CCCS: A Top-down Associative Classifier for Imbalanced Class Distribution

Bavani Arunasalam * and Sanjay Chawla
School of Information Technologies, University of Sydney
NSW, Australia
bavani@it.usyd.edu.au, chawla@it.usyd.edu.au

ABSTRACT

In this paper we propose CCCS, a new algorithm for classification based on association rule mining. The key innovation in CCCS is the use of a new measure, the "Complement Class Support (CCS)" whose application results in rules which are guaranteed to be positively correlated. Furthermore, the anti-monotonic property that CCS possesses has very different semantics *vis-a-vis* the traditional support measure. In particular, "good" rules have a low CCS value. This makes CCS an ideal measure to use in conjunction with a top-down algorithm. Finally, the nature of CCS allows the pruning of rules without the setting of any threshold parameter! To the best of our knowledge this is the first threshold-free algorithm in association rule mining for classification.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

algorithms,experimentation

Keywords

Association Rules Mining, Classification, Imbalanced Data sets, Parameter-free mining

1. INTRODUCTION

Classification is a core data mining and machine learning task. The objective in classification is to build a model (the classifier) which maps objects into pre-defined classes based on the attributes of objects. The methodology of evaluating a classifier consists of partitioning a data set into two subsets called the training and test data. The model is built on the training data and its accuracy is determined by measuring its ability to correctly classify the objects in the test data. Interested readers should consult a recent textbook on data mining (for example, [14]) for a survey of classification techniques.

In 1998, Liu, Hsu and Ma [13] introduced CBA, a new classification algorithm based on association rule mining, a foundational paradigm in data mining [1]. CBA mines association rules of the form $A \rightarrow C$, where A is a subset of the attribute set and C is an element of the set of pre-defined classes. The mining is performed on the training data. The set of mined rules are then used to classify instances in the test data. Several variations on the original CBA algorithm have emerged in the research literature [13, 15, 10] with the main challenges being:

1. To select an optimal subset of rules mined to build a classifier, and

2. Selecting an appropriate rule in order to classify new instances.

While several CBA-type algorithms are available (collectively called "Associative Classifiers") all of them use the traditional support measure to mine association rules. However using a minimum support threshold is inappropriate for data sets with highly imbalanced class distribution. For example in direct marketing applications, the response rate, the people who actually buy the product after responding to a promotion, is typically $1\% - 2\%$ [11]. Similarly in medical databases the relative size of the samples of people who suffer from a disease is extremely small. In such applications setting the minimum support to even 1% is likely to prune most of the cases of interest. On the other hand using a low minimum support results in the generation of extremely large amount of rules.

While *support* is often used as a measure of significance of the rule (rules with high support intuitively capture an underlying process rather than being artifacts of the particular data set), the strength of the rule is captured by its *confidence*. However, even confidence can often be misleading measure in the case of imbalanced distribution. Consider the example, shown in Table.1 which shows the association between an item A and class C_1. A and $\bar{A}$ represent the presence and absence of item A respectively , C_1 represents a class and $\bar{C_1}$ represents the complement of C_1.

Now consider the association rule $A \Rightarrow C_1$. The confidence of this rule is given by $conf(A \Rightarrow C_1) = \sigma(A \cup C_1)/\sigma(A) = 20/25 = 80\%$ and the support of the rule is

*The work of this author was supported by The Capital Markets CRC.

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.

	C_1	$\bar{C_1}$	Total
A	20	5	25
$\bar{A}$	70	5	75
Total	90	10	100

Table 1: Example

$\sigma(A \cup C_1)/N$ =20%. Hence this rule has high confidence and support.

Now lets calculate the correlation between A and C_1. As shown in [5], one way of calculating the correlation is to compute $P(A \cup C_1)/P(A) \times P(C_1) = 0.2/(0.25 \times 0.90) = 0.89$. The fact that this quantity is less than 1 indicates negative correlation between A and C_1. Such situations occur when the class distribution is imbalanced as in the example where class C_1 makes up 90% of the data.

We have designed a new measure, which we call the Complement Class Support (CCS), which tightly captures the relationship between the antecedent of the rule and its class. Given a rule $A \rightarrow C$,

$$CCS(A \rightarrow C) = \frac{\sigma(A \cup \bar{C})}{\sigma(\bar{C})}$$

Intuitively, CCS captures the strength of the rule in the complement class. If a rule is *strong* then its CCS should be small. In fact in Section 3, we will prove the following properties which explain why CCS can be used to design an efficient and accurate classifier for imbalanced data sets.

LEMMA 1. *For a rule $A \rightarrow C$, the following are equivalent*

a) A and C are positively correlated.

b) $CCS(A \rightarrow C) < \frac{\sigma(A)}{N}$.

c) $conf(A \rightarrow C) > \frac{\sigma(C)}{N}$.

d) $conf(A \rightarrow C) > 1 - \frac{\sigma(\bar{C})}{N}$.

Lemma 1.(b) asserts that for a rule to be positively correlated it is necessary and sufficient that the CCS of the rule be bounded by the support of the antecedent. This explains why the rule $A \rightarrow C_1$ in Table 1 is not positively correlated even though it has a high confidence: $CCS(A \rightarrow C_1) = 5/10 = 0.5 > \frac{\sigma(A)}{N}$.

Lemma 1.(c) shows that when the classes are equally distributed, by setting the *minconf* to 0.5, we can make sure that all the rules discovered are positively correlated. However, when the class distribution is imbalanced, the *minconf* has to be set to the support of the majority class in order for the rules discovered to be positively correlated rules. This is likely to prune most of the rules from the minority class and hence affect the accuracy of classification.

Assuming that we are dealing with a binary classification, Lemma 1.(d) asserts that the *minconf* for each class should be set in terms of the support of the other class to get positively correlated rules. If the class distributions are equal the *minconf* for each class would be 0.5 and this would be the same as using confidence alone. However, if the class distributions are imbalanced such as 90% and 10%, for the majority class the *minconf*=1-0.1=0.9 and for the minority class, *minconf*=1-0.9=0.1. By setting a lower value for the minority class we avoid pruning its rules and by setting a higher value for the majority class we avoid generating large number of rules.

The above discussion leads us to conclude that CCS is a better measure for imbalanced class distribution. It automatically selects the right confidence level for each class in order to guarantee that the rules discovered are positively correlated. Finally, it should be noted that CCS does not have a prescribed a pre-defined threshold value. At each node in the lattice, the CCS value will dynamically adjust (based on the support of the antecedent at that node) in order to generate only positively correlated rules.

In many cases, for reasons of efficiency, we may want to set a threshold value for CCS. In such situations we can take advantage of the fact that CCS enjoys the anti-monotonic property.

LEMMA 2. *Given a rule $A_i \Rightarrow C_j$, if $CCS(A_i \Rightarrow C_j) > t$ then, $CCS(subsetOf(A_i) \Rightarrow C_j) > t$.*

It is important to note that the anti-monotonicity of CCS has different semantics *vis-a-vis* the traditional *support* measure. In particular, "good" rules have low CCS value. As we will show later, we can combine CCS with top-down row enumeration algorithms to efficiently discover class rules. Even though row enumeration algorithms [6, 7, 8] use the support measure for pruning they cannot exploit the anti-monotonic property as they descend down the tree. Instead at each node, an upper bound on the maximum support value of all nodes rooted at that node is derived. The subtree is pruned if the maximum is below the support threshold. For CCS, no such bound is required. If a node has a high CCS value, then it (and possibly its descendants) can be pruned.

2. RELATED WORK

Associative classifier is a classification model that has shown a lot of promise recently. The first associative classifier, CBA, was proposed in 1998 [13]. Since then many algorithms have been developed to improve the efficiency and accuracy of classifications. These methods differ in three aspects as follows:

1. The technique used to mine the association rules efficiently
2. The measures and method used to select the best set of rules for the classifier.
3. The measures used to effectively predict the class of a new instance.

In CBA and Harmony [15], the rule selection is purely based on confidence. However this method may not always give the correct classification especially when the class distribution is highly imbalanced. As discussed in Section 1, in imbalanced data sets, rules with high confidence could be actually negatively correlated.

CMAR [10] overcomes this problem by selecting rules for the classifier only if they pass the χ^2 test and ARC-PAN [2] by calculating the correlation coefficient. We will show in Section 3 that by using Complement Class Support for pruning, we only generate positively correlated rules and this

removes the need for further testing. Refer to our technical report [3] for a detailed discussion on related work.

Recently the *row enumeration* method used in algorithms such as Farmer, RERII and RCBT [6, 7, 8] has proved to be very efficient in mining association rules from databases with extremely large number of attributes such as the micro array data sets. We adopt this method in our algorithm for reasons discussed in [3].

The remainder of the paper is arranged as follows: In Section 3 we present the basic definitions and notations of the concepts used in this paper and also present the theorems.In Section 4 we present our classification algorithm CCCS with an example. Finally in section 5 we present the experiments and results and Section 6 concludes the paper with a summary of the research.

3. BASIC DEFINITIONS AND NOTATION

Let I be a finite set of items and C a finite set of class labels. A row (or instance) is an element of the set $(2^I \times C)$. D is the set of all rows given. Assume $|D| = N$. A class rule is an implication of the form $A \rightarrow C_1$ where $A \subset I$ and $C_1 \in C$. Traditionally, the strength of of a class rule is defined in terms of *support* and *confidence* (Refer to [1] for definitions).

Definition 1. Given a rule $A \rightarrow C$, we measure the correlation between A and C based on the magnitude of the ratio $\frac{P(A \cup C)}{P(A)P(C)}$. In particular if

$$\frac{P(A \cup C)}{P(A)P(C)} \quad \begin{cases} > 1 & \text{then } A \text{ and } C \text{ are positively correlated} \\ < 1 & \text{then } A \text{ and } C \text{ are negatively correlated} \end{cases}$$

Definition 2. The Class Support of rule $A_i \Rightarrow C_j$ is the support of A_i in class C_j.

$$ClSup(A_i \Rightarrow C_j) = \frac{\sigma(A_i \cup C_j)}{\sigma(C_j)}$$

Definition 3. The Complement Class Support (CCS) of a rule $A_i \Rightarrow C_j$ is the support of A_i in the classes other than C_j. Let $\overline{C_j}$ denote all the classes in D other than C_j. Then

$$CCS(A_i \Rightarrow C_j) = \frac{\sigma(A_i \cup \overline{C_j})}{\sigma(\overline{C_j})}$$

Definition 4. The Strength Score of a rule $A_i \Rightarrow C_j$ is given by

$$SS(A_i \Rightarrow C_j) = \frac{Conf(A_i \Rightarrow C_i) \times ClSup(A_i \Rightarrow C_i)}{max(CCS(A_i \Rightarrow C_j), t)}$$

where t is set to a very low value such as 0.001 to avoid division by 0 when CCS =0.

We now show that CCS is anti-monotonic.

LEMMA 3. *Given a rule $A_i \Rightarrow C_j$, if $CCS(A_i \Rightarrow C_j) > t$ then, $CCS(subsetOf(A_i) \Rightarrow C_j) > t$.*

PROOF. Given

$$CCS(A_i \Rightarrow C_j) > t$$

$$\frac{\sigma(A_i \cup \overline{C_j})}{\sigma(\overline{C_j})} > t$$

But

$$\sigma(subsetOf(A_i) \cup \overline{C_j}) > \sigma(A_i \cup \overline{C_j})$$

/* anti-monotone property of support*/

Therefore

$$\frac{\sigma(subsetOf(A_i) \cup \overline{C_j})}{\sigma(\overline{C_j})} > t$$

$$CCS(subsetOf(A_i) \Rightarrow C_j) > t$$

□

We now prove Theorem 1, the main theoretical result of this paper which underpins the design of our algorithm CCCS. We restate the lemma for convenience.

THEOREM 1. *For a class rule $A \rightarrow C$, the following are equivalent*

a) A and C are positively correlated.

b) $CCS(A \rightarrow C) < \frac{\sigma(A)}{N}$.

c) $conf(A \rightarrow C) > \frac{\sigma(C)}{N}$.

d) $conf(A \rightarrow C) > 1 - \frac{\sigma(\bar{C})}{N}$.

PROOF. $(a) \rightarrow (b)$

Given that A and C are positively correlated

$$\Rightarrow \frac{P(A \cup C)}{P(A)P(C)} > 1$$

$$\Rightarrow \frac{\frac{\sigma(A \cup C)}{N}}{\frac{\sigma(A)}{N}\frac{\sigma(C)}{N}} > 1$$

$$\Rightarrow \frac{N\sigma(A \cup C)}{\sigma(A)\sigma(C)} > 1$$

$$\Rightarrow N\sigma(A \cup C) > \sigma(A)\sigma(C)$$

$$\Rightarrow -N\sigma(A \cup C) < -\sigma(A)\sigma(C)$$

$$\Rightarrow N\sigma(A) - N\sigma(A \cup C) < N\sigma(A) - \sigma(A)\sigma(C)$$

$$\Rightarrow N[\sigma(A) - \sigma(A \cup C)] < \sigma(A)[N - \sigma(C)]$$

$$\Rightarrow N\sigma(A \cup \overline{C}) < \sigma(A)\sigma(\overline{C})$$

$$\Rightarrow \frac{\sigma(A \cup \overline{C})}{\sigma(\overline{C})^1} < \frac{\sigma(A)}{N}$$

$$CCS(A \Rightarrow C) < \frac{\sigma(A)}{N}$$

$(b) \Rightarrow (c)$

Given

$$CCS(A \Rightarrow C) < \frac{\sigma(A)}{N}$$

$$\Rightarrow \frac{\sigma(A \cup \overline{C})}{\sigma(\overline{C})} < \frac{\sigma(A)}{N}$$

$$\Rightarrow N\sigma(A \cup \overline{C}) < \sigma(A)\sigma(\overline{C})$$

$$\Rightarrow N[\sigma(A) - \sigma(A \cup C)] < \sigma(A)[N - \sigma(C)]$$

$$\Rightarrow N\sigma(A) - N\sigma(A \cup C) < N\sigma(A) - \sigma(A)\sigma(C)$$

$$\Rightarrow N\sigma(A \cup C) > \sigma(A)\sigma(C)$$

$$\Rightarrow \frac{\sigma(A \cup C)}{\sigma(A)} > \frac{\sigma(C)}{N}$$

$(c) \Rightarrow (d)$
This is a simple rewrite of

$$\frac{\sigma(C)}{N} = 1 - \frac{\sigma(\bar{C})}{N}$$

$(d) \Rightarrow (a)$

Given

$$conf(A \rightarrow C) > 1 - \frac{\sigma(\overline{C})}{N}$$

$$\Rightarrow \frac{\sigma(A \cup C)}{\sigma(A)} > \frac{N - \sigma(\overline{C})}{N}$$

$$\Rightarrow \frac{\sigma(A \cup C)}{\sigma(A)} > \frac{\sigma(C)}{N}$$

$$\Rightarrow \frac{N\sigma(A \cup C)}{\sigma(A)\sigma(C)} > 1$$

$$\Rightarrow \frac{\frac{\sigma(A \cup C)}{N}}{\frac{\sigma(A)}{N}\frac{\sigma(C)}{N}} > 1$$

$$\Rightarrow \frac{P(A \cup C)}{P(A)P(C)} > 1$$

Therefore, A and C are positively correlated □

4. CCCS ALGORITHM

We now introduce the Classification using Complement Class Support (CCCS) Algorithm.

The definition of CCS suggests that rules with lower CCS values are likely to be *stronger* as they are less frequently seen in other classes. This property along with the anti-monotonic property of CCS makes it possible for it to be integrated with a *row enumeration* method. As the tree grows, the CCS of the rules increase and hence the rules become less desirable and are candidates for being pruned. Row-enumeration algorithms were designed for data sets where the number of rows is much smaller than the number of columns. For example, microarray data sets fall in this category. However they can be used for small to moderate size data sets (like those in the UCI Repository[4]) even if the data sets do not match these characteristics.

The CCCS Algorithm is given in Figure 1. Basically CCCS takes each transaction and generates rules such that the itemset and the class are positively correlated. Initially each transaction is added to the root of the enumeration tree. Then item sets are generated from a transaction by performing intersection with sibling nodes in the enumeration tree and these itemsets are added as child nodes of the transaction and the class of the transaction is passed on as the class of all its child nodes for the purpose of calculating the CCS. After intersection with all the siblings on the right of the enumeration tree, if a node is found to be positively correlated ($CCS < support$), that rule is considered as potential candidate for the class of the corresponding transaction.

The tree is grown in a depth-first fashion by recursively performing intersection of each node with its sibling node. The intersection at each node is performed in the same way as RERII [7]. From the candidate rules, our algorithm builds the classifier by selecting the best rule for each transaction similar to Harmony [15]. In our case the best rule is considered as the rule with the highest *Score Strength (SS)*.

Refer to our technical report [3] for a detailed explanation of the CCCS algorithm with an example.

Data set	# attributes	# rows	CBA	CCCS
Breast	10	699	4.2	3.1
Heart	14	270	18.5	18.1
Diabetes	9	768	25.3	23.9
Cleve	12	303	16.7	16.4
Pima	9	768	27.6	27.9
Mushroom	23	8124	0.0	1.31
Horse	19	368	18.7	19.0
Australian	15	368	13.4	14.4
Average			15.5	15.4

Table 2: Comparison of Error Rates in UCI Data sets

5. EXPERIMENTAL EVALUATION

In this section we report on the experiments that we have carried out to measure and assess the accuracy of CCCS. All our comparisons are with CBA. The executable of the CBA program was downloaded from [12]. Our objective is to experimentally validate the hypothesis:

For data sets with an imbalanced (skewed) class distribution, CCCS will be more accurate compared to CBA

5.1 Data Preparation

Eight data sets were obtained from the UCI ML Repository[4]. For a fair comparison the continuous variables were discretization using the same techniques as described in [13]. For each original data set, three versions of minority class size, 5%, 10% and 15% were created. Refer to our technical report [3] for the method we used to create the imbalanced datasets.

5.2 Results

A comparison of the error rates of the eight data sets is shown in Table 2. All the error rates reported are the average values of 10-fold cross validation. For CBA, the *minconf* was set to 50% and *minsup* was set to 1%.

While the error rates of CCCS is lower than CBA on the original data, a bootstrapping analysis (Section 5.3) shows that the differences are not significant.

Table 3 shows the results of CCCS and CBA on three separate imbalanced versions each of the eight data sets. For the 5% data sets, CCCS outperforms CBA except on the Diabetes data. For the 10% data sets, CCCS again outperforms CBA except on the Pima data. At 15%, the accuracy of CCCS and CBA begins to converge. This confirms our hypothesis that CCCS is more suitable than CBA for imbalanced data sets. **It should be noted that CCCS**

Input: Transaction table, t
Output: Classification Rules
1. Let D be the set of n transactions Tr_1,Tr_2,.....,Tr_n.
2. Let C be the set of class labels $c_1, c_2, c_3, ...c_m$ in D.
3. Let $Tr_i.items$ be the set of non-class attributes of Tr_i.
4. Let $Tr_i.class$ be the class label of Tr_i.
5. $Rules \leftarrow 0$
6. $PC \leftarrow 0$
7. Add the transactions to the root node
8. For each $Tr_i \epsilon D$
9. Let c_k be the class attribute of Tr_i.
10. Let N be the set of transactions $Tr_{i+1...n}$
11. For each node n in PC
12. if n.class = c_k
13. $N = N \cup n$
14. }
15. End For
16. MineRules[Tr_i, N, c_k,Rules]
17. End for

MineRules[Tr, N, c,Rules]
18. For each $n \epsilon N$
19. $N_i \leftarrow 0$ //set of intersections is set to 0//
20. $d = Tr.items \cap n.items$
21. if $|d| > 0$
22. if $Tr.items = n.items$
23. remove n from N
24. add n.id to Trn and $n'(n' \epsilon N)$
25. corresponding to n.class
26. }
27. if $Tr.items \subset n.items$
28. add n.id to Tr and $n'(n' \epsilon N)$
29. corresponding to n.class
30. }
31. if $Tr.items \supset n.items$
32. remove n from N
33. if d is not discovered before
34. add n' to N_i
35. $n'.tems = d$
36. add ids of Tr and n to n'
37. }
38. }
39. if $Tr.items \neq n.items$
40. if d is not discovered before
41. add n' to N_i
42. $n'.tems = d$
43. add ids of Tr and n to n'
44. }
45. }
46. }
47. End For
48. If $Tr.CCS < sup(Tr.itemlist)$
49. BuildClassifier[Rules,D,Tr,c]
50. if $|N_i| > 0$
51. For each $n_i' \epsilon N_i$
52. MineRules[n_i', N_i, c,Rules]
53. End For
54. }
55. else if $Tr.ClSup < sup(Tr.itemlist)$
56. $PC = PC \cup Tr$
57. }

Figure 1: Algorithm CCCS.

achieves a significantly higher percentage of *True Positives* (minority class) compared to CBA even when the *minsup* for CBA was 1%. In Table 3, TP% is the True Positive Rate. The True Positive Rate is a better measure of accuracy on data sets which are characterized by an imbalanced class distribution.

5.3 Bootstrap Analysis

To determine if the differences between the error rates of CCCS and CBA are real (as opposed to being artifacts of this particular instances of the data) we carried out a bootstrapping analysis. The advantage of bootstrapping (as opposed to a paired t-test), is that no distributional assumption about the error rates is required.

In bootstrapping, a resampling method is used to determine the confidence bounds of statistical estimators [9]. In our case we have two vectors, *cbaerror* and *cccserror* of length eight (the number of data sets). We want an estimate on the average of vector difference between them.

In order to create multiple samples of the vector difference, *sampling with replacement* is carried out and the average of the difference is computed for each sample. In sampling with replacement the data point sampled is returned to the data set and made available to be selected again. This way several samples can be created and "histogrammed". For our particular case if the zero point (0) lies in the bulk of the histogram then one can conclude that there is no significant difference between the two error vectors.

Figure 2 shows the bootstrapping results (on 1000 samples) on the four different versions of the data set. For the original data, the zero point lies in the bulk of the histogram and we can conclude that there is no significant difference between the CBA and CCCS error rate. For the 5% data, the zero point lies on the edge of the histogram which allows us to conclude that the differences between the CBA and CCCS errors are different. Similarly for the 10% data, the zero point is far removed from the bulk of the histogram. We again conclude that the differences between the CBA and CCCS errors are significant. Finally, for the the 15% data set, the zero point again returns to the middle of the histogram - the error differences are not significant.

Note that from Figure 2, it appears that CCCS does much better than CBA for the 10% data compared to the 5% data. This is because, for the 5% data, the chance to create errors is limited to 5% for the positive class.

6. CONCLUSION

In the last decade extensive research has been carried out in the mining of association rules. In 1998, Liu, Hsu and Ma [13] introduced the CBA algorithm for classification using association rules. Since then several variations on the original algorithm have been introduced. However, till date all algorithms have used the traditional support measure to mine association rules. We have shown that for imbalanced class data sets, the support/confidence framework is inadequate. In order to address this problem we have introduced a new measure, the Complement Class Support (CCS). CCS has several properties which make it extremely suitable for mining imbalanced data sets. The nature of CCS makes it an ideal candidate to be used in conjunction with a top-down row enumeration type algorithm. This is the essence of CCCS - a row enumeration algorithm with CCS guaranteed to generate positively correlated rules.

Data Set	# Instances		Error Rates		TP %	
	#Pos	#Neg	CBA	CCCS	CBA	CCCS
Breast						
5%	25	458	3.21	2.63	48.2	64.2
10%	51	458	5.41	4.47	59.2	74.3
15%	81	458	4.69	3.62	75.6	86.3
Heart						
5%	8	150	6.50	5.33	0 .0	26.6
10%	17	150	14.10	12.50	0.0	18.7
15%	26	150	12.50	12.94	47.0	50.9
diabetes						
5%	26	500	5.10	5.95	15.2	27.0
10%	56	500	10.50	8.91	36.3	30.9
15%	88	500	14.40	15.48	38.6	45.1
Cleve						
5%	9	165	7.20	4.71	0.0	24.0
10%	18	165	13.70	12.78	18.5	27.8
15%	29	165	16.10	15.79	20.66	38.5
Pima						
5%	26	500	5.10	5.00	0.0	10.0
10%	56	500	9.80	10.00	12.1	9.0
15%	88	500	14.40	14.53	22.5	27.1
Mushroom						
5%	221	4208	5.20	1.90	31.6	66.6
10%	447	4208	1.8	1.52	60.6	89.3
15%	742	4208	3.00	2.17	96.9	86.9
Horse						
5%	12	232	8.50	7.53	0.0	22.0
10%	26	232	11.20	10.42	29.0	41.6
15%	41	232	13.50	13.30	80.0	66.0
Australian						
5%	20	383	6.00	5.25	33.34	60.0
10%	43	383	8.08	6.19	79.3	64.2
15%	68	383	10.40	10.00	54.5	65.2
Average			8.77	8.04	35.8	46.7

Table 3: Comparison of Error Rates in Imbalanced Data sets

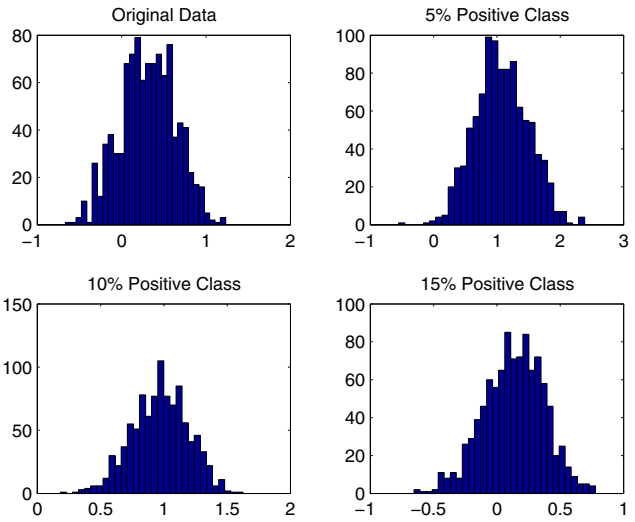

Figure 2: 1000 bootstrapped samples of the difference between the error rate of CBA and CCCS for the four different types of data sets. For the 5% and 10% data sets the zero point lies outside the bulk of the histogram.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases VLDB Proceedings*, pages 487–499, 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) Proceedings*, pages 64–69, 2004.

[3] B. Arunasalam and S. Chawla. Cccs: A top-down associative classifier for imbalanced class distribution. Technical Report TR 584, University of Sydney, 2006.

[4] C. Blake and C. Merz. *UCI KDD Archive*. http://kdd.ics.uci.edu/, 2000.

[5] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD International Conference on Management of Data Proceedings*, pages 265–276, 1997.

[6] G. Cong, A. K.H.Tung, X. Xu, F. Pan, and J. Yang. Farmer: Finiding interesting rule groups in microarray datasets. In *23rd ACM SIGMOD International Conference on Management of Data Proceedings*, pages 145–154, 2004.

[7] G. Cong, K.-L. Tan, A. K.H.Tung, and F. Pan. Mining frequent closed patterns in microarray data. In *2004 IEEE International Conference on Data Mining (ICDM'04) Proceedings*, pages 363–366, 2004.

[8] 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.

[9] M. Inc. *Matlab Statistical Toolbox*. Mathworks, 2005.

[10] W. Li, J. Han, and J. Pei. Cmar:accurate and efficient classification based on multiple class-association rules. In *2001 IEEE International Conference on Data Mining (ICDM'01) Proceedings*, pages 369–376, 2001.

[11] C. X. Ling and C. Li. Data mining for direct marketing:problems and solutions. In *International Conference on Knowledge Discovery and Data Mining(KDD'98) Proceedings*, pages 73–79, 1998.

[12] B. Liu, W. Hsu, and Y. Ma. *Data Mining II*. http://www.comp.nus.edu.sg/ dm2/index.html, 1998.

[13] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *International Conference on Knowledge Discovery and Data Mining(KDD'98) Proceedings*, pages 80–86, 1998.

[14] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[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.