Analysis of Sequential Mining Algorithms

Surbhi Chandhok
Bachelors of technology
Computer Science
Galgotias College of Engineering & Technology
Uttar Pradesh, India

Romil Anand
Bachelors of Technology
Computer Science
Galgotias College of Engineering & Technology
Uttar Pradesh, India

Soumay Gupta
Bachelors of Technology
Computer Science
Galgotias college of Engineering &Technology
Uttar Pradesh, India

Aatif Jamshed
Masters of Technology
Computer Science
Galgotias college of Engineering &Technology
Uttar Pradesh, India

ABSTRACT
This paper essentially analyses the sequential pattern of mining algorithms. The discovery of Association relationship seeks more attention in data mining due to the constantly increasing amount of data stored in the real application system. Mining for association rules has its usage in several areas of business such as the process of decision making and the development of customized marketing programs & strategies. Therefore, the primary objective of data mining is to transform "data into knowledge". As a result of which, mining association rules from enormous databases has been a significant topic in recent research for knowledge discovery in databases.

It is known that database can be both dynamic and static. Static databases are the ones that do not change or alter with the passage of time. On the other hand, in dynamic databases, various new transactions append as time passes by. This might result in the production of some new itemsets while it is possible that certain frequent itemsets might as well become invalid. Therefore, in dynamic databases, the maintenance of large itemsets can be extremely expensive, in case rerun of previous mining algorithms on updated database is applied as it repeats a major portion of work done during previous computations. Apart from this, there is also lack of space for the storage of all the data and its processing. Therefore, it is recommended that instead of finding enormous itemsets again, certain heuristics be used for mining of dynamic databases. It brings forth the study of sequential pattern-mining algorithms, classified into five varied classes.

1. on the basis of Apriori-based algorithm.
2. on the basis of FP-Growth Algorithm.
3. on the bassis of Fast Algorithm.
4. on Partition Based Algorithm.
5. on the basis of Fast Update algorithm.

Keywords
Sequential Pattern, Data Mining, Pattern analysis,

1. INTRODUCTION
It can be said that the topic of Sequential Pattern mining is primarily concerned with looking for statistically approved patterns between different data examples where the valuesare delivered in a sequence. Usually, it is presumed that the values are distinct and therefore, we can say that time series mining is also closely related and similar, but it's usually considered a separate activity. Sequential pattern mining is also known as a special case of structured data mining.

2. CATEGORIS OF PATTERN MINING ALGORITHM SEQUENTIALLY
Algorithms for sequential pattern mining can be classified into the following classes:-

1. Apriori-likealgorithms
2. BFS Breadth First Search)-based algorithms
3. DFS (Depth First Search)-basedalgorithms
4. closed sequential pattern based algorithms
5. incremental-based algorithms

2.1 Static Datamining
Static Data Mining can be defined as one that uses static database for mining. There can be a wide range of static data mining algorithms such as Fp- Tree, Partition based algorithm, Apriori, Fast algorithmetc.

2.1.1. Apriori Algorithm
The most widely accepted static data mining algorithm-Apriori is often described as a "fast algorithm used for mining association rules". It is enforced by marketbasket data. Also, it effectively produces large itemsets along with candidate itemsets through the process of repeatedly scanning the database.

Apriori algorithm is the one based on candidate set generation accompanied with the test method. The unresolved issue that often appears during the process of mining frequent relations is, repeated scanning of original database, large number of candidate generation along with workload of support counting of the candidates. Hence, there is a need to begin reduction of passes of transaction database scans, in order to lessen the number of candidates and to help with support counting of candidates. Apriori algorithm isn'tefficient enough, regardless of it being the building ground for several efficient algorithms.

Suppose there is a transaction database which includes customer sequences. This database is composed majorly by three attributes:

1. customer-id
2. transaction time
3. purchased-item

Decomposed with five steps, the mining process takes place as follows:

Sort step: In this step, the transaction database is sorted according the customer- id.

L-itemset step: The main goal of this step is to get the large itemsets from the sorted database, in accordance with the support threshold.

Transformation step: In this step, the sequences are replaced by the huge itemset that they contain. In order to make the process of mining efficient, all of the large itemsets are mapped in the form of an integer series after which, the original database is transformed into set of customer sequences, of which the large itemsets are representative

Sequence step: Now, from all the sequential database that is transformed, this step produces frequent sequential patterns and forms.

Maximal step: This final step extracts the sequential patterns contained in several other super sequential patterns, since our prime concern is to obtain maximum sequential patterns.

2.1.2. FP-Growth Algorithm

Known to be an order of magnitude faster than the Apriori algorithm, FP-Tree is primarily used for mining static databases. This algorithm involves the generation process of frequent patterns and basically includes two processes:

1. Constructing the FP-tree.
2. Generating frequent patterns from the FP tree.

This involves divide-and-conquer method and 2 scans of database are taken. Candidate itemsets generation does not takes place during these steps.

Mainly, 2 approaches are followed to generate FP- Growth algorithm:

- Generate a compact data structure FP tree

Take some items directly from FP tree FP-Tree structure construction

The frequent-pattern tree, commonly known as the

FP-tree is basically a compact structure that helps in the storage process of quantitative information regarding frequent patterns formed in a database. One of the roots is labeled as "null" which comes with a set of itemprefix subtrees such as children along with a frequentitem- header table.

- Each node located in the item-prefix subtree comprises of mainly three fields:
    - Item-name- It registers whichever item is being represented by the node.
    - Count- Its function is to count the number of transactions represented by the path that reaches the node.
    - Node link- It connects a node to the next node in the FP-tree which carries the same item name, or null in case there is none.
- Every entry in the frequent-item-header table comprises of mainly two fields: ○Item-name ○

Head of node-link: It can be described as a pointer to the very first node in the FP-tree which is carrying the item-name.

2.1.3. Fast Algorithm

Keeping a track of the frequency of the occurrences of the intriguing subsets of items termed candidates often the most time consuming task while discovering the association rules from the database. Therefore, this calls for a strong need to develop a method that would avoid or lessen the production of candidates and would test and utilize some novel data structures to decrease the monetary cost in frequent pattern mining. Fast algorithm makes use of TreeMap- a structure in java that store key/ value pair. Furthermore, the Arraylist technique that majorly decreases the requirement of traversing through the database is also used. This also avoids excessive usage of storage memory.

This algorithm brings forward an improvement of the candidate generation and also supports the counting of GSP algorithm. The generating and pruning method is used by this algorithm to produce and validate candidates, according to the previous mining result. As per the Performance study, it is shown that the performance of this algorithm is better than previous algorithms used for the purpose of maintenance of sequential patterns in terms of speed. FASTUP includes the same limitations as GSP, nevertheless.

2.2 Dynamic Data Mining

Data Mining that primarily makes use of dynamic databases and considers all updates (insert, delete and update problems) into account is defined as dynamic data mining. There exists varied kinds of dynamic data mining algorithms such as Fast Update (FUp), incremental method that includes promising based algorithm along with probability based algorithm.

2.2.1 Partition Based Algorithm

Partition based algorithm's prime function is to divide the database into segments that lessens the number of database scans to two. This algorithm is used to reduce CPU as well as I/O overheads and is especially suitable for extremely large size databases. During the first scan, it divides the database into multiple partitions and generates frequent itemsets in various partitions separately by scanning the database, once in each partition. Then, during the second scan, counters for each of these itemsets are organized and their actual strength is measured so as to decide if they are large enough for entire database. A major fraction of itemsets would turn out to be large if the items are evenly distributed across the partitions.

2.2.2 Fast Update Algorithm

When new transaction data is added to a transaction base, effective maintenance of discovered association rules, for which an incremental updating technique is used, called as FUp (Fast Update) algorithm. In this technique, we segregate winners (ones that remain large in updated database) from losers (ones that aren't large in updated database) among all other large items in the original database and also find new winners that are large in original database (DB) and incremental database (db) i.e. (DB U db). This algorithm is known to be 2 to 16 times faster than the Apriori.

3. NEW STATIC DATA MINING ALGORITHM (PAPRIORI)

Based upon basic data mining algorithm(Apriori), PAPRIORI algorithm is used to generate frequent itemsets uniformly in static database through the process of reading K transactions at a time. For the initial K transactions, m large itemsets will be produced and then, for next K transactions m, m+1 large itemsets will be produced consecutively and so on. This algorithm is based on the followingconsiderations

- Those itemsets that do not satisfy minimum support or are counted initially are termed as 'Estimated Infrequent' (EI) itemsets.
- Estimated Frequent (EF) itemsets are the ones satisfying the minimum support threshold.
- Itemsets that have been counted throughout the whole database only once and they satisfy minimum support are called as Confirmed Frequent(CF).
- The itemsets that are counted throughout whole database once and do not satisfy minimum support are termed as Confirmed Infrequent (CI).
- Following are the steps followed in thisalgorithm:

  Step 1: Initially, set all itemsets as Estimated Infrequent (EI) itemsets.

  Step2: Now, go on to read the database with K transactions at a time (until transaction reads is less than the total number of transactions in database).

- Keep increasing the counter for the itemset for every transaction.
- For every itemset that belongs to Estimated Infrequent, if value of counter satisfies minimum support, then label the itemset as Estimated Frequent.
- The immediate superset of itemsets is set as EI if they belong to either EF or CF.
- For every itemset that belongs to Estimated Frequent, if it is read throughout the whole database once, it is moved to the CF.
- On the other hand, if any itemsets belong to the Estimated Infrequent and is read throughout the whole database once, it is moved to the CI.

4. ACKNOWLEDGMENT

This research would not have been completed without the guidance of Mr. Manish Kumar Sharma (Project Coordinator) and HOD CSE Dr. Bhawna Mallick. We would like to thank all the professors who guide us in dis project.

5. CONCLUSION

In the future, extensive research and experiments on the proposed algorithm will be brought forward. We have also proposed static data mining algorithm that helps generate itemsets progressively with minimal execution time at intermediate no. of transactions read. We are sure in the upcoming years, detailed research and well planned experiments could very well bring forth even more interesting details on this topic.

6. REFERENCES

[1] Mabroukeh, N. R.; Ezeife, C. I. (2010). "A taxonomy of sequential pattern mining algorithms"

[2] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216

[3] Rakesh Agrawal & Ramakrishan Srikant, “Fast algorithm for mining Association rules”, IBM Almaden Research Center, 650 Harry road, San Jose, CA 95120: In proceedings of the 20th VLDB conference Santiago, Chile, pp 487-499

[4] J. Han, J. Pei, and Y. Yin.” Mining frequent patterns without candidate generation”, in W.Chen, J. Naughton, and P. A.Bernstein, editors, 2000 ACM SIGMOD Intl. Conference on Management of Data, Vol. 29, No.2 pp 1-12.

[5] M.H.Margahny and A.A.Mitwaly,” Fast Algorithm for Mining Association Rules”, AIML 05 Conference, pp 19-21, December 2005, CICC, Cairo, Egypt.

[6] Ashok Savasere, Edward Omiecinski, Shamkant Navathe, “An Efficient Algorithm for Mining Association Rules in Large Databases”, in proceedings of 21st VLDB Conference , Zurich , Switzerland, pp432-444, 1995.

[7] David W. Cheung, Jiawei Han, Vincent T. Ngt C.Y. Wongj,” Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique”, in proceedings of the 12th ICDE, New Orleans, Louisiania (IEEE) , pp 106-114,February 1996.

[8] Lin MY, Lee SY (1998) Incremental update on sequential patterns in large databases. In Proc of the 10th IEEE IntConf on Tools with Artificial Intelligence.Nov. 1998, Taipei, Taiwan.24–31.

[9] Tian Lan, Runtong Zhang and Hong Dai, “A New Frame of Knowledge Discovery,” in Proc. 1st International Workshop on Knowledge Discovery and Data Mining, WKDD 2008, Jan. 2008, pp 607– 611.

[10] Hebah H. O. Nasereddin, “Stream Data Mining”, International Journal of Web Applications, Volume 1, No. 4, December 2009, pp183-190.