Mining Positive and Negative Sequential Pattern in Incremental Transaction Databases

Vinay Kumar Khare,Vedant Rastogi
M.Tech Student, Assistant Professor
Department of Computer Science Engineering
Institute of Engineering & Technology, Alwar, Rajasthan, India

ABSTRACT

Positive and negative sequential patterns mining is used to discover interesting sequential patterns in a incremental transaction databases, and it is one of the essential data mining tasks widely used in various application fields. Implementation of this approach, construct tree for appended transactions (new upcoming data) and will merge this tree with existing tree (tree of existing transactions) to get the Updated tree. Positive and negative sequential Patterns mining is an aim to find more interesting sequential patterns, considering the minimum support of each data item in a sequence database. Generally, the generation order of data elements is considered to find sequential patterns. Positive sequential patterns states that these items were occur with one and another. Actually the absence of certain itemset may imply the appearance of other itemsets as well. The absence of itemsets thus is becoming measurable in many applications. Negative sequential patterns could assist product recommendation systems to make more accurate decisions. This approach will reduce the mining time for incremental database if the Existing database has lots of transactions and Appended database having few transactions.

Keywords— **Appended, Database, Existing, Itemsets, Negative, Positive, Patterns, Sequential**

INTRODUCTION

Data mining is the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data Mining is the process of discovering new patterns from large data sets. Data mining (sometimes called knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining can be viewed as a result of the natural evolution of information technology. The database system industry has an evolutionary growth in the development of the following functionalities data collection and database creation, data management (including data storage and retrieval, and database transaction processing), and advanced data analysis (involving data warehousing and data mining). For instance, the early development of data collection and database creation mechanisms served as a prerequisite for later development of effective mechanisms for data storage and retrieval, and query and transaction processing. With numerous database systems offering query and transaction processing as common practice, advanced data analysis has naturally become the next target.

1.1 ASSOCIATION RULES

Association rules are statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk. Obtained from reference [3] "An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent. Association rules are created by analyzing data for frequent 'if/then' patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true. In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, and catalog design and store layout [4]. Support is an indication of how frequently the items appear in the database. The support of an association rule is the percentage of groups that contain all of the items listed in that association rule.

Support in an association rule

Support is an indication of how frequently the items appear in the database. The support of an association rule is the percentage of groups that contain all of the items listed in that association rule [6].

Support of item 'a' = (p/q)

p = Number of groups containing that item 'a'
q = Total number of groups

Confidence in an association rule

Confidence indicates the number of times the if/then statements have been found to be true. The confidence value indicates how reliable this rule is. The higher the value, the more often this set of items is associated together [7].

Confidence ('A' -> 'B') = Support ('A'U'B') / Support ('A')

Support ('A'U'B') = number of groups containing 'A' and 'B'.
Support ('A') = number of group containing 'A'.
In data mining, association rules are useful for analyzing and predicting customer behaviour. They play an important part in shopping basket data analysis, product clustering, and catalogue design and store layout [8].

1.2 FREQUENT PATTERNS

As stated by Jiawei Han et al [9] Frequent Patterns are itemsets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. For example, a set of items, such as milk and bread that appear frequently together in a transaction data set is a frequent itemset. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data.

Moreover, it helps in data indexing, classification, clustering, and other data mining tasks as well. Thus, frequent pattern mining has become an important data mining task. Frequent pattern mining was first proposed by Agrawal et al. (1993) for market basket analysis in the form of association rule mining. It analyses customer buying habits by finding associations between the different items that customers place in their “shopping baskets”. For instance, if customers are buying milk, how likely are they going to also buy cereal on the same trip to the supermarket. Such information can lead to increased sales by helping retailers do selective marketing and arrange their shelf space [10].

1.3 SEQUENCE DATABASE

A sequence database consists of sequences of ordered elements or events, recorded with or without a concrete notion of time. There are many applications involving sequence data. Typical examples include customer shopping sequences, Web click streams, biological sequences, sequences of events in science and engineering, and in natural and social developments. So a sequence is represented as an ordered list of data elements. A sequence database can be represented as a tuple <SID, Sequence Item List>, where SID: represents the sequence identifier and Sequence Item List specifies the sequence [11].

1.4 FINDING SEQUENTIAL PATTERNS

Terminology the length of a sequence is the number of item sets in the sequence. A sequence of length k is called a k-sequence. The sequence formed by the concatenation of two sequences x and y is denoted as x.y. The support for an item set i is defined as the fraction of customers who bought the items in i in a single transaction. Thus the itemset i and the l-sequence (i) have the same support. An item set with minimum support is called a large itemset or Litemset. Note that each itemset in a large sequence must have minimum support. Hence, any large sequence must be a list of Litemsets. Sequential pattern mining is the mining of frequently occurring ordered events or Subsequence’s as patterns. An example of a sequential pattern is “Customers who buy a Canon digital camera are likely to buy an HP colour printer within a month.” For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection. Notice that most studies of sequential pattern mining concentrate on categorical (or symbolic) patterns, whereas numerical curve analysis usually belongs to the scope of trend analysis and forecasting in statistical time-series analysis [13].

1.5 THE PROPOSED WORK

Mining Positive & Negative Sequential pattern from databases is useful for knowledge discovery. The patterns were mined only from the Existing transaction database. New upcoming transactions databases cannot be merged into existing transaction database. So every time new transactions database is mined separately. In this approach we can easily update existing transaction database with the appended transaction database. The Merged transaction database (updated database) will be mined to get the Positive & Negative Sequential patterns. Merging of Existing and Appended database is performed by using the updated compact pattern tree approach. Proposed model is Mining Positive and Negative Sequential patterns in incremental transaction Databases. To mine Positive and Negative Sequential patterns in incremental transaction database in this Approach we can update, existing transaction database with appended transaction database by the use of Updated Compact pattern tree approach then according to their support the new updated transaction database table is maintained and we can mine positive and negative sequential patterns with the help of CPNFSP algorithms proposed by Weimin Quyang and Qinhua Huang. First of all we will have the Existing database,. While constructing the tree we will maintain the I_{list} which is the list of all the items in existing database. Now we will sort this I_{list} in frequency descending order to get the I_{sort}. After getting I_{sort} we will rearrange the constructed tree based on I_{sort}. So we will get rearranged tree. Now when new transactions will come then we will take them in Appended database. This appended database contains all the new upcoming transactions. So for this appended database we will construct the tree, maintain the I_{list} and then we will sort the I_{list} in frequency descending order to get I_{sort} and then we will rearranged the appended tree according to the appended I_{sort}. So at this point of time we will have existing tree, I_{sort} of existing tree, appended tree and the I_{sort} of appended tree. Now we will merge both the I_{sort} and we will get Merged I_{list} and then this Merged I_{list} will be sorted in frequency descending order to get Merged I_{sort}. This Merged I_{sort} is the frequency of all the items in Existing Database and Appended Database in descending order. Now existing tree and appended tree will be reordered bases on this Merged I_{sort}. Finally these reordered trees are merged to get the updated Compact pattern tree. The Updated transaction table is now created on behave of supports of an particular item. Now with the use of CPNFSP Algorithm we can easily mine positive and negative sequential patterns in incremental databases.

1.6 UPDATED COMPACT PATTERN TREE

Step 1: Scanning Existing Transaction Database for construction of Tree and getting I-list1.

Step 2: Sorting I-list based on frequency of each item according to descending order in existing transaction database and getting I-sort1.

Step 3: Scanning Appended Transaction Database for construction of Tree and getting I-list2.

Step 4: Sorting I-list based on frequency of each items according to descending order in appended transaction database and getting I-sort2.

Step 5: Merge these two I-sort to get Merged I-list and then Sort this merged I-list in frequency descending order to get merged I-sort.

Step 6: Restructuring Tree to get updated Tree.

Step 7: We get Updates Transaction table from Tree.

Step 8: Candidate generation for each item with other item occurred.

Step 9: CPNFSP Algorithm is applied on the updated Transaction Table.

Step 10: Positive and Negative Sequential patterns are mined.

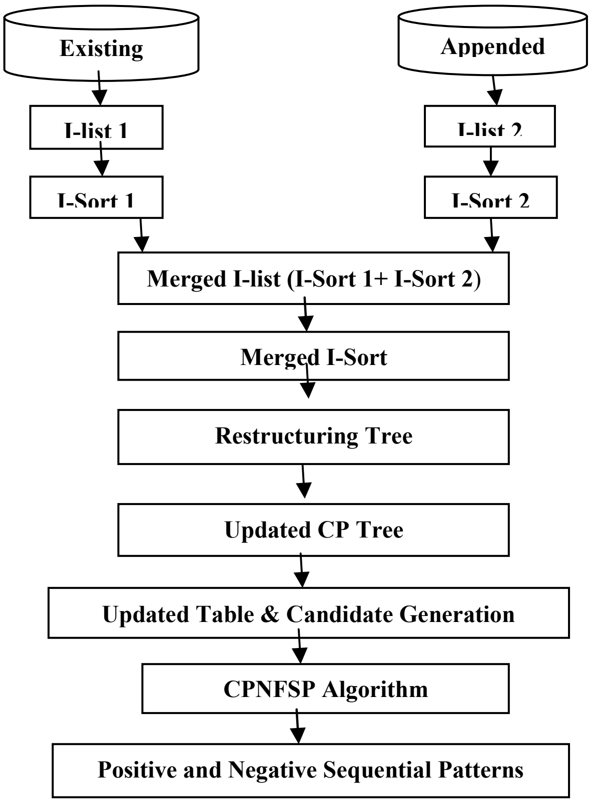

Existing
Appended
I-list 1
I-list 2
I-Sort 1
I-Sort 2
Merged I-list (I-Sort 1+ I-Sort 2)
Merged I-Sort
Restructuring Tree
Updated CP Tree
Updated Table & Candidate Generation
CPNFSP Algorithm
Positive and Negative Sequential Patterns

Figure1.1 Architecture of Updated Compact pattern Tree

1.7 POSITIVE SEQUENTIAL PATTERNS

In a large database of customer transactions, where each transaction consists of customer-id, transaction time, and the items bought in the transaction. Mining sequential patterns over such databases is needed in many real time applications. Data mining is motivated by the decision support Problem faced by most large retail organizations. Database D contains customer transactions. Each transaction consists of the following fields: customer-id, transaction-time, and the items purchased in the transaction. No customer has more than one transaction with the same transaction-time. Each item is a binary variable representing whether an item was bought or not. An item set is a non-empty set of items. A sequence is an ordered list of item sets without loss of generality. By default sequential patterns means positive sequential patterns only [1, 14, 15].

1.8 NEGATIVE SEQUENTIAL PATTERNS

Corresponding to a positive sequential pattern such as (A, B), there are three possible negative sequential patterns, (A, ¬B), (¬A, B) and (¬A, ¬B). For a sequence (A, ¬B) and a certain transaction sequence CS, if A ⊆ CS and ¬B ⊄ CS, we say that CS supports (A, ¬B). Assume there is a negative sequential pattern such as ({i1}, ¬{ i2 , i3}), which means that if i1 is in a customer sequence CS, i2 and i3 would not appeared in the customer sequence CS, but there is a possibility that one of the i2 and i3 is in this customer sequence [1].

For an interesting negative sequential pattern (A, ¬B) as:

(1) A ∩ B = ∅;

(2) Sup (A)>=minsup, sup (B)>=minsup, sup (A∪ ¬B)>=minsup。

(3) Sup (A∪¬B) − sup (A) × sup (¬B)>= mininterest. Here minsup and mininterest given by user mainly domain experts.

By the same way we can define conditions of negative sequential patterns forms as (¬A, B) and (¬A, ¬B).

The ways to find the supports to the negative sequential patterns [11] are

- s (A) =1-s (¬ A);
- s (A∪¬ B) =s (A)-s (A∪B);
- s (¬ A∪B) =s (B)-s (A∪B);
- s (¬ A∪¬ B) =1-s (A)-s (B) +s (A∪B).

1.9 RESULT ANALYSIS

All the experiments have been performed in computer with Intel(R) Pentium(R) Core 2 duo 2.10 GHz CPU and 1 GB RAM. The coding of existing algorithm and proposed algorithm is done in java. The Experiment is done for dataset of 1000 transaction treating 800 transactions as existing transactions and last 200 transactions as appended transactions. Existing transactions such as for 800 transactions the Updated compact pattern tree is constructed and then individually for the rest 200 transactions appended Updated compact pattern tree is constructed and merged these trees in order to mine Positive and Negative Sequential patterns. The least support is 1 in whole experiments. Mining Positive and Negative Sequential patterns from incremental transaction database is useful for knowledge discovery. Tree based approach has been proposed by researchers such as FP tree, CATS tree, CAN tree, CP tree etc are considered for frequent pattern mining. In this dissertation we used Updated Compact Pattern tree for constructing the tree for entire database and to mine Positive and Negative sequentional patterns we have used CPNFSP Algorithm. By Merging this two Approaches we have mined Positive and Negative sequential patterns in incremental transactions databases. The tree is constructed for all the items in transactional database while the new transactions updated. We have evaluated the performance of proposed algorithm on database such as Synthetic dataset.

Figure 2 represents the time required for extracting Positive and Negative Sequential patterns in seconds with various Transaction Databases for proposed CPNFSP algorithm (Dataset with 1000 transactions). In the given figure 2, On X-axis, we have taken Time starting from 14.4 to 16.4 with time interval of 0.2 seconds and on Y-axis, we have the proposed CPNFSP algorithm and the existing CPNFSP algorithm and we have taken a dataset of 1000 for existing CPNFSP algorithm and for proposed CPNFSP we have taken 800 as Existing transaction database and 200 for appended transaction database. We found that the time required for Mining Positive and Negative Sequential Patterns for the proposed algorithm is less as compare to the existing CPNFSP algorithm. In experiment we have found that the proposed algorithm is time efficient as compare to existing algorithm. The Proposed algorithm mines Positive and Negative sequential patterns in incremental database. So by this approach we can easily mines existing transaction database with the appended transaction database and no need to mine transaction database separately for each transaction we can merge it to existing database by using updated compact tree and then mine positive and negative sequential patterns.

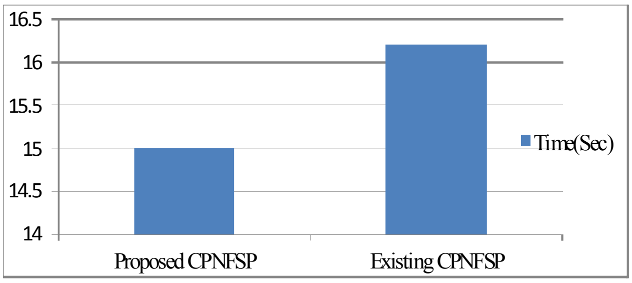


Figure 2, times required for extracting positive and negative sequential patterns in milliseconds with synthetic dataset of 1000 transaction for Existing CPNFSP and for Proposed CPNFSP (Existing transaction dataset-800 & Appended transaction dataset-200).

In the given figure 3, On X-axis, we have taken Time starting from 0 to 35 with time interval of 5 seconds and on Y-axis, we have taken synthetic datasets starting from 1000 to 5000 synthetic datasets. The proposed CPNFSP algorithm and the existing CPNFSP algorithm is taken for time analysis when datasets are increasing. In experiment we have found that the proposed algorithm is time efficient as compare to existing algorithm. We found that as the datasets are increasing from 1000 to 5000 then time is also increasing. At every 1000 dataset the proposed algorithm consumes less time as compare to the existing algorithm. The Proposed algorithm mines Positive and Negative sequential patterns in incremental database.

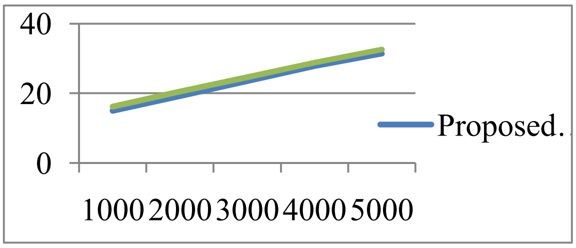
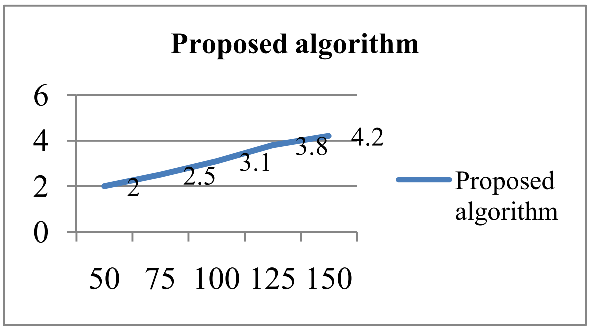


Figure 3, times required for extracting positive and negative sequential patterns in milliseconds with synthetic dataset of 1000 transactions to 5000 transactions for Existing CPNFSP and for Proposed CPNFSP (Existing transaction dataset-80% & Appended transaction dataset-20%).

To examine the scalability of proposed algorithm we increased the numbers of datasets from 5000 to 15000, with least support-1 .the result are shown in Figure 5.5. The executing time is increased almost linearly with the increasing of dataset size. So it can be said that our proposed algorithm has good scalable performance



CONCLUSION & FUTURE SCOPE

The proposed model mines positive and negative Sequential Patterns and gives good results, the tree is constructed for all the items from transactional database while the new transactions updated. In our approach we constructed tree independently and merged to the existing one, this reduces the time complexity in constructing the tree for entire database. Then we get updated table of transaction so in this updated transaction table positive and negative sequential patterns are mined.

As a future work, we will take supports according to user requirement and then mine positive and negative sequential patterns and we can use fuzzy logic for different supports as user specified. We can take large transaction database for mining positive and negative sequential patterns.

REFERENCES

[1] Weimin Ouyang, Qinhua Huang,"Mining Positive and Negative Sequential Patterns with Multiple Minimum Supports in Large Transaction Databases", IEEE Second WRI Global Congress on Intelligent Systems 2010.

[2] Sue-Chen Hsueh1, Ming-Yen Lin2, Chien-Liang Chen2, "Mining Negative Sequential Patterns for E-Commerce Recommendations", IEEE, Asia-Pacific Services Computing Conference, 2008.

[3] R. Uday Kiran and P. Krishna Reddy," An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules", Computational Intelligence and Data Mining, CIDM '09, IEEE Symposium on 2009.

[4] William Cheung and Osmar R. Zaiane - Incremental Mining of Frequent Patterns Without Candidate Generation or Support Constraint, University of Alberta, Edmonton, Canada { wcheung, zaiane} @cs.ualberta.ca 2003.

[5] Yun, H., Ha, D., Hwang, B., and Ryu K. H. "Mining association rules on significant rare data using relative support.", The Journal of Systems and Software 67, 2003, pp. 181-191.

[6] Savasere A, Omiecinski E and Navathe S. "Mining for Strong Negative Associations in a Large Database of Customer Transactions" In Proc. 1998 Int. Conf. on Data Engineering, pp.494-502.

[7] Luis, R., Redol, J., Simoes, D., Horta, N., "Data Warehousing and Data Mining System Applied to ELearning, Proceedings of the II International Conference on Multimedia and Information & Communication Technologies in Education, Badajoz, Spain, December 3-6th 2003.

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

[9] Jiawai Han, Jian Pai, Yiwen Yin, Runying Mao-Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach © Kluwer Academic Publishers 2004.

[10] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, R (1996). "The KDD Process for Extracting Useful Knowledge from Volumes of Data," Communications of the ACM, (39:11), pp.27-34.

[11] Han, J., Kamber, M. (2001), Data Mining: Concepts and Techniques, Morgan-Kaufmann Academic Press, San Francisco.

[12] Hand, D. J. (1998), “Data Mining: Statistics and More?”, The American Statistician, May (52:2), 112-118.

[13] Han, J., Kamber, M. (2001), Data Mining: Concepts and Techniques, Morgan-Kaufmann Academic Press, San Francisco.

[14] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R, Editors (1996), Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, MA.

[15] Rajagopalan, B., Krovi, R. (2002), “Benchmarking Data Mining Algorithms”, Journal of Database Management, Jan-Mar, 13, 25-36.