ELSEVIER

Expert Systems with Applications 30 (2006) 650–657

Expert Systems with Applications

www.elsevier.com/locate/eswa

An efficient data mining approach for discovering interesting knowledge from customer transactions

Show-Jane Yen*, Yue-Shi Lee

Department of Computer Science and Information Engineering, Ming Chuan University, Taipei, Taiwan, ROC

Abstract

Mining association rules and *mining sequential patterns* both are to discover customer purchasing behaviors from a transaction database, such that the quality of business decision can be improved. However, the size of the transaction database can be very large. It is very time consuming to find all the association rules and sequential patterns from a large database, and users may be only interested in some information.

Moreover, the criteria of the discovered association rules and sequential patterns for the user requirements may not be the same. Many uninteresting information for the user requirements can be generated when traditional mining methods are applied. Hence, a data mining language needs to be provided such that users can query only interesting knowledge to them from a large database of customer transactions. In this paper, a data mining language is presented. From the data mining language, users can specify the interested items and the criteria of the association rules or sequential patterns to be discovered. Also, the efficient data mining techniques are proposed to extract the association rules and the sequential patterns according to the user requirements.

Keywords: Data mining; Association rule; Sequential pattern; Interesting knowledge; Transaction database

1. Introduction

An association rule (Han and Pei, 2000) describes the association among items in which when some items are purchased in a transaction, others are purchased too. An association rule has the form $X \Rightarrow Y$, in which X and Y are two sets of items. In this paper, we refer to X as an antecedent and Y as a consequent of this rule. The *length* of an itemset *i* is the number of items in the itemset *i*, and an itemset of length *k* is called a *k-itemset*. A transaction *t* *supports* an itemset *i* if *i* is contained in *t*. The *support* for an itemset *i* is defined as the ratio of the number of transactions that supports the itemset *i* to the total number of transactions. If the support for an itemset *i* satisfies the user-specified *minimum support* threshold, then *i* is called *frequent itemset*, and a frequent itemset of length *k* a *frequent k-itemset*. The *confidence* of a rule $X \Rightarrow Y$ is defined as the ratio of the support for the itemsets $X \cup Y$ to the support for the itemset X. If itemset $Z = X \cup Y$ is a frequent itemset and the confidence of $X \Rightarrow Y$ is no less than the user-specified *minimum confidence*, then the rule $X \Rightarrow Y$ is an association rule.

Mining sequential patterns (Pie et al., 2001) is to find the sequential purchasing behavior for most customers from a large transaction database. A sequence is an ordered list of the itemsets$\langle s_1, s_2, \ldots, s_n \rangle$, where s_i is a set of items. A *customer sequence* is the list of all the transactions of a customer, which is ordered by increasing transaction-time. A customer sequence *c* *supports* a sequence *s* if *s* is contained in *c*. The *support* for a sequence *s* is defined as the ratio of the number of customer sequences that supports *s* to the total number of customer sequences. If the support for a sequence *s* satisfies the user-specified *minimum support* threshold, then *s* is called *frequent sequence*. The *length* of a sequence *s* is the number of itemsets in the sequence. A sequence of length *k* is called a *k-sequence*, and a frequent sequence of length *k* a *frequent k-sequence*. A *sequential pattern* is a frequent sequence that is not contained in any other frequent sequence.

In this paper, we present a data mining language, from which users only need to specify the criteria and the interested items for discovering the association rules and

* Corresponding author. Tel.: +886 33507001; fax: +886 33593874.
E-mail address: sjyen@mcu.edu.tw (S.-J. Yen).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.07.035

sequential patterns. We also propose efficient data mining algorithms for the data mining language processing. For the data mining algorithms, we focus on discovering the associations among interested items and all the other items. For our data mining system, a user can make a query through our query language, and the system answers to the query according to user specified items and criteria immediately. If the answers do not satisfy user's needs, then user can resubmit his/her query by adjusting the criteria and item constraints.

Many constraint-based mining methods have been proposed. Hipp and Guntzer (2002) presented that data mining process should be an initial unconstrained and costly mining run. The mining queries are answered from the initial mining result such that response time can be minimized. However, the discovered association rules may become invalid or inappropriate since the transactions are increasing any time. It is very costly to re-run the unconstrained mining algorithm to obtain the up-to-date initial mining result. Ng, Lakshmanan, Han, and Mah (1999) considered aggregate constraints and item constraints for mining association rules. For item constraints, the items in the discovered frequent itemset must exactly be contained in the specified items. Pei and Han (2000, 2002) developed pattern-growth methods for constrained frequent pattern mining and sequential pattern mining. An item constraint specifies what is the particular individual or group of items that should or should not be presented in the pattern, that is, the items in the discovered patterns have to be contained in the specified itemset. In (Pei et al., 2002), they discussed about mining sequential patterns with regular expression, the items in the discovered patterns must appear in the sequence defined in the regular expression. All the above approaches cannot discover the associations among certain items and all the other items. Hence, the item constraints in the above approaches are different from our work.

Meo, Psaila and Ceri (1996) proposed a SQL-like operator for extracting association rules. However, SQL-like operator cannot completely express the associations among certain items and all the other items. Furthermore, the SQL-like operator performs set-oriented operations (i.e. join operations), which are very inefficient operations. Yen and Chen (1997) proposed a data mining language for mining interesting association rules. They presented a user-friendly mining language and users can specify the interested items and the criteria of the rules to be discovered. This approach constructs an association graph and generates all the frequent itemsets by traveling the association graph. However, it needs to take a lot of memory space to record the related information. In this paper, we successfully integrate two kinds of patterns and use the similar style of the data mining language proposed in (Yen and Chen, 1997). Besides, we also propose efficient data mining algorithms to find all the associations among certain items and all the other items.

2. Data mining language and database transformation

The data mining language is defined as follows. Users can query association rules or sequential patterns by specifying the related parameters in the data mining language.

Mining ⟨Data Mining Technology⟩
From ⟨CSD⟩
With ⟨(D_1),(D_2), …,(D_m)⟩
Support ⟨s%⟩
Confidence ⟨c%⟩

In the **Mining** clause, ⟨Data Mining Technology⟩ can be ⟨association rules⟩ or ⟨sequential patterns⟩. The former is to discover association rules and the later is to discover sequential patterns.

In the **From** clause, ⟨CSD⟩ is used to specify the database name to which users query the association rules or sequential patterns.

In the **With** clause, if the ⟨Data Mining Technology⟩ is ⟨sequential patterns⟩, ⟨(D_1),(D_2), …,(D_m)⟩ are user-specified itemsets which ordered by increasing purchasing time, and (D_i) can be the notation '*' which represents any sequences. If the ⟨Data Mining Technology⟩ is ⟨association rules⟩, then m is equal to 2, and D_1 and D_2 are the itemsets in the antecedent and consequent, respectively, of the discovered rules. Besides, (D_i) and the items in D_i can be the notation '*' which represents any items.

Support clause is followed by the user-specified minimum support *s*%.

Confidence clause is followed by the user-specified minimum confidence c% if the ⟨Data Mining Technology⟩ is ⟨association rules⟩. If the ⟨Data Mining Technology⟩ is ⟨sequential patterns⟩, this clause is ignored.

In order to find the interesting association rules and sequential patterns efficiently, we need to transform the original transaction data into another type. Each item in each customer sequence is transformed into a bit string. The length of a bit string is the number of the transactions in the customer sequence. If the *i*th transaction of the customer sequence contains an item, then the *i*th bit in the bit string for this item is set to 1. Otherwise, the *i*th bit is set to 0. For example, in Table 1, the bit string for item A in CID 1 is 011. Hence, we can transform the customer sequence database (Table 1) into the *bit-string database* (Table 2).

From the bit-string database, we can easily compute the number of the transactions in a customer sequence, which contain an itemset. For example, in Table 1, if we want to know how many transactions in CID 1 support the itemset (A,C,E). We can perform logical AND operations on the bit strings for items A, C and E in CID 1. The number of 1's in the resultant bit string is the number of the transactions which contain the itemset (A,C,E) in CID 1.