Using Multiple and Negative Target Rules to Make Classifiers More Understandable

Jiuyong Li [1]

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

Jason Jones

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

Abstract

One major goal for data mining is to understand data. Rule based methods are better than other methods in making mining results comprehensible. However, the current rule based classifiers make use a small number of rules and a default prediction to build a concise predictive model. This reduces the explanatory ability of a rule based classifier. In this paper, we propose to use multiple and negative target rules to improve explanatory ability of rule based classifiers. We show experimentally that this understandability is not at the cost of accuracy of rule based classifiers.

Key words: classification, association rule, negative and multiple rule.

1 Introduction

Two major tasks of data mining are to understand data and predict trends. A rule based method is one of the best methods to achieve the both goals.

[1] Corresponding author, tel: 61 7 46315548, fax: 61 7 46315550.
This project has been partially supported by Australian Research Council Discovery Grant DP0559090.

In most applications, knowledge discovered by data mining is used to assist humans to make decisions. For example, doctors use diagnosis systems and sale managers use cross-sale promotion systems in this way. Therefore, understandable reasons form an indispensable part of predictions.

In the decision making, understandable reasons help decision makers to forecast the possible favourable and adverse consequences of a decision, and to find possible mistakes of a decision. More importantly, the understandability avoids many blunt decisions.

Rules are a type of the most human-understandable knowledge, and therefore rule-based methods are very popular in building decision systems. Last twenty years saw a lot of rule discovery methods, such as, the covering algorithm based methods, e.g. AQ15 [18], CN2 [8,9], decision tree based method, e.g. C4.5rules [19], association rule mining methods, e.g. Apriori [1] and FP-growth [12], and optimal rule mining methods, e.g. PC optimality rule mining [4] and optimal class association rule mining [14]. However, understandable rules may not build understandable classifiers.

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, it 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 for it to make predictions. Highly accurate classifiers are generally preferred.

There are roughly two types of models for building rule based classifiers.

(1) Ordered rule based classifiers: rules are organised as a sequence, e.g. in the 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 [19] and CBA [16] employ this model.
(2) Unordered rule based classifiers: rules are not organised in a sequence and all (or most) matching rules participate in the determination of the class of a coming record. A straightforward way is to accept the majority votes of rules like in CPAR [21]. A more complex method is to compare the combined accuracies obtained from the multiple rules for all possible classes. The one obtaining the highest accuracy will be the final prediction. Improved CN2 [8] and CMAR [15] employ this method.

We do not discuss committee prediction, e.g. Bagging [7] and Boosting [10,11], since they use multiple classifiers.

The second model looks more attractive and has great potential for improving classification accuracy, but it may not be good for understanding. The characteristic of the second model is to vote in the prediction. Simple voting is subject to bias since all rules may not be independent. A more complex statistical model may avoid the bias, but becomes too complex for users, such as online shop owners, to understand. Further, it is very difficult to trace down to find the reasons for wrong predictions when a complex model is used.

In contrast, the first model is simple and effective. Its predictions are easy to be understood 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 large. However an ordered rule based classifier has the following two major drawbacks.

Firstly, predictions made by low accurate rules are too weak for users to use. When a prediction is made by a rule with low accuracy, it has great uncertainty. For example, if we have a rule $A \rightarrow c_1$ with 60% accuracy, then it roughly makes 40% wrong predictions. Most people would like to know more about this 40% data before a decision is made. However, the ordered rule based classifiers do not provide such information.

Secondly, 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 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 complement a weak regular rule in prediction. Negative rules summarise exceptional cases of the weak regular rule and multiple target rules provide alternative predictions. We also drop the misleading default predictions in the classifiers so that all predictions relate back to rules. These remedies overcome two obstacles to providing understandable predictions of ordered rule based classifiers as discussed above. The 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 show that our proposed classifiers do not sacrifice the accuracy.

2 Motivation

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 a in the basket, he/she is recommended to buy b. Should store mangers promote b targeting this group of customers buying 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 the product b at all.

A solution is to find the 10% customers and exclude them from the mailing 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.

Traditional classification problems include only two classes, and the prediction is one or the other. However, 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 it to any 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.

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 so do rules. We may have a rule $A \rightarrow c_1$ and its exceptional rule $AX \rightarrow \neg c_1$. This means that pattern A generally associates with class c_1 but does not when it occurs with pattern X. These rules are very interesting when being considered in conjunction with a regular rule.

Only few rules are highly accurate, and many rules have exceptions which may discourage the original prediction target. If we can identify the exceptional cases for lower accuracy rules, then we may avoid many of blunt prediction errors.

3 Multiple and negative target rules

We define the multiple and negative target rules by the association rule terminology.

Given a data set $D = \{T_1, T_2, \ldots, T_n\}$. A pattern is a subset of a record $P \subset T_i$. The support for pattern P is the ratio of the number of records containing the pattern to the number of all records, denoted by $\operatorname{supp}(P)$. $P \to c_i$ is an implication if $c_i \notin P$. $\operatorname{supp}(P \cup c_i)$ and $\operatorname{supp}(P \cup c_i)/\operatorname{supp}(P)$ [2] are the support and the confidence of the implication. The confidence is denoted by $\operatorname{conf}(P \to c_i)$. We call an implication a rule if its support and confidence are greater than the predefined minimum support and confidence respectively.

In this paper, we will extend the concept of rules to multiple and negative target rules.

A multiple target rule takes a number of disjunctive targets as its consequent. In contrast, an association rule takes a number of conjunctive targets as its target. $P \to c_1 \vee c_2$ means patterns P associates with targets c_1 or c_2.

$$\begin{aligned}\operatorname{supp}(A \to c_1 \vee c_2 \vee \cdots \vee c_k) = &\sum_{i=1}^{k} \operatorname{supp}(A \to c_i) - \sum_{1 \le i_1 < i_2 \le k} \operatorname{supp}(A \to c_{i_1} c_{i_2}) \\ &+ \sum_{1 \le i_1 < i_2 < i_3 \le k} \operatorname{supp}(A \to c_{i_1} c_{i_2} c_{i_3}) + \cdots + \\ &(-1)^{k-1} \operatorname{supp}(A \to c_1 c_2 \ldots c_k) \end{aligned} \tag{1}$$

$$\begin{aligned}\operatorname{conf}(A \to c_1 \vee c_2 \vee \cdots \vee c_k) = &\sum_{i=1}^{k} \operatorname{conf}(A \to c_i) - \sum_{1 \le i_1 < i_2 \le k} \operatorname{conf}(A \to c_{i_1} c_{i_2}) \\ &+ \sum_{1 \le i_1 < i_2 < i_3 \le k} \operatorname{conf}(A \to c_{i_1} c_{i_2} c_{i_3}) + \cdots + \\ &(-1)^{k-1} \operatorname{conf}(A \to c_1 c_2 \ldots c_k) \end{aligned} \tag{2}$$

The support and confidence of multiple target rules look very complex. However, k is usually very small since otherwise multiple target rules are less interesting in practice. For example, when $k = 2$, their support and confidence become simple.

[2] In the rest of this paper, we use a capital letter to stand for a set of items, and a lower case letter for an item. AB means $A \cup B$, Ac means $A \cup \{c\}$, and ab means $\{a, b\}$.

$$\mathrm{supp}(A \rightarrow c_1 \vee c_2) = \mathrm{supp}(A \rightarrow c_1) + \mathrm{supp}(A \rightarrow c_2) - \mathrm{supp}(A \rightarrow c_1 c_2) \quad (3)$$
$$\mathrm{conf}(A \rightarrow c_1 \vee c_2) = \mathrm{conf}(A \rightarrow c_1) + \mathrm{conf}(A \rightarrow c_2) - \mathrm{conf}(A \rightarrow c_1 c_2) \quad (4)$$

The definitions become even simpler when all targets are disjoint, such as in most classification problems.

$$\mathrm{supp}(A \rightarrow c_1 \vee c_2) = \mathrm{supp}(A \rightarrow c_1) + \mathrm{supp}(A \rightarrow c_2) \quad (5)$$
$$\mathrm{conf}(A \rightarrow c_1 \vee c_2) = \mathrm{conf}(A \rightarrow c_1) + \mathrm{conf}(A \rightarrow c_2) \quad (6)$$

A negative rule takes an absent target as its consequent. The support and confidence of a negative rule are defined as follows.

$$\mathrm{supp}(AB \rightarrow \neg c_1) = \mathrm{supp}(AB) - \mathrm{supp}(ABc_1) \quad (7)$$
$$\mathrm{conf}(AB \rightarrow \neg c_1) = \frac{\mathrm{supp}(AB) - \mathrm{supp}(ABc_1)}{\mathrm{supp}(AB)} = 1 - \mathrm{conf}(AB \rightarrow c_1) \quad (8)$$

We note that a negative target rule is a low confidence implication. The number of all negative target rules is significantly bigger than that of regular rules. We do not intend to generate them all, but only use negative target rules to complement a regular rule. For example, $AB \rightarrow \neg c_1$ couples with $A \rightarrow c_1$. In this case, the negative target rule is an exceptional rule.

This emphasis makes our work distinct from other negative association rule mining research [3,2,20,22], which focuses on solving efficiency problems of general negative association rule mining. Our negative target rules are very similar to those in [13]. In this paper, we use negative target rules together with regular and multiple target rules to make predictions more understandable.

In general, a negative target rule is a special case for the multiple target rules. A rule targeting $(m-1)$ targets, where m is all possible targets, is a negative target rule targeting the absence of the remaining target. In practice, we do not seek all multiple target rules, but some with very small number of targets, for example, 2 or 3. Further, we use multiple target rules and negative target rules for different purposes. Therefore, we distinguish them in this paper.

For the convenience of the following discussions, we define that a rule covers a record if the antecedent of a rule is a subset of the record. The covered set of the rule is the set of all records containing the antecedent of the rule, denoted by $\mathrm{cov}(P \rightarrow c)$.

We have the following relationship immediately.

$$\mathrm{cov}(A \rightarrow c_1) = \mathrm{cov}(A \rightarrow c_1 \vee c_2 \vee \cdots \vee c_k) \tag{9}$$
$$\mathrm{cov}(AB \rightarrow \neg c_1) \subseteq \mathrm{cov}(A \rightarrow c_1) \tag{10}$$

A multiple target rule set covers the same amount of records as its component rule, but with different confidence. A negative target rule covers a part of records covered by its corresponding regular rule, and picks up exceptional cases for the regular rule.

4 Mining MTNT rules

We first discuss the nature of mining multiple and negative target rules.

Mining multiple target rules is an extension for mining association rules under the following assumptions. First a rule is frequent if its antecedent pattern, e.g. A in Equations 1 and 2, is frequent. Second, the number of targets is limited, e.g. 2 or 3. Under the assumptions, infrequent rules are pruned by the support property and the cost for combining regular rules to multiple target rules is not big.

Mining negative target rules is equivalent to mining association rule with very low minimum support. We are interested in negative rules that couple with regular rules. Let rule $AB \rightarrow \neg c_1$, which is an opposite rule of $AB \rightarrow c_1$ according to Equations 7 and 8, be a negative target rule of $A \rightarrow c_1$. Rule $A \rightarrow c_1$ is frequent, but $AB \rightarrow c_1$ may not be. Therefore, the minimum support should be set very low in order to find negative rules.

However, it is inefficient to mine multiple and negative target rules by association rule mining approach, although conceptually they are variants of association rule mining. First, the number of rules will be too many and the secondary data mining is required to process these rules. Association rule mining produces a large number of rules and this number will be increased significantly when the multiple and negative target are considered. Secondly, it may be infeasible to directly apply an association rule mining method to some data sets when the minimum support is low.

We use a two-tier algorithm to generate multiple and negative rules. First, we generate a small rule base, min-optimal class association rule set. Second, we generate multiple and negative rules to complement rules in the min-optimal rule set. Assume that we have a class association rule set for a data set. The rule with the highest accuracy is called the best match rule for a record among all rules covering the record. The min-optimal rule set is the set of all best match rules for records in the data set. The min-optimal rule set is very small in comparison with an association rule set. Therefore, mining multiple and negative rule to complement rules in this small rule set is very efficient.

A straightforward method to generate the min-optimal rule set is to generate all rules by an association rule mining method and then select the set of best match rules. This method is inefficient since the relational data set is usually dense and makes an association rule mining method inefficient.

We do not need to generate all rules to find the min-optimal rule set. Consider the two rules $A \rightarrow c_1$ and $AB \rightarrow c_1$. The covered set of the latter rule is a subset of that of the former rule, i.e. every record covered by the latter rule is also covered by the former rule. If the latter rule is less accurate than the former rule, then it never has a chance to be a best match rule for any record.

The optimal class association rule set [14] is a subset of the class association rule set. It excludes all more specific rules, e.g. $AB \rightarrow c_1$ with lower accuracy than its more general rules, e.g. $A \rightarrow c_1$. It is much smaller than the class association rule set and can be generated more efficiently.

In our implementation, we extend our optimal class rule discovery algorithm [14] and the main procedure is summarised as follows.

(1) generate the optimal class association rule set;
(2) choose the min-optimal class association rule set including the highest accurate rules matching every record;
(3) find negative and multiple target rules for rules in the min-optimal rule set if their accuracies are lower than a high accuracy threshold, e.g. 90%.

An alternative for the first step is to use the constraint association rule mining method [5] by setting the minimum confidence improvement as zero. If we choose to do so, we need to discover the optimal rule sets for every class first and then union them as the optimal rule set. When the number of classes is large, this involves some redundant computation.

We generate multiple target rules in the following way. Given rule $A \rightarrow c_1$ with a low accuracy. We scan a data set once to count support for Ac_2, Ac_3, and so on. We generate multiple target rules whose accuracy is greater than the high accuracy threshold, e.g. 90%. We only generate multiple rules with at most two targets.

We generate negative target rules in the following way. Given rule $A \rightarrow c_1$. We extract the covered set of the rule, and relabel record classes that are not c_1 as $\neg c_1$. We then apply optimal class association rule discovery algorithm to the data set to find optimal class association rules for class $\neg c_1$. Since we only generate negative target rules with short antecedent, two or three conditions in addition to the regular rule, this procedure is efficient. We select negative rules whose confidence is greater than that of $A \rightarrow c_1$.

This is a regular rule guiding method for multiple and negative target rule discovery. The efficiency depends on the number of rules in the min-optimal class association rule set. We will show that the number of min-optimal class association rules is small, and therefore, this method is efficient.

5 MTNT Classifiers

A typical classifier is an ordered rule set tailed by a default prediction. When a new record is coming, the first matching rule in the sequence classifies it. When there is no rule in the classifier matching the record, it is classified as the default class. It is simple and effective. It is understandable since it associates a prediction with a rule. However, the default prediction does not provide much useful information. Further predictions made by low accurate rules need some backup explanations.

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 [14] 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

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 [19] and CBA [16]. 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 [19] and CBA [16], there is no default class. Remaining rules are appended to this sequence in the accuracy decreasing order.

Rule 1: $A \rightarrow c_1$		acc = 95%
Rule 2: $B \rightarrow c_2$		acc = 92%
...	...	...
...	...	...
Rule 11: $D \rightarrow c_5$		acc = 70%
$\quad DE \rightarrow \neg c_5$		
$\quad DF \rightarrow \neg c_5$		
...	...	...
$\quad D \rightarrow c_2 \vee c_5$		
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 cases of Rule 11 are explained by the following negative rules.

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 cases. 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 cases of a regular rule are removed by the negative rules. As a result, the rule group, including a regular rule and some negative rules, 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 large. Therefore, we put a set of multiple target rules at the end of classifier to for this purpose.

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 cases 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.

6 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 [6]. 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 [19]. 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.

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 [16] and its enhancement [17], have been proposed. They are more accurate than

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 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 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

Table 2
Accuracy of different classifiers (in %)

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.

7 Conclusions

In this paper, We proposed to use multiple and negative rules to increase the understandability of rule based clarifiers. They improve explanatory ability of predictions made by low accurate rules by characterising their exceptional cases and providing possible alternatives. We have discussed a regular rule guide method to generate multiple and negative rules and proposed two classifier models to make use of multiple and negative rules. We have shown experimentally that we successfully bind the multiple and negative rules to classifiers and remove the vague default prediction without sacrificing classification accuracy.

The utilisation 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 utilisation of multiple and negative rules is an approach to obtain more certain information in these data.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the Twentieth International Conference on Very Large Databases*, pages 487–499, Santiago, Chile, 1994.

[2] M. Antonie and O. R. Zaiane. Mining positive and negative association rules: An approach for confined rules. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 27 – 38, 2004.

[3] A.Savasere, E.Omiecinski, and S. B. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of International Conference on Data Engineering ICDE98*, pages 494–502, 1998.

[4] R. Bayardo and R. Agrawal. Mining the most interesting rules. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154, N.Y., 1999. ACM Press.

[5] R. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense database. *Data Mining and Knowledge Discovery Journal*, 4(2/3):217–240, 2000.

[6] E. K. C. Blake and C. J. Merz. UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[7] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[8] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning - EWSL-91*, pages 151–163, 1991.

[9] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

[10] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[11] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[12] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery Journal*, pages 53–87, 2004.

[13] F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining with a relative interestingness measure. In *Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD 00)*, pages 86–97. LNCS, 2000.

[14] J. Li, H. Shen, and R. Topor. Mining the optimal class association rule set. *Knowledge-Based System*, 15(7):399–405, 2002.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[20] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381 – 405, 2004.

[21] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *Proceedings of 2003 SIAM International Conference on Data Mining (SDM'03)*, 2003.

[22] X. Yuan, B. P. Buckles, Z. Yuan, and J. Zhang. Mining negative association rules. In *Proc of the Seventh IEEE Symposium on Computers and Communications*, pages 623 – 628, 2002.