A Review of Frequent sequential Pattern Mining Methods

[1]Deepti chandnani, [2]Prof. Ravi Gedam

[1]M.Tech Scholar, [2]Assistant Professor
SBITM, Betul (M.P)

***Abstract*: With the mining capabilities of the several data mining methodologies, there are several interesting extensions on frequent pattern mining. The discovery of sequential patterns is one of them. It has a vast array of real world applications. It is worthy of study on extending the memory indexing approach for efficient mining of generalized sequential patterns. This paper proposes a critical review of the sequential pattern mining methods.**

***Index Terms*- Data Mining, KDD Process, Sequential Pattern Mining,**

1. INTRODUCTION:

Data mining is generally thought of as the process of finding hidden, non trivial and previously unknown information in large collection of data. Association rule mining is an essential component of data mining. Basic objective of finding association rules is to find all co-occurrence relationship called associations. Most of the research efforts in the scope of association rules have been oriented to simplify the rule set and to improve performance of algorithm. But these are not the only problems that can be found and when rules are generated

A typical process of knowledge discovery in databases is illustrated in Fig. 1-1.

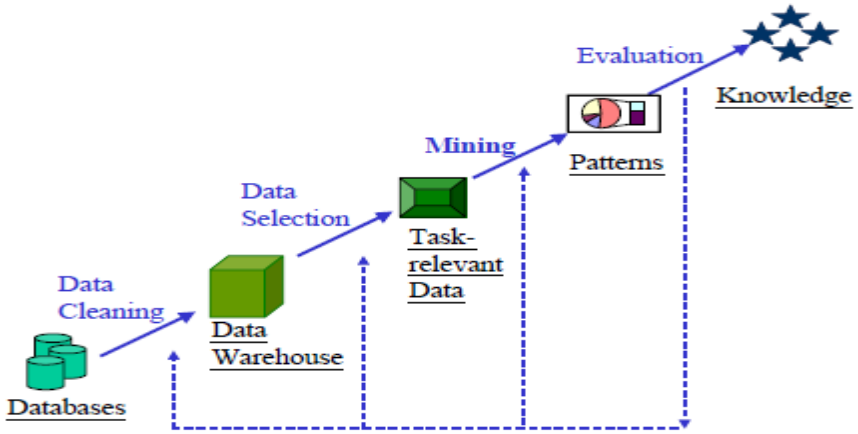

Fig 1. The process of knowledge discovery in databases

2. BACKGROUND:

Sequence Database Each sequence is a time-ordered list of item sets. An item set is an unordered set of items (symbols), considered to occur simultaneously.

S.No	ID	Sequences
01	Seq1	{a,b},{c},{f},{g},{e}
02	Seq2	{a,d},{c},{b},{a,b,e,f}
03	Seq3	{a},{b},{f},{e}
04	Seq4	{b},{f,g}

Table 1: Data Mining Sequences

Sequential Pattern Mining (SPM) [2,3,7] is perhaps the foremost standard set of techniques for locating temporal patterns in sequence databases. SPM finds sub- sequences that ar common to over minsup sequences. SPM is restricted for creating predictions. for instance, take into account the pattern. It"s attainable that y seems often when an x but that there are also several cases wherever x isn't followed by y. For prediction, we'd like a mensuration of the confidence that if x happens, y can occur afterward.

A sequential rule usually has the shape X->Y . A sequential rule X⇒Y has 2 properties:

1. Support: the number of sequences where X happens before Y, divided by the number of sequences.
2. Confidence the number of sequences where X happens before Y, divided by the number of sequences where X occurs.

3. LITERATURE SURVEY

Yan [12] uses weight constraints to reduce the number of unimportant patterns. During the mining process, they consider not only supports but also weights of patterns. Based on the framework, they present a weighted sequential pattern mining algorithm (WSpan).

Chen, Cao, Li, & Qian [4] incorporate user-defined tough aggregate constraints so that the discovered knowledge better meets user needs (19). They propose a novel algorithm called PTAC (sequential frequent Patterns mining with Tough Aggregate Constraints) to reduce the cost of using tough aggregate constraints by incorporating two effective strategies. One avoids checking data items one by one by utilizing the features of "promising-ness" exhibited by some other items and validity of the corresponding prefix. The other avoids constructing an unnecessary projected database by effectively pruning those unpromising new patterns that may, otherwise, serve as new prefixes.

(Masseglia, Poncelet, & Teisseire, 2003) [5] propose an approach called GTC (Graph for Time Constraints) for mining time constraint based patterns (as defined in GSP algorithm) in very large databases(20). It is based on the idea that handling time constraints in the earlier stage of the data mining process can be highly beneficial. One of the most significant new features of their approach is that handling of time constraints can be easily taken into account in traditional level- wise approaches since it is carried out prior to and separately from the counting step of a data sequence.

(Wang, Chirn, Marr, Shapiro, Shasha, & Zhang, 1994) [6] looked at the problem of discovering approximate structural patterns from a genetic sequences database (21). Besides the minimum support threshold, their solution allows the users to specify:

1. The desired form of patterns as sequences of consecutive symbols separated by variable length don't cares,
2. a lower bound on the length of the discovered patterns, and
3. an upper bound on the edit distance allowed between a mined pattern and the data sequence that contains it.

Their algorithm uses a random sample of the input sequences to build a main memory data structure, termed generalized suffix tree, that is used to obtain an initial set of candidate pattern segments and screen out candidates that are unlikely to be frequent based on their occurrence counts in the sample. The entire database is then scanned and filtered to verify that the remaining candidates are indeed frequent answers to the user query.

GSP (Generalized Sequential Pattern) was introduced by Srikant and Agrawal (1996) [1,8] it is also an Apriori-based pattern mining algorithm. The whole algorithm has two subprocesses: candidate pattern generation and frequent pattern generation.

In the candidate generation process, candidate k-sequences are generated based on the large (k-1) – sequences using the same method described by Agrawal and Srikant (1994).

The candidate sequences are generated in two steps: joining phase and pruning phase. In the joining phase, candidate k-sequences are generated by joining two (k-1) sequences that have the same contiguous subsequences. When joining the two sequences the item can be inserted as a part of the element or as a separate element. For example, <(a,b)(c)> and <(a,b)(d)> have the same contiguous subsequence <(a,b)>, based on those candidate 4-sequence <(a,b)(c,d)>,<(a,b),(c)(d)> and <(a,b)(d)(c)> can be generated. While in the pruning phase, those candidate sequences that have a contiguous subsequence whose support count is less than the minimal support are deleted. It also uses the hash-tree structure to reduce the number of candidates to be checked in the next phase.

PrefixSpan (Pei et al., 2001) [11] is a more efficient algorithm for mining sequential patterns compared with Aprioriall. PrefixSpan is also capable of dealing with very large databases. PrefixSpan mainly employs the method of database projection to make the database for next pass much smaller and consequently increasing the speed of the algorithm. Also in PrefixSpan there is no need for candidate generation, this step is instead by recursively generating projected database according to the sequence prefix. PrefixSpan mainly avoids generating and counting candidate sequences, which is the most time-consuming part of Apriori-like sequential mining methods.

By using projection, the database that PrefixSpan scans each subsequent time is much smaller than the original database. The main cost of PrefixSpan is the projected database generation process, and in order to improve the performance a bi-level projection method that uses the triangle S-Matrix is introduced.

SPAM (Sequential PAttern Mining) is a typical algorithm which integrates a variety of old and new algorithmic contributions. It is introduced by [9,10] a lexicographic tree has been used to store all the sequences. SPAM traverses the sequence tree in a standard depth-first search (DFS) manner. At each node n, the support of each sequence-extended child is tested. If the support of a generated sequence s is greater than or equal to minimum support, SPAM stores that sequence and repeats the DFS recursively on s. (Note that the maximum length of any sequence is limited since the input database is finite.) If the support of s is less than minimum support, then SPAM does not need to repeat the DFS on s by the Apriori principle[13,14] , since any child sequence generated from s will not be frequent. If none of the generated children are frequent, then the node is a leaf and user can backtrack up the tree.

4. CONCLUSION:

Sequential pattern mining is a popular area of research. There are many real world applications are related to it. A survey of frequent sequential pattern mining methods have n=been proposed. This paper also elaborated the concept of sequential pattern mining and knowledge discovery in data base in a lucrative manner.

References:

1. Srikant R, Agrawal R., 1995. "Mining generalized association rules". In: Dayal U, Gray P M D, Nishio Seds. Proceedings of the International Conference on Very Large Databases. San Francisco, CA: Morgan Kanfman Press, pp. 406-419.
2. M. Houtsma, and Arun Swami, 1995. "Set-Oriented Mining for Association Rules in Relational Databases". IEEE International Conference on Data Engineering, pp. 25–33.
3. S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, 1997. "Dynamic itemset counting and implication rules for market basket data". In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, volume 26(2) of SIGMOD Record, pp. 255–264. ACM Press.
4. Chen, E., Cao, H., Li, Q., & Qian, T. (2008). Efficient strategies for tough aggregate constraint-based sequential pattern mining. Inf. Sci., 178(6), 1498-1518.

5. Masseglia, F., Poncelet, P., & Teisseire, M. (2003). Incremental mining of sequential patterns in large databases. Data Knowl. Eng., 46(1), 97–121.

6. Wang, J. L., Chirn, G., Marr, T., Shapiro, B., Shasha, D., & Zhang, K. (1994). Combinatorial pattern discovery for scientific data: Some preliminary results. Proc. ACM SIGMOD Int'l Conf. Management of Data, (pp. 115-125).
7. Yang, J., Wang, W., & Yu, P. S. (2001). Infominer: mining surprising periodic patterns. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
8. Srikant, R., & Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology.
9. Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences.
Journal of Machine Learning, 42(1-2), 31-60.

10. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. (2000). FreeSpan: frequent pattern-projected sequential pattern mining. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
11. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., et al. (2001). PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. Proceedings of the 17th International Conference on Data Engineering.
12. Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining Closed Sequential Patterns in Large Datasets. Proceedings of the 2003 SIAM International Conference on Data Mining (SDM'03).
13. Tzvetkov, P., Yan, X., & Han, J. (2005). TSP: Mining top-k closed sequential patterns. Knowledge and Information Systems, 7(4), 438-457.
14. Vincent Tseng, Fournier. (2012). Mining Top K sequential rules. ADMA 2012 proceedings, springer publications.