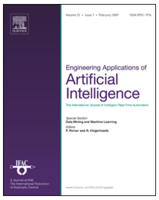
Engineering Applications of Artificial Intelligence 36 (2014) 64–80

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Simultaneous mining of frequent closed itemsets and their generators: Foundation and algorithm

Anh Tran [a,*], Tin Truong [a], Bac Le [b]

[a] Department of Mathematics and Computer Science, University of Dalat, Dalat, Vietnam
[b] Department of Computer Science, University of Science, VNU – Ho Chi Minh, Ho Chi Minh City, Vietnam

ARTICLE INFO

Article history:
Received 2 December 2013
Received in revised form
31 May 2014
Accepted 7 July 2014
Available online 7 August 2014

Keywords:
Closed itemset
Generator
Association rule
Frequent itemset
Data mining

ABSTRACT

Closed itemsets and their generators play an important role in frequent itemset and association rule mining. They allow a lossless representation of all frequent itemsets and association rules and facilitate mining. Some recent approaches discover frequent closed itemsets and generators separately. The Close algorithm mines them simultaneously but it needs to scan the database many times. Based on the properties and relationships of closed itemsets and generators, this study proposes GENCLOSE, an efficient algorithm for mining frequent closed itemsets and generators simultaneously. The level-wise search over an ItemsetObject–setGenerator–Tree enumerates the generators by using a necessary and sufficient condition to produce $(i+1)$-item generators from i-item generators. This condition, based on transaction (object) sets that can be efficiently implemented using diffsets, is very convenient and reliably proved. In the search, pre-closed itemsets are gradually extended using three proposed extension operators. It is shown that these itemsets produce the expected closed itemsets. Extensive experiments on many benchmark databases confirm the efficiency of the proposed approach.

1. Introduction

Association rule mining (Agrawal et al., 1993) from transaction databases is a fundamental technique in data mining. The task is to determine the association rules that satisfy the pre-defined minimum support and confidence from a given database. It was originally designed for market basket applications (Agrawal et al., 1993), but has been extended to various domains, such as risk management, telecommunication networks, and bio-sequences. Association rule mining has two phases (Agrawal and Srikant, 1994): (a) extraction of all *frequent itemsets* whose occurrences exceed the minimum support, and (b) generation of *association rules* that satisfy the given minimum confidence from the itemsets. If all frequent itemsets and their supports are known, association rule generation is straightforward. Hence, most researchers have concentrated on finding efficient methods for mining frequent itemsets.

The basic algorithms for mining frequent itemsets are Apriori, FP-growth, and Eclat. Apriori and its variations (Agrawal et al., 1993; Agrawal and Srikant, 1994) are based on the Apriori property, which states that every subset of a frequent itemset is also frequent, i.e., the *support* of an itemset never exceeds the supports of its subsets. Although this anti-monotone property helps significantly reduce the search space, Apriori-based algorithms are not efficient as they generate many redundant candidates, which increase the CPU and memory burden. Further, they have to scan the database multiple times. To overcome these issues, frequent pattern tree-based algorithms were proposed by Han and Pei (2000) and Han et al. (2004). The original database is compressed into a FP-tree or a similar tree structure. Using divide-and-conquer and depth-first search approaches, all large itemsets are mined from frequent 1-itemsets[1] without having to rescan the database. However, in interactive or incremental mining systems, where the users often change the minimum support and insert new transactions into the original database, FP-tree-inspired structures are unsuitable because the trees need to be rebuilt. Both the Apriori and FP-tree based methods work with a horizontal data format. Zaki proposed Eclat (Zaki, 2000) and DEclat (Zaki and Gouda, 2003) for mining with a vertical data format. These algorithms all show good performance for sparse databases with short itemsets, such as

* Corresponding author. Postal address: 01 Phu dong thien vuong Street, Dalat City, Vietnam. Tel.: +84 983 185 834.
E-mail addresses: anhtn@dlu.edu.vn (A. Tran), tintc@dlu.edu.vn (T. Truong), lhbac@fit.hcmus.edu.vn (B. Le).

[1] Briefly, a set of i items is denoted as i-set, e.g., i-itemset, i-generator.

http://dx.doi.org/10.1016/j.engappai.2014.07.004
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

market databases. For dense databases, which produce long frequent itemsets, such as bio-sequences and telecommunication networks, the frequent itemset class can grow to be unwieldy even if the minimum support is large (Bayardo, 1998). A frequent itemset of length n produces 2^{n-1} frequent non-empty, proper sub-itemsets. Hence, the generation of frequent itemsets not only has the large time complexity $O(2^N)$ (where N is the number of items) but also produces many duplicates in the huge search space. Mining only *maximal frequent itemsets* is one of the solutions for overcoming the drawbacks mentioned above. Many algorithms have been proposed for mining such itemsets (Bayardo, 1998; Burdick et al., 2001). An itemset is maximal frequent if none of its proper supersets are frequent. The number of maximal itemsets is much smaller than that of all frequent itemsets (Zaki and Hsiao, 2005). Although all sub-itemsets of a maximal itemset are frequent, their actual supports are unknown. Further, since frequent itemsets can come from different maximal ones, it takes a lot of time to mine and delete the duplicates. Therefore, maximal frequent itemsets are unsuitable for frequent itemset and association rule generation.

A more suitable approach to overcome this difficulty is using the closures of itemsets, i.e., closed itemsets. The maximal itemset class is contained in the closed itemset class, which, in turn, is a subset of the itemset class. An extensive experimental evaluation conducted by Zaki and Hsiao (2005) showed that, for real databases, the number of frequent closed itemsets is about 10 times greater than that of maximal frequent itemsets, but about 100 times smaller than the cardinality of frequent itemsets. Hence, mining *frequent closed itemsets* has received the attention of many researchers (Pasquier et al., 1999; Pei et al., 2000; Wang et al., 2003; Singh et al., 2005). An itemset is *closed* iff[2] it is identical to its *closure*. This concept is similar to the concept lattice (Birkhoff, 1967; Wille, 1982, 1992; Davey and Priestley, 1994; Ganter and Wille, 1999) and has been recently applied (Boulicaut et al, 2003; Zaki, 2004; Pasquier et al., 2005). A *generator* (Pasquier et al., 1999; Szathmary et al., 2009) of an itemset is its minimal subset that has the same closure as its own. Generators are also called *minimal generators* (Zaki, 2004; Dong et al., 2005), *key patterns* (Bastide et al., 2000), and *free-sets* (Boulicaut et al., 2003). Although there are many definitions of closed itemsets and generators, they are equivalent (see Section 2.1).

Closed itemsets together with their lattice structure and generators, called $\mathscr{LG}_{\mathcal{A}}$, play an important role in both itemset and association rule mining. First, their cardinality is typically orders of magnitude much lower than that of all itemsets. Whenever the user changes the minimum support, all frequently closed itemsets and their generators can be quickly derived from $\mathscr{LG}_{\mathcal{A}}$. Second, two itemsets are equivalent if they have the same closure. In the study by Anh et al. (2012b), based on this *equivalence relation*, all itemsets were partitioned into disjoint equivalence classes. This decreases most duplication in the generation of all itemsets. Further, each class can be explored independently. In each class, a closed itemset is a maximum set, its generators are minimal subsets, and each remaining itemset has a unique representation through its closure and generators. Thus, the duplication in the generation of all frequent itemsets is completely removed. Hence, frequent closed itemsets together with their generators produce a *lossless representation* of all frequent itemsets. Third, many studies have used the generators of closed itemset for mining association rules (Balcazar, 2010; Bay et al., 2012; Anh et al., 2012a; Pasquier et al., 1999, 2005; Zaki, 2004; Tin and Anh, 2010a; Tin et al., 2010b). All rules can be obtained based on frequent closed itemsets and their generators. The lattice of frequent closed itemsets and generators with constraints is essential for the discovery of frequent itemsets and association rules with item constraints, especially when the minimum support and confidence thresholds and item constraints often change (Anh et al., 2011, 2012b, 2014; Hai et al., 2013, 2014).

The problem of mining frequent closed itemsets and generators is stated as follows: given a transaction database and a minimum support threshold, the task is to find all frequent closed itemsets together with their generators. The algorithms for mining closed itemsets can be divided into three approaches, namely generate-and-test, divide-and-conquer and hybrid. Many algorithms have been proposed for mining closed itemsets, including Close (Pasquier et al., 1999) (generate-and-test), Closet (Pei et al., 2000) and Closet+ (Wang et al., 2003) (divide-and-conquer), and Charm (Zaki and Hsiao, 2005) and CloseMiner (Singh et al., 2005) (hybrid). Algorithms for mining generators include Talky-G (Szathmary et al., 2009) and MinimalGenerator (Zaki, 2004). However, most of these algorithms discover frequent closed itemsets and generators separately. The exception is Close, which mines them simultaneously. However, its execution is computationally expensive. The present study proposes GENCLOSE, which has the following key features:

1) Generators and frequent closed itemsets are simultaneously found using *breadth-first search* over an *IOG*-tree (Itemset-Object-set[3]-Generator tree).
2) At each level, the generators are first mined using a *necessary and sufficient condition* to determine the class of $(i+1)$-generators from the class of i-generators $(i \geq 1)$ based on the object-sets (or diffsets in practice).
3) Three *extension operators* are proposed to extend itemsets toward their closures when mining generators.

The rest of this paper is organized as follows. Section 2 gives the background of closed sets, generators, and their definitions. Related work is also discussed in this section. Section 3 proposes some necessary and sufficient conditions for producing generators and three operators for extending generators to their closures. Based on these results, the GENCLOSE algorithm is constructed. In Section 4, GENCLOSE is compared to CharmLMG and DTouch, two well-known algorithms for finding closed itemsets and generators. The conclusion is given in Section 5. For better readability, some proofs and implemented techniques are given in the appendices.

2. Foundation of mining closed itemsets and their generators

Consider two non-empty sets: $\mathcal{O}$ containing objects (or transactions) and $\mathcal{A}$ containing all items (attributes) related to transactions $o \in \mathcal{O}$. Let $\mathscr{R}$ be a binary relation in $\mathcal{O} \times \mathcal{A}$. A triple $\mathcal{D} = (\mathcal{O}, \mathcal{A}, \mathscr{R})$ is called a transaction database or a binary database, or briefly a *database*. A set of items $A \subseteq \mathcal{A}$ and a set $O \subseteq \mathcal{O}$ are called an *itemset* and an *object-set*, respectively. Let $2^{\mathcal{O}}$ and $2^{\mathcal{A}}$ be the power sets of $\mathcal{O}$ and $\mathcal{A}$. Two set functions of λ: $2^{\mathcal{O}} \rightarrow 2^{\mathcal{A}}$ and ρ: $2^{\mathcal{A}} \rightarrow 2^{\mathcal{O}}$ are determined as follows: $\forall \mathrm{A} \subseteq \mathcal{A}$, $O \subseteq \mathcal{O}$, $\mathrm{A} \neq \varnothing$, $O \neq \varnothing$, $\lambda(O) = \{a \in \mathcal{A} | (o, a) \in \mathscr{R}, \forall o \in O\}$, $\rho(A) = \{o \in \mathcal{O} | (o, a) \in \mathscr{R}, \forall a \in A\}$, and as convention $\rho(\varnothing) := \mathcal{O}$, $\lambda(\varnothing) := \mathcal{A}$. The *closure* of an itemset A is defined by $h(A) = \lambda(\rho(A))$, and that of an object-set O is defined by $h'(O) = \rho(\lambda(O))$. Thus, itemset A is called *closed* iff it is identical to its closure, i.e., $A = h(A)$. Similarly, an object-set O is closed iff $O = h'(O)$ (for details, refer to the studies of Birkhoff (1967), Wille (1982), Wille (1992), Davey and Priestley (1994), and Ganter and Wille (1999)).

[2] For convenience, we write "if and only if" simply as "iff".

[3] Object-set means set of objects.