Temporal Association Rule Mining in Event Sequences

Sherri K. Harms
University of Nebraska at Kearney, USA

INTRODUCTION

The emergence of remote sensing, scientific simulation and other survey technologies has dramatically enhanced our capabilities to collect temporal data. However, the explosive growth in data makes the management, analysis, and use of data both difficult and expensive. To meet these challenges, there is an increased use of data mining techniques to index, cluster, classify and mine association rules from time series data (Roddick & Spiliopoulou, 2002; Han, 2001). A major focus of these algorithms is to characterize and predict complex, irregular, or suspicious activity (Han, 2001).

BACKGROUND

A time series database contains sequences of values typically measured at equal time intervals. There are two main categories of temporal sequences: *transaction-based sequences* and *event sequences*. A transaction-based sequence includes an identifier such as a customer ID, and data mining revolves around finding patterns within transactions that have matching identifiers. An example pattern is "A customer who bought Microsoft and Intel stock is likely to buy Google stock later." Thus, the transaction has a definite boundary around known items of interest. There are many techniques that address these problems (Roddick & Spiliopoulou, 2002; Han, 2001).

Data analysis on event sequences is enormously more complex than transactional data analysis. There are no inherently defined boundaries around factors that might be of interest. The factors of interest themselves may not be obvious to domain experts. Temporal event sequence mining algorithms must be able to compute inference from volumes of data, find the interesting events involved, and define the boundaries around them. An example pattern is "A La Niña weather pattern is likely to precede drought in the western United States." La Niña weather data is based on Pacific Ocean surface temperatures and atmospheric values, and drought data is based on precipitation data from several weather stations located in the western United States. As illustrated by this example, sequential data analysis must be able to find relationships among multiple time series. The sheer number of possible combinations of interesting factors and relationships between them can easily overwhelm human analytical abilities. Often there is a delay between the occurrence of an event and its influence on the dependent variables. These factors make finding interesting patterns difficult.

One of the most common techniques to find interesting patterns is *association rule mining*. Association rules are implications between variables in the database. The problem was first defined in the context of the market basket data to identify customers' buying habits (Agrawal et al., 1993), where the Apriori algorithm was introduced. Let $I = I_1, I_2, \ldots, I_m$ be a set of binary attributes, called items. Let T be a database of transactions. An association rule r is an implication of the form $X \Rightarrow Y$, where X and Y are sets of items in I, and $X \cap Y = \varnothing$. X is the rule *antecedent*, Y is the rule *consequent*. *Support* of rule $X \Rightarrow Y$ in database T is the percentage of transactions in T that contain $X \cup Y$. The rule holds in T with *confidence c* if *c%* of the transactions in T that contain X also contain Y. For example, it is of interest to a supermarket to find that 80% of the transactions that contain milk also contain eggs and 5% of all transactions include both milk and eggs. Here the association rule is milk $\Rightarrow$ eggs, with 80% is the confidence of the rule and 5% support.

This paper provides the status of current temporal association rule mining methods used to infer knowledge for a group of event sequences. The goal of these tools is to find periodic occurrences of factors of interest, rather than to calculate the global correlation between the sequences. Mining association rules is usually decomposed into three sub-problems: 1) prepare the data for analysis, 2) find frequent patterns, and 3) generate association rules from the sets representing those frequent patterns.

MAIN THRUST

Events and Episodes

To prepare time series data for association rule mining, the data is discretized and partitioned into sequences of

Copyright © 2006, Idea Group Inc., distributing in print or electronic forms without written permission of IGI is prohibited.

events. Typically, the time series is normalized and segmented into partitions that have similar characteristics of data within a given interval. Each partition identifier is called an *event type*. Partitioning methods include symbolizing (Lin et al., 2003) and intervals (Hoppner, 2002). Different partitioning methods and interval sizes produce diverse discretized versions of the same dataset. This step relies on domain-expert involvement for proper discretization. When multivariate sequences are used, each variable is normalized and discretized independently. The time granularity (duration) is converted to a single (finest) granularity before the discovery algorithms are applied to the combined sequences (Bettini et al., 1998).

A discretized version of the time series is referred to as an *event sequence*. An event sequence $\hat{S}$ is a finite, time-ordered sequence of events (Mannila et al., 1995). That is, $\hat{S} = (e_1, e_2, ...e_n)$. An event is an occurrence of an event type at a given timestamp. The time that a given event e_i occurs is denoted *i,* and $i \leq i+1$ for all *i* timestamps in the event sequence. A sequence includes events from a single finite set of event types. An event type can be repeated multiple times in a sequence. For example, the event sequence $\hat{S}_1 = AABCAB$ is a sequence of 6 events, from a set of 3 event types {A,B,C}. In this event sequence, an A event occurs at time 1, followed by another A event, followed by a B event, and so on. The step size between events is constant for a given sequence.

An *episode* in an event sequence is a combination of events with partially specified order (Mannila et al., 1997). It occurs in a sequence if there are occurrences of events in an order consistent with the given order, within a given time bound (window width). Formally, an episode a is a pair *(V, ordering),* where *V* is a collection of events and the ordering is *parallel* if no order is specified, and *serial* if the events of the episode have fixed order. The episode length is defined as the number of events in the episode.

Finding Frequent Episodes Based on Sliding Window Technologies

The founding work on finding frequent episodes in sequences is Mannila et al. (1995). Frequent episodes are discovered by using a sliding window approach, *WINEPI*. A *window* on an event sequence $\hat{S}$ is an event subsequence, $w = e_i, e_{i+1}, ...e_{i+d}$ where the width of window *w*, denoted *d,* is the time interval of interest. The set of all windows *w* on $\hat{S}$, with a width of *d* is denoted $\hat{W}(\hat{S},d)$. In this system, the value of the window width is user-specified, varying the closeness of event occurrences. To process data, the algorithm sequentially slides the window of width *d* one step at a time through the data. The *frequency* of an episode a is defined as the fraction of windows in which the episode occurs. For example, in the sequence $\hat{S}_1$ above, if a sliding window of width 3 is used, serial episode a = AB occurs in the first window (AAB), the second window (ABC), and the fourth window (CAB).[1] The guiding principle of the algorithm lies in the "downward-closed'' property of frequency, which means every subepisode is at least as frequent as its superepisode (Mannila et al., 1995). As with the Apriori method, candidate episodes with *(k+1)* events are generated by joining frequent episodes that have *k* events in common, and episodes that do not meet a user-specified frequency threshold are pruned.

The *WINEPI* algorithm was improved by Harms et al. (2001) to use only a subset of frequent episodes, called *frequent closed episodes,* based on closures and formal concept analysis (Wille, 1982). A frequent closed episode *X* is the intersection of all frequent episodes containing *X*. For example, in the $\hat{S}_1$ sequence, using a window width *d = 3,* and a minimum frequency of three windows, serial episode α= AB is a frequent closed episode since no larger frequent episode contains it[2], and it meets the minimum frequency threshold. Using closed episodes results in a reduced input size and in a faster generation of the episodal association rules, especially when events occur in clusters. Harms et al. (2001) use an inclusion constraint set to target specific subsets of episodes.

In Hoppner (2002), multivariate sequences are divided into small segments and discretized based on their qualitative description (such as increasing, high value, convexly decreasing, etc.). Patterns are discovered in the interval sequences based on Allen's temporal interval logic (Allan, 1983). For example, the pattern "A meets B" occurs if interval A terminates at the same point in time at which B starts. For any pair of intervals there is a set of 13 possible relationships, including after, before, meets, is-met-by, starts, is-started-by, finishes, is-finished-by, overlaps, is-overlapped-by, during, contains, and equals. As with *WINEPI*, this approach finds frequent patterns by using sliding windows and creating a set of candidate *(k+1)*-patterns from the set of frequent patterns of size *k*.

An approach to detecting suspicious subsequences in event sequences is presented in Gwadera et al. (2003). Using an approach based on *WINEPI*, they quantify: 1) the probability of a suspicious subsequence $\acute{s}$ occurring in a sequence $\hat{S}$ of events within a window of size *d,* 2) the number of distinct windows containing $\acute{s}$ as a subsequence, 3) the expected number of such occurrences, and 4) the variance of the subsequence $\acute{s}$. They also establish its limiting distribution that allows users to set an alarm threshold so that the probability of false alarms is small.

Ng & Fu (2003) presented a method to mine frequent episodes using a tree-based approach for event sequences. The process is comprised of two phases: 1) tree construction and 2) mining frequent episodes. Each

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/temporal-association-rule-mining-event/10760

Related Content

Mining in Spatio-Temporal Databases
Junmei Wang, Wynne Hsuand Mong Li Lee (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications (pp. 3477-3492).*
www.irma-international.org/chapter/mining-spatio-temporal-databases/7844

A Porter Framework for Understanding the Strategic Potential of Data Mining for the Australian Banking Industry
Kate A. Smithand Mark S. Dale (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications (pp. 2772-2791).*
www.irma-international.org/chapter/porter-framework-understanding-strategic-potential/7799

World Wide Web Usage Mining
Wen-Chen Hu, Hung-Jen Yang, Chung-wei Leeand Jyh-haw Yeh (2005). *Encyclopedia of Data Warehousing and Mining (pp. 1242-1248).*
www.irma-international.org/chapter/world-wide-web-usage-mining/10788

Artificial Neural Networks for Prediction
Rafael Marti (2005). *Encyclopedia of Data Warehousing and Mining (pp. 54-58).*
www.irma-international.org/chapter/artificial-neural-networks-prediction/10565

Discovering Unknown Patterns in Free Text
Jan H. Kroeze (2005). *Encyclopedia of Data Warehousing and Mining (pp. 382-386).*
www.irma-international.org/chapter/discovering-unknown-patterns-free-text/10627