

# Survey on Mining High Utility Patterns in One Phase

Harshita Taran  
M.Tech Student,  
Computer Science and Engineering  
Kavikulguru Institute of Technology And Science  
Rmatek(M.H.),India.

Shilpa Ghode  
Assistant Professor,  
Computer Science and Engineering  
Kavikulguru Institute of Technology And Science  
Rmatek(M.H.), India.

**Abstract**—This Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Conventional data mining techniques have focused largely on finding the items that are more frequent in the transaction databases, which is also called frequent itemset mining. These data mining techniques were based on support confidence model. Itemsets which appear more frequently in the database must be of more meaning to the user from the business point of view. High Utility Itemset Mining that discovers the itemsets considering not only the frequency of the itemset but also utility associated with the itemset. Every itemset have a value like quantity, profit and other user's interest. This value associated with every item in a database is called the utility of that itemset. Those itemsets having utility values greater than given threshold are called high utility itemsets. Prior works on this problem all employ a two-phase, candidate generation approach with one exception that is however inefficient and not scalable with large databases. The two-phase approach suffers from scalability issue due to the huge number of candidates. In this paper we present survey on a novel algorithm that finds high utility patterns in a single phase without generating candidates. The novelties lie in a high utility pattern growth approach, a lookahead strategy, and a linear data structure. Concretely, pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding. Look ahead strategy is to identify high utility patterns without enumeration by a closure property and a singleton property. The linear data structure is to compute a tight bound for powerful pruning and to directly identify high utility patterns in an efficient and scalable way, which targets the root cause with prior algorithms.

**Keywords**— Data Mining, Frequent Itemset Mining, Utility Mining, High Utility Itemset Mining, D2HUP

## I. INTRODUCTION

Data mining and knowledge discovery from data bases has received much attention in recent years. Data mining, the extraction of hidden predictive information from largedatabases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, previously unknown and potentially useful patterns in data. These patterns are used to make predictions or classifications about new data, explain existing data, summarize the contents of a large database to support decision making and provide graphical data visualization to

aid humans in discovering deeper patterns. Data mining is the process of revealing nontrivial, previously unknown and potentially useful information from large databases. Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining. Among them, frequent pattern mining is a fundamental research topic that has been applied to different kinds of databases, such as transactional databases, streaming databases, and time series databases, and various application domains, such as bioinformatics, Web click-stream analysis and mobile environments.

Subfield of data mining that is called pattern mining. Pattern mining consists of using/developing data mining algorithms to discover interesting, unexpected and useful patterns in databases. Pattern mining algorithms can be applied on various types of data such as transaction databases, sequence databases, streams, strings, spatial data, graphs, etc. Pattern mining algorithms can be designed to discover various types of patterns: subgraphs, associations, indirect associations, trends, periodic patterns, sequentialrules, lattices, high-utility patterns, etc. frequent pattern mining has become an important data mining task and a focused theme in data mining Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data indexing, classification research.The problem of high-utility itemset mining is an extension of the problem of frequent pattern mining. frequent pattern mining has become an important data mining task and a focused theme in data mining Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data indexing, classification research.The problem of high-utility itemset mining is an extension of the problem of frequent pattern mining.utility mining emerges as an important topic in data mining field. Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance or profitability of an item to users. Utility of items in a transaction database consists of two aspects:

- The importance of distinct items, which is called external utility, and
- Importance of items in transactions, which is called internal utility.

A high utility pattern growth approach is proposed, which we argue is one without candidate generation because while the two-phase, candidate generation approach employed by prior algorithms first generates high TWU patterns (candidates) with TWU being an interim, anti-monotone measure and then identifies high utility patterns from high TWU patterns, our approach directly discovers high utility patterns in a single phase without generating high TWU patterns (candidates). The strength of our approach comes from powerful pruning techniques based on tight upper bounds on utilities. A lookahead strategy is incorporated with our approach, which tries to identify high utility patterns earlier without recursive enumeration. Such a strategy is based on a closure property and a singleton property, and enhances the efficiency in dealing with dense data. A linear data structure, CAUL, is proposed to represent original utility information in raw data, which targets the root cause with prior algorithms, that is, all employ a data structure to maintain the utility estimates instead of the original utility information, and thus can only determine the candidacy of a pattern but not the actual utility of the pattern in their first phase.

## II. LITERATURE SURVEY

### A. Frequent Pattern Mining

Frequent pattern mining was first proposed by Agrawal et al (1993), which is to discover all patterns whose supports are no less than a user-defined minimum support threshold [2]. Frequent pattern mining employs the anti-monotonicity property: support of a superset of a pattern is no more than the support of the pattern. Algorithms for mining frequent patterns as well as algorithms for mining high utility patterns fall into three categories, breadth-first search, depth first search, and hybrid search.

Apriori by Agrawal and Srikant(1994) is a very famous breadth-first algorithm for mining frequent patterns, which scans the disk-resident database as many times as the maximum length of frequent patterns[3]. Consider the problem of discovering association rules between items in a large database of sales transactions. Two new algorithms for solving this problem that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. Also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

FP-growth by Han et al. (2000) is a well-known depth-first algorithm, which compresses the database by FP-trees in main memory [18]. Mining frequent patterns in transaction databases, time-series databases, and many other kinds of databases has been studied popularly in data mining research. Most of the previous studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist prolific patterns and/or long patterns.

Propose a novel frequent pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and

develop an efficient FP-tree-based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth.

Efficiency of mining is achieved with three techniques: (1) a large database is compressed into a highly condensed, much smaller data structure, which avoids costly, repeated database scans, (2) our FP-tree-based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining connected patterns in conditional databases, which dramatically reduces the search space. FP-growth method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm and also faster than some recently reported new frequent pattern mining methods.

Eclat by Zaki (2000) is a famous hybrid algorithm. It keeps a database or a database partition [34] in memory by a vertical tid-list layout [21] and can work in either depth-first or breadth-first manner.

D2HUP adopts a depth-first strategy since breadth first search is typically more memory-intensive and more likely to exhaust main memory and thus slower. Concretely, our algorithm depth-first searches a reverse set enumeration tree, which can be thought of as exploring a regular set enumeration tree [1], [18], [33] right-to-left in a reverse lexicographic order [43].

While Eclat (2000) also explores such an order, our algorithm is the first fully exploiting the benefit in mining high utility patterns[43]. Association rule discovery has emerged as an important problem in knowledge discovery and data mining. The association mining task consists of identifying the frequent itemsets and then, forming conditional implication rules among them. Efficient algorithms for the discovery of frequent itemsets which forms the compute intensive phase of the task. The algorithms utilize the structural properties of frequent itemsets to facilitate fast discovery. The items are organized into a subset lattice search space, which is decomposed into small independent chunks or sublattices, which can be solved in memory. Efficient lattice traversal techniques are presented which quickly identify all the long frequent itemsets and their subsets if required. Effect of using different database layout schemes combined with the proposed decomposition and traversal techniques. Experimentally compare the new algorithms against the previous approaches, obtaining improvements of more than an order of magnitude for test databases.

### B. Constraint-Based Mining

Constraint-based mining is a milestone in evolving from frequent pattern mining to utility mining. Works on this area mainly focus on how to push constraints into frequent pattern mining algorithms.

Pei et al. (2004) discussed constraints that are similar to (normalized) weighted supports [10], Mining of association rules has been found useful in many applications. Many previous works focused on the basket (binary) association rules, which is in the form of The transactions show that there are many customers who purchase product A will purchase

the product B". All the products are treated uniformly, and all the rules are mined based on the counts of the products. However, in the social science research, the analysts may want to mine the rules based on the importance of the products, items or attributes. For example, total income attribute is more interesting than the height of a person in a household. Based on this, generalize this to the case where the items are given weights to reflect the importance to the users. As the downward closure property of the support measure in the mining of association rules no longer exists, previous algorithms cannot be applied. In this thesis, we make use of a metric, called support bounds, in the mining of binary and quantitative (fuzzy) association rules. Furthermore, introduce the simple sample method and the data maintenance method, based on the statistical approach, to mine the rules. Experiments show that by using

the weights, we can prune the items without any interest at an earlier step, and hence saving time in mining of the association rules. and first observed an interesting property, called convertible anti-monotonicity, by arranging the items in weight-descending order [32]. The authors demonstrated how to push them into the FP-growth algorithm [18].

Bucila et al. (2003) considered mining patterns that satisfy a conjunction of anti-monotone and monotone constraints, and proposed an algorithm, DualMiner, that efficiently prunes its search space using both anti-monotone and monotone constraints [9]. Bonchi et al. [6] introduced the ExAnte property which states that any transaction that does not satisfy the given monotone constraint can be removed from the input database, and integrated the property with Apriori-style algorithms.

Bonchi and Goethals (2004) applied the ExAnte property with the FP-growth algorithm [7]. integrate the proposed ExAnte data reduction technique within the FP-growth algorithm. Together, result in a very efficient frequent itemset mining algorithm that effectively exploits monotone constraints.

Bonchi and Lucchese (2007) generalized the data reduction technique to a unified framework [8]. The constraint-based pattern discovery paradigm was introduced with the aim of providing to the user a tool to drive the discovery process towards potentially interesting patterns, with the positive side effect of achieving a more efficient computation. Review and extend the state-of-the-art of the constraints that can be pushed in a frequent pattern computation. Novel data reduction techniques which are able to exploit convertible anti-monotone constraints (e.g., constraints on average or median) as well as tougher constraints (e.g., constraints on variance or standard deviation). A thorough experimental study is performed and it confirms that our framework outperforms previous algorithms for convertible constraints, and exploit the tougher ones with the same effectiveness. Finally, highlight that the main advantage of our approach, i.e., pushing constraints by means of data reduction in a level-wise framework, is that different properties of different constraints can be exploited all together, and the total benefit is always greater than the sum of the individual benefits. This consideration leads to the definition of a general Apriori-like algorithm which is able to exploit all possible kinds of constraints studied so far.

De Raedt et al. (2008) investigated how standard constraint programming techniques can be applied to constraint-based mining problems with constraints that are monotone, anti-monotone, and convertible [14]. Bayardo and Agrawal (1999), and Morishita and Sese (2000) proposed techniques of pruning based on upper bounds when the constraint is neither monotone, anti-monotone, nor convertible [5][31]. Project also employs such a standard technique. contribution is to develop tight upper bounds on the utility.

### C. Some Categories of Utility Mining

Interestingness measures can be classified as objective measures, subjective measures, and semantic measures [17]. Objective measures [20], [37], such as support or confidence, are based only on data; Subjective measures [13], [36], such as unexpectedness or novelty, take into account the user's domain knowledge; Semantic measures [41], also known as utilities, consider the data as well as the user's expectation.

Hilderman et al. (1998) proposed the itemset share framework that takes into account the weights both on attributes, for example, the price of a product, and on attribute-value pairs, for example, the quantity of a product in a shopping basket [19]. Then, support and confidence measures can be generalized based on count-shares as well as on amount-shares.

Yao et al. (2004) proposed a utility measure that instantiates this framework. Falls into three category [39], [40]. Sequential pattern mining is an important data mining problem with broad applications. It is challenging since one may need to examine a combinatorially explosive number of possible subsequence patterns. Most of the previously developed sequential pattern mining methods follow the methodology of A priori which may substantially reduce the number of combinations to be examined. However, Apriori still encounters problems when a sequence database is large and/or when sequential patterns to be mined are numerous and long. A novel sequential pattern mining method, called Prefixspan (i.e., Prefix-projected Sequential Pattern Mining), which explores prefix projection in sequential pattern mining. Prefixspan mines the complete set of patterns but greatly reduces the efforts of candidate subsequence generation. Moreover, prefix-projection substantially reduces the size of projected databases and leads to efficient processing. performance study shows that Prefixspan outperforms both the Apriori-based GSP algorithm and another recently proposed method; FreeSpan, in mining large sequence data bases.

Cai et al. (1998) proposed weighted itemset mining [10]. Lin et al. (2002) proposed value added association mining [25]. Both works assign each item a weight representing its importance, which results in (normalized) weighted supports, also known as horizontal weights. Lu et al. (2001) [30] proposed to assign a weight to each transaction representing the significance of the transaction, also known as vertical weights.

Shen et al. (2002) and Chan et al. (2003) proposed objective-oriented utility-based association mining that explicitly models associations of a specific form "Pattern ! Objective" where, Pattern is a set of non-objective-attribute value pairs, and Objective is a logic expression asserting objective-attributes with each objective attribute value satisfying

(violating) Objective assigned a positive (negative) utility[35] [11].

#### D. Algorithms with the Itemset Share Framework

As the utility measure with the itemset share framework is neither anti-monotone, monotone, nor convertible, most prior algorithms resort to an interim measure, (TWU), proposed by Liu et al(2005) and adopt a two-phase, candidate generation approach[29]. Transaction weighted utilization of a pattern is the sum of the transaction utilities of all the transactions containing the pattern.

TWU is anti-monotone. TWU or its variants is employed by most prior algorithms, which first invoke either Apriori [3] or FP-growth [18] to find high TWU patterns (candidates), and then scan the raw data once more to identify high utility patterns from the candidates. An exception is that Yao et al. [40], [41]

Liu et al. (2005) proposed the anti-monotonicity property with TWU, based on which they developed the Two Phase algorithm by adapting Apriori [3], [29].

Li et al. (2008) proposed an isolated items discarding strategy (IIDS) [24]. An isolated item is one that is not contained in any length-k candidate, and hence it will not occur in any candidate with a length greater than k. Any multi-pass, level-wise algorithm can employ IIDS to reduce the number of redundant candidates.

Lan et al. (2014) proposed an projection-based algorithm, based on the TWU model [29], that speeds up the execution by an indexing mechanism [23] efficient mining approach which adopts an indexing mechanism to speed up the execution efficiency and reduce the memory requirement in the utility mining process. Besides, a pruning strategy is designed to reduce the number of unpromising itemsets for mining. The experimental results also show the performance of the proposed approach is better than that of traditional two-phase utility mining algorithm under different parameters,.

Erwin et al.(2008) proposed the CTU-PROL algorithm for mining high utility patterns that integrates the TWU anti-monotonicity property and pattern growth approach [18] in first phase, which is facilitated by a compact utility pattern tree structure, CUP-tree.

Ahmed et al. (2009) proposed a tree-based algorithm, IHUPTWU, for mining high utility patterns, which uses an IHUPTWU-tree to maintain the TWU information of transactions, and mines the set of candidates of high utility patterns by adapting FP-growth [18]. Notice that CTU-PROL [15] and IHUPTWU produce the same amount of candidates in the first phase [4].

Tseng et al.(2013) proposed the latest, FP-growth based algorithm, UP-Growth, which uses an UP-tree to maintain the revised TWU information, improves the TWU property based pruning, and thus generates fewer candidates in the first phase[34], [24].

Yun et al. (2014) and Dawar and Goyal (2015) improved UPGrowth [38] by pruning more candidates, while the inherent issue of the two-phase approach remains. preliminary work [27] and Liu and Qu [28], simultaneously and independently, proposed to mine high utility patterns without candidate generation[42][12].

The HUIMiner algorithm by Liu and Qu (2012) employs a vertical data structure to represent utility information, which employs inefficient join operations and is also not scalable[28]. HUIMiner is even less efficient than an improved version of UP-Growth [38] when mining large databases. Therefore, scalability and efficiency remains to be a challenge with HUIMiner [28]. work addresses such a challenge with large databases.

Fournier-Viger et al. (2014) improved HUIMiner [28] by, pre-computing the TWUs of pairs of items to reduce the number of join operations [16].

Krishnamoorthy (2015) improved HUIMiner [28] by a partition strategy [22]. Their improvement is within a factor of 2 to 6, while D2HUP algorithm is up to 45 times faster than HUIMiner [28] on the same databases.

### III. CONCLUSION

The Frequent itemset mining is based on the principle that the itemsets which appear more frequently in the transaction databases are of more importance to the user. However in reality the benefit of frequent itemset mining by considering only frequency of itemset is challenged in many research areas such as retail, marketing etc. It has been seen that in many real application domains that the itemsets that contribute the most are not necessarily the frequent itemsets. Utility mining is an era of research which tries to bridge this gap by using item utilities as an analytical measurement of the importance of that item in the user's point of view. Utility mining is a comparatively new area of research and most of the literature work is focused towards reducing the search space while searching for the high utility itemsets. New algorithm, d2HUP, for utility mining with the itemset share framework, which finds high utility patterns without candidate generation. include: 1) A linear data structure, CAUL, is proposed, which targets the root cause of the two phase,candidate generation approach adopted by prior algorithms, that is, their data structures cannot keep the original utility information. 2) A high utility pattern growth approach is presented, which integrates a pattern enumeration strategy, pruning by utility upper bounding, and CAUL. This basic approach outperforms prior algorithms strikingly. 3) approach is enhanced significantly by the lookahead strategy that identifies high utility patterns without enumeration.

In the future, we will work on implementation of high utility pattern mining algorithm which directly discover high utility patterns.

### REFERENCES

- [1] R. Agarwal, C. Aggarwal, and V. Prasad, "Depth first generation of long patterns," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 108–118,2000.
- [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, pp. 207–216, 1993.
- [3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Databases,pp. 487–499 ,1994.
- [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, pp. 145–154,1999.

- [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/June 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*, pp. 155–160, 2004.
- [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.*, pp. 68–77, 1998.
- [11] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," in *Proc. Int. Conf. Data Mining*, pp. 19–26, 2003.
- [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.*, pp. 56–61, 2015.
- [13] T. De Bie, "Maximum entropy models and subjective interestingness: An application to tiles in binary databases," *Data Mining Knowl. 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*, pp. 204–212, 2002.
- [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*, pp. 554–561, 2002.
- [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.*, pp. 83–92, 2014.
- [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*, pp. 72–86, 1998.
- [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*, pp. 150–155, 1995.
- [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 projection-based 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*, pp. 328–333, 2002.
- [26] J. Liu, Y. Pan, K. Wang, and J. Han, "Mining frequent item sets by opportunistic projection," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp. 229–238, 2002.
- [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*, pp. 984–989, 2012.
- [28] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, pp. 55–64, 2012.
- [29] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in *Proc. Utility-Based Data Mining Workshop SIGKDD*, pp. 253–262, 2005.
- [30] S. Lu, H. Hu, and F. Li, "Mining weighted association rules," *Intell. Data Anal.*, vol. 5, no. 3, pp. 211–225, 2001.
- [31] S. Morishita and J. Sese, "Traversing itemset lattice with statistical metric pruning," in *Proc. 19th ACM Symp. Principles Database Syst.*, pp. 226–236, 2002.
- [32] J. Pei, J. Han, and V. Lakshmanan, "Pushing convertible constraints in frequent itemset mining," *Data Mining Knowl. Discovery*, vol. 8, no. 3, pp. 227–252, 2004.
- [33] J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. 17th Int. Conf. Data Eng.*, pp. 215–224, 2001.
- [34] A. Savasere, E. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," in *Proc. 21st Int. Conf. Very Large Databases*, pp. 432–444, 1995.
- [35] Y. Shen, Q. Yang, and Z. Zhang, "Objective-oriented utility-based association mining," in *Proc. IEEE Int. Conf. Data Mining*, pp. 426–433, 2002.
- [36] A. Silberschatz and A. Tuzhilin, "On subjective measures of interestingness in knowledge discovery," in *Proc. ACM 1st Int. Conf. Knowl. Discovery Data Mining*, pp. 275–281, 1995.
- [37] P. N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," *Inf. Syst.*, vol. 29, no. 4, pp. 293–313, 2004.
- [38] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1772–1786, Aug. 2013.
- [39] H. Yao and H. J. Hamilton, "Mining itemset utilities from transaction databases," *Data Knowl. Eng.*, vol. 59, no. 3, pp. 603–626, 2006.
- [40] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proc. SIAM Int. Conf. Data Mining*, pp. 482–486, 2004.
- [41] H. Yao, H. J. Hamilton, and L. Geng, "A unified framework for utility-based measures for mining itemsets," in *Proc. ACM SIGKDD 2nd Workshop Utility-Based Data Mining*, 2006, pp. 28–37.
- [42] U. Yun, H. Ryang, and K. H. Ryu, "High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3861–3878, 2014.
- [43] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 3, pp. 372–390, May/June 2000.
- [44] M. J. Zaki and C. Hsiao, "Efficient algorithms for mining closed itemsets and their lattice structure," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 462–478, Apr. 2005.