Implementing Advanced Association Rule Mining for Top K-Item Set through Shopping Package

K.Dharani[1] Dr.Antony Selvadoss Thanamani[2]
[1]Mphil Research Scholar [2]Associate Professor & Head of Dept.
[1,2]Department of Computer Science & Engineering
[1,2]NGM College Pollcahi

Abstract— *To mine the frequent item set and maximum threshold signature of the shopping package software item set. The frequent item set deals with the whole database of the shopping application. It contains various transactions like sales data, purchase data, customer data, item data and etc. TOP K – Item set has been implemented (TKI) which gives more accuracy and performance than TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase), which are implemented in the exiting methods for mining such item sets without the consideration of entire database. TOP K-Item set Algorithm is used to analyze the items which are sold frequently that are reported by the client. The decisions can be made on the result of analysis, so that the item can be identified. Normally an input given by the client to the sale the item in project is taken as it is and service is provided without analyzing the input. This leads to wastage of time in decision making and also the delay in finding the frequently sold items. If the frequently sold items in project are analyzed, then it is easy to find out the relationship or association among the items. So, that The reason for sold item and how frequently sold item on each other can be found in a project. Here advanced association rules based on pattern mining is used in the system to find associations or relationships among the frequently sold items. So that both data mining and networking concepts are implemented. In this article we propose a new method as K-mine. It is used to find out the deep historical value from the database. The system provides the information about the associations among the frequently sold item in an item level. The system has used two data mining techniques namely association rules and its algorithms to finding the frequently sold items. Some of the results are displayed in a graphical manner also. The results of these techniques would be helpful in decision making, so that the client needs can be satisfied in a faster way.*

Keywords— TKU (mining Top-K Utility item sets), TKO (mining Top-K utility item sets in One phase), Decision Making, K-mine, Frequent Item mining, Association rules, Pattern mining

I. INTRODUCTION

Generally any organization receives a report sold the items in projects from its user and provides services to its user based on their input without analyzing them. This leads to wastage of time in Decision making and also delays in finding the frequently sold items. It is because they do not know the how much items sold. The results of the above queries are unknown and they can be answered only by using data mining techniques. Hence, a system is needed with data mining techniques to analyze the sold items in package in order to find the relationship among the frequently sold items. The dependencies among modules and programs can also be found using data mining techniques. The frequently sold items can also be done quickly by using information generated by data mining techniques. The hidden or unknown information is useful in decision making and the decisions quickly.

II. RELATED WORKS

The subgroup discovery algorithm CN2-SD pattern, based on a separate and conquer strategy, has to face the scaling problem which appears in the evaluation of large size data sets. To avoid this problem, in this article we propose the use of instance selection algorithms for scaling down the data sets before the subgroup discovery task. The results show that CN2-SD can be executed on large data set sizes pre-processed, maintaining and improving the quality of the subgroups discovered [1]. We 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. We also present results of applying this algorithm to sales data obtained from a large retailing company, which shows the effectiveness of the algorithm [2]. If the cost parameters are not known at training time, Receiver Operating Characteristic (ROC) analysis can be applied (Provost & Fawcett 1997; Swets, Dawes & Monahan 2000). ROC analysis provides tools to distinguish classifiers that are optimal under some class and cost distributions from classifiers that are always sub-optimal, and to select the optimal classifier once the cost parameters are known [3]. ROC analysis for two classes is based on plotting the true-positive rate (TPR) on the y-axis and the false-positive rate (FPR) on the x-axis. This gives a point for each classifier [4].

III. EXISTING SYSTEM

In existing system general association rules are used is a prominent and a well explored method for determining relations among variables in large databases. But the there is no data handling techniques and special algorithm. Let us take a look at the formal definition of the problem of association rules given by Rakish Agrawal, the President and Founder of the Data Insights Laboratories.

Let $I = \{i_1, i_2, i_3, ..., i_n\}$ be a set of n attributes called items and $D = \{t_1, t_2, ..., t_n\}$ be the set of transactions. It is called database. Every transaction, t_i in D has a unique transaction ID, and it consists of a subset of item sets in I.

A rule can be defined as an implication, $X \longrightarrow Y$ where X and Y are subsets of $I(X, Y \subseteq I)$, and they have no element in common, i.e., $X \cap Y$. X and Y are the antecedent and the consequent of the rule, respectively. Let's take an easy example from the supermarket sphere. The example that we

are considering is quite small and in practical situations, datasets contain millions or billions of transactions. (E.g) The set of item sets, I ={Onion, Burger, Potato, Milk, Beer} and a database consisting of five transactions. Each transaction is a tuple of 0's and 1's where 0 represents the absence of an item and 1 the presence. Here only low data consideration has been done. This cannot be applicable for a huge data set and huge data transactions.

IV. PROBLEM DEFINITION

The main objective of this article is to mine the frequent item set and maximum threshold signature of shopping package. High average utility itemset algorithm is used to analyze the items, which are sold frequently that are reported by the clients. The decision can be made on the result of analysis so that the item can be identified. If the frequently sold items in project are analyzed then it is easy to find out the relationships.HUI is used as the input dataset for the entire thesis, it contains huge collection of utility items. Some association rule is implemented for frequent item mining and finally top-K itemset has been implemented. K-mine is proposed a new method. It is used to find out the deep historical data from data base. The K-Mine using pattern mining can make a major impact in the data mining concept. Also it may usefully for sales business like retail, whole sale and online business.

V. IMPLEMENTATION

K – Mine through pattern mining Algorithm Pseudo code
L1=Least frequent items,
Ck=candidate key;

1) Step 1: procedure kmine (T, min Support) {//T is the database and min Support is the minimum support
2) Step 2: L1= {frequent items};
3) Step 3:for (k= 2; Lk-1 !=∅; k++) {
4) Step4: Ck= candidates generated from Lk-1
   //that is Cartesian product Lk-1 x Lk-1 and
   Eliminating any k-1 size itemset that is not
   //frequent
5) Step 5: for each transaction t in database do
   {
6) Step 6: #increment the count of all candidates in Ck that are contained in t
7) Step 7: Lk = candidates in Ck with min Support
8) Sep 8: Generate pattern
   }
   //end for each}
   //end for return ∪ ;
   }

A. Implementation Process

As is common in association rule mining, given a set of item sets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. K - Mine uses a "bottom up" approach, wing pattern mining where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Sample usage of K - Mine algorithm in pattern mining, a large supermarket tracks sales data by Stock-keeping unit (SKU) for each item, and thus is able to know what items are typically purchased together. K - Mine is a moderately efficient way to build a list of frequent purchased item pairs from this data. Let the database of transactions consist of the sets {1,2,3,4}, {1,2,3,4,5}, {2,3,4}, {2,3,5}, {1,2,4}, {1,3,4}, {2,3,4,5}, {1,3,4,5}, {3,4,5}, {1,2,3,5}. Each number corresponds to a product such as "butter" or "water". The first step of K - Mine is to count up the frequencies, called the supports, of each member item separately: Item Support 1 2 3 4 5 We can define a minimum support level to qualify as "frequent," which depends on the context. For this case, let min support = 4. Therefore, all are frequent. The next step is to generate a list of all 2-pairs of the frequent items. Had any of the above items not been frequent, they wouldn't have been included as a possible member of possible 2-item pairs. In this way, K - Mine prunes the tree of all possible sets. In next step we again select only these items (now 2-pairs are items) which are frequent. Item Support {1,2} 4 {1,3} 5 {1,4} 5 {1,5} 3 {2,3} 1 {2,4} 5 {2,5} 4 {3,4} 2 {3,5} 5 {4,5} 4 We generate the list of all 3-triples of the frequent items (by connecting frequent pair with frequent single item). Item Support {1,3,4} 4 {2,3,4} 4 {2,3,5} 4 {3,4,5} 4 The algorithm will end here because the pair {2,3,4,5} generated at the next step does not have the desired support. We will now apply the same algorithm on the same set of data considering that the min support is 5. We get the following results: Step 1: Item Support 1 2 3 4 5

Step 2: Item Support {1,2} 4 {1,3} 5 {1,4} 5 {1,5} 3 {2,3} 1 {2,4} 5 {2,5} 4 {3,4} 2 {3,5} 5 {4,5} 4 The algorithm ends here because none of the 3-triples generated at Step 3 have de desired support.

The conviction of a rule can be defined as:

$$conv(X \longrightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \longrightarrow Y)}$$

For the rule {Item1, Item2}=>{Item2}

If the value of lift is greater than 1, it means that the itemset Y is likely to be bought with itemset X, while a value less than 1 implies that itemset Y is unlikely to be bought if the itemset X is bought.

The lift of a rule is defined as:

$$lift(X \longrightarrow Y) = \frac{supp(X \cup Y)}{supp(X) * supp(Y)}$$

This signifies the likelihood of the itemset Y being purchased when item X is purchased while taking into account the popularity of Y.

VI. RESULT AND ANALYSIS

A. Interpreting and Comparing Results

When comparing the results of applying association rules with pattern mining to those from simple frequency or cross-tabulation tables, you may notice that in some cases very high-frequency codes or text values (items) are not part of any association rule. This can sometimes be perplexing.

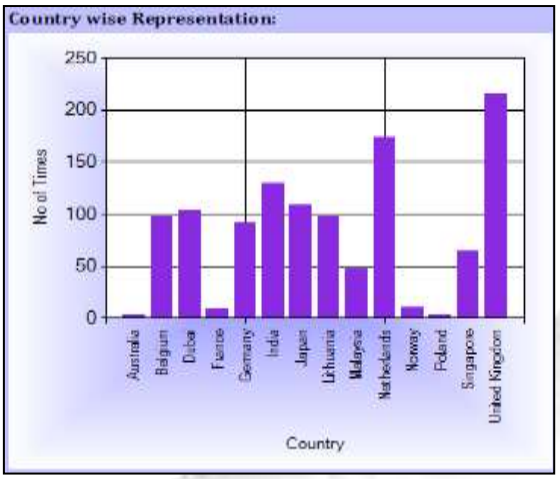

Fig. 1: Country Wise usage

Fig 1: illustrating the most product usage countries from the available dataset. A union association rule has been used to fetch data following the association rules.

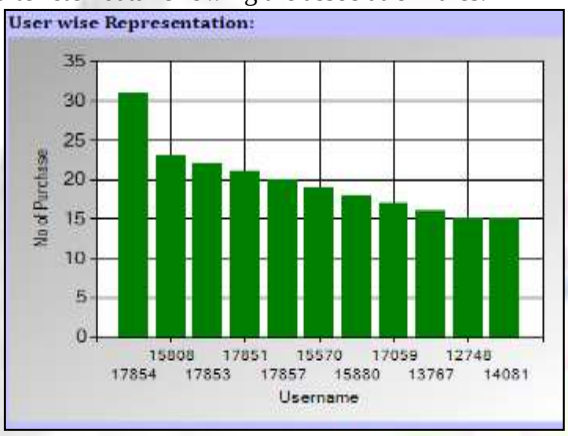

Fig. 2: High Utility Items

Fig 2 shows the high utility item set. All the products are mentioned in their product ids. From the product id the product can be shown in the final result. For finding association rules, we need to find all rules having support greater than the threshold support and confidence greater than the threshold confidence. From our research perspective, the problem of high-utility item set mining is more challenging. In frequent itemset mining, there is a well-known property of the frequency (support) of itemsets that states that given an itemset, all its supersets must have a support that is lower or equal. This is often called the "Association property" or "anti-monotonicity" property and is very powerful to prune the search space because if an itemset is infrequent then we know that all its supersets are also infrequent and may be pruned. In high-utility itemset mining there is no such property. Thus given an itemset, the utility of its supersets may be higher, lower or the same.

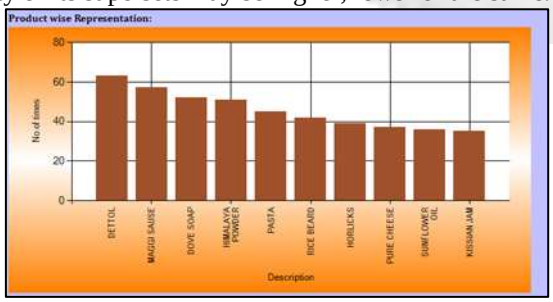

Fig. 3: Product wise high utility item

B. Screenshots

Fig. 4: Dataset View

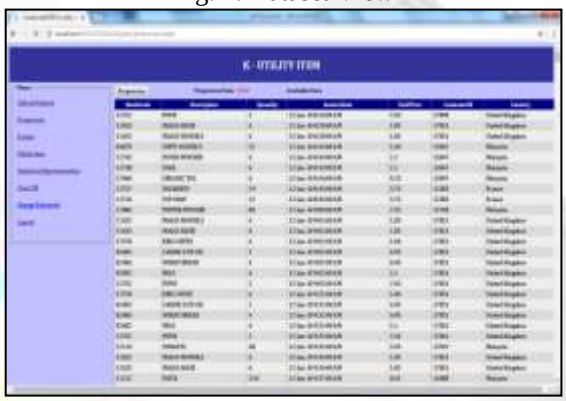

Fig. 5: Data after preprocessing

Fig. 6: K – Mine process

Fig. 7: K-Mine product similarity view

Fig. 8: K-Mine Customer based product view

VII. CONCLUSION

The output has been verified as per the committed abstract. K mine has been implemented successfully as per (1) Constructing the Decision-Tree, (2) Populating Potential K – mine for High Utility Item Sets (PKHUIs) using pattern mining (3) Identifying top-k and UT s from the set of PKHUIs. A huge data set of data has been executed successfully. The output has been verified with three different data types in various conditions. Output has been verified, according to the given input. The obtained result is prompt according to the given commitments. The K-Mine using pattern mining can make a major impact in the data mining concept. Also it may usefully for sales business like retail, whole sale and online business. Clustering of the data has been implemented successfully in the pre processing stage itself. More over many categorization processes has been done for generating various outputs from single dataset. For graphical representation, various charts have been developed. User can clear the database for fresh use of these methods. So that this article has been implemented successfully and result has been verified.

VIII. FUTURE WORK

Mining high utility item sets are becoming more significant. In this article, the Improved TKO evaluated with TKU with association rules is discussed. These algorithms are experimented on synthetic datasets and real time datasets for different support threshold. From the experimental observation, the enhancement is that, IUPG – Improved UP growth algorithm performs well than UPG algorithm for different support values. Also the IUPG algorithm scales well as the size of the transaction database increases. The future work would focus on the different issues to improve in terms of execution and memory space cost.

REFERENCE

[1] The CN2 induction algorithm", Machine Learning.Clark, P. and Niblett, T., "The CN2 induction algorithm", Machine Learning, Vol. 3(4), pp. 261–283, 1999.

[2] Mining Association with pattern Rules between Sets of Items in Large Databases. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large database.In Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'93), pp:207-216, Washington, DC, May 1993.

[3] Learning Decision Trees Using the Area under the ROC Curve Cesar Ferri-Ram´ ırez, Peter A. Flach, and Jose Hernandez-Orallo. Learning decision trees using the area under the roc curve. In Proceedings of the Nineteenth International Conference on Machine Learning, pages 139–146, Morgan Kaufmann, 2002.

[4] Discovering business intelligence from online product reviews: A rule-induction framework. W. Chung and H. Chen. Web-Based Business Intelligence Systems: A Review and Case Studies. In G. Adomavicius and A. Gupta, editors, Business Computing, volume 3, chapter 14, pages 373–396. Emerald Group Publishing, 2009.

[5] Logical Design of Data Warehouses from XML. M. Banek, Z. Skocir, and B. Vrdoljak.Logical Design of Data Warehouses from XML . In ConTEL '05: Proceedings of the 8th international conference on Telecommunications, volume 1, pages 289–295, 2005.

[6] A multisession-based multidimensional model. M. Body, M. Miquel, Y. B´edard, and A. Tchounikine. A multidimensional and multi version structure for OLAP applications. In DOLAP '02: Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP, pages 1–6, New York, NY, USA,2002. ACM.

[7] Transaction Management for a Main-Memory Database. P. Burte, B. Aleman-meza, D. B. Weatherly, R. Wu, S. Professor, and J. A. Miller. Transaction Management for a Main-Memory Database. The 38th Annual South eastern ACM Conference, Athens, Georgia, pages 263–268, January 2001.

[8] A rule-induction framework. W. Chung and H. Chen. Web-Based Business Intelligence Systems: A Review and Case Studies. In G. Adomavicius and A. Gupta, editors, Business Computing, volume 3, chapter 14, pages 373–396. Emerald Group Publishing, 2009.

[9] Crowd sourcing Predictors of Behavioural Outcomes using pattern mining. Josh C. Bongard, Member, IEEE, Paul D. H. Hines, Member, IEEE, Dylan Conger, Peter Hurd, and Zhenyu Lu. ieee transactions on knowledge and data engineering year 2013.