Expert Systems with Applications 42 (2015) 4361–4369

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An improved algorithm for mining class association rules using the difference of Obidsets ☆

Loan T.T. Nguyen [a,c], Ngoc Thanh Nguyen [b,c,*]

[a] Faculty of Information Technology, VOV College, HCM City, Viet Nam
[b] Division of Knowledge and System Engineering for ICT, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
[c] Department of Information Systems, Faculty of Computer Science and Management, Wroclaw University of Technology, Poland

ARTICLE INFO

Article history:
Available online 13 January 2015

Keywords:
Classification
Class association rules
Data mining
Difference of Obidsets
Tree structure

ABSTRACT

Class association rules play an important role in decision support systems and have thus been extensively studied. Recently, an efficient algorithm for mining class association rules, named CAR-Miner, has been proposed. It, however, consumes a lot of memory for storing the Obidsets (sets of object identifiers that contain itemsets) of itemsets and requires a lot of time to compute the intersection between two Obidsets, especially in the large datasets. This paper proposes an improved algorithm for mining class association rules that uses the difference between two Obidsets (d2O) to save memory usage and run time. Firstly, the d2O concept is developed. A strategy for reducing the storage space and computation time of d2O is then derived. Experimental results show that the proposed algorithm is more efficient than CAR-Miner in terms of run time and memory usage.

1. Introduction

Association rule mining has been extensively studied due to its application in numerous fields such as market basket analysis, medicine, protein sequencing, census data processing, and fraud detection. Many subjects have attracted researchers, including mining association rules (Duong, Tin, & Vo, 2014; Grahne & Zhu, 2005; Lucchese, Orlando, & Perego, 2006; Vo, Hong, & Le, 2012; Vo, Hong, & Le, 2013; Zaki & Hsiao, 2005) and classification based on association rules (Abdelhamid, Ayesh, Thabtah, Ahmadi, & Hadi, 2012; Chien & Chen, 2010; Coenen, Leng, & Zhang, 2007; Li, Han, & Pei, 2001; Lim & Lee, 2010; Liu, Hsu, & Ma, 1998; Liu, Jiang, Liu, & Yang, 2008; Liu, Ma, & Wong, 2000; Nguyen & Vo, 2014; Nguyen, Vo, Hong, & Thanh, 2012; Nguyen, Vo, Hong, & Thanh, 2013; Nguyen, Vo, & Le, 2014; Nguyen, Vo, & Le, 2015; Thabtah, Cowling, & Peng, 2004; Thabtah, Cowling, & Hammoud, 2006; Veloso, Meira, Goncalves, Almeida, & Zaki, 2007; Veloso, Meira, Goncalves, Almeida, & Zaki, 2011; Veloso, Meira, & Zaki, 2006; Vo & Le, 2008; Yang, Mabu, Shimada, & Hirasawa, 2011; Yin & Han, 2003; Zhang, Chen, & Wei, 2011; Zhao, Tsang, Chen, & Wang, 2010). A common issue in these problems is frequent itemset mining. Frequent itemset mining methods include the Apriori algorithm (Agrawal & Srikant, 1994), the FP-Growth algorithm (Han, Pei, & Yin, 2000), methods based on IT-tree (Zaki & Hsiao, 2005), and methods for mining frequent itemsets in incremental databases (Hong, Lin, & Wu, 2009; Le, Vo, Hong, & Le, 2012; La, Le, & Vo, 2014; Van, Vo, & Le, 2014; Vo, Le, Hong, & Le, 2014).

Rule-based classification is an important method in classification systems because of its high accuracy and good interpretation. Methods for mining rules include decision tree learning (Quinlan, 1986; Quinlan, 1992), the inductive learning algorithm (ILA) (Tolun & Abu-Soud, 1998; Tolun, Sever, Uludag, & Abu-Soud, 1999), and associative classification (AC) (Liu et al., 1998). Some studies showed that classification based on AC is more accurate than those based on decision tree learning (Veloso et al., 2006; Veloso et al., 2007; Veloso et al., 2011). Therefore, a lot of algorithms based on AC have been developed in recent years. They can be divided into two types. The first type focuses on accuracy (Chien & Chen, 2010; Coenen et al., 2007; Giuffrida, Chu, & Hanssens, 2000; Li et al., 2001; Lim & Lee, 2010; Liu et al., 1998; Liu et al., 2000; Liu et al., 2008; Priss 2002; Sun, Wang, & Wong, 2006; Thabtah et al., 2004; Vo & Le, 2008; Yang et al., 2011; Yin & Han, 2003; Zhang et al., 2011; Zhao et al., 2010). These studies have developed approaches to prune rules and predict the classes of unknown cases based on the built classifiers. The second type

☆ This work is a revised and expanded version of the paper "Mining Class Association Rules with the Difference of Obidsets" presented at ACIIDS 2014, Bangkok, Thailand (Nguyen, 2014).

* Corresponding author at: Division of Knowledge and System Engineering for ICT, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Viet Nam.

E-mail addresses: Thuy.Loan.Nguyen.Thi@pwr.edu.pl, nthithuyloan@gmail.com (L.T.T. Nguyen), nguyenngocthanh@tdt.edu.vn, Ngoc-Thanh.Nguyen@pwr.edu.pl (N.T. Nguyen).

http://dx.doi.org/10.1016/j.eswa.2015.01.002
0957-4174/

Input: Dataset D, *minSup*, and *minConf*

Output: all CARs that satisfy *minSup* and *minConf*

Procedure:

CAR-Miner-Diff(L_r, *minSup*, *minConf*)

1. CARs = ∅;
2. L_r= {Single items whose supports satisfy *minSup*}
3. Call **CAR-Miner-Diff-Recursive**(L_r, *minSup*, *minConf*)
4. return CARs

CAR-Miner-Diff-Recursive(L_r, *minSup*, *minConf*)

5. for all $l_i \in L_r$.children do
6. **ENUMERATE-CAR**(l_i, *minConf*)
7. P_i = ∅;
8. for all $l_j \in L_r$.children, with j >i do
9. if l_i.att ≠l_j.att then// by Theorem 1
10. O.att = l_i.att ∪l_j.att;
11. O.values = l_i.values ∪l_j.values;
12. if L_r = null then
13. O.*Obidset* = l_i.*Obidset* \ l_j.*Obidset*;// by definition 5
14. else
15. O.*Obidset* = l_j.*Obidset* \ l_i.*Obidset*;// by Theorem 2
16. O.count = l_i.count - {count(x ∈ O.*Obidset* | class(x) = c_i,∀i∈[1,k]};
17. O.pos = $\arg\max_{i\in[1,k]}\{l.count_i\}$;
18. O.*total* = l_i.*total* - |O.*Obidset*|;
19. if O.count[O.pos] ≥ *minSup* then
20. $P_i = P_i \cup O$;
21. **CAR-Miner-Diff-Recursive**(P_i, *minSup*, *minConf*)

ENUMERATE-CAR-Diff(l, *minConf*)

22. conf = l.count[l.pos] / l.*total*;
23. if conf ≥ *minConf* then
24. CARs = CARs ∪ {l.itemset→c_{pos} (l.count[l.pos], conf)}

Fig. 1. CAR-Miner-Diff algorithm.

focuses on the time required to mine classification association rules (CARs) (Li et al., 2001; Nguyen et al., 2012; Nguyen et al., 2013; Vo & Le, 2008). These studies have also developed methods for fast pruning redundant rules (the rule set after pruning is called pCARs).

This paper focuses on the second type, i.e., proposing a method for efficiently mining CARs in large datasets. With this type, there are some methods for improving the mining time. CMAR adopted an FP-tree-based approach for mining CARs (Li et al., 2001). CMAR scans the dataset two times and uses a complex data structure (using FP-tree with class label) to mine CARs. ECR-CARM (Vo & Le, 2008) proposed a new data structure (ECR-tree) to store itemsets. It only scans dataset one time. However, each node in ECR-tree stores many itemsets which makes ECR-CARM generate-and-check many candidates. Recently, the CAR-Miner algorithm has been proposed for efficiently mining CARs. CAR-Miner utilizes the MECR-tree (a modified data structure of ECR-tree) and the divide-and-conquer technique, using which the problem is transformed into sub-problems that can be solved separately. CAR-Miner used Obidset to compute the supports of rules fast. However, the size of Obidsets grows quickly in large datasets, which means that a lot of memory is required for storing them and a lot of computation time is required to compute the intersection of Obidsets. Consequently, reducing the memory consumption and run time of CAR-Miner is an important issue. Therefore, this paper proposes an improved version of CAR-Miner. Instead of storing the whole intersection of two Obidsets, only their difference is stored. A strategy for reducing the storage space of d2O is also proposed. The experimental results show that the proposed algorithm is superior to CAR-Miner in terms of memory usage and run time.

The main contributions of this paper are as follows:

(i) We prove that Obidset can be replaced by d2O to compute the support of itemsets (Theorem 2 in Section 4).
(ii) Sorting strategy is derived for fast computing d2O of itemsets. This strategy is verified by Theorems 3 and 4 in Section 4.
(iii) An algorithm for fast mining CARs using d2O is also developed.

The rest of this paper is organized as follows. Section 2 presents works related to CAR mining and Section 3 presents some definitions of CARs. The main contributions are described in Section 4, including the concept of the difference of two Obidsets. An example is also given. Section 5 presents the experiments and Section 6 gives the conclusions and suggestions for future work.

2. Related works

As mentioned above, rule-based classification is very efficient in terms of high accuracy and good interpretation. Thus, numerous rule-based classification methods have been developed recently. Quinlan proposed decision tree learning (Quinlan, 1986) and the algorithm ID3, which uses the information gain (IG) measure. ID3 selects the attribute with the highest IG to be the root node and partitions unclassified data into subsets of data. Each subset is computed recursively until no subsets can be generated. C4.5 was proposed with some improvements over ID3 (Quinlan, 1992). C4.5 uses the gain ratio measure to handle data with missing attribute values. ILA, an inductive algorithm, was proposed by Tolun and Abu-Soud (1998). ILA often has higher accuracy than those of ID3 and C4.5 since it mines general rules.

C4.5 and ILA, which are based on heuristics and greedy approaches, generate rule sets that are either too general or too overfitting for a given dataset. They thus often yield high error ratios. Recently, a new method for classification, called classification based on associations (CBA) (Liu et al., 1998), has been proposed for mining CARs. This method has advantages over the heuristic and greedy methods, including higher accuracy due to its ability to easily remove noise. It can additionally generate a rule set that is more complete than those obtained using C4.5 and ILA. One of the weaknesses of mining CARs algorithms is that they require more computation time than C4.5 and ILA because they have to check the generated rules with a set of other rules. Many methods have thus been developed to reduce the mining time and memory consumption of CBA. The first method for mining CARs was proposed by Liu et al. (1998). The authors proposed CBA-RG, an Apriori-based algorithm, for mining CARs. An algorithm for building the classifier (CBA-CB) based on mined CARs was also proposed. Although it improved accuracy, it required more time to generate-and-test candidates in the CBA-RG phase. Li et al. proposed a method called CMAR that uses the FP-tree to compress the dataset and the CR-tree to store rules (Li et al., 2001). To predict unseen data, the method finds all rules that satisfy the data and uses a weighted χ^2 measure to determine the class. CMAR must scan the datasets two times and uses a complex data structure for mining CARs. Although FP-tree-based approach is often efficient for mining frequent itemsets. However, when it is applied in mining CARs, we must determine class label along with each record in the tree's path. It makes the tree more complex and it consumes more time to project in this tree. Thabtah et al.