Sequential Pattern Mining

Florent Masseglia
INRIA Sophia Antipolis, France

Maguelonne Teisseire
University of Montpellier II, France

Pascal Poncelet
Ecole des Mines d' Alè, 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 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

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

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 (Masseglí, 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.

An original approach for mining sequential patterns aims at recursively projecting the data sequences into smaller databases. Proposed in (Han, et al., 2000), FreeSpan is the first algorithm considering the pattern-projection method for mining sequential patterns. This work has been continued with PrefixSpan, (Pei, et al., 2001), based on a study about the number of candidates proposed by a Generating-Pruning method. Starting from the frequent items of the database, PrefixSpan generates projected databases with the remaining data-sequences. The projected databases thus contain suffixes of the data-sequences from the original database, grouped by prefixes. The process is recursively repeated until no frequent item is found in the projected database. At this level the frequent sequential pattern is the path of frequent items driving to this projected database.

Closed Sequential Patterns

A closed sequential pattern is a sequential pattern included in no other sequential pattern having exactly the same support. Let us consider the database illustrated in Table 1. The frequent sequential pattern "(DVD Rec) (Video Soft)" is not closed because it is included in the sequential pattern S2 which has the same support (50%).

On the other hand, the sequential pattern "(Camcorder, MiniDV)" (with a support of 75%) is closed because it is included in other sequential patterns but with a different support (for instance, S1, which has a support of 50%). The first algorithm designed to extract closed sequential patterns is CloSpan (Yan, Han, & Afshar, 2003) with a detection of non-closed sequential patterns avoiding a large number of recursive calls. CloSpan is based on the detection of frequent sequences of length 2 such that "*A* always occurs before/after *B*". Let us consider the database given in Table 1. We know that "(DVD Rec) (Video Soft)" is a frequent pattern. The authors of CloSpan proposed relevant techniques to show that "(DVD-R)" always occurs before "(Video Soft)". Based on this observation CloSpan is able to find that "(DVD Rec, DVD-R) (Video Soft)" is frequent without anymore scans over the database. BIDE (Wang & Han, 2004) extends the previous algorithm in the following way. First, it adopts a novel sequence extension, called BI-Directional Extension, which is used both to grow the prefix pattern and to check the closure property. Second, in order to prune the search space more deeply than previous approaches, it proposes a BackScan pruning method. The main idea of this method is to avoid extending a sequence by detecting in advance that the extension is already included in a sequence.

Incremental Mining of Sequential Patterns

As databases evolve, the problem of maintaining sequential patterns over a significantly long period of time becomes essential since a large number of new records may be added to a database. To reflect the current state of the database, in which previous sequential patterns would become irrelevant and new sequential patterns might appear, new efficient approaches were proposed. (Masseglia, Poncelet, & Teisseire, 2003) proposes an efficient algorithm, called ISE, for computing the frequent sequences in the updated database. ISE minimizes computational costs by re-using the minimal information from the old frequent sequences, i.e. the support of frequent sequences. The main new feature of ISE is that the set of candidate sequences to be tested is substantially reduced. The SPADE algorithm was extended into the ISM algorithm (Parthasarathy, Zaki, Ogihara, & Dwarkadas., 1999). In order to update the supports and enumerate frequent sequences, ISM maintains "maximally frequent sequences" and "minimally infrequent sequences" (also known as negative border). KISP (Lin and Lee, 2003) also proposes to take advantage of the knowledge previously computed and generates a knowledge base for further queries about sequential patterns of various support values.

Extended Problems Based on the Sequential Pattern Extraction

Motivated by the potential applications for the sequential patterns, numerous extensions of the initial definition have been proposed which may be related to the addition of constraints or to the form of the patterns themselves. In (Pei, Han, & Wang, 2002) the authors enumerate some of the most useful constraints for extracting sequential patterns. These constraints can be considered as filters applied to the extracted patterns, but most methods generally take them into account during the mining process. These filters may concern the items ("extract patterns containing the item *Camcorder* only") or the length of the pattern, regular expressions describing the pattern, and so on. The definition of the sequential patterns has also been adapted by some research work. For instance (Kum, Pei, Wang, & Duncan, 2003) proposed ApproxMap to mine approximate sequential patterns. ApproxMap first proposes to cluster the data sequences depending on their items. Then for each cluster ApproxMap allows extraction of the approximate sequential patterns related to this cluster. Let us consider the database in Table 1 as a cluster. The first step of the extraction process is to provide the data sequences of the cluster with an alignment similar to those of bioinformatics. Table 2 illustrates such an alignment.

The last sequence in Table 2 represents the weighted sequence obtained by ApproxMap on the sequences of Table 1. With a support of 50%, the weighted sequence gives the following approximate pattern: "(Camcorder: 3, MiniDV: 3) (DVD Rec: 3, DVD-R: 3) (MemCard: 2) (Video Soft: 2) (USB Key: 2)". It is interesting to observe that this sequential pattern does not correspond to any of the recorded behavior, whereas it represents a trend for this kind of customer.

FUTURE TRENDS

Today several methods are available for efficiently discovering sequential patterns according to the initial definition. Such patterns are widely applicable for a large number of applications. Specific methods, widely inspired from previous algorithms, exist in a wide range of domains. Nevertheless, existing methods have to be reconsidered since handled data is much more complex. For example, existing algorithms consider that data is binary and static. Today, according to the huge volume of data available, stream data mining represents an emerging class of data-intensive applications where data flows in and out dynamically. Such

Table 2. Alignment proposed for the data sequences of Table 1.

Camcorder, MiniDV	DigiCam		MemCard		USB Key
Camcorder, MiniDV		DVD Rec, DVD-R		Video Soft	
		DVD Rec, DVD-R	MemCard	Video Soft	USB Key
Camcorder, MiniDV	Laptop	DVD Rec, DVD-R			
Camcorder: 3 MiniDV: 3	DigiCam: 1 Laptop: 1	DVD Rec: 3 DVD-R: 3	MemCard: 2	Video Soft: 2	USB Key: 2

applications also need very fast or even real-time responses (Giannella, Han, Pei, Yan, & Yu, 2003; Cai et al., 2004). In order to increase the immediate usefulness of sequential rules, it is very important to consider much more information. Hence, by associating sequential patterns with a customer category or multi-dimensional information, the main objective of multi-dimensional sequential pattern mining is to provide the end-user with more useful classified patterns (Pinto et al., 2001). With such patterns, an auto-dealer would find, for example, an enriched sequential rule stating that "*Customers who bought an SUV on monthly payment installments 2 years ago are likely to respond favorably to a trade-in now*".

CONCLUSION

Since they have been defined in 1995, sequential patterns have received a great deal of attention. First work on this topic focused on improving the efficiency of the algorithms either with new structures, new representations or by managing the database in the main memory. More recently extensions were proposed by taking into account constraints associated with real life applications. In fact, the increasing contributions on sequential pattern mining are mainly due to their adaptability to such applications. The management of timestamp within the recorded data is a difficulty for designing algorithms; on the other hand this is the reason why sequential pattern mining is one of the most promising technologies for the next generation of knowledge discovery problems.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington, D.C, USA, 207-216.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceeding of the 11th International Conference on Data Engineering*, Taipei, Taiwan, 3-14.

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential pattern mining using bitmap representation. *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, Alberta, Canada, 429-435.

Cai, Y., Clutter, D., Pape, G., Han, J., Welge, M., & Auvil, L. (2004). MAIDS: Mining alarming incidents from data streams. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Paris, France, 919-920.

Giannella, G., Han, J., Pei, J., Yan, X., & Yu, P. (2003). Mining frequent patterns in data Streams at multiple time granularities. In H. Kargupta, A. Joshi, K. Sivakumar & Y. Yesha (Eds.), *Next Generation Data Mining* (chap. 3), MIT Press.

Han J., Pei J., Mortazavi-asl B., Chen Q., Dayal U., & Hsu, M. (2000). FreeSpan: Frequent pattern-projected sequential pattern mining. *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, Boston, USA, 355-359.

Hu, Y., & Panda, B. (2004). A Data mining approach for database intrusion detection. *Proceedings of the 19 th ACM Symposium on Applied Computing*, Nicosia, Cyprus, 711-716.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations*, *2*(1), 1-15.

Kum, H.-C., Pei, J., Wang, W., & Duncan, D. (2003). ApproxMAP: Approximate mining of consensus sequential patterns. *Proceedings of the 3rd SIAM International Conference on Data Mining*, San Francisco, CA, 311-315.

Lin, M., & Lee, S. (2003). Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Big Island, USA, CDROM, 68.

Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP approach for mining sequential patterns. *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery*, Nantes, France, 176-184.

Masseglia, F., Poncelet, P., & Teisseire, M. (2003). Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering*, *46*(1), 97-121.

Parthasarathy, S., Zaki, M., Ogihara, M., & Dwarkadas, S. (1999). Incremental and interactive sequence mining. *Proceedings of the 8th International Conference on Information and Knowledge Management*, Kansas City, USA, 251-258.

Pei, J., Han, J., Mortazavi-asl, B., Pinto, H., Chen, Q., Dayal, U., et al. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of 17th International Conference on Data Engineering*, Heidelberg, Germany, 215-224.

Pei, J., Han, J., & Wang, W. (2002). Mining sequential patterns with constraints in large databases. *Proceedings of the11th Conference on Information and Knowledge Management*, McLean, USA, 18-25.

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q., & Dayal, U. (2001). Multi-dimensional sequential pattern mining. *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, USA, 81-88.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Proceeding of the 5th International Conference on Extending Database Technology,* Avignon, France, 3-17.

Wang, J., & Han, J., (2004). BIDE: Efficient mining of frequent closed sequences. *Proceedings of the 20th International Conference of Data Engineering*, Boston, USA, 79-90.

Yan X., Han, J., & Afshar, R. (2003). CloSpan: Mining closed sequential patterns in large databases. *Proceedings of the 3rd SIAM International Conference on Data Mining*, San Francisco, CA.

Zaki, M. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning, 42*(1/2), 31-60.

Zaki, M. (2003). Mining data in bioinformatics. In Nong Ye (Ed.) *Handbook of Data Mining*, Lawrence Earlbaum Associates, 573-596.

KEY TERMS

Apriori: The method of generating candidates before testing them during a scan over the database, insuring that if a candidate may be frequent then it will be generated. See also *Generating-Pruning*.

Breadth-First: The method of growing the intermediate result by adding items both at the beginning and the end of the sequences. See also *Generating-Pruning*

Closed Sequential Pattern: A frequent sequential pattern that is not included in another frequent sequential pattern having exactly the same support.

Data Sequence: The sequence of itemsets representing the behavior of a client over a specific period. The database involved in a sequential pattern mining process is a (usually large) set of data sequences.

Depth-first: The method of generating candidates by adding specific items at the end of the sequences. See also *Generating-Pruning*.

Generating-Pruning: The method of finding frequent sequential patterns by *generating* candidates sequences (from size 2 to the maximal size) step by step. At each step a new generation of candidates having the same length is generated and tested over the databases. Only frequent sequences are kept (*pruning*) and used in the next step to create a new generation of (longer) candidate sequences.

Itemset: Set of items that occur together.

Maximal Frequent Sequential Pattern: A sequential pattern included in at least *n* data sequences (with *n* the minimum support specified by the user). A sequential pattern is maximal when it is not included in another frequent sequential pattern. A frequent sequential pattern may represent, for instance, a frequent behavior of a set of customers, or a frequent navigation of the users of a Web site.

Negative Border: The collection of all sequences that are not frequent but both of whose generating sub sequences are frequent.

Sequential Pattern: A sequence included in a data sequence such that each item in the sequential pattern appears in this data sequence with respect to the order between the itemsets in both sequences.