Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1178–1189
Expert Systems with Applications
www.elsevier.com/locate/eswa

Discovering during-temporal patterns (DTPs) in large temporal databases ☆

Li Zhang [a], Guoqing Chen [a,*], Tom Brijs [b], Xing Zhang [a]

[a] School of Economic and Management, Tsinghua University, Beijing 100084, PR China
[b] Transportation Research Institute, Hasselt University, Diepenbeek B3920, Belgium

Abstract

Large temporal databases (TDBs) usually contain a wealth of data about temporal events. Aimed at discovering temporal patterns with *during* relationship (*during*-temporal patterns, DTPs), which is deemed common and potentially valuable in real-world applications, this paper presents an approach to finding such DTPs by investigating some of their properties and incorporating them as desirable pruning strategies into the corresponding algorithm, so as to optimize the mining process. Results from synthetic reveal that the algorithm is efficient and linearly scalable with regard to the number of temporal events. Finally, we apply the algorithm into the weather forecast field and obtain effective results.

Keywords: Data mining; *During* relationship; Temporal pattern

1. Introduction

In recent years, discovery of association rules (Agrawal, Imielinski, & Swami, 1993) and sequential patterns (Agrawal & Srikant, 1995) has been a major research issue in the area of data mining. While typical association rules usually reflect related events occurring at the same time, sequential patterns represent commonly occurring sequences that are in a time order. However, real-world businesses often generate a massive volume of data in daily operations and decision-making processes, which are of a richer temporal nature. For instance, a customer could buy a DVD machine after TV was bought; the duration of an ERP project partially overlapped the duration of a BPR project; and a patient suffered from cough during the period of fever. Apparently, such temporal relationships (e.g., *after*, *overlap*, *during*, etc.) are kinds of real-world semantics that are, in many cases, considered meaningful and useful in practice. Usually, temporal relationships between events with different time stamps could be categorized into several types in forms of temporal comparison predicates such as *after*, *meet*, *overlap*, *during*, *start*, *finish*, and *equal* (Allen, 1983). Though recent years have witnessed several efforts on discovering the after relationship (Agrawal & Srikant, 1995; Chen, Ai, & Yu, 2002; Das, Lin, & Mannila, 1998; Zaki, 2000), more in-depth investigations of the relationship are still badly needed, let along their explorations of other types of temporal relationships. Furthermore, results from the studies on the after relationship could hardly be simply extended to the case of some other relationships such as *during*, *overlap*, etc. This may be attributed to the fact that in the after relationship, events can generally be dealt with on a time point, whereas in other relationships, events are considered to be of a time interval nature.

On the other hand, both Rainsford et al. (1999) and Hoppner (2001) have recently discussed the issues of finding temporal relationships between time-interval-based events using temporal comparison predicates (Allen, 1983), but with different mining approaches. Rainsford

☆ The work was partly supported by the National Natural Science Foundation of China (70231010/70321001), Tsinghua University's Research Center for Contemporary Management, and the Bilateral Scientific and Technological Cooperation between China and the Flanders.
* Corresponding author. Tel.: +86 10 6277 2940.
E-mail address: chengq@sem.tsinghua.edu.cn (G. Chen).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.12.024

introduced temporal semantics into association rules, in forms of $X \Rightarrow Y \wedge P_1 \wedge P_2 \wedge \cdots \wedge P_n$ $(n \geqslant 0)$, where X and Y are itemsets, and $X \cap Y = \emptyset$. $P_1 \wedge P_2 \wedge \cdots \wedge P_n$ is a conjunction of binary temporal predicates. While mining a database D_T, a rule is accepted when its confidence factor $0 \leqslant c \leqslant 1$ is equal to or larger than the given threshold. Similarly, each predicate P_i is measured with a temporal confidence factor $0 \leqslant tc_{P_i} \leqslant 1$. The algorithm firstly generates the traditional association rules without considering the temporal factors, and then finds all of the possible pairings of temporal items in each rule. Subsequently, these pairings are tested so that strong temporal relationships could be found. Obviously, the complexity of this sequentially executed algorithm rises rapidly as the number of typical rules grows. Differently, Hoppner proposed another technique for discovering temporal patterns in state sequences. He defined the supporting level of a pattern as the total time in which the pattern can be observed within a sliding window, which should be predetermined by the user. However, a major concern for this technique is how to decide a proper size for the sliding window, since the sliding window can affect the mining results. Furthermore, The changes of the sliding window will lead to a sub-patterns check. The check requires some backtracking mechanism, which is computationally expensive. Like many existing data mining algorithms, the algorithm needs to scan the database repeatedly, which would significantly lower its efficiency.

This paper will focus on a particular type of temporal relationships, namely *during*, which represents that one event starts and ends within the duration of another event. Notably, this *during* relationship could reflect the temporal semantics of during, start, finish and equal described in Allen (1983). An approach will be proposed to discover the so-called *during*-temporal patterns (DTPs) in larger temporal databases, which are considered common and potentially valuable in real-world applications. One idea behind the approach is to design the corresponding algorithm so as to reduce the workload in scanning the database. In doing so, the database is partitioned into some disjoined datasets with two operations when calculating the support level of each pattern, so that scanning the whole database could be avoided. Furthermore, some properties of DTPs are investigated and then incorporated into the algorithm as pruning strategies to optimize the mining process for efficiency purposes.

The remainder of this paper is organized as follows. Section 2 formulates the problem and introduces related notions. In Section 3, the algorithmic details are provided, along with some of the related properties. The experiments on synthetic data and real weather data are discussed in Sections 4 and 5 concludes the paper.

2. The problem formulation

Let $\mathscr{A} = \{a_1, a_2, \ldots, a_m\}$ be a set of states, and $\mathscr{D}_{\mathscr{T}}$ a temporal database as shown in Table 1. Given a database

Table 1
A temporal database

Event	State	Starting time	Ending time
e_1	a_1	1	20
e_2	a_3	1	4
e_3	a_4	5	7
e_4	a_1	22	28
e_5	a_2	2	8
e_6	a_3	10	13
e_7	a_5	25	35
e_8	a_3	23	28
e_9	a_4	25	27
e_{10}	a_6	25	26
e_{11}	a_1	30	40
e_{12}	a_3	30	38
e_{13}	a_4	34	38
e_{14}	a_6	37	37

$\mathscr{D}_{\mathscr{T}}$ with N records, each of which is in the form of $\{a,(st,et)\}$ with respect to event e, i.e., $e = (a,t)$, where a is the state involved in the event, $t = (st,et)$ is the time interval which indicates starting time (st) and ending time (et) of state a in the event. A specific event is denoted as $e_l = (a_i, t_l)$ $(1 \leqslant l \leqslant N$ and $1 \leqslant i \leqslant m)$ and $t_l = (st_l, et_l)$, i.e., $S(e_l) = st_l$ and $E(e_l) = et_l$. For example, with $a_1 =$ rain, $e_1 = (a_1, (1, 20))$ in Table 1 means that it began to rain at 1:00 h and ended at 20:00 h.

Definition 1. Let $e_l = (a_i, t_l)$ and $e_k = (a_j, t_k)$ be two events in $\mathscr{D}_{\mathscr{T}}$. We call e_l *during* e_k (or e_k *contains* e_l), denoted as $e_l <^d e_k$, if

$$S(e_l) \geqslant S(e_k) \quad \text{and} \quad E(e_l) \leqslant E(e_k)$$

Generally, given a set of events $\{e_1, e_2, \ldots, e_k\}$, if e_{i+1} *during* e_i is satisfied for all $i = 1, 2, \ldots, k-1$, we have $e_k <^d e_{k-1} <^d \cdots <^d e_2 <^d e_1$.

For any two states a_i and a_j, a_i is called to be *during* a_j, denoted as pattern $a_i \Rightarrow^d a_j$, if state a_i occurs during the period of another state a_j, which is a *during*-temporal pattern (DTP) with length 1. Generally, a DTP of length $(k-1)$ $(k \geqslant 1)$, namely DTP_{k-1}, is of the form:

$$a_k \Rightarrow^d a_{k-1} \Rightarrow^d \cdots \Rightarrow^d a_2 \Rightarrow^d a_1$$

When the length is 0 (i.e., DTP_0), the pattern is a single state actually. More generally, given two patterns α and β, the form $\alpha \Rightarrow^d \beta$ is also a DTP ($A_\alpha \cap A_\beta = \emptyset$, where A_α and A_β are the sets of states included in pattern α and β respectively). As a special case, it retrogresses to $a_i \Rightarrow^d a_j$ when the lengths of both patterns are 0.

Given a pattern $a_k \Rightarrow^d a_{k-1} \Rightarrow^d \cdots \Rightarrow^d a_2 \Rightarrow^d a_1, e_k <^d e_{k-1} <^d \cdots <^d e_2 <^d e_1$ supports this pattern if a_i is the state of e_i for all $i = 1, 2, \ldots, k$. In the case, $e_k <^d e_{k-1} <^d \cdots <^d e_2 <^d e_1$ can be considered as an instance of this pattern. For example, in Table 1, $e_{10} <^d e_9 <^d e_8$ is an instance of the pattern $a_6 \Rightarrow^d a_4 \Rightarrow^d a_3$, and $e_{14} <^d e_{13} <^d e_{12}$ is another instance of this pattern. Further, a DTP α