On Pruning and Tuning Rules for Associative Classifiers

Osmar R. Zaïane and Maria-Luiza Antonie

University of Alberta, Edmonton, Canada {zaiane, luiza}@cs.ualberta.ca
Partially supported by Alberta Ingenuity Fund, iCORE and NSERC Canada.

Abstract. The integration of supervised classification and association rules for building classification models is not new. One major advantage is that models are human readable and can be edited. However, it is common knowledge that association rule mining typically yields a sheer number of rules defeating the purpose of a human readable model. Pruning unnecessary rules without jeopardizing the classification accuracy is paramount but very challenging. In this paper we study strategies for classification rule pruning in the case of associative classifiers.

1 Associative Classifiers and their massive model

Association rules are typically known as an important and common means for market basket analysis. However, it has been observed that association rules could be used to model relationships between class labels and features from a training set [4]. Therefore, association rules were used to efficiently build a classification model from very large training datasets. Since then, many associative classifiers were proposed mainly differing in the strategies used to select rules for classification and in the heuristics used for pruning rules [7, 6, 9]. Among the many advantages of associative classifiers we can highlight four major ones:

- The training is very efficient regardless of the size of the training set;
- Training sets with high dimensionality can be handled with ease and no assumptions are made on dependence or independence of attributes;
- The classification is very fast;
- The classification model is a set of rules easily understandable by humans and can be edited.

The problems with associative classifiers are also remarkable. First, they inherit two complicated parameters from association rule mining, namely *support* and *confidence*. These are difficult to set and tune. Second, association rule mining generates a sheer number of rules commonly outnumbering the observations in the training set. This defeats the purpose of readability of the classification model since no human would be willing to sift through hundreds of thousands of rules for editing purposes. This leads to two other issues: How can we reduce the number of rules in the model and how can we effectively select rules to apply during classification? In this paper we address one of these issues: the reduction of classification rules. This problem is challenging because the goal is to prune rules while preventing the accuracy of the classifier from dipping.

1.1 Motivation and contributions

Our strategy, as will be explained later on in the paper, is to generate association rules for each class in the training set separately. This strategy has advantages and disadvantages. The advantage is that with unbalanced training sets (i.e. training sets with rare classes) the small classes do not get overshadowed by the large classes, as is the case with other associative classification approaches. On the other hand, small classes end up generating a huge number of rules since, as will be explained later with the association rules, every feature in the few observations representing the rare classes becomes locally frequent and thus generates rules with high confidence. So dealing with rare classes is what initially motivated this work concerning pruning classification rules. However, in order to generalize the concepts, instead of using the rule generation by class using our *ARC-BC* algorithm [2], we use herein our *ARC-AC* [2] classifier which considers all classes together like other associative classifiers in the literature [7, 6].

In this paper we present an approach to prune the large set of classification rules using the rule performance on the training set. We show with progressive pruning techniques how the number of rules is reduced significantly without jeopardizing the accuracy of the overall classifier. In some cases, the accuracy is actually improved.

In the reminder of the paper we will briefly present the concepts related to association rule mining in Section 2 and will illustrate how these can be integrated to generate an associative classifier. In the same section, we will also introduce related work and highlight their different strategies. The rule pruning approaches will be presented in Section 3 and some experimental results will be illustrated in Section 4. Some conclusions are offered in Section 5.

2 Association rules and their integration in classifiers

The problem of mining association rules over market basket analysis was introduced in [1]. The problem consists of finding associations between items or itemsets in transactional data. The data is typically retail sales in the form of customer transactions, but can be any data that can be modeled into transactions. For example medical images where each image is modeled by a transaction of visual features from the image, or text data where each document is modeled by a transaction representing a bag of words, or web access data where click-stream visitation is modeled by sets of transactions, all are well suited applications for association rules or frequent itemsets.

Formally, the problem is stated as follows: Let $I = \{i_1, i_2, ...i_m\}$ be a set of literals, called items where m is considered the dimensionality of the problem. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A unique identifier, *TID*, is given to each transaction. A transaction T is said to contain X, a set of items in I, if $X \subseteq T$. An *association rule* is an implication of the form "$X \Rightarrow Y$", where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. An itemset X is said to be *frequent* if its *support* s is greater or equal than a given

minimum support σ. The rule $X \Rightarrow Y$ has a *support* s in the transaction set $\mathcal{D}$ if $s\%$ of the transactions in $\mathcal{D}$ contain $X \cup Y$. In other words, the support of a rule is the probability that X and Y hold together in $\mathcal{D}$. It is said that the rule $X \Rightarrow Y$ holds in the transaction set $\mathcal{D}$ with *confidence* c if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. In other words, the confidence of the rule is the conditional probability that the consequent Y is true under the condition of the antecedent X. The problem of discovering all association rules from a set of transactions $\mathcal{D}$ consists of generating the rules that have a *support* and *confidence* greater than given thresholds. These rules are called *strong rules*.

The first reference to using association rules as classification rules is credited to [4] while the first classifier using these association rules was CBA introduced in [7] and later improved in CMAR [6], and ARC-AC and ARC-BC [9]. The idea is relatively simple. Given a training set 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 association rules to 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 (feature set). The difference between CBA, CMAR and ARC-AC and ARC-BC lies in the strategy for rule selection during the classification. They also have some differences in pruning rules. CBA ranks all discovered rules by *precedence* ordering (using confidence then support) and simply selects the first ranked rule that applies given an object to classify [7]. CMAR takes all rules that apply within a confidence range and selects from this set the one with the highest χ^2 measure. ARC-AC and ARC-BC also take all rules that apply within a confidence range, but instead, calculate the average confidence for each set of rules grouped by class label in the consequent and select the class label of the group with the highest confidence average. Another difference is that ARC-AC, CMAR and CBA generate the association rules from all training transactions together. ARC-BC, on the other hand, generates association rules for transactions grouped by class label, each class at a time, giving this way a chance to small classes to have representative classification rules. Another interesting but not very convincing approach proposed in [5] suggests to consider the size of the antecedent and favour long rules before making an allowance for confidence and support. Their experimental results are unfortunately not compelling.

3 Pruning rules

As stated in [4, 7, 6], associative classifiers generate an overwhelming number of classification rules and it is very important to prune the rules to make the classifier effective and more efficient. We argue that pruning is also very important in order to allow domain experts to tune a classifier by editing rules if necessary. Our previous experiments show that manual alteration of the rules can lead to significant improvement in the classification [3]. The techniques proposed to prune the rules are based on redundancy and noise elimination and precedence ranking. For example contradictory rules such as $X \Rightarrow C1$ and $X \Rightarrow C2$

are eliminated in the case of single class classification. More specific rules are favoured. For example given two rules $R_1 : X \Rightarrow C$ and $R_2 : Y \Rightarrow C$ if both have the same confidence and $X \subset Y$, only R_1 is kept and R_2 is eliminated. Another accepted method of pruning is *database coverage* introduced in [7] and used in [6]. Database coverage consists of going over all the rules and evaluating them against the training instances. Whenever a rule applies correctly on some instances, the rule is marked and the instances eliminated until all training instances are covered. Finally, the unmarked rules are simply pruned.

Our associative classifier ARC-AC uses only database coverage because other prunings influence the accuracy on many real application datasets. While many rules are eliminated this way, we still find that the number of remaining rules is crushing and further pruning is required. The question is how can we remove more rules without jeopardizing the accuracy of the classifier. We propose to study the performance of each rule in re-classifying the training set and plotting the graph for correct classifications and incorrect classifications for each rule. Figure 1 shows an example. Each rule is plotted with the number of true positives and false positives scored on the training set. The rules plotted high on the graph incorrectly classified many instances. Rules that are plotted towards the right of the graph correctly classified many instances. Note that correct classification does not exclude incorrect classification. One given rule can do both for a large number of instances. The idea of exploiting the graph is to identify culprits of many misclassifications. To do this, we suggest four alternatives that can be executed progressively. Figure 1 illustrates these alternatives. We can visually identify the good and the poor rules.

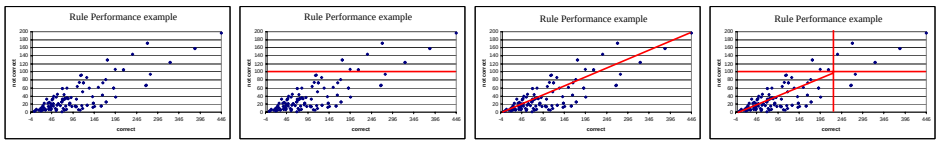

Fig. 1. Filtering by quadrant and diagonal slicing

1. Eliminate the high offender rules: By tracing a horizontal line at a given threshold, we can eliminate all the rules above the line. We suggest a line at 50% by default but a sliding line can also be possible aiming at a certain percentage of rules to eliminate.
2. Eliminate the rules that misclassify more than they classify correctly: By tracing a diagonal line such rules can be identified. Notice that when the axes of the plot are normalized, the diagonal indicates the rules that correctly classify as many times as they misclassify. When the axes are not normalized, the diagonal indicates a relative ratio, which we advocate.
3. Elimination by quadrant slicing: The plot could be divided into four regions. The top left ($RegionA$) contains rules that are incorrect more than they are correct. The top right ($RegionB$) contains rules that are frequently used but equally misclassify and correctly classify. The bottom left ($RegionC$)

has rules that are infrequently used but equally misclassify and correctly classify. Finally, the bottom right ($RegionD$) contains the good rules which frequently classify correctly but seldom misclassify. The idea is to successively remove the rules that are in $RegionA$, then $RegionB$, then $RegionC$.

4. A combination of the above methods: After removing regions A and B, eliminating the rules in $RegionC$ (bottom left) can be costly because many rules may be seldom used but have no replacements. Once removed, other rules are "forced" to play their role and can in consequence misclassify. The idea is to use a diagonal line to identify within $RegionC$ the rules that misclassify more than they are correct. This strategy is a good compromise.

Pruning classification rules is a delicate enterprise because even if a rule misclassifies some objects, it has a role in correctly classifying other objects. When removed, there is no guarantee that the object the rule used to correctly classify will be correctly classified by the remaining rules. This is why we advocate the progressive strategies depending upon the datasets at hand. We found that Strategy 4 (combining quadrant and diagonal pruning) allows in general a good result. A good result here means that we reduce the number of rules while keeping or improving the accuracy of the classifier as much as possible.

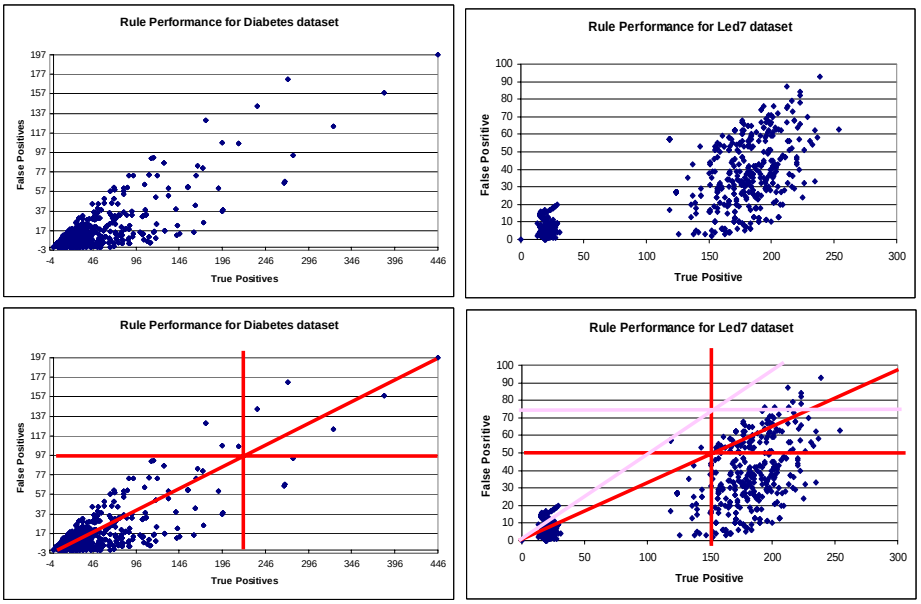

Fig. 2. Rule Performance Plot and Rule pruning for *Diabetes* and *Led7* datasets.

4 Experimental results and performance study

We run a battery of experiments to evaluate our strategies. For lack of space, we report herein a representative fraction of these experiments with good and less encouraging results. We used datasets from the UCI ML repository [8] and

the performance of CBA and CMAR are from their respective authors' papers [7, 6]. We used a 10 fold cross-validation method for each dataset and what is reported are averages. The table in Figure 3 shows the comparative results for five datasets namely *Breast, Diabetes, Iris, Led7* and *Pima*. We first tested our strategies without any additional pruning then added the best strategy to the database coverage technique. We chose to report on datasets that have different distributions of rules. For instance Figure 2 shows the plots for *Diabetes* with rules concentrating in the bottom left corner and distributed sparsely along the diagonal, and *Led7* with two clusters of rules, one of rarely used rules and one of frequently used rules. The table in Figure 3 first compares ARC-AC with and without database coverage pruning against CBA and CMAR. ARC-AC is the winner on this small collection. With database coverage, the accuracy is still very good while the number of rules drops significantly. Figure 3 also shows the effect of the pruning strategies when no other pruning technique is applied. While Strategy 3 has the worst accuracy result overall, it drastically reduces the number of classification rules without bringing the accuracy too low. In the case of *Led7*, this strategy was actually very good. By eliminating the entire cluster of rules in *RegionC* the performance was better than eliminating a portion of it in Strategy 4. This is because when the entire cluster is eliminated, the remaining rules sharing the task of classifying objects, normally classified by a rule from the cluster, do an excellent job at it. When removing only part of the cluster in Strategy 4, the rules that fire for the objects classified by the pruned rules are actually those remaining from the small cluster and they misclassify indeed. Strategy 3 is thus too drastic, while Strategy 4 is a good compromise.

Dataset	Breast	Diabetes	Iris	Led7	Pima	*Average*
CBA	**96.30**	74.50	**94.70**	71.90	72.90	82.06
CMAR	96.40	75.80	94.00	**72.50**	75.10	82.76
ARC-AC w/o any pruning	95.14	79.17	94.00	71.57	78.46	83.67
number of rules w/o pruning	16738	4086	135	656	4083	
ARC-AC + database coverage pruning	94.29	78.14	94.00	71.24	78.52	83.24
number of rules (with db coverage)	146	205	35	250	205	
Strategy 1 (horizontal slicing)	95.29	**79.44**	94.67	71.57	**78.52**	**83.90**
Number of rules after strategy 1	14800	3500	100	645	3900	
Strategy 2 (diagonal slicing)	95.58	78.26	94.67	64.66	77.61	82.16
Number of rules after strategy 2	13000	2500	100	520	2500	
Strategy 3 (Quadrant A+B+C)	65.53	65.11	94	71.69	65.11	72.29
Number of rules after strategy 3	1006	**120**	32	435	**119**	
Strategy 4 (quadrant AB + diagonal C)	95.86	79.18	94.67	68.44	77.61	83.15
Number of rules after strategy 4	13000	2500	98	520	2400	
ARC-AC + DB cov + Strategy 4	93.43	78.27	94.00	62.10	78.00	81.16
Number of rules (strategy 4 + Db cov.)	**135**	180	**30**	**208**	190	

Fig. 3. Comparison of CBA, CMAR, ARC-AC and the pruning strategies

The winning strategy overall (for the reported datasets) is the simple horizontal slicing of Strategy 1. It outperforms CBA and CMAR on two datasets. In most datasets the bar was put at 50% except for *Led7* for which it was set at 75% since many of its rules are in *RegionB*. However, based on our other evaluations, Strategy 4 is typically the winner overall. By slicing horizontally

at a given percentage of the *False Positives* and then vertically at a certain percentage of the *True Positives*, we generate four regions of unequaled areas A, B, C and D. Then by removing rules in A, B and above the diagonal of C we make sure that we eliminate the rules with the highest ratio of incorrect versus correct classifications, yielding a smaller set of classification rules but a good overall accuracy. Combining Strategy 4 with database coverage further reduces the number of rules while the performance in accuracy remains adequate. In the case of *Diabetes* the accuracy actually improved while the number of rules was reduced by 10%.

5 Conclusion and future work

Associative classifiers by piggybacking on the association rule mining technology are cursed by the combinatorial explosion in the number of classification rules generated. This extraordinary number of rules has a consequence on the efficiency of a classifier, but more seriously makes it impossible to manually edit and improve the rules by adding domain knowledge in the model. Inserting domain knowledge is often desirable and association rules are in theory readable by humans. In this paper we propose some strategies to prune the classification rules without severely hindering on the classifier's performance, and sometimes even improve its accuracy. The pruning strategies are simple and are based on individual rule performance when re-classifying the training set. A visual and interactive user application for rule pruning is desired. We are currently working on such interface using the plot presented above that allows interactive selection of rules for pruning and editing, visualizing rule performance and colour coded confidence and support.

References

1. Agrawal, R., Srikant, R. Fast algorithms for mining association rules. In *Proc. Intl. Conf. on Very Large Data Bases*, (1994) 487–499
2. Antonie, M.-L., Zaïane, O. R. Text document categorization by term association. In *Proc. of the IEEE International Conference on Data Mining (ICDM'02)*, (2002) 19–26
3. Antonie, M.-L., Zaïane, O. R., Coman, A. Associative Classifiers for Medical Images, In Mining Multimedia and Complex Data (LNAI 2797), Springer-Verlag, (2003) 68–83
4. Bayardo, R.: Brute-force mining of high-confidence classification rules. In *3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, (1997) 123–126
5. Coenen, F., Leng, P. An evaluation of approaches to classification rule selection. In *IEEE International Conference on Data Mining (ICDM'04)*, (2004) 359–362
6. Li, W., Han, J., Pei J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In *IEEE International Conference on Data Mining (ICDM'01)*, (2001) 369–376.
7. Liu, B., Hsu, H., Ma, Y.: Integrating classification and association rule mining. In *4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD'98)*, (1998) 80–86.
8. UCI repository, http://www.ics.uci.edu/~mlearn/MLRepository.html
9. Zaïane, O. R., and Antonie, M.-L. Classifying text documents by associating terms with text categories. In *Proc. of the Thirteenth Australasian Database Conference (ADC'02)*, (2002) 215–222.