Volume: 02 Issue: 01 Pages: 470-474 (2010)

Combinatorial Approach of Associative Classification

P. R. Pal
Department of Computer Applications, Shri Vaishnav Institute of Management, Indore, M.P.
E-mail: prpal@rediffmail.com

R.C. Jain
Department of Computer Applications, Samrat Ashok Technological Institute, Vidisha, M.P.
E-mail: dr.jain.rc@gmail.com

ABSTRACT

Association rule mining and classification are two important techniques of data mining in knowledge discovery process. Integration of these two has produced class association rule mining or associative classification techniques, which in many cases have shown better classification accuracy than conventional classifiers. Motivated by this study we have explored and applied the combinatorial mathematics in class association rule mining in this paper. Our algorithm is based on producing combinations of itemsets including classes using combinatorial mathematics and subsequently finds an associative classifier. In our approach, experimental results have given accuracy, near to other popular classification methods.

Keywords: Associative classification, combinatorial mathematics, Data mining, Knowledge discovery.

Date of Submission: February 23, 2010 Date of Acceptance: April 28, 2010

1. Introduction

Data mining algorithms have well taken up challenges for data analysis in large database. Association rule mining [1, 2, 3] is one of the key data-mining tasks in which associability of the items is discovered in a training database. Classification [4, 5] is another data mining tasks. The objective of classification is to build a model in training dataset to predict the class of future objects whose class label is not known.

The idea of using association rule mining in classification rule mining was first introduced in 1997 by [4] and [6] and it was named as class association rule mining or associative classification. The first classifier based on association rules was CBA [7] given by Liu et al. in 1998. Later, some improved classifiers were given by Li et al. CMAR [8] in 2001, Yin et al. CPAR [9] in 2003, and Fadi et al. MCAR [10] in 2005. More research is going on to design even improved classifiers.

Class association rule mining process can be decomposed in three parts. First we find frequent itemsets and frequent class association rules. The provided support threshold value is used to remove the uninterested elements. Second we find the strong class association rules. Confidence threshold value helps to accomplish this task and prune the weak rules. Third only a subset of selected class association rules is used to design a classifier and rest of the class association rules are removed. Various methods [4, 6, 7, 8, 9, 11, 12, 13, 14, 15] are common to accomplish the class association rule mining process.

In this paper we propose an algorithm CAAC (Combinatorial Approach of Associative Classification). CAAC is based on the concept of combinatorial mathematics. It works in two steps:

- CAAC_RG_CB (CAAC Rule Generator and Classifier Builder): in this step a strong class association rule set is generated and a classifier is produced.
- CAAC_CP (CAAC Class Predictor): this step predicts the class of objects whose class label is not known and calculates the accuracy of classifier.

2. Class association Rule mining

The idea of class association rule mining is as follows. We have given a training database where each transaction contains all features of an object in addition to the class label of that object. We can derive the association rules to always have a class label as consequent i.e. the problem states of finding a subset of an association rule set of the $X \Rightarrow C$, where X is association of some or all object features and C is class label of that object.

Class association rule mining is a special case of association rule mining and associative classification finds a subset of class association rule set to predict the class of previously unseen data (test data) as accurate as possible with minimum efforts. This subset of class association rule set is called associative classifier or simply a classifier.

Let we illustrate the class association rule mining with the training data shown in table 1. It consists three attributes X (X1, X2, X3), Y (Y1, Y2, Y3), Z (Z1, Z2, Z3) and two class labels (C1, C2). We assume the min_sup = 30% and min_conf = 70%. Table 2 shows the strong class association rules along with their support and confidence. The table 2 also represents a classifier as the rules are sorted according to confidence they hold.

Table 1. Training database

TID	X	Y	Z	Class
1.	X2	Y2	Z1	C1
2.	X1	Y2	Z2	C2
3.	X1	Y3	Z3	C2
4.	X3	Y1	Z2	C1
5.	X1	Y1	Z3	C2
6.	X2	Y3	Z1	C1
7.	X3	Y3	Z2	C1
8.	X1	Y1	Z1	C1
9.	X2	Y3	Z1	C1
10.	X1	Y1	Z1	C2

Table 2. Strong class association rule set

Class association rule		Support	Confidence
Antecedent	Consequent		
X2	C1	3/10	3/3
Y3	C1	3/10	3/3
X2Z1	C1	3/10	3/3
X1	C2	4/10	4/5
Z1	C1	4/10	4/5

3. COMBINATORIAL MATHEMATICS

A combination is an unordered collection of unique sizes. An ordered collection is called permutations. Given S, the set of all possible unique elements, a combination is a subset of the elements of S. The order of elements is not considered in combinations. Two or more subsets with same elements in different orders are considered as one combination e.g. ab and ba represents two different permutations but only one combination. Also elements cannot be repeated in a combination. Every element appears uniquely once; this is because the combinations are defined by the set of elements contained by them in unordered manner e.g. aba is not a legal combination.

A k_combination is a subset with k elements. The number of k_combinations each of size k from a set S with n elements is the binomial coefficient and represented by:

$$^{n}C_{k} = \frac{n!}{k!\,(n-k)!}$$

k_combination is also defined as the k elements taken at a time out of n elements.

The sum of all the possible combinations of a set S with n elements can be calculated by adding all the 0_combinations, 1_combinations, 2_combinations, up to n_combinations. Sum of all the combinations is equal to 2^n. It can be represented as follows:

$$^{n}C_{0} + {}^{n}C_{1} + {}^{n}C_{2} + \ldots\ldots + {}^{n}C_{n} = 2^{n}$$

For example, set S has 3 elements i.e. S = (a, b, c). The set of all possible combinations of S is C = (ϕ, a, b, c, ab, bc, ac, abc), i.e. there are total 8 combinations, which is 2^3.

The above discussion finds the number of combinations taking k elements at a time out of n elements. It also finds the total number of all the possible combinations for which k will vary from 0 to n, is 2^n.

The elements in each combination of a set S with n unique elements can be found as follows: Generate 2^n unique binary patterns. Each binary pattern will consist an n digits binary string of 0 and 1. Here each digit of the binary pattern corresponds to a unique element of the set S i.e. 1st digit of binary pattern corresponds to 1st element of S, 2nd digit of binary pattern corresponds to 2nd element of S and so on up to nth digit of the binary pattern. Here, a binary pattern represents a combination and each 0 in a binary pattern shows the absence of corresponding element and each 1 shows the presence of the corresponding element in that combination. Therefore, in each binary pattern, the elements having corresponding binary digits 1 are combined to form the subset of elements in that combination because each 1 in the binary pattern represents that corresponding element to be included in the combination. In such a way we will find a subset of elements in each combination. It will produce total 2^n subsets (each subset will represent a combination) that will represent the set of all combinations.

For example, let S = (a, b, c) here n is 3. The total numbers of combinations are $2^3 = 8$. The unique binary patterns for n = 3 can be represented as:

B = (000, 001, 010, 011, 100, 101, 110, 111)

It gives: $C_0 = \phi$, $C_1 = c$, $C_2 = b$, $C_3 = bc$, $C_4 = c$, $C_5 = ac$,

$C_6 = ab$, $C_7 = abc$.

Now C = (ϕ, a, b, c, ab, bc, ac, abc).

Here C is containing all the possible subsets of combinations for set S.

Combinatorial study tells about the number of combinations ($^{n}C_{k}$) to be generated, but it doesn't tell any thing that how the subsets of these combinations will be generated? In this section we have explored the systematic method that generates the subsets of these combinations for a set

4. COMBINATORIAL APPROACH OF ASSOCIATIVE CLASSIFICATION

The proposed algorithm is CAAC (Combinatorial Approach of Associative Classification). It consists of two parts:

- CAAC_RG_CB (CAAC Rule Generator and Classifier Builder)
- CAAC_CP (CAAC Class Predictor)

Volume: 02 Issue: 01 Pages: 470-474 (2010)

Algorithm CAAC_RG_CB

```
Input:  training dataset trn in bitmap, minimum support (min_supp), minimum confidence (min_conf),
        number of items (n), number of attributes (NAttr), number of values in each attributes (NVAttr)

Output: classification model (classifier)
begin
a=2^(NVAttr(NAttr));
b=(2^(n-1))+(2^(n-NVAttr(1)));
for i=a:b
   comb=dec2bin(i,n); // generate the n bit binary string (combination)
   f = CombValidity(comb,NAttr,NVAttr); // check the validity of combination
   if (f==1)
      mat = Comb2Mat(comb, n); // convert the combination into the binary matrix
      sup = CombSup(mat, trn, rows, cols); // calculate the support of combination
      if (sup >= min_sup)
         L= [L; mat, sup]; // store large combination along with its support
      end if
   end if
end for
//here we find the strong class association rules set
ILen=n-NVAttr(NAttr); //itemset length excluding class length
SCARSet = StrongCARSet(L, min_conf, ILen, n); //get strong class association rules set
// here we select a subset of strong class association rules i.e. build a classifier
classifier = ClassifierRuleSet(SCARSet, n); //select a subset of strong class association rules set
end
```

Figure 1: CAAC_RG_CB algorithm

Algorithm CAAC_CP

```
Input:  test dataset tst in bitmap form, classifier (produced by CAAC_RG_CB)
        number of items (n), number of attributes (NAttr), number of values in each attributes (NVAttr)
Output: accuracy of the classifier
begin
objLen=n-NVAttr(NAttr);
[tstR tstC]=size(tst);
true=0;
[classifierR, classifierC]=size(classifier);
for t=1:tstR
   obj=tst(t,1:objLen); // get the object from the test database
   [r, c]=size(obj);
   // here we match the attributes of the object with each rule of the classifier &  derive a matrix
   // ClassMat; Last column of ClassMat stores the maximum number of attributes of the object matched
   // with that rule.
   [ClassMat, mc ] = MatchObjAttr(classifier, obj);
   // here we derive a matrix ClassFreq; it contains the frequency of each class that an object has been
   // classified by the classifier with maximum number of attributes and finally we find the class index of
   // the object
   [ClassFreq] = ObjClassFreq(ClassMat, mc); // get the class index of the object
   // here we get the number of objects correctly classified by the classifier
   for y=class1:classN
      if ((tst(t,y)==1) & (y==ClassIndex))
            true=true+1;
      end if
   end for
end for
acc = (true*100)/tstR; // here the accuracy of the classifier is calculated
end
```

Figure 2: CAAC_CP algorithm

The pseudo codes for CAAC_RG_CB and CAAC_CP are presented in figure 1 and figure 2 respectively.

CAAC_RG_CB: This algorithm performs following key tasks.

First, it generates binary combination of items including classes. Combinatorial mathematics is used to produce the combination. Algorithm prunes the invalid combinations and only possible valid combinations are processed to find all frequent binary combinations in bitmap training database.

Next, it finds strong class association rules set using confidence threshold (min_conf); and finally it produces a classifier (a subset of strong class association rules set) by eliminating all small (consisting less items) strong rules that are contained by large (consisting more items) strong rules.

CAAC_CP: This algorithm performs the following tasks.

First, it gets object from test database. Then we match the attributes of the object with each rule of the classifier & derive a matrix called ClassMat. Last column of ClassMat stores the maximum number of attributes of the object matched with that rule.

Matrix ClassFreq contains the frequency of each class that the classifier with maximum number of attributes has classified an object. The class index with maximum value is the class of that object. Finally it gets the number of correctly classified objects (from training database) by the classifier, subsequently calculates the accuracy of the classifier.

5. EXPERIMENTAL RESULTS

To evaluate the accuracy of our classifier (CAAC), we choose the dataset from UCI machine learning repository [16] reported in table 3.

Table 3: Dataset

Dataset Name	No of Attributes	No of Items	No of Classes	No of Records
Tic-tac-tow	9	27	2	958

We have set the min_sup to 1% and min_conf to 50% for all experiments as these parameters with same values are also considered by CBA and CMAR and are reported to produce the best accuracy. We have performed all the experiments on 1.7 GHz Celeron PC with 256 MB main memory.

We have chosen randomly 90% objects for training the classifier and remaining 10% for testing of the classifier. We have taken 10 such observations so that training and testing can be performed thoroughly in the dataset. Accuracy encountered in experimental results is shown in the table 4.

For Tic-tac data set popular classification methods C4.5, RIPPER, CBA, CMAR and CPAR gives 99.4%, 98.0%, 99.6%, 99.2%, and 98.6% accuracy respectively, Yin and Han, 2003 (6). The accuracy (in table 4) obtained by we people in our experiments is nearing to these results that confirms our approach.

Table 4: Accuracy encountered in experimental results

Observation No.	Tic-tac-tow	
	Time (sec.)	Accuracy (%)
1.	77.78	98.96
2.	76.06	100.00
3.	77.31	98.96
4.	77.10	100.00
5.	78.18	100.00
6.	73.45	98.96
7.	75.26	98.96
8.	77.35	98.96
9.	74.14	98.96
10	72.17	98.96
Average	75.88	99.27

6. CONCLUSION

In this study, we have proposed a new approach CAAC (Combinatorial Approach of Associative Classification). CAAC exploits the combinatorial technique to generate the class association rule set from the training dataset and subsequently forms an associative classifier.

Our present study on tic-tac dataset of UCI machine learning database repository show that the accuracy of our technique is near to other popular associative classification methods like C4.5, RIPPER, CBA, CMAR and CPAR. It shows the significance of our approach to produce the associative classifiers for better results in future.

REFERENCES

[1]. R. Agrawal and R. Srikant, “Fast Algorithm for Mining Association Rules”, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB-94), Morgan Kaufman Publishers, Santiago de Chile, Chile, September 1994, pp. 487-499.

[2]. P. R. Pal, R. C. Jain, CAARMSAD: “Combinatorial Approach of Association Rule Mining for Sparsely Associated Databases”. Journal of Computer Science, Tamilnadu India, Vol. 2, No 5, pp 717, July 2008.

[3]. J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation”, proceedings of international conference on management of data (ACM SIGMOD’00), pp 1-12, Dallas, TX, May 2000.

[4]. R. Bayardo, “Brute-force mining of high-confidence classification rules”, Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97), AAAI Press, Newport Beach, CA, United States, August 1997, pp. 123-126.

Volume: 02 Issue: 01 Pages: 470-474 (2010)

[5]. J. Quinlan, C4.5, “Programs for machine learning”, San Mateo, CA: Morgan Kaufmann, 1993.

[6]. K. Ali, S. Manganaris, and R. Srikant, “Partial Classification using Association Rules”, Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97), AAAI Press, Newport Beach, CA, United States, August 1997, pp. 115-118.

[7]. B. Liu, W. Hsu, and Y. Ma, “Integrating Classification and Association Rule Mining”, Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), AAAI Press, New York City, NY, United States, 1998, pp. 80-86.

[8]. W. Li, J. Han, and J. Pei, “CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules”, In Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM-01), IEEE Computer Society, San Jose, CA, United States, 2001, pp. 369-376.

[9]. X. Yin and J. Han, “CPAR: Classification based on Predictive Association Rules”, Proceedings of the Third SIAM International Conference on Data Mining (SDM-03), SIAM, San Francisco, CA, United States, 2003, pp. 331-335.

[10]. F. Thabtah, P. Cowling, and Y. Peng, “MCAR: Multi-class classification based on association rule approach”, Proceedings of 3rd IEEE International Conference on Computer System and Applications Cairo, Egypt, 2005, pp. 1-7.

[11]. F. Coenen and P. Leng, “An Evaluation of Approaches to Classification Rule Selection”, Proceedings of the 4th IEEE International Conference on Data Mining (ICDM-04), IEEE Computer Society, Brighton, United Kingdom, November 2004, pp. 359-362.

[12]. Veloso A., Meira W. “Rule Generation and Selection Techniques for Cost Sensitive Associative Classification”, 19th Brazilian Symposium on Software Engineering, 2005.

[13]. J. Wang, G. Karypis, “On Mining Instance-Centric Classification Rules”, IEEE Transactions on Knowledge and Data Engineering, Vol. XX, No. XX, 2006, pp. 1-13.

[14]. Y.J. Wang, Q. Xin, and F. Coenen, “A Novel Rule Ordering Approach in Classification Association Rule Mining”, Proceedings of the 5th International Conference on Machine Learning and Data Mining (MLDM-07), Springer-Verlag, Leipzig, Germany, July 2007, pp. 339-348.

[15]. Y.J. Wang, Q. Xin, and F. Coenen, “A Novel Rule Weighting Approach in Classification Association Rule Mining”, Proceedings of the 7th IEEE International Conference on Data Mining, October 2007, pp. 271-276.

[16]. C. Merz and P. Murphy, “UCI repository of machine learning databases”, Irvine, CA, University of California, Department of Information and Computer Science.

Authors Biography

R. C. Jain, M.Sc., M. Tech., Ph. D., is a Director of S.A.T.I. (Engg. College) Vidisha (M. P.) India. He is Dean, Faculty of Computer and Information Technology, Rajeev Gandhi Technical University Bhopal (M.P.). He has 35 years of teaching experience. He is actively involved in Research with area of interest as Fuzzy Systems, DIP, Mobile Computing, Data Mining and Adhoc Networks. He has published more than 125 research papers, produced 7 Ph. Ds. And 10 Ph. Ds are under progress

P. R. Pal, B. Sc., M.C.A., Ph.D., is a Reader in Department of Computer Applications of Shri Vaishnav Institute of Management Indore (M P) India. He is working on Data Mining Algorithms to improve their efficiency under guidance of Dr. R. C. Jain. His area of interest is DBMS, Data Mining, Computer Graphics and Computer Architecture.