A Review of Frequent Sequential Pattern Mining Over Large Data Set

[1]Rajesh Gupta, [2]Prof. Rakesh Pandit

[1]Research Scholar, [2]Assistant Professor

***Abstract*: As with the new innovation of the IT technologies, the amount of accumulated data is also increasing. It has resulted in large amount of data stored in databases, warehouses and other repositories. Thus the Data mining comes into representation to explore and analyze the databases to extract the interesting and previously unknown patterns and rules known as association rule mining. In data mining, Sequential Pattern mining becomes one of the important tasks of descriptive technique which can be defined as discovering meaningful patterns from large collection of database. This paper presents a review of sequential pattern mining techniques.**

1. Introduction:

As with the new innovation of the IT technologies, the amount of accumulated data is also increasing. It has resulted in large amount of data stored in databases, warehouses and other repositories. Thus the Data mining comes into representation to explore and analyze the databases to extract the interesting and previously unknown patterns and rules known as association rule mining.
In data mining, Association rule mining becomes one of the important tasks of descriptive technique which can be defined as discovering meaningful patterns from large collection of database. Mining frequent itemset is very crucial part of association rule mining.
Many algorithms have been proposed from last many decades including horizontal layout based techniques, vertical layout based techniques, and projected layout based techniques. But most of the techniques suffer from repeated database scan, Candidate generation (Apriori Algorithms), memory consumption problem (FP-tree Algorithms) and many more for mining frequent patterns. As many industries transactional databases contain same set of transactions many times, to apply this thought, this thesis presents a new technique which uses time & space efficient algorithm that guarantee the better performance than classical apriori algorithm.

It is the combination of Multiple Disciplines. Figure 1 shows the different disciplines that take part in data mining.

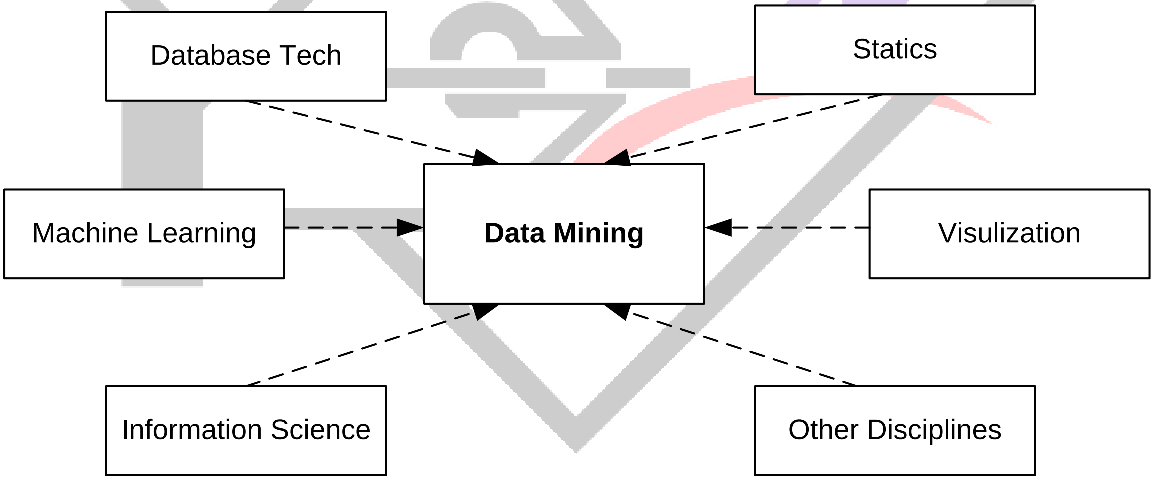

Figure 1 Show the Multiple Disciplines for data mining

Data mining is not new. People who first discovered how to start fire and that the earth is round also discovered knowledge which is the main idea of Data mining. Even before technologies were used for Data mining, statisticians were using probability and regressing techniques to model historical data [1]. Today technology allows to capture and store vast quantities of data. Finding and summarizing the patterns, trends, and anomalies in these data sets is one of the big challenges in today's information age. "With the unprecedented growth-rate at which data is being collected [2] and stored electronically today in almost all fields of human endeavor, the efficient extraction of useful information from the data available is becoming an increasing scientific challenge and a massive economic need" [3].
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.

2. Literature Survey:

Generally, frequent-pattern mining results in a huge number of patterns of which most can be found to be insignificant according to application and/or user requirements. As a result, there have been efforts in the literature to mine constraint-based and/or user-interest based frequent patterns [4], [5], [6], [7]. In recent times, temporal periodicity of frequent patterns has been used as an interestingness criterion to discover a class of user-interest based frequent patterns, called periodic-frequent patterns [8]. A pattern is said to be periodic-frequent if it satisfies both the minimum support (minsup) and the minimum confidence (minconf) constraints. Minsup constraint controls the minimum number of transactions that a pattern must cover in a database. Minconf constraint controls the minimum number of items that a pattern must cover in all the transactions.

Since a single minsup and a single minconf constraint are used for all items in the database, this model implicitly assumes that all items have similar frequencies and occurrence behavior. However, this is not the case in real-world datasets. Real-world datasets are non-uniform in nature containing both frequent items

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.

3. Conclusion:

The data mining is helpful for analysis the data, when the manually analysis of the data is not feasible then the data mining techniques are applied for analysis. The data mining techniques are the computer based algorithms which identify the relationship among the data and extraction of the similar pattern data on which they are trained. This paper has given a literature review for mining all the sequentially frequent patterns from a sequential data set.

References:

[1] Groth Robert. "Data Mining: A Hands-on Approach for Business Professionals". Prentice Hall PTR, 1998.
[2] Witten Ian and Eibe. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". San Francisco: Morgan Kaufmann Publishers, 2000.
[3] Zaki Mohammed and Ho Ching-Tien, "Large-Scale Parallel Data Mining". Berlin: Springer, 2000.
[4] Hu T., Sung S. Y., Xiong H., and Fu Q., "Discovery of maximum length frequent itemsets". Inf. Sci., 178:69–87, January 2008.
[5] Quang T. M. , Oyanagi S. , and Yamazaki K., "Mining the k-most interesting frequent patterns sequentially". In E. Corchado, H. Yin, V. J. Botti, and C. Fyfe, editors, IDEAL, volume 4224 of Lecture Notes in Computer Science, pages 620–628. Springer, 2006.
[6] Kumar, R. ; Dixit, M., Analysis on probabilistic and binary datasets through frequent itemset mining, Page(s): 263 - 267 Conference Publications, IEEE 2012.
[7] Schmidt Jana and Kramer Stefan , “The Augmented Itemset Tree: A Data Structure for Online Maximum Frequent Pattern Mining”, pp 277-291 Springer 2011.
[8] Tanbeer S. K., Ahmed C. F., Jeong B.-S., and Lee Y.-K., "Discovering periodic-frequent patterns in transactional databases". In PAKDD ’09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pages 242–253, Berlin, Heidelberg, Springer-Verlag 2009.
[9] Mannila, H., Toivonen and H., Verkano, A.I. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1, 1 (1997), 259-289

[10] Dr P padmaja, P Naga Jyoti, m Bhargava “*Recursive Prefix Suffix Pattern Detection Approach for Mining Sequential Patterns”* IJCA September 2011

[11] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of International Conference on Very Large Data Bases*, pages 487– 499, 1994.

[12] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of Inter- national Conference on Data Engineering*, pages 3–14, 1995.

[13] R. Agrawal and E. Wimmers. A framework for expressing and combining pref- erences. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 297–306, 2000.

[14] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using a bitmap representation. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 429–435, 2002