



School of IT  
Technical Report



**The University of Sydney**

USING SIGNIFICANT, POSITIVELY ASSOCIATED AND  
RELATIVELY CLASS CORRELATED RULES FOR  
ASSOCIATIVE CLASSIFICATION OF IMBALANCED DATASETS  
TECHNICAL REPORT 614

FLORIAN VERHEIN AND SANJAY CHAWLA  
SCHOOL OF INFORMATION TECHNOLOGIES  
THE UNIVERSITY OF SYDNEY

AUGUST 2007

# Using Significant, Positively Associated and Relatively Class Correlated Rules for Associative Classification of Imbalanced Datasets

Florian Verhein  
The School of Information Technologies  
University of Sydney  
fverhein@it.usyd.edu.au

Sanjay Chawla  
The School of Information Technologies  
University of Sydney  
chawla@it.usyd.edu.au

## Abstract

*The application of association rule mining to classification has led to a new family of classifiers which are often referred to as “Associative Classifiers (ACs)”. The advantage of ACs is that they are rule-based and thus lend themselves to an easier interpretation. Another advantage that ACs enjoy is that they are based on a global search criterion, unlike other rule-based classifiers – e.g. decision trees – which use a greedy search strategy.*

*Rule-based classifiers can play a very important role in applications such as medical diagnosis and fraud detection where “imbalanced data sets” are the norm and not the exception.*

*The focus of this paper is to extend and modify ACs for classification on imbalanced data sets using only statistical techniques. We combine the use of statistically significant rules with a new measure, the Class Correlation Ratio (CCR), to build an AC which we call SPARCCC. Experiments show that in terms of classification quality, SPARCCC performs comparably on balanced datasets and outperforms other AC techniques on imbalanced data sets. It also has a significantly smaller rule base and is much more computationally efficient.*

## 1 Introduction

Since the introduction of CBA [8] many variations on Associative Classifiers have been proposed in the literature [7, 2, 17, 15, 4, 5, 3, 13]. Most of the ACs are based on rules discovered using the *support-confidence* paradigm and the classifier itself is a collection of rules ranked using confidence or variation thereof. In many application domains, the data sets are imbalanced, i.e., the proportion of samples from one class is much smaller than the other class(es). Additionally, the smaller class is the class of interest. Unfortunately, the *support-confidence* framework does not perform

well in such cases. Can AC techniques be used for the imbalanced situation?

Recently Webb [16] has shown the value of using statistically significant rules and has demonstrated that many of the rules mined using *support-confidence* are spurious and are irregularities in the data rather than properties of the underlying population. We believe that the same holds true from rules used for classification. It is also well known that confidence has non-intuitive properties in imbalanced data sets. For example, high confidence rules can also be negatively correlated. In this paper we combine statistically significant rules with a new measure, the *Class Correlation Ratio (CCR)*, which leads to a better classifier. As we will show, *CCR* overcomes the weaknesses of *correlation* and *confidence*. Furthermore, our method does *not* use the *support-confidence* paradigm.

We make the following **contributions**:

- We propose the *Class Correlation Ratio (CCR)*, which measures the relative class correlation of a rule and overcomes the downsides of *correlation*. A high *CCR* is desirable because it means the rule is more positively correlated with the class it predicts than the alternative(s). *CCR* also forms the basis of an effective rule ranking method that does not require *confidence*. This method could also be beneficially employed in other algorithms.
- We prove that confidence and support are biased toward the majority class in imbalanced datasets in the context of *CCR*.
- We propose an Associative Classifier that is based purely on statistical techniques. We call the method **Significant, Positively Associated and Relatively Class Correlated Classification (SPARCCC)** because we use only rules that are statistically significant (using a one sided test so the antecedent is positively associated with the class), and where the antecedent is more correlated with the class than with

the other class(es). We also search directly for significant rules – using this to prune the search space. This classifier outperforms *support-confidence* based associative classifiers on imbalanced datasets. Since there are standard levels of significance, it is also relatively parameter free. Finally, since the rules are significant and relatively class correlated, they may be examined for insights into the data.

The the remainder of this paper is organised as follows: Section 2 gives a brief summary of AC. Section 3 describes a significance test we use and the *class correlation ratio*. Section 4 proves that confidence (and support) is biased against the minority class. Section 5 describes our technique in detail. We provide experiments in Section 6 and survey related work in Section 7.

## 2 Associative Classification (Background)

In **Association Rule Mining (ARM)** the data is a set of transactions  $T = \{t_1, \dots, t_{|T|}\}$ , each of which is a subset of the set of items:  $t_i \subseteq I, I = \{i_1, \dots, i_{|I|}\}$ . The *support* of an *itemset*  $X \subseteq I$  is  $sup(X) = |\{t_i : X \subseteq t_i \wedge t_i \in T\}|$ . An association rule  $X \rightarrow Y$  is an implication between two mutually exclusive itemsets  $X$  and  $Y$ . The *support* of  $X \rightarrow Y$  is  $sup(X \rightarrow Y) = sup(X \cup Y)$  and its *confidence* is  $conf(X \rightarrow Y) = sup(X \rightarrow Y) / sup(X)$ .

In the **Associative Classification** problem, we assume a discrete dataset  $D$  with attributes  $A = \{a_1, a_2, \dots, a_{|A|}\}$ , one of which is the class attribute  $a_c$ . In every instance  $d \in D$ , each attribute  $a_i \in A$  takes one of a finite number of possible values  $V_i = \{v_{i,1}, \dots, v_{i,|V_i|}\}$  so that  $d = [v_{1,j}, v_{2,k}, \dots, v_{|A|,l}]$  (for some  $j, k, \dots, l$ ). As an ARM task, the attribute-value pairs become items (Namely,  $i_{|V_1|+\dots+|V_{i-1}|+j} \equiv (a_i = v_{i,j})$ ) and the instances become corresponding transactions. The previous instance  $d$  then becomes a transaction  $t = \{(a_1 = v_{1,i}), (a_2 = v_{2,j}), \dots, (a_{|A|} = v_{|A|,k})\}$ . Clearly, there will be  $\sum_{i=1}^{|A|} |V_i| = |I|$  items and each transaction will have size  $|A|$ . Since the described mapping is a bijection, we can freely interchange *instances* and *transactions* when convenient. The **Classification Rule Mining** task is to find *interesting* rules  $X \rightarrow y$  where  $X$  is a set of *legal* (an attribute cannot occur more than once) attribute value pairs and  $y$  is one of the class attribute value pairs. By *interesting*, we mean rules that, in conjunction with other mined rules, are likely to perform well for classification of unseen data.

## 3 Significance and Class Correlation Ratio

There are strong arguments for mining statistically significant rules [16]. These also hold true when the rules are

	$X$	$\neg X$	$\Sigma$ rows
$y$	$a$	$b$	$a + b$
$\neg y$	$c$	$d$	$c + d$
$\Sigma$ cols	$a + c$	$b + d$	$n = a + b + c + d$

**Figure 1.**  $2 \times 2$  Contingency Table for  $X \rightarrow y$ . We will often use the notation  $[a, b; c, d]$ .

used for classification, as we would like to make a decision based on significant evidence. We are interested in rules  $X \rightarrow y$  that are statistically significant in the positively associated direction. Toward that end, we use **Fisher’s Exact Test (FET)** on *contingency tables* of the form of Figure 1. FET is an *exact test (permutation test)*. Given a table  $[a, b; c, d]$ , FET will find the probability ( $p$ -value) of obtaining the given table or a table where  $X$  and  $y$  are more *positively associated* under the null hypothesis that  $\{X, \neg X\}$  and  $\{y, \neg y\}$  are independent, and that the margin sums are fixed. The  $p$ -value is given by:

$$p([a, b; c, d]) = \sum_{i=0}^{\min(b,c)} \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!(a+i)!(b-i)!(c-i)!(d+i)!}$$

We only use rules whose  $p$ -values are below the level of significance desired. Rules that pass this test are therefore statistically significant in the positively associated direction. We also prune the search space using significance, rather than support, as outlined in Section 5.2. FET’s continuous approximation – the  $\chi^2$  test – could also be used, however it is less desirable as it cannot distinguish positive associations (it is a two-sided test).

In addition to statistical significance, **correlation** also forms a very important component of our technique. We are interested in rules  $X \rightarrow y$  where  $X$  is more *positively correlated with  $y$  than it is with  $\neg y$* . In this paper we use the following definition of correlation<sup>1</sup>:

$$c\hat{o}rr(X \rightarrow y) = \frac{sup(X \cup y) \cdot |T|}{sup(X) \cdot sup(y)} = \frac{a \cdot n}{(a+c) \cdot (a+b)}$$

$X$  and  $y$  are positively (negatively) correlated if  $c\hat{o}rr(X \rightarrow y) > 1 (< 1)$ , and independent otherwise. Note that  $c\hat{o}rr(X \rightarrow y) = I(X, y)$ , where  $I(X, y)$  is the *Interest Factor* [12]. This measure has downsides when used by itself. It is clear to see that increasing the size of the dataset

<sup>1</sup>To be more precise,  $c\hat{o}rr(X \rightarrow y)$  this is the *estimate* of  $corr(X \rightarrow y) = \frac{P(X \cup y \subseteq t)}{P(X \subseteq t) \cdot P(y \subseteq t)}$ , where  $corr(X \rightarrow y)$  is defined over the underlying process that generates the data. Also recall that *support* was defined as the number of transactions supporting an itemset – hence the  $|T|$  in  $c\hat{o}rr$ .

by increasing  $d$  (refer to Figure 1) will increase the correlation between  $X$  and  $y$  – even though it is actually increasing the association between  $\neg X$  and  $\neg y$ . The reverse holds for decreasing  $d$ . For example, consider the table  $T_1 = [100, 20; 20, 10]$  where  $X$  and  $y$  are have a strong association but  $\hat{c}orr(X \rightarrow y) = 1.04$  (almost independent!). If we increase  $d$  to get  $T_2 = [100, 20; 20, 200]$  then clearly  $\neg X$  and  $\neg y$  are strongly associated, but  $\hat{c}orr(\neg X \rightarrow \neg y) = 1.4$  while now  $\hat{c}orr(X \rightarrow y) = 2.36!$  This is clearly undesirable. This problem arises only in imbalanced datasets however – note how changing  $d$  changes the class distribution. We therefore do *not* search for positively correlated rules using it. When we speak of a rule being positively associated or correlated, we mean by using the *one sided* test of significance described above. The FET does not have this downside because of the constant margin sum restriction. Indeed,  $p(T_1) = 0.041$  (significant at the 0.05 level) and  $p(T_2) = 1.07 \cdot 10^{-44}$  (highly significant).

We use  $\hat{c}orr(\cdot)$  to measure how correlated  $X$  is with  $y$  compared to  $\neg y$ . That is, we use what we call the **Class Correlation Ratio (CCR)**:

$$CCR(X \rightarrow y) = \frac{\hat{c}orr(X \rightarrow y)}{\hat{c}orr(X \rightarrow \neg y)} = \frac{a \cdot (b + d)}{b \cdot (a + c)}.$$

This measures how much more positively the antecedent is correlated with the class it predicts, *relative* to the alternative class(es). This avoids the downsides of using an absolute correlation measure – indeed, terms cancel out. Furthermore, it makes a lot of sense intuitively – you would not want to use a rule that is more correlated with classes other than that it predicts! Returning to the example,  $CCR(\cdot) = 1.25$  for  $T_1$  and  $CCR(\cdot) = 9.17$  for  $T_2$ . This also says that  $X \rightarrow y$  is a better rule under  $T_2$  than under  $T_1$ . This is true – it is much more discriminative because under  $T_1$ ,  $y$  is already the majority class and therefore the rule does not provide much additional information. In fact, the *Information Gain* of using  $X \rightarrow y$  over  $\emptyset \rightarrow y$  is only 0.072 bits under  $T_1$  but is 0.215 bits under  $T_2$ . Recall also that the rule was much more significant under  $T_2$ . *We only use rules with  $CCR > 1$ , so that no rules are used that are more positively associated with the classes they do not predict. Furthermore, we use CCR in our Strength Score.*

#### 4 Relative Correlation Bias of Confidence (and Support) on Imbalanced Datasets

Confidence is widely used as a measure of strength of a classification rule  $X \rightarrow y$  because it is an *estimate* (the dataset is a sample) of the probability that, given the attribute-value pairs in  $X$  appear in a transaction (instance)  $t$  generated by the underlying process, the instance will have the class label  $y$ . That is,  $conf(X \rightarrow y) \sim P(y \in t | X \subset$

A	$sup(y) < sup(\neg y)$
B	$\hat{c}orr(X \rightarrow y) > \hat{c}orr(X \rightarrow \neg y)$ i.e. $CCR(X \rightarrow y) > 1$
B'	$\hat{c}orr(X \rightarrow y) < \hat{c}orr(X \rightarrow \neg y)$ i.e. $CCR(X \rightarrow y) < 1$
C	$conf(X \rightarrow y) > conf(X \rightarrow \neg y)$ $\equiv sup(X \rightarrow y) > sup(X \rightarrow \neg y)$
C'	$conf(X \rightarrow y) < conf(X \rightarrow \neg y)$ $\equiv sup(X \rightarrow y) < sup(X \rightarrow \neg y)$

**Figure 2. Statements for Lemma 1.**  $\neg y$  means all class attribute-values other than  $y$ .

$t$ ). The confidence of a *significant* rule (it does not make sense to use insignificant rules, and their confidences are unlikely to mean anything anyhow) is therefore a useful measure of the rule strength in classification – *but only in balanced datasets*: We show that *confidence* (and *support*, while we're at it) are biased toward the majority class under the CCR. This is useful for explaining why using confidence to rank rules for classification of imbalanced datasets can give poor performance. It also provides us with additional reasons to use CCR for ranking rules, and Section 5.1 describes how we can correct for the bias. In our previous example, note that  $conf(X \rightarrow y) = 0.83$  in both  $T_1$  and  $T_2$ , despite the rule being clearly better in  $T_2$ .

**Lemma 1** *Confidence (and support) are biased toward the majority class under the Class Correlation Ratio. Specifically (the statements in parentheses are defined in Figure 2):*

1. *If  $X \rightarrow y$  is more positively correlated than  $X \rightarrow \neg y$  but has a lower confidence (support), then  $y$  must be the minority class ( $B \wedge C' \implies A$ ).*
2. *If  $X \rightarrow y$  is more positively correlated and more confident (frequent) than  $X \rightarrow \neg y$ , we cannot say anything about whether  $y$  is the minority or majority class ( $B \wedge C \not\implies A$  and  $B \wedge C \not\implies \neg A$ ).*
3. *If  $y$  is the minority class and  $X \rightarrow y$  is more confident (frequent) than  $X \rightarrow \neg y$ , then it is also more positively correlated ( $A \wedge C \implies B$ ).*
4. *If  $y$  is the minority class and  $X \rightarrow y$  is less confident (frequent) than  $X \rightarrow \neg y$ , there is no relationship between the correlation of the rules ( $A \wedge C' \not\implies B$  and  $A \wedge C' \not\implies \neg B$ ).*
5. *If  $y$  is the minority class and  $X \rightarrow y$  is less positively correlated than  $X \rightarrow \neg y$ , it is also less confident (frequent) ( $A \wedge B' \implies C'$ ).*

6. If  $y$  is the minority class and  $X \rightarrow y$  is more positively correlated than  $X \rightarrow \neg y$ , then we cannot say anything about their confidences (supports) ( $A \wedge B \not\Rightarrow C'$  and  $A \wedge B \not\Rightarrow \neg C'$ ).

**Proof:**

1.  $C' \Rightarrow 1 > \text{sup}(X \cup y)/\text{sup}(X \cup \neg y)$ ,  
 $B \Rightarrow \text{sup}(X \cup y)/\text{sup}(X \cup \neg y) > \text{sup}(y)/\text{sup}(\neg y)$ ,  
hence  $B \wedge C' \Rightarrow A$ .
2. Counter examples: If  $\text{sup}(y) = 0.3 \cdot n = n - \text{sup}(\neg y)$ ,  $\text{sup}(X) = 0.5 \cdot n$ ,  $\text{sup}(X \cup y) = 0.3 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.2 \cdot n$  we contradict  $B \wedge C' \Rightarrow \neg A$ . If  $\text{sup}(y) = 0.7 \cdot n = n - \text{sup}(\neg y)$ ,  $\text{sup}(X) = 0.8 \cdot n$ ,  $\text{sup}(X \cup y) = 0.6 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.2 \cdot n$  we contradict  $B \wedge C' \Rightarrow A$ .
3.  $C' \Rightarrow \frac{\text{sup}(X \cup y) \cdot n}{\text{sup}(X) \cdot \text{sup}(y)} > \frac{\text{sup}(X \cup \neg y) \cdot n}{\text{sup}(X) \cdot \text{sup}(\neg y)} \cdot \frac{\text{sup}(\neg y)}{\text{sup}(y)}$ , which, using  $A$ , is greater than  $\frac{\text{sup}(X \cup \neg y) \cdot n}{\text{sup}(X) \cdot \text{sup}(\neg y)} = \text{corr}(X \rightarrow \neg y)$ .
4. Counter examples: Let  $\text{sup}(y) = 0.3 \cdot n = n - \text{sup}(\neg y)$ .  
If  $\text{sup}(X \cup y) = 0.2 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.3 \cdot n$  and  $\text{sup}(X) = 0.5 \cdot n$  we contradict  $A \wedge C' \not\Rightarrow \neg B$ .  
If  $\text{sup}(X \cup y) = 0.2 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.6 \cdot n$  and  $\text{sup}(X) = 0.8 \cdot n$  we contradict  $A \wedge C' \not\Rightarrow B$ .
5.  $B' \Rightarrow \frac{\text{sup}(X \cup y)}{\text{sup}(X)} < \frac{\text{sup}(X \cup \neg y)}{\text{sup}(X)} \cdot \frac{\text{sup}(y)}{\text{sup}(\neg y)}$ , which, using  $A$ , is less than  $\frac{\text{sup}(X \cup \neg y)}{\text{sup}(X)} = \text{conf}(X \rightarrow y)$ .
6. Counter examples: Let  $\text{sup}(y) = 0.3 \cdot n = n - \text{sup}(\neg y)$  and  $\text{sup}(X) = 0.5 \cdot n$ . If  $\text{sup}(X \cup y) = 0.2 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.3 \cdot n$  we contradict  $A \wedge B \Rightarrow \neg C'$ . If  $\text{sup}(X \cup y) = 0.3 \cdot n$  and  $\text{sup}(X \cup \neg y) = 0.2 \cdot n$  we contradict  $A \wedge B \Rightarrow C'$ .

Suppose we have a two class problem and  $y$  describes the minority class. 3) tells us that if  $X \rightarrow y$  is more confident than  $X \rightarrow \neg y$ , then it is also more positively correlated. However, the reverse does not hold as described by 4). That is, if  $X \rightarrow \neg y$  is more confident than  $X \rightarrow y$ , then it may or may not be more positively correlated. This means we may have a highly confident rule for the majority class,  $X \rightarrow \neg y$  (that is more confident than  $X \rightarrow y$ ), but is actually less positively correlated than  $X \rightarrow y$  – very undesirable! In the opposite case, 5) tells us that a rule in the minority class,  $X \rightarrow y$ , with lower relative correlation will also have lower confidence than  $X \rightarrow \neg y$ . Again, this does not hold for the majority class. *Since higher confidence (support) for a rule in the minority class implies higher relative correlation ( $CCR > 1$ ), and lower relative correlation ( $CCR < 1$ ) in the minority class implies lower confidence, but neither of these are true for the majority class, we say that confidence (support) tends to bias the majority class – because confidence (support) and  $CCR$  can only ‘contradict’ each other in the majority class.* In a related matter, 1) tells us that if  $X \rightarrow y$  is more positively correlated than  $X \rightarrow \neg y$  but is less confident, then  $y$  must be the minority class. Again, the reverse does not hold in general. *All this means that if we choose high confidence (support) rules (e.g. by using a*

*threshold) we are more likely to miss rules that are relatively positively correlated (have  $CCR > 1$ ) applying to the minority class than in the majority class. Furthermore, when ranking by confidence (support) we are likely to use rules that are relatively negatively correlated (have  $CCR < 1$ ) predicting the majority class over relatively positively correlated ( $CCR > 1$ ) rules predicting the minority class.* Experiments on imbalanced datasets also readily verify that ranking based on confidence is biased toward the majority class.

**Example 1** Consider an imbalanced dataset with  $\text{sup}(y) = 15$  and  $\text{sup}(\neg y) = 100$ . A possible contingency table is  $[5, 10; 10, 90]$ . Even though  $\text{conf}(X \rightarrow y) = \frac{1}{3} < \text{conf}(X \rightarrow \neg y) = \frac{2}{3}$ ,  $X$  has a significant positive association with  $y$  ( $p_{\text{value}} = 0.02$ ). Furthermore,  $\text{corr}(X \rightarrow y) = 2.56$  and  $\text{corr}(X \rightarrow \neg y) = 0.77$  so this rule has a high  $CCR$  ( $CCR = 3.32 \gg 1$ ) and is thus a very good rule.

## 5 SPARCCC

There are four components to SPARCCC. We describe how they fit together here, and outline them in detail in the subsequent sections.

1. The *Interestingness and Rule Ranking* technique (Section 5.1) determines which of the *potentially interesting* rules mined by the *Search and Pruning Strategy* are in fact *interesting*. It also determines the *Strength Score* we assign to the resulting *interesting* rules.
2. The *Search and Pruning Strategy* (Section 5.2) determines how the space of all possible rules is examined and pruned. This determines the candidate rules we consider as possibly being interesting – i.e. the *potentially interesting* rules. The choice of strategy determines the computational performance and we evaluate three possibilities.
3. The *Rule Selection Method* (Section 5.3) determines which of the *interesting* rules are to be used for classification. It makes use of the *Rule Ranking* strategy and outputs *selected* rules.
4. The *Classification Method* (Section 5.4) determines how we classify an unseen instance by using the *selected* rules. It makes use of the *Rule Ranking* strategy.

### 5.1 Interestingness and Rule Ranking

We perform the following tests to determine whether a *potentially interesting* rule is **interesting**:

- We check the significance of a rule  $X \rightarrow y$  by performing Fisher’s Exact Test on the contingency table of Figure 1, as earlier described. We record the  $p_{value}$ .
- We check whether  $corr(X \rightarrow y) > corr(X \rightarrow \neg y)$  ( $CCR(X \rightarrow y) > 1$ ). If this is not the case, the rule is not interesting because it is more correlated with the alternative class(es) than it is with the class it predicts.

The *interesting* rules – those that pass the above two tests – are candidates for the classification task.

In order to use the rules to make a classification, we need a ranking (ordering) of the rules. That is, we need a measure of how interesting a rule is, in the sense that it best captures the ability of the rule to make a correct classification. This ordering is defined by the **Strength Score** of the rule,  $SS(X \rightarrow y)$ . Since this paper focuses on significant and correlated rules, one might consider using  $SS_p(X \rightarrow y) = (1 - p_{value})$  to rank the *interesting* rules by their level of significance in the positively correlated direction. Experiments show that this does work, however the performance is not good enough to compete with other techniques. The reasons are intuitive: the  $p_{value}$  does not vary much, and by ranking according to it we do not take into account how likely the rule is to produce a correct classification.

Based on the discussions in Sections 3 we therefore use:

$$SS_{p,CCR}(X \rightarrow y) = (1 - p_{value}) \cdot CCR(X \rightarrow y)$$

Confidence is an *estimate* of the probability that, given  $X$  occurs,  $y$  will occur. Therefore in balanced datasets, choosing the rule with the highest confidence gives the highest expected probability of making a correct classification. Therefore, for comparison, we also evaluate:

$$SS_{p,conf}(X \rightarrow y) = (1 - p_{value}) \cdot conf(X \rightarrow y)$$

But as Lemma 1 shows, *confidence* has a bias toward the majority class. While  $SS_{p,conf}$  performs well on balanced datasets, it performs very poorly on imbalanced datasets. Recall that **a**) a highly confident rule predicting the majority class may in fact be more negatively correlated than the same rule predicting the other class(es), and **b**) a rule that is more positively correlated but predicts the minority class may have much lower confidence than the same rule predicting the other class(es). Now, our interestingness criteria above excludes case a), but it does not correct for the bias in confidence for less extreme cases and it does nothing to fix case b). We propose to correct this using  $CCR$ :

$$SS_{p,conf,CCR}(r) = (1 - p_{value}) \cdot conf(r) \cdot CCR(r)$$

For the rule  $r = X \rightarrow y$ . This works by giving poor rules a lower score (in comparison to better rules) and scaling up cases of b):  $CCR(X \rightarrow y) > 1$ .

In terms of a suitable classification performance  $P(\cdot)$ , experiments show that on relatively balanced datasets:

$$P(SS_{p,CCR}) \approx P(SS_{p,conf,CCR}) \approx P(SS_{p,conf})$$

While on imbalanced datasets:

$$P(SS_{p,CCR}) \gg P(SS_{p,conf,CCR}) \gg P(SS_{p,conf})$$

That is, the use of  $CCR$  achieves the highest performance on imbalanced datasets while performing comparably on balanced datasets. As expected, this agrees nicely with our discussions and theoretical results in Sections 3 and 4. Furthermore, note that in a completely balanced dataset,  $CCR(X \rightarrow y)$  reduces to  $\frac{sup(X \rightarrow y)}{sup(X \rightarrow \neg y)} = \frac{conf(X \rightarrow y)}{conf(X \rightarrow \neg y)}$ . This is a nice result, because it shows that the *Class Correlation Ratio* reduces to what we shall call the *Class Support Ratio* and the *Class Confidence Ratio*.

Finally, we note that the  $p_{value}$  has little impact in the final score, because it varies at most by the significance level. It’s inclusion therefore favors more significant rules only if the other components of  $SS$  are similar.

**Example 2** Recall Example 1 where a highly positively correlated and significant rule had a very low confidence of  $\frac{1}{3}$ , so  $SS_{p,conf} = 0.33$ . However,  $CCR(X \rightarrow y) = \frac{2.56}{0.77} = 3.33$ . Inclusion of this in the strength score raises it from 0.33 to  $SS_{p,conf,CCR} = 0.33 \cdot 3.33 = 1.09$ . In comparison, if the classes had been equally distributed, the rule would have been negatively correlated, insignificant and  $CCR(X \rightarrow y)$  would have been  $\frac{1}{2}$ . This demonstrates how  $CCR$  can be used to counteract the bias of  $conf(\cdot)$  in imbalanced datasets. Clearly,  $SS_{p,CCR} = 3.27$ .

## 5.2 Search and Pruning Strategies

The overall strategy is a bottom up item enumeration technique, as all the rules  $X' \rightarrow y : X' \subset X$  will be examined before  $X \rightarrow y$  and the search is over the item space (attribute-value space). The underlying algorithm used to do this is a variation of GLIMIT [14]. It performs this task in a depth first fashion. It uses linear space in the number of instances, linear time in the number of itemsets (classification rules and their antecedents) that need to be considered, and one pass over the dataset. While this is faster than alternatives such as Apriori [1] or FP-Growth [6], either of these could potentially be used.

The general idea of a rule being statistically significant is not anti-monotonic. To avoid examining the entire space<sup>2</sup>, we use search strategies that ensure the concept of being *potentially interesting* is anti-monotonic. That is,  $X \rightarrow y$

<sup>2</sup>The search space for finding classification rules is  $\prod_{i \in \{1, \dots, |A|\}} (|V_i| + 1)$

	$t : X \subset t$	$t : X - \{z\} \subset t \wedge z \notin t$	$t : X - \{z\} \subset t$
$t : y \in t$	$a = \text{sup}(X \rightarrow y)$	$b = \text{sup}(X - \{z\} \rightarrow y) - \text{sup}(X \rightarrow y)$	$a + b = \text{sup}(X - \{z\} \rightarrow y)$
$t : \neg y \in t$	$c = \text{sup}(X \rightarrow \neg y)$	$d = \text{sup}(X - \{z\} \rightarrow \neg y) - \text{sup}(X \rightarrow \neg y)$	$c + d = \text{sup}(X - \{z\} \rightarrow \neg y)$
	$a + c = \text{sup}(X)$	$b + d = \text{sup}(X - \{z\}) - \text{sup}(X)$	$a + b + c + d = \text{sup}(X - \{z\})$

**Figure 3. The contingency table  $[a, b; c, d]$  used to test for the significance of the rule  $X \rightarrow y$  in comparison to *one* of its generalizations  $X - \{z\} \rightarrow y$  for the Aggressive-S search strategy.**

might be considered as *potentially interesting* if and only if all  $\{X' \rightarrow y | X' \subset X\}$  have been found to be *potentially interesting*. We use the following search strategies:

- Select a new attribute-value in such a way that it makes a significant positive contribution to the rule, when compared to all *immediate* generalizations. Specifically, Figure 3 describes how we test for the significance of the rule  $X \rightarrow y$  in comparison to *one* of its generalizations  $X - \{z\} \rightarrow y$ . The rule  $X \rightarrow y$  is *potentially interesting* only if the test passes for all immediate generalizations  $\{X - \{z\} \rightarrow y : z \in X\}$ . This effectively tells us that, when compared to the immediate generalizations, all of the attribute-value pairs in the antecedent of the rule make a significant positive contribution to the rule’s association with the class. This technique prunes the search space most aggressively, as it performs  $|X|$  tests per rule. However, this also means that it greatly favors shorter rules, as they have fewer tests to pass. It is almost the same strategy as used by Webb [16]. In our experiments, this technique is called **Aggressive-S**
- Use FET as described in Section 3 and force it to be anti-monotonic. That is, if and only if all rules  $\{X - \{z\} \rightarrow y : z \in X\}$  are *potentially interesting*, then we use the contingency table of Figure 1 to determine whether  $X \rightarrow y$  is *potentially interesting*. Note that this is recursive. Also note, therefore, that all rules found to be *potentially interesting* are already half way to being *interesting*. In our experiments, this technique is called **Simple-S**, for simple significance based search. It performs only one test per rule and examines more of the search space.
- Use a minimum support threshold. All rules with  $\text{supp}(X \rightarrow y) \geq \text{minSup}$  are *potentially interesting*. We do this only as a point of comparison. In our experiments, this technique is called **Support**. Unlike the other techniques, it does not direct the search by significance.

For the first two techniques, we define  $\text{sup}(\emptyset) = |T|$ , the number of transactions (instances). This is necessary so

that we can evaluate a  $p_{\text{value}}$  for so-called “default rules” – rules with no antecedent. The  $p_{\text{values}}$  for these tend to be high, but we keep them in case no other rules match.

We advise using Aggressive-S first, as it is faster. If this runs quickly then Simple-S should be tried, as it allows more rules to be considered and may have slightly higher classification performance.

### 5.3 Rule Selection Method

The algorithm in Figure 4(a) returns the set of highest ranking rules so that each training instance is covered by (and correctly classified by) enough rules for it to have *minGroups* groups of rules, where each group is made up of rules with the same scores. This is a type of covering technique. We use the concept of groups, which is driven by the *Classification Method* below.

### 5.4 Classification Method

The algorithm in Figure 4(b) makes the decision based on the highest ranked (according to the *Strength Score*) matching rules. If there is one rule with the highest score, or multiple rules with the same score but predicting the same class, then the choice is straightforward – simply pick the class predicted by the rules. However, if there are multiple rules with the same score but predicting different classes, then we pick the class predicted by the majority of the rules in the group. However, there may be cases where there is no single majority class in the group. In this case, we first remove from consideration any classes that are not in the majority. Then, we use the next group of rules to make a decision between the remaining classes. We continue to do this until there is a majority. If we run out of rules, we make a random choice between the *remaining* classes. In addition to this, we ensure that we do not run out of rules for any class. For example, suppose we have 3 matching rules, and further suppose there are two rules predicting different classes in the top group. We cannot make a decision based on the top group alone. If we used the next group, we’d predict based on the only remaining rule, even if it has a very low score. So in this case there is a bias toward the class that has the most rules, even if they are of poor quality. This is

```

// R is the set of rules found
// T is the set of training instances (transactions)
SR = ruleSelectionByGroups(R, minGroups)
  sort R in descending order by the rule's score (r.score)
  SR =  $\emptyset$  // the selected rules
  for each t  $\in$  T
    prevScore =  $\infty$ , groups = 0
    for each r  $\in$  R and while groups < minGroups
      if (r.X  $\subseteq$  t.X  $\wedge$  r.y == t.y)
        // the rule applies to and correctly classifies t
        SR = SR  $\cup$  r
        if (r.score < prevScore)
          groups ++
          prevScore = r.score
  return SR

```

(a) Rule Selection Algorithm.

```

// t is an instance to classify
// SR is the set of selected rules.
c = classifyByGroups(t)
  M = {r|r.X  $\subseteq$  t.X  $\wedge$  r  $\in$  SR} // the matching rules.
  C = {r.y|r  $\in$  M} // classes predicted by matching rules.
  min = minc |{r.y == c  $\wedge$  r  $\in$  M}|
  //the minimum number of matching rules for a class.
  min = min · |C| // the number of rules we can use without
  // running out of rules for any class
  keep the first min rules in M when sorted in descending
  order by r.score and delete the rest
  group the rules in M by equal score
  counts[|C|] = [0, ..., 0]
  for each group g, from highest to lowest score
    for each c  $\in$  C
      counts[c] += |{r|r  $\in$  g  $\wedge$  r.y == c}|
      // the number of rules in g predicting c
    max = maxc  $\in$  C {counts[c]}
    for each c  $\in$  C
      if (counts[c] < max) // not a majority.
        C = C - c
    if (|C| == 1) // have one standout majority class
      return the only c  $\in$  C
  return a randomly chosen c  $\in$  C.

```

(b) Classification Algorithm.

## Figure 4. Algorithms

not 'fair' to the class for which fewer rules were found (for whatever reason – for example, this could happen if it was the minority class). Indeed, we have found that making a prediction based solely on the class that has the majority of rules (i.e.: ignoring the score) can have poor performance. So in this case, we make the random choice. In practice, when testing this on a few datasets, the random choice was never exercised.

Note that the rule selection pruning algorithm ensures that there are at most *minGroups* groups of rules with the

same score for each training instance. Therefore, we can expect to have up to *minGroups* groups to base a decision on when classifying. *minGroups* can be set quite low, since usually the top group is enough. We choose *minGroups* = 3.

## 6 Experiments

We performed experiments<sup>3</sup> on both relatively balanced datasets as well as imbalanced variations of them. The datasets are well known UCI datasets [10], with continuous variables discretised using the technique of [8]. We used stratified 10-fold cross validation for measuring all performance indicators. The methods we describe in this paper are denoted by “SPARCCC”, with the search strategy being used in parentheses. For comparison we also use a purely support and confidence based technique denoted by “Support-Confidence”. It finds all rules satisfying the support and confidence thresholds, and uses confidence as the strength score<sup>4</sup>.

### 6.1 Original (Balanced) Datasets

Note that in Figure 5(a), the average accuracy of SPARCCC is relatively insensitive to the significance level and compares favorably to CBA, CMAR and C4.5<sup>5</sup>. Note that the choice of *Strength Score* makes almost no difference (on average) to the accuracy on these datasets. However, sometimes it may be wise to experiment, as the results on the “Horse” and “Diabetes” datasets show. There is also little difference on average between the different search strategies considered in terms of classification performance. Finally, there is a small trend in significance level – favoring the use of more significant rules.

However, there are large differences in the search space examined and hence the run times, as shown in Figures 6(a) and 6(b). Our methods have a much smaller search space and hence computational complexity than the other AC methods. Despite having very similar accuracy, the search space explored by “Aggressive-S” is {1.6%, 1.4%, 1.3%} (for significances of {0.05, 0.01, 0.001} respectively) of that explored using a support based technique with *minSup* = 1%<sup>6</sup>. If we use the less aggressive search, “Simple-S”, it is {18.9%, 10.0%, 6%}. These are quite dramatic savings.

<sup>3</sup>performed on a laptop with: Intel Pentium M 2.0GHz, 1GB of RAM, Windows XP Pro. Programs written in Java.

<sup>4</sup>That is, there is no use of significance tests or correlation at all. The rule selection and classification procedure is as described in this paper

<sup>5</sup>The reported accuracy levels for C4.5, CBA and CMAR were obtained from [7]

<sup>6</sup>We should note that *minSup* = 1% is usually recommended – *minSup* = 5% performs worse, and as we shall see, is terrible on skewed datasets.

Algorithm	Strength Score	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	SS <sub>p,conf,CCR</sub>	na	na	0.05	84.1	95.6	81.1	77.0	81.1	73.0	<b>82.0</b>
		na	na	0.01	84.5	96.1	81.5	77.3	80.7	78.4	<b>83.1</b>
		na	na	0.001	84.2	96.1	82.8	78.0	82.2	78.4	<b>83.6</b>
	SS <sub>p,conf</sub>	na	na	0.05	85.8	94.3	82.1	75.0	81.9	75.4	<b>82.4</b>
		na	na	0.01	86.4	95.1	82.1	73.7	80.0	80.3	<b>82.9</b>
		na	na	0.001	85.9	95.3	83.4	71.5	82.6	80.1	<b>83.1</b>
	SS <sub>p,CCR</sub>	na	na	0.05	84.1	95.4	81.8	77.1	81.1	72.4	<b>82.0</b>
		na	na	0.01	84.2	96.0	81.8	76.3	81.1	78.4	<b>83.0</b>
		na	na	0.001	84.2	96.0	83.1	76.3	82.2	78.4	<b>83.4</b>
SPARCCC (Simple-S)	SS <sub>p,conf,CCR</sub>	na	na	0.05	83.3	95.7	82.8	77.6	83.0	75.7	<b>83.0</b>
		na	na	0.01	83.5	95.9	82.1	77.6	83.0	75.7	<b>82.9</b>
		na	na	0.001	85.1	95.0	82.8	77.1	83.0	74.6	<b>82.9</b>
	SS <sub>p,CCR</sub>	na	na	0.05	83.6	95.7	82.8	78.0	83.0	75.4	<b>83.1</b>
		na	na	0.01	83.2	95.9	82.1	78.0	83.0	75.4	<b>82.9</b>
		na	na	0.001	85.1	95.1	82.8	77.9	83.0	74.3	<b>83.0</b>
SPARCCC (Support)	SS <sub>p,conf,CCR</sub>	1%	na	0.05	84.2	95.7	82.1	76.7	81.9	78.4	<b>83.2</b>
		5%	na	0.05	84.6	93.3	81.8	77.5	83.0	79.0	<b>83.2</b>
	SS <sub>p,CCR</sub>	1%	na	0.05	84.3	95.7	81.8	76.6	82.2	77.0	<b>82.9</b>
		5%	na	0.05	85.5	92.7	82.1	73.0	83.3	80.1	<b>82.8</b>
Support-Confidence	conf	1%	0.5	na	85.4	95.6	82.8	76.7	83.3	80.3	<b>84.0</b>
		5%	0.5	na	85.5	92.7	81.8	73.0	83.0	80.1	<b>82.7</b>
CBA	na	1%	0.5	na	84.9	96.3	82.8	74.5	81.9	82.1	<b>83.8</b>
CMAR	na	1%	0.5	na	86.1	96.4	82.2	75.8	82.2	82.6	<b>84.2</b>
C4.5	na	na	na	na	84.7	95.0	78.2	74.2	80.8	82.6	<b>82.6</b>

(a) Accuracy on Original Datasets.

Algorithm	Strength Score	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	SS <sub>p,conf,CCR</sub>	na	na	0.05	53.5	88.2	31.6	19.6	23.5	30.8	<b>41.2</b>
		na	na	0.01	48.8	90.2	5.3	19.6	11.8	19.2	<b>32.5</b>
		na	na	0.001	32.6	82.4	0.0	5.4	0.0	19.2	<b>23.2</b>
	SS <sub>p,conf</sub>	na	na	0.05	2.3	68.6	0.0	0.0	0.0	0.0	<b>11.8</b>
		na	na	0.01	2.3	70.6	0.0	0.0	0.0	0.0	<b>12.2</b>
		na	na	0.001	0.0	56.9	0.0	0.0	0.0	0.0	<b>9.5</b>
	SS <sub>p,CCR</sub>	na	na	0.05	69.8	88.2	42.1	46.4	47.1	30.8	<b>54.1</b>
		na	na	0.01	74.4	90.2	31.6	44.6	17.6	42.3	<b>50.1</b>
		na	na	0.001	58.1	86.3	26.3	39.3	5.9	46.2	<b>43.7</b>
SPARCCC (Simple-S)	SS <sub>p,conf,CCR</sub>	na	na	0.05	41.9	74.5	42.1	32.1	41.2	26.9	<b>43.1</b>
		na	na	0.01	39.5	78.4	31.6	33.9	23.5	26.9	<b>39.0</b>
		na	na	0.001	39.5	76.5	5.3	17.9	0.0	19.2	<b>26.4</b>
	SS <sub>p,CCR</sub>	na	na	0.05	41.9	74.5	42.1	55.4	41.2	26.9	<b>47.0</b>
		na	na	0.01	39.5	80.4	31.6	55.4	35.3	26.9	<b>44.8</b>
		na	na	0.001	39.5	80.4	21.1	39.3	11.8	34.6	<b>37.8</b>
SPARCCC (Support)	SS <sub>p,conf,CCR</sub>	1%	na	0.05	58.1	70.6	31.6	23.2	29.4	42.3	<b>42.5</b>
		5%	na	0.05	23.3	0.0	0.0	0.0	0.0	7.7	<b>5.2</b>
	SS <sub>p,conf</sub>	1%	na	0.05	58.1	52.9	21.1	0.0	23.5	42.3	<b>33.0</b>
		5%	na	0.05	23.3	0.0	0.0	0.0	0.0	3.8	<b>4.5</b>
	SS <sub>p,CCR</sub>	1%	na	0.05	58.1	70.6	31.6	42.9	29.4	42.3	<b>45.8</b>
		5%	na	0.05	25.6	17.6	5.3	0.0	11.8	7.7	<b>11.3</b>
Support-Confidence	conf	1%	0.5	na	14.0	37.3	5.3	0.0	11.8	0.0	<b>11.4</b>
		5%	0.5	na	0.0	0.0	0.0	0.0	0.0	0.0	<b>0.0</b>
CBA	na	1%	0.5	na	79.3	59.2	18.5	36.3	0.0	29.0	<b>37.1</b>
CCCS	na	na	na	na	64.2	74.3	27.8	30.9	18.7	41.6	<b>42.9</b>

(b) True Positive Rate (Recall, Sensitivity) of the Minority Class on Imbalanced Versions of the Datasets.

Figure 5. Classification Performance on Original and Imbalanced Versions of the Datasets.

Algorithm	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	na	na	0.05	2,840	4,805	1,251	874	3,361	29,023	7,026
	na	na	0.01	2,089	4,102	929	724	1,921	27,831	6,266
	na	na	0.001	1,525	3,404	661	519	1,436	25,894	5,573
SPARCCC (Simple-S)	na	na	0.05	168,762	55,149	16,520	3,136	47,105	210,735	83,568
	na	na	0.01	102,804	35,854	14,047	2,680	47,105	61,749	44,040
	na	na	0.001	55,218	23,532	11,376	2,136	47,057	19,347	26,444
any Support method	1%	any	any	501,939	10,527	71,038	8,363	321,577	1,733,455	441,150
	5%	any	any	38,095	2,202	11,460	2,454	72,060	19,152	24,237

(a) Search Space Size on the Original Datasets.

Algorithm	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	na	na	0.05	0.184	0.125	0.067	0.073	0.062	0.106	0.103
	na	na	0.01	0.135	0.112	0.050	0.057	0.056	0.070	0.080
	na	na	0.001	0.115	0.100	0.043	0.057	0.045	0.060	0.070
SPARCCC (Simple-S)	na	na	0.05	17,024	1,281	0.917	0.557	3.909	13,556	6.207
	na	na	0.01	10,239	0.954	0.756	0.484	3.911	4.732	3.513
	na	na	0.001	4,587	0.740	0.575	0.410	3.870	1.474	1.943
any Support method	1%	any	any	116.873	3.850	6.795	1.190	30.917	353.708	85.556
	5%	any	any	3.284	0.822	0.700	0.379	5.198	1.076	1.910

(b) Training Time on the Original Datasets.

Algorithm	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	na	na	0.05	172	128	128	48	113	101	115
	na	na	0.01	107	105	85	39	85	50	79
	na	na	0.001	66	85	56	29	59	35	55
SPARCCC (Simple-S)	na	na	0.05	39,653	5,375	5,028	1,104	16,384	36,059	17,267
	na	na	0.01	21,061	3,669	4,181	908	16,384	9,250	9,242
	na	na	0.001	8,455	2,461	3,144	710	16,336	2,723	5,638
SPARCCC (Support)	1%	na	0.05	182,996	5,094	19,926	2,762	69,187	344,860	104,138
	5%	na	0.05	15,708	1,090	4,760	896	29,326	6,339	9,687
any Support- Confidence	1%	0.5	na	218,637	5,175	31,091	3,415	137,545	772,506	194,728
	5%	0.5	na	15,953	1,091	4,932	965	31,164	7,065	10,195

(c) Number of Rules Found (Prior to Rule Selection) on the Original Datasets.

Algorithm	minSup	minConf	significance	Australia	Breast	Cleve	Diabetes	Heart	Horse	Average
SPARCCC (Aggressive-S)	na	na	0.05	1,877	3,272	424	238	420	2,341	1,429
	na	na	0.01	1,318	2,762	309	194	293	1,459	1,056
	na	na	0.001	1,072	1,951	246	170	232	736	735
SPARCCC (Simple-S)	na	na	0.05	89,496	31,416	8,865	3,336	25,815	24,771	30,617
	na	na	0.01	54,738	23,382	5,213	2,705	22,617	5,043	18,950
	na	na	0.001	31,670	15,125	2,986	1,813	17,889	1,289	11,795
any Support	1%	any	any / na	501,132	10,701	93,387	7,314	270,694	2,080,527	493,959
	5%	any	any / na	50,843	3,152	13,355	2,534	67,169	39,039	29,349

(d) Search Space Size on Imbalanced Versions of the Datasets.

**Figure 6. Computational Performance and Rules Found (Averaged over Folds).**

Even more pronounced results are shown in the run times. For example, “Aggressive-S” takes at most 0.12% of the time, and “Simple-S” takes at most 7.3% of the time of using  $minSup = 1\%$ . These are dramatic savings and suggest that these techniques are finding the best rules much more efficiently. Finally, much fewer rules are found, as can be seen in Figure 6(c).

Note that the rules found by the “Support-Confidence” algorithm are those that pass the  $minSup$  and  $minConf$  thresholds. This is equal to the rules CMAR finds without

the  $\chi^2$  test, and the rules CBA finds without error based pruning. Therefore, the search space and runtime are directly comparable (actually, they are faster than CBA and CMAR due to a faster underlying algorithm).

So picking the best accuracy (83.6%, “Aggressive-S” using significance of 0.001 and  $SS_{p,conf,CCR}$ ) we can obtain comparable accuracy while searching only 1.3% of the space, using 0.08% of the time and finding 0.03% of the rules, when compared to support based methods – for example; CBA and CMAR.

## 6.2 Imbalanced Datasets

Highly imbalanced versions of the datasets were obtained by keeping the majority class as is while randomly selecting a subset of the minority class so that the percentage of instances with the minority class was 10%.

Figure 5(b) shows the *True Positive Rate (TPR)* of the minority class. Accuracy is a poor performance measure for imbalanced datasets because one can obtain high accuracy by predicting the majority class. TPR (also known as *sensitivity* and *recall*) is a much better performance indicator. The accuracy is therefore not shown. It remains high however – and the accuracy on the original datasets also implies this.

The effect of using *CCR* in the *Strength Score* is dramatic. One can clearly see the following relationship:

$$TPR(SS_{p,CCR}) \gg TPR(SS_{p,conf,CCR}) \gg TPR(SS_{p,conf})$$

For example, when using “Aggressive-S”,  $SS_{p,conf,CCR}$  is on average (over datasets and significance levels) 2.87 times better than  $SS_{p,conf}$  and  $SS_{p,CCR}$  is 1.58 times better than  $SS_{p,conf,CCR}$  and 4.44 times better than  $SS_{p,conf}$ ! This is a very significant improvement. A similar, though slightly smaller effect occurs for “Simple-S” and “Support-S”.

The improvement of our methods over other rule based techniques such as CBA is dramatic. We also get higher results than CCCS [3] which was designed specifically for imbalanced datasets. The highest average TPR overall is for “Aggressive-S” with a significance level of 0.05. This was 45.8% better than CBA and 26.1% better than CCCS.

Unlike for the original datasets, the significance level has a large impact on the classification performance on imbalanced datasets, likely due to the pruning of the search space. Interestingly, the use of *CCR* had the unexpected benefit of reducing this effect. We also found that much fewer rules were generated overall.

Finally, the computational performance favors our techniques even more on imbalanced datasets. Figure 6(d) for example shows that “Aggressive-S”, at a significance level of 0.05, explores only 0.29% of the space considered by a support based method with  $minSup = 1\%$  – and the training time is even less. For “Simple-S” it is 6.2%. Note also that the performance of any support based technique with  $minSup = 5\%$  is terrible, as is to be expected.

*So overall, using SPARCCC with a significance based search strategy we achieve much better classification performance on skewed datasets than techniques such as CBA (we outperform it by up to 45.8% when using a significance level of 0.05) while using dramatically fewer computational resources (0.29% of those used by support based methods).*

## 7 Related Work

CBA [8] was the first Associative Classifier (AC) proposed and almost all other ACs are variations on the original CBA design. For *rule mining*, CBA mines all rules passing support and confidence thresholds ( $minSup$  and  $minConf$ ). Additionally, it ignores rules based on a “pessimistic error based pruning method” borrowed from C4.5 [11]. Unfortunately this still generates thousands of rules – most of which perform poorly. Therefore, a *rule selection* process is needed to select a small subset likely to perform well. New instances are *classified* according to the highest ranked rule that is applicable. Rules are ranked according to confidence, support, and size.

CMAR [7] has many similarities to CBA. The main differences<sup>7</sup> are the use of a  $\chi^2$  test instead of the error based pruning and a more complicated *classification* procedure involving an empirically chosen *weighted*  $\chi^2$  measure applied to *multiple* matching rules. CMAR uses the same contingency table (Figure 1) as we do for evaluating one of our interestingness criteria. However, the  $\chi^2$  test does not distinguish between directions of association and therefore the claim in [7] that only positively correlated rules are found is incorrect. Negatively associated rules are just as likely to pass the test as positively associated ones. Even though it checks for significance, it is still based on the *support-confidence* framework.

In general, rules with  $CCR(\cdot) < 1$  will incorrectly classify the training data. These techniques still work because, in balanced datasets, choosing high support and confidence rules tends to favor positively correlated rules, but this is *not* the case in imbalanced datasets as Lemma 1 shows. It is not surprising then, that techniques using the *support-confidence* framework perform poorly on imbalanced datasets.

The CCCS [3] technique was proposed to find positively correlated rules. It takes into account imbalanced class distributions, enabling it to outperform other techniques on imbalanced datasets. It forces *correlation* to be locally monotonic and uses a *top down row enumeration* algorithm. However, there is no guarantee that the rules found are statistically significant, and this algorithm generates many thousands of rules. It is also very computationally intensive and does not scale well for traditional datasets where there are more instances than attributes.

Morishita et al. [9] use the same test as CMAR but find upper-bounds on  $\chi^2$  for search space pruning. It is an ARM technique and is not used for classification.

Webb [16] recently proposed the use of Fisher’s Exact

<sup>7</sup>We note that CBA is based on the Apriori algorithm, while CMAR is based on FP-Growth. Such differences do not change the rules that are found, just the way in which they are found, and hence are irrelevant for this discussion.

Test (FET) to examine the significance of association rules in more detail than [7, 9]. The technique is almost identical to the *Aggressive-S* search strategy we employ for pruning the search space. Webb does *not* use the rules for classification – instead it is used for knowledge discovery. This requires consideration of the issue of multiple tests. Since the mined rules are not validated, it is very difficult to determine whether the rules are in fact useful. In this paper we mine significant rules under a number of *different* strategies and use them for classification – which also requires additional work such as rule selection, ranking and classification. We therefore have a very good performance indicator – performance on unseen data in comparison to other algorithms.

## Acknowledgments

We are grateful to Bavani Arunasalam for providing the preprocessed data and the CCCS results. This research was partially funded by the Australian Research Council (ARC) Discovery Grant, Project ID: DP055900.

## References

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases VLDB*, pages 487–499. Morgan Kaufmann, 1994.
- [2] M.-L. Antonie and O. R. Zaiane. An associative classifier based on positive and negative rules. In *9th ACM SIGMOD workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD-04)*, pages 64–69, 2004.
- [3] B. Arunasalam and S. Chawla. Cccs: a top-down associative classifier for imbalanced class distribution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, New York, NY, USA, 2006. ACM Press.
- [4] G. Cong, A. K.H.Tung, X. Xu, F. Pan, and J. Yang. Farmer: Finding interesting rule groups in microarray datasets. In *23rd ACM SIGMOD International Conference on Management of Data Proceedings*, pages 145–154, 2004.
- [5] G. Cong, K.-L. Tan, A. K.H.Tung, and X. Xu. Mining top-k covering rule groups for gene expression data. In *ACM SIGMOD/PODS 2005 Proceedings*, pages 670–681, 2005.
- [6] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12. ACM Press, May 2000.
- [7] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 369–376, Washington, DC, USA, 2001. IEEE Computer Society.
- [8] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80–86, 1998.
- [9] S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In *Symposium on Principles of Database Systems*, pages 226–236, 2000.
- [10] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Machine-readable data repository, University of California, Department of Information and Computer Science, Irvine, CA, 1992.
- [11] R. Quinlan. *C4.5: Program for Machine Learning*. Morgan Kaufmann.
- [12] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.
- [13] A. Veloso, W. M. Jr., and M. J. Zaki. Lazy associative classification. In *IEEE ICDM*, volume 0, pages 645–654, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
- [14] F. Verhein and S. Chawla. Geometrically inspired itemset mining. In *2006 International Conference on Data Mining (ICDM'06)*, pages 655–666. IEEE Computer Society, 2006.
- [15] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *2005 SIAM International Conference on Data Mining (SDM'05) Proceedings*, 2005.
- [16] G. I. Webb. Discovering significant rules. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 434–443, New York, NY, USA, 2006. ACM Press.
- [17] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In D. Barbará and C. Kamath, editors, *SDM*. SIAM, 2003.

School of Information Technologies, J12  
The University of Sydney  
NSW 2006 AUSTRALIA  
T +61 2 9351 3423  
F +61 2 9351 3838  
[www.it.usyd.edu.au](http://www.it.usyd.edu.au)

ISBN 978 1 86487 923 0