Enhancing High Utility Pattern Mining using Gradual Pruning

[1] N.Asha Pranuthi, [2]Dr. Kondapalli Venkata Ramana
[1]M.Tech Scholar, Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India.
[2]Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India

Abstract:
Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits or revenues. Although there are many algorithms to extract utility itemsets, they incur the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space requirement. Earlier work shows this on two phase candidate generation. This approach suffers from scalability issue due to the huge number of candidates. Our paper presents the efficient approach where we can generate high utility patterns in one phase without generating candidates. Here we have take experiments on linear data structure, our pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding. Also high utility patterns are identified by a closure property and singleton property. This sort of large wide variety of candidate object units degrades the mining overall performance in phrases of execution time and area requirement. The situation may additionally end up worse at the same time as the database consists of prolonged transactions or long immoderate software object units. To overcome and speed up process we implemented pruning techniques. This paper proposes a novel algorithm that finds high utility patterns in a single phase without generating candidates. It uses a high utility pattern growth approach, a lookahead strategy, and a linear data structure. The pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding. We thus propose an efficient utility mining algorithm, which is a pruning approach (also termed the gradual pruning approach, or GPA), to discover high utility itemsets from a database. The Experimental consequences show that the proposed set of rules, especially application sample increase plus, required a lot less execution time and decreased reminiscence usage even as databases encompass lots of the excessive transactions.

***Index Terms* – Data mining, utility mining, high utility patterns, frequent patterns, pattern mining, gradual pruning approach**

I. INTRODUCTION

Data mining is the process of discovering hidden patterns and knowledge with in large amounts of data and also making predictions for behaviors or outcomes [2]. Data mining is commonly used to discover interesting patterns and knowledge discovery from massive data. Interestingness measures play an important role in knowledge discovery process and these measures are intended for selecting and ranking patterns according to their potential interest to the user. Finding interesting patterns is essential for variety of applications such as financial data analysis, retail system, market basket analysis, customer profiling, targeting, inventory prediction, and condition monitoring and genome analysis [9]. The discovered knowledge from data mining is generally classified as association rule, sequential patterns, frequent patterns and utility patterns. In recent past, utility pattern mining has emerging as an important research topic since the quantity and profit factors are both used to discover interesting patterns from huge amounts of data. Utility pattern mining addresses the limitation of frequent pattern mining by considering user's expectation as well as the raw data [18]. In frequent pattern mining, the user can easily express their perception regarding the itemset and its attributes and can mine the itemset by defining a threshold value. But in the case of utility pattern mining, these factors take a change. The utility value changes with the preferences of the employed user. The restrictions of frequent or rare itemset mining inspired researchers to conceive a utility based mining approach, which allows a user to conveniently express the perspectives concerning the usefulness of itemsets as utility values and then find itemsets with high utility values higher than a threshold [5].

The various approaches and the process for high utility pattern mining have been reviewed [16]. Among utility mining problems, high utility pattern mining with the itemset framework is more challenging than the other categories of utility mining and frequent pattern mining. This research paper proposes an enhanced high utility pattern approach (EHUPA) for mining high utility itemsets. Concretely, the interestingness measures in the latter classes study anti-monotonicity properties, that is, a superset of an uninteresting pattern is likewise. Such a belongings may be hired in pruning are trying to find region, which is likewise the foundation of all common pattern mining algorithms [3]. The anti-monotonicity assets does no longer examine to software mining with the itemset proportion framework [39], [40]. Therefore, utility mining with the itemset percent framework is extra hard than the other classes of software program mining in addition to frequent sample mining. most of the preceding software mining algorithms with the itemset proportion framework [4], [15], [24], [29], [38], [39] adopt a -section, candidate technology technique, that is, first find candidates of immoderate utility styles in the first segment, after which test the raw records one greater time to identify excessive application patterns from the applicants inside the second phase. The venture is that the range of candidates may be big, that is the scalability and performance bottleneck. Even though a diffusion of effort has been made [4], [15], [24], [38] to reduce the variety of candidates generated within the first phase, the mission however persists while the raw records includes many lengthy transactions or the minimal application threshold is small. This sort of large quantity of candidates reasons scalability problem not simplest in the first section but also inside the second section, and therefore degrades the performance. One exception is the HUIMiner set of rules [28], it is but even much less in experienced than -section algorithms mining massive databases due to inefficient be a part of operations, loss of robust pruning, and scalability trouble with its vertical statistics shape. The high utility itemsets, which had utility values larger than or equal to a predefined threshold could thus be easily found. However, since the downward-closure property in association-rule mining [1] cannot be directly used in utility mining [3].

To deal with this, Liu et al. thus proposed a two-phase utility mining (TP) algorithm to find high utility itemsets from databases by adopting their specific downward-closure property [12]. By this property, the utility values of all the items in a transaction were summed up as the transaction utility and used as the upper bound of any itemset in that transaction. This property was called the transaction-weighted utilization (TWU) model [12]. By using the TWU model, their proposed algorithm can effectively handle the problem of utility mining. However, when using the model, it was observed that many unpromising candidates were still generated in each pass for mining. In this study, we thus propose an efficient utility mining algorithm, which is a level-wise pruning approach (also termed the gradual pruning approach, or GPA), to discover high utility itemsets from a database. In particular, a new pruning strategy can be applied to reduce the number of candidates in each pass, in which unpromising items are removed early from transactions to make more precise utility upper bounds for itemsets. Also, the data size in each pass can be gradually reduced to save the scanning time needed for mining. The experimental results show that the number of candidates required by GPA is obviously less than that required by the TP algorithm [12]. The GPA algorithm executes faster than the TP algorithm as well. The remaining parts of this paper are organized as follows. The related works are reviewed. The problem to be solved and the proposed mining algorithm with a pruning strategy are stated. An example is then given to illustrate the detailed process of the proposed algorithm. The experimental results are next shown in Section 5, and the conclusions and directions for future work are given in. To address the venture, this paper proposes a new set of rules, d2HUP, for utility mining with the itemset share framework, which employs numerous techniques proposed for mining frequent styles, which include exploring a regular set enumeration in a reverse lexicographic order [43] and Heuristics for ordering gadgets [18], [43].

II. UTILITY MINING PROBLEM

Utility mining is one of the emerging data mining techniques. It is a challenging data mining task to discover high utility itemsets efficiently from a large data base. Identification of the itemsets with high utilities is called as Utility Mining. Utility may be defined as the measure of usefulness an item. The utility can be measured according to the user preferences, utility can be measured in terms of cost, profit or revenues or some other expressions. The utility measures can be changed based on user's interest and their requirements. In frequent pattern mining, the patterns are mined based on the frequency of their occurance which ignores the quantity and profits earned. These limitations of frequent pattern mining has motivated to start research on utility based mining , It allows a user to conveniently express their perspectives concerning the usefulness of itemsets as utility values. Based on the user's requirements , a threshold value is obtained which can be mininmum utitlity value and then if the utility of an itemset exceeds the minimum utility value which is threshold then that itemset will be high utility itemset . In utility based mining the term utility refers to the quantitative representation of user preference i.e. according to an itemsets utility value is the measurement of the importance of that itemset in the user's perspective. The goal of utility pattern mining is to discover high utility itemsets which results in large portions of the total utility.

III. RELATED WORKS

High utility pattern mining problem is closely related to frequent pattern mining, including constraint-based mining. In this section, we briefly review prior works both on frequent pattern mining and on utility mining, and discuss how our work connects to and differs from the prior works. Frequent Pattern mining -Frequent pattern mining was first proposed by Agrawal et al. [2], which is to discover all patterns whose supports are no less than a user-defined minimum support threshold. The field of high utility pattern mining is gaining more importance in the recent past due to the increase in data generation and the need to get un-identified patterns from the known data sets. Several research works have been proposed to meet the issues of high utility pattern mining. The various proposed algorithms for mining high utility patterns are described as follows.

Liu et al., [13] have proposed pseudo projection algorithm which is fundamentally different from those proposed in the past. This algorithm uses both array based and tree-based structures to represent projected transaction subsets and heuristically decides to build unfiltered pseudo projection to make a filtered copy according to features of the subsets. As a result both CPU time and memory efficiency are improved. This algorithm grows the frequent itemset tree by depth first search, where as breadth first search is used to build the upper portion of the tree if necessary. This algorithm is not only efficient on sparse and dense databases at all levels of support threshold and also highly scalable to very large databases. The disadvantage of this algorithm is, it only support minimum description code length with small number of patterns.

Han et al., [8] have proposed a frequent pattern growth (FP-Growth) algorithm for mining frequent pattern with constraints. In this work the frequent pattern tree (FP-tree) structure is an extended prefix tree structure developed for storing crucial information about frequent patterns. The pattern fragment growth mines the complete set constructs a highly compact FP-tree and applies a pattern growth method for database scans which is usually substantially smaller than the original database by which costly database scans are saved in the subsequent mining processes. The disadvantage of this algorithm is it reduces multi-pass candidate generation process in the first phase by discarding isolated items to reduce the number of candidates. This method shrink the database scanned in each pass which results in more computation time.

Liu et al., [13] have proposed a two-phase algorithm to find high utility itemsets. This algorithm efficiently prunes down the number of candidates and discovers the complete set of high utility itemsets. This algorithm works in two phases. In first phasees transaction-weighted downward closure property is applied to add high transaction weighted utilization. In second phase, over estimated low utility itemsets are filtered using an extra database scan. This algorithm requires fewer database scans,which results in less memory and computational cost for large databases and performs very well in terms of speed and memory cost on both synthetic and real database. The main disadvantage of this algorithm is, the insufficient frequent counts for repeated candidate itemset which can lose interesting patterns.

Erwin et al., [6] have proposed a transaction weighted utility (TWU) algorithm. This algorithm is based on compact utility pattern tree data structures. This work implements the parallel projection scheme to utilize the disk storage. This algorithm first identifies the TWU items from transaction database and the compressed utility pattern tree is constructed for mining complete set of high utility patterns. In this algorithm parallel projection is used to create subdivision for subsequently mining. This algorithm has anti-monotone property which is used to discover the pruning space. In this work the task of high utility itemset mining discovers all the utility which has utility higher than the user specified-utility. Generation of frequent graphs results in high in memory usage and low in accuracy.

Shankar et al., [17] have proposed a fast utility mining (FUM) algorithm that finds all high utility itemset within the given utility constraint threshold. It is faster and simpler than the original Utility Mining algorithm. This algorithm efficiently handles the redundant itemsets by checking redundancy and removing redundant itemsets which reduces the execution time of the algorithm. This algorithm provides accurate and efficient discovery of high utility itemset from the transactions in the database. This algorithm executes transaction datasets exceptionally faster when more itemset are identified as high utility itemset and when the number of distinct items in the database increases.

Ahmed et al., [1] have proposed a tree-based incremental high utility pattern mining (IHUPM) algorithm. In this algorithm a tree based structure called IHUP-Tree is used to maintain the information about itemsets and their utilities. It proposes three tree structures to perform incremental and interactive high utility pattern mining efficiently. This reduces the calculations when a minimum threshold is changed or a database is updated. The first tree structure is an incremental high utility pattern lexicographic tree (IHUPLTree) that is arranged according to an item's lexicographic order. It can capture the incremental data without any restructuring operation. The second tree structure is the incremental high utility pattern transaction frequency tree (IHUPTF-Tree) which is simple and easy to construct and handle. In this tree the items are arranged according to their transaction frequency. It does not require any restructuring operation even when the data base is incrementally updated. They have achieved the less memory consumption. The third tree structure is the incremental high utility pattern transaction weighted utilization tree (IHUPTWU-Tree) and this tree is based on the transaction weighted utility value of items in descending order. This algorithm takes insufficient memory usage and outperforms with earlier lexicographical approaches. Tseng et al., [18] have proposed an efficient algorithm called as utility pattern growth plus (UP-Growth+) which is an improved version of utility pattern growth (UP-Growth) mining algorithm. In this work the information of high utility itemset is maintained in a special data structure named utility pattern tree (UP-Tree) and the candidate itemsets are generated with one scans of the database.

Liu and Qu., [12] have proposed a high utility itemset miner (HUI-Miner) for high utility itemset mining. This algorithm uses a novel structure called utility-list which is used to store both the utility information about an itemset and the heuristic information for pruning the search space. This algorithm first creates an initial utility list for itemsets of recursively a utility list for each itemset of the length k using a pair of utility lists for itemset of the length k-1 for mining high utility itemset, each utility list for an itemset keeps the information of indicates transaction for all of transactions containing the itemset, utility values of the item set in the transactions, and the sum of utilities of the remaining items that can be included to super itemset of the itemset in the transactions. This algorithm first estimate the utilities of the itemsets and generate the candidate itemsets and then by scanning the database compute the exact utilities of the itemset to generate the high utility itemset. This algorithm mines the high utility itemset without generation of the candidates and the algorithm outperforms in terms of both running time and memory consumption.

Fournier-Viger et al., [7] have proposed an algorithm fast high utility miner (FHM) which extends the high utility itemset miner (HUI-Miner) algorithm. It is a depth-first search algorithm that relies on utility-lists to calculate the exact utility of itemsets. This algorithm consists of discovering frequent itemset in transactions. This work integrates a novel strategy named estimated utility co-occurrence pruning (EUCP) to reduce the number of joins operations when mining high utility itemset using the utility list data structure. The estimated utility co-occurrence pruning structure (EUCP) stores the transaction weighted utility of all itemsets. It built during the initial database scans. The memory footprint of the estimate utility co-occurrence pruning structure is small. This algorithm performs high utility itemset miner based on the analysis of item co-occurrences to reduce the number of join operations that need to be performed. An important limitation of this algorithm is it assumes that each item cannot appear more than once in each transaction and that all items have the same importance.

IV. PROPOSED SYSTEM

To provide the efficient solution to mine the large transactional datasets, recently improved methods presented propose two novel algorithms as well as a compact data structure for efficiently discovering high utility itemsets from transactional databases. Experimental results show that d2HUP and CAUL outperform other algorithms substantially in terms of execution time. But these algorithms further needs to be extend so that system with less memory will also able to handle large datasets efficiently.

In-order to address the above problems, we proposed a new algorithm d^2HUP (Direct discovery of high Utility patterns) for utility mining with item-set share frame work.

- This algorithm employs several techniques which includes exploring a regular set enumeration tree in reverse lexicographic order.
- A high utility pattern growth approach that directly discovers high utility patterns in single phase without generating high TWU patterns (candidates).
- The strength of the approach comes from pruning techniques which are base on tight upper bounds.
- A lookahead strategy to identify high utility patterns without recursive enumeration based on closure property and singleton property.
- This property enhances efficiency while dealing with dense data.
- A linear data CAUL (Chain Accurate Utility List) – It represents original utility information in raw data for each enumerated pattern. Which enables us to compute the utility and to estimate tight utility upper bounds efficiently
- And GPA (Gradual Pruning Approach) improves the performance while mining itemsets. The experimental results show that the number of candidates required by GPA is obviously less than that required.

SYSTEM DESIGN

The proposed enhanced high utility pattern mining has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility, transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed system design is shown below. The proposed system builds transaction set (TS) by scanning the database D and build the external utility table(XUT) to compute s(i), u(i), uBitem(i), and uBfpe(i) for each item i. The proposed system first pre processes the datasets then applies GPA pruning and then starts searching high utility patterns from the construction of the utility pattern tree by calling the depth first search (DFS) approach. For the each node N currently being visited, DFS prints pattern (N) as a high utility pattern if its utility is no less than the threshold which makes the set W of relevant items.

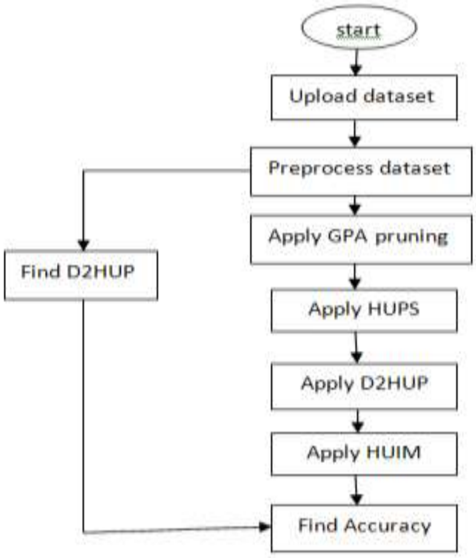

Figure: 4.1 Implementation process.

V. EXPERIMENTS ON DENSE DATASETS

In this work five real-world datasets are used for evaluation. The first dataset is T10I6D1M which contains the items are selected such as milk, bread, butter, jam. The data from used for 1-itemset, 2-itemset, 3-itemset.The second one is Chess which is a dense dataset used for transaction items. The third dataset is Chain store generated the itemsets. The fourth one is T20I6DIM in mixed dataset, it increases with the transactions. The last dataset is foodmart which contains real utility values generated from high utility values. For T10I6D1M, Chess, Chain store and T20I6DIM, food mart divide each part generated itemsets in 10,000 itemsare selected for each transaction. The transaction database that contains items which find effective high utility patterns for improving the performance of mining itemsets. The method provides scalability and efficiency for mining utility itemsets along execution time and memory space on database transactions. The first column is the name of a dataset, the second (|t|) is the average and maximum length of transactions, the third (|I|) is the number of distinct items, the fourth (|D|) is the number of transactions, and the fifth (Type) is a rough categorization based on the number of high utility patterns to be mined, partially depending on the minimum utility threshold. The detailed of the five datasets which include transaction, distinct items, number of transaction and type are shown in Table 4.1.The proposed system eliminates the local unfavorable items and reduces local node utility. The proposed enhanced high utility pattern approach has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed algorithm has been executed on same minimum utility value as per the datasets to generate itemsets. Experiments are performed to evaluate the performance of the proposed EHUPM with the existing d2HUP algorithm based on five datasets.

Data Set	Transaction $\|t\|$	Distinct Items $\|I\|$	Number of Transaction $\|D\|$	Type
T10I6D1M	10:33	1000	933,493	Mixed
Chess	37:37	76	3,197	Dense
Chain-Store	7.2:170	46,086	1,112,949	Sparse
T2016DIM	20:49	1,000	999,287	Mixed
Foodmart	4.8:27	1,559	34,015	Dense

TABLE 1

Characteristics of datasets

Performance analysis and results

Experiments are performed to evaluate the performance of the proposed EHUPA algorithm based on the five datasets. The performance of EHUPA algorithm has been compared with existing d2HUP algorithm based on the metrics such as memory usage and running time for mining high utility itemsets.

Memory usage for high utility pattern

The proposed EHUPA is compared with existing d2HUP method for memory usage for mining high utility itemsets and the performance graph is shown in Figure 4.1. In the graph, x-axis represents the datasets and y-axis represents the memory space. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less memory space usage than the existing d2HUPmethod.

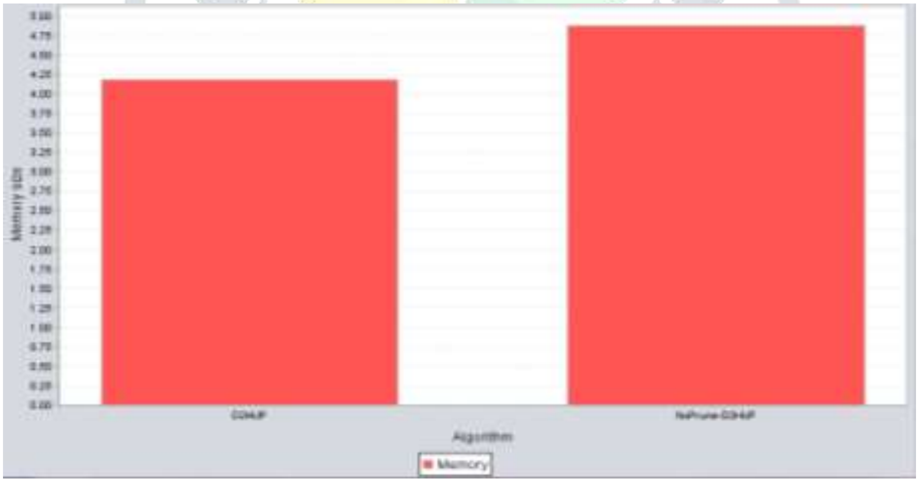

Figure 5.1 Memory Graph

Running time for high utility pattern

The proposed EHUPA is compared with existing d2HUP method to mine high utility itemsets for running time and the performance graph is shown in Figure 4.2. In the graph, x-axis represents the datasets and y-axis represents the running time of utility itemsets. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less running time than the existing d2HUP Method.

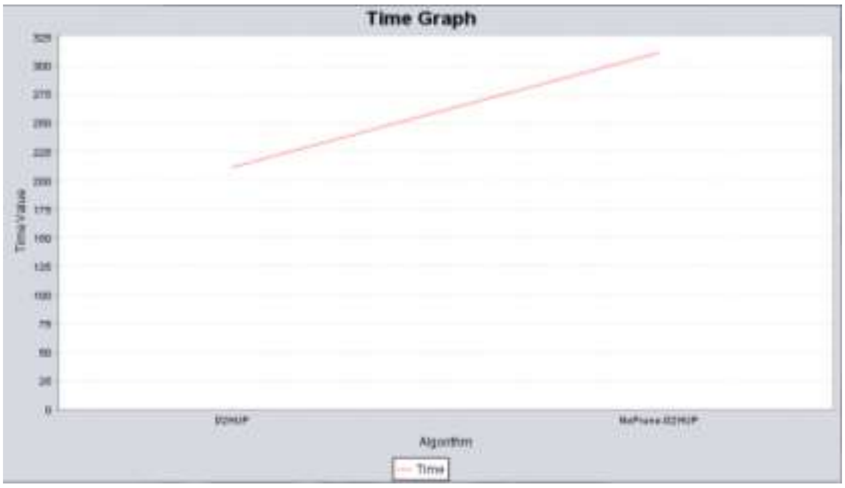

Figure: 5.2 Time Graph

VI. CONCLUSION AND FUTURE WORK

This paper proposes the algorithm, d2HUP, for data mining with the itemset proportion framework, which reveals patterns without candidate technology. The high utility pattern mining with the itemset share framework is more challenging than the other categories of utility mining such as weighted itemset mining, association rule mining and frequent pattern mining. During the knowledge discovery process, utility based measures are used to find the unidentified patterns to improve the mining efficiency. In this research paper an enhanced high utility pattern approach (EHUPA) has been proposed to mine high utility itemsets. To Speed up process we add GPA pruning techniques. The experimental results show that the proposed system provides better performance than the existing direct discovery high utility pattern algorithm in terms of memory usage and execution time for mining high utility itemsets. In future we will be developing an algorithm for Mining High Utility Itemsets from Distributed Databases. Most of research on high utility itemset focuses on static databases. 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.

REFERENCES

[1] R. Agawam, C. Aggarwal, and V. Prasad, “Depth first generation of long patterns,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2000, pp. 108–118.

[2] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1993, pp. 207–216.

[3] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. 20th Int. Conf. Very Large Databases, 1994,pp. 487–499.

[4] 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. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[5] R. Bayardo and R. Agrawal, “Mining the most interesting rules,”in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining,1999, pp. 145–154.

[6] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, “ExAnte: A preprocessing method for frequent-pattern mining,” IEEE Intell.Syst., vol. 20, no. 3, pp. 25–31, May/Jun. 2005.

[7] F. Bonchi and B. Goethals, “FP-Bonsai: The art of growing and pruning small FP-trees,” in Proc. 8th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2004, pp. 155–160.

[8] F. Bonchi and C. Lucchese, “Extending the state-of-the-art of constraint-based pattern discovery,” Data Knowl. Eng., vol. 60,no. 2, pp. 377–399, 2007.

[9] C. Bucila, J. Gehrke, D. Kifer, and W. M. White, “Dualminer: A dual-pruning algorithm for itemsets with constraints,” Data Mining Knowl. Discovery, vol. 7, no. 3, pp. 241–272, 2003.

[10] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, “Mining association rules with weighted items,” in Proc. Int. Database Eng.Appl. Symp., 1998, pp. 68–77.

[11] R. Chan, Q. Yang, and Y. Shen, “Mining high utility itemsets,” in Proc. Int. Conf. Data Mining, 2003, pp. 19–26.

[12] S. Dawar and V. Goyal, “UP-Hist tree: An efficient data structure for mining high utility patterns from transaction databases,” in Proc. 19th Int. Database Eng. Appl. Symp., 2015, pp. 56–61.

[13] T. De Bie, “Maximum entropy models and subjective interestingness: An application to tiles in binary databases,” Data MiningKnowl. Discovery, vol. 23, no. 3, pp. 407–446, 2011.

[14] L. De Raedt, T. Guns, and S. Nijssen, “Constraint programming for itemset mining,” in Proc. ACM SIGKDD, 2008, pp. 204–212.

[15] A. Erwin, R. P. Gopalan, and N. R. Achuthan, “Efficient mining of high utility itemsets from large datasets,” in Proc. 12th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2008, pp. 554–561.

[16] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, “FHM:Faster high-utility itemset mining using estimated utility cooccurrence pruning,” in Proc. 21st Int. Symp. Found. Intell. Syst.,2014, pp. 83–92.

[17] L. Geng and H. J. Hamilton, “Interestingness measures for data mining: A survey,” ACM Comput. Surveys, vol. 38, no. 3, p. 9, 2006.

[18] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2000, pp. 1–12.

[19] R. J. Hilderman, C. L. Carter, H. J. Hamilton, and N. Cercone,“Mining market basket data using share measures and characterized itemsets,” in Proc. PAKDD, 1998, pp. 72–86.

[20] R. J. Hilderman and H. J. Hamilton, “Measuring the interestingness of discovered knowledge: A principled approach,” Intell.Data Anal., vol. 7, no. 4, pp. 347–382, 2003.

[21] M. Holsheimer, M. Kersten, H. Mannila, and H. Toivonen, “A perspective on databases and data mining,” in Proc. 1st Int. Conf.Knowl. Discovery Data Mining, 1995, pp. 150–155.

[22] S. Krishnamoorthy, “Pruning strategies for mining high utility itemsets,” Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.

[23] G.-C. Lan, T.-P. Hong, and V. S. Tseng, “An efficient projectionbased indexing approach for mining high utility itemsets,” Knowl.Inf.Syst., vol. 38, no. 1, pp. 85–107, 2014.

[24] Y.-C. Li, J.-S.Yeh, and C.-C. Chang, “Isolated items discarding strategy for discovering high utility itemsets,” Data Knowl.Eng.,vol. 64, no. 1, pp. 198–217, 2008.

[25] T. Y. Lin, Y. Y. Yao, and E. Louie, “Value added association rules,”in Proc. 6th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining,2002, pp. 328–333.

[26] J. Liu, Y. Pan, K. Wang, and J. Han, “Mining frequent item sets byopportunistic projection,” in Proc. 8th ACM SIGKDD Int. Conf.Knowl. Discovery Data Mining, 2002, pp. 229–238.

[27] J. Liu, K. Wang, and B. Fung, “Direct discovery of high utility itemsets without candidate generation,” in Proc. IEEE 12th Int. Conf. Data Mining, 2012, pp. 984–989.

[28] M. Liu and J. Qu, “Mining high utility itemsets without candidate generation,” in Proc. ACMConf. Inf. Knowl.Manage., 2012, pp. 55–64.

[29] Y. Liu, W. Liao, and A. Choudhary, “A fast high utility itemsets mining algorithm,” in Proc. Utility-Based Data Mining Workshop SIGKDD, 2005, pp. 253–262.

[30] S. Lu, H. Hu, and F. Li, “Mining weighted association rules,”Intell.Data Anal., vol. 5, no. 3, pp. 211–225, 2001.