www.ioirp.com
International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)
ISSN: 2394-6364, Volume – 1, Issue – 6. May 2015

AUP-GROWTH METHORD FOR MINING HIGH UTILITY ITEMSET FROM TRANSACTIONAL DATABASE

[1]Drishya.T, [2]Mrs.A.S.Shanthi.M.E.,(Ph.D).,
[12]Department of Computer Science and Engineering, Tamilnadu College of Engineering, Coimbatore, India.
[1]drishya.sub@gmail.com, [2]babushanthi@gmail.com

***Abstract*— Data Mining is an interdisciplinary subset of Computer Science. It is applied whenever information from a dataset is to be extracted and transformed into some understandable format for easy usage. This paper introduces an efficient technique to mine high utility itemsets from a transactional database by using the proposed algorithm AUP (Advanced UP- Growth algorithm). UP-growth algorithm is an efficient technique to mine high utility itemsets,yet the numbers of Potential High Utility Itemsets generated are high. In this paper, the traditional UP-Growth method and UP-Growth+ is extended to form the AUP–Growth and AUP-Growth+ method to reduce the number of PHUI'S.**

Keywords*— Candidate pruning, frequent itemset, high utility itemset, utility mining, data mining.

I. INTRODUCTION

Generally, Data mining is the process of analyzing of data from different angles and thereby extracting useful information. Data mining software act as an analytic tool which allows the users to analyze, categorize and summarize data based upon the relationships identified. There are many data mining tasks such as frequent data mining, weighted frequent data mining and high utility pattern mining which require the discovery of useful patterns that are hidden in the database. The application of frequent data mining can be seen in different types of databases including streaming databases, transactional databases and time series databases and also in many application domains.

Weighted Association Rule mining emerged as a research area which overcame the limitation of frequent pattern mining which does not considers the relative importance of items. In WAR each items in a transaction is associated with a weight which reflects its importance within the transaction. In certain cases infrequent items with may also be important and Weighted Association Rule helps in finding out such items with high weights. Even though this approach identifies the relative weights of each item it does not take into account their quantities in each transaction.

Utility Mining gives the simple idea to mine items with high utility or high profit. Every item in a transaction contains two utilities, internal and external. External utility is the utility or importance of each item and internal utility is the utility of an item in a particular transaction. Utility of an itemset as a whole is the product of both i.e., its internal utility and external utility. Based upon this utility items can be put into two classes, high utility itemset and low utility itemset. A High utility item always has utility greater than a user specified limit and low utility item has utility less the user specified limit. Utility mining can be applied in many areas like website click stream analysis, online e-commerce management and many more.

II. PROBLEM STATEMENT

Let I = {i1,i2,i3,....,in}, im (1≤m≤n) be a finite set of items each having a unit profit p(im). Let A be an itemset with j distinct elements (i1,i2,....,ij) where ik I, 1≤k≤j.let T={T1,T2,....,Tn} represent a transactional database in which each transaction ty has an identifier Tid . A quantity q(im,Ty) is assigned to each item in each transcation which represents the quantity of item impurcashed in transaction Ty.

- util(im,Ty) denotes the utility of item im in transaction Ty and is defined as
$$util(im,Ty) = p(im) \times q(im,Ty) \quad (1)$$
- util(A,Ty) denotes the utility of itemset in transaction Ty and is defined as
$$util(A,Ty) = \sum_{i_m \in A \cap A \subseteq T} util(i_m, T_y) \quad (2)$$
- util(A) denotes the utility of itemset A in database T and is defined as

$$\text{util}(A) = \sum_{A \in T_y \cap T_y \subseteq T} \text{util}(A, T_y) \quad (3)$$

- An itemset A is called a High Utility Itemset if its utility is always greater than a user specified threshold utility level denoted as thres_util().
- trans_util(Ty) denotes the transaction utility of Ty and is defined as util(Ty,Ty).
- trans_wght_util(A) denotes the transaction weighted utility of itemset A and is defined as

$$\text{trans\_wght\_util}(A) = \sum_{A \in T_y \cap T_y \in T} \text{trans\_util}(T_y) \quad (4)$$

- HTWUIS denotes a High Transaction Weighted Utility Itemset and is defined as an Itemset with trans_wght_util(A) greater than thres_util().

Given a transactional database T containing the purchase details of customers and user specified threshold utility level thres_util(),mining High UtiityItemsets using AUP refers to the efficient discovery of itemsets with high utility,i.e. high profit items.

III. Related Works

Extraction of frequent patterns [1], [5], [7], [13], [21], [22], [36], [40] in research field gained a lot of attention for many years.Association Rule mining [1], [7], [13], [21], [36], [40] and sequential pattern mining [5], [22] are based on frequent pattern mining.Apriorialgorithm[1], and FP-Growth [13] are both based on association rule mining.

Weighted association rule mining was then introduced to overcome the limtitations of association rule mining [3], [26], [28], [31], [37], [38], [39]. Weighted pattern mining [6], [26], [37]introduced new methods of pattern mining. Utility mining [2], [4], [11], [16], [17], [19], [24], [25], [29], [30], [32], [33] was proposed after ARM since it does not consider the quantity of each items in a transaction.

Two Phase algorithm [19] that was proposed by liuet al. performed mining of itemsets in two phases. Phase1 generates the HTWUIusing Apriori and in phase2 high utility itemsets are identified from the output of previous phase. Two Phase algorithm has the problem of generating too many HTWUI's.

Lie et al[17] proposed another method,the IIDS i.e. isolated items discarding strategy which efficiently reduced the number of generated candidate itemsets.The number of HTWUI's generated during the phase 1 is reduced by pruning out the isolated items while performing the level-wise search.

Ahmed et al [2] introduced a tree based algorithm called IHUP.IHUP efficiently generated THWUI's during the first phase by using a tree based structure called IHUP-tree which maintained the details of itemsets and their corresponding utilities.The use of IHUP –tree also reduced the number of database scans.

IV. Proposed Method

The proposed methods consist of three steps. First scan the database twice to construct a global UP Tree with two strategies Discarding Global Unpromising Items (DGU) and Decreasing Global Node Utilities (DGN) during constructing a Global AUP-Tree. Second recursively generate PHUIs from global AUP-Tree and local AUP-Trees by AUP-Growth with two strategies Discarding Local Unpromising items (DLU) and Decreasing Local Node utilities (DLN) or by AUP-Growth+ with two strategies Discarding Local unpromising items(DNU) and Decreasing node utilities (DNN) and then identify actual high utility itemsets from the set of PHUIs .

A. Construction of AUP-Tree

To enhance the mining performance and avoid scanning original database repeatedly, we use a compact tree structure, named AUP-Tree, to maintain the information of transactions and high utility itemsets along with two strategies to minimise the overestimated utilities stored in the nodes of global AUP-Tree.

In an AUP-Tree, each node M contains M.name, M.count , M.nu, M.parent and a set of child nodes. M.name denotes the node's item name. M.count denotes the node's support count. M.nu denotes the node's node utility, which is the overestimated

utility of the node. M.parent represents the parent node of M. A table named header table is employed to facilitate the traversal of AUP-Tree. In header table, each entry contains an item name and an overestimated utility.

The construction of a global AUP-Tree can be performed with two scans of the original database. In the first scan, Trans_Util (Transaction Utility) of each transaction is computed. At the same time, Trans_Wght_Util (Transaction Weighted Utility) of each single item is also calculated. An item and its supersets are high utility itemsets if its Trans_Wght_Util is less than the minimum utility threshold and such an item is referred to as an unpromising item. AUP-tree is constructed with two strategies DGU and DGN. First strategy is discarding global unpromising items and their actual utilities from transactions and transaction utilities of the database. New Trans_Util after pruning unpromising items is called reorganized transaction utility (RTU). RTU (Tr) denotes the RTU of a reorganized transaction Tr. Strategy DGU make use of RTU to overestimate the utilities of itemsets instead of Trans_Wght_Util. The second strategy is DGN, decreasing global node utilities for the nodes of global AUP-Tree by actual utilities of descendant nodes during the construction of global AUP-Tree.

B. The Proposed Mining Method: AUP-Growth

AUP growth algorithm is applied after the construction of global AUP-Tree for mining high utility itemsets. The common method for generating patterns in tree-based algorithms contain three steps: 1) Generate conditional pattern bases by tracing the paths in the original tree; 2) construct conditional trees (also called local trees) by the information in conditional pattern bases; and 3) mine patterns from the conditional trees. However, the two strategies DGU and DGN are not applicable in conditional AUP-Trees since actual utilities of items in different transactions are not maintained in a global AUP-Tree. The actual utilities of unpromising items that need to be discarded in conditional pattern bases cannot be determined unless an additional database scan is performed. This problem can be overcome by using a naıve solution i.e. is to maintain items' actual utilities in each transaction into each node of global AUP-Tree. However, this is impractical since it needs lots of memory space. The two strategies requires a minimum item utility table to keep minimum item utilities for all global promising items in the database. Minimum item utility of item I_{tm} in database D, denoted as miu(I_{tm}), is I_{tm}'s utility in transaction Td if there does not exist a transaction Td' in D such that util(I_{tm},Td')<u(I_{tm},Td).

Minimum item utilities are utilized to reduce utilities of local unpromising items in conditional pattern bases instead of exact utilities. From the path utility of an extracted path an estimated value for each local unpromising item is subtracted. Path utility of a path p in I_{tm}'s conditional pattern base (abbreviated as {i_{tm}}-CPB) is denoted as PU(p, {I_{tm}}-CPB) and defined as NI_{tm}'s node utility where p is retrieved by tracing NI_{tm} in the AUP-Tree.

AUP growth is implemented by using two strategies. First strategy is DLU (Discarding Local Unpromising items). Local unpromising items and their estimated utilities are discarding from the paths and path utilities of conditional pattern bases. It provides a useful schema to reduce overestimated utilities locally without an extra scan of original database. Second strategy is DLN (Decreasing Local Node utilities) for the nodes of local AUP-Tree by estimated utilities of descendant nodes.

The process of mining PHUIs by AUP-Growth is a step by step process: First the bottom entry in header table i.e. item im is considered and it's node links in AUP-Tree is traced. All found nodes are traversed up to root of the AUP-Tree to retrieve every paths related to im. The resultant conditional pattern base of im consists of all retrieved paths along with their path utilities and support counts. A conditional AUP-Tree can be constructed by two scans of a conditional pattern base. In the first sacn local promising and unpromising items are determined by summing the path utility for each item in the conditional pattern base. Then, DLU is used to decrease the overestimated utilities during the second scan of the conditional pattern base. After retrieving a path, unpromising items and their estimated utilities are eliminated from the path and its path utility by (1). Then the path is reorganized by the descending order of path utility of the items in the conditional pattern base. During the insertion of reorganized paths into a conditional AUP-Tree DLN applied.

After the candidate itemsets are generated the next phase of AUP growth is initiated. All candidates items that has a transaction support above minimum support is found out. The number of transaction having the items is the support count of that item. Itemset having support greater than or equal to minimum support value is known as large itemsets and those having a lesser support value is known as small itemsets. The algorithm performs multiple passes on the data to determine the large itemsets. The support count of individual items are determined the first pass to find out items having minimum support. These resultant items are known as seed itemsets. These seed itemsets of previous step becomes the input for the next pass and is used for generation potentially large itemsets. For each potentially large itemset it's actual support count is determined and actual large itemsets are identified. This process continues iteratively until no large itemsets are left.

C. The proposed mining method: AUP-Growth+

AUP-Growth achieves better performance than FP-Growth by using DLU and DLN to decrease overestimated utilities of itemsets. The overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes. There is also an improved method, named AUP-Growth+, for reducing over estimated utilities more effectively. In AUP-Growth, minimum item utility table is used to reduce the overestimated utilities. In AUP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

Minimal node utility for each node can be acquired during the construction of a global AUP-Tree. First, we add an element, namely M.mnu, into each node of AUP-Tree. M.mnu denotes the minimal node utility of N. When N is traced, M.mnu keeps track of the minimal value of M.name's utility in different transactions. If M.mnu is larger than u(M. name, Tcurrent), M. mnu is set to u (M.name, Tcurrent). When a local AUP Tree is being constructed, minimal node utilities can also be acquired by the same steps of global AUP-Tree. In the mining process, when a path and the minimal node utility of each and every node in the path is also retrieved. In AUP-Growth+ algorithm, minimum item utilities are replaced with minimum node utilities. AUP-Growth+ algorithm uses two strategies .That are, DNU and DNN. First strategy is discarding local unpromising items and their estimated node utilities from the paths and path utilities of conditional pattern bases and Second strategy is decreasing local node utilities for the nodes of local AUP-Tree by estimated utilities of descendant nodes. After the generation of PHUI's ,items that has a greater transaction support is found out. Those items are considered as High Utility Items.

V. EXPERIMENTAL RESULT

To understand the performance of the proposed AUP Growth and AUP Growth+ algorithm many experiments was performed on real and synthetic data sets 2.80 GHz Intel Pentium D Processor with 3.5 GB memory is used to perform the experiments. Microsoft Windows 7 is the operating system used. Dot net framework is used for implementation. The result of mining process in case of UP-Growth, UP-Growth+, AUP-Growth and AUP-Growth+ is shown in the figures below.

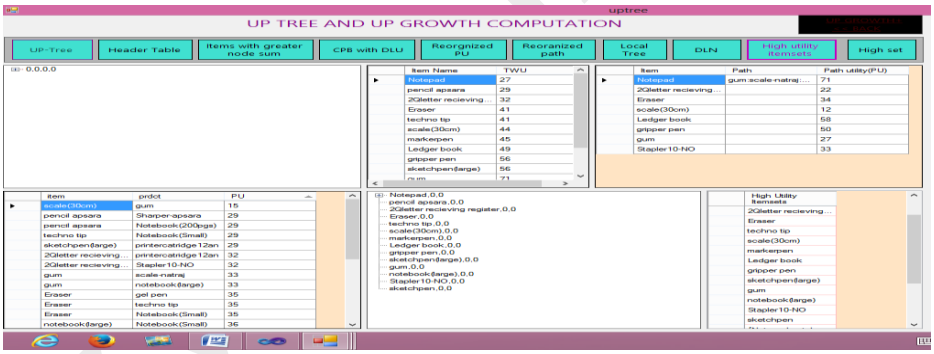

Fig 1: Result of mining process in UP-Growth

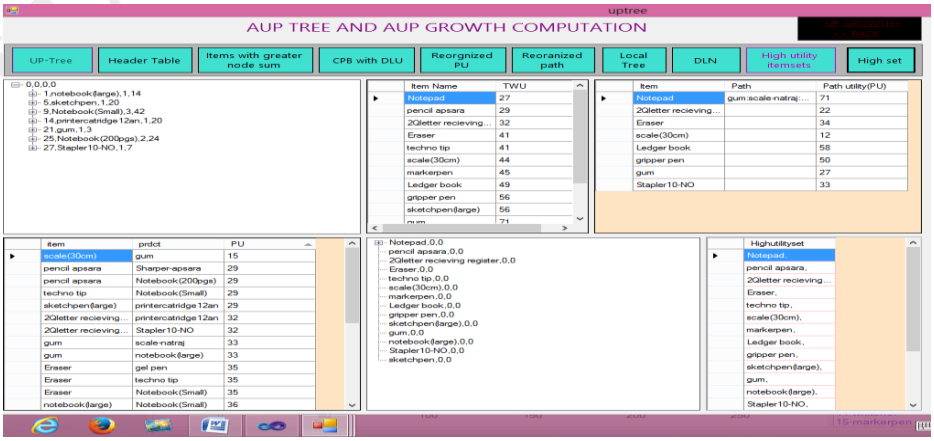

Fig 2 : The result of mining process in AUP-Growth

From the experimental results it is seen that the number of PHUI's generated by AUP-Growth is lesser than that of normal UP-Growth algorithm. AUP-Growth+ algorithm is found to outperform AUP-Growth algorithm since it utilizes minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

VI. CONCLUSION

This paper proposes a method to efficiently mine high utility itemsets from a transaction database by using three algorithms UP-Growth, UP-Growth+ and Apriori. UP-tree is a data structure used to store information about the high utility itemsets. The Potential High Utility Itemsets are generated by the UP-tree by scanning the database just twice.On the basis of experimental results, it can be seen that these algorithms reduces both the search space and also the number of candidate itemsets generated.

References

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB), pp. 487-499, 1994.

[2] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12,pp. 1708-1721, Dec. 2009.

[3] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items," Proc. Int'l Database Eng.and Applications Symp. (IDEAS '98), pp. 68-77, 1998.

[4] R. Chan, Q. Yang, and Y. Shen, "Mining High Utility Itemsets,"Proc. IEEE Third Int'l Conf. Data Mining, pp. 19-26, Nov. 2003.

[5] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc.11th Int'l Conf. Data Eng., pp. 3-14, Mar. 1995.

[6] J.H. Chang, "Mining Weighted Sequential Patterns in a Sequence Database with a Time-Interval Weight," Knowledge-Based Systems,vol. 24, no. 1, pp. 1-9, 2011.

[7] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," Proc. 21th Int'l Conf. Very Large Data Bases,pp. 420-431, Sept. 1995.

[8] M.-S. Chen, J.-S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns," IEEE Trans. Knowledge and Data Eng., vol. 10,no. 2, pp. 209-221, Mar. 1998.

[9] C. Creighton and S. Hanash, "Mining Gene Expression Databases for Association Rules," Bioinformatics, vol. 19, no. 1, pp. 79-86,2003.

[10] M.Y. Eltabakh, M. Ouzzani, M.A. Khalil, W.G. Aref, and A.K.Elmagarmid, "Incremental Mining for Frequent Patterns in Evolving Time Series Databases," Technical Report CSD TR#08-02, Purdue Univ., 2008.

[11] A. Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Data Sets," Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD),pp. 554-561, 2008.

[12] E. Georgii, L. Richter, U. Ru¨ ckert, and S. Kramer, "Analyzing Microarray Data Using Quantitative Association Rules," Bioinformatics,vol. 21, pp. 123-129, 2005.

[13] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM-SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000.

[14] J. Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database," Proc. Int'l Conf. on Data Eng.,pp. 106-115, 1999.

[15] S.C. Lee, J. Paik, J. Ok, I. Song, and U.M. Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns," Int'l J.Computer Science Security, vol. 7, no. 2, pp. 285-291, 2007.

[16] 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," Proc. IEEE Eighth Int'l Conf. on Data Mining, pp. 881-886, 2008.

[17] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated Items Discarding Strategy for Discovering High Utility Itemsets," Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.

[18] C.H. Lin, D.Y. Chiu, Y.H. Wu, and A.L.P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," Proc. SIAM Int'l Conf. Data Mining (SDM '05), 2005.

[19] Y. Liu, W. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop,2005.

[20] R. Martinez, N. Pasquier, and C. Pasquier, "GenMiner: Mining nonredundant Association Rules from Integrated Gene Expression Data and Annotations," Bioinformatics, vol. 24, pp. 2643-2644,2008.

[21] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, "H-Mine:Fast and Space-Preserving Frequent Pattern Mining in Large Databases," IIE Trans. Inst. of Industrial Engineers, vol. 39, no. 6,pp. 593-605, June 2007.

[22] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Moal, and M.C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The Prefixspan Approach," IEEE Trans. Knowledge and Data Eng.,vol.16, no.10, pp. 1424-1440, Oct. 2004.

[23] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W.K. Liao, A.Choudhary, and G. Memik NU-MineBench Version 2.0 Data Set and Technical Report, http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html, 2012.

[24] B.-E. Shie, H.-F. Hsiao, V., S. Tseng, and P.S. Yu, "Mining High Utility Mobile Sequential Patterns in Mobile Commerce Environments,"Proc. 16th Int'l Conf. DAtabase Systems for Advanced Applications (DASFAA '11), vol. 6587/2011, pp. 224-238, 2011.

[25] B.-E. Shie, V.S. Tseng, and P.S. Yu, "Online Mining of Temporal Maximal Utility Itemsets from Data Streams," Proc. 25th Ann.ACMSymp. Applied Computing, Mar. 2010.

[26] K. Sun and F. Bai, "Mining Weighted Association Rules without Preassigned Weights," IEEE Trans. Knowledge and Data Eng.,vol. 20, no. 4, pp. 489-495, Apr. 2008.

[27] S.K. Tanbeer, C.F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Efficient Frequent Pattern Mining over Data Streams," Proc. ACM 17th Conf. Information and Knowledge Management, 2008.

[28] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining(KDD '03), pp. 661-666, 2003.

[29] V.S. Tseng, C.J. Chu, and T. Liang, "Efficient Mining of TemporalHigh Utility Itemsets from Data Streams," Proc. ACM KDD Workshop Utility-Based Data Mining Workshop (UBDM '06), Aug.2006.

[30] V.S. Tseng, C.-W. Wu, B.-E. Shie, and P.S. Yu, "UP-Growth: AnEfficient Algorithm for High Utility Itemsets Mining," Proc. 16thACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10), pp. 253-262, 2010.

[31] W. Wang, J. Yang, and P. Yu, "Efficient Mining of WeightedAssociation Rules (WAR)," Proc. ACM SIGKDD Conf. KnowledgeDiscovery and Data Mining (KDD '00), pp. 270-274, 2000.

[32] H. Yao, H.J. Hamilton, and L. Geng, "A Unified Framework forUtility-Based Measures for Mining Itemsets," Proc. ACM SIGKDDSecond Workshop Utility-Based Data Mining, pp. 28-37, Aug. 2006.

[33] S.J. Yen and Y.S. Lee, "Mining High Utility QuantitativeAssociation Rules." Proc. Ninth Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK), pp. 283-292, Sept. 2007.

[34] C.-H. Yun and M.-S. Chen, "Using Pattern-Join and Purchase-Combination for Mining Web Transaction Patterns in an ElectronicCommerce Environment," Proc. IEEE 24th Ann. Int'l ComputerSoftware and Application Conf., pp. 99-104, Oct. 2000.

[35] C.-H. Yun and M.-S. Chen, "Mining Mobile Sequential Patterns ina Mobile Commerce Environment," IEEE Trans. Systems, Man, andCybernetics-Part C: Applications and Rev., vol. 37, no. 2, pp. 278-295,Mar. 2007.

[36] S.J. Yen, Y.S. Lee, C.K. Wang, C.W. Wu, and L.-Y. Ouyang, "The Studies of Mining Frequent Patterns Based on Frequent Pattern Tree," Proc. 13th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), vol. 5476, pp. 232-241, 2009.

[37] U. Yun, "An Efficient Mining of Weighted Frequent Patterns withLength Decreasing Support Constraints," Knowledge-Based Systems,vol. 21, no. 8, pp. 741-752, Dec. 2008.

[38] U. Yun and J.J. Leggett, "WFIM: Weighted Frequent ItemsetMining with a Weight Range and a Minimum Weight," Proc. SIAM Int'l Conf. Data Mining (SDM '05), pp. 636-640, 2005.

[39] U. Yun and J.J. Leggett, "WIP: Mining Weighted InterestingPatterns with a Strong Weight and/or Support Affinity," Proc. SIAM Int'l Conf. Data Mining (SDM '06), pp. 623-627, Apr. 2006.

[40] M.J. Zaki, "Scalable Algorithms for Association Mining," IEEETrans. Knowledge and Data Eng., vol. 12, no. 3, pp. 372-390, May2000.

[41] Frequent Itemset Mining Implementations Repository, http://fimi.cs.helsinki.fi/, 2012.