Classification Algorithm based on MS Apriori for Rare Classes

Devashree Rai
Department of Electrical Engineering
National Institute of Technology, Raipur
Chhattisgarh, India

Kesari Verma
Department of Computer Application
National Institute of Technology, Raipur
Chhattisgarh, India

A. S. Thoke
Department of Electrical Engineering
National Institute of Technology, Raipur
Chhattisgarh, India

ABSTRACT

Most of the data mining algorithm focuses on frequent patterns, few algorithm emphases on rare items, but rare items [1] also have importance, for example, network intrusion detection, where among various normal connections we need to detect the rare malicious connections. Classification of such a non-uniform data set is a challenging issue. Most classifiers perform poorly in such a data set. Realizing the importance of rare class classification, in this paper we propose a classification algorithm (CBMR Algorithm) that is based on association rules mined by MSApriori approach [2] and is capable of classifying rare classes. The performance evaluation of the proposed algorithm has been done for different data sets [3] and in comparison with existing technique like [4], it is found that algorithm has efficient and superior performance for classifying rare cases.

General Terms

Rare classes, MSApriori Algorithm, Classification, Data Mining.

1. INTRODUCTION

Data mining [5] is concerned with extracting useful information from large databases. Association rule mining [6] [7] and classification [4] [8] are two important data mining techniques. Association rule mining aims to discover associations among items in databases. This is done by first identifying frequent item sets and then obtaining association rules from these frequent item sets. Association rule mining is an unsupervised learning since it extracts rule without any prior target information whereas classification is supervised learning where rule extraction is done on the basis of pre-determined target (class). Classification rule mining extracts classification rules by using data sets containing set of labeled training examples and the objective is to build a classifier that is capable of classifying "unseen" data records. The two techniques, association rule mining and classification rule mining has been integrated in [4] to build an efficient classifier from association rules. This is done by considering subset of association rules that has class attribute as its consequent.

Real world datasets contains frequent as well as rare items The challenging issue of rare class classification arises in various data mining applications like oil spills detection in satellite radar images [9], identifying fraudulent credit card transaction [10], predicting failures in telecommunication equipments [11] and [12] [13] so on. All this applications has common problem of having target class samples extremely rare and other class samples sufficiently large. Most classifiers perform poorly in such data sets.

Association based classification methods like [4] that employ single minimum support criteria for association rule mining fails to give satisfactory results in classifying rare class. Single minimum support based approaches [6] [7] suffer from "rare item problem" [14] dilemma. If high minimum support value is used, rare item sets fails to satisfy minimum support criteria and thus could not be extracted. If low value is used, item sets explodes. Therefore, to extract frequent item sets involving rare items, an improved approach known as Multiple Support Apriori (MSApriori) has been proposed in [2], that uses multiple supports instead of single. To extract frequent item sets involving rare items, each item is assigned with minimum item support (MIS) value. Then item sets has to satisfy the lowest MIS value among the respective items. The rules generated are then pruned based on confidence value. Efforts are being made in researches to develop improved algorithms based on multiple supports [15] [16].

In this paper we propose a Classification algorithm that is Based on MSApriori algorithm for mining association rules and is capable of classifying Rare classes (CBMR). In this approach each target class is assigned with user specified MIS value and therefore allows giving special importance to rare class. Frequent class will be assigned with relatively higher MIS value and rare class with lower value. This approach is efficient in classifying rare class as well as frequent class. Experimental evaluation of algorithm has been done for different data sets [3] containing rare class and is giving better results in all the cases in comparison with approach like [4].

2. CBMR ALGORITHM

2.1 Overall System

The proposed algorithm (CBMR) works in two phases. In first phase class association rules (CARs) are generated using MSApriori algorithm [2] and in second phase classifier is build up using these class association rules. The two phases of CBMR algorithm are presented below.

2.2 CARs Generation

2.2.1 Basic Concepts

This phase generates rules of the form $X \rightarrow Y$, where X is a set of items and Y is a class label. Support of rule is calculated as follows:

$$Support = \frac{ruleCount}{|D|} * 100\%$$

Where *ruleCount* is number of cases in data set D that contains X and are labeled with class Y, |D| is size of data set. Confidence of rule is calculated as:

$$Confidence = \frac{ruleCount}{X.Count} * 100\%$$

Where, *X.Count* is the number of cases in D that contains X.

2.2.2 Rule Generation Phase

This phase begins with assigning MIS value to each item in a data set. Target classes are assigned with user specified MIS values and for every other items i_j, $MIS(i_j)$ is calculated as per equation 1.

$$MIS(i_j) = \beta \times S(i_j), \text{ if } \beta \times S(i_j) \succ LS \qquad (1)$$

$$= LS \quad \text{else}$$

Where, LS corresponds to user-specified least support value. β is user specified proportional value that can vary between 0 and 1. $S(i_j)$ refers to support of an item equal to (i_j .Count)/|D|. After assignment of MIS values to class and all other items, CARs generation process proceeds with following steps:

Let L_K denote set of large K-item sets, C_k denote set of candidates K-item sets. (For simplicity, we will interchangeably use the terms support and count).

```
1)  Sort all items in ascending order of their MIS values and insert into Q.
2)  G= initial-pass(Q,D)
3)  $L_1 = \{< g > | g \in G, g.Count \geq MIS(g)\}$
4)  $CAR_1$= generate-CRs($L_1$)
5)  $for(k = 2; L_{k-1} \neq \phi; k++)$
6)  If k=2 then $C_2$= level2CandidateGen(G)
7)  Else $C_k$=Candidate-Gen($L_{k-1}$)
8)  End
9)  For each record d∊ D do
10) $C_d$ = Subset($C_k$,d)
11) For each candidate c ∊ $C_d$ do
12) c. Count++;
13) If d.class == c.class then c.ruleCount++
14) End
15) End
16) $L_k$={c ∊ $C_k$| c.Count ≥ MIS(c[1])}
17) $CAR_k$ = generate-CRs($L_k$)
18) End
19) $CAR_k = \cup_k CAR_k$
```

Line 1 sort the items in ascending order of their MIS values and insert sorted items in Q. Line 2 calls the function initial pass which performs following function:

a) Finds actual count of each item in Q.
b) Finds first item *i* in Q, such that i.Count ≤ MIS (i), *i* is inserted into G.
c) For each subsequent item j after I, if j.Count ≥ MIS (i), j is inserted into G.

Line 3 generates frequent 1-item set L_1. Function generate-CRs() is called in line 4 that generates rules if the rule's confidence is greater than or equal to user specified confidence value. Loop starts at line 5 which iteratively generate candidates [2] line 6, 7, then each record in the dataset D is checked to contain candidate items by calling subset () in line 10. For each candidate that is a subset its count value is incremented in line 12. If the candidate's class and record class matches its corresponding rule count is incremented in line 13. Line 16 finds large k-item sets if candidate's count value is greater than lowest MIS value amongst its items, which in this case will be the first item since items are sorted in ascending order of their MIS values. In line 17 generate-CRs is called to generate CARs for all large k-items. The algorithm finally returns CARs that will be processed in next phase to generate classifier.

2.3 Building a classifier

To build a classifier from the generated CARs we follow same approach as in [4]. First step is to impose ordering on generated rules, ordering is done according to following criteria:

For any two rules, r1 and r2, r1>r2 that is, r1 has higher precedence than r2, if,

a) Confidence (r1) > confidence (r2).
b) Confidence (r1) == confidence (r2) and support (r1) > support (r2).
c) Confidence (r1) == confidence (r2), support (r1) == support (r2) and r1 is generated earlier than r2.

The classifier is build up in four stages that we discuss in sections below:

2.3.1 Stage 1: C and W rule identification.

For each record in the data set D, identify first rule in the ordered list that correctly classifies the record (cRule) and the first rule that wrongly classifies the record (wRule). The identified C and W rule are than compared.

a) If no cRule found, do nothing.
b) If cRule exists but no wRule exist, then mark cRule as "Strong" cRule to indicate that it classifies a record correctly.
c) If both cRule and wRule exists, and cRule has greater precedence than wRule, then mark cRule as "strong" cRule.
d) If both cRule and wRule exists, and wRule has greater precedence than cRule, create a structure say A of the form <ID, Y, cRule, wRule>, where ID is unique identification number of particular record in D, Y is the class of record with the given ID, cRule and wRule are associated cRule and wRule.

On completion of this stage all records that are wrongly classified but for which there exists corresponding cRule are stored as structure A ready for further consideration and all cRule that are "strong" with respect to at least one record will be identified. For each cRule, we also keep track of number of cases it covers of each class in the field classCasesCovered.

2.3.2 Stage 2: Process wrongly classified records.

Process the list of records that has been wrongly classified. If wRule associated with record correctly classifies at least one other record, update claassCasesCovered for both wRule and corresponding cRule. If wRule is not a cRule for any record, than find all the rules that wrongly classify record and have higher precedence than its corresponding cRule (For this we only have to consider rules that are cRule for any record), and place the result in rules "Replace" list. These rules are all those rules that we would like to remove, if possible, so that the corresponding record will be correctly classified.

2.3.3 Stage 3: Process rule list

In this stage rule list is processed to identify, for each "strong" cRule, the default class and total error count. We find the number of records in training data set that corresponds to individual classes. Next for each "strong" cRule replace list is processed. If the ID case in the training data set has been wrongly classified by a strong cRule with higher precedence than cRule will not replace rule in rule list otherwise it is replaced to cover the case. ClassCasesCovered field is updated accordingly (for more details refer [4]).

Then the default class is identified and the total error count is calculated.

2.3.4 Stage 4: Generate classifier

The classifier is generated by identifying first “strong” cRule that is, the rule that satisfies at least one record, with the lowest total error that is the cutoff rule. The final classifier will consist of all the rules up to and including the identified rule. As all the rules after this rule only produce more error they can be discarded. The final classifier will also consist of default rule that produces the default class associated with the identified rule.

3. EXPERIMENTAL EVALUATION

The proposed algorithm is implemented in java 1.7.0, windows 7 operating system and i5 core processor. Below we present the comparative classification result of CBMR (proposed algorithm) and LUCS KDD [17] implementation of CBA algorithm. These results are obtained by experimenting with different data set from [3].

3.1 Data set: Car

- Number of records: 1728
- Classes: 4 (<22>,<23>,<24>,<25>)
- Minimum support of CBA (corresponding LS of CBMR): 172 records. (For simplicity considering count as support).
- MIS (<22>, <23>, <24>, <25>): (172, 10, 5, 5).

```
Algorithm:  CBA
Accuracy: 80.79
Rules generated:
<7> → <22>
<1> → <22>
<14>→ <22>
<10> → <22>
Default → <23>
```

Fig 1: Output of CBA Algorithm for Car data set.

```
Algorithm: CBMR
Accuracy: 81.59
Rules generated:
<19> → <22>
<13> →<22>
<4 18 21 15> → <25>
<3 18 20 15> → <23>
<8 3 14 21> → <25>
<3 18 21 15> → <25>
<5> → <22>
<1> → <22>
Default → <23>
```

Fig 2: Output of CBMR Algorithm for Car data set.

3.2 Data set: Glass

- Number of records: 214
- Classes:7(<42>,<43>,<44>,<45>,<46>,<47>, 48>)
- Minimum support of CBA (corresponding LS of CBMR): 10 records (for simplicity considering count as support).
- MIS (<42><43><44> <45> <46> <47> <48>): (7, 12, 3, 3, 3, 3, 12).

```
Algorithm:  CBA
Accuracy: 57.94
Rules generated:
<16 24> → <48>
<8 24> → <48>
<3 5 25>→ <42>
<3 6 25> → <42>
<9 10 13> → <43>
<15 25> → <42>
<3 7 15> → <42>
<14 24> → <48>
<2 7 12> → <43>
<4 7 12> → <43>
<4 9 13> → <43>
<6 23> → <42>
<3 6 21> → <42>
<2 5 12> → <43>
<7 10 12> → <43>
<4 12 13> → <43>
<27 31> → <43>
<4 7 15> → <43>
Default → <42>
```

Fig 3: Output of CBA Algorithm for Glass data set.

```
Algorithm: CBMR
Accuracy: 63.55
Rules generated:
<36 9> → <48>
<6 30 3> →<42>
<6 16 3> → <42>
<33 17> → <46>
<24 17> → <47>
<9 7 38> → <46>
<2 30 28 12> → <44>
<36 8> → <48>
<29 2 12 38> → <43>
<32 16> → <42>
<19 16 3> → <42>
<21 3 12 38> → <42>
<29 28 38 34> → <43>
<17 29 7> → <43>
<17 2 28> → <43>
<17 12 38> → <43>
<2 30 12> → <44>
<30 28 3 12 38> → <42>
Default → <43>
```

Fig 4: Output of CBMR Algorithm for Glass data set.

3.3 Data set: Heart

- Number of records: 303
- Classes:5(<48>,<49>,<50>,<51>,<52>)
- Minimum support of CBA (corresponding LS of CBMR): 45 records (for simplicity considering count as support).
- MIS (<48>, <49>, <50>, <51>, <52>): (45, 30, 25, 3, 26).

```
Algorithm:  CBA
Accuracy: 58.94
Rules generated:
<7 8 18> → <48>
<7 10 18> → <48>
<8 10 11>→ <48>
<8 10 18> → <48>
<1 11 19> → <48>
<5 10 18> → <48>
<9 10 14> → <48>
<8 10 19> → <48>
<7 11 14> → <48>
<7 9 10> → <48>
<9 19> → <48>
<7 9 14> → <48>
<5 9 14> → <48>
<7 9 11> → <48>
<7 8 9> → <48>
<1 7 9> → <48>
Default → <49>
```

Fig 5: Output of CBA Algorithm for Heart data set.

```
Algorithm: CBMR
Accuracy: 58.27
Rules generated:
<5 47 10 16> → <51>
<47 10 39 41 6> →<51>
<45 41 31 30> → <48>
<45 31 16 30> → <48>
<45 41 31> → <48>
<45 41 33 30> → <48>
<45 1 11> → <48>
<45 31 21 30> → <48>
<41 31 16 30> → <48>
<45 16 33 30> → <48>
<41 31 30> → <48>
<38 31> → <48>
Default → <49>
```

Fig 6: Output of CBMR Algorithm for Heart data set.

3.4 Analysis

By observing the above results it is clear that, CBMR algorithm gives better performance in classifying the rare classes in comparison to CBA algorithm. In figure 1, where CBA algorithm only produces rules pertaining to class 22, the proposed algorithm produces rules for the classes 23 25 as well (figure 2). The CBMR algorithm also gives better accuracy in this case. (Accuracy can increase or decrease depending on the MIS values provided by the user for the classes). For the data set glass (figure 3), CBA algorithm fails to produce any rules corresponding to classes 44, 45, 46, 47. We assigned lower MIS values to these classes in comparison to more frequent classes and we found better results both in rules produced as well as in accuracy (figure 4). Similarly for data set heart, CBA (figure 5) only produces rules corresponding to class 48; in this case if we are particularly interested in the class 51 we can get the desired result by CBMR algorithm (figure 6). The algorithm not only produces rules for frequent classes or rules only for rare classes but for both of them together balanced on the basis of MIS value provided.

4. CONCLUSION

Our work mainly focuses on rare class classification. Since most of the classifier gives poor performance in the case. Realizing the importance of rare classes in many applications we proposed an algorithm CBMR which is capable of classifying rare classes efficiently. We have tested performance of CBMR by experimenting with different data sets [3] and found the desired results. In future the idea can be used to enhance other existing classification algorithms to make them efficient in classifying both frequent and rare classes.

5. REFERENCES

[1] Weiss, G. M. “Mining With Rarity: A Unifying Framework.” SIGKDD Explorations, 2004, Vol. 6, Issue 1, pp. 7 – 19.

[2] Liu, B., Hsu, W., and Ma, Y. “Mining Association Rules with Multiple Minimum Supports.” SIGKDD Explorations, 1999wman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[3] Coenen, F. (2003), The LUCS-KDD Discretized/normalised ARM and CARM data library http://www.csc.liv.ac.uk/~frans /KDD/Software /LUCS_KDD_DN/, Department of Computer Science, The University of Liverpool, UK.

[4] B. Liu, W. Hsu and Y. Ma, “Integrating classification and association rule mining”, proceedings of the fourth international conference on knowledge discovery and data mining, 1998, pp. 80-86.

[5] M.S. Chen, J. Han, P.S. Yu, “Data mining: an overview from a database perspective”, *IEEE Transactions on Knowledge and Data Engineering*, 1996, 8, pp. 866-883.

[6] Agrawal, R., Imielinski, T., and Swami, A. “Mining association rules between sets of items in large databases.” SIGMOD, 1993, pp. 207-216.

[7] Agrawal, R., and Srikanth, R. “Fast algorithms for mining association rules.” VLDB, 1994.

[8] W. Li, J. Han and J. Pei, “CMAR: Accurate and efficient classification based on multiple class-association Rules”, In ICDM'01, San Jose, CA, Nov.2001, pp. 369-376.

[9] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. Machine Learning, 30(2):195-215, 1998.

[10] P. K. Chan, and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164-168, 2001.

[11] G. M. Weiss, and H. Hirsh. Learning to predict rare events in event sequences. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 359-363, 1998.

[12] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia.Event detection and analysis from video streams. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(8): 873-889, 2001.

[13] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. Proc. CVPR’04, Washington, DC, 2004, Vol.2, pp.819-826.

[14] Mannila, H. “Methods and Problems in Data Mining.” ICDT, 1997.

[15] R. Kiran and P. Reddy “An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules” IEEE 2009.

[16] I. Kouris, C. Makris, A. Tsakalidis”An improved algorithm for mining association rules using multiple support values” FLAIRS 2003.

[17] Coenen, F. (2004). LUCS KDD implementation of CBA (Classification Based on associations). http://www.csc.liv.ac.uk/~frans/KDD/Software/CMAR/cba.html, Department of Computer Science, The University of Liverpool, UK.