International Journal of Soft Computing Applications
ISSN: 1453-2277 Issue 4 (2009), pp.81-95

http://www.eurojournals.com/IJSCA.htm

Utility Sentient Frequent Itemset Mining and Association Rule Mining: A Literature Survey and Comparative Study

S.Shankar
Senior Lecturer (IT), Sri Krishna College of Engineering and Technology
Coimbatore, India
E-mail: shanx_80@yahoo.co.in

T.Purusothaman
Assistant Professor (CSE), Government College of Technology
Coimbatore, India
E-mail: purushgct@yahoo.com

Abstract

It is a well accepted verity that the process of data mining produces numerous patterns from the given data. The most significant tasks in data mining are the process of discovering frequent itemsets and association rules. Numerous efficient algorithms are available in the literature for mining frequent itemsets and association rules. Incorporating utility considerations in data mining tasks is gaining popularity in recent years. Certain association rules enhance the business value and the data mining community has acknowledged the mining of these rules of interest since a long time. Several business applications have been found to benefit from the discovery of frequent itemsets and association rules from transaction databases. A comprehensive survey and study of various methods in existence for frequent itemset mining, association rule mining with utility considerations have been presented in this paper.

Keywords: Data Mining, Frequent Itemset Mining, Association Rule Mining, Utility, High Utility Itemset Mining.

1. Introduction

Recent developments in information science have caused large scale data digitalization, swelling up digital databases and data warehouses. As a result, it is necessary to develop mechanisms that effectively handle large quantities of sequential data and expeditiously extract useful knowledge on the basis of data [8]. Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Data mining, also known as Knowledge Discovery in Databases, has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [2]. Data mining is used to extract structured knowledge automatically from large data sets [48]. The information that is 'mined' is expressed as a model of the semantic structure of the dataset, where in the prediction or classification of the obtained data is facilitated with the aid of the model [26].

Descriptive mining and Predictive mining are the two categories of data mining tasks. The descriptive mining refers to the method in which the essential characteristics or general properties of the data in the database are depicted. The descriptive mining techniques involve tasks like Clustering,

Association and Sequential mining. The method of predictive mining deduces patterns from the data such that predictions can be made. The predictive mining techniques involve tasks like Classification, Regression and Deviation detection. Mining Frequent Itemsets from transaction databases is a fundamental task for several forms of knowledge discovery such as association rules, sequential patterns, and classification [33]. The subsets frequently occurring in a collection of sets of items are known as the frequent itemsets. Frequent itemsets are typically used to generate association rules. The objective of Frequent Item set Mining is the identification of items that co-occur above a user given value of frequency, in the transaction database [45].

One of the popular descriptive data mining techniques is Association rule mining (ARM) [27], owing to its extensive use in marketing and retail communities in addition to many other diverse fields. Mining association rules is particularly useful for discovering relationships among items from large databases [34]. The "market-basket analysis" which performs a study on the habits of customers [3] is the source of motivation behind ARM. The extraction of interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories is the main objective of ARM [40]. As the target of discovery is not pre-determined, it is possible to identify all association rules that exist in the database. This feature of the association rules can be said as its major strength. The development of marketing and placement strategies in addition to the preparation of logistics for inventory management can be greatly assisted by the discovery of association rules.

The alignment of the data mining process and algorithms with the extensive economic objectives of the tasks supported by data mining is essential so as to permit the additional impact of data mining on business applications. The ultimate economic utility obtained as the outcome of the data mining product has the impact of all the diverse stages of the data mining processes. It is important to consider the economic utility of acquiring data, extracting a model, and applying the acquired knowledge [42]. The evaluation of the decisions made on the basis of the learned knowledge is influenced by the economic utility. The economic measures, for example, profitability and return on investment have replaced the simple assessment measures such as predictive accuracy.

A new research area known as utility-based data mining is concerned with all types of utility factors in data mining processes [37, 42, 49, 50]. The aim of utility-based data mining is to integrate utility considerations in both predictive and descriptive data mining tasks. One of the research areas of utility based descriptive data mining is high utility item set mining. The high utility item set mining mainly contributes to the total utility by the identification of item sets. The identification of all item sets that offer utility greater or equal to the user specified threshold [50] is the objective of high utility item set mining. The high utility item set mining uses subjectively defined utility in place of statistics-based support measure, which upgrades the standard frequent item set mining framework.

1.1. Frequent Itemset Mining

The task of frequent itemset mining was first introduced by Agrawal et al. [3] in 1993. A frequent itemset is a set of items that appears at least in a pre-specified number of transactions. Frequent itemsets are typically used to generate association rules. The task of frequent itemset mining is defined as follows:

Let I be a set of items. A set $X = \{i_i,....,i_k\} \subseteq I$ is called an itemset, or a k-itemset, if it contains k items. A transaction over I is a couple T = (tid, I) where tid is the transaction identifier and I is an itemset. A transaction T = (tid, I) is said to support an itemset $X \subseteq I$, if $X \subseteq I$ .A transaction database D over I is a set of transactions over I. The support of an itemset X in D is the number of transactions in D that supports X:

$$Support(X, D) = \left|\{tid \mid (tid, I) \in D, X \subseteq I\}\right| \quad (1)$$

The frequency of an itemset X in D is the probability of X occurring in a transaction $T \in D$:

$$Frequency(X, D) = P(X) = \frac{Support(X, D)}{|D|} \quad (2)$$

Note that |D| = support ({}, D). An itemset is called frequent if its support is no less than a given absolute minimal support threshold $\sigma_{abs,}$ with $0 \leq \sigma_{abs} \leq |D|$. The frequent itemsets discovered does not reflect the impact of any other factor except frequency of the presence or absence of an item.

1.2. Association Rule Mining

Since its introduction in 1993 by Agrawal et al. [3], the task of association rule mining has received a great deal of attention. Today the mining of such rules is still one of the most popular pattern-discovery methods in Knowledge Discovery and Data mining (KDD) [3]. Association rule mining [25] is a popular data mining technique because of its wide application in marketing and retail communities as well as other more diverse fields [55]. Association rule mining is a method of finding relationships of the form X→Y amongst itemsets that occur together in a database where X and Y are disjoint itemsets [48].Support and confidence measures serve as the basis for customary techniques in association rule mining. The support and confidence are predefined by users to drop the rules that are not so interesting or useful. The association rule indicates that the transactions that contain X tend to also contain Y. Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain purchasing of this item [40]. The task of mining association rules is defined as follows:

Let IS =$\{i_1,i_2,i_3,\ldots,i_m\}$ a set of items and TDI =$\{t_1,t_2,t_3, \ldots,t_n\}$be a set of transaction data items, where t_i = $\{IS_{i1}, IS_{i2,} IS_{i3, \ldots,} IS_{ip}\}$, $p \leq m$ and $IS_{ij} \in IS$, if $X \subseteq I$ with k = |X| is called a k-item set or simply an itemset. An expression, where X, Y are itemsets and $X \cap Y = \phi$. holds is called an association rule X → Y.

The measure of number of transactions T supporting an item set X with respect to TDI is termed as the Support of an itemset.

$$Support(X) = \left|\{T \in TDI \mid X \subseteq T\}/TDI\right| \quad (3)$$

The ratio of the number of transactions that hold $X \cup Y$ to the number of transactions that hold X is said to be the confidence of an association rule X → Y

$$Conf(X \rightarrow Y) = Support(X \cup Y)/Support(X) \quad (4)$$

In this paper, we have presented a comprehensive survey of the algorithms and techniques available for frequent itemset mining and association rule mining. The algorithms with the incorporation of economic utility factors have also been presented. A comparative study has been performed through the thorough assessment of the results of the algorithms and techniques on the basis of parameters utilized. The execution time and the utilization of memory in conjunction with the minimum threshold for mining frequent itemsets were the chief factors deliberated during the comparison.

2. Literature Survey

This section presents a comprehensive survey, mainly focused on the study of research methods for mining the frequent itemsets and association rules with utility considerations. Most of the existing works paid attention to performance and memory perceptions.

The AIS (Agrawal, Imielinski, Swami) algorithm put forth by Agrawal et al. [3] was the forerunner of all the algorithms used to generate the frequent itemsets and confident association rules, the description of which has been given along with the introduction of mining problem. The algorithm comprises of two phases. The first phase constitutes the generation of the frequent itemsets. This is followed by the generation of the confident and frequent association rules in the second phase. The exploitation of the monotonicity property of the support of itemsets and the confidence of association rules led to the enhancement of the algorithm and it was renamed Apriori in a later point of time by Agrawal et al [4, 6]. Though a number of algorithms were put forth following the introduction of

Apriori algorithm, a majority of them dealt with the optimization of one or more steps of the Apriori bearing the similar general structure. Alongside Apriori, Agrawal et al. [4, 9] proposed the AprioriTid and AprioriHybrid algorithms as well. Apriori outperforms AIS on problems of various sizes. It beats by a factor of two for high minimum support and more than an order magnitude for low levels of support. SETM (SET-oriented Mining of association rules) [59] was constantly outperformed by AIS. AprioriTid performed equivalently well as Apriori for smaller problem sizes however performance degraded twice slow when applied to large problems.

The support counting procedure of the Apriori algorithm has attracted voluminous research owing to the fact that the performance of the algorithm mostly relies on this aspect. Park et al. proposed an optimization, called DHP (Direct Hashing and Pruning) intended towards restricting the number of candidate itemstes, shortly following the Apriori algorithms mentioned above [5]. Brin et al put forth the DIC algorithm that partitions the database into intervals of a fixed size so as to reduce the number of traversals through the database [10]. Another algorithm called the CARMA algorithm (Continuous Association Rule Mining Algorithm) employs an identical technique in order to restrict the interval size to 1.

A methodology that is entirely different from that of the aforesaid ones was proposed by Savasere et al [7]. In this case, the vertical data base layout comes into action while storing the database in main memory besides the computation of an itemset being done with the intersection of the covers of two of its subsets. The Eclat algorithm put forth by Zaki [19] is considered to be the archetype in the depth first manner of generation of frequent itemsets. This was followed by the introduction of diverse depth first algorithms [18, 20] among which the FP-growth algorithm by Han et al. [18] is the most famous and widely used. The numerous algorithms available are categorized based on their attention towards the parameters: performance, memory and discussed briefly with comparison and other related works in the following sub-sections.

2.1. Performance Emphasized Works

The two algorithms namely Apriori and AprioriTid, which discover all significant association rules between items in a large database of transactions was proposed by Agrawal et al. [4]. The best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments demonstrated that AprioriHybrid scales linearly with the number of transactions. In addition, the execution time decreases a little as the number of items in the database increases. As the average transaction size increases (while keeping the database size constant), the execution time increases only gradually. AIS and SETM have always been outperformed by the Apriori and AprioriTid algorithms. There was considerable increase in the performance gap with the increase in problem size, ranging from a factor of three for tiny problems to more than an order of magnitude for huge ones.

S. Brin et al. [10] have presented an algorithm for finding large itemsets which uses fewer passes over the data than classic algorithms, and yet uses fewer candidate itemsets than methods based on sampling. In addition they have presented a new way of generating "implication rules", which are normalized based on both the antecedent and the consequent. They produced more intuitive results than other methods.

C. Hidber [12] has presented a novel algorithm named CARMA (Continuous Association Rule Mining Algorithm), which is used to compute large itemsets online. It continuously produced large itemsets along with a shrinking support interval for each itemset. He has showed that CARMA's itemset lattice quickly approximates a superset of all large itemsets while the sizes of the corresponding support intervals shrink rapidly. The memory efficiency of CARMA was an order of magnitude greater than Apriori. Apriori and DIC (Dynamic Itemset Counting) [60] fell behind CARMA on low support thresholds. Besides, the CARMA has been found to be sixty times more memory efficient.

J.S. Park et al. [5] have proposed a DHP (direct hashing and pruning) algorithm for efficient large itemset generation. The proposed algorithm has two major features: one is efficient generation for large itemsets and other is effective reduction on transaction database size. By utilizing the hash techniques, DHP is very efficient for the generation of candidate set for large 2-itemsets, in orders of magnitude, smaller than that by previous methods, thus resolving the performance bottleneck.

Compared with Apriori [4] and its variants which need several database scans, the FP-growth method proposed by Jiawei Han et al. [32] only needs two database scans when mining all frequent itemsets. Jiawei Han et al have proposed a novel data structure, frequent pattern tree (FP-tree), for storing compressed, crucial information about frequent patterns, and developed a pattern growth method, FP-growth, for efficient mining of frequent patterns in large databases. Their method ensured that it never generates any combinations of new candidate sets which are not in the database because the itemset in any transaction is always encoded in the corresponding path of the FP-trees. The FP-growth method is about an order of magnitude faster than the Apriori algorithm and some recently reported frequent-pattern mining methods besides being efficient and scalable for mining both long and short frequent patterns. A conditional FP-tree is in orders of magnitude smaller compared to the global FP-tree. Therefore the size of the FP-trees to be handled would be greatly decreased when a conditional FP-tree is created out of each projected database. This has been proved to be faster than the Tree-Projection algorithm [16] where in the database is projected recursively into a tree of databases.

Mohammed J. Zaki et al. [11] have presented CHARM (Closed Association Rule Mining; the 'H' is gratuitous), an efficient algorithm for mining all frequent closed itemsets. It has enumerated closed sets using a dual itemset-tidset search tree, using an efficient hybrid search that skips many levels. It also uses a technique called diffsets to reduce the memory footprint of intermediate computations. An extensive experimental evaluation on a number of real and synthetic databases shows that CHARM significantly outperforms previous methods. Besides being several orders of magnitude better than Pascal [56], CHARM can also be run on very low support values [56]. Characteristically Pascal is twice as quick as A-Close [57], and ten times quicker than Apriori. CHARM performs better than Closet [58] by an order of magnitude or more, particularly in case of lowered support.

M. J. Zaki [19] has presented new algorithms for discovering the set of frequent itemsets. He also presented a lattice-theoretic approach to partition the frequent itemset search space into small, independent sub-spaces using either prefix-based or maximal-clique-based methods. The evaluated results showed that the maximal-clique based decomposition is more precise and leads to smaller classes.

A new class of interesting problem called weighted association rule (WAR) problem was identified by Wei Wang, et al. [17]. They have proposed an approach which mines WARs by first ignoring the weight and finding the frequent itemsets and it was followed by introducing the weight during the rule generation. Their approach not only results in shorter average execution times, but also produces high quality results than the generalization of known methods on quantitative association rules.

Mohammed J. Zaki et al. [28] have presented a novel vertical data representation called Diffset that only keeps track of differences in the tids of a candidate pattern from its generating frequent patterns. They have showed that diffsets drastically cut down (by orders of magnitude) the size of memory required for storing intermediate results. The running time of vertical algorithms like Eclat [19] and CHARM [45] were improved by several orders of magnitude with the aid of Diffsets. Tidset based methods are outperformed in several orders of magnitude by diffset algorithms. The average diffset size corresponding to long patterns is several orders of magnitude smaller than the analogous average tidset size i.e on dense sets, it is four to five orders of magnitude smaller whereas by only two to three orders for sparse sets.

Ferenc Bodon [27] has analyzed theoretically and experimentally Apriori [4], the most established algorithm for frequent itemset mining. The implementations of the Apriori algorithm have

displayed large differences in running time and memory need. He has modified Apriori and named it as Apriori_Brave that appears to be faster than the original algorithm.

An enhancement with a memory efficient data structure of a quantitative approach to mine association rules from data was proposed by Liang Dong et al. [24]. The best features of the three algorithms (the Quantitative Approach, DHP, and Apriori) were combined to constitute the proposed approach. The obtained results accurately reflected the knowledge hidden in the datasets under examination.

A new single-pass algorithm, called DSM-FI (Data Stream Mining for Frequent Itemsets) was proposed by Hua-Fu Li, et al. [30], which mines all frequent itemsets over the entire history of data streams. DSM-FI outperforms the Lossy Counting [23] in terms of execution time and memory usage between the large datasets.

PRICES: an efficient algorithm for mining association rules was proposed by Chuan Wang [31], which first identifies all large itemsets and then generates association rules. His approach reduced large itemset generation time, known to be the most time-consuming step, by scanning the database only once and using logical operations in the process. PRICES is competent and proficient and can sometimes be ten times as quick as Apriori

Mingjun Song et al. [38] have presented a novel transaction algorithm for mining complete frequent itemsets. In their approach, transaction ids of each itemset were transformed and compressed to continuous transaction interval lists in a different space using the transaction tree and frequent itemsets were found by transaction intervals intersection along a lexicographic tree [16] in depth first order. This compression greatly saves the intersection time. Their algorithm has outperformed FP-growth [29] and dEclat [28] on the basis of runtime and storage cost.

Yanbin Ye et al. [39] have implemented a parallel Apriori algorithm based on Bodon's work [27] and analyzed its performance on a parallel computer. Their implementation was a partition based Apriori algorithm that partitions a transaction database. They have also shown that partitioning a transaction database has improved the performance of frequent itemsets mining by fitting each partition into limited main memory for quick access and allowing incremental generation of frequent itemsets.

An effective and efficient Fuzzy Healthy Association Rule Mining Algorithm (FHARM) [47] has been proposed by M. Sulaiman Khan, et al. In their approach, edible attributes were filtered from transactional input data by projections and were then converted to Required Daily Allowance (RDA) numeric values. The averaged RDA database was then converted to a fuzzy database that contains normalized fuzzy attributes comprising different fuzzy sets. Their algorithm produced more interesting and quality rules by introducing new quality measures.

As a replacement for standard support and confidence measure, a weighted support and confidence framework for mining weighted association rules (Boolean and quantitative data) by validating the downward closure property (DCP) has been presented by M. Sulaiman Khan, et al. [53]. The classical and fuzzy ARM was used to solve the issue of invalidation of DCP in weighted ARM. The problem of invalidation of downward closure property was solved by using improved model of weighted support and confidence framework for classical and fuzzy association rule mining.

The frequent itemsets discovered using the above algorithms does not reflect the impact of any other factor except frequency of the presence or absence of an item. Frequent itemsets may only contribute a small portion of the overall profit, whereas non-frequent itemsets may contribute a large portion of the profit. This leads to the necessity of high utility itemset mining. Several authors have proposed algorithms for high utility itemset mining. A brief overview of some of the algorithms for high utility itemset mining is as follows:

Ying Liu et al. [37] have proposed a two-phase algorithm that can discover high utility itemsets very efficiently. The accuracy, effectiveness and scalability of their algorithm are demonstrated using both real and synthetic data on shared memory parallel machines. Their algorithm can handle very large databases with ease and it requires fewer database scans, less memory space and less

computational cost. The transaction-weighted utilization mining not only effectively restricts the search space, but also covers all the high utility itemsets.

Yu-Chiang Li et al. [34] have evaluated the significance of itemsets for the mining of association rules from databases. They have proposed an algorithm; Enhanced FSM (EFSM), which efficiently reduces the time complexity of the join step [36]. In addition, two additional algorithms were presented namely, SuFSM and ShFSM, developed from EFSM. SuFSM and ShFSM prune the candidates more efficiently than FSM and therefore it can improve the performance significantly.

In [49] Jieh-Shan Yeh et al. have proposed a bottom-up two-phase algorithm, BU-UFM, for efficiently mining utility-frequent itemsets. They have introduced a concept, quasi-utility-frequency, which is upward, closed with respect to the lattice of all itemsets. A top-down two-phase algorithm, TD-UFM, for mining utility-frequent itemsets was also proposed by them. An efficient algorithm FUFM (Fast Utility-Frequent Mining) was presented by Vid Podpecan, et al. [50], which finds all utility-frequent itemsets within the given utility and support constraints threshold. It is faster and simpler than the original BU-UFM algorithm (Bottom-up Utility-Frequent Mining) [49], as it is based on efficient methods for frequent itemset mining.

A new algorithm named CTU-Mine (Compressed Transaction Utility-Mine) was proposed by Alva Erwin, et al. [48], which mines high utility itemsets using the pattern growth approach. An evaluation of the performance of CTU-Mine was done on several dense data sets and compared against the Two-Phase algorithm [37]. CTU-Mine has performed more efficiently with regard to execution time with varying minimum utility thresholds on synthetic dense datasets.

Chun-Jung Chu et al. [51] have proposed a novel method, namely THUI (Temporal High Utility Itemsets)-Mine, used for mining temporal high utility itemsets from data streams efficiently and effectively. The uniqueness of THUI-Mine is that it can effectively identify the temporal high utility itemsets by generating fewer candidate itemsets such that the execution time can be reduced substantially in mining all high utility itemsets in data streams with less memory space. THUI-Mine is in orders of magnitude quicker than Two-Phase and the margin develops with the decrease in minimum utility threshold.

Guangzhu Yu et al. [52] have proposed a hybrid method, which is composed of a row enumeration algorithm (i.e., Inter-transaction) and a column enumeration algorithm (i.e., Two-phase), to discover high utility itemsets from two directions: Two-phase seeks short high utility itemsets from the bottom, while Inter-transaction seeks long high utility itemsets from the top. In addition, optimization technique was adapted to improve the performance of computing the intersection of transactions.

CTU-PRO algorithm was proposed by Alva Erwin et al. [45] to mine the complete set of high utility itemsets from both sparse and relatively dense datasets with short or long high utility patterns. Their data structure and algorithm have extended the pattern growth approach, taking into account the lack of anti-monotone property for pruning utility based patterns. The performance of CTU-PRO was compared against the Two Phase algorithm [37] and CTU-Mine [48] and shown that CTU-PRO works more efficiently than Two Phase and CTU-Mine on dense data sets.

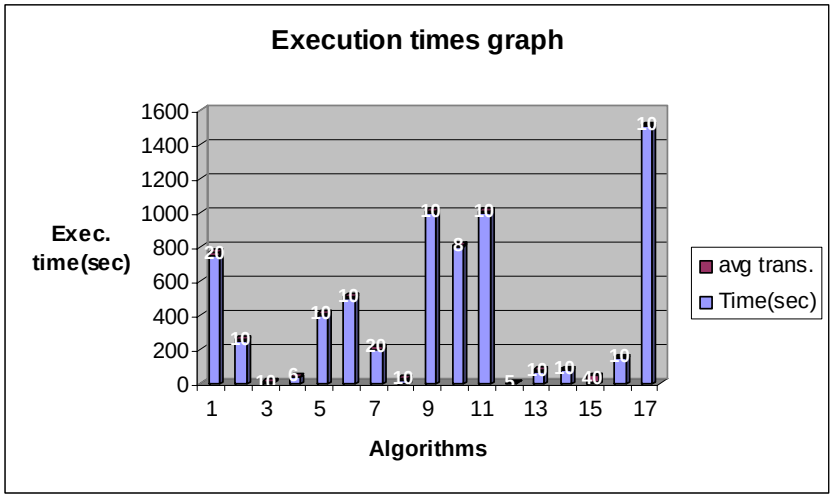
In Table 1, the results of various algorithms have been evaluated based on their execution time for mining the frequent itemsets and association rules from large databases. The table shows the file size, average size of transactions, average size of itemsets, the minimum threshold value and the corresponding execution time.

Table 1: Comparison of Execution times of different Algorithms

Algorithm Name	File Size	Avg. size of transactions	Avg. size of itemsets	Threshold	Time(sec)
Two-Phase[37]	1000K	20	6	1	750
FUFM[50]	100K	10	4	0.5	250
AprioriHybrid[4]	100K	10	4	0.75	7.5
EFSM[34]	100K	6	4	0.4	40
DSM-FI [30]	2000K	10	5	0.01	400
CTU-Mine[48]	10K	10	10	0.5	500
THUI[51]	100K	20	6	1	200
Transaction[38]	100K	10	4	1	22
Parallel Apriori [39]	100K	10	4	0.005	998
Inter-transaction[52]	8000K	8	6	0.01	800
WAR[17]	1000K	10	4	0.1	1000
CTU-PRO[45]	100K	5	5	0.5	5
FHARM[47]	100K	10	4	0.3	75
FWARM[53]	100K	10	4	1	80
Apriori_Brave[27]	100K	40	10	0.05	8.3
PRICES[31]	100K	10	4	5	150
Combined Approach[24]	100K	10	4	4	1500

Figure 1 shows the graphical representation of the execution times of all the algorithms given in Table 1.The graph has been plotted with the results obtained by corresponding authors, which provides a clear idea about execution times of all the algorithms given in Table. The execution time and average size of transactions in the datasets of all the algorithms have been employed to plot the graph. The x-axis represents the number of algorithms and y-axis represents the execution time and z-axis represents the average size of transactions.

Figure 1: Execution Times Graph of Performance emphasized Algorithms



2.2. Memory Emphasized Works

A frequent pattern mining algorithm named H-mine using the data structure H-struct was proposed by Jian Pei et al. [44]. Their algorithm have taken the advantage of the data structure H-struct and dynamically adjusted links in the mining process. It can be scaled up to very large databases using database partitioning. H-mine has high performance and is scalable in many kinds of data, with a very limited and precisely predictable main memory overhead, and outperforms currently existing algorithms with various settings. Pattern-growth methods such as FP-growth and TreeProjection fall

behind H-mine (Mem) in case of mining sparse datasets since the latter has polynomial space complexity. Moreover, H-mine (Mem) is more beneficial than the Apriori-based methods that generate numerous candidates.

The algorithms put forth by Chun-Jung Chu et al. [51], Hua-Fu Li, et al. [30] and Liang Dong, et al.[24] have been discussed concisely in the aforementioned subsection. These algorithms tend to focus on the memory usage besides dealing with the performance of mining.

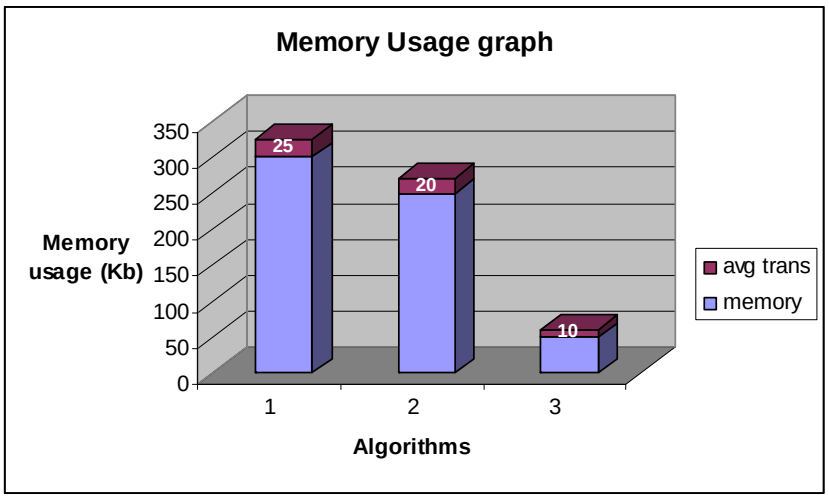
In Table 2, the results of various algorithms have been evaluated based on their memory usage for mining the frequent itemsets and association rules from large databases. The table shows the file size, average size of transactions, average size of itemsets, the minimum threshold value and the corresponding memory usage.

Table 2: Comparison of Memory Usage of different Algorithms

Algorithm Name	File size	Avg. size of transactions	Avg. size of itemsets	Threshold	Memory Usage
H-mine [44]	10K	25	15	1	300Kb
THUI [51]	10K	20	6	0.2	250Kb
DSM-FI [30]	1000K	10	5	0.01	50Kb
Combined Approach[24]	10K	8	4	10	13Mb

Figure 2 shows the graphical representation of the memory usage of all the algorithms given in Table 2. The graph has been plotted with the results obtained by corresponding authors, which provides a clear idea about the memory usage of all the algorithms given in Table. The memory usage and average size of transactions in the datasets of all the algorithms have been employed to plot the graph. The x-axis represents the number of algorithms and y-axis represents the memory usage and z-axis represents the average size of transactions.

Figure 2: Memory Usage Graph of Memory emphasized Algorithms



2.3. Other Related Works

Ashok Savasere et al. [7] have described an algorithm which is not only efficient but also fast for discovering association rules in large databases. An important contribution of their algorithm is that it drastically reduces the I/O overhead associated with previous algorithms. Their algorithm not only reduces the I/O overhead significantly but also has lower CPU overhead for most cases. Moreover their algorithm is especially suitable for very large size databases. The complexity of their algorithm is explained subsequently: The CPU overhead was decreased by a factor of four accompanied by almost an order of magnitude reduction in I/O in case of voluminous databases.

A simple, effective and highly interactive approach proposed by Bing Liu et al. [13] solves the interestingness problem completely. The proposed methodology was designed to perform the post-analysis of the discovered patterns to help the user identify the interesting ones. Their technique is based on fuzzy matching [1] of the discovered patterns with a set of user-specified patterns. The degrees of match are then used to rank the discovered patterns according to various interestingness measures, such as unexpectedness and actionability. The runtime complexity of their approach has been described as follows: The fundamental algorithm is the same for recognizing actionable patterns. Attribute value matching consumes constant time. The worst-case time complexity of the procedure is given by $O(E||B|N^2)$ where N is the maximal number of propositions in a pattern.

An algorithm that exploits all user specified constraints including minimum support, minimum confidence, and a new constraint was proposed by Roberto J. Bayardo Jr et al. [15]. Their algorithm ensured that every mined rule offers a predictive advantage over any of its simplifications showed how Dense-Miner exploits rule constraints to efficiently mine consequent constrained rules from large and dense data-sets, even at low supports [15].

Roberto J. Bayardo Jr et al. [14] have proposed an approach to select most-interesting rule according to several interestingness metrics including support, confidence, gain, Laplace value, conviction, lift, entropy gain, gini, and chi-squared value. Their approach is capable of mining all rules that are best according to any of these criteria with respect to an arbitrary subset of the population of interest. The techniques allowed for improved insight into the data and support more user-interaction in the optimized rule-mining process.

Ke Wang et al. [21] have presented a profit-based data mining approach called profit mining. The goal of profit mining is to construct a recommender that recommends target items and promotion codes on the basis of maximizing the profit of target sales on future customers. They presented a scalable construction of recommends to address several important requirements in profit mining: pruning specific rules on a profit-sensitive basis, dealing with the behavior of shopping on unavailability, dealing with sparse and explosive search space, ensuring optimality and interpretability of recommenders.

A fresh ideology, to model the association patterns that bear explicit relation with the user's objective and relativity, called the Objective-Oriented utility based Association (OOA) mining has been projected by the authors in [22]. With regard to the usage of user's objective and objective utility as the key semantic information in the determination of usefulness of association patterns the task of decision making is found to be incorporated flawlessly within the association mining procedure.

The incremental utility mining can identify all high temporal utility itemsets in a specified time period on an incremental transaction database. Two efficient algorithms, Incremental Utility Mining (IUM) and Fast Incremental Utility Mining (FIUM), have been proposed by Jieh-Shan Yeh et al. [54]. These algorithms can efficiently identify all high temporal utility itemsets that users will be interested in particular periods when the new transaction data are added into the original transaction database. The algorithm not only finds the temporal high utility itemsets for particular time periods, but also can find the high utility itemsets for the entire transaction database. The time complexity of the C^k_p generation is $O(n^{2p-2})$, where n is the number of RU^k_{p-1}. Nevertheless the time complexity of FIUM-Algorithm generation is reduced to $O(n^p)$.

The approach proposed by Hong Yao et al. [41] permits the users to quantify their preferences concerning the usefulness of itemsets using utility values. They have proposed two algorithms namely UMining and UMining_H which mainly reduce the execution time and the number of passes while mining the utility values. The UMining algorithm can efficiently find all useful itemsets from both a synthetic and real world database, while methods for frequent itemset mining, convertible constraint based mining, and share based mining cannot.

Jing Wang et al. [43] have proposed a model called general utility mining, which takes both frequency and utility into consideration simultaneously. It balanced the impact of frequency and utility

by adjusting their weights. By adjusting the weight of the frequency factor or the utility factor, this model can meet the different preferences of different applications.

GenMax, a backtracking search based algorithm for mining maximal frequent itemsets was proposed by Karam Gouda et al. [35]. GenMax uses a number of optimizations to prune the search space. It uses a novel technique called progressive focusing to perform maximality checking, and diffset propagation to perform fast frequency computation. They have shown that GenMax is a highly efficient method to mine the exact set of maximal patterns. Considering the dense datasets, the MaxMiner is distinctively outperformed in orders of magnitude by the GenMax. Even though MaxMiner performs well on sparse datasets the new procedure outperforms MaxMiner when it comes to low support value (such as 0.1% on t40).

Most association rule mining algorithms suffer from the twin problems of too much execution time and generating too many association rules. In order to solve this problem, Girish K. Palshikar.et al. [46] proposed a solution to address the latter problem and proposed the concept of heavy itemset, which compactly represents an exponential number of rules. An efficient greedy algorithm was projected to generate a collection of disjoint heavy itemsets in a given transaction database. They also presented a modified Apriori algorithm that uses the given collection of heavy itemsets and detects more heavy itemsets.

3. Conclusion

Frequent itemset mining and association rule mining are the two important tasks of data mining. Numerous efficient algorithms are available in the literature for mining frequent itemsets and association rules. Incorporating utility considerations in data mining tasks is gaining popularity in recent years. Discovering association rules used to ascend the business of an enterprise has long been recognized in data mining community. In this paper, we have performed a comprehensive survey of the algorithms and methods in existence for the mining of frequent itemsets and association rules with utility considerations. A brief discussion of a number of algorithms was presented along with a comparative study of a few significant ones based on their performance and memory usage.

References

[1] H. J. Zimmermann, 1991. "Fuzzy set theory and its applications", Second Edition, Kluwer Academic Publishers.

[2] W. J. Frawely, G. Piatesky-Shapiro, C. J. Matheus, 1991. "Knowledge discovery in databases: An overview", AAAI/MIT Press, pp. 1-27.

[3] R.Agrawal, T.Imielinski, and A.Swami, 1993. "Mining association rules between sets of items in large databases", in proceedings of the ACM SIGMOD Int'l Conf. on Management of data, pp. 207-216.

[4] Rakesh Agrawal, and Ramakrishnan Srikant, 1994. "Fast Algorithms for Mining Association Rules", In Proceedings of the 20th Int. Conf. Very Large Data Bases, pp. 487-499.

[5] J.S. Park, M.-S. Chen, and P.S. Yu, 1995. "An effective hash based algorithm for mining association rules". In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, volume 24(2) of SIGMOD Record, pp. 175–186. ACM Press.

[6] Srikant R, Agrawal R., 1995. "Mining generalized association rules". In: Dayal U, Gray P M D, Nishio Seds. Proceedings of the International Conference on Very Large Databases. San Francisco, CA: Morgan Kanfman Press, pp. 406-419

[7] Ashok Savasere, Edward Omieinski and Shankant Navathe, 1995. "An Efficient Algorithm for Mining Association Rules in Large Databases", Proceedings of the 21st International Conference on Very Large Data Bases, pp. 432 – 444.

[8] M. S. Chen, J. Han, and P. S. Yu, 1996 "Data mining: An overview from a database perspective", IEEE Transactions on Knowledge Data Engineering, Vol.8, pp. 866-883.

[9] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, 1996. "Fast discovery of association rules" In U.M. Fayyad, G. Piatetsky - Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pp. 307–328. MIT Press.

[10] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, 1997. "Dynamic itemset counting and implication rules for market basket data". In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, volume 26(2) of SIGMOD Record, pp. 255–264. ACM Press.

[11] M. J. Zaki and C.-J. Hsiao, October 1999. "CHARM: An efficient algorithm for closed association rule mining". Technical Report 99-10, Computer Science Dept., Rensselaer Polytechnic Institute.

[12] C. Hidber, 1999. "Online association rule mining". In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, volume 28(2) of SIGMOD Record, pp. 145–156. ACM Press.

[13] Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee, 1999. "Finding Interesting Patterns Using User Expectations", IEEE Transactions on Knowledge and Data Engineering, Vol 11, No. 6, pp. 817-832.

[14] Roberto J. Bayardo Jr., and Rakesh Agrawal, 1999. "Mining the Most Interesting Rules", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, pp. 145-154.

[15] Roberto J. Bayardo Jr., Rakesh Agrawal and Dimitrios Gunopulos, 1999. "Constraint-Based Rule Mining in Large, Dense Databases", Research Report – IBM, In Proceedings of the 15th International Conference on Data Engineering, pp.188-197.

[16] R. Agrawal, C. Aggarwal, and V. Prasad, 2000. "A Tree Projection Algorithm for Generation of Frequent Item Sets", Parallel and Distributed Computing, pp. 350-371.

[17] Wei Wang, jiong yang and philip S. Yu, 2000. "Efficient Mining of Weighted Association Rules (WAR)", Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining,pp. 270 - 274.

[18] Han, J., J. Pei, and Y. Yin, 2000. "Mining Frequent Patterns without Candidate Generation" in ACM SIGMOD Int'l Conference on Management of Data.

[19] M.J. Zaki, May/June 2000. “Scalable algorithms for association mining”. IEEE Transactions on Knowledge and Data Engineering, 12(3):372–390.

[20] C. Borgelt, 2005. “An Implementation of the FP-growth Algorithm”, Workshop Open Source data Mining Software, OSDM'05, Chicago, IL, 1-5.ACM Press, USA.

[21] Ke Wang, Senqiang Zhou, and Jiawei Han, 2002. “Profit Mining: From Patterns to Action”, Proceedings of International Conference on Extending Database Technology, pp. 70-87.

[22] Yi-Dong Shen, Zhong Zhang, 2002. “Objective-Oriented Utility-Based Association Mining”, Proceedings of the IEEE International Conference on Data Mining,pp. 426- 433.

[23] G. S. Manku and R. Motwani, 2002. “Approximate Frequency Counts Over Data Streams”. In Proc. of the 28th VLDB conference.

[24] Liang Dong and Christos Tjortjis, 2003. “Experiences of Using a Quantitative Approach for Mining Association Rules”, in Lecture Notes Computer Science, Vol. 2690, pp. 693-700.

[25] Srikant, R. and Agrawal, R., 1996. “Mining Quantitative Association Rules in Large Relational Tables.” In Proc. of ACM SIGMOD Conf. on Management of Data. ACM Press, pp. 1-12.

[26] Cunningham, S. J. and Holmes, G., 1999. "Developing innovative applications in agriculture using data mining," In the Proceedings of the Southeast Asia Regional Computer Confederation Conference, Singapore.

[27] F. Bodon, 2003. “A Fast Apriori Implementation”, In B. Goethals and M. J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Vol. 90 of CEUR Workshop Proceedings.

[28] M.J. Zaki and K. Gouda, 2003. “Fast Vertical Mining Using Diffsets”, Proc. Ninth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 326-335.

[29] G. Grahne and J. Zhu, 2003. “Efficiently Using Prefix-Trees in Mining Frequent Itemsets”, In proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI).

[30] Hua-Fu Li, Suh-Yin Lee and Man-Kwan Shan, 2004. “An Efficient Algorithm for Mining Frequent Itemsets over the Entire History of Data Streams”, In Proceedings of the 1st Int’l. Workshop on Knowledge Discovery in Data Streams, pp. 20- 24.

[31] Chuan Wang, Christos Tjortjis, 2004. “PRICES: An efficient algorithm for mining association rules”, in Lecture Notes Computer Science Vol. 3177, pp. 352-358, ISSN: 0302-9743.

[32] Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, January 2004. "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Volume 8, Issue 1, pp. 53 – 87.

[33] Marek Wojciechowski, Krzysztof Galecki, Krzysztof Gawronek, 2005. “Concurrent Processing of Frequent Itemset Queries Using FP-Growth Algorithm”, Proc. of the 1st ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD'05), Tallinn, Estonia.

[34] Yu-Chiang Li, Jieh-Shan Yeh, Chin-Chen Chang, 2005. “Efficient Algorithms for Mining Share-Frequent Itemsets”, In Proceedings of the 11th World Congress of Intl. Fuzzy Systems Association.

[35] Karam Gouda and Mohammed J. Zaki, 2005. “GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets, Data Mining and Knowledge Discovery, Vol. 11, No. 3, pp. 223-242.

[36] Y. C. Li, J. S. Yeh, and C. C. Chang, 2005. “A fast algorithm for mining share-frequent itemsets”, Lecture Notes in Computer Science, Springer-Verlag, Vol. 3399, pp. 417-428.

[37] Ying Liu, W. K. Liao, and A. Choudhary, 2005. "A Fast High Utility Itemsets Mining Algorithm", Proc. UBDM'05, Chicago Illinois.

[38] Mingjun Song and Sanguthevar Rajasekaran, 2006. ”A Transaction Mapping Algorithm for Frequent Itemsets Mining”, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No.4, pp. 472-481.

[39] Yanbin Ye and Chia-Chu Chiang, 2006. “A Parallel Apriori Algorithm for Frequent Itemsets Mining”, Fourth International Conference on Software Engineering Research, Management and Applications, pp. 87- 94.

[40] S. Kotsiantis, D. Kanellopoulos, 2006. “Association Rules Mining: A Recent Overview”, GESTS International Transactions on Computer Science and Engineering, Vol.32, No. 1, pp. 71-82.

[41] Hong Yao, Howard J. Hamilton, 2006. “Mining itemset utilities from transaction databases, Data & Knowledge Engineering”, vol. 59, pp. 603-626.

[42] Weiss G., Zadrozny B., Saar-Tsechansky M., 2006. “Utility-based data mining”, workshop report. SIGKDD Explorations, Vol. 8, No. 2.

[43] Jing Wang, Ying Liu, Lin Zhou, Yong Shi, and Xingquan Zhu, 2007. “Pushing Frequency Constraint to Utility Mining Model”, Lecture notes in computer science, Springer, pp. 685-692.

[44] Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang and Dongqing Yang, 2007. “H-Mine: Fast and space-preserving frequent pattern mining in large databases”, IIE Transactions, Vol 39; No. 6, pp. 593-605.

[45] Alva Erwin, Raj P. Gopalan, N.R. Achuthan, 2007. “A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets”, In Proceedings of the 2nd international workshop on Integrating artificial intelligence and data mining,Vol. 84, pp. 3-11.

[46] Girish K. Palshikar, Mandar S. Kale, Manoj M. Apte, 2007. “Association Rules Mining Using Heavy Itemsets”, Data & Knowledge Engineering, Vol. 61, No. 1, pp. 93-113.

[47] M. Sulaiman Khan, Maybin Muyeba, Christos Tjortjis, Frans Coenen, 2006. “An effective Fuzzy Healthy Association Rule Mining Algorithm (FHARM)”, In Lecture Notes Computer Science, Vol. 4224, pp.1014-1022, ISSN: 0302-9743.

[48] Erwin, A., Gopalan, R. P., and Achuthan, N. R., 2007. “CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach”, IEEE 7th International Conferences on Computer and Information Technology, pp. 71-76.

[49] Yeh J. S., Li, Y. C., Chang C. C., 2007. " A Two-Phase Algorithm for Utility-Frequent Mining", To appear in Lecture Notes in Computer Science, International Workshop on High Performance Data Mining and Applications.

[50] Vid Podpecan, Nada Lavrac, and Igor Kononenko, 2007. “A Fast Algorithm for Mining Utility-Frequent Itemsets”, International Workshop on Constraint-based Mining and Learning at ECML/PKDD.

[51] Chun-Jung Chu, Vincent S. Tseng, Tyne Liang, 2008. “An efficient algorithm for mining temporal high utility itemsets from data streams”, Journal of System Software, Vol. 81, No. 7, pp. 1105-1117.

[52] Guangzhu Yu, Keqing Li, Shihuang Shao, 2008. “Mining High Utility Itemsets in Large High Dimensional Data”, International Workshop on Knowledge Discovery and Data Mining WKDD, pp. 17-20.

[53] M. Sulaiman Khan, Maybin Muyeba, Frans Coenen, 2008. “Fuzzy Weighted Association Rule Mining with Weighted Support and Confidence Framework”, to appear in ALSIP (PAKDD), pp. 52-64.

[54] Jieh-Shan Yeh, Chih-Yang Chang and Yao-Te Wang, 2008. “Efficient Algorithms for Incremental Utility Mining”, Proceedings of the 2nd international conference on Ubiquitous information management and communication,pp. 212-217.

[55] Khan, M.S. Muyeba, M. Coenen, F., 2008. “A Weighted Utility Framework for Mining Association Rules”, Second UKSIM European Symposium on Computer Modeling and Simulation, pp. 87-92.

[56] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal, December 2000. “Mining frequent patterns with counting inference”. SIGKDD Explorations, 2(2).

[57] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, January 1999. “Discovering frequent closed itemsets for association rules”. In 7th Intl. Conf. on Database Theory.

[58] J. Pei, J. Han, and R. Mao,May 2000. “Closet: An efficient algorithm for mining frequent closed itemsets”. In SIGMOD Int’l Workshop on Data Mining and Knowledge Discovery.

[59] M. Houtsma, and Arun Swami, 1995. “Set-Oriented Mining for Association Rules in Relational Databases”. IEEE International Conference on Data Engineering, pp. 25–33.

[60] Brin, S., Motwani, R., Ullman, J.D. and Tsur. S, 1997. “Dynamic Itemset Counting and Implication Rules for Market Basket Data”. in Proceedings of the ACM SIGMOD Conference, pp. 255-264.