Improved Apriori Algorithm using Incremental Technique

Sudha Devi Kore
M.Tech. Scholar, School of Future Studies and Planning, DAVV, INDORE

Avinash Navlani
Lecturer, School of Future Studies and Planning, DAVV, INDORE

Abstract - Finding frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems.

This paper proposed a new algorithm for mining frequent sequences that uses information collected during an earlier mining process to minimize the time and cost for finding new sequential patterns in the updated database. The algorithm is designed to work faster than the other approaches of such frequent pattern mining tasks. This could be useful for mining sequential patterns by applying this proposed algorithm than to mine sequential patterns using a standard algorithm, by dividing the database into an original database and an increment.

Keywords: Incremental Mining, Sequential Pattern Mining, Minimum Support, Split database, interestingness of frequent itemsets.

I. INTRODUCTION

The studies propose incremental frequent pattern mining method, which can discover sequential frequent pattern itemsets. It can efficiently identify all frequent itemsets that occur in incremental database in particular periods when the new transaction data are added into the original transaction database.

Consider an original and an incremental customer transaction database. Incremental database may contain new transactions for new customers. To calculate the itemset of sequential patterns in the updated database, we want to avoid counting everything from the scratch. Some main things one has to consider are as follows [2]:

- Find out all sequential patterns not frequent in the original database but become frequent with the increment.
- Observe all transactions in the original database which can be extended to become frequent.
- Old frequent sequences may become invalid when adding new entries.

II. ASSOCIATION RULE MINING

This method is proposed by Rakesh Agrawal in 1993 [1,6,7,8,9]. Initially used for market basket analysis to find how frequent itemsets purchased by customers. It generates a set of rules to help understanding the relationships that might exist in data. It is used to find interesting associations and correlation relationship among large amount of itemsets. Association rules show attributes value conditions that occur frequently together in a given dataset. It works on "If- then" relationship. Customer buying habits by finding associations and correlations between the different items that customers place in their "shopping basket" is known as market basket analysis.

Frequent patterns are pattern that appear frequently in a dataset example a set of items such as milk and bread that appear frequently in the transaction. Items that occur often together can be associated to each other together occurring items from a frequent itemsets. Frequent itemset mining leads to the discovery of associations and correlations among items in large data sets. The process analyzes customers between the different items that customers place in their shopping baskets.

III. BACKGROUND & RELATED WORK

Most research of data mining has focused on the problem of mining association rules. Some research studies the time constraint data mining and sequence mining. In such a case, the data being mined has a timestamp, the data will increase with the time. If we re-run the algorithm of data mining to analyze the whole database including

incremental data and original data, it is obviously inefficient and time consuming [2,3,4,5,6]. There are some popular algorithms used in data mining.

1. *Apriori Algorithm:* The name of algorithm is based on fact that algorithm uses level-wise search. Apriori uses generate & test approach. Generation of candidate itemsets is expensive in both space & time. Support counting is expensive while performing subset checking and multiple database scan.
2. *FP Growth:* Allow frequent itemsets, generating without candidate itemsets generation. It perform two step approach:
    a. Build a compact data structure called from FP tree.
    b. Extract frequent itemset directly from FP tree.
3. *FUP (Fast Update):* Cheung and Han et al. [10] first proposed an algorithm, called FUP (Fast Update), for the incremental mining association rules. Subsequently, other researchers have proposed many algorithms [11,14,15,16,17] to solve the incremental updating association patterns. When new transactions are added to the database, the FUP algorithm updates the association rules in a database. Algorithm FUP is used the concept of Apriori and is designed to discover the new frequent itemsets iteratively.
4. *ISM (Incremental Sequence Mining):* In [13]**,** an algorithm called ISM (Incremental Sequence Mining) was proposed based on SPADE approach [18], which can update the frequent sequences when new transactions and new customers are added to the database. It builds an increment sequence Lattice that consists of all the frequent sequences and the negative border sequences [4]. When new data arrive, the incremental part is scanned once and the result of scanning the new data is merged into the Lattice. If the transaction database is very large, the size of negative border will be very large, which will consume a lot of memory [13].
5. *ISE (Incremental Sequence Extraction):* In [12], an algorithm called ISE (Incremental Sequence Extraction) was proposed for mining frequent sequence, which generates candidates in the whole database by attaching the sequence of the incremental database to the frequent sequence of the original database. Therefore, it avoids keeping the large number of negative border sequences and re-computing those sequences when the data in the original database have been updated. However, since the ISE algorithm does not keep negative border sequences, it will need searching the database more. Furthermore, the ISE algorithm only extends the suffix of frequent sequence of the original database, but not extends the prefix of frequent sequence of the original database.
6. *IUS (Incrementally Updating Sequences):* In [19], an efficient algorithm, called IUS (Incrementally updating sequences) for computing the frequent sequence when new data are added into the origin database. The IUS algorithm minimizes computing costs by reusing the negative border sequence and frequent sequence in the original database.
7. *UWEP (Updated With Early Pruning***):** The UWEP is based on partition algorithm [20] in data mining. The major advantage of UWEP is to construct a transaction list for each larger itemsets by scanning the database exactly twice.
8. *AprioriAll:* Each itemset in a large sequence must have minimum support. Any large sequence must be a list of itemsets [7,8]. Finding all sequential patterns in five phases:-
    a. Sort Phase
    b. Litemset Phase
    c. Transformation Phase
    d. Sequence Phase
    e. Maximal Phase

IV. PROBLEM STATEMENT

Let Database *D* with item transactions and db be the incremental updates of this database. The database displays only the items purchased. While the quantities of items purchased are not concerned.

The task is to find the maximal sequences among all sequences in the given a database *D* of transactions with the incremental db datasets which are added after a time period. Each such maximal sequence represents a sequential pattern, which is the output of proposed algorithm.

V. An Efficient Algorithm for finding frequent Itemset in Incremental Database

The proposed algorithm follows the approaches of UWEP and AprioriAll. It prunes an item set that will become small from the set of generated candidates as early as possible by a dynamic look ahead pruning strategy. It generates and counts the less number of candidates in the new database.

The Length of a sequence is the number of itemsets in the sequence. A sequence of length *k* is called k-sequence. A sequence concatenated from sequences *x* and *y* is denoted by *x.y*. *DB* is the original database, while *db* is the increment database. $U = DB \cup db$ is the updated database containing all sequences from *DB* and *db*. L^{DB} is the set of frequent sequences in *DB*. The task is to find frequent sequences in *U*, noted L^{U}.

VI. Main Algorithm

1) *FS (DB ,db , L_D , minsup)*
2) *C^1_{db} = all 1- sequence in db*
3) *k=1while C^K_{db} ¹ f do*
4) *find the counts of all the sequences of C^K_{db} in db*
5) *T^K_{db}= All K-sequence in C^K_{db} with support ≥ minsup in db*
6) *P_set = $L^K_{DB} – T^K_{db}$*
7) *if P_set ¹ f then*
   i) *start_prune(P_set)*
8) *end if*
9) *for all X e T^K_{db} do*
   a) *if X ε L_{DB} then*
      i) *Add X to L_{DB+db} and L^k_{db}*
   b) *else find sup_{DB} (X)*
      i) *if X is frequent in DB+db then*
      ii) *Add X to L_{DB+db} and L_{db}*
      iii) *end if*
10) *end if*
11) *end for*
12) *k =k+1*
13) *C_{db}^K=generate _candidate (L^K_{db}-1)*
14) *End while*
15) *for (k=n; k>1; k--)*
   i) *for each k-sequences S_k do*
   b) *delete all subsequences from L_{DB+db}*
16) *end for*

VII. Pruning Algorithm

7.1.1 *Start _prune(P_set)*
7.2.1 *While P_set ¹ f do*
7.3.1 *X = first element of P_set*
7.4.1 *find SUP_d (X)*
7.5.1 *If X is not frequent in DB+db then*
   a. *remove X and its supersets from L_{DB}*
   b. *remove X and its supersets from P_set*
7.6.1 *else*
   a. *append supersets of X in L_{DB} to P_set*
   b. *add X to L_{DB} + db and remove X from L_{DB}*

c. remove X from P_set
7.7.1 end if
7.8.1 end While

VIII. CONCLUSION

We developed a new method that considers sequential data mining of retail websites as an effective tool that participates greatly in having well-structured retail websites. The advantage of our method is that it saves a lot of maintenance efforts needed in the future. The proposed algorithm generates less number of candidate sets due to its look ahead pruning strategy. It traverses on the old database only where it is really required. The algorithm can be used in the fields where the database is dynamic.

We also presented a new measure that defines the interestingness of frequent itemsets. The interestingness measure is based on the idea that interesting frequent itemsets are supported by many recent transactions. This method can be used either as a preprocessing step to search for frequent itemsets within a determined interval, or as an extension to the Apriori algorithm to prune non-interesting frequent itemsets. A huge number of possible sequential patterns are hidden in databases. A mining algorithm should find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold be highly efficient, scalable, involving only a small number of database scans be able to incorporate various kinds of user-specific constraints.

The general problem of updating the transaction database includes also the deletion of transaction. Discover the new set of frequent itemsets after changing the support threshold. An improvement in the performance of the developed algorithm can be achieved by a well choice of the data structure. As a future work to our method of using data mining to support sequential frequent pattern mining process, our method can be tested on synthetic and real life datasets. Through these results, the customers influence each other in the decision of buying some product. In this way, the purchasing process can be modeled much more realistically.

REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. Swami, “Efficient similarity search in sequence databases”, Proceedings of Conference on Foundations of Data Organization and Algorithms, pp. 69-84, 1993.

[2] M. Last, Y. Klein, and A. Kandel, “Knowledge Discovery in Time Series Databases”, IEEE transactions on systems, man, and cybernetics, Vol. 31, No. 1, pp. 160-168, 2001.

[3] B. LeBaron and A. S. Weigend, “A Bootstrap Evaluation of the Effect of Data Splitting on Financial Time Series”, IEEE Transactions on Neural Networks, Vol. 9, No. 1, pp. 213-220, 1998.

[4] C. Y. Chang, M. S. Chen, and C. H. Lee, “Mining general temporal association rules for items with different exhibition periods”, IEEE International Conference on Data Mining, pp. 59-66, 2002.

[5] C. H. Lee, M. S. Chen, and C. R. Lin, “Progressive partition miner: an efficient algorithm for mining general temporal association rules”, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, pp. 1004-1017, 2003.

[6] Y. Li, P. Ning, X. S. Wang, and S. Jajodia, “Discovering calendar-based temporal association rules”, Data & Knowledge Engineering, Vol. 44, No. 2, pp. 193-218, 2003.

[7] R. Agrawal and R. Srikant, “Mining sequential patterns”, Proceedings of 1995 International Conference Data Engineering, pp. 3-14, 1995.

[8] R. Srikant and R. Agrawal, “Mining sequential patterns: generalizations and performance improvements”, Proceedings of the 5th International Conference on Extending Database Technology, pp. 3-17, Avignon, France, 1996.

[9] X. Yan, J. Han, and R. Afshar, “CloSpan: Mining Closed Sequential Patterns in Large Datasets”, Proceedings of the 2003 SIAM International Conference on Data Mining (SDM'03), pp. 166-177, San Francisco, CA, 2003.

[10] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, .Maintenance of discovered association rules in large databases: An incremental update technique,. *In Proceedings* of *12th Intl. Conf. on Data Engineering (ICDE.96)*, pages 106-114, New Orleans, Louisiana, USA, February 1996.

[11] D. W. Cheung, S. D. Lee, and B. Kao, .A General Incremental Technique for Maintaining Discovered Association Rules,. *In Proceedings of the 5th Intl. Conf. on Database Systems for Advanced Applications (DASFAA.97)*, pages 185-194, Melbourne, Australia, April 1997.

[12] F. Masseglia, P. Poncelet and M.Teisseire, .Incremental Mining of Sequential Patterns in Large Databases (PS),. *Actes des 16ièmes Journées Bases de Données Avancées (BDA’00)*, Blois, France, October 2000.

[13] S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas, .Incremental and Interactive Sequence Mining,. *In Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM.99)*, pages 251-258, Kansas City, MO, USA, November 1999.

[14] Necip Fazil Ayan, .UPDATETING LAEFW ITEMSETS WITH EARLY PRUNING,. Master Thesis, The institute of engineer and science of Bilkent University, July 1999.

[15] Necip Fazil Ayan, Abdullah Uz Tansel, and Erol Arkun, .An efficient algorithm to update large itemsets with early pruning,. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD.99)*, August 15-18, 1999, San Diego, CA USA pp.287-291.

[16] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, .An efficient algorithm for the incremental updating of association rules in large database,. *In Proceedings of the 3rd Intl. Conf. On Knowledge Discovery and Data Mining (KDD.97)*, pages263-266, Newport Beach, California, USA, August 1997.

[17] Ahmed Ayad, Nagwa El-Makky and Yousry Taha, .Incremental Mining of Constrained Association Rules,. *First SIAM International Conference on DATA MINING*, April 5-7, 2001, Chicago USA
[18] M. Zaki, .Scalable Data Mining for Rules,. PHD Dissertation, University of Rochester-New York, 1998
[19] Q. Zheng, K. Xu, W. Lv, and S. Ma, .Intelligent Search of Correlated Alarms from Database Containing Noise Data,. *Proceedings of the 8th International IFIP/IEEE Network Operations and Management Symposium (NOMS 2002)*, April, 2002, to be published, available at http://arXiv.ogr/abs/cs.NI/0109042.
[20] Ashoka Savasere, Edward Omiecinski, and Shamkant Navathe. An efficient algorithm for mining association rules in large database. In Proceedings of 21st Intl. Conf. on Very Large Databases (VLDB'95). Pages 432-444, Zurich, Switzerland, September 1995.