Time-Constrained Sequential Pattern Mining

Ming-Yen Lin
Feng Chia University, Taiwan

INTRODUCTION

Sequential pattern mining is one of the important issues in the research of data mining (Agrawal & Srikant, 1995; Ayres, Gehrke, & Yiu, 2002; Han, Pei, & Yan, 2004; Lin & Lee, 2004; Lin & Lee, 2005b; Roddick & Spiliopoulou, 2002). A typical example is a retail database where each record corresponds to a customer's purchasing sequence, called data sequence. A data sequence is composed of all the customer's transactions ordered by transaction time. Each transaction is represented by a set of literals indicating the set of items (called itemset) purchased in the transaction. The objective is to find all the frequent sub-sequences (called sequential patterns) in the sequence database. Whether a sub-sequence is frequent or not is determined by its frequency, named support, in the sequence database.

An example sequential pattern might be that 40% customers bought PC and printer, followed by the purchase of scanner and graphics-software, and then digital camera. Such a pattern, denoted by <(PC, printer)(scanner, graphics-software)(digital camera)>, has three elements where each element is an itemset. Although the issue is motivated by the retail industry, the mining technique is applicable to domains bearing sequence characteristics, including the analysis of Web traversal patterns, medical treatments, natural disasters, DNA sequences, and so forth.

In order to have more accurate results, constraints in addition to the support threshold need to be specified in the mining (Pei, Han, & Wang, 2007; Chen & Yu, 2006; Garofalakis, Rastogi, & Shim, 2002; Lin & Lee, 2005a; Masseglia, Poncelet, & Teisseire, 2004). Most time-independent constraints can be handled, without modifying the fundamental mining algorithm, by a post-processing on the result of sequential pattern mining without constraints. Time-constraints, however, cannot be managed by retrieving patterns because the support computation of patterns must validate the time attributes for every data sequence in the mining process. Therefore, time-constrained sequential pattern mining (Lin & Lee, 2005a; Lin, Hsueh, & Chang, 2006; Masseglia, Poncelet, & Teisseire, 2004;) is more challenging, and more important in the aspect of temporal relationship discovery, than conventional pattern mining.

BACKGROUND

The issue of mining sequential patterns with time constraints was first addressed by Srikant and Agrawal in 1996 (Srikant & Agrawal 1996). Three time constraints including minimum gap, maximum gap and sliding time-window are specified to enhance conventional sequence discovery. For example, without time constraints, one may find a pattern <(b, d, f)(a, e)>. However, the pattern could be insignificant if the time interval between (b, d, f) and (a, e) is too long. Such patterns could be filtered out if the maximum gap constraint is specified.

Analogously, one might discover the pattern <(b, c, e)(d, g)> from many data sequences consisting of itemset (d, g) occurring one day after the occurrence of itemset (b, c, e). Nonetheless, such a pattern is a false pattern in discovering weekly patterns, i.e. the minimum gap of 7 days. In other words, the sale of (b, c, e) might not trigger the sale of (d, g) in next week. Therefore, time constraints including maximum gap and minimum gap should be incorporated in the mining to reinforce the accuracy and significance of mining results.

Moreover, conventional definition of an element of a sequential pattern is too rigid for some applications. Essentially, a data sequence is defined to support a pattern if each element of the pattern is contained in an individual transaction of the data sequence. However, the user may not care whether the items in an element (of the pattern) come from a single transaction or from adjoining transactions of a data sequence if the adjoining transactions occur close in time (within a specified time interval). The specified interval is named sliding time-

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

window. For instance, given a sliding time-window of 6, a data sequence $< {}_{t1}(a, e)\ {}_{t2}(b)\ {}_{t3}(f)>$ can support the pattern <(a, b, e)(f)> if the difference between time t1 and time t2 is no greater than 6. Adding sliding time-window constraint to relax the definition of an element will broaden the applications of sequential patterns.

In addition to the three time constraints, duration and exact gap constraints are usually specified for finding actionable patterns (Lin, Hsueh, & Chang, 2006; Zaki, 2000). Duration specifies the maximum total time-span allowed for a pattern. A pattern having transactions conducted over one year will be filter out if the duration of 365 days is given. Exact gap can be used to find patterns, within which adjacent transactions occur exactly the specified time difference. The discovery of sequential patterns with additionally specified time constraints is referred to as mining time-constrained sequential patterns.

A typical example of mining time-constrained sequential patterns might be that finding out frequent sub-sequences having minimum support of 40%, minimum gap of 7 days, maximum gap of 30 days, sliding time-window of 2 days, and duration of 90 days.

MAIN FOCUS

Sequential pattern mining is more complex than association rule mining because the patterns are formed not only by combinations of items but also by permutations of itemsets. The number of potential sequences is by far larger than that of potential itemsets. Given 100 possible items in the database, the total number of possible itemsets is

$$\sum_{i=0}^{100}\binom{100}{i} = 2100.$$

Let the size of a sequence (sequence size) be the total number of items in that sequence. The number of potential sequences of size k is

$$\sum_{i_1=1}^{k}\binom{100}{i_1}\sum_{i_2=1}^{k-i_1}\binom{100}{i_2}\sum_{i_3=1}^{k-i_1-i_2}\binom{100}{i_3}\cdots\sum_{i_k=1}^{k-i_1-\cdots-i_{k-1}}\binom{100}{i_k}.$$

The total number of potential sequences, accumulating from size one to size 100 and more, could be enormous. Adding time constraints complicates the mining much more so that the focus of time-constrained sequential pattern mining is to design efficient algorithms for mining large sequence databases. In general, these algorithms can be categorized into Apriori based and pattern-growth based approaches, as well as vertical mining approaches.

Apriori-Based Approaches

Although there are many algorithms dealing with sequential pattern mining, few handle the mining with the addition of time constraints. The GSP (Generalized Sequential Pattern) algorithm (Srikant & Agrawal 1996) is the first algorithm that discovers sequential patterns with time constraints (including minimum gap, maximum gap, and sliding time-window) within Apriori framework. GSP solves the problem by generating and testing candidate patterns in multiple database scans and it scans the database k times to discover patterns having k items. Candidate patterns having any non-frequent sub-sequence are pruned before testing to reduce the search space. In a database scan, each data sequence is transformed into items' transaction-time lists for fast finding of certain element with a time tag. Since the start-time and end-time of an element (may comprise several transactions) must be considered, GSP defines 'contiguous sub-sequence' for candidate generation, and move between 'forward phase' and 'backward phase' for checking whether a data sequence contains a certain candidate.

Pattern-Growth Based Approaches

A general pattern-growth framework was presented for constraint-based sequential pattern mining (Pei, Han, & Wang, 2007). From the application point of view, seven categories of constrains including item, length, super-pattern, aggregate, regular expression, duration, and gap constraints were covered. Among these constraints, duration and gap constraints are tightly coupled with the support counting process because they confine how a data sequence contains a pattern. Orthogonally classifying constraints by their roles in mining, monotonic, anti-monotonic, and succinct constraints were characterized and the prefix-monotone constraint was introduced. The prefix-growth framework which pushes prefix-monotone constraints into PrefixSpan (Pei, Han, Mortazavi-Asl, Wang, Pinto,

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/time-constrained-sequential-pattern-mining/11089

Related Content

Scientific Web Intelligence
Mike Thelwall (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1714-1719).*
www.irma-international.org/chapter/scientific-web-intelligence/11049

Mining Email Data
Steffen Bickel (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1262-1267).*
www.irma-international.org/chapter/mining-email-data/10984

A Data Mining Methodology for Product Family Design
Seung Ki Moon (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 497-505).*
www.irma-international.org/chapter/data-mining-methodology-product-family/10866

Hybrid Genetic Algorithms in Data Mining Applications
Sancho Salcedo-Sanz, Gustavo Camps-Vallsand Carlos Bousoño-Calzón (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 993-998).*
www.irma-international.org/chapter/hybrid-genetic-algorithms-data-mining/10942

Constraint-Based Association Rule Mining
Carson Kai-Sang Leung (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 307-312).*
www.irma-international.org/chapter/constraint-based-association-rule-mining/10837