Classification Using Multiple and Negative Target Rules

Jiuyong Li and Jason Jones

Department of Mathematics and Computing,
University of Southern Queensland,
Australia, 4350
jiuyong@usq.edu.au, jonesj@usq.edu.au

Abstract. Rules are a type of human-understandable knowledge, and rule-based methods are very popular in building decision support systems. However, most current rule based classification systems build small classifiers where no rules account for exceptional instances and a default prediction plays a major role in the prediction. In this paper, we discuss two schemes to build rule based classifiers using multiple and negative target rules. In such schemes, negative rules pick up exceptional instances and multiple rules provide alternative predictions. The default prediction is removed and hence all predictions relate to rules providing explanations for the predictions. One risk for building a large rule based classifier is that it may overfit training data and results in low predictive accuracy. We show experimentally that one classifier is more accurate than a benchmark rule based classifier, C4.5rules.

Keywords: classification, association rule, negative and multiple rule.

1 Introduction

Rules are a type of human-understandable knowledge, and therefore rule-based methods are very popular in building decision support systems. Last twenty years saw a lot of rule classification systems, such as, the covering algorithm based systems, e.g. AQ15 [8], CN2 [2,3], decision tree based systems, e.g. C4.5rules [9], and association rule based systems, e.g. CBA [6] and CMAR [5].

Apart from some routine processes, such as data preprocessing and result presentation, rule based classification usually involves two stages, training and test. Consider a relational data set where each record is assigned a category (class), called a training data set. In the training stage, a rule set is generated from the training data set. Each rule associates a pattern with a class. In the test stage, rules are used to predict classes of records that have no class information. If the predictive class is the class that a record is supposed to belong to, then the prediction is correct. Otherwise, the prediction is wrong. The proportional of correct predictions on the test data is the accuracy of a classifier.

A classifier refers to a rule set and the mechanism to make predictions. Highly accurate classifiers are generally preferred. One commonly used rule based classifier is ordered rule based classifiers. Rules are organised as a sequence, e.g. in the

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part I, LNAI 4251, pp. 212–219, 2006.
© Springer-Verlag Berlin Heidelberg 2006

descending accuracy order. When it classifies a coming record, the first matching rule in the sequence makes the prediction. This sequence is usually tailed by a default class. When there is no rules in the sequence matching the coming record, the class of the record is predicted as the default one. C4.5rules [9] and CBA [6] employ this model.

An ordered rule based classifier is simple and effective. Its predictions are easy to be interpreted and its wrong predictions are easy to be traced down since only one rule is used. It makes a prediction based on the most likelihood. This is because that a rule with higher accuracy usually precede a rule with lower accuracy and the accuracy approximates the conditional probability when a data set is big.

Traditional classification problems normally involve two classes and hence a prediction is either one class or the other class. When the number of classes is big, it is difficult to predict all instances to one class. One reason is that some patterns are association with multiple classes. If we restrict predictive class to one, this results in low accuracy of some rules. These low accurate rules make predictions unreliable and lose their utility significantly. It is desirable to have some multiple target rules to overcome this drawback of single target rules. Practically, predicting an instance to belong to two classes out of ten possible classes with 90% accuracy is more useful than predicting it to belong to one class with 50% accuracy.

In additional, predictions made by the default class may be misleading. For example, in data set Hypothyroid, 95.2% records belong to class Negative and only 4.8 % records belong to class Hypothyroid. So, if we set the default prediction as Negative, then a classifier that has no rule will give 95.2% accuracy. You can see that how accuracy is floated by the default prediction. Further, this distribution knowledge is too general to be useful. For example, a doctor uses his patient data to build a rule based diagnosis system. 95% patients coming to see him are healthy, and hence the system sets the default as healthy. Though the default easily picks up 95% accuracy, this accuracy is meaningless for the doctor.

In this paper, we propose to use negative and multiple rules to build rule based classifiers. Negative rules summarise exceptional instances of low accurate rules and multiple target rules provide alternative predictions. We also drop the misleading default predictions in the classifiers. A risk of building a classifier from a large rule set is that it may reduce the accuracy of the classifier because a large model usually tends to overfit data. We experimentally demonstrate that the new schemes do not overfit data, and can improve accuracy of classifiers over a benchmark rule based classifier, C4.5rules.

2 Multiple and Negative Rule Based Classifier

2.1 Multiple and Negative Rules

Let us start with an example. We observer that 60% customers buying product a also buy product b, and hence summarise this phenomenon as rule $a \rightarrow b$.

Afterwards, when a customer put product a in his/her shopping trolley, he/she is recommended to buy product b. Should store mangers promote product b targeting this group of customers buying product a? We look at some possible consequences. There may be 10% customers buying product a hate the product b, and therefore this promotion angers these customers. Further, other 30% customers may be annoyed since they are not interested in product b at all.

A solution is to find the 10% customers and exclude them from the the promotion list; and to find another product that is of interest to the 30% of customers and bind two products in one promotion. This solution minimises the probability of annoying customers. This is the basic idea for negative and multiple target rules.

Many real world problems have a number of classes. In some cases, a pattern is association with two or three classes, and it is impossible to have an accurate rule to associate an instance to one class. Therefore it is necessary to extend the traditional classification rules to target multiple classes. Practically, these rules are interesting if they restrict the choices to two or three among ten or twenty classes. In e-commence applications the number of classes may be thousands.

For example, there are ten classes in a data set, and we have two rules $A \rightarrow c_1$ (conf = 48%) and $A \rightarrow c_2$ (conf = 42%). Either rule is accurate enough to be a rule alone, but the joint rule $a \rightarrow c_1 \vee c_2$ has the confidence of 90% and is an accurate rule. Predictively, eliminating eight of the potential ten classes as outcomes means that the rule is still informative.

Multiple target rules are different from general association rules. The consequent of a general association rule can be a number of conjunctive items, but the consequent of a multiple target rule is a number of disjunctive items.

Almost all statements have exceptions, and rules have exceptions too. Rule $AX \rightarrow \neg c_1$ is an exceptional rule of rule $A \rightarrow c_1$. This means that pattern A generally associates with class c_1 but does not when it occurs with pattern X. Using negative rules to filter out exceptional instances of a regular rule can increase the accuracy of the regular rule.

For example, $A \rightarrow c_1$ (supp = 15%, conf = 50%) and $AX \rightarrow \neg c_1$ (supp = 5%, conf = 100%). The combination of two rules will produce the confidence of 75% since 5% exceptional instances are removed from the former rule.

2.2 Two MTNT Classifiers

We will present two classifier models with multiple and negative rules in this section.

The first model (MTNT1) does not directly use multiple and negative target rules for making prediction, but simply provides some backup explanations for low accurate rules. A low accurate rule is coupled by some negative and multiple target rules, and as a result users will be aware of possible adverse effects of the prediction made by the rule and possible alternative prediction.

In MTNT1, all regular rules are sorted by their estimated predictive accuracy and there is no default class at the end of this rule sequence. Confidence is

accuracy on the training data, but accuracy on test data is required for prediction. We use method in [4] to estimate predictive accuracy of a rule.

In classification, the first matching regular rule classifies a record. Multiple and negative target rules do not participate in the classification, but provide some backup explanations. An example of MTNT1 classifier is shown in Figure 1.

Rule 1: $A \rightarrow c_1$		acc = 95%
Rule 2: $B \rightarrow c_2$		acc = 92%
...	...	...
...	...	...
Rule 11: $D \rightarrow c_5$		acc = 70%
$DE \rightarrow \neg c_5$		
$DF \rightarrow \neg c_5$		
...	...	...
$D \rightarrow c_2 \vee c_5$		
Rule 12: ...		...
...	...	...
...	...	...
No default class		

Fig. 1. An example for MTNT1 classifier. Rules are sorted by their estimated accuracy and low accurate rules are coupled by multiple and negative rules for better understanding. For example, c_2 is an alternative prediction for Rule 11 and some exceptional instances of Rule 11 are explained by the following negative rules.

The second model (MTNT2) makes use of multiple and negative target rules in predictions. We use a covering algorithm based method to sort regular rules in classifier as C4.5rules [9] and CBA [6]. The rule with the fewest false positive errors is put the first. Then all covered records by the selected rule are removed, and false positive errors are recomputed for the remaining rules. This procedure repeats until there is no records or rules left. Unlike C4.5rules [9] and CBA [6], there is no default class. Remaining rules are appended to this sequence in the accuracy decreasing order.

Multiple and negative rules are inserted in the classifier in the following way. Negative target rules are put before their corresponding regular rule to filter exceptional instances. All multiple target rules are appended to the end in the accuracy decreasing order. An example of MTNT2 is shown in Figure 2.

In classification, for a record to be classified, it is compared with rules from the top to the bottom. If a matching rule is a single target rule without any negative rules, then the rule classifies it. If the matching rule has negative rules, we move into the negative rule subset and match each of them. If no negative rule matches the record, then the rule classifies it. Otherwise, no prediction is made. We move on to the next rule and repeat the process.

Exceptional instances of a regular rule are removed by the negative rules. As a result, the rule group, including a regular rule and some negative rules,

Rule 1: $A \rightarrow c_1$		acc = %95
Rule 2: $B \rightarrow c_2$		acc = %92
...	...	...
...	...	...
Rule 11: $DE \rightarrow \neg c_5$		
$DF \rightarrow \neg c_5$		
$D \rightarrow c_5$		acc = %70
...	...	...
...	...	...
Rule 49: $E \rightarrow c_2 \vee c_3$		acc = %92
Rule 50: $D \rightarrow c_2 \vee c_5$		acc = %90
...	...	...
...	...	...
No default class		

Fig. 2. An example for MTNT2 classifier. Regular rules are ordered by a covering algorithm based method to minimise false positive errors. Negative target rules filter the exceptional instances for the following regular rule, and as a result the regular rule makes more accurate predictions. All multiple target rules reside at the end in the accuracy decreasing order to make alternative predictions when an accurate prediction is impossible.

is more accurate than the single regular rule. Alternative predictions are made by multiple target rules. In some cases, we may not define a coming record in a single class. It is still useful to classify it into two or three possible classes when the number of all possible classes is big. Therefore, we put a set of multiple target rules at the end of classifier to for this purpose.

This classifier sometimes results in the classification of two or more classes simultaneously, however, we do not consider this as the conflicting prediction. When the number of classes is big, this prediction narrows the choice to a smaller number, e.g. 2. This prediction is useful since it excludes many other choices. This prediction is not as accurate as the prediction of one class, but in some cases it is impossible to predict one class accurately. We scale down the predictive accuracy of multiple target rules and details are presented in the following section.

3 Experimental Results

In the previous section, we mainly discuss how to incorporate multiple and negative target rules in a classifier to improve the explanatory ability of a rule based classifier. We also drop the default prediction in the classifier so that every prediction relates to rules. However, a major concern is that a large classifier may overfit data and reduce its predictive accuracy. In this section, we will show that our classifiers do not sacrifice the accuracy of classifiers.

We carried out experiments by using 10-fold cross validation on 7 data sets from the UCI Machine Learning Repository [1]. We chose them since they contain 4 or more classes each. Multiple and negative target rules are not suitable for two-class data sets.

We compare the MTNT classifiers with c4.5rules [9]. We chose c4.5rules since we are able to modify the code to drop its default prediction. In addition, nearly all new classifiers have been compared with C4.5, and hence interesting readers can compare the MTNT classifiers with other classifiers indirectly.

In the experiments, we used the local support. The *local support* of rule $A \rightarrow c$ is $\text{supp}(Ac)/\text{supp}(c)$. It avoids too many rules in the large distributed classes and too few rules in the small distributed classes. We explored negative rules to depth 3, and therefore the maximum length of any negative rule is the length of regular + 3. The maximum classification rule length was set as 6, the minimum accuracy threshold was set as 50%, the high accuracy threshold was set as 90%, and the number of multiple targets was set as 2.

There is no default prediction for MTNT classifiers. If no one rule matches a record, an error is counted.

A brief description of data sets and classifiers is given in Table 1. MTNT classifiers are four times larger than C4.5rules on average. This is because the default prediction is a vital part in a C4.5rules classifier. The construction of C4.5rules classifiers has a post-pruning process. All rules and the default prediction are considered as a whole. Any rule that does not contribute to increase the accuracy is eliminated. Removed rules may not be as good as the default in terms of accuracy, but they provide direct reasons for correct or wrong predictions. In other words, they make a classifier more understandable. Therefore, we keep larger classifiers.

On average, a regular rule in MTNT classifiers has 2 negative target rules and four regular rules have 1 multiple target rule. We did not set the minimum support requirement for negative target rules and therefore their number is comparatively big. In contrast,the number of the multiple target rules is small since the they have to reach the high accuracy threshold.

Table 1. A brief description of data sets and classifiers. In classifier size columns, (M) means #multiple rules, (N) means #negative rules, and no symbol means #regular rules.

Data sets			classifier size	
Name	#records	#classes	MTNT	C4.5rules
anneal	898	5	42 + 30(M) + 51(N)	22
auto	204	7	50 + 6(M) + 244(N)	27
glass	214	7	29 + 2(M) + 21(N)	12
led7	3200	10	161 + 32(M) + 76(N)	32
lymph	148	4	33 + 10(M) + 123(N)	11
vehicle	846	4	242 + 42(M) + 510(N)	47
zoo	101	7	8 + 6(M) + 7(N)	9
Average	n/a	n/a	80 + 18(M) + 147(N)	23

Table 2. Accuracy of different classifiers (in %)

data set name	MTNT1 no default	MTNT2 no default	C4.5rules no default	C4.5rules default
anneal	95.6	96.9	90.3	93.5
auto	70.3	77.5	75.1	78.0
glass	71.6	74.9	63.6	72.5
led7	72.0	73.9	73.2	73.2
lymph	80.5	83.1	73.1	78.4
vehicle	69.8	70.6	67.3	71.9
zoo	93.1	94.8	92.1	92.1
Average	79.0	81.6	76.4	79.9

To have a fair comparison, predictions made by multiple target rules are scaled down so that they scored $1 - \frac{Num\ Target}{Max\ Class}$ for a correct prediction instead of 1. For example, for a data set with 4 classes, a multiple target rule with two classes makes a correct prediction. We consider this as a 0.5 correct prediction instead of 1. Therefore, we do not expect the multiple target rules to increase the accuracy of classifiers, but to improve the understandability of the predictions.

MTNT2 is more accurate than C4.5rules (with default) and MTNT1 is nearly as accurate as C4.5rules (with default), as shown in Table 2. Consider both MTNT classifiers do not include the default prediction. The MTNT classifiers have successfully dropped the default prediction while maintaining accuracy. The default is replaced by some additional rules, which make correct or wrong predictions directly associate with rules.

We are also aware that a number of new rule based classifiers, e.g. CBA [6] and its enhancement [7], have been proposed. They are more accurate than the C4.5rules. However, they make use of the default prediction, a factor that reduces the explanatory ability of a rule based classifier. We did not compare with them since we are unable to drop their default predictions.

4 Conclusions

In this paper, We proposed to incorporate multiple and negative rules in rule based classifiers. Negative rules are used to pick up exceptional instances and multiple rules are used to provide alternative predictions. They improve explanatory ability of predictions made by low accurate rules by characterising their exceptional instances and providing alternatives. We proposed two schemes to bind the multiple and negative rules to classifiers and remove the default prediction. We experimentally shows that one proposed classifier is more accurate than a benchmark rule based classifier, C4.5rules.

The utility of multiple and negative rules has strong practical implication in eCommerce applications, e.g. target-commercial and personalization. In these applications, the number of possible targets is very big, and rules are usually low accurate. The utility of multiple and negative rules is an approach to obtain more certain information in these data.

Acknowledgement

This project has been partially supported by Australian Research Council Discovery Grant DP0559090.

References

1. E. K. C. Blake and C. J. Merz. UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.
2. P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning - EWSL-91*, pages 151–163, 1991.
3. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
4. J. Li, H. Shen, and R. Topor. Mining the optimal class association rule set. *Knowledge-Based System*, 15(7):399–405, 2002.
5. W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE International Conference on Data Mining (ICDM 2001)*, pages 369–376. IEEE Computer Society Press, 2001.
6. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 27–31, 1998.
7. B. Liu, Y. Ma, and C. Wong. Improving an association rule based classifier. In *4th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD*, pages 504–509, 2000.
8. R. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The AQ15 inductive learning system: an overview and experiments. In *Proceedings of IMAL 1986*, Orsay, 1986. Université de Paris-Sud.
9. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.