Received: February 25, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Proficient Algorithms for Mining HUI in One Phase Using Direct Discovery of High Utility Patterns (D2HUP)

Ghetiya Rutvi Babulal[1] **Gayathri Prakasam[1]***

[1]*School of Computing Science and Engineering, VIT University, Vellore, India*
* Corresponding author's Email: pgayathri@vit.ac.in

Abstract: Proficient Algorithms for Mining High Utility Itemset in One Phase utilizing Direct Discovery of High Utility Patterns refers to the disclosure of object/item sets with high utility. The situation may end up being more terrible when the database contains long high utility itemsets.The information of high utility itemsets is kept up in a tree-based data structure UP-Tree to such a degree, to the point that required itemsets can be created productivity with only two yields of database. The execution of existing algorithm issue all utilize a two-stage, applicant era approach with an exemption which is however neither efficient nor versatile in case of huge databases. This two-stage approach experiences adaptability issue because of the enormous number of matching itemsets. This paper proposes a novel calculation that finds high utility examples in a solitary stage without creating candidates. Our straight information structure empowers us to process a tight technique for pruning and to specifically distinguish high utility examples in an efficient and versatile way, which focuses on the main driver with earlier technique. Test comes to conclusion that the proposed computations, especially Direct Discovery of High Utility Pattern reduce the amount of hopefuls satisfactorily and additionally beat diverse figuring extensively to the extent runtime, especially when databases contain clusters of long trades.

Keywords: Data mining, Utility mining, Frequent itemsets, High utility itemsets.

1. Introduction

Frequent itemset mining (FIM) [1] is an essential study point in information mining. Sole of its well-known implementation is market basket [2] examination, which alludes to disclosure of sets of things/items (itemsets) which are much of the time purchased together by clients. Be that as it may, in this implementation, the conventional framework of FIM may find a lot of frequent however little income/profit itemsets and lose the data on important objectsets having little offering frequencies [3]. Mentioned issues are brought about with the truths that First, FIM regards each one of the things as having a similar significance/unit profit/weight and Second,this implementation accept that each thing inside an exchange shows up in a dual format, means, a thing may be either available or missing while doing transaction, that doesn't demonstrate its buy amount in transaction. Subsequently, FIM can't fulfil client's prerequisite who yearning to find objectsets with favourable high utilities [4], for example, high profits.

To approach these problem,utility mining [5] develops as a critical theme in information mining [6]. In utility mining, everything owns a weight (for example, unit profit) and may seem multiple times in every transaction (For example, buy amount). An utility of a objectset speaks to significance of itemset, which one can measure as far as cost, weight, amount/quantity [7], profit or other data relying upon the client inclination. An itemset is known as a high utility itemset (HUI) if utility of it is no not exactly or less than client specified least utility i.e. threshold; else, it known as a low utility object/item set. This utility mining is a critical assignment and has an extensive variety of utilizations, for example, click stream examination of websites [8], cross showcasing in retail shops [9],

versatile business environment and biomedical applications [7].

The majority of an earlier utility mining techniques having an itemset share structure embrace a two-stage, applicant era technique, in that,candidates which are having high utility(HUI) examples are found in the first phase [9],afterward in second stage ,It checks raw dataset/information once again to distinguish high utility (HUI) itemsets from the candidates which have been already obtained in first phase [10]. The test is that candidates' quality can be tremendous, which is a versatility as well as efficiency bottleneck. In spite of the fact that a great deal of exertion has been made to diminish candidates' quality, produced after first stage, a test however endures when dataset information contains numerous long exchanges or utility limit is little [11]. This type of numerous candidates causes adaptability issue in first stage as well as in the second stage, and thus debases the efficiency.

To resolve the drawback, this paper proposes another calculation, D2HUP, for utility mining having an objectset share structure, that utilizes a few systems suggested to mine frequent patterns, along with investigating a standard set list in the switch lexicographic pattern and heuristics for requesting things [12]. D2HUP algorithm scans the whole database in one phase only and find the itemsets which contains high utility patterns with respect to minimum utility which is considered as a threshold utility. D2HUP is more efficient than other existing techniques in terms of execution time as well as memory usage. D2HUP algorithm beats all state-of-arts techniques in terms on execution time.This is because the other existing techniques generates a huge number of candidates in first phase and first phase only takes more time to find all candidates from huge database and then they requires a next phase to find high utility patterns while D2HUP does not need second phase.This is the main advantage of using D2HUP algorithm to improve the performance in case of large database compared to existing techniques.

The next section lists down the existing works on different mining techniques. Section 3 elaborates the proposed methodology which includes approach and algorithm used for mining HUI effectively. Section 4 discusses experimental results and end section concludes the paper.

2. Literature Survey

R. Agrawal et al. [13] presented an efficient technique that produces all noteworthy association rules among objects in the database of consumer transaction which composed of things bought by a consumer.Technique incorporates buffer organization along with novel estimation as well as pruning capability. Also shown outcome after applying this method to retail company's huge sales data to show productiveness of their idea.

R. Srikant et al. [1] considered the issue of finding association rules between things in an expansive database of sales transactions and exhibited two unfamiliar computations for taking care of this problem are in a general motive not quite the same as the familiar methods. Experimental analysis demonstrated that those techniques beat the familiar techniques. Furthermore indicated how these best elements of these two suggested computations can be combined into hybrid computation, populated as AprioriHybrid.Scale-up tests reveals that it scales linearly with number of transactions.

R. Bayardo and R. Agrawal [14] recommended that the best rule as per any measurements must live along a support/confidence. Facilitate, on account of conjunctive rule mining inside categorical information, the quantity of principles along this border is advantageously little, and can be mined effectively from various real life information sets. They likewise indicated how this idea can be summed up to mine every rules which are optimal as indicated by some of these criteria regarding a discretionary subset of the number of interest and contended that by giving back a more extensive arrangement of standards.

F. Bonchi and B. Goethals [15] incorporated the as of late proposed ExAnte information reduction method inside the FP-growth technique. Together, they brought about an extremely proficient successive itemset mining calculation that viably abuses monotone constraints.

C. Lucchese et al. [16] presented limitation based example discovery with the point of giving to the client an instrument to drive the uncovering procedure towards conceivably interesting patterns, accompanied by an affirmative reaction of accomplishing an additional productive calculation likewise performed that their system beats past algorithms for convertible constraints, and endeavor the harder ones with a similar adequacy.

R. Chan et al. [7] tossed light upon a developing area known as Utility Mining that accounts the occurrence of the object/item sets as well as looks on utility connected with objectsets. In their paper they introduced a writing audit of the present situation with research and different algorithms for high utility mining.

R. J. Hilderman et al. [17] proposed the share-condense structure for information disclosure from databases that approaches the issue of mining object sets from market basket information and introduced another method for arranging objectsets according to characteristics features separated from registration or life-style information.At last they displayed experimental comes that exhibit the utility of the sharecon dence system.

From above mentioned literature survey we can make observation that High Utility Pattern mining problem is closely related to frequent pattern mining, including constraint-based mining. In reviewed prior works both on frequent pattern mining as well as on utility mining i.e, weighted item set mining and up-growth algorithm etc. are discussed.

Association mining and frequent itemset mining algorithms scans datasets as many times as the maximum length of frequent pattern in result so it leads to less time efficient. Also this techniques generates FP-trees in each iteration leads to memory as well as time inefficient.

Hybrid algorithm 'AprioryHybrid' dictated above, also stores the partitions of the datasets into the main memory and then it follows either BFS or DFS algorithm to prune the relevant itemsets which leads to poor performance in pruning technique.

From existing algorithm we observed that when two-phase algorithm is used to generate high utility itemsets then it first generates candidates in first phase and then in second phase it identify the high utility itemsets with the help of candidates which are generated in first phase. So it takes more time to scan whole data base when there are more relevant itemsets are present in database. Addition to this performance issue we can also include that this algorithm consumes more primary memory because of its tree based data structure. Form above observations we can conclude that some techniques should be required to mine high utility pattern which should be more efficient in terms of time as well as space. So we have proposed a technique called D2HUP which find high utility itemsets from database in one phase without finding a candidates.

3. Proposed work

Mining of high utility [14] object set/itemsets from database which has vast transactions mentions to disclosure of object sets with more profits, weights [7] etc. Despite the fact that various significant approaches have been presented as of late, they cause the issue of delivering countless itemsets with high utility. The circumstance may turned out to be more terrible when the database has heaps of long transactions.

From above study of related work we can make observation that High utility pattern mining issue is almost similar to frequent pattern mining [1], with constraint-based mining. In reviewed existing works both on frequent pattern mining as well as on utility mining i.e, weighted item set mining and up-gowth [9] algorithm etc. are discussed and how our proposed work relates to and diverges from these existing works is mentioned. Form above observations we can conclude that some techniques should be required to mine high utility pattern which should be more efficient in terms of time as well as memory. With this aim in our mind high utility pattern mining approach is proposed which discovers HUI from large dataset in a single phase only without generating candidates.

3.1 Architecture

Figure 1 shows overall flow of the project. Admin loads the data/records from the database and applies D2HUP algorithm on that data along with minUtility given by user. High utility item sets will be generated after successful execution of D2HUP and result will be stored into the output file which can be used by user/client for further analysis.

3.2 Modules

3.2.1 User

User is a client who want to get high utility item sets from the data items which he/she has.

-Using this high utility item sets they can get idea about item sets which can be used to generate more profit.

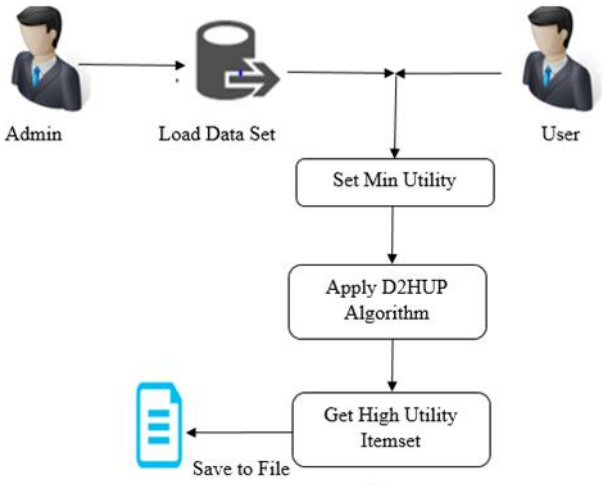

Figure.1 System Architecture

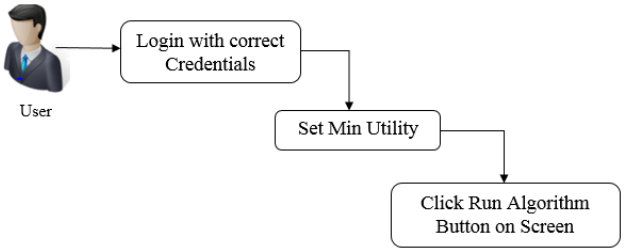

Figure.2 Flow of User Module

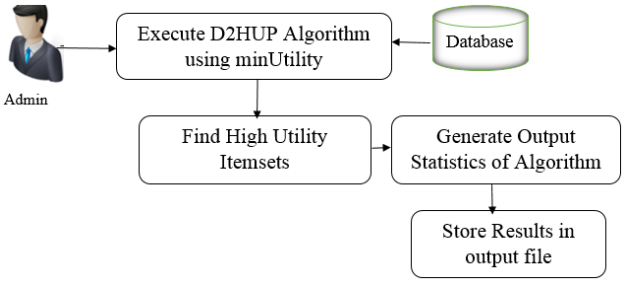

Figure.3 Flow of Admin Module

Algorithm: D2HUP (D, XUT, minUtility)

- Build Transaction set from D and XUT
- N <- root of reverse set enumeration tree
  DFS (N, TS (pat (N)), minUtility)
- If utility of high utility pattern pat(N) >= minimum Utility
  Then DFS prints high utility pattern
- Set W <- { i/ i is high utility pattern pat(N) where pat(N) >= minUtility
- If closure (pat(N),W,minUtility) is satisfied
  Then prints non-empty subset of W U pat (N)
- Else if singleton(pat(N),W,minUtility) is satisfied
  Then prints W U pat (N) as an High utility Pattern (HUP)
- Else for each item i ∈ W do
  If uB_{fpe} ({i} U pat (N)) >= minUtility
  Then C <- the child node for N for i
  TS (pat(C)) <- project (TS (pat (N)), i)
  DFS (C, TS (pat(C)), minUtility)

End for each

Figure.4 D2HUP Algorithm

-User has to login into system first if they want to find high utility pattern.

-After successfully logon, User need to give minimum utility value which can be considered as a threshold utility for administrator to find high utility item sets.

-After setting minUtility they can click on run algorithm button provided in the system to find high utility pattern.

3.2.2 Admin

-Admin module has the control over the algorithm.

-When client/user gives minimum utility value and click on run algorithm button at that time admin executes the algorithm and finds the high utility item sets from the database based on the minimum utility value given by user.

-Admin is also responsible of generating the statistics of algorithm and also result file which stores the output of algorithm.

3.3 Algorithm

Algorithm in Fig.4 generates TransactionSet TS{} by examining database 'D' along with utility table XUT to find utility u(i), uBfpe(i) and uBitem(i) for every item/thing 'i' and begins searching of HUI patterns from reverse set enumeration tree's root.

For root N which is being visited, DFS prints pat(N) as a HUI pattern if its utility is no not as much as the threshold (minimum) utility and prepare set named as 'W' of these kind of significant items. If closure property satisfies then DFS yields HU items as an extension of pat(N). If singleton property satisfies then it outputs union of every single relevant item and pat(N) as an HUI pattern. If (prefixExtension{i} U pat(N))'s utilities's upper bound is no not as much as threshold then DFS makes set of transactions of child node (C) and looks subtree of C recursively.

4. Result and discussion

In this section we will discuss about the result which we have got after implementing this algorithm. Our D2HUP algorithm has been executed with sample market transaction dataset and its utility table as an input.D2HUP algorithm scans the whole database in one phase only and find the itemsets which contains high utility patterns with respect to minimum utility which is considered as a threshold utility.

After successfully execution of algorithm User can see the 'Login Page' to login into a system as shown in Fig.5.

Figure.5 Login Page

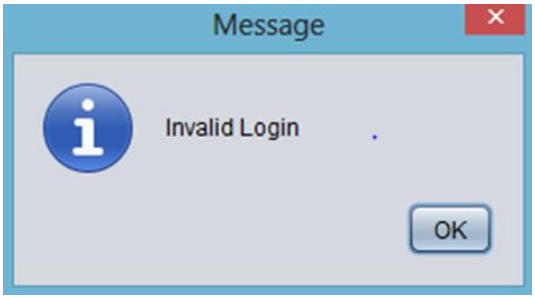

Figure.6 Login status

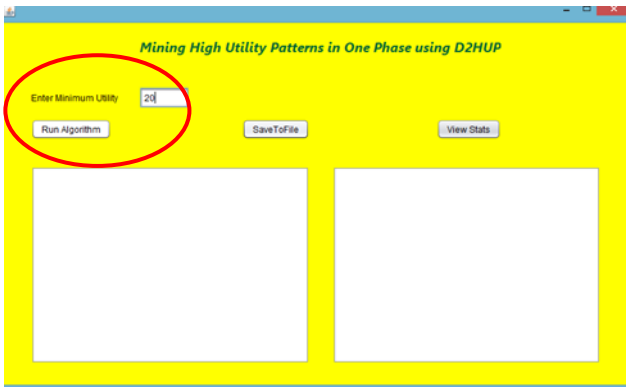

Figure.7 User screen

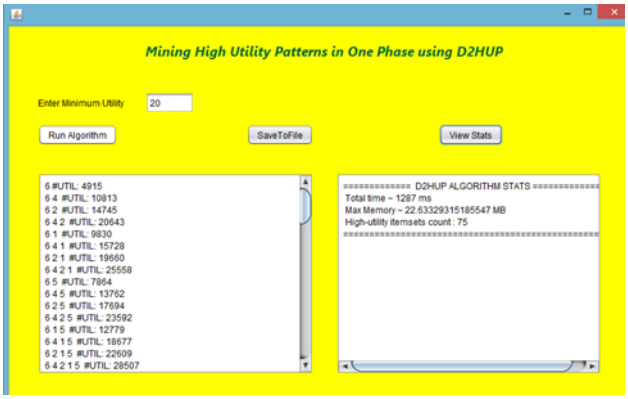

Figure.8 Outcome of Algorithm

User need to enter valid credential to login. If user enter wrong credential then system shows error pop-up to user as shown in Fig.6.

After successful login, User can see the main screen where user can set the minimum utility as threshold value. Once minUtility value is set then User can click on 'Run Algorithm' button highlighted in Fig.7.

When User click on 'Run Algorithm' button, Admin module starts the execution of algorithm at back-end by considering minimum utility as a threshold which is given by user.

Once the execution is done, it shows the list of all high utility itemsets along with its utility value at left panel given in Figure 8.When User click on 'view Stats' button, Admin module find the statistics of algorithm run i.e execution time taken by algorithm to run along with memory consumed by it. After calculating these statistics Admin module displays the result at right panel shown in Fig.8.

To permanently store, the outcome of an algorithm, this system has a facility to save the result in an output file. When user click on a 'Save to File' button, Admin module generate a file where it stores the result which is generated by execution of algorithm.

As mentioned earlier, this proposed approach directly find the HUI patterns in one phase only in absence of candidate generation. Algorithm scans the database only once and find the high utility patterns in that phase only by using CAUL based data structure to prune the prefix-extension of itemsets at back-end. CAUL based framework manages utility details on transaction sets{} for every enumerated itemset in a reverse set enumeration tree. Because of this advanced behavior of this algorithm at implementation side, it takes less time and less memory to execute as compare to other existing techniques. The result of an algorithm with respect to execution time and memory consumption is discussed in following part of this section. In Table 1, the after effects of different algorithms have been assessed in view of their execution time for mining the frequent objectsets and association rules from vast databases. The table demonstrates input file size and relating execution time.

Figure 9 demonstrates the graphical view of all algorithm's execution time mentioned in Table 1.The chart has been represented with the outcomes acquired by relating authors, which gives a clear thought regarding execution time of the considerable number of algorithms mentioned in Table 1. The execution time and size of input file of all algorithms have been utilized to plot a chart. The x-axis speaks to the various algorithm shown in Table 1 what's more, y-axis speaks to the execution time taken for input file size having 100KB and 1000KB.

In Table 2, the outcome of various algorithms has been analyzed with respect to memory consumption of respective algorithm. The table 2 demonstrates input file size and relating execution time of various algorithms.

Table 1. Execution time of various Algorithm

Algorithms	**Input File Size (Byte)**	**Execution time**
Two-Phase [18]	1000K	750 sec
Combined Approach [19]	100K	1500 sec
AprioriHybrid [1]	100K	7.5 sec
WARM [20]	1000K	1000 sec
D2HUP (Proposed)	100K	1.3 sec

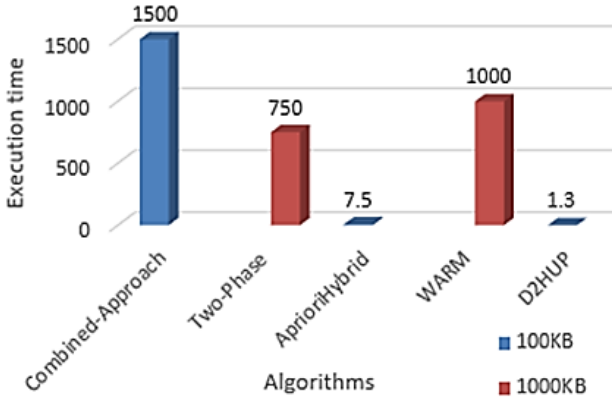

Figure.9 Execution Time graph of various algorithms for different input size

Table 2. Memory Consumption of various Algorithm

Algorithms	**Input File Size (Byte)**	**Memory Consumption**
Combined Approach [19]	10K	13 MB
H-Mine [21]	10K	300 KB
THUI [22]	10K	250 KB
D2HUP (Proposed)	100K	20 MB

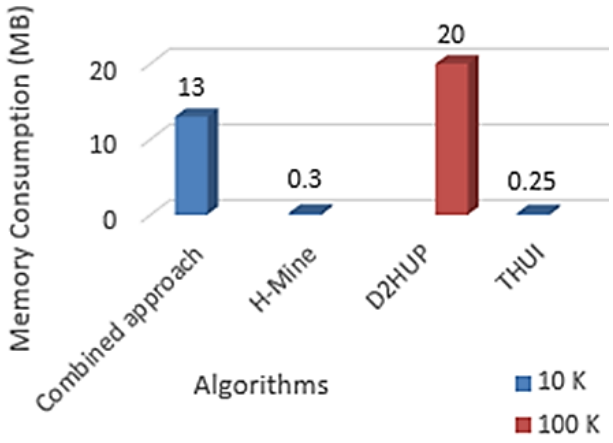

Figure.10 Memory consumption graph of various algorithms for different input size

Figure 10 demonstrates the graphical view all algorithms' memory consumption during its execution. The graph has been plotted with outcome gathered by relating authors, which shows clear idea of memory consumption of some algorithms with different input file size that mentioned in Table 2. The X-axis represents the various algorithms given in Table 2 and Y-axis represents memory consumed in Megabytes by various algorithms for input file size 10KB as well as 100KB mentioned in Table 2.

From above analysis it is observed that our proposed technique D2HUP is more efficient than other existing techniques in terms of execution time as well as memory usage. D2HUP algorithm beats all algorithms mentioned in Table 1 in terms on execution time while it is more memory efficient than Combined-Approach technique mentioned in Table 2. The other existing techniques generates a huge number of candidates in first phase and first phase only takes more time to find all candidates from huge database and they requires a next phase to find high utility patterns while D2HUP does not need second phase. This is the main advantage of using D2HUP algorithm to improve the performance in case of large database.

5. Conclusion

Mining of frequent objectset and association rule is a critical undertakings of information mining. Considering utility with mining undertakings is picking up prominence as of late.Proposed technique D2HUP discovers HighUtility patterns without candidate generation in single phase without scanning dataset multiple times. D2HUP does not need second phase to generate high utility patterns so its takes less time compared to other algorithms. The root cause reason of generation of high utility itemsets without candidate generation in single phase is CAUL based framework which is used along with reverse set enumeration. From obtained result we conclude that our D2HUP technique is more efficient in terms of execution time with different input file size and beats earlier algorithms incredibly in terms of high performance. As a future work of this research we can enhance this work to optimize memory consumption as compared to other existing methods as well.

Acknowledgments

I, Ghetiya Rutvi Babulal would like to avail this opportunity to express my sincere thanks and gratitude to VIT University. The successful completion of this work was made possible with the guidance received from my guide Dr. P. Gayathri. She has guided me for successful completion of the project with her enthusiasm, wide support, co-operation and valuable advices.

References

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *VLDB*, Vol. 1215, pp. 487-499, 1994.

[2] S. Brin, J.D.U.R. Motawani, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data", In: *Proc. of ACM*

SIGMOD international conference on Management of data, Vol. 26, No. 2, pp. 255-264, 1997.

[3] M.J. Zaki and C.J. Hsiao, "Efficient algorithms for mining closed itemsets and their lattice structure", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 4, pp. 462-478, 2005.

[4] V.S. Tseng, C.W. Wu, and P.F. Viger, "Efficient Algorithms for Mining Top-K High Utility Itemsets", *Efficient Algorithms for Mining Top-K High Utility Itemsets*, Vol. 28, No. 1, pp.54-67, 2016.

[5] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets", In: *Proc. of the 12th Pacific-Asia Conference*, pp. 554-561, 2008.

[6] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and Y. K. Lee, "Efficient tree structure for high utility pattern mining in incremental databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 12, pp. 1708-1721, 2009.

[7] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," In: *Proc. of ICDM. IEEE*, pp. 19-26, 2003.

[8] H.F. Li, H.Y. Huang, Y.C. Chen, Y.J. Liu, and S.Y. Lee, "Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams", In: *Proc. of the Eighth IEEE International Conference on Data Mining*, pp. 881-886, 2008.

[9] V.S. Tseng, C.W. Wu, B.E. Shie, and P.S. Yu, "UP-Growth:An efficient algorithm for high utility itemset mining", In: *Proc. of ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pp. 253-262, 2010.

[10] J. Liu, K. Wang, and B. C.M.Fung, "Mining High Utility Patterns in One Phase without Generating Candidates", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 5, pp. 1245-1257, 2016.

[11] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", In: *Proc. of SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, Vol. 29, No. 2, pp. 1-12, 2000.

[12] C.W. Lin, T.P. Hong, and W.H. Lu, "An effective tree structure for mining high utility itemsets", *Expert Systems with Applications*, Vol. 38, No. 6,pp. 7419-7424, 2011.

[13] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", In: *Proc. of SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, Vol. 22,No. 2, pp. 207-216, 1993.

[14] R. Bayardo and R. Agrawal, "Mining the most interesting rules", In: *Proc. of SIGKDD. ACM*, pp. 145-154, 1999.

[15] F. Bonchi and B. Goethals, "Fp-bonsai: The art of growing and pruning small fp-trees," In: *Proc. of PAKDD*, pp. 155-160, 2004.

[16] F. Bonchi and C. Lucchese, "Extending the state-of-the-art of constraint-based pattern discovery", *Data and Knowledge Engineering*, Vol. 60, No. 2, pp. 377-399, 2007.

[17] R. J. Hilderman, C. L. Carter, H. J. Hamilton, and N. Cercone, "Mining market basket data using share measures and characterized itemsets", In: *Proc. of PAKDD*, pp. 159-173, 1998.

[18] Y. Liu, W.-K. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm", In: *Proc. of UBDM '05 Proceeding of the 1st international workshop on Utility-based data mining*, pp. 90-99, 2005.

[19] L. Dong and C. Tjortjis, "Experiences of Using a Quatitative Approach for Mining Association Rules", *Intelligent Data Engineering and Automated Learning*, pp. 693-700, 2003.

[20] W. Wang, J. Yang, and P. S. Yu, "Efficient mining of weighted association rules (WAR)", In: *Proc. of KDD '00 Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 270-274, 2000.

[21] J. Pei, J. Han, H. Lut, S. Nishio, S. Tang, and D. Yang, "H-Mine: Fast and space-preserving frequent pattern mining in large databases", *IIE Transactions*, Vol. 39, No. 6, 2007.

[22] C.J. Chu, V. S. Tseng, and T. Liang, "An efficient algorithm for mining temporal high utility itemsets from data streams", *Journal of*

System and Software, Vol. 81, No. 7, pp. 1105-1117, 2008.