

Algorithm 3 describes three heuristic classification methods for an unlabeled new instance.

4 Experiments

4.1 Datasets

We evaluate our SigDirect method on 20 datasets from UCI Machine Learning Repository [8]. In these datasets, the numerical attributes have been discretized by the author of [12], the discretization strategy is different from that used in [25, 23], thus the classification performance may be different from the results reported before. All the following experimental results on each dataset are reported as an average of a 10-fold cross validation.

```
**Data**: A new instance $o$ to be classified. Set of CARs $\mathcal{R}_{new}$ from rule pruning phase.
**Result**: Class label of the new instance $o$.
$T = \emptyset$ ;                                          // set of CARs matching $o$
for each CAR $r$ in $\mathcal{R}_{new}$ do
    $i = 1$;
    while $i \leq r.count$ do
        if $r.antecedent \subseteq o.antecedent$ then
            $T$.add($r$);
        end
        $i = i + 1$;
    end
end
divide $T$ into $n$ subsets by class labels: $T_1, T_2, ..., T_n$;
// Classification with S1
for each subset $T_1, T_2, ..., T_n$ do
    sum up the $ln(p_F)$ values of matching CARs in each subset;
end
assign the class with the lowest sum of $ln(p_F)$ value to the instance $o$;
// Classification with S2
for each subset $T_1, T_2, ..., T_n$ do
    sum up the confidence values of matching CARs in each subset;
end
assign the class with the highest sum of confidence value to the instance $o$;
// Classification with S3
for each subset $T_1, T_2, ..., T_n$ do
    sum up the $ln(p_F)$.confidence values of matching CARs in each subset;
end
assign the class with the lowest sum of $ln(p_F)$.confidence value to the instance $o$;
```

Algorithm 3: Classification Phase.

4.2 Classification Accuracy

We evaluate our SigDirect with three different classification strategies S1, S2, S3 against two rule-based classifiers C4.5 [27] and FOIL [26], two associative classifiers CBA [25], CMAR [23] and a hybrid between rule-based and associative classifier CPAR [31] on the previous mentioned 20 discretized UCI datasets. The results are reported in the form of average classification accuracy over 10-folds. All classification methods are evaluated on the same generated 10-folds to ensure a fair comparison. The parameters of C4.5 are set as default values [27]. In FOIL, we allow a maximum of 3 attributes in the antecedent of a rule. In CBA and CMAR, the minimum support is set to be 1%, the minimum confidence is 50%, the maximum number of antecedent items and the maximum number of mined CARs are set to be 6 and $80,000$, respectively. In CPAR, we also follow the same parameter settings as [31], minimum gain threshold set to 0.7, total weight threshold to 0.05 and decay factor to 2/3.

Table 2 presents the classification accuracy of the following methods: C4.5, FOIL, CBA, CMAR, CPAR and our SigDirect method with three different classification heuristics S1, S2 and S3. Along with the accuracy result, the name of the dataset, the number of antecedent attributes on the discretized transaction dataset, the number of classes and the number of records are also reported.

Table 2: Comparison of classification results: C4.5, FOIL, CBA, CMAR, CPAR and SigDirect

Dataset	#attr	#cls	#rec	C4.5	FOIL	CBA	CMAR	CPAR	SigDirect		
									S1	S2	S3
adult	97	2	48842	78.8	**84.6**	84.2	81.3	77.3	84.0	83.9	84.0
anneal	73	6	898	76.7	**98.8**	94.5	90.7	95.1	92.1	91.6	93.5
breast	20	2	699	91.5	89.3	**94.1**	89.9	93.0	91.4	91.7	91.6
cylBands	124	2	540	69.1	74.1	76.1	**76.5**	70.0	74.8	76.1	75.2
flare	39	9	1389	82.1	83.8	84.2	**84.3**	63.9	80.6	81.8	81.6
glass	48	7	214	65.9	66.5	68.4	**71.1**	64.9	66.3	69.2	68.7
heart	52	5	303	**61.5**	55.2	57.8	56.2	53.8	55.8	58.1	57.4
hepatitis	56	2	155	84.1	77.8	42.2	79.6	75.5	83.2	**85.2**	82.6
horseColic	85	2	368	70.9	**83.4**	78.8	82.3	81.2	81.0	80.2	81.0
ionosphere	157	2	351	84.6	86.6	32.5	**91.5**	88.9	90.0	90.3	90.3
iris	19	3	150	91.3	94.0	93.3	94.0	**94.7**	89.3	89.3	89.3
led7	24	10	3200	**73.9**	60.5	73.1	73.2	71.3	73.5	73.4	73.6
letRecog	106	26	20000	50.4	50.0	32.5	28.3	58.2	48.0	**58.8**	52.4
mushroom	90	2	8124	92.8	99.5	46.7	**100.0**	98.5	**100.0**	**100.0**	**100.0**
pageBlocks	46	5	5473	92.0	92.4	90.9	90.1	**92.5**	91.2	91.2	91.2
penDigits	89	10	10992	70.5	84.1	**92.3**	87.4	80.5	84.3	88.4	84.6
pima	38	2	768	71.7	71.9	74.6	74.4	74.0	74.6	**75.1**	74.6
soybean	118	19	683	60.3	88.0	89.2	88.1	83.1	89.3	**89.9**	89.6
wine	68	3	178	75.8	88.2	49.6	92.7	88.2	92.7	**93.3**	92.7
zoo	42	7	101	91.0	93.1	40.7	93.0	**94.1**	**94.1**	**94.1**	**94.1**
Average				76.7	81.1	69.8	81.2	79.9	81.8	**83.1**	82.4

As can be observed from Table 2, the proposed SigDirect with S2 achieves the best overall classification accuracy, followed by SigDirect with S3 and SigDirect with S1. All of these three classifiers outperform C4.5, FOIL, CBA, CMAR and CPAR on the average over the 20 datasets.

To have a more fair comparison between these classifiers, we show how many times the classifier is the best and how many times it is the runner-up. Table 3 shows the comparison results, SigDirect with S2 (classify by the sum of confidence) is still the best among these classifiers. It wins 7 out of 20 datasets, i.e., 35% of all datasets, and is the runner-up 5 times. CMAR, in the second place, wins in 5 datasets and gets the runner-up three times. Combing the comparison results from Table 2 and Table 3 together, SigDirect with S2 is always the best, SigDirect with S1 and SigDirect with S3 can be considered as competitive classifiers. It demonstrates that in the classification accuracy aspect, our SigDirect classification method can be viewed as a better classifier when measured against state-of-the-art rule based and associative classifiers.

4.3 Number of Rules

In associative classification, the number of CARs before and after rule pruning are both very important indicators to measure a classifier. On one hand, if we get a small number of CARs after rule generation, people are able to sift through these rules to determine validity, to choose a subset of them or even to

Table 3: Best and runner-up counts comparison.

Classifiers	Best	Runner-up
C4.5	1	2
FOIL	3	4
CBA	2	4
CMAR	6	2
CPAR	3	3
SigDirect with S1	3	4
SigDirect with S2	8	5
SigDirect with S3	2	5

edit them to inject domain knowledge not reflected in the training data. Moreover, rule pruning strategies are possible since these rules are more readable. On the other hand, a small number of rules after rule pruning can make the classification phase faster. In addition, after rule pruning, because of transparency of the rules, manually updating some rules is favourable and practical in many applications if the number of rules is reasonable. Therefore, in this subsection, we evaluate the number of CARs generated by our SigDirect algorithm and the number of CARs after pruning strategy. Table 4 shows the number of rules of two associative classifiers CBA, CMAR and our SigDirect method. The number of rules before and after rule pruning are both presented. We also list the number of rules in C4.5, FOIL and CPAR in Table 4. All the parameter settings of these classifiers are the same as the previous subsection. We also need to notice that CBA and CMAR use Apriori and FP-growth to generate CARs. The number of rules generated by these two methods should be the same. In CBA and CMAR, the rule generation stops if the number of rules is larger than 80,000, but even in this situation, we can find that the number of CARs generated by SigDirect is much smaller than that generated by CBA and CMAR. In most datasets, the number is even an order of magnitude smaller. It can also be observed, after rule pruning, the number of rules by SigDirect is smaller than that by CBA in 13 datasets, when compared with CMAR, the number of rules are smaller in all 20 datasets. Furthermore, in 17 datasets, the number of rules is below 100, which make it more readable and more manually editable.

All in all, SigDirect dramatically reduce the number of CARs compared with CBA and CMAR in the rule generation phase without jeopardizing accuracy and even improving it. After rule pruning, the number of rules for classification is still smaller than that by CBA and CMAR. The overall smaller number of rules makes SigDirect superior to other associative classifiers when there is slight difference between classification accuracies. The number of rules remains comparable and even smaller than the case of C4.5, FOIL and CPAR.

4.4 Effects of Pruning Strategies and Classification Heuristics

In SigDirect, we propose an instance-centric method to do rule pruning to reduce the number of CARs. Here, we first compare the effect of this pruning strategy

Table 4: Comparison of the number of rules generated by different classifiers

Dataset	C4.5	FOIL	CPAR	CBA		CMAR		SigDirect	
				before rule pruning	after rule pruning	before rule pruning	after rule pruning	before rule pruning	after rule pruning
adult	1176.5	229.4	84.6	87942.6	691.8	87942.6	2982.5	136.1	50.1
anneal	17.0	29.5	25.2	89101.9	27.3	89101.9	208.4	340.5	23.2
breast	8.8	13.9	6.0	2711.4	13.5	2711.4	69.4	19.8	10.1
cylBands	37.2	45.5	35.8	64194.3	135.4	64194.3	622.8	700	34.2
flare	54.4	95.5	48.1	11910.2	115.1	11910.2	347.1	600.9	45.1
glass	14.8	47.1	34.8	10171.0	63.7	10171.0	274.5	340.5	23.2
heart	23.9	63.4	44.0	41899.4	78.4	41899.4	464.2	19.8	10.1
hepatitis	8.1	23.6	14.3	181441.1	2.3	181441.1	165.7	185.7	19.0
horseColic	25.6	64	19	178353.3	116.4	178353.3	499.9	319.9	33.3
ionosphere	18.3	24.9	22.8	95242.1	27.3	95242.1	272.7	284.3	15.8
iris	8.4	7.9	7.4	171.0	12.3	171.0	63.4	8.1	6.0
led7	63.2	79.1	31.7	453.6	71.2	453.6	206.3	269.0	108.7
letRecog	1565.2	559.3	789.1	2402.9	151.4	2402.9	1132.5	19252.9	468.0
mushroom	121.2	11.7	11.1	104666.3	2.0	104666.3	102.6	906.4	16.0
pageBlocks	16.3	43.6	29.9	1546.6	7.6	1546.6	80.6	230.1	25.8
penDigits	758.3	163.3	135.1	91125.5	657.6	91125.5	4501.5	8406	212.0
pima	24.4	58.7	21.7	1769.6	43.2	1769.6	203.3	116.8	33.2
soybean	57.1	46.3	76.6	26912.0	65.8	26912.0	293.2	314242	31.5
wine	12.8	15.9	15.2	82120.6	4.7	82120.6	122.7	8721.7	11.1
zoo	5.3	9.9	16.9	82616.7	2.0	82616.7	35.0	4597	7.8

with the database coverage paradigm. In Table 5, the classification results with these two different rule pruning strategies are presented and compared. As can be observed, the classification accuracy indeed improves when we take the instance-centric pruning strategy, no matter what kind of classification heuristics are used. The average classification accuracy is higher around 1% to 2%.

In order to investigate the efficacy of some measurement M ($ln(p_F)$, confidence or $ln(p_F)$.confidence), to see if classifying by the sum of M can overcome the bias problem caused by classifying with only the best rule or by the average of M, we compare S1, S2 and S3 with their corresponding alternatives (B1, A1), (B2, A2) and (B3, A3). The compared classification heuristics B1, A1, B2, A2, B3 and A3 are listed below:

- B1: Select the matching rule with the lowest $ln(p_F)$ value, the class label of the new instance is determined by the selected rule
- A1: Calculate the average value of $ln(p_F)$ for matching rules in each class, the class label of the new instance is determined by the class of the lowest value
- B2: Select the matching rule with the highest confidence value, the class label of the new instance is determined by the selected rule
- A2: Calculate the average of confidence value for matching rules in each class, the class label of the new instance is determined by the class of the highest value

Table 5: Comparison of instance-centric and database coverage pruning methods.

Dataset	S1		S2		S3	
	instance centric	database coverage	instance centric	database coverage	instance centric	database coverage
adult	**83.9**	**83.9**	**83.9**	83.2	**84.1**	83.6
anneal	**96.8**	96.1	**94.0**	88.0	**96.7**	94.4
breast	**91.4**	90.7	**91.7**	91.3	**91.6**	90.7
cylBands	**74.4**	73.3	**73.7**	72.0	**74.4**	73.9
flare	**83.0**	80.3	**84.2**	83.2	**84.2**	83.7
glass	**66.8**	66.4	69.6	**72.0**	**68.7**	67.8
heart	56.4	**57.1**	**58.1**	56.8	**57.4**	56.4
hepatitis	**83.2**	82.6	**85.2**	83.2	**82.6**	81.9
horseColic	**81.3**	80.2	**80.7**	76.7	**81.3**	80.7
ionosphere	87.2	**88.9**	**85.5**	85.0	87.2	**88.9**
iris	**94.0**	93.3	94.0	**94.7**	**93.3**	**93.3**
led7	**73.8**	73.5	**73.8**	73.5	**73.7**	72.7
letRecog	**48.2**	46.7	58.8	**61.8**	**52.6**	51.1
mushroom	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
pageBlocks	**91.2**	90.7	**91.2**	91.1	**91.2**	90.7
penDigits	**84.3**	81.5	88.4	**90.3**	**84.6**	81.5
pima	**74.6**	68.5	**75.1**	67.7	**74.6**	68.6
soybean	**89.5**	87.6	**90.0**	89.6	**89.8**	88.4
wine	92.1	**92.7**	**92.7**	88.2	92.1	**92.7**
zoo	**94.1**	93.1	**94.1**	93.1	**94.1**	**94.1**
Average	**82.3**	81.3	**83.2**	82.1	**82.7**	81.8

- B3: Select the matching rule with the lowest $ln(p_F)$. confidence value, the class label of the new instance is determined by the selected rule
- A3: Calculate the average of $ln(p_F)$.confidence value for matching rules in each class, the class label of the new instance is determined by the class of the lowest value

As shown in Table 6, S1, S2 and S3 have a better classification performance than their counterpart (B1, A1), (B2, A2), (B3, A3), respectively. Table 7 shows the count of wins, losses and ties for S1, S2 and S3 when compared with their alternatives.

From these two tables, it can be concluded that the classification heuristics in the "A" category are always the worst, "B" category heuristics are better than "A" category, but are still not as good as "S" category heuristics. Therefore, the classification heuristic that classifying a new instance by the sum of measurement M ($ln(p_F)$, confidence or $ln(p_F)$.confidence) of all matching rules in SigDirect indeed helps to improve the classification performance. When the measurement M is the rule's confidence, the associative classifier is the best.

4.5 Statistical Analysis

From Table 2, we can conclude that our SigDirect algorithm gets better classification performance compared to other methods and the confidence is a better

Table 6: Comparison of classification heuristics S1 with (B1, A1), S2 with (B2, A2) and S3 with (B3, A3)

Dataset	SigDirect								
	S1	B1	A1	S2	B2	A2	S3	B3	A3
adult	**84.0**	73.4	71.4	**83.9**	81.0	77.4	**84.0**	80.6	80.7
anneal	92.1	**92.9**	92.1	91.6	**92.9**	90.9	**93.5**	90.5	89.8
breas	**91.4**	89.7	88.1	**91.7**	90.0	90.0	**91.6**	91.4	90.7
cylBands	**74.8**	69.3	68.7	**76.1**	73.7	73.1	**75.2**	69.3	69.8
flare	**80.6**	78.2	77.0	81.8	**82.1**	**82.1**	**81.6**	81.4	81.5
glass	**66.3**	64.5	62.6	69.2	**71.5**	68.7	**68.7**	65.9	65.4
heart	**55.8**	55.4	**55.8**	**58.1**	51.5	52.1	**57.4**	55.4	56.4
hepatitis	**83.2**	75.4	73.5	85.2	**85.8**	83.2	**82.6**	81.3	80.0
horseColic	**81.0**	**81.0**	78.5	**80.2**	70.9	74.7	**81.0**	**81.0**	78.8
ionosphere	90.0	**90.9**	87.7	**90.3**	86.0	85.2	90.3	**91.2**	88.9
iris	**89.3**	**89.3**	**89.3**	**89.3**	**89.3**	**89.3**	**89.3**	**89.3**	**89.3**
led7	**73.5**	69.7	68.3	**73.4**	70.9	69.8	**73.6**	70.5	69.3
letRecog	**48.0**	27.7	18.8	**58.8**	54.4	49.2	**52.4**	42.3	39.5
mushroom	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
pageBlocks	**91.2**	90.6	90.6	**91.2**	90.7	90.7	**91.2**	90.7	90.7
penDigits	**84.3**	68.0	50.9	**88.4**	87.4	84.2	**84.6**	74.6	63.3
pima	**74.6**	**74.6**	**74.6**	75.1	75.1	**75.3**	**74.6**	**74.6**	**74.6**
soybean	**89.3**	85.5	81.6	89.9	89.9	**90.2**	**89.6**	87.1	87.0
wine	**92.7**	91.6	91.0	**93.3**	84.8	80.9	92.7	**93.3**	91.0
zoo	**94.1**	93.1	93.1	**94.1**	93.1	93.1	**94.1**	**94.1**	**94.1**
Average	**81.8**	78.0	75.7	**83.1**	81.1	80.0	**82.4**	80.2	79.0

measure when measured against $ln(p_F)$ and $ln(p_F)$.confidence in the classification phase. Table 4 shows that SigDirect gets a small number of CARs both before and after rule pruning phase. Table 5 and Table 6 indicate the superiority of the proposed instance-centric rule pruning strategy and the summation effect when perform classification, respectively. These conclusions are obtained mainly by measuring average classification accuracies and winning times. Although it gives 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 [14] method, con-

Table 7: Classification heuristics S1, S2, S3 compared with their alternatives

	wins	losses	ties
S1 vs. B1	14	2	4
S1 vs. A1	15	0	5
B1 vs. A1	14	1	5
S2 vs. B2	12	4	4
S2 vs. A2	15	3	2
B2 vs. A2	10	4	6
S3 vs. B3	13	2	5
S3 vs. A3	16	0	4
B2 vs. A2	11	4	5

ducting 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 2. We first rank different classifiers on each dataset separately, r_i^j denotes 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:

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

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 classifiers, otherwise, there is no statistically significant difference among the k classifiers over these N datasets.

The Friedman statistic of 8 classification methods from Table 2 exceeds the critical value, so we continue to use Wilcoxon signed-ranks test to compare the differences between different methods pairwisely. In Wilcoxon signed-ranks test, 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 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 tests from Table 2, Table 4, Table 5 and Table 6 are listed in Table 8. It shows the count of wins, losses, ties and corresponding p-value for pairwise post-hoc comparisons. **Rows 2-6** show the differences between the proposed SigDirect algorithm with 5 other well-known rule based and associative classifiers. SigDirect is significantly better than C4.5, FOIL, CBA and CPAR and is as good as CMAR. From **Rows 7-8**, we can

Table 8: Statistical analysis of Table 2, Table 4, Table 5 and Table 6 ; (*) indicates statistically significant difference with a p-value of 0.05.

row	comparisons	wins	losses	ties	p-value
2	SigDirect(S2) vs. C4.5*	17	2	1	0.001
3	SigDirect(S2) vs. FOIL*	13	6	1	0.040
4	SigDirect(S2) vs. CBA*	14	5	1	0.033
5	SigDirect(S2) vs. CMAR	12	5	3	0.136
6	SigDirect(S2) vs. CPAR*	13	6	1	0.010
7	SigDirect(S2) vs. SigDirect(S1)	10	4	6	0.158
8	SigDirect(S2) vs. SigDirect(S3)	11	5	4	0.214
9	#bef. prun: SigDirect vs. CBA*	18	2	0	0.004
10	#bef. prun: SigDirect vs. CMAR*	18	2	0	0.006
11	#aft. prun: SigDirect vs. CBA	9	11	0	0.435
12	#aft. prun: SigDirect vs. CMAR*	19	1	0	0.001
13	S1: instance-cetric vs. db coverage*	15	3	2	0.001
14	S2: instance-cetric vs. db coverage	15	4	1	0.056
15	S3: instance-cetric vs. db coverage*	15	2	3	0.008
16	SigDirect(S1) vs. SigDirect(B1)*	16	2	2	0.001
17	SigDirect(S1) vs. SigDirect(A1)*	15	3	2	0.001
18	SigDirect(S2) vs. SigDirect(B2)*	13	5	2	0.006
19	SigDirect(S2) vs. SigDirect(A2)*	15	3	2	0.001
20	SigDirect(S3) vs. SigDirect(B3)*	15	1	4	0.001
21	SigDirect(S3) vs. SigDirect(A3)*	16	2	2	0.001

see that the difference between three different classification heuristics is not statistically significant, but since S2 gets a more higher average classification accuracy, we choose to use S2 in the classification phase. **Rows 9-12** list the number of CARs differences between SigDirect, CBA, CMAR before and after rule pruning phase. SigDirect gets a significantly smaller number of CARs in the rule generation phase when measured against CBA and CMAR, the number of CARs is still significantly smaller than CMAR even after the rule pruning phase. The effect of the instance-centric rule pruning strategy is shown in **Rows 13-15**, when classification heuristics S1 and S3 are used, the instance-centric method is significantly better than the database coverage method. Although the difference is not statistically significant with S2, the corresponding p-value is still very close to 0.05 and the instance-centric strategy wins 15 time and only loses 4 times. Therefore, the instance-centric rule pruning strategy is better than the database coverage method. The last 6 rows compare different classification heuristics, the “S” category is much better than the “B” and “A” category. In this way, to classify a new instance, we should choose to sum up the measure M of multiple matching CARs to make a final prediction.

5 Conclusions

In this paper, we study the problem using statistically significant dependent CARs for classification and propose a novel associative classifier SigDirect. The proposed associative classifier consists of three steps: rule generation, rule pruning and rule classification. In the first phase, we adapt upon the Kingfisher algorithm by pushing the rule constraint in the rule generation phase to enable the discovery of statistically significant 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 an instance-centric rule pruning strategy to select a subset of CARs of high quality. 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. The proposed SigDirect classifier achieves better classification results on many real-world datasets when measured against state-of-the-art rule based and associative classifiers. Apart from the promising classification performance, the number of CARs before and after rule pruning phase are both very small, making SigDirect more appealing than other methods when there is little difference in classification performance. The number of CARs before rule pruning phase is even an order of magnitude smaller than that by CBA and CMAR. After rule pruning phase, the number of rules is still very small. The small set of CARs in both phases makes it possible and practical for users to sift through them to edit and update according to their own needs, which can be very important in many applications.

References

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

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[3] Luiza Antonie, Jundong Li, and Osmar Zaiane. Negative association rules. In *Frequent Pattern Mining*, pages 135–145. Springer, 2014.

[4] M-L Antonie and Osmar R Zaiane. Text document categorization by term association. In *Proceedings of 2002 IEEE International Conference on Data Mining (ICDM)*, pages 19–26, 2002.

[5] M-L Antonie, Osmar R Zaiane, and Robert C Holte. Learning to use a learned model: A two-stage approach to classification. In *Proceedings of 6th IEEE International Conference on Data Mining (ICDM)*, pages 33–42, 2006.

[6] Maria-Luiza Antonie and Osmar R Zaïane. An associative classifier based on positive and negative rules. In *Proceedings of the 9th ACM SIGMOD*

workshop on Research Issues in Data Mining and Knowledge Discovery, pages 64–69, 2004.

[7] Bavani Arunasalam and Sanjay Chawla. Cccs: a top-down associative classifier for imbalanced class distribution. In *Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 517–522, 2006.

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

[9] Elena Baralis and Paolo Garza. A lazy approach to pruning classification rules. In *Proceedings of 2002 IEEE International Conference on Data Mining (ICDM)*, pages 35–42, 2002.

[10] Roberto J Bayardo Jr. Brute-force mining of high-confidence classification rules. In *Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 123–126, 1997.

[11] Loïc Cerf, Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. A parameter-free associative classification method. In *Proceedings of 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, pages 293–304, 2008.

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

[13] William W Cohen. Fast effective rule induction. In *Proceedings of 12th International Conference on Machine Learning (ICML)*, pages 115–123, 1995.

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

[15] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[16] Wilhelmiina Hämäläinen. Efficient discovery of the top-k optimal dependency rules with fisher's exact test of significance. In *Proceedings of 10th IEEE International Conference on Data Mining (ICDM)*, pages 196–205, 2010.

[17] Wilhelmiina 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.

[18] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

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

[20] Jundong Li, Aibek Adilmagambetovm, Mohomed Shazan Mohomed Jabbar, Osmar R Zaïane, Alvaro Osornio-Vargas, and Osnat Wine. On discovering co-location patterns in datasets: A case study of pollutants and child cancers. *GeoInformatica*, 2016.

[21] Jundong Li and Osmar R Zaïane. Associative classification with statistically significant positive and negative rules. In *Proceedings of the 24th ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2015.

[22] Jundong Li, Osmar R Zaïane, and Alvaro Osornio-Vargas. Discovering statistically significant co-location rules in datasets with extended spatial objects. In *Data Warehousing and Knowledge Discovery*, pages 124–135. Springer, 2014.

[23] Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of 2001 IEEE International Conference on Data Mining (ICDM)*, pages 369–376, 2001.

[24] Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 40(3):203–228, 2000.

[25] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of 4th International Conference on Knowledge Discovery and Data mining (KDD)*, pages 80–86, 1998.

[26] J Ross Quinlan and R Mike Cameron-Jones. Foil: A midterm report. In *Proceedings of 1993 European Conference on Machine Learning (ECML)*, pages 1–20, 1993.

[27] John Ross Quinlan. *C4.5: Programs for Machine Learning*, volume 1. Morgan kaufmann, 1993.

[28] Myra Spiliopoulou. Web usage mining for web site evaluation. *Communications of the ACM*, 43(8):127–134, 2000.

[29] Florian Verhein and Sanjay Chawla. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proceedings of 7th IEEE International Conference on Data Mining (ICDM)*, pages 679–684, 2007.

[30] Jianyong Wang and George Karypis. Harmony: Efficiently mining the best rules for classification. In *Proceedings of 2005 SIAM International Conference on Data Mining (SDM)*, pages 205–216, 2005.

[31] X Yin and J Han. Cpar: Classification based on predictive association rules. In *Proceedings of 3rd SIAM International Conference on Data Mining (SDM)*, pages 331–335, 2003.

[32] Osmar R Zaïane and Maria-Luiza Antonie. On pruning and tuning rules for associative classifiers. In *Proceedings of 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES)*, pages 966–973, 2005.