Sequential Pattern Mining

Florent Masseglia
INRIA Sophia Antipolis, France

Maguelonne Teisseire
University of Montpellier II, France

Pascal Poncelet
Ecole des Mines d'Alès, France

INTRODUCTION

Sequential pattern mining deals with data represented as sequences (a sequence contains sorted sets of items). Compared to the association rule problem, a study of such data provides "inter-transaction" analysis (Agrawal & Srikant, 1995). Applications for sequential pattern extraction are numerous and the problem definition has been slightly modified in different ways. Associated to elegant solutions, these problems can match with real-life timestamped data (when association rules fail) and provide useful results.

BACKGROUND

In (Agrawal & Srikant, 1995) the authors assume that we are given a database of customer's transactions, each of which having the following characteristics: sequence-id or customer-id, transaction-time and the item involved in the transaction. Such a database is called a base of data sequences. More precisely, each transaction is a set of items (itemset) and each sequence is a list of transactions ordered by transaction time. For efficiently aiding decision-making, the aim is to obtain typical behaviors according to the user's viewpoint. Performing such a task requires providing data sequences in the database with a support value giving its number of actual occurrences in the database. A frequent sequential pattern is a sequence whose statistical significance in the database is above user-specified threshold. Finding all the frequent patterns from huge data sets is a very time-consuming task. In the general case, the examination of all possible combination is intractable and new algorithms are required to focus on those sequences that are considered important to an organization.

Sequential pattern mining is applicable in a wide range of applications since many types of data are in a time-related format. For example, from a customer purchase database a sequential pattern can be used to develop marketing and product strategies. By way of a Web Log analysis, data patterns are very useful to better structure a company's website for providing easier access to the most popular links (Kosala & Blockeel, 2000). We also can notice telecommunication network alarm databases, intrusion detection (Hu & Panda, 2004), DNA sequences (Zaki, 2003), etc.

MAIN THRUST

Definitions related to the sequential pattern extraction will first be given. They will help understanding the various problems and methods presented hereafter.

Definitions

The item is the basic value for numerous data mining problems. It can be considered as the object bought by a customer, or the page requested by the user of a website, etc. An itemset is the set of items that are grouped by timestamp (e.g. all the pages requested by the user on June 04, 2004). A data sequence is a sequence of itemsets associated to a customer. In table 1, the data sequence of C2 is the following: "(Camcorder, MiniDV) (DVD Rec, DVD-R) (Video Soft)" which means that the customer bought a *camcorder* and *miniDV* the same day, followed by a *DVD recorder* and *DVD-R* the day after, and finally a *video software* a few days later.

A sequential pattern is included in a data sequence (for instance "(MiniDV) (Video Soft)" is included in the data sequence of C2, whereas "(DVD Rec) (Camcorder)" is not included according to the order of the

Table 1. Data sequences of four customers over four days

Cust	June 04, 2004	June 05, 2004	June 06, 2004	June 07, 2004
C1	Camcorder, MiniDV	Digital Camera	MemCard	USB Key
C2	Camcorder, MiniDV	DVD Rec, DVD-R		Video Soft
C3	DVD Rec, DVD-R	MemCard	Video Soft	USB Key
C4		Camcorder, MiniDV	Laptop	DVD Rec, DVD-R

timestamps). The minimum support is specified by the user and stands for the minimum number of occurrences of a sequential pattern to be considered as frequent. A maximal frequent sequential pattern is included in at least "minimum support" data sequences and is not included in any other frequent sequential pattern. Table 1 gives a simple example of 4 customers and their activity over 4 days in a shop. With a minimum support of "50%" a sequential pattern can be considered as frequent if it occurs at least in the data sequences of 2 customers (2/4). In this case a maximal sequential pattern mining process will find three patterns:

- **S1:** "(Camcorder, MiniDV) (DVD Rec, DVD-R)"
- **S2:** "(DVD Rec, DVD-R) (Video Soft)"
- **S3:** "(Memory Card) (USB Key)"

One can observe that S1 is included in the data sequences of C2 and C4, S2 is included in those of C2 and C3, and S3 in those of C1 and C2. Furthermore the sequences do not have the same length (S1 has length 4, S2 has length 3 and S3 has length 2).

Methods for Mining Sequential Patterns

The problem of mining sequential patterns is stated in (Agrawal & Srikant, 1995) and improved, both for the problem and the method, in (Srikant & Agrawal, 1996). In the latter, the GSP algorithm is based on a breadth-first principle since it is an extension of the A-priori model to the sequential aspect of the data. GSP uses the "Generating-Pruning" method defined in (Agrawal, Imielinski, & Swami, 1993) and performs in the following way. A candidate sequence of length (k+1) is generated from two frequent sequences, s_1 and s_2, having length k, if the subsequence obtained by pruning the first item of s_1 is the same as the subsequence obtained by pruning the last item of s_2. With the example in Table 1, and k=2, let s_1 be "(DVD Rec, DVD-R)" and s_2 be "(DVD-R) (Video Soft)", then the candidate sequence will be "(DVD Rec, DVD-R) (Video Soft)" since the subsequence described above (common to s_1 and s_2) is "(DVD-R)". Another method based on the Generating-Pruning principle is PSP (Massegli, Cathala, & Poncelet, 1998). The main difference to GSP is that the candidates as well as the frequent sequences are managed in a more efficient structure. The methods presented so far are designed to depend as little as possible on main memory. The methods presented thereafter need to load the database (or a rewriting of the database) in main memory. This results in efficient methods when the database can fit into the memory.

In (Zaki, 2001), the authors proposed the SPADE algorithm. The main idea in this method is a clustering of the frequent sequences based on their common prefixes and the enumeration of the candidate sequences, thanks to a rewriting of the database (loaded in main memory). SPADE needs only three database scans in order to extract the sequential patterns. The first scan aims at finding the frequent items, the second at finding the frequent sequences of length 2 and the last one associate to frequent sequences of length 2, a table of the corresponding sequences id and itemsets id in the database (e.g. data sequences containing the frequent sequence and the corresponding timestamp). Based on this representation in main memory, the support of the candidate sequences of length k is the result of join operations on the tables related to the frequent sequences of length (k-1) able to generate this candidate (so, every operation after the discovery of frequent sequences having length 2 is done in memory). SPAM (Ayres, Flannick, Gehrke, & Yiu, 2002) is another method which needs to represent the database in the main memory. The authors proposed a vertical bitmap representation of the database for both candidate representation and support counting.