International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

A Combinatorial Approach for High Utility Item Set Mining using FRUP and Direct Discovery Approach without Candidate Generation

Mansi Jaiswal[1], Vijay Prakash[2]

[1]PG Student, CSE Department, Shri Vaishnav Vidhyapeeth Vishwavidyalaya, Indore, India
[2]Assistant Professor, CSE Department, Shri Vaishnav Vidhyapeeth Vishwavidyalaya, Indore, India

Abstract: ***The main purpose of data mining and analytics is to find novel, potentially useful patterns that can be utilized in real-world applications to derive beneficial knowledge. For identifying and evaluating the usefulness of different kinds of patterns, many techniques/constraints have been proposed, such as support, confidence, sequence order, and utility parameters (e.g., weight, price, profit, quantity, etc.). In recent years, there has been an increasing demand for utility-oriented pattern mining (UPM). UPM is a vital task, with numerous high-impact applications, includes cross-marketing, e-commerce, finance, medical, and biomedical applications. In this research work we have undertook two different approach as proposed in [1] and [2]. One approach uses RUP/FRUP growth algorithm while the other method uses direct discovery algorithm which does not uses candidate generation. The FRUP/FRUP approach is more extensive in a sense that not only it is helpful in determining frequent itemset but it also helps in finding the utility of the item set in a more cohesive manner. We used Matlab programming environment to combine the two approaches. The experimental results show that RUP/FRUP when combined with direct discovery approach gives better results.***

Keywords: RUP/FRUP-GROWTH algorithm, HUI, data mining, apriori, big data

1. Introduction

The rapid growth of data generated and stored has led us to the new era of Big Data [3, 4, 14, 18, 19]. Nowadays, we are surrounded by different types of big data, such as enterprise data, sensor data, machine-generated data and social data. Extracting valuable information and insightful knowledge from big data has become an urgent need in many disciplines. In view of this, big data analytics [3, 4, 14, 18, 19] has emerged as a novel topic in recent years. This technology is particularly important to enterprises and business organizations because it can help them to increase revenues, retain customers and make more intelligent decisions. Due to its high impact in many areas, more and more systems and analytical tools have been developed for big data analytics, such as Apache Mahout [14], MOA [3], SAMOA [19] and Vowpal Wabbit [20]. However, to the best of our knowledge, no existing studies have incorporated the concept of utility mining [2, 6, 7, 8, 11, 12, 13] into big data analytics.

Utility mining is an important research topic in data mining. The main objective of utility mining is to extract valuable and useful information from data by considering profit, quantity, cost or other user preferences. High utility itemset (HUI) mining is one of the most important tasks in utility mining, which can be used to discover sets of items carrying high utilities (e.g., high profits). This technology has been applied to many applications such as market analysis, web mining, mobile computing and even bioinformatics. Due to its wide range of applications, many studies [2, 6, 7, 8, 11, 12, 13] have been proposed for mining HUIs in databases. However, most of them assume that data are stored in centralized databases with a single machine performing the mining tasks. However, in big data environments, data may be originated from different sources and highly distributed. A large volume of data also makes it difficult to be moved to a centralized database. Thus, existing algorithms are not suitable for the applications of big data. Although mining HUIs from big data is very desirable for many applications, it is a challenging task due to the following problems posed: First, due to a large amount of transactions and varied items in big data, it would face the large search space and the combination explosion problem. This leads the mining task to suffer from very expensive computational costs in practical. Second, pruning the search space in HUI mining is more difficult than that in frequent pattern mining because the downward closure property [1] does not hold for the utility of itemsets. Therefore, many search space pruning techniques developed for frequent pattern mining cannot be directly transferred to the scenario of HUI mining. Third, a large amount of data cannot be efficiently processed by a single machine. A well-designed algorithm incorporated with parallel programming architecture is needed. However, implementing a parallel algorithm involves several problematic issues, such as search space decomposition, avoidance of duplicating works, minimization of synchronization and communication overheads, fault tolerance and scalability problems.

2. Literature Review

In 2017, Jue Jin and Shui Wang in their research work titled "RUP/FRUP-GROWTH: AN EFFICIENT ALGORITHM FOR MINING HIGH UTILITY ITEMSETS" proposed an improvement of the UP-Growth algorithm called RUP-Growth, and develops a new algorithm called FRUP-Growth to take into consideration both the minimum support number and the minimum utility value to mine frequent & high utility itemsets. The experimental results show that their proposed strategies are more efficient and effective; especially with real-life marketing database, the advantage is more obvious.

In 2015, Ying Chun Lin and others in their research work titled "Mining High Utility Itemsets in Big Data" propose a

new framework for mining high utility itemsets in big data. A novel algorithm named PHUI-Growth (Parallel mining High Utility Itemsets by pattern-Growth) is proposed for parallel mining HUIs on Hadoop platform, which inherits several nice properties of Hadoop, including easy deployment, fault recovery, low communication overheads and high scalability. Moreover, it adopts the MapReduce architecture to partition the whole mining tasks into smaller independent subtasks and uses Hadoop distributed file system to manage distributed data so that it allows to parallel discover HUIs from distributed data across multiple commodity computers in a reliable, fault tolerance manner.

In 2012, Adinarayanareddy B and others in their research work titled "An Improved UP-Growth High Utility Itemset Mining" adopted UP-Tree (Utility Pattern Tree), which scans database only twice to obtain candidate items and manage them in an efficient data structured way. Applying UP-Tree to the UP-Growth takes more execution time for Phase II. Hence this paper presents modified algorithm aiming to reduce the execution time by effectively identifying high utility itemsets.

In 2014, Junqiang Liu and others in their research work titled "Direct Discovery of High Utility Itemsets without Candidate Generation" proposed a high utility itemset growth approach that works in a single phase without generating candidates. Their basic approach is to enumerate itemsets by prefix extensions, to prune search space by utility upper bounding, and to maintain original utility information in the mining process by a novel data structure. Such a data structure enables them to compute a tight bound for powerful pruning and to directly identify high utility itemsets in an efficient and scalable way. We further enhance the efficiency significantly by introducing recursive irrelevant item filtering with sparse data, and a lookahead strategy with dense data. Extensive experiments on sparse and dense, synthetic and real data suggest that their algorithm outperforms the state-of-the-art algorithms over one order of magnitude.

In 2007, Alva Erwin and others in their research work titled "A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets" proposed a new algorithm called CTU-PRO that mines high utility itemsets by bottom up traversal of a compressed utility pattern (CUP) tree. They have tested the algorithm on several sparse and dense data sets, comparing it with the recent algorithms for High Utility Itemset Mining and the results show that their algorithm works more efficiently.

3. RUP/FRUP Growth Algorithm

Mining frequent itemsets from a transaction database is an important task in the field of data mining. Its goal is to identify the itemsets with their appearing frequencies above a certain threshold. Rakesh Agrawal developed the first frequent itemset mining algorithm, named Apriori [1], for mining association rules from sales data in 1994. Since then, new algorithms are proposed constantly, such as FP-Growth [2], COFI [3], COFI2 [4], Pincer-search [5], MAFIA [6], CLOSET [7], CHARM [8], and so on. The existing algorithms of mining frequent itemsets only consider each item in a transaction database as a 0/1 value. Moreover, items having high/low selling frequencies may have low/high profits, respectively. The information of profit and quantity of each item in a transaction itemset is of great importance for market analysis. In view of this, high utility itemsets mining emerges as an important topic in data mining. Yao et al. proposed the high utility itemsets mining model [9] in 2004. They defined two types of utilities for items: internal utility and external utility. The internal utility of an item in a transaction is defined according to the information stored in the transaction itself, such as quantity of the merchandise. The external utility of an item is based on information not available in the transaction database, for example, the profit value of the merchandise sold in the marketplace. The utility of an item is defined as its internal utility multiplied by its external utility. The utility of an itemset is defined as the sum of its all items' utilities. An itemset X is a high utility itemset if its utility is not less than a user-specified minimum threshold. High utility itemsets mining is widely used in applications such as sales data analysis, etc., to find the suitable combinations of items that are more profitable; and many algorithms have been proposed recently, such as those described in [9-17]. The existing algorithms mentioned above apply overestimated method; they firstly find candidates for high utility itemsets, and then identify the high utility itemsets from the candidates by one additional database scan. The performance bottleneck of these algorithms is the generating & processing of the candidate itemsets; and with the increasing of the number of long transaction itemsets and the decreasing of the minimum utility threshold, the situation may become worse. Additionally, as stated above, the existing algorithms mine either frequent itemsets or high utility itemsets. However, in real world, an itemset may be a high utility itemset, but not a frequent itemset, so we can not get an association rule from this itemset, and can not get useful knowledge for the future business arrangement. Thus, in this paper, we firstly propose new strategies to reduce the number of candidates; then propose a new idea to mine frequent & high utility itemsets, which satisfy both the minimum support threshold as well as the minimum utility threshold, from transaction datasets.

4. FP Growth Algorithm

d 2 HUP searches the prefix extension tree in a depth-first manner. When visiting a node N, it first computes utilities and upper bounds for the children of N by building a pseudo CAUL, make a materialized copy of CAUL if a space-time tradeoff is beneficial, outputs each child whose utility is no less than minUtil as a high utility itemset, depth-first searches each child whose upper bound is no less than minUtil, and then purges the subtree rooted at N and continues with the next sibling of N. The pseudo code of d 2 HUP is shown in Algorithm 1. First, d 2 HUP creates the root of prefix extension tree (line 1), builds TS caul ({}) by scanning the database D and the external utility table XUT to compute s({i}), u({i}), uB item (i,{}), and uB fpe ({i}) for each item i (line 2), and starts the depth-first search from the root node (line 3). Second, d 2 HUP makes the set W of relevant items by Corollary 2 for the node N currently being visited (line 4). If the closure property holds, d 2 HUP outputs every prefix extension of pat(N) with relevant items

as a high utility itemset (line 5). If the singleton property holds, d 2 HUP outputs the union of all the relevant items and pat(N) as a high utility itemset (lines 6 - 7). Third, for each relevant item i ∈ W, d 2 HUP outputs {i}∪pat(N) as a high utility itemset if u({i}∪pat(N)) ≥ minUtil, and creates a child node C of N with item(C) ← i and pat(C) ← {i} ∪ pat(N), if uB fpe ({i} ∪ pat(N)) ≥ minUtil (lines 8 - 11). Fourth, d 2 HUP continues by purging the branch that has been searched and making N to represent the next node in the depth-first order (lines 12 - 15), and by computing s({j}∪pat(N)), u({j}∪pat(N)), uB item (j,pat(N)), and uB fpe ({j}∪pat(N)) for each item j in TS caul ({pat(N)}) by projection from TS caul (pat(P)) (line 16). For the running example, d 2 HUP only enumerates the nodes 0, 4, 6, 8, 10, 16, 24, 32, and 64 in Figure 1 to find all the high utility itemsets.

5. Results

We have made a graphical user interface in matlab environment which helps us in comparing and evaluating the two approaches in a more holistic manner. The screenshot of GUI is as shown below:

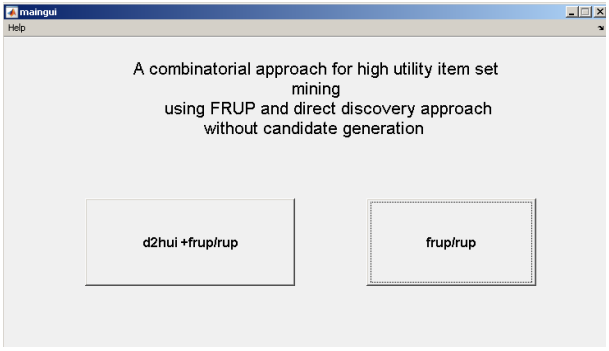

Figure 5.1: Primary GUI window

Tools in the Graphic User Interface

Consider a data set with a number of observations (rows) measured over a set of features or variables (columns). The MEDA Graphic User Interface includes the following PCA and PLS tools as to analyze data sets:

1) Score plots [1]: this tool allows the user to visualize the distribution of the observations in the reduced set of LVs. This results in a bi-dimensional scatter plot for a pair of LVs selected by the user.
2) Loading plots [1]: this tool allows the user to visualize the distribution of the features or variables, in order to explore the relationship among variables in the data set. Again, this operation will give the user a bi-dimensional scatter plot for a pair of LVs selected by the user.
3) MEDA (Missing Data Methods for EDA) [4]: this tool shows the relationship among the features in the data within a simple red-blue grid graphic. The features to display are selected by the user.
4) oMEDA (observation-based MEDA) [5]: is a variant of MEDA to connect observations and features. This is a very powerful tool in order to learn how and what features affect each observation or group of observations. The group of observations to display is selected by the user.
5) Model and Residue: these buttons allow the user to represent the model or the residual variance corresponding to the selected LVs or PCs.

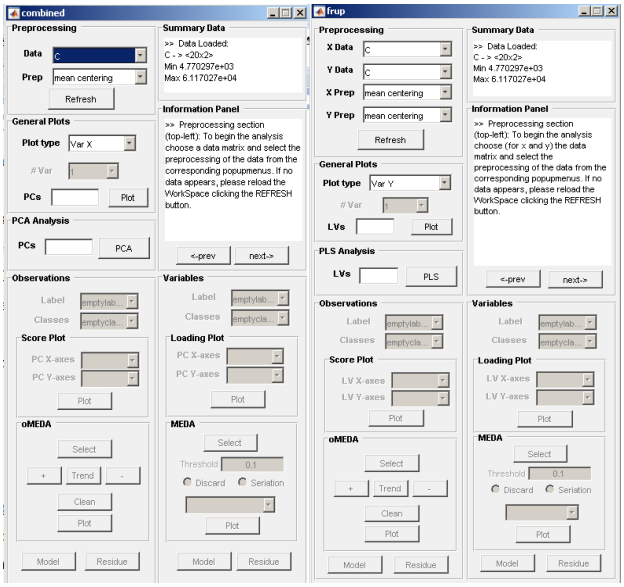

Figure 5.2: Secondary GUI windows for d2hui and frup/rup approach

The data of the example in this tutorial is available at the folder named *Examples*, file *MANET.mat*. Load the data matrices in the *Workspace* in Matlab® using command *load*. The following matrices will be loaded, see Table 2. Push the *Refresh* button in the GUI in order to load the new data matrices. Now select the data from the *X Data* and *Y Data* popup menus. In this case, it is recommended to use auto-scaling as the preprocessing method due to the different nature of the X-block variables in this data set. An important question arises here; how many LVs are enough to run the model? As an aid in this matter, you can use C area and initially select a large number of LVs and use a plot type between the following ones: Var Y, Var Y + Scores, Y-SVI Plot or Y-crossval. In this example we are working with 18 variables (listed in Table 1). Let's consider 10 LVs. Write 10 on the LVs editText of area C, select a plot type and click on the Plot button. The result for a "Var Y + scores" plot type is shown in Fig. 4. As it is shown, with just 3 LVs, 70% of the variance is captured, whereas with 5 LVs, 80% of the variance is captured. We will work with 3 LVs. For that, write 3 on the LVs editText of area E and press the PLS button. After that new areas on the interface are enabled (Areas F to I).

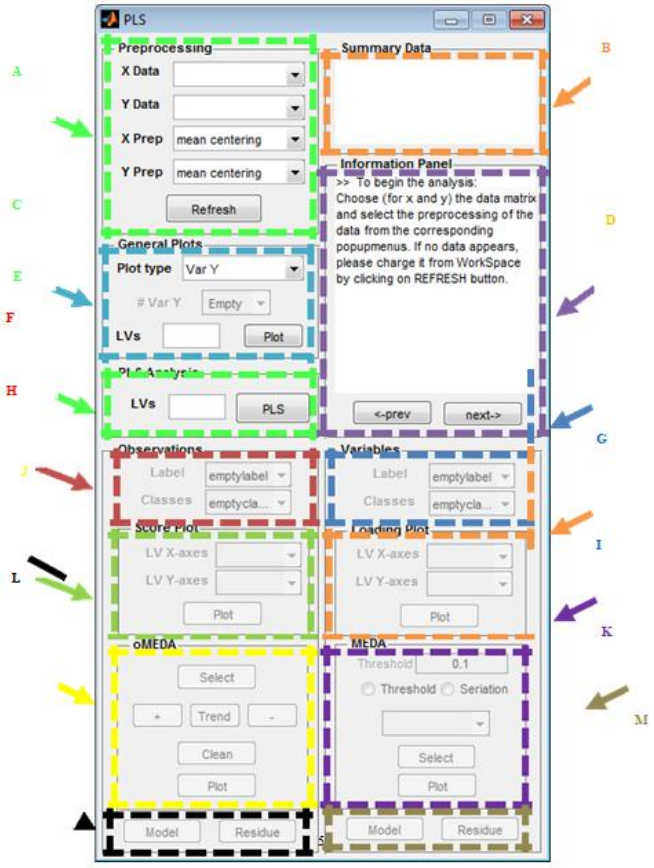

Table 2: Data loaded from the MANET.mat file

Name	Size	Observations
x	70x18	Data matrix. Charge on X Data (area A of the GUI).
y	70x4	Data set of predicted variables. Charge on Y Data (area A of the GUI).
clases	70x1	Array containing as many entries as the number of observations in the data ser. This is an optional field that colors the observations according to the value assigned to each of them, classifying the observations in the data set.
label_v	1x18	Array containing as many entries as the number of variables in the data set. This is an optional field that assigns name to each of the variables.
laby	1x70	Array containing as many entries as the number of observations in the data set.

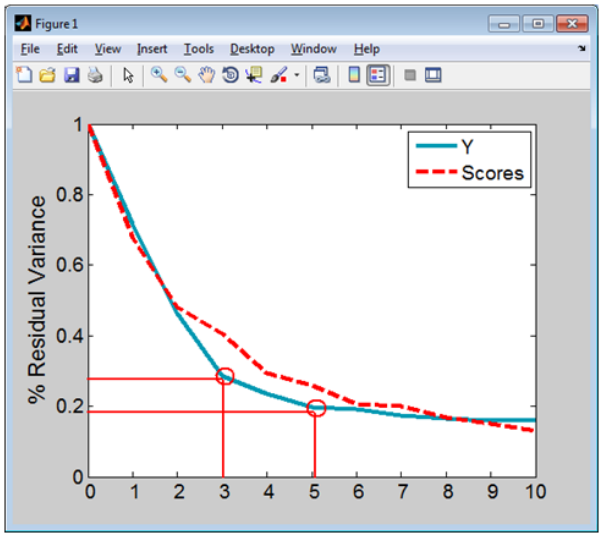

Figure 4: Var Y + scores LVs plot of the MANET data set

Both the *Score Plot* area in the *Observation* menu and the *Loading Plot* area in the *Variables* menu are enabled. It is possible to begin the analysis working with the observations or the variables. In this case we will start with the observations. Select the LVs you want to display on the Score Plot by clicking on the *LV X-axes* and *LV Y-axes* popup menus. Push on the *Plot* button, in area C, to obtain the graphics. In Fig. 5 you can see 2score plots of the data. We can improve the visualization with colors. For this, we define a *Classes* vector in the GUI. This vector is a row vector, containing 70 values with the assignment of each observation to one class; the assignment is a number from 1 to 4 depending on the routing protocol each observation belongs to. This vector is loaded with the rest of the data from the MANET.mat file. Load this vector on the *Classes* popup menu and repeat the steps to obtain the score plots. The result is shown in Fig. 6. The colored plot is easier to interpret. In the second plot, the four classes are distributed in different locations. We are working with that score plot from now on. The plot suggests that we can identify the class of an observation from the PLS-DA model with the design variables considered in Table 1. This means that there are significant differences between the algorithms in the data under analysis. We will learn how to unveil the reason for these differences below.

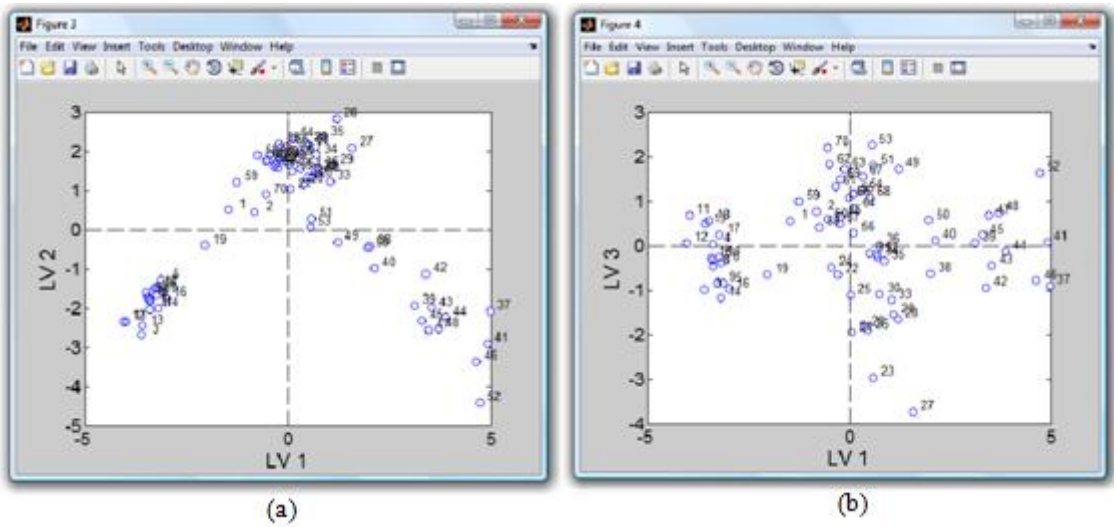

Figure 5: Examples of score plots: (a) LV1 vs LV2 and (b) LV1 vs LV3.

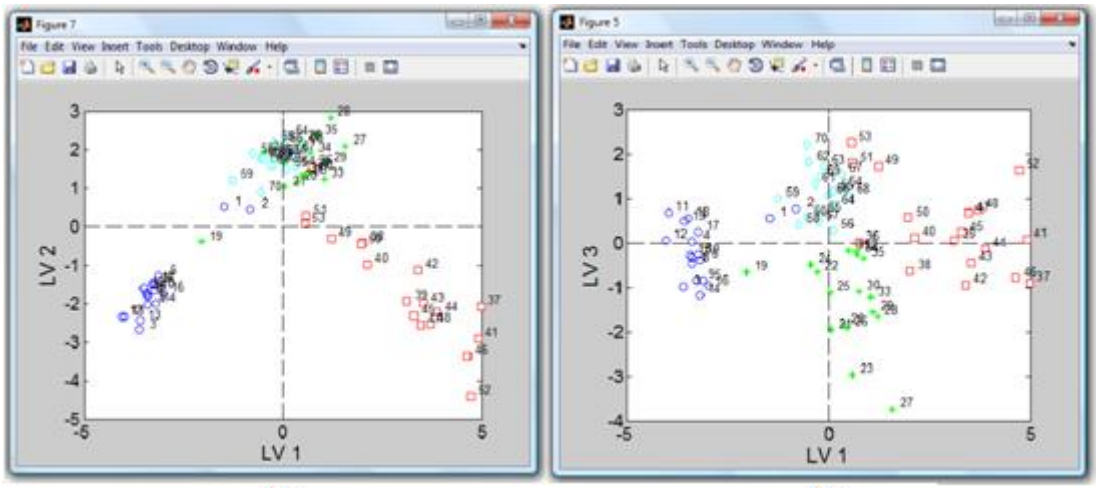

Figure 6: Examples of colored score plots: (a) LV1 vs LV2 and (b) LV1 vs LV3

Maybe you have noticed that after pressing the *Plot* button, the whole interface is enabled, as illustrated in Fig. 7. At this point, it is possible to work with MEDA, oMEDA and the model and residues buttons too.

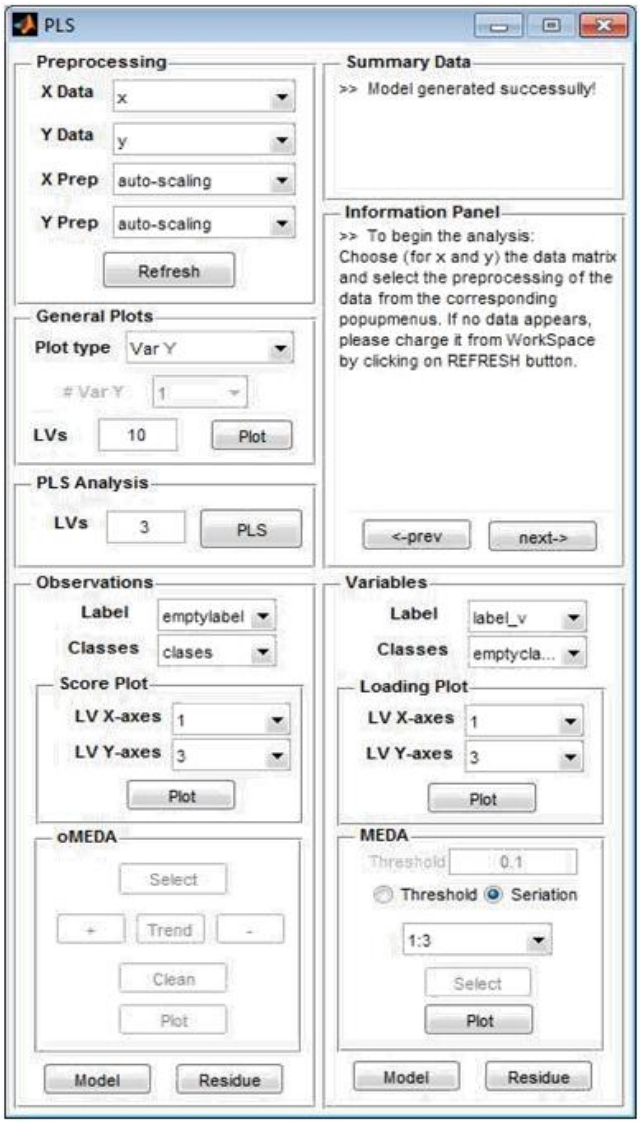

Figure 7: PLS GUI completely enabled

In EDA it is customary to interpret both a score and loading plot of the same subspace together, as done in Fig. 8. Sometimes this is done in a single plot, the so called bi-plot. In the plots corresponding to a given subspace, it is widely accepted that variables and observations displayed in the same zone or the opposite across the axis of coordinates are related. But this is not a fact. It is important to know that loading and score plots tend to be quite confusing when a high number of points are displayed. For example, take a look at variable cY in Fig. 8(b). Considering its locations, it may be thought that this variable may be related to deviations among the observations in the direction of the arrow in the score plot, Fig. 8(a). This is direction the main discriminant direction between two groups of observations: AODV and STARA. We will see in Fig. 10 that there is no influence from variable cY over the difference between the groups. To avoid the misinterpretation of the relationship between observations and variables, the tool oMEDA is very useful.

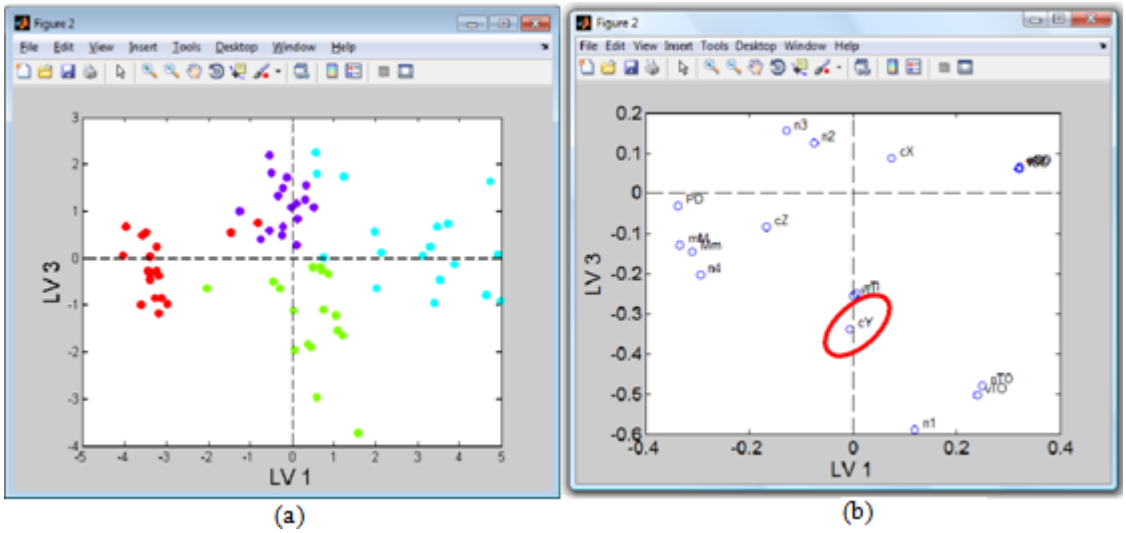

Figure 8: Score plot (a) and loading plot (b) corresponding to LV1 vs LV3

To use oMEDA in the GUI, a number of steps are followed. Firstly, we need to select one score plot to work with, the one in Fig. 8(b). We are going to compare two clusters of scores: the blue one (these observations correspond to STARA) vs the green one (these observations correspond to AODV).

Select (click on) the score plot you are going to work with. Go to the oMEDA submenu in PLS interface and click on the *Select* button. If you put your mouse on the score plot, you will see a cross that allows you to draw an irregular polygon around the observations you want to select. Draw the polygon selecting the AODV scores and click on the + (plus) button. By doing this, we assign the value 1 in oMEDA to the selected scores. Repeat the operations, click on the *Select* button and draw a polygon around the STARA scores, and click over the – *(minus)* button. They are assigned the value -1 in oMEDA. At this point our score plot will look like the one in Fig.

9(a). The unselected scores take the value 0 and are not considered for the oMEDA plot. (*Note:* It is possible to add a trend line to include weights to each of the selected scores but we are not considering that possibility in our example.) Finally click on the *Plot* button. A graphic oMEDA plot like the one shown in Fig. 9(b) is generated. In an oMEDA plot, there is a bar for each of the variables. Thus, the Y axis in Fig. 9(b) represents numbers from 1 to 18, corresponding to the 18 variables in the data set. Alternatively, the name of the variables could be displayed to clarify the plot. To display the name of the variables, first define a vector containing the names, then go to Variables submenu (area G) and load that vector in the *Label* popup menu. By doing this the variables' names will be shown. Re-plot the previous oMEDA plot and you will obtain a graphic like the one in Fig. 10.

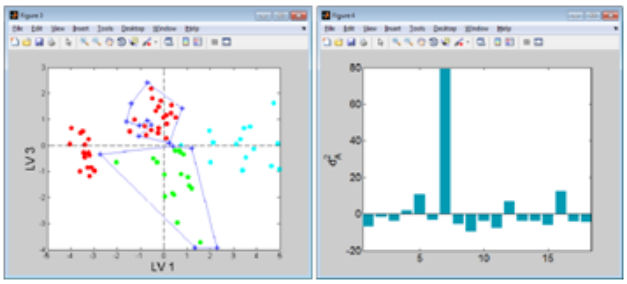

Figure 9: Score plot (a) and oMEDA result (b)

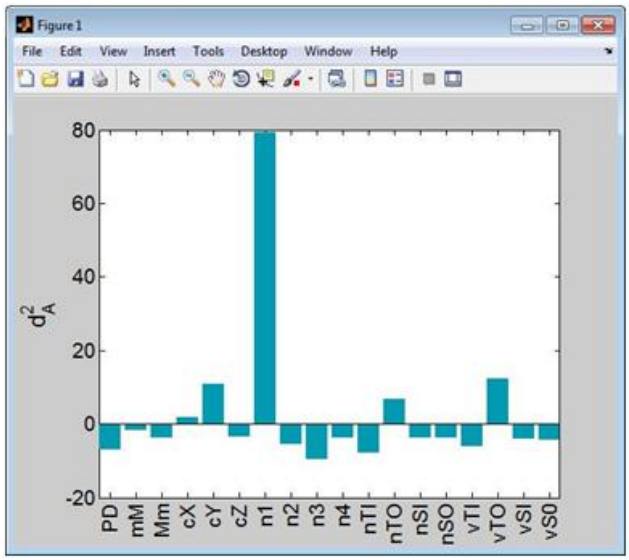

Figure 10: oMEDA plot of Fig. 9(b) with the name of the variables

The oMEDA plot of Fig. 10 displays the differences between AODV and STARA. Remember that the value +1 was assigned to the AODV observations, and the value -1 to the STARA observations. Thus, the bars with a large positive value represent variables that take a larger value in the first group (AODV) in comparison to the second group (STARA), while negative values represent the other way round. Do you remember that we were expecting to solve the confusion caused in figure 9? Look at figure 11, variable cY does not take a high value for AODV. Thus, there is not a clear difference in terms of cY in both groups of observations.

6. Conclusion

Most of research on high utility itemset focuses on static databases (e.g. Transaction database). With the emergence of the new application, the data processed may be in the continuous dynamic data streams. Because the data in streams come with high speed and are continuous and unbounded, mining result should be generated as fast as possible and make only one pass over a data. In this paper, we have proposed combinational algorithm for mining high utility itemsets from transaction databases. A data structure named UP-Tree was proposed for maintaining the information of high utility itemsets. PHUIs can be efficiently generated from UP-Tree with only two database scans. Moreover, we developed several strategies to decrease overestimated utility and enhance the performance of utility mining. Comparison results show that the strategies considerably improved performance by reducing both the search space and the number of candidates. Proposed algorithms, , outperform the state of-the-art algorithms substantially especially when databases contain lots of long transactions or a low minimum utility threshold is used. Proposed system applications in Website click stream analysis, Business promotion in chain hypermarkets, Cross marketing in retail stores, online e-commerce management, Mobile commerce environment planning and even finding important patterns in biomedical applications.

References

[1] Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. In: ACM SIGMOD International Conference on Management of Data (1993).
[2] A. Erwin, R.P. Gopalan, and N.R. Achuthan,"Efficient Mining of High Utility Itemsets from Large Datasets", T. Washio et al. (Eds.):PAKDD2008, LNAI 5012, pp. 554–561, 2008. © Springer-Verlag Berlin Heidelberg 2008.
[3] Yao, H., Hamilton, H.J., Buzz, C. J., "A Foundational Approach to Mining Itemset Utilities from Databases", In: 4th SIAM International Conference on Data Mining, Florida USA (2004).
[4] "A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets", Ying Liu, Wei-Keng Liao, and Alok Choudhary, Northwestern University, Evans.
[5] "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach" In: Seventh International Conference on Computer and Information Technology (2007).
[6] "UP-Growth: An Efficient Algorithm or High Utility Itemset Mining ", Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu. University of Illinois at Chicago, Chicago, Illinois, USA, 2010.
[7] Mengchi Liu Junfeng Qu, "Mining High Utility Itemsets without Candidate Generation", 2012.
[8] "FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning", Philippe Fournier-Viger1, Cheng-Wei Wu 2014.
[9] Smita R. Londhe,, Rupali A. Mahajan,, Bhagyashree J. Bhoyar,"Overview on Methods for Mining High Utility Itemset from Transactional Database", International Journal of Scientific Engineering and Research (IJSER), Volume 1 Issue 4,December2013.
[10] Y. Liu, W. Liao and A. Choudhary, "A fast high utility itemsets mining algorithm," in Proc. of the Utility-Based Data Mining Workshop, 2005.
[11] S.Shankar, T.P.Purusothoman, S. Jayanthi,N.Babu, "A fast agorithm for mining high utility itemsets",in :Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.
[12] Sucahyo, Y.G., Gopalan, R.P., CT-PRO: "A BottomUp Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure", In: IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI), Brighton UK (2004).
[13] G. Salton, Automatic Text Processing, AddisonWesley Publishing, 1989.
[14] J. Pei, J. Han, L.V.S. Lakshmanan, "Pushing convertible constraints in frequent itemset mining", Data Mining and Knowledge Discovery 8 (3) (2004) 227–252.
[15] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets", T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 554–561, 2008. © SpringerVerlag Berlin Heidelberg 2008.
[16] Bin Chen, Peter Hass, Peter Scheuermann, "A New Two-Phase Sampling Based Algorithm for Discovering Association Rules", SIGKDD '02 Edmonton, Albetra, Canada © 2002 ACM 1 58113 567 X/02/2007.
[17] Ming-Yen lin, Tzer-Fu Tu, Sue-Chen Hsueh, "High utility pattern mining using the maximal itemset property and, lexicographic tree structures", Information Science 215(2012) 1-14.
[18] Sudip Bhattacharya, Deepty Dubey, "High utility itemset mining, International Journal of Emerging Technology and advanced Engineering", ISSN 2250-2459, Volume 2, issue 8, August 2012.