Improvement of Business analysis using frequent pattern tree

Kilari Teja Seethal Kumar[1], Mr. K. John Paul[2]

[1] Student(Mtech),

[2] Associative Professor, CSE department, Nova College of Engineering & Technology

Abstract: Increasing business is the main task for every company. Planning, implementation and improvements of the company growth is also very important for us. Many business companies generate large data from their daily transactions. It is very useful to store all the transactions for the future improvement. In data mining research frequent patterns in transactional databases, time series databases, relational databases and various kinds of databases. In this paper, proposed system focus on the steps taken to improve the business according to the association rules and an advanced novel frequent pattern tree (FP-tree) structure. Data mining shows the better techniques for the improvement of the global business. The proposed system focus on two techniques: 1. huge databases are compressed into small data structure which reduces the cost. 2.) Apriori algorithm with support and confidence. The results show the performance of the proposed system.

Keywords: Business, association rules, data mining.

Introduction:

Frequent pattern mining plays an essential role in mining associations, correlations sequential patterns episodes, multi-dimensional patterns, max-patterns, partial periodicity, emerging patterns, and many other important data mining tasks.

Data mining techniques can be categorized according to the objectives they follow and the results they offer, which obtains computer as a tool and makes use of the skill and knowledge significance to comprehend and explain the problem. Various data mining techniques such as, decision trees, association rules, and neural networks are already presented and become the point of attention for several years. Association rule mining technique is the most efficient data mining technique to search hidden or desired pattern among the huge amount of data. It is responsible to get correlation relationships among various data attributes in a large set of items in a database. A huge quantity of interesting relevance or related association across the itemsets has been determined by association rules mining. A typical example of the association rules mining is the market basket analysis. Association rules research assists to find the relationship among different products (items) in transaction databases and to find out the customer buyer behaviors, such as the purchase of a commodity impact on other goods. The results can be applied to goods shelf layout, storage arrangements, and classification of user according to buying patterns. Association Analysis is the detection of hidden pattern or condition that occurs frequently together in a given data. Association Rule mining techniques finds interesting associations and correlations among data set. An association rule is a rule, which entails certain association relationships with objects or items, for example the interrelationship of the data item as whether they occur simultaneously with other data item and how often. These rules are computed from the data and, association rules are calculated with help of probability. It has a mentionable amount of practical applications, including classification, XML mining, spatial data analysis, and share market and recommendation systems. This rule measure with support to ensure every dataset treated equally in classical model. The perception of association rule mining suggests the support confidence level outline and condensed association rule mining to the discovery of frequent item sets. Rule support and confidence are two measures of interestingness. Association rules are regarded as appealing if a minimum support and a minimum confidence threshold is satisfied. Boolean association rule mining is more extensively used than other kinds of association rule mining. Association rule mining procedure can be finished in four steps. First data preparation and pick the required data, second produce itemsets that determines the rule constraints for knowledge, third mine k frequent itemsets using the new database and fourth produce the association rule that sets up the knowledge base. The paper discusses an algorithm to mine association rules and the support and confidence are studied.

For Apriori algorithm, there are two disadvantages. First, it must scan data sets repeatedly, which may direct to generate a large number of candidate itemsets. Second, the rare information is difficult to dig because the limitation of algorithm. The competence of Apriori algorithm has a greatly effect on performance and practicality of related data mining system. When there are several elements and minimum support threshold is low, the competence of Apriori algorithm is easy to become a bottleneck of performance

Related Work:

Clustering of FP-tree nodes by path and by item prefix sub-tree. Since there are many operations localized to single paths or individual item prefix sub-trees, such as pattern matching for node insertion, creation of transformed prefix paths for each node ai, etc., it is important to cluster FP-tree nodes according to the tree/subtree structure. That is, (1) store each item prefix sub-tree on the same page, if possible, or at least on a sequence of continuous pages on disk; (2) store each subtree on the same page, and put the shared prefix path as the header information of the page, and (3) cluster the node-links belonging to the same paged nodes together, etc. This also facilitates a breadth-first search fashion for mining all the patterns starting from all the nodes in the header in parallel.

In the incremental mining, data are not only added but also obsolete data are being deleted. The main aim of incremental mining algorithm is to re-run the mining algorithm on the only incremented database. However, it is obviously less efficient than traditional association rule mining since previous mining rules are not utilized for discovering new rules while the updated portion is usually small compared to the whole dataset. Consequently the efficiency and the effectiveness are most crucial issue of incremental mining. Algorithms should be such that only updated transactions and previous mined rules to be taken into account for generating new rules [4].

Apriori is a seminal algorithm proposed by R. Agrawal and R.Srikant in 1994 for Apriori is the best-known algorithm to mine association rules. It uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support.

Improvement of Business Environment:

To improve the business analysis, the proposed system focus on association rule mining and apriori algorithm. Here some of the steps to improve the global market. Using data mining techniques there are no of business analytics have been introduced for the feature enhancement.

Proposed Association Analysis:

It is used to discover the associations based on the relationships hidden in large datasets. There are no of transactions based on the on the items and item sets.

The sample transactions on super market:

TID	Items
1	{Bread,Milk}
2	{Bread,Diapers,Beer,Eggs}
3	{Milk,Diapers,Beer,Cola}
4	{Bread,Milk,Diapers,Beer}

Table: 1, Sample transactions for super market

From the above transactions we can extract the data from the above transaction.

{Talcum powder→Face wash→Face cream}

{Milks→Eggs}

{Beer→Breezer→Liquor}

In the above transactions, we are showing the associations of the items present in the dataset as per the transactions. To form the real and exact associations we are using two techniques that are support and confidence.

Support: It is very important to know that a support of the rule may occur by very low chance.

Confidence: On the other hand, measures the reliability of the rule.

In this paper, our consideration is mainly based on support and confidence. Because to improve the business analytics these two are the important for formation of association rules.

For Support:

Support is the important rule because it measures the similar association rule.

Consider the rule {Milk, Diapers}→Beer Since the support count of the {Milk, Diapers, Beer} is 2 and total no of transactions is 5.

Support = total similar transactions/total transactions

Support=2/5=0.4

Support for second rule= {Bread,Milk}

Occurrence= 4/6= 0.67

For Confidence:

Confidence determines how frequently items appear in transactions.

Confidence= Support count / items contain in transactions

For first transaction Confidence= 2/3=0.67

For second transaction =4/5=0.8

Mining Frequent Patterns using FP-tree:

Construction of a compact FP-tree ensures that subsequent mining can be performed in a rather compact data structure. However, this does not automatically guarantee that subsequent mining will be highly efficient since one may still encounter the combinatorial problem of candidate generation if we simply use this FP-tree to generate and check all the candidate patterns.

In this section, we will study how to explore the compact information stored in an FP-tree and develop an efficient mining method for frequent pattern mining. Although there are many kinds of frequent patterns that can be mined using FP-tree, this study will focus only on the most popularly studied one [3]: mining all patterns, i.e., the complete set of frequent patterns. Methods for mining other frequent patterns, such as max-pattern [5], i.e., those not subsumed by other frequent patterns, will be covered by subsequent studies.

Algorithm 1 (FP-growth: Mining frequent patterns with FP-tree and by pattern fragment growth)

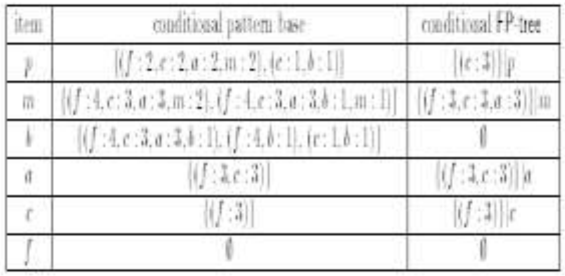
Input: FP-tree constructed based on Algorithm 1, using DB and a minimum support threshold €.

Output: The complete set of frequent patterns.

Method: Call FP-growth (FP-tree ; null), which is implemented as follows.

```
Procedure FP-growth (Tree,$\eta$ )

    IF Tree contains a single path P
      THEN FOR EACH combination (denoted as β) of the nodes in the path P DO
        generate pattern β U $\eta$ with support = minimum support of nodes in β;
      ELSE FOR EACH $\eta_i$ in the header of Tree DO {
     generate pattern β = $\eta_i$ U $\eta$ with support = $\eta_i$ ,support;
     Construct _'s conditional pattern base and then β 's conditional FP-tree $Tree_\beta$;
     IF $Tree_\beta \neq \varphi$
      THEN Call FP-growth ($Tree_\beta$ ,β)}
```



item	conditional pattern base	conditional FP-tree
p	$\{(f:2,c:2,a:2,m:2),(c:1,b:1)\}$	$\{(c:3)\}\|p$
m	$\{(f:4,c:3,a:3,m:2),(f:4,c:3,a:3,b:1,m:1)\}$	$\{(f:3,c:3,a:3)\}\|m$
b	$\{(f:4,c:3,a:3,b:1),(f:4,b:1),(c:1,b:1)\}$	$\emptyset$
a	$\{(f:3,c:3)\}$	$\{(f:3,c:3)\}\|a$
c	$\{(f:3)\}$	$\{(f:3)\}\|c$
f	$\emptyset$	$\emptyset$

Table: Mining of all-patterns by creating conditional (sub)-pattern bases

From the algorithm and its reasoning, one can see that the FP-growth mining process is a divide-and-conquer process, and the scale of shrinking is usually quite dramatic. If the shrinking factor is around 20~100 for constructing an FP-tree from a database, it is expected to be another hundreds of times reduction for constructing each conditional FP-tree from its already quite small conditional frequent pattern base

Conclusion:

In this Paper, Frequent pattern tree (FP-tree), for storing compressed, crucial in-formation about frequent patterns, and developed a pattern growth method, FP-growth, for efficient mining of frequent patterns in large databases. There are several advantages of FP-growth over other approaches: (1) It constructs a highly compact FP-tree, which is usually substantially smaller than the original database, and thus saves the costly database scans in the subsequent mining processes. (2) It applies a pattern growth method which avoids costly candidate sets generation and test by successively concatenating frequent 1-itemset found in the (conditional) FP-trees : It never generates any combinations of new candidate sets which are not in the database because the item set in any transaction is always encoded in the corresponding path of the FP-trees . In this context, the mining methodology is not Apriori-like (restricted) generation-and-test but frequent pattern (fragment) growth only. The major operations of mining are count accumulation and prefix path count adjustment, which are usually much less costly than candidate generation and pattern matching operations performed in most Apriori-like algorithms. (3) It applies a partitioning-based divide-and-conquer method which dramatically reduces the size of the subsequent conditional pattern bases and conditional FP-trees. Several other optimization techniques, including ordering of frequent items, and employing the least frequent events as sufix, also contribute to the efficiency of the method.

References:

[1] R. Agrawal, T. Imielinski, and A. N. Swami, “Mining association rules between sets of items in large databases,” in *Proc. SIGMOD*,
Washington, DC, USA, 1993, pp. 207–216.

[2] B. Goethals and M. J. Zaki, “Advances in frequent itemset mining implementations: Introduction to FIMI’03,” in *Proc. ICDM*, 2003.

[3] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules,” in *Proc. 7th ICDT*,
Jerusalem, Israel, 1999, pp. 398–416.

[4] D. Xin, J. Han, X. Yan, and H. Cheng, “Mining compressed frequent-pattern sets,” in *Proc. 31st Int. Conf. VLDB*, Trondheim,
Norway, 2005, pp. 709–720.

[5] V. Chvatal, “A greedy heuristic for the set-covering problem,” *Math. Oper. Res.*, vol. 4, no. 3, pp. 233–235, 1979.

[6] G. Grahne and J. Zhu, “Efficiently using prefix-trees in mining frequent itemsets,” in *Proc. FIMI*, 2003.

[7] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. Depth-first generation of large itemsets for association rules. In IBM Technical Report RC21538, October, 1999.

[8] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. In Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining), (to appear), 2000.

[9] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases, pages 487{499, Santiago, Chile, September 1994.

[10] R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering, pages 3{14, Taipei, Taiwan, March 1995.

[11] R. J. Bayardo. E_ciently mining long patterns from databases. In Proc. 1998 ACM-SIGMOD Int.

Conf. Management of Data, pages 85{93, Seattle, Washington, June 1998.

[12] R. J. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining on large, dense data sets. In Proc. 1999 Int. Conf. Data Engineering (ICDE'99), Sydney, Australia, April 1999.