International Journal of Research in Advent Technology, Vol.2, No.10, October 2014
E-ISSN: 2321-9637

An Efficient Frequent Item Mining using Various Hybrid Data Mining Techniques in Super Market Dataset

P.Abinaya[1], Dr. (Mrs) D.Suganyadevi[2]
M.Phil. Scholar[1], Department of Computer Science,STC,Pollachi[1].
Director, PG Department of Computer Applications, STC, Pollachi[2]
Email: p.abinaya.msc@gmail.com[1], sugan.devi1@gmail.com[2]

Abstract- The computerization of many business and government transactions, huge amounts of data have been stored in computers. The existing database systems do not provide the users with the necessary tools and functionalities to capture all stored information easily. Most enterprise generates unlimited amounts of data and stores them in databases. This large volume of data contains valuable hidden knowledge. The techniques that allow the enterprise to extract the information from Database are known as knowledge discovery in database (KDD) or data mining. Market basket analysis is an important component of analytical system in retail organizations to determine the placement of goods, designing sales promotions for different segments of customers to improve customer satisfaction and profit of the supermarket. In the departmental store each customer purchases a different set of products, in different quantities, at different times. In this paper using the frequent item mining concept which is derived through FP tree growth algorithm, Association rule mining, k means algorithm and Top k rule algorithm. Association rule mining is one of the most popular data mining methods. In this paper proposed the performance comparison of K-Apriori and Top k rule based FP-growth algorithms. The performance is analyzed based on the execution time for different number of instances and confidence in Super market data set. These algorithms are presented together with some experimental data. Our performance study shows that the Top k rule FP-growth method is efficient and scalable and is about an order of magnitude faster than the k-Apriori algorithm.

Index Terms- KDD, Data mining, Market basket analysis, FP tree growth, Association rule, K-means.

1. INTRODUCTION

Data mining plays a vital role in computer application and their development. Data mining is one of the large areas and this is used in various type of real time application. This is mainly used in marketing area. One of the challenges for companies that have invested heavily in customer data collection is how to extract important information from their vast customer databases and product feature databases, in order to gain competitive advantage. Market basket analysis using customer interest profile and interests on particular products for one-to-one marketing, purchasing patterns in a multi-store environment to improve the sales.[6] Market basket analysis has been intensively used in many companies as a means to discover product associations and base a retailer's promotion strategy on them.

Informed decision can be made easily about product placement, pricing, promotion, profitability and also finds out, if there are any successful products that have no significant related elements. Similar products can be found so those can be placed near each other or it can be cross-sold. A retailer must know the needs of customers and adapt to them. Market basket analysis is one possible way to find out which items can be put together. Market basket analysis gives retailer good information about related sales on group of goods basis Customers who buys bread often also buy several products related to bread like milk, butter or jam.[8] It makes sense that these groups are placed side by side in a retail center so that customers can access them quickly. Such related groups of goods also must be located side-by-side in order to remind customers of related items and to lead them through the center in a logical manner.

2. OVERVIEW OF DATA MINING

The term data mining and knowledge discovery in database are often used interchangeable.KDD (Knowledge Discovery in Database) is the process of finding useful information and pattern in data. Data mining is the use of algorithm to extract the information and patterns derived by the KDD process.[1]

Data mining is the analysis of (often large) observational data sets to find unsuspected

relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. Data mining is an interdisciplinary field bringing together techniques from machine learning, pattern recognition, statistics, databases, and visualization to address the issue of information extraction from large data bases.

The most common data mining tasks are Description, Estimation, Prediction, Classification, Clustering, and Association. In **Association** the Presence of one set of items in a transaction implies other set of items. **Classification** develops profiles of different groups. **Sequential Patterns** identifies sequential patterns subject to user constraints. **Clustering** helps segmenting database into subsets or clusters. [5]

The two fundamental goals of Data-Mining are Prediction and Description. Prediction makes use of the existing variables in the database in order to predict unknown or future values of interest. Description focuses on finding patterns describing the data and the subsequent presentation for user interpretation.

3. RELATED WORKS

Agrawal R, Srikant R [1] (1994) consider the problem of discovering association rules between items in a large database of sales transactions. This paper provide two new algorithms for solving this problem that are fundamentally different from the known algorithms. Experiments with synthetic as well as real-life data show that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. This paper is to show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called Apriori Hybrid. Scale-up experiments show that Apriori Hybrid scales linearly with the number of transactions. Apriori Hybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

R. Agrawal, T. Imielinski [2] (1993) Provide large database of customer transactions. Each transaction consists of items purchased by a customer in a visit. This paper present an efficient algorithm that generates all significant association rules between items in the database. The algorithm incorporates buffer management and novel estimation and pruning techniques. Present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm.

Pavel Berkhin, [3] In this paper, discussion on different clustering techniques and then comparison by analyzing various types of cluster.

Madhuri V.Joseph et al [4] (2013) In this paper provides an explanation and comparative study on some of the most common data mining techniques and methods in use today in day life and business predictions. Data mining may be viewed as the extraction of patterns and models from observed data or a method used for analytical process designed to explore data. There are many different methods, which may be used to predict the appropriate class for the objects. The majority of data mining techniques can deal with different data types. Though there are a number of other techniques and many variations of the methods described, one of the techniques from the mentioned group is almost always used in real world deployments of data mining systems.

Aastha Joshi et al [5] (2013) In this paper, discuss on clustering is a process of putting similar data into groups. Clustering cab be considered the most important unsupervised learning technique so as every other problem of this kind; it deals with finding a structure in a collection of unlabeled data. This paper uses six types of clustering techniques k-Means clustering, Hierarchial Clustering, DBSCAN Clustering, OPTICS, STING.

Weng, S.-S., Liu, J.-L [6] (2004) Most recommendation systems face challenges from products that change with time, such as popular or seasonal products, since traditional market basket analysis or collaborative filtering analysis are unable to recommend new products to customers due to the fact that the products are not yet purchased by customers. Although the recommendation systems can find customer groups that have similar interests as target customers, brand new products often lack ratings and comments. The advantage of this research is its ability of recommending to customers brand new products or rarely purchased products as long as they fit customer interest profiles; a deduction which traditional market basket analysis and collaborative filtering methods are unable to do.

4. METHODOLOGY

The main objective of this dissertation is to increase the customer satisfaction and increase the revenue of the TS super market.

Following algorithm were used in this dissertation

- Association Rule mining
- Cluster analysis
  - K means algorithm
  - Top K rule algorithm

Market Basket Analysis is taken from TS store. In this dissertation transaction is observed from copy bills or invoice copies which contain the items purchased by

different customers. Copy bill is the duplicate copy of the bills generated in the system which is used for future reference.

4.1 K MEANS

K Means algorithm for mining frequent item sets and deriving Association rules from binary data are proposed here. K-FP is an enhanced version of FP growth algorithm based on the FP property and the Association rule generation procedure.

In K-Means, k random points are chosen initially from the whole Market transaction dataset. Each point is assigned to one of the k groups based on the smallest distance between the point and the chosen centers. The distance can be measured using a distance measure such as Euclidean Distance. Once a pass over the dataset is completed, the new centers of the newly formed k groups are found. The algorithm terminates if the new centers and the previous centers only differ by a satisfiable threshold, called convergence delta. Otherwise, it continues to make a new pass over the dataset with the new centers.

***Proposed Algorithm*:** K-Means clustering uses multiple passes over data to refine the cluster centers. The most computationally intensive part of the algorithm is the number of scans that have to be performed. We also eliminate a further overhead by delaying storing the association of point to the cluster centers until after all the iterations complete.

***Algorithm 1:* K-Means**
Step1: centers - stores k centers
Step2: statVector - statVector is an array of k Statistics object whose each index corresponds to one of the k centers. Also Statistics object stores int sum1, sum2, ... (for each dimensions of the point) and int count
Step3: AddPoint(int X, int Y, int Z) // Adding a Point to Statistics Object
*sum*1+ = *X*; *sum*2+ = *Y* ; *sum*3+ = *Z*; + + *count*;
Step4: Merge(Statistics other)
*sum*1+ = *other* .*sum*1; *sum*2+ = *other* .*sum*2; *sum*3+ = *other* .*sum*3; *count*+ =
other .*count*;
Step5: Finalize()
*sum*1= = *count*; *sum*2= = *count*; *sum*3= = *count*;

4.2 FP GROWTH

FP stands for frequent pattern. It generates frequent item sets without the use for candidate generation. FP-growth adopts the divide and conquer strategy. FP-Growth algorithm encodes the data set using compact data structure called an FP-tree and then extracts frequent item sets directly from this structure. It is based on a prefix tree representation of the given database of transactions, which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as to generate the FP-tree for all the transactions.

Algorithm 2:Improved FP Growth

***Input*:** A transaction database DB and a minimum support threshold ξ.
***Output*:** FP-tree, the frequent-pattern tree of DB.
***Method*:** The FP-tree is constructed as follows:
A. Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.
B. Create the root of an *FP-Tree*, T, and label it as "null". For each transaction Trans in DB do the following-
(b1) Select the frequent items in Trans and sort them according to the order of FList. Let the sorted frequent-item list in Trans be [p | P], where p is the first element and P is the remaining list. Call insert tree ([p | P], T).
(b2) The function insert tree([p | P], T) is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N‟s count by 1; else create a new node N, with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree (P, N) recursively.

Algorithm to find Frequent Item sets using FP-Growth algorithm:

The *FP- Growth* algorithm for mining frequent patterns using *FP-Tree* by pattern fragment growth is:
Input: a *FP-Tree* constructed with the algorithm mentioned in Algorithm for FP-tree construction
D - Transaction database
ξ - Minimum support Threshold.
Output: The complete set of frequent patterns.
Method:
Call FP-growth (*FP-tree*, *null*)
Procedure FP-growth (*Tree, A*)
{
if (*Tree* contains a single path *P)*
then for each **(**combination (denoted as *B*) of the nodes in the path *P*) *do*
generate pattern *B*∪*A* with support = minimum support of nodes in *B*;
else (for each *ai* in the header of *Tree*) do
{
generate pattern *B = ai*∪*A* with *support = ai.support*;
construct *B*‟s conditional pattern base and then *B*‟s conditional *FP-Tree TreeB*;
if (*TreeB* ≠ ∅)
{

call *FP-growth* (*TreeB, B*)
}
}
}

The next step is to generate positive and negative class association rules. It firstly finds the rules contained in F which satisfy min_sup and min_conf threshold. Then, it will determined the rules whether belong to the set of positve class correlation rules P_AR or the set of negative class correlation rules N_AR.

4.3 TOP K RULE

The algorithm main idea is the following. TopKRules first sets an internal *minsup* variable to 0. Then, the algorithm starts searching for rules. As soon as a rule is found, it is added to a list of rules L ordered by the support. The list is used to maintain the top-k rules found until now. Once *k* valid rules are found, the internal *minsup* variable is raised to the support of the rule with the lowest support in L. Raising the *minsup* value is used to prune the search space when searching for more rules. Thereafter, each time a valid rule is found, the rule is inserted in L, the rules in L not respecting *minsup* anymore are removed from L, and *minsup* is raised to the value of the least interesting rule in L. The algorithm continues searching for more rules until no rule are found, which means that it has found the top-k rules.

To search for rules, TopKRules does not rely on the classical two steps approach to generate rules because it would not be efficient as a top-k algorithm (as explained in the introduction). The strategy used by TopKRules instead consists of generating rules containing a single item in the antecedent and a single item in the consequent. Then, each rule is recursively grown by adding items to the antecedent or consequent.

To select the items that are added to a rule to grow it, TopKRules scans the transactions containing the rule to find single items that could expand its left or right part. We name the two processes for expanding rules in TopKRules *left expansion* and *right expansion*. These processes are applied recursively to explore the search space of association rules.

Another idea incorporated in TopKRules is to try to generate the most promising rules first. This is because if rules with high support are found earlier, TopKRules can raise its internal *minsup* variable faster to prune the search space. To perform this, TopKRules uses an internal variable R to store all the rules that can be expanded to have a chance of finding more valid rules. TopKRules uses this set to determine the rules that are the most likely to produce valid rules with a high support to raise *minsup* more quickly and prune a larger part of the search space.

TOP K Rule Algorithm

TOPKRULES(T, k, minconf) R := Ø. L := Ø. minsup := 0.
1. Scan the database T once to record the tidset of each item.
2. FOR each pairs of items i, j such that |tids(i)| ×|T| ≥ minsup and |tids(j)| ×|T| ≥ minsup:
3. sup({i}→{j}) := |tids(i) ∩ tids(j)| / |T |.
4. sup({j}→{i}) := |tids(i) ∩ tids(j)| / |T|.
5. conf({i}→{j}) := |tids(i) ∩ tids(j)| / |tids(i)|.
6. conf({j}→{i}) := |tids(i) ∩ tids(j)| / |tids(j)|.
7. IF sup({i}→{j}) ≥ minsup THEN
8. IF conf({i}→{j}) ≥ minconf THEN SAVE({i}→{j}, L, k, minsup).
9. IF conf({j}→{i}) ≥ minconf THEN SAVE({j}→{i}, L, k, minsup).
10. Set flag expandLR of {i}→{j}to true.
11. Set flag expandLR of {j}→{i}to true.
12. R := R∪{{i}→{j}, {j}→{i}}.
13. END IF
14. END FOR
15. WHILE ∃r ∈ R AND sup(r) ≥ minsup DO
16. Select the rule rule having the highest support in R
17. IF rule.expandLR = true THEN
18. EXPAND-L(rule, L, R, k, minsup, minconf).
19. EXPAND-R(rule, L, R, k, minsup, minconf).
20. ELSE EXPAND-R(rule, L, R, k, minsup, minconf).
21. REMOVE rule from R.
22. REMOVE from R all rules r ∈ R | sup(r) <minsup.
23. END WHILE

After that, a loop is performed to recursively select the rule *r* with the highest support in R such that *sup*(*r*) ≥ *minsup* and expand it (line *15 to 23*). The idea is to always expand the rule having the highest support because it is more likely to generate rules having a high support and thus to allow to raise *minsup* more quickly for pruning the search space. The loop terminates when there is no more rule in R with a support higher than *minsup*. For each rule, a flag *expand LR* indicates if the rule should be left and right expanded by calling the procedure EXPAND-L and EXPAND-R or just left expanded by calling EXPAND-L.

For all rules of size 1*1, this flag is set to true. The utility of this flag for larger rules will be explained later. Before describing the procedure EXPAND-L and EXPAND-LR, we describe the SAVE procedure. Its role is to raise *minsup* and update the list L when a new valid rule *r* is found. The first step of SAVE is to add the rule *r* to L (line *1*). Then, if L contains more than *k* rules and the support is higher than *minsup*, rules from L that have exactly the support equal to *minsup* can be removed until only

k rules are kept (line *3 to 5*). Finally, *minsup* is raised to the support of the rule in L having the lowest support. (line *6*). By this simple scheme, the top-k rules found are maintained in L.

5. RESULTS AND DISCUSSION

The performance of the proposed algorithm K FP growth with Top k rule and compare with the algorithms K-Apriori on market dataset transaction datasets. All algorithms were written in Java programming language. The configuration of the testing platform is as follows: Windows 7 operating system, 4GB memory, Intel(R) Core duo CPU @ 2.60 GHz.

Table 5.1 Frequent Itemsets of Supermarket dataset with different Support Value

Algorithm	0.1	0.2	0.3	0.4
Apriori	45	13	5	1
KApriori	71	15	8	2
FP –Top K Rule	85	17	11	3

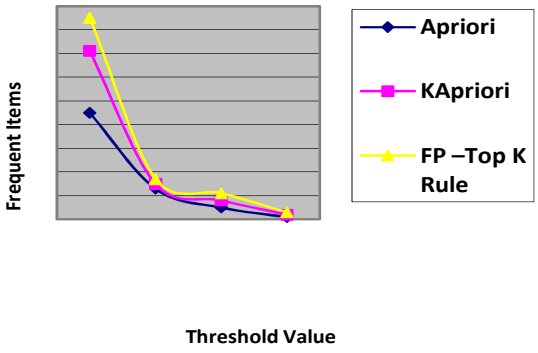

Figure 5.1 Comparison of different algorithm vs Threshold value

Table 5.2 Memory of Supermarket dataset with different Support Value

Algorithm	0.1	0.2	0.3	0.4
Apriori	11.2	15.3	21.3	24.32
KApriori	8.43	10.4	15.32	19.43
FP –Top K Rule	6.348	8.2	5.27	6.68

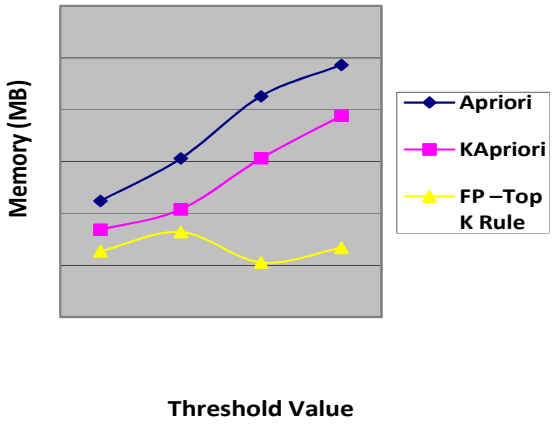

Figure 5.2 Comparison of different algorithm memory used svs Threshold value

Table 5.3 Times of Supermarket dataset with different Support Value.

Algorithm	0.1	0.2	0.3	0.4
Apriori	23	19.2	16.9	14.2
K- Apriori	14	8.2	5.9	3.1
FP –Top K Rule	5	3.8	3.4	2

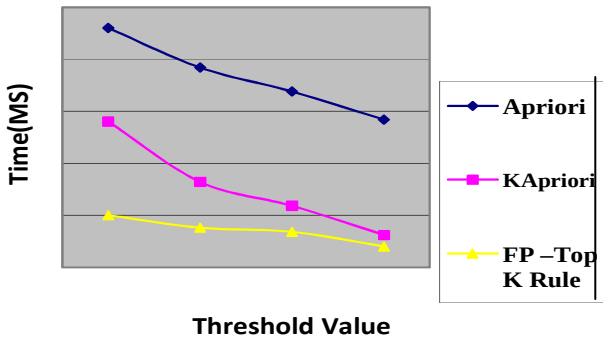

Figure 5.3 Comparison of different algorithm time takes svs Threshold value.

6. CONCLUSION

Market basket analyses is most widely used in the TS store is to manage the placement of goods in their store layout. Related products are placed together in such a manner that customers can logically find items he/she might buy which increases the customer satisfaction and hence the profit. Customers are segmented and association rules are separately generated to satisfy their specific needs in a cost effective manner using some special promotions for the common groups. The proposed FP growth and TOP k Rule based association rule mining is to discover the *top-k* rules having the highest support, where *k* is set by the user. To generate rules, Top K Rules relies on a novel approach called rule expansions and also includes several optimizations that improve its performance. Experimental results show that Top K Rules has excellent performance and scalability, and that it is an advantageous alternative to classical association rule mining algorithms when the user wants to control the number of association rules generated.

7. FUTURE ENHANCEMENT

The mining of association rules in binary data. However, a transaction database can be quantitative type. In the experiments mentioned above, we transfer the quantitative database to the binary type by using discretization and it may cause losing much information. In this case, we should utilize fuzzy set to overcome this problem. A fuzzy set is represented by a membership function which assigns to each value of the attribute a value between 0 and 1 to indicate how much this value belongs to the fuzzy set.

REFERENCES

[1]. Agrawal R, Srikant R, Fast algorithms for mining association rules. In: *Proceedings of the 20th VLDB conference*,1994, pp 487–499.

[2].R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, ashington,D.C., May 1993.

[3]. Pavel Berkhin," Survey of Clustering Data Mining Techniques" Accrue Software, Inc.

[4]. Aastha Joshi, Rajneet Kaur, "A Review: Comparitive Study of Various Clustering Techniques in Data Mining(IJARCSSE) Volume 3, Issue 3, March 2013, pp. 55 – 57.

[5]. Madhuri V.Joseph, Lipsa Sadath, Vanaja Rajan, "Data mining: A comparative study on Various Techniques and Methods" (2013) International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 3, Issue 2, February 2013, pp.106-113.

[6].Weng,S.-S., Liu, J.-L.: Feature-based recommendations for one-to-one marketing, *Expert Systems with Applications, Vol. 26*, 2004, pp. 493-508.

[7]. Chen, Y.-L., Tang, K., Shen, R.-J., Hu, Y.-H.: Market basket analysis in a multiple store environment, *Decision Support Systems*, 2004.

[8]. Berry, M.J.A., Linoff, G.S.: *Data Mining Techniques: for Marketing, Sales and Customer Relationship Management (second edition)*, Hungry Minds Inc., 2004.

[9]. J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: *Proc. Conf. on the Management of Data SIGMOD'00*, ACM Press, New York, NY, USA 2000.

[10]. Loraine Charlet Annie M.C. and Ashok Kumar D, "Frequent Item set mining for Market Basket Data using K-Apriori algorithm" , *International Journal of Computational Intelligence and Informatics*, Volume 1, No. 1, 2011, pp.14-18.

[11]. Verykios V. S., Elmagarmid A., Bertino E., Saygin Y., Dasseni E., "Association Rule Hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, April 2004.

[12]. Agrawal R, Srikant R., Mining Sequential Patterns, Proc. of the Int'l Conference on Data Engineering, Taipei, Taiwan, 1995.

[13]. Chakravarthy S., Krishnaprasad V., Tamizuddin Z., and Badani R. H., ECA Rule Integration into an OODBMS: Architecture and Implementaion, 11th International Conference on Data Engineering, Taipei, Taiwan, March 1995.

[14]. Dudgikar M., A Layered Approach for Mining Association Rules over Relational DBMS, Master's thesis, University of Florida, Gainesville, 2000.

[15]. J. Han, J. Pei, Y. Yin. "Mining Frequent Patterns without Candidate Generation". Proc. of ACM-SIGMOD, 2000.