Algorithm for Classification Based on Positive and Negative Class Association Rules

Luo Junwei  
College of Computer Science and Technology  
Henan Polytechnic University  
JiaoZuo, China  
ljwonly@yahoo.com.cn

Luo Huimin  
School of Computer and Information Engineering  
Henan University  
Kaifeng, China  
hmluo_henu@yahoo.com.cn

Abstract—The negative class association rules are important to build accurate and efficient classifiers. Despite a great deal of research, a number of challenges still exist. In order to solve the problem of “difficult to build precise classifier”, the paper presents a new algorithm for classification which integrates positive class association rules and negative class association rules. The algorithm applies Apriori method and correlation between itemsets and class labels to compute all positive and negative class association rules from training dataset. Moreover, a classifier will be built to predict the label of a new data object. The performance study shows that the method is highly efficient and accurate in comparison with other reported associative classification methods.

Keywords-Classification; Positive Class Association Rule; Negative Class Association Rule; Associative Classification;

I. INTRODUCTION

Building accurate and efficient classifiers for large databases is one of the important data mining techniques. Given a set of cases with class labels as a training dataset, classification aims to build a model (called classifier) to predict future data objects for which the class label is unknown[1].

Previous studies propose that associative classification has high classification accuracy and strong flexibility at handling unstructured data[2]. Until now, there are two general approaches which use association rules for classification:

(1) The first kind of approach is using association rules for classification directly. In this way, all association rules in the training dataset will be mined firstly, secondly these methods use the association rules to build classifiers, and lastly the classifiers use the strength of association rules or select a subset of association rules which match the data object to judge the class label of data object;

(2) The second kind of approach is using association rules as classification attributes. In this way, association rules will be used as classification attributes to enhance the accuracy, efficiency and scalability of training dataset, and then the methods use the traditional classifier for classification.

However, these approaches may also suffer some weakness.

On one hand, it is not easy to find interesting or useful rules. Traditional techniques always select some simple criterions such as confidence, distance functions, rough set and so on. The simple pick may affect the classification accuracy.

On the other hand, a training dataset often generates a huge set of rules. It is challenging to store, retrieve, prune, and sort a large number of rules efficiently for classification[3].

To solve these problems, in this paper, we develop a new algorithm for classification based on positive and negative association rules. Because the traditional approaches building classifiers often are based on the positive association rules, while ignoring the value of negative association rules for classification. Actually some useful discovery can be dug out through negative association rules from training dataset; we can find the class labels which test case does not belong to and some contradictory judgments. For example, there is a data object which has some attributes, and it is consistent with the relevant association rules: $X \rightarrow c, X \rightarrow \neg c$, then the data object can be determined that the possibility of it belongs to the class label $c$ is very small.

Therefore, in this paper we will find the positive and negative association rules from training dataset and prune contradictory positive and negative association rules. According to the collection of positive and negative association rules, this paper presents a new classifier to classify the data object. The final comparative experiment shows that the algorithm for classification based on positive and negative association rules has a higher recall rate and precision rate, is feasible and effective.

This work makes the following contributions:

(1) It proposes a new classifier instead of relying on positive association rules for classification. This classifier uses a small set of high confidence positive and negative rules to determine the class label of test case, and experimental results show that this way is, in general, more accurate than other techniques;

(2) It proposes a collection of positive and negative associative rules which is small, useful and reasonable for data object.
The remaining of the paper is arranged as follows. Section II revisits the general idea of associative classification. Section III presents the algorithm for generating positive and negative class association and building a new classifier. The experimental results on classification accuracy and the performance study on efficiency and scalability are reported in Section IV. The paper is concluded in Section V.

II. ASSOCIATIVE CLASSIFICATIONS

This paper assumes that the training dataset is a normal relational table, which consists of N data objects described by L distinct attributes. These N data objects have been classified into q known classes. Attributes can be categorical or continuous. For a categorical attribute, we assume that all the possible values are mapped to a set of consecutive positive integers. For a continuous attribute, we assume that its value range is discretized into intervals, and the intervals are also mapped to consecutive positive integers. So, we treat all the attributes uniformly in this study[4].

Let D be the training dataset. In the training dataset, let I be the set of all items in D, that means every data object has some attributes following the form I={i1, i2, ..., iL} and there exists a class label associated with it. Let C={c1, c2, ..., cn} be a set of class labels. We say that a data object d ∈D contains X ⊆I, a subset of items(called itemset). A class association rule (CAR) is an implication of the form X →ci, where X ⊆I, and ci ⊆C. The number of data objects in D matching X and having class label ci is called the support of the rule X →ci, denoted as sup(X →ci). The ratio of the number of objects matching X and having class label ci versus the total number of objects matching X is called the confidence of the rule X →ci, denoted as conf(X →ci). In general, given a training dataset, the task of classification is to build a classifier from the training dataset such that it can be used to predict class labels of unknown objects with high accuracy[5].

For example, if 80% of customers who have bought apples also buy oranges, i.e., the confidence of rule: apples→oranges is 80%, then we can use the rule to classify future data objects. To avoid noise, a rule is used for classification only if it has enough support. Given a support threshold and a confidence threshold, the method finds the complete set of class-association rules passing the thresholds. When a new object comes, the classifier selects the rule which matches the data object and has the highest confidence and uses it to predict the class label of the new object.

In the training dataset, there also exists other class association rules: X →¬ci, ¬X →ci, and ¬X →¬ci. The rule X →¬ci means the data objects which have itemset X do not have the label ci. The rule ¬X →ci means the data objects which do not have itemset X have the label ci. The rule ¬X →¬ci means the data objects which do not have itemset X do not have the label ci. These rules can be called negative class association rules. The rule X →ci can be called positive class association rule. The support and confidence of negative class association rules can be defined as the support and confidence of positive class association rule[6].

III. ALGORITHM BASED ON POSITIVE AND NEGATIVE CLASS ASSOCIATION RULES

The algorithm in this paper consists of two phases: rule generating and classification.

In the first phase: the algorithm computes the complete set of positive and negative class association rules such that sup(R) and conf(R) pass the given support and confidence thresholds, respectively. Furthermore, the algorithm prunes some contradictory rules and only selects a subset of high quality rules for classification.

In the second phase: classification, for a given data object, the algorithm extracts a subset of rules fund in the first phase matching the data object and predicts the class label of the data object by analyzing this subset of rules.

A. Generating Rules

To find rules for classification, the algorithm first mines the training dataset to find the complete set of rules passing certain support and confidence thresholds. This is a typical frequent pattern or association rule mining task. The algorithm adopts Apriori method to find frequent itemset. Apriori method is a frequent itemset mining algorithm which is fast[7]. The algorithm also uses the correlation between itemsets to find positive and negative class association rules[8]. The correlation between itemsets can be defined as:

\[
\text{corr}(X, Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X) \text{sup}(Y)}
\]

X and Y are itemsets. When \(\text{corr}(X, Y)>1\), X and Y have positive correlation. When \(\text{corr}(X, Y)=1\), X and Y are independent. When \(\text{corr}(X, Y)<1\), X and Y have negative correlation. Also when \(\text{corr}(X, Y)>1\), we can deduce that \(\text{corr}(X, ¬Y)<1\) and \(\text{corr}(¬X, Y)<1\).

So, we can use the correlation between itemset X and class label ci to judge the class association rules.

When \(\text{corr}(X, ci)>1\), we can deduce that there exists the positive class association rule X→ci.

When \(\text{corr}(X, ci)>1\), we can deduce that there exists the negative class association rule X→¬ci[9].

So, the first step is to generate all the frequent itemsets by making multiple passes over the data. In the first pass, it counts the support of individual itemsets and determines whether it is frequent. In each subsequent pass, it starts with the seed set of itemsets found to be frequent in the previous pass. It uses this seed set to generate new possibly frequent itemsets, called candidate itemsets. The actual supports for these candidate itemsets are calculated during the pass over
the data. At the end of the pass, it determines which of the
candidate itemsets are actually frequent[10].

The algorithm of generating frequent itemsets is shown as follow:

Algorithm 3.1
Input: tranining dataset T, min_sup
Output: frequent itemsets F
(1)P1=InitPass(T);
(2)F1={f | f ∈ P1, sup(f) >= minsup};
(3)for(k=2;Fk-1!=NULL); k++)
{Pk=CandidateGen(Fk-1);
for (each t ∈ T) {
for (each candidate p ∈ Pk) {
if (p is contained in t) (the number of p)++;
}
Fk={p ∈ Pk | sup(p)=minsup};
}
(4)Return F= Fk;

In this algorithm, there is an important function
CandidateGen() which generates k-itemsets based on
Fk-1.

The code of it is shown as follow:

Algorithm 3.2
Input: Fk-1
Output: Pk
(1)Pk=NULL;
(2)for(all f1, f2 ∈ Fk-1, f1={i1, i2, ... ik-2, ik-1}, f2={i1, i2, ... ik-2, jk-1), and ik-1< jk-1) {
p={i1, i2, ... ik-2, jk-1};
Pk=Pk∪{p};
For(each (k-1)-subset s of p) {
if (s! ∈ Pk-1) delete p from Pk;
}
}
(3)return Pk;

Then, the next step is to generate positive and negative
class association rules. It firstly finds the rules contained in
F which satisfy min_sup and min_conf threshold. Then, it
will determined the rules whether belong to the set of
positive class correlation rules P_AR or the set of negative
class correlation rules N_AR.

The algorithm of generating positive and negative class
association rules is shown as follow:

Algorithm 3.3
Input: training dataset T, min_sup, min_conf
Output: P_AR, N_AR
(1)P_AR=NULL, N_AR=NULL;
(2)for (any frequent itemset X in F and c_i in C) {
if (sup(X→c_i)>min_sup and conf(X→c_i)>min_conf) {
if (corr(X, c_i)>1) {
P_AR= P_AR ∪ {X→c_i};
} else if corr(X, c_i)<1 {
N_AR= N_AR ∪ {X→c_i};
}
}
(3) return P_AR and N_AR;

In this algorithm, we use Apriori method generates the
set of frequent itemsets F. In F, there are some itemsets
passing certain support and confidence thresholds. And the
correlation between itemsets and class labels is used as an
important criterion to judge whether or not the correlation
rule is positive. Lastly, P_AR and N_AR are returned.

B. Classification

After P_AR and N_AR are selected for classification, the
algorithm is ready to classify new objects. Given a new data
object, the algorithm collects the subset of rules matching
the new object. In this section, we discuss how to determine
the class label based on the subset of rules.

First, the algorithm finds all the rules matching the new
object, generates PL set which includes all the positive rules
from P_AR and sorts the itemset by descending support
values. The algorithm also generates NL set which includes
all the negative rules from N_AR and sort the itemset by
descending support values. Second, the algorithm will
compare the positive rules in PL with the negative rules in
NL and decides the class label of the data object.

The algorithm of classification is shown as follow:

Algorithm 3.4
Input: data object, P_AR, N_AR
Output: the class label of data object Cd
(1) PL=Sort(P_AR); NL=Sort(N_AR); i=j=1;
(2)p_rule=GetElem(PL, i); n_rule=GetElem(NL, j);
(3)while i<=PL_Length and j<=NL_Length {
if(RuleCompare(p_rule, n_rule)) {
if(p_rule>n_rule) {
Cd = the label of p_rule;
Break;
}
if(p_rule=n_rule) {
Cd = the label of p_rule;
break;
}
if(p_rule<n_rule) {
j++;
}
IV. EXPERIMENTAL RESULTS

To evaluate the accuracy and efficiency of the algorithm, in this section, we report our experimental results on comparing the algorithm against the popular classification method: CBA.

All the experiments are performed on a 2.2GHz Core PC with 1G main memory, running Microsoft Windows Server 2003. CBA was implemented by its authors, respectively. We choose a training dataset including 1000 objects which have 12 attributes and 7 class labels.

In our experiments, \( min\_conf \) is set to 50%. For \( min\_sup \), it is more complex. \( min\_sup \) has a strong effect on the quality of the classifier produced. If \( min\_sup \) is set too high, those possible rules that cannot satisfy \( min\_sup \) but with high confidences will not be included, and also the rules may fail to cover all the training cases. In the experiments reported before, we set \( min\_sup \) to 6%.

The results are shown in Table I.

As can be seen from the table, the algorithm outperforms CBA on recall ratio and precision ratio. It is clear from these objects that our algorithm produces more accurate classifiers. Our Recall Ratio and Precision Ratio is higher than CBA. So, it shows that the algorithm outperforms CBA in terms of average accuracy and efficiency.

There are two important parameters, database coverage threshold and confidence difference threshold. As discussed before, these two thresholds control the number of rules selected for classification.

In general, if the set of rules is too small, some effective rules may be missed. On the other hand, if the rule set is too large, the training data set may be over fit. Thus, we need to test the sensitivities of the two thresholds for classification accuracy.

According to our experimental results, there seems no way to pre-determine the best threshold values. Fortunately, both curves are quite plain. That means the accuracy is not very sensitive to the two thresholds values.

V. CONCLUSIONS

In this paper, we examined a number of problems that exist in current classification techniques. A new algorithm is presented to generate all positive and negative class association rules and to build an accurate classifier. The method has several distinguished features: (1) its classification is performed based on positive and negative class association rules, which leads to better overall classification accuracy; (2) it prunes contradictory positive and negative class association rules effectively based on correlation between itemsets. Our experiment shows that the algorithm is highly effective at classification and has better average classification accuracy and efficiency in comparison with CBA. In our future work, we will focus on building more accurate classifiers by using more sophisticated techniques and mining more useful positive and negative class association rules.

REFERENCES


