Knowledge-Based Systems 103 (2016) 73–88

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Efficient mining of class association rules with the itemset constraint☆

Dang Nguyen[a,c], Loan T.T. Nguyen[b,c,*], Bay Vo[d,e], Witold Pedrycz[f,g,h]

[a] Division of Data Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[b] Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[c] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[d] Faculty of Information Technology, Ho Chi Minh City University of Technology, Vietnam
[e] College of Electronics and Information Engineering, Sejong University, Seoul, Republic of Korea
[f] Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6R 2V4 AB Canada, Canada
[g] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[h] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

ARTICLE INFO

Article history:
Received 31 August 2015
Revised 23 March 2016
Accepted 25 March 2016
Available online 14 April 2016

Keywords:
Associative classification
Class association rule
Data mining
Useful rules

ABSTRACT

Mining class association rules (CARs) with the itemset constraint is concerned with the discovery of rules, which contain a set of specific items in the rule antecedent and a class label in the rule consequent. This task is commonly encountered in mining medical data. For example, when classifying which section of the population is at high risk for the HIV infection, epidemiologists often concentrate on rules which include demographic information such as gender, age, and marital status in the rule antecedent, and HIV-Positive in the rule consequent. There are two naive strategies to solve this problem, namely pre-processing and post-processing. The post-processing methods have to generate and consider a huge number of candidate CARs while the performance of the pre-processing methods depend on the number of records filtered out. Therefore, such approaches are time consuming. This study proposes an efficient method for mining CARs with the itemset constraint based on a lattice structure and the difference between two sets of object identifiers (*diffset*). Firstly, a lattice structure is built to store all frequent itemsets in the dataset. To reduce memory usage, instead of the entire set of object identifiers, the *diffset* is used. Secondly, the lattice is traversed to generate only rules which satisfy the itemset constraint. The experimental results show that the proposed algorithm outperforms existing methods in terms of both the mining time and memory usage.

1. Introduction

A class association rule is a special case of association rule where the rule consequent contains only a class label. The problem of mining class association rules (CARs) is to find a complete set of CARs, which satisfy user-specified minimum support and minimum confidence thresholds. In recent years, numerous approaches have been proposed to solve this problem. Table 1 provides an overview of several existing algorithms for CAR mining.

CAR mining has been applied in many practical various domains, such as healthcare [1–3], hotel and tourism management [4,5], social security [6], and education [7]. However, mining all CARs often produces redundant results. In the real-world, end users often consider only a subset of CARs, for instance, those that contain a set of specific items in the rule antecedent. For example, in cancer treatment applications researchers often focus on rules involving new drugs to understand the effectiveness of new treatment strategies. Mining CARs with the itemset constraint has also been demonstrated in the public health domain [8]. Therefore, an ongoing problem is mining CARs with the itemset constraint. This task can be accomplished by checking the rules obtained with the constraint in the pre-processing or post-processing step. However, these approaches have to generate a complete set of CARs, and thus require much time and effort. In this work, our goal is to overcome this limitation by proposing a lattice-based approach for mining CARs with the itemset constraint. The primary contributions of this paper are as follows. First, we propose a lattice structure for storing all frequent itemsets in the dataset. Note that rather than generating and storing all rules, like post-processing approaches do, we only find and keep frequent itemsets

☆ This work is a revised and expanded version of the paper entitled "A novel method for mining class association rules with itemset constraints" presented at ICCCI 2014, Seoul, Korea.

* Corresponding author. Tel.: +840839744186.
E-mail addresses: nguyenphamhaidang@tdt.edu.vn (D. Nguyen), nguyenthithuyloan@tdt.edu.vn, nthithuyloan@gmail.com (L.T.T. Nguyen), bayvodinh@gmail.com (B. Vo), wpedrycz@ualberta.ca (W. Pedrycz).

http://dx.doi.org/10.1016/j.knosys.2016.03.025
0950-7051/

Table 1
CAR mining algorithms: an overview.

Algorithms	Approach to generate CARs
CBA [17], CAAR [29], CACA [30], and CBC [31]	Use Apriori-like algorithms to generate rules
GARC [32] and GEAR [33]	Generate rules based on Apriori-like algorithms with information gain
CMAR [18], L3 [34], and G-L3 [35]	Base on FP-growth like algorithms to generate rules
MMAC [36], MCAR [37], and MCAR [38]	Mine CARs based on the vertical dataset format
ECR-CARM [19]	Use the equivalence class rule tree (ECR-tree) to generate candidate rules
MAC [20] and PAM [39]	Use the TID list intersection approach to compute the support of a rule
CAR-Miner [21]	Mine rules based on the modified ECR-tree with *Obidset*
GA-ACR [40]	Base on GA approach to generate rules
MR-MCAR [41] and PMCAR [42]	Mine CARs with parallel and distributed approaches

because generating rules from frequent itemsets requires much effort and time [9]. We also use the *diffset* technique (the difference between two sets of object identifiers) in order to reduce the memory consumption for storing the lattice structure. Second, we use the paternity relations among nodes to discover rules which satisfy the constraint without re-building the lattice. Finally, an efficient and fast algorithm for mining CARs with the itemset constraint is developed. In comparison with existing methods, the proposed method significantly reduces the time needed for mining.

The originality and efficiency of the proposed algorithm are implied by the two main contributions of this work, as follows:

(1) Formation of an efficient lattice structure for mining CARs with the itemset constraint. Although the attributes and values in the dataset are fixed, the itemsets in the constraint can be changed by the time, depending on the end user's requirements. Unlike existing non-lattice-based methods, the proposed algorithm does not re-build its data structure when the constraint is changed, as the computational cost of this is very high and time-intensive. Because the lattice stores all frequent itemsets, the proposed algorithm searches only nodes containing constrained itemsets on the lattice and uses their paternity relations to discover expected CARs. This feature can reduce the search space, significantly reducing computing overhead.
(2) Usage of an efficient memory reduction strategy based on the *diffset* technique. Instead of storing the intersection of two sets of object identifiers, the method stores only the difference between them. This feature has an obvious advantage when working with dense datasets, because most object identifiers are nearly identical on such datasets. Moreover, the method can deal with multiple itemset constraints and does not produce duplicate rules. The soundness and completeness of the proposed algorithm is proved.

The rest of this paper is organized as follows. In Section 2, the problem statement and definitions of CAR mining are briefly given. Related works, including those on mining association rules with the itemset constraint and mining class association rules with the itemset constraint, are introduced in Section 3. The main contributions of this work are presented in Section 4, in which the lattice structure with *diffset* is presented. The proposed algorithm, LD-CARM-IC, for efficiently mining CARs with the itemset constraint, is also described in this section. The experimental results are presented in Section 5 while the conclusions and directions for future work are discussed in Section 6.

2. Problem statement and definitions

Let D be a dataset with n attributes $\{A_1, A_2, ..., A_n\}$ and $|D|$ denotes records (objects) where each record has an object identifier (*OID*). A set of *OID*s is called an *Obidset*. Let $C = \{c_1, c_2, ..., c_k\}$ be a list of class labels. A specific value of an attribute A_i and class C are denoted by lower-case letters a_{mi} and c_j, respectively.

Definition 1. An *item* is described as an attribute and a specific value for that attribute, denoted by $\langle(A_i, a_{mi})\rangle$. Let I be the set of all items in the dataset. A set $X \subseteq I$ is called an *itemset*.

Definition 2. A class association rule R has the form $X \rightarrow c_j$, where $c_j \in C$ is a class label and X is an *itemset*.

Definition 3. The actual occurrence *ActOcc(R)* of rule R in D is the number of records of D that match R's antecedent.

Definition 4. The support of rule R, denoted by *Supp(R)*, is the number of records of D that match R's antecedent and are labeled with R's class.

Definition 5. The confidence of rule R, denoted by *Conf(R)*, is defined as:

$$Conf(R) = \frac{Supp(R)}{ActOcc(R)}$$

Definition 6. (Support constraint): Given a *minimum support* threshold δ, a rule R satisfies the support constraint iff $Supp(R) \geq \delta$.

Definition 7. (Confidence constraint): Given a *minimum confidence* threshold σ, a rule R satisfies the confidence constraint iff $Conf(R) \geq \sigma$.

Definition 8. (Itemset constraint): Let β be a set of itemsets (called an *itemset constraint*). We assume without loss of generality that the itemsets in β are disjoint. That is, β is of the form $\beta = \{X_1, X_2, ..., X_h\}$. A rule $X \rightarrow c_j$ satisfies the itemset constraint β iff $\exists X_i \in \beta$ and $X_i \subseteq X$.

Example 1. We consider $\beta = \langle(A, a3), (B, b3)\rangle, \langle(C, c1)\rangle$. Three rules $\langle(A, a3), (B, b3)\rangle \rightarrow 1$, $\langle(C, c1)\rangle \rightarrow 1$, and $\langle(A, a3), (B, b3), (C, c1)\rangle \rightarrow 1$ are said to satisfy β. However, rule $\langle(A, a3)\rangle \rightarrow 1$ does not satisfy β.

Problem statement: Given a dataset D, an itemset constraint β, a minimum support threshold δ, and a minimum confidence threshold σ, the problem of mining CARs with the itemset constraint is to find all CARs satisfying three constraints: the support constraint, the confidence constraint, and the itemset constraint.

A sample dataset is shown in Table 2. It contains eight objects, three attributes (A, B, and C), and two classes (1 and 2). For example, consider rule R: $\langle(A, a1)\rangle \rightarrow 1$. We have *ActOcc(R)*=3 and *Supp(R)*=2 because there are three objects with *A=a1*, in that two objects have the same class 1 in Table 2. We also have $Conf(R) = \frac{Supp(R)}{ActOcc(R)} = \frac{2}{3}$.