Expert Systems with Applications 41 (2014) 2703–2712

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Incrementally building frequent closed itemset lattice

Phuong-Thanh La [a], Bac Le [a], Bay Vo [b,*]

[a] Computer Science Department, University of Science, VNU-Ho Chi Minh, Viet Nam
[b] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Keywords:
Data mining
Incremental mining
Iceberg lattice
Frequent closed itemsets

ABSTRACT

A concept lattice is an ordered structure between concepts. It is particularly effective in mining association rules. However, a concept lattice is not efficient for large databases because the lattice size increases with the number of transactions. Finding an efficient strategy for dynamically updating the lattice is an important issue for real-world applications, where new transactions are constantly inserted into databases. To build an efficient storage structure for mining association rules, this study proposes a method for building the initial frequent closed itemset lattice from the original database. The lattice is updated when new transactions are inserted. The number of database rescans over the entire database is reduced in the maintenance process. The proposed algorithm is compared with building a lattice in batch mode to demonstrate the effectiveness of the proposed algorithm.

1. Introduction

The goal of data mining is to extract information from a data set and transform it into an understandable structure for further use. Incremental mining is a major challenge in data mining as the real application database is always changed time by time. Therefore, the incremental method avoids reading the full database many times. Data mining consists of three main problems, namely mining association rules, classification, and clustering.

Mining frequent closed itemsets (FCIs) is very important in the process of mining association rules. Many algorithms have been proposed for mining FCIs, including FP-Close (Han, Cheng, Xin, & Yan, 2007), DBV-Miner (Vo, Hong, & Le, 2012), Closet+ (Wang, Han, & Pei, 2003), CHARM and CHARM-L (Zaki & Hsiao, 2005), which use a horizontal or vertical database format. Horizontal database formats are similar to real-world databases, with each line in a database being a transaction with one or more items. In vertical database formats, each line has an item and a list of all transactions (referred to as the transaction IDs) that contain the corresponding item. For such databases, the entire database might need to be updated when transactions are added.

Concept lattices, widely used in mathematics (Ganter & Wille, 1999), are ordered structures between concepts. Each concept consists of three parts, namely a set of objects, a set of attributes, and a relationship between these two sets. Alternatively, each concept consists of the closed itemset (CI), the closed transaction set, and the relationship between these two closed sets (Szathmary et al., 2013). Building a concept lattice requires building parent–child relationships directly between the CIs. Formal concept analysis (FCA) Ganter & Wille, 1999, a mathematical method used to mine CIs, has been proposed. Zaki and Hsiao (2005) defined the concept intent as CIs and the concept extent as a set of transaction identifiers. Methods for maintaining a concept lattice can be grouped into direct-update-based and merge-based methods.

Concept lattices are rarely used in real-world data mining with massive databases because the concept lattice size increases with the number of transactions. Therefore, Pasquier et al. Rouane-Hacene, Huchard, Napoli, and Valtchev (2013) and Szathmary et al. (2011) proposed the iceberg concept lattice, also known as an FCI lattice. Each node of the iceberg lattice contains an FCI, so the number of nodes in the lattice is small. The iceberg lattice theory is based on the concept lattice theory. Thus, methods for maintaining iceberg lattices are direct-update-based or merge-based. Based on a direct-update-based method, individual insertion of transactions, Valtchev, Missaoui, and Godin (2008) proposed an algorithm that uses minimal generators. Gupta, Bhatnagar, and Kumar (2010) proposed using an index to improve searching and updating a lattice. Merge-based methods often build a new lattice from added transactions and then combined it with the initial lattice, which was created from the original database. Ceglar and Roddick (2007) proposed an algorithm that builds an updated lattice, merges lattices, and deletes a concept. Valtchev and Missaoui (2001) proposed an algorithm that merges two lattices by building an immediate lattice, known as a nested line diagram. In Vo, Hong, and Le (2013) proposed an efficient method for building an iceberg lattice and used this lattice for mining most generalization association rules (Vo et al., 2012).

* Corresponding author.
E-mail addresses: thanh.laphuong@gmail.com (P.-T. La), lhbac@fit.hcmus.edu.vn (B. Le), vdbay@it.tdt.edu.vn (B. Vo).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.11.002

The present study proposes a method for constructing a lattice from the original database and updating it when a new transaction is added. The framework is based on FCA and lattice structure, which benefits for the association rule mining. The approach reduces the FCI mining problem by individual insertion of transactions into the database and analyzing their impact on mining results.

The proposed algorithm has the following advantages:

1. Exact CIs are mined using upper and lower support thresholds.
2. No generators are used for updating the iceberg lattice.
3. The number of full database rescans is reduced.
4. The support threshold can be modified at the rescanning point.

The rest of this paper is organized as follows. Section 2 reviews related works. The proposed algorithm is described in Section 3. Section 4 demonstrates the use of the proposed algorithm with an example. Section 5 presents the experimental results to show the effectiveness of the proposed algorithm. A discussion of the proposed algorithm is given in Section 6. Finally, the conclusions and future work are presented in Section 7.

2. Related work

2.1. Frequent itemsets and frequent closed itemsets

Let D be a transaction database with a pair $\{I, T\}$, where $I = \{i_1, i_2, \ldots, i_m\}$ is a set of all items and $T = \{t_1, t_2, \ldots, t_n\}$ is a set of all transaction identifiers. A transaction $t \in D$ is a collection of items associated with a unique identifier, called TID. A set $X \subseteq I$ is called an *itemset* and a set $Y \subseteq T$ is called a *tidset*. The support of itemset X, denoted as $\sigma(X)$, is the number of transactions in which it occurs as a subset. Itemset $X \subseteq I$ is a frequent itemset if its support is greater than or equal to a user-defined *minimum support*. A frequent itemset X is called *closed* if there exists no proper superset $Y \supset X$ with $\sigma(X) = \sigma(Y)$.

2.2. Lattice structure

2.2.1. Formal concept analysis

Valtchev and Missaoui (2001) and Gupta et al. (2010) provided three definitions and two theorems for FCA.

Definition 1 Valtchev and Missaoui (2001). A formal context is a triple $K = (O, I, A)$, where O and A are sets (objects and attributes, respectively) and I is an incident relation, i.e., $I \subseteq O \times A$.

Following standard FCA notations, objects are denoted by numbers and attributes are denoted by small letters. A set is given, whenever possible, in a separator-free form. For example, 148 denotes the set objects {1, 4, 8} and adfh denotes the set attributes {a, d, f, h}.

Definition 2 Valtchev and Missaoui (2001). The function f maps a set of objects into the set of common attributes, and g maps a set of attributes into the set of common objects.

$$f : P(O) \to P(A), \quad f(X) = \{a \in A | \forall o \in X, oIa\}$$

$$g : P(A) \to P(O), \quad g(Y) = \{o \in A | \forall a \in Y, oIa\}$$

Functions f and g are also denoted by a quote (‘). The compound operators $g \circ f(X)$ and $f \circ g(Y)$ are denoted by double quotes (”), where $g \circ f(x)$ is the closure operator over $P(O)$, and $f \circ g(Y)$ is the closure operator over $P(A)$. For example, based on the concept lattice in Fig. 1, $f(134)$ is fgh and g(abc) is 127. If $X = \{2, 7\}$, then $X'' = \{1, 2, 7\}$.

Definition 3 Valtchev and Missaoui (2001). A formal concept is a couple (X, Y), where $X \in P(O)$, $Y \in P(A)$, $X = Y'$, and $Y = X'$. X is called the extent and Y is called the intent of concept (X, Y). X' is called the object intent of X and Y' is called the attribute extent of Y. For example, (134, fgh) is a concept, whereas (16, efh) is not a concept.

Theorem 1 Ganter and Wille (1999). *Each concept of a context (O, I, A) has the form (X'', X') for subset $X \subseteq O$ and the form (Y', Y'') for some subset $Y \subseteq A$.*

This theorem means that all extents are CIs and all indents are closed tidsets.

Theorem 2 Ganter and Wille (1999). *Every extent is the intersection of attribute extents and every intent is the intersection of object intents.*

This theorem means that the intersection of extents of nodes in a lattice is always an extent of a node in the lattice. The intersection of intents of nodes in a lattice is always an intent of a node in the lattice.

For proofs of these theorems, please refer to Ganter and Wille (1999).

2.2.2. Lattice order, upper bound, and lower bound

According to Szathmary et al. (2011), the set C_K of all concepts of the context $K = (O, A, I)$, C_k, is partially ordered by intent/extent inclusion:

$$(X_1, Y_1) \leqslant_k (X_2, Y_2) \iff X_1 \subseteq X_2 (Y_2 \subseteq Y_1)$$

The partial order $L = \langle C_k, \leqslant_k \rangle$ is a complete lattice least upper bound (LUB) and greatest lower bound (GLB) as follows:

$$\vee_{i=1}^{k}(X_i, Y_i) = ((\cup_{i=1}^{k} X_i)'', \cap_{i=1}^{k} Y_i)$$

$$\wedge_{i=1}^{k}(X_i, Y_i) = (\cap_{i=1}^{k} X_i, (\cup_{i=1}^{k} Y_i)'')$$

The LUB is also called the join, denoted , and the GLB is also called the meet, denoted . For example, with $C_1 = (123, cf)$ and $C_2 = (1246, ef)$, the join of C_1 and C_2 is (123, cf) (1246, ef) = (12346, f) and the meet of C_1 and C_2 is (123, cf) (1246, ef) = (12, abcef).

2.2.3. Iceberg lattice (frequent closed itemset lattice) definition

Based on the study of Szathmary et al. (2011), iceberg lattices are upper sets of concept lattices that are generated by a specific sort of maximal anti-chains of the lattice, i.e., a sort of concept extent. All concepts in iceberg lattices are frequent concepts. Support γ of a concept $c = (X, Y)$ is $\gamma(c) = ||X||/||Y||$.

Alternatively, an iceberg lattice can be created by a complete horizontal cut of the lattice $L = \langle C_k, \leqslant_k \rangle$ into two parts with respect to a minimal support threshold $\alpha \in (0, 1]$. The upper part is denoted $L^{\alpha} = \langle C^{\alpha}, \leqslant_k \rangle$, where $C^{\alpha} = \{c | c \in C, \gamma(c) \geqslant \alpha\}$.

2.3. Building concept/iceberg lattices

This section reviews the building of concept/iceberg lattices.

Zaki and Hsiao (2005) proposed CHARM-L, which is an iceberg lattice construction algorithm based on a tree structure known as IT-tree. Each node in IT-tree is formed as $X \times t(X)$, where X is an FCI and $t(X)$ is the support of itemset X. When a CI X is found, the set S of all closed parent itemsets is determined to create a link between X and the minimum node in S. A vertical database format is used, so a conversion of horizontal format data is required. Therefore, CHARM-L does not support dynamic updates when a transaction is inserted.

Valtchev, Missaoui, and Lebrun (2000) proposed an algorithm for constructing lattices based on a Hasse diagram from predefined CIs.