Asian Journal of Information Technology 11 (3): 94-99, 2012
ISSN: 1682-3915

STEM: A Novel Approach for Spatiotemporal Sequence Mining

[1]Kelvin Leong and [2]Stephen Chan
[1]School of Accounting and Finance, [2]Department of Computing,
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract: Building on the skeleton of Generalized Sequential Pattern (GSP), researchers propose a new approach-Spatio-Temporal Events Miner (STEM) for sequential pattern analysis. The STEM extends the traditional finding by coverage of both temporal and spatial attributes. Furthermore, researchers are interested in studying whether AprioriAll Method in traditional GSP is most suitable in STEM.

Key words: Skelton, GSP, STEM, attributes, analysis

INTRODUCTION

Sequential pattern mining has been widely studied in many real life applications such as stock market analysis, climate prediction, diseases control, sales forecasting, etc. (Ramirez *et al*., 1998, 2000). Since, the sequential pattern has been introduced in (Agrawal and Srikant, 1995), many new initiatives has been proposed by different researchers such as SPADE algorithm (Zaki, 2001) 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. FreeSpan (Han *et al*., 2000) is the first algorithm considering the pattern-projection method for mining sequential patterns. PrefixSpan (Pei *et al*., 2001) mainly employs the method of database projection to make the database for next pass much smaller and consequently make the algorithm faster, also in PrefixSpan there is no need for candidate's generation only recursively project the database according to their prefix. In detail Masseglia *et al*. (2005) and Zhao and Bhowmick (2003) has made a summary for different approaches. However, all these new initiatives were focus on how to improve the effectiveness of finding sequential pattern; none of them were studying its application in spatiotemporal databases.

Traditional sequential pattern mining is the mining of frequently occurring patterns related to time or other sequence among different agents. Researchers use the term agent, it refers any person (or object) that causes an event. For example:

- Case 1 Customer buys Narnia from bookshop
- Case 2 Grave sweepers cause hill fires

In Case 1, Customer is an agent (buy Narnia) is an event. In Case 2, Grave sweepers is an agent (hill fire) is an event. In simplicity, the approach-STEM use the location-id to replace agent-id thus spatial data can be incorporated. In other words, the traditional sequential pattern indicates agent based sequence while STEM represents location based sequence.

RELATED WORKS

Discovering event's pattern by data mining: Data mining (Chen *et al*., 1996) is the process of extracting interesting non-trivial, implicit, previously unknown and potentially useful information or patterns from large information repositories. Association Rule (AR) mining (Agrawal *et al*., 1993) aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. AR has been applied widely on transactional, spatial and temporal data mining. It is also the foundation of sequential pattern mining.

Sequential Pattern (SP) mining was first introduced in (Agrawal and Srikant, 1995). Sequence mining is focused on finding statistically relevant patterns between data examples where the values are delivered in a sequence. AR and SP mining are important in this study because they are core techniques in DEEP mining.

Temporal association rule: Temporal Association Rules (TAR) (Wang *et al*., 2001) intend to describe the temporal changes of attribute values of some object histories. An attribute evolution is E (A_i) over m consecutive snapshots, denoted by:

Corresponding Author: Kelvin Leong, School of Accounting and Finance, the Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

$$E(A_i) = (A_i [l_1,u_1] \text{-->} A_i [l_2,u_2] \text{ ....} A_i [l_m,u_m])$$

It represents the change of attribute A_i over m consecutive snapshots where $[l_m, u_m]$ is the interval of values of A_i. The u and l stand for upper and lower interval, respectively. For example, an employee's salary changes from an interval of $40,000 and 45,000 to an interval of $50,000-55,000 and then to an interval $60,000-65,000. Let A_1, A_2 ... A_n be a subset of attributes. A temporal association rule R of length m is defined as:

$$E(A_1) \cap E(A_2) \cap ... E(A_{k-1}) \cap E(A_{k+1}) ... \cap E(A_n) = > E(A_k)$$

and $E(A_i)$ is an evolution of length m with attributes A_i ($1 <= i <= n$). TAR has been used in crime analysis (Ng *et al*., 2007). However, the limitation of TAR is the mined attributes must be in numerical form. Thus, users have to change the original narrative attributes into numerical form.

Spatio-temporal sequential patterns mining on single object: In (Cao *et al*., 2005), first transforms the original sequence into a list of sequence segments and detects frequent regions in a heuristic way. Then, use the proposed substring tree structure and improving Apriori technique to find frequently repeated paths of an object. However, the algorithm is more focus on tracing an individual object movement history. This is not applicable to discover the relationship of different events over the time and space.

Topological pattern: Given a distance threshold D and time window threshold W, topological pattern framework (Wang *et al*., 2005) is able to find the intra-relationships of events within a time window. Topological patterns are set of collocated features that satisfy additional pre-defined spatial relationships. This is not only satisfying the spatial proximity relationship but also the temporal proximity relationship. Even though it can research out spatio-temporal pattern, it is still unable to disclose the inter-relationship of events in different time windows.

Flow pattern and generalized spatio-temporal pattern: Flow pattern and Generalized Spatio-Temporal pattern (GST pattern) are discussed by Wang *et al*. (2004) and Hsu *et al*. (2005). These two algorithms intend to describe how one event in some location implies the occurrence of another event in a second location or how changes of event in one location can affect the events in another location. A flow pattern is a sequence of reflexive eventsets sorted by time such that for any two consecutive eventsets while GST patterns emphasis relative addresses, this is a sequence of RelativeEventsets and all the RelativeEventsets are CloseNeighbors of each other. A sequence is a list of related eventsets sorted by time within a time window. Let e is event, l is location, t is timestamp and R is threshold of related. Two events $e_1 (l_1, t_1)$ and $e_2 (l_2, t_2)$, $t_1 \leq t_2$ are said to be related or Closed Neighbors if and only if $(l_1, l_2) \in R$ and t_1 is near t_2. A set of location-based events that occur at the same time is called and eventset, denoted as $E = < e_1 (l_1, t_1) ... e_m (l_m, t_m) >$. Two eventsets E_1 and E_2 are said to be CloseNeighbors if and only if every event in E_1 is related to every event in E_2. However, these two algorithms link the changes in one location to the changes in a nearby location. It means non-CloseNeighbors events will not be considered.

Sequential pattern mining: Sequential pattern indicate the correlation between transactions while association rule represents intra transaction relationship. For example in a supermarket database, the results of association rule mining are about which items are brought together frequently and those items must come from the same transaction. While the results of sequential pattern mining are about which items are brought in a certain order by the same customer, those items come from different transactions. An improved sequential pattern is generalized sequential patterns.

Generalized sequential patterns: Generalized Sequential Patterns (GSP) is first described by Srikant and Agrawal (1996). Sequential pattern mining is the process of extracting certain sequential patterns whose support exceeds a predefined minimal support threshold. The support for a sequence is defined as the fraction of total data-sequences that contain this sequence. As per (Agrawal and Srikant, 1995), a data-sequence contains a sequence s if s is a subsequence of the data-sequence. Given a database D of sequences called data-sequences. Each data-sequence is a list of transactions, ordered by increasing transaction-time. $I = I_1, I_2, ... ,I_m$ be a set of attributes called items, T be transaction which contains a set of items from I.

Lets an itemset a non-empty set of items. A sequence is an ordered list of itemsets. A sequence s is denoted by $<s_1 s_2 ... s_n>$ where s_j is an itemset. Researchers also call s_j an element of sequence. An element of a sequence is denoted by $(x_1, x_2 ... x_m)$ where x_j is an item. An item can occur only once in an element of a sequence but can occur multiple times in different elements. GSP is more comprehensive than SP because it integrates with time constraints and relaxes the definition of transaction also

it considers the knowledge of taxonomies. In order to include the core test and 3 additional conditions, GSP involves 4 processes.

Process 1 subsequence test: A sequence $<a_1a_2...a_n>$ is contained in another sequence $<b_1b_2...b_n>$ if there exist integers $i_1<i_2<...<i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, ... ,$a_n \subseteq b_{in}$. For example, the sequence <(3) (6; 7; 9) (7; 9)> is contained in <(2) (3) (6; 7; 8; 9) (7) (7; 9)>, since (3)⊆(3), (6, 7, 9)⊆ (6, 7, 8, 9), (7, 9)⊆(7,9). However, sequence <(2)(3)> is not contained in sequence <(2; 3)>, since the former sequence means (3) is bought after (2) being bought while the latter represents item 2 and 3 being bought together.

Process 2 plus taxonomies: Many datasets have a user-defined taxonomy (is a hierarchy) over the items in the data and users want to find patterns that include items across different levels of the taxonomy. A transaction T contains an item $x \in I$ if x is in T or x is an ancestor of some item in T. Researchers say that a transaction T contains an itemset $y \subseteq I$ if T contains every item in y. A data-sequence $d = <d_1...d_m>$ contains a sequence $s = <s_1...s_n>$ if there exist integers $i_1<i_2<...<i_n$ such that s_1 is contained in d_{i1}, s_2 is contained in d_{i2},..., s_n is contained in d_{in}. If there is no taxonomy, this degenerates into a simple subsequence test.

Process 3 plus sliding windows: The sliding window generalization relaxes the definition of when a data-sequence contributes to the support of a sequence by allowing a set of transactions to contain an element of a sequence as long as the difference in transaction-times between the transactions in the set is less than the user-specified window-size. For example, if the book-club specifies a time window of a week, a customer who ordered the Foundation on Monday, Ringworld on Saturday and then Foundation and Empire and Ringworld Engineers in a single order a few weeks later would still support the pattern Foundation and Ringworld followed by Foundation and Empire and Ringworld engineers. Formally, a data-sequence $d = <d_1...d_m>$ contains a sequence $s = <s_1...s_n>$ if there exist integers $l_1 \le u_1 < l_2 \le u_2 < ... < l_n \le u_n$ such that:

- s_i is contained in $U^{ui}_{k=li}\ d_k$, $1 \le i < n$
- Transaction-time (d_{ui})-transaction-time (d_{li})≤ Window-size, $1 \le i \le n$

Process 4 plus time constraints: Time constraints restrict the time gap between sets of transactions that contain consecutive elements of the sequence. For example, a book club probably does not care if someone bought Foundation followed by foundation and empire 3 years later they may want to specify that a customer should support a sequential pattern only if adjacent elements occur within a specified time interval, say 3 months (So for a customer to support this pattern, the customer should have bought Foundation and Empire within 3 months of buying Foundation).

Given user-specified window-size, max-gap and min-gap, a data-sequence $d = <d_1...d_m>$ contains a sequence $s = <s_1...s_n>$ if there exist integers $l_1 \le u_1 < l_2 \le u_2 < ... < l_n \le u_n$ such that:

- s_i is contained in $U^{ui}_{k=li}\ d_k$, $1 \le i < n$
- Transaction-time (d_{ui})-transaction-time (d_{li})≤ Window-size, $1 \le i \le n$
- Transaction-time (d_{li})-transaction-time (d_{ui-1}) >min-gap, $2 \le i \le n$
- Transaction-time (d_{ui})-transaction-time (d_{li-1})≤Max-gap, $2 \le i \le n$

The first two conditions are the same as in the earlier definition of when a data-sequence contains a pattern. The third condition specifies the minimum time-gap constraint and the last the maximum time-gap constraint.

STEM

Conventional sequential pattern indicate agent based sequence while STEM represents location based sequence. Researchers propose STEM as a new approach for finding events pattern over the time and space. This algorithm is derived from GSP.

Problem statement: Given a database contains three fields in each transaction Location id, time and events. Where the location id represents the location of event's occur. In real application, it could be in different forms such as (x, y) coordinate, branches or districts, etc. The Time is timestamp of event occurred in the example, they are in date form. The last field is event-set, an event-sets is a non-empty set of events. It refers what kinds of event are occurring in specified time and specified location id. Researchers do not consider quantities of event occurred in a day thus it is a Boolean variable representing whether an event is occurred or not. Researchers are interested in discovering the frequent spatio-temporal sequence of events. The sequence is an ordered list of event-sets. The event-sets in the sequence are not necessarily contiguous.

Definitions: Let $T = (T_1, T_2,...,T_i)$ be the ordering timestamp where $1<2...<i$. Let $E = (E_1, E_2,...,E_m)$ be an event-set. An event-set contains a set of event e. Researchers also let $L = (L_1, L_2,...,L_m)$ be a location set of

the event occurs where 1< >2< >...< >m. Let s be a sequence. This is a list of transactions from same location, ordered by increasing transaction time where each of transaction is a set of events. Researchers call this sequence a location sequence. The length of a sequence is the number of event-sets in the sequence. A sequence of length k is called a k-sequence. A location supports s if s is contained in the location-sequence for this location. The support for a sequence is defined as fraction of total locations where support this sequence. Out interest is to find the maximal sequences among all sequences that have a certain user-specified minimum support.

Example: In Table 1, a database has been sorted on location id and time. Assume researchers want to find the frequent sequence that with minimum support set to 25% (i.e., a minimum support of 2 locations) and two-sequences. Firstly, researchers convert the original database from Table 1 to the format of location-sequence database as at Table 2. Based on the location-sequence in Table 2, assume the minimum support is 25% researchers obtain two desired frequent 2-sequences: <(B) (C)>, <(A B) (F)>. The first sequence <(B) (C)> is supported by location id 2, 3 and 5 while the second sequence <(A B) (F)> is supported by location id 2 and 4.

Sequential patterns with support >25%:

- < (B) (C)>
- < (A B) (F)>

Table 1: Database sorted by location id, transaction time and events

Location id	Time	Events set
1	23-Dec-07	A
1	24-Dec-07	G
2	23-Dec-07	A, B
2	26-Dec-07	C
2	27-Dec-07	A, B, F
3	25-Dec-07	B
3	28-Dec-07	C, D, E
4	1-Dec-07	A, B
4	25-Dec-07	F
5	22-Dec-07	B, C, H

Table 2: Location-sequeence format of the database

Location id	Location sequence
1	<(A) (G)>
2	<(A B) (C) (A B F)>
3	<(B) (C D E)>
4	<(A B) (F)>
5	<(B) (C) (H)>

The algorithm: The algorithm can be decomposed into five phases.

Sort phase: The original transaction database is sorted. Using the agent-id as the major key and transaction time as the minor key; the result is set of agent sequences. Table 1 is an example of sorted database.

L-eventsets phase (or leventset phase): The sorted database is scanned to obtain large 1-event-set. A large event-set is referred to a sequence satisfying the minimum support as per predefined support threshold. Base on the example of Table 1, assume the minimum support is 25%, the satisfied 1-event-sets are (A), (B), (C), (F) and (A B). For convenience purpose then researchers map these to alternative representation as per Table 3.

Transformation phase: Base on the result of L-event-set phase, the original transaction database transform into location sequences format. In the transformed location sequences, each transaction is replaced by the set of all leventset contained in that transaction. If a transaction does not contain any leventset, it is not retained in the also transform each transformed location sequence into an alternative representation. Table 4 is an example of transformed database.

Sequence phase: All frequent sequential patterns are generated from the transformed sequential database. As per GSP, AprioriAll was proposed to find the frequent sequential patterns.

Maximal phase: Those sequential patterns that are contained in other super sequential patterns are pruned in this phase, since researchers are only interested in maximum sequential patterns. The AprioriAll is based on the Apriori algorithm in association rule mining, similarly there are two subprocess. The first is to generate those sequences that may be frequent which is also called

Table 3: Litemset table

Large event-sets	Mapped to
{(A)}	1
{(B)}	2
{(C)}	3
{(F)}	4
{(A B)}	5

Table 4: Transformed database

Location id	Original location sequence	Transformed location sequence	After mapping
1	<(A) (G)>	<{(A)}>	<{1}>
2	<(A B) (C) (A B F)>	<{(A), (B), (A B)} {(C)}{(A), (B), (A B), (F)}>	<{1, 2, 5}, {3}, {1, 2, 4, 5}>
3	<(B) (C D E)>	<{(B)} {(C)}	<{2}, {3}>
4	<(A B) (F)>	<{(A), (B), (A B)} {(F)}>	<{1, 2, 5}, {4}>
5	<(B) (C) (H)	<{(B)} {(C)}>	<{2}, {3}>

candidate sequences. Then, the sequential database is scanned to check the support of each candidate to determine frequent sequential patterns according to minimal support.

THE APPLICATION OF STEM FOR CRIME ANALYSIS

Researchers often use historical data to analysis crime trend. Han and Kamber (2000) has summarized four classes of time series pattern:

- Trend analysis: This is to find the evolution patterns of attributes over time
- Similarity search: It tries to find sequences that differ only slightly
- Sequential patterns: Trying to find the relationship between occurrences of sequential events
- Periodical patterns: These are those recurring patterns in the time series database

Regression is one most typical tool for crime analysis. Such as Brown and Oxford (2001), it based on Routine Activity Theory (RAT) to predict crime trend in sub-city regions of Richmond, Virginia, US RAT is first described by Marcus and Cohen (1980). The theory states that in order for a crime to occur, a motivated offender, a suitable target and the lack of a capable guardian must converge in both space and time. Researchers could use STEM to analysis crime sequence.

RECOMMENDATIONS

In GSP, a frequent sequential pattern is a sequence whose statistical significance in the database is above user-specified threshold. As STEM is derived from GSP, researchers have to compare both results. Currently, the AprioriAll algorithm is applied for finding frequent sequence in GSP. Researchers are also interested in studying whether AprioriAll is most suitable for spatio temporal data. The following question is also considerable. Given the transaction database with three attributes agent-id, transaction-time and items, the mining process is sorted with agent-id as the major key and transaction-time as the minor key, the result is set of agent-sequences. However, the method does not mention how to handle the recurring patterns within the long sequence. For example, GSP emphasis only maximum sequential patterns are interested. Let (A), (B) and (C) are events, assume the finding is (ABABAB). It is interesting to discuss either (AB) or (ABABAB) is more useful.

REFERENCES

Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. Proceedings of the 11th International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan, pp: 3-14.

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 25-28, 1993 Washington, DC., USA., pp: 207-216.

Brown, D.E. and R.B. Oxford, 2001. Data mining time series with applications to crime analysis. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 3, October 7-10, 2001, Tucson, AZ, pp: 1453-1458.

Cao, H., N. Mamoulis and D.W. Cheung, 2005. Mining frequent spatio-temporal sequential patterns. Proceedings of the 5th IEEE International Conference on Data Mining, Houston, Texas, November 27-30, 2005, IEEE Computer Society, USA., pp: 82-89.

Chen, M.S., J. Han and P.S. Yu, 1996. Data mining: An overview from a database perspective. IEEE Trans. Knowledge Data Eng., 8: 866-883.

Han, J. and M. Kamber, 2000. Data Mining Concepts and Techniques. Morgan Kaufmann, San Francisco, USA.

Han, J., J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal and M. Hsu, 2000. FreeSpan: Frequent pattern-projected sequential pattern mining. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 20-23, 2000, Boston, MA., USA., pp: 355-359.

Hsu, W., W. Hsu, M.L. Lee and J. Wang, 2005. Mining generalized spatio-temporal pattern. Proceedings of the 10th International Conference on Database Systems for Advanced Applications, April 17-20, 2005, Beijing, China, pp: 649-661.

Marcus, F. and L.E. Cohen, 1980. Human ecology and crime: A routine activity approach. Hum. Ecol., 8: 389-406.

Masseglia, F., M. Teisseire and P. Poncelet, 2005. Sequential pattern mining: A survey on issues and approaches. Encyclopedia of Data Warehousing and Mining, Citeseer, pp: 1028-1032.

Ng, V., S. Chan, D. Lau and C.M. Ying, 2007. Incremental mining for temporal association rules for crime pattern discoveries. Proceedings of the 18th Conference on Australasian Database, Volume 63, January 30-February 2, 2007, Australian Computer Society Inc., Darlinghurst, Australia, pp: 123-132.

Pei, J., J. Han, M.A. Behzad, P. Helen, Q. Chen and M.C. Hsu, 2001. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, Germany, pp: 215-224.

Ramirez, J.C.G., D.J. Cook, L.L. Peterson and D.M. Peterson, 2000. An event set approach to sequence discovery in medical data. J. Intell. Data Anal., 4: 513-530.

Ramirez, J.C.G., L.L. Peterson and D.M. Peterson, 1998. A sequence building approach to pattern discovery in medical data. Proceedings of the 11th International Florida Artificial Intelligence Research Society Conference, May 18-20, 1998, AAAI Press, Menlo Park, CA., USA., pp: 188-192.

Srikant, R. and R. Agrawal, 1996. Mining sequential patterns: Generalizations and performance improvements. Proc. Int. Conf. Extend. Database Technol., 1057: 3-17.

Wang, J., W. Hsu and M.L. Lee, 2005. A framework for mining topological patterns in spatio-temporal databases. Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, October 31-November 05, 2005, ACM, New York, USA., pp: 429-436.

Wang, J., W. Hsu, M.L. Lee and J. Wang, 2004. FlowMiner: Finding flow patterns in spatio-temporal databases. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, November 15-17, 2004, Boca Raton, Florida, pp: 14-21.

Wang, W., J. Yang and R. Muntz, 2001. TAR: Temporal association rules on evolving numerical attributes. Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, pp: 283-292.

Zaki, M.J., 2001. SPADE: An efficient algorithm for mining frequent sequences. Mach. Learn. J., 40: 31-60.

Zhao, Q. and S.S. Bhowmick, 2003. Sequential pattern mining: A survey. ITechnical Report CAIS Nayang Technological University Singapore, pp: 1-26.