Frequent Itemset mining Models: Contemporary affirmation of the recent literature

K.Vinuthna[1] P.V.S.Srinivas[2]

Abstract: Distinguishing the association rules in large databases is having high degree of presence in data mining. This paper is for the most part meant at taking into account of previous explorations, current operational standing and to conclude the gaps flanked by them with current identified information. The two problems identified regarding this context are: identifying all frequent item sets and to generate constraints from them. Here, first problem, as it obtains more processing time, is computationally expensive. As a result, many algorithms are projected to handle this problem. Current study focused on these algorithms and their related issues.

Keywords: Frequent itemset mining, multiple minimum supports, Utility mining, tree based frequent itemset mining, Frequent patterns, Minimum support Count, Association Rule Mining

1. INTRODUCTION

In the operation of association rule mining, frequent pattern mining is an important stage that has been aimed at and in which remarkable improvements have been made. The research varies from efficient and scalable algorithms to most research frontiers; including sequential, structured, correlative mining, associative classification and frequent pattern based clustering. Let us discuss present status of this step including the analyzed challenges.

Frequent patterns are item sets or substructures which occur in a dataset more than specified minimum no. of times. A substructure can be a sub-graph or sub-tree. If such substructure occurs more than particular threshold, it is called a frequent structural pattern. Identifying frequent patterns is significant in mining associations and correlations. It contributed in data indexing, classification and clustering. It is proposed by Agarwal et al. [1] for market basket analysis which explores customer characteristics from the associations between objects in the basket. There are a number of proposed algorithms for generating frequent item sets which vary in the way of traversing item set lattice, use of anti-monotone property and the way to handle database. Based on these variations, representative set of algorithms is explained.

2. NOMENCLATURE

Apriori [1] algorithm improved the studies over frequent pattern mining. Here, we distinguished the pattern detection techniques based on their similarities.

The initial algorithm suggested was AIS, by Agrawal et al. (1993) [1] for association rule mining problem. It is a multi-pass algorithm; where candidate item sets are formed while passing the database by extending prior frequent item sets with each transaction items. But, it creates more no. of candidates which may convert into infrequent in future. Moreover, data structures for maintenance are not specified. SETM, which uses SQL, is another algorithm which represents frequent item set in the form of <TID (Transaction unique ID), item set>. Its drawback is similar to that of AIS. The main problem with SETM was identified by, Agrawal and Srikant (1994) [2] and Sarawagi et al. [28].

Apriori seemed to be better algorithm in next generation, which completely includes the subset frequency based pruning optimization. It uses hash tree to save counter. But, it passes over the database length of longest frequent item set times (n). In Kth pass, counts of K item sets are obtained. Another drawback is that it follows tuple-by-tuple approach after every transaction which is an overhead. As there are large no. of single items in database which may form big no. of item sets, it is difficult to develop a scalable algorithm for it. Agrawal and Srikant [2] noticed an interesting downward closure property, called Apriori which refers that an item set is frequent only if all its subsets are frequent. This is the extract of the Apriori algorithm [2] and its alternative [3]. With the proposal of Apriori algorithm, no. of database passes also decremented.

2.1 Divergence Approaches

Partitioning technique: It is proposed by Savasere et al [5], in which the database is logically partitioned into disjoint sets. It needs only 2 passes and based on observation that an item set is globally frequent only if it is frequent at local in at least one divided set. TID lists are calculated for each set and for each candidate in the item set.

[1]*Associate Professor, Department of Computer Science & Engineering, Keshav Memorial Institute of Technology, Hyderabad, A.P-500 029, India. #+919849204569, vinuthnar@yahoo.com*

[2] *Professor & Head, Department of Computer Science & Engineering, Geethanjali College of Engineering & Technology, Cheeryal (V), Keesara (M), Hyderabad, A.P-501 301, India, pvssrinivas23@gmail.com*

Sampling approach: It is proposed by Toivonen et al [6] in which a random sample is mined to identify frequent item sets. These item sets are considered as representative of actual frequent item sets. For more precise results, it needs 1 or 2 passes over database. It is similar to Apriori in case of drawbacks.

Non level wise algorithm [7]: Here, candidates are updated after every M transaction, M is a parameter. It is a multi pass algorithm which is completed in 2 passes. It is closer to sampling approach.

Continuous Association Rule Mining Algorithm [8]: It computes frequent item sets online. It allows changing parameters, minimum support and minimum confidence at any transaction during first pass. It is a 2 pass algorithm allowing non static update of candidates. Hidber, 1999 [8] show that it is less capable than Apriori, but has less memory use.

2.2 Tree Structures based Mining frequent item sets

FP tree based algorithm: Sometimes, when Apriori algorithm decreases no. of candidates abruptly, it faces two nontrivial costs. They are, creating large no. of candidate sets and frequently passing the database and comparing the candidates using pattern matching. Han et al. [4] proposed this algorithm that determines complete set of frequent item sets without candidate generation, based on divide and conquer technique. It expand a list of frequent items in initial pass and sorts them in frequency decrement way. The database is then condensed into FP tree. The FP tree is mined, initiating from a frequent length 1 pattern, forming its sub database (having set of prefix paths co occurring with suffix pattern). This is done iteratively. Drawbacks of FP tree are time consumption, no flexibility and no reusability.

This algorithm divides the problem of identifying lengthy frequent patterns into small patterns and concatenating the suffixes and thus minimizing search time. The extensions of FP growth approach include, Depth first generation, proposed by Agarwal et al [27], H-Mine, By Pei et al [11], Top-down and Bottom-up traversals by Liu et al [12] and array based prefix tree structure by Grahne and Zhu [13].

2.3 Interesting Itemset Mining

Sometimes it becomes essential for user to consider only required patterns.

Constraint-based mining: A user may be interested in the patterns satisfying some specified constraint. Constraints can be of different types based on their communication with mining operation. For example, succinct constraints must be inserted at the initiation of mining; anti monotonic must be inserted deep to keep pattern growth under control during mining and monotonic constraints which require only one constraint checking [14]. The push of monotonic constraints was discussed by Grahne et al [15]. The insertion of convertible constraints (e.g. avg() = v), is performed by sorting them in an order for constrained pattern growth [11]. Bonchi et al [17] proposed dual mining. ExAnte was also proposed by him [18] to further prune the data search space with monotone constraints. Gade et al [19] proposed a block constraint which determines item set's significance based on dense block formed by pattern's items. Bonchi and Lucchese [20] proposed an algorithm for mining closed constrained patterns. Yun and Leggett [21] suggested a weighted frequent item set mining algorithm to insert weight constraint while balancing downward closure property.

3. CONTEMPORARY AFFIRMATION OF THE RECENT LITERATURE

3.1 Utility Itemset Mining

Conventional methods of ARM weights equally the all elements involved in transactions of the given dataset. These models prioritize the items by their existence the transaction. The items frequency in given transactions that these conventional models consider are not realistic to conclude the elements, which are frequent and highly profitable. It may also be a practice of the conventional models that sometimes highly profitable elements may be discarded by concluding them as infrequent. In most of the market basket practices it is significant to identify the frequency of elements with regard to profits they generate, which often not found in transactions with required support. Hence a mining strategy referred as Utility mining can be considered to perform profitable frequent itemset mining in real time practices.

Here in this practice the utility is an identity measure to conclude the significance of an itemset in regard to profits. The conventional ARM methods are utterly fails in this contest since these models always gives utility of each element as 1 and the quantity of that element is always 1 if it exists in a given transaction or 0 if not exists.

The high utility mining models were defined in [28], [29], [30], [31] and [32]. These allow exploring the significance of the itemset by utility value. The transactions significance can be identified for several important decisions in business area similar to exploiting profits diminishing costs related to product promotion or inventory can be in use and more vital knowledge about elements of the transactions or customer's causative to the greater part of the revenue can be exposed.

The models related to utility mining defined in [28], [29], [30], [32] having limits with level-wise candidature. Thus the memory resource usage is extremely high due to huge candidature maintenance also evident of poor scalability due to the need of multiple scans of transaction dataset, which proportionate to maximum length of the candidature itemset.

The model called Mining with Expected Utility in short referred as MEU defined in [28], is not able to manage the downward closure property of Apriori. A heuristic has been used by MEU to determine the state of an itemset is a candidate or not. This heuristic usually not appropriate since an overestimation can be suspected. In particular at the initial levels that leads to generate candidates from all combinations of items. This is not viable in the context of huge number of divergent items low utility. From the authors of [28], there are two more algorithms called UMining and UMining-H [29] to estimate the high usefulness itemsets. A pruning by utility upper bound approach was used in UMining. In the similar context UMining-H was defined but the pruning is performing by a heuristic approach. But, the heuristic method is pruning some high utility itemsets inaccurately. As the same model defined in [28] these methods also not able to perform downward closure property of Apriori and fails to avoid the consequences in level-wise candidature and test methodology.

The Two-Phase [30] algorithm was developed based on the definitions of [28] to find high utility itemsets using the downward closure property. They have defined “transaction weighted utilization”, and proved it can preserve the descending closure property. For the first database scan, their algorithm finds all the 1- element transaction weighted exploitation itemset and based on that engenders the candidates for 2-element transaction weighted use itemsets. In the second database scan, it finds all the 2- element transaction weighted exploitation itemset and based on that engenders the candidates for 3-element transaction weighted exploitation itemsets and so on. At the last scan, it finds out the real high utility itemsets from high transaction weighted utilization itemsets. This algorithm undergo from the same problem of the level-wise candidate generation-and- test methodology. CTU- Mine [31] proposed an algorithm that is efficient over Two-Phase algorithm only in dense database when the minimum utility threshold is very low. The isolated items discarding strategy (IIDS) [32] for discovering high utility itemsets was proposed to reduce some candidates in every pass of databases. They developed efficient high utility itemset mining algorithm FUM and DCG+ and showed that their work is better than all previous high utility pattern mining works. But still their algorithms suffers from the level-wise candidate generation-and- test problem and needs multiple database scans depending on the maximum length of the candidate patterns. In this regard C.F. Ahmed et al [33] proposed a novel tree-based candidate pruning technique HUC- Prune (high utility candidates prune) to efficiently mine high utility patterns without level-wise candidate set generation-and-test. It prunes a big number of unnecessary candidates during the mining process. It exploits a pattern development mining approach and needs a maximum of three database scans in contrast to several database scans of the existing algorithms.

Later Wu et al [34], Y.L.Chen et al[35] discussed the freshness of the data referred as “recency” provided as utility factor that is change of data distribution between the past data and the new data.

Although the models [28, 29, 30, 31, 32, 33, 34, 35] referred under utility itemset mining category, they still fail to fully reflect the utility factors Recency, Frequency and Monetary (RFM) in the mining process.

An example of the constraints observed under utility factor "recency" is follows.

In the consideration of utility factor recency, the occurring time of the last transaction of a pattern, should satisfy the recency threshold. Similarly to recency, monetary is defined as that the total amount of money spent by all customers in a pattern should fall into the range between the maximum and minimum monetary thresholds. The algorithm for mining patterns [35] satisfying RFM constraints; they used a fixed time gap that every occurrence of patterns within the given recent time period will have the same influence regardless of the occurring time. However, their approach cannot reflect the different length of stripes of recency in measuring the importance of transactions.

With the knowledge of observations disclosed above Ya-Han Hu et al [22] proposed a scoring-based method in the context of performing the complete concept of RFM on measuring the importance of patterns. In this regard they introduced the RFMP-tree structure and the RFMP-growth algorithm, which are modified from the well-known FP-tree structure and FP-growth algorithm [36].

A complete RFMP-tree contains a list, called RFM-header, and a RFMP-tree. A RFM-header is a list containing all the 1-RFT-patterns (i.e. a RFT-pattern containing one item only),which are sorted according to their Fscore in descending order. Each entry in RFM-header consists five fields: item-name, Rscore, Fscore, Tta, and head of node-link, where item-name registers which item this entry presents, Rscore, Fscore, and Tta record the recency score, frequency score, and total transaction amount of this item, respectively, and head of node link points to the first node in RFMP-tree carrying the same item-name.

Each node in RFMP-tree consists of five fields: item-name, score, Fscore, Tta, parent-link, child-link, and sibling-link, where item-name registers which item this node represents, Rscore, Fscore, and Tta register the recency score, frequency score, and total transaction amount from all the transactions that have the corresponding patterns represented at this node, and parent-link, child-link, and sibling-link register the addresses of the parent node, child nodes, and the next node carrying the same item-name in the RFMP-tree, respectively. The performance analysis indicating that the three thresholds used are having good impact on the proposed model. The importance of the values assumed to these thresholds is not generalized. The experiments done on synthetic datasets are justifying the threshold values considered. Finally it can be concluded that it is necessary to generalize the values of the thresholds proposed and experiments need to be carried out on real datasets that entertains noise and uncertainty.

3.2 Minimum Support Parameter Value Determination Based Itemset Mining

The data mining requirements of a data source can be frequent set of items and similarity. The frequent can be indicated as fuzzy threshold in fuzzy point of view. The authorities of the data source or end users to whom the mining results meant for often expects to find frequency of items is less, more extremely high, or frequent with total coverage. Here all this parameters representing the state of the frequency can be fuzzy thresholds. Hence it is obvious to generate fuzzy sets with significantly purposeful itemsets. In this context the considerable issue is finding frequent itemsets without predefined coverage threshold, which indicates the need of bridge to fix the gap conventional frequent itemset mining and fuzzy threshold. The other mining requirement of the authority of the data sources is measuring similarity, which can be found with given range of relatively minimum support between 0 and 1

Shichao Zhang et al [37] devised a polynomial approximation model for a précised minimum of range between 0 and 1 and "fuzzy estimation" to determine minimum-support for fuzzy sets. **P**rimarily, the proposed model takes a predefined required coverage threshold and computationally converts the specified threshold into an actual minimum support. This facilitates to specify mining concerns regardless of the transaction dataset state. In the absence of transaction dataset knowledge, It is quite common to fail to fix the appropriate minimum coverage threshold. Even A minimum coverage threshold is defined with the knowledge of transaction dataset, often can observe the resultant frequent itemsets away from the user requirements. Under this context Shichao Zhang et al [37] have devised a strategic approach to compute the minimum coverage threshold. Due to its ability to compute minimum coverage threshold, on other hand existing models demand initial support to be set by end users then converts this specified threshold into actual minimum coverage threshold.

The process of the frequent itemset mining can be explored in two folds such as

1) Approximating the possible count of itemsets that compatible to a coverage value opted.
2) Approximating the probability of a specific itemset that satisfies the given coverage value.

In the context of contemporary affirmation of the literature, we can conclude that the first fold of the frequent itemset mining is much complex than second. Much of the research work attained solutions in the consideration of the probability evaluation of the specific itemset that satisfies the given coverage threshold constraints. The models in [38] and [39] evaluate the viable distributions of frequent itemsets. The model in [38] aimed on the sort of distributions can be expected for divergent transaction datasets and modeled such that it can determine the possibility of extracting maximal frequent itemsets collection of expected number and with expected length. The model in [39] uses probabilistic approach to determine the possible count of closed itemsets. The other model devised by Geerts et al. [40] is fixing the limit on candidatures to be generated during frequent itemset mining. in an ordered setting. The upper bound of the candidature count is dynamic and can differ at each level of the order. This upper bound can be set based on the number of candidates allowed and number of itemsets generated in previous level.

Under concern to best levels of our knowledge, the only approach devised by J. Besson, et al [42] is estimating the count of itemsets with desired coverage threshold without developing a "global analytical model". This approach requires a computational strategy to measure the probability of the given itemset to justify the given coverage threshold, and it is ordered in real practices by opting to itemset space sampling scheme.

M. Boley, et al[43] have evaluated the algorithmic processes to estimate a maximum frequent set. This exploration is aggravated by the necessitate for an competent parameter assessment method that can be applied earlier than an exponential time pattern mining algorithm. The estimating algorithms have been considered since both are affected by NP- hard. This is a familiar maximization issue. Since the minimization is to be similar to min set cover issue, hence the hardness remains same even in minimization. With regard to computational complexity observed in performance analysis, M. Boley, et al[43] convinced that a nontrivial approach of estimating algorithm for maximal frequent set is suspect to exist. This parameter evaluation can also be done by considering the number of closed or maximal frequent itemsets under given parameter. In this regard it is interesting to find that in generalized scenario how fast this parameter evaluation is?

The counting of itemsets leads to NP-Hard, since the limit is to assess the max and min support of the itemsets. The pitfall of randomized counting is computational complexities. In that the sever issue is recurrent counting.

In this regard Zalizah Awang Long et al [23] described a design to conclude the interesting pattern, minimum support and item set count. The proposed model is sampling method specific that let to conclude the minimum support, which computed by mean to define possible frequent itemsets. It is broadly divided in three stages: 1.Pre-processing 2.frequent item set algorithm and 3.data re-transformation. The data preparation is two folds and they are

1) Assortment and conversion fold. With the help of WSARE dataset potential attributes are determined in comparison with the synthetic outbreak cases. The Apriori algorithm is constructed in the data transformation module.
2) In the second fold the minimum support and sequence length 'k' is determined. The occurrence discovery process uses results of frequent mining algorithm as coverage threshold.

The results of explored experiments showed that the maximum quantity of resultant itemsets against to lessen minimum support are much significant. On its rest fold of the frequent itemset mining process the count of item set taken for each generated length and positions of high gain are identified.

3.3 Multiple Minimum Supports based Item set Mining

The conventional and initial standard of frequent itemset mining is using single minimum support threshold for entire transaction dataset is inadequate for the reason that it cannot confine the intrinsic properties and/or divergence in frequencies of the items of the given transaction dataset since a part of transactions reflect the frequency of some itemsets as more compared to other itemsets.

In this regard the model devised by Liu et al. [47] can be considered as initial significant work with divergent coverage, which allows considering the item specific divergent coverage values. The coverage of minimal frequent itemset is considered as the lowest minimum supports amongst the items in the itemset. This assignment is, though, not all the times appropriate. When the least coverage of an itemset is considered as the minimal support of the elements in it, then sometimes a big itemset can be with less number of elements, which can be susceptible. Hence it is worthy to quote that the incident frequency of an appealing itemset ought to be superior to the maximal of the coverage values of elements belongs to that itemset. In this regard at first Wang et al. [48] devised a bin-oriented, non-uniform support constraint model that allows the coverage of an itemset can be any of the minimal coverage of elements those belongs to the itemset. Here in this model the elements of the transactions clustered into bins, and neglects the difference of the minimum support value of the elements belongs to same bin. Even if this model is supple in fix the minimum supports to itemsets, the methodology of frequent itemset mining that recommended as an algorithm is a small multifaceted due to its oversimplification. Although this model is comprehensive to handle such issues, the time complexity is high. Besides, the elements with quantitative values in multiple folds in said transactions are not considered in this model. The Apriori based frequent itemset mining with multiple minimum supports is defined Y. C. Lee et al [49][50], which aimed to find the maximal length itemsets with multiple itemset coverage values, which is scalable and simple over Wang et al.'s model.

In recent literature the model proposed by Ya-Han Hu et al[25] is interesting, which have laid a detailed emphasis on Sequential pattern Frequent Mining with multiple minimum supports. Frequent itemset mining is a significant data-mining methodology for determining associations amongst elements or itemsets. Owing to the varied frequencies of different itemsets, stating a unique minimum support not able to be exactly discovering interesting itemsets. The authors argued that the solutions available, which are based on multiple minimum supports (MMS) [9] is complicated. So, based on the new characterizations of frequent itemsets with MMS, a refined PLWAP-tree [10] formation has been proposed, labeled as Preorder Linked Multiple Supports tree (PLMS-tree), to condense and maintains the required information from transaction dataset. Then, PLWAP-Mine is further extended as MSCPG (Multiple Supports Candidate Pattern Growth) that aims to scalable mining of the frequent itemsets under MMS. MSCPG scalable with compact structure due to its underlying concepts called PLWAP-tree and PLWAP-Mine. The MSCPG is working based an algorithm referred as PLMS-Tree, which similar to PLWAP-tree that saves information about all elements and transaction in given dataset in a compact structure that referred as PLMS-Tree. Then this PLMS-Tree that indicates the compact structure of the transaction dataset information will be used by MSCPG to generate frequent itemsets under MMS. The PLMS contradict with PLWAP in the way the items selected to have in tree structure. The PLWAP selects elements with support more than the given support threshold, but in the case of PLMS the items with minimum interesting support will be selected. The exceptional element issue in the mining of frequent itemsets is also addressed in this model

. However the model described in [25] is more specific in presenting tree representation of the data to predict itemsets, in this regard to achieve the scalability in memory usage, the essential pruning process has not been discussed. More over the performance of the proposed PLMS-Tree structure under extended length of itemsets was not explored.

3.4 Frequent Itemset Mining without Candidate Maintenance

In the concerns of avoiding candidature maintenance Han et al. [51] devised a model referred as FP-growth that limits only to two scans of the given set. This is an effective model, which is beginning of the frequent itemset mining with no candidature maintenance. Another mining model presented in [52], which referred as H-Mine. This H-Mine keeps the sparse set in primary memory by using a hyper structure that referred as H-structure. Then it appeal to FP-growth to perform mining on dense set. Another tree based model referred as COFI-Trees can be found in [53]. This model initially structures a matrix from the given transactions and inverses that matrix, then constructs COFI-Trees by using the relatively sparse parts of the inverted matrix. Another Tree based model devised in [54], which referred as CFP-tree. In concern to balance the memory space, the CFP-Tree approach publishes the determined frequent itemsets on disk. In contrast to FP-Tree that publishes transactions on disk, the CFP-tree publishes discovered frequent itemsets. In this sequence of tree based frequent itemset mining models, the other considerable model defined in [55] that referred as pattern fragment growth methodology. This frequent Fragment growth model with underlying concept of FP-Tree aimed to determine maximum length frequent itemsets. The tree model referred as CATS (Compressed and Arranged Transaction Sequences)-Tree devised by William Cheung et al[56] which helps to use determine frequent itemsets under divergent supports. The other conventional properties of this model are finding frequent itemsets with single scan of the set and ability to perform mining in incremental approach since the tree can be updated dynamically. The CATS-Tree model is not hesitating to use large volume of the memory, which can be tolerable since 1) it is compatible and scalable even for sensibly dense transaction set 2) the memory resource are vast and less expensive in current and future state. 3) Current memory management and data compression techniques are reliable and compatible to this CATS Tree. Later, Grouping Compressed tree (GC tree) was devised by Liou et al [57] to further improve the performance of CATS tree.

In this regard Chuang-Kai Chiou et al [26] presented a new tree structure called Sorted Compression (SC) tree and a mining algorithm for association rule are proposed. Advantages of several algorithms are combined in this algorithm, and as per the performance analysis report observed, the SC outperforms (CATS tree) and GC tree mining algorithms. The CATS tree is one of them and it builds its tree structure dynamically so that the mining process is complex and tedious. Hence, an enhanced algorithm called the Sorted Compression tree (SC tree) where association rules can be mined in a bottom-up style instead of bi-directional or recursive is proposed. As a result, the cost of association rule mining is reduced and it is not only more efficient but also space saving.

CATS-tree allows users to adjust the minimum support value. The efficiency of tree construction and association rule mining are improved by SC tree. By pre-sorting the data set, the data arrangement of SC tree is consistent, so that dynamic adjustment of the tree structure can be avoided.

This SC Tree algorithm can simplifies the process of tree construction and another is to simplify the rule mining method. Constructing and mining the tree structure done by SC tree with fewer memory resources that compared to other methods and it is the most efficient algorithm. By applying the SC tree algorithm in large database by employing the technique of parallel and distributed computing, the scalability also increases.

Under the experiment and evaluation section the performance of SC tree, CATS tree and GC tree algorithm was verified. Focus is on execution performance and memory requirements under different values of transaction size attribute. Under the efficiency evaluation, the experiments focused on efficiency of tree construction and rule mining. Under the memory requirements of algorithms, the experiments focused on memory requirements for tree construction and memory requirements for association rule mining.

Table 1: Feature exploration of the models in recent literature

Reference	**Model Type**	**Candidature**	**Support Constraint**	**Support**	**Utility based**	**Multiple Supports**	**Scan Count**
[22]	Tree	No	Yes	user defined	Yes	No	Multiple
[23]	Apriori	Yes	Yes	Compute by sampling	No	No	Multiple
[25]	Tree	No	Yes	Compute by preorder Link	No	Yes	Multiple
[26]	Tree	No	No	NA	NA	NA	Multiple
[28]	Apriori	Yes	Yes	user defined	Yes	No	Multiple
[29]	Apriori	Yes	Yes	user defined	Yes	No	Multiple

[30]	Apriori	Yes	Yes	user defined	Yes	No	Two
[31]	Tree	No	Yes	user defined	Yes	No	Two
[32]	Apriori	Yes	Yes	user defined	Yes	No	Multiple
[33]	Tree	No	Yes	user defined	Yes	No	Multiple
[34]	Tree	No	Yes	user defined	Yes	No	Multiple
[35]	Tree	No	Yes	user defined	Yes	No	Multiple
[37]	Apriori	Yes	Yes	Compute by probability	No	No	Multiple
[38]	Apriori	Yes	Yes	Compute by Averages	No	No	Multiple
[39]	Apriori	Yes	Yes	Compute by Averages	No	No	Multiple
[40]	Apriori	Yes	Yes	Compute by upper bound	No	No	Multiple
[41]	Apriori	Yes	Yes	Compute by estimation	No	No	Multiple
[42]	Apriori	Yes	Yes	Compute by estimation	No	No	Multiple
[43]	Apriori	Yes	Yes	Compute algorithmically	No	No	Multiple
[47]	Apriori	Yes	Yes	User Defined	No	Yes	Multiple
[48]	Apriori	Yes	Yes	Compute by Bin Oriented	No	Yes	Multiple
[49]	Apriori	Yes	Yes	Compute by maximal constraint	No	Yes	Multiple
[50]	Apriori	Yes	Yes	Compute by maximal constraint	No	Yes	Multiple
[51]	tree	No	No	NA	NA	NA	Multiple
[52]	Tree	No	No	NA	NA	NA	Multiple
[53]	Tree	No	No	NA	NA	NA	Multiple
[54]	Tree	No	No	NA	NA	NA	Multiple
[55]	Tree	No	No	NA	NA	NA	Multiple
[56]	Tree	No	No	NA	NA	NA	Multiple
[57]	Tree	No	No	NA	NA	NA	Multiple

4. CONCLUSION

With copious research published about frequent pattern mining models that affirmed in this paper, evident that numerous significant research issues at to be resolved. The methods scalability in resource utilization is the initial concerns of the research in frequent pattern mining, which is still a considerable issue since the current trends and requirements elevated new scalability related issues and challenges. Hence we optimistic to consider this as our further research direction to settle for efficient mining methodologies. In this regard we can plot a research direction that aims to devise frequent pattern mining models that result compact frequent itemsets, which are sensible under inference validation, temporal validation, transitional validation. In regard to achieve the compactness in total number of frequent itemsets determined, few models are already available such as closed frequent itemset mining, maximal frequent itemset mining, approximate frequent itemset mining condensed frequent itemset bases, representative frequent itemset mining, utility mining of frequent itemsets, and discriminative frequent itemset mining. However, it is still a considerable dilemma that what category of models will be substantial for finding frequent itemsets that are compact and meets inference, temporal and transitional validation in regard to the source of the data. Hence it is quite remarkable to conclude that much research is still needed under consideration of the constraints such as transitional, temporal and inference validity. There is a big scope to perform research to identify post itemset mining models to determine the coherent itemsets. In this regard we continue our research to determine scalable closed itemset mining models under the constraints explored.

REFERENCES

[1] Agrawal, R., T, Imielinski and A, Swami, 1993, Mining association rules between sets of items in large databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, May 25-28, ACM, New York, USA., pp: 207-216

[2] Agrawal, R, and R, Srikant, 1994, Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very Large Data Bases, Sept, 12-15, San Francisco, CA., USA., pp: 487-499

[3]Mannila, H., H, Toivonen and A, Inkeri Verkamo, 1994 Efficient algorithms for discovering association rules Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, (KDD-94), IEEE, pp: 181-192

[4]Han, J., J, Pei, Y, Yin and R, Mao, 2004, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Mining Knowledge Discovery, 8: 53-87

[5]Savasere, A., E, Omieccinski and S, Navathe, 1995, An efficient algorithm for mining association rules in large databases, Proceedings of the 21st International Conference on Very Large Databases, Sept, 11-15, Zurich, Switzerland, pp: 432-443

[6]Toivonen, H., 1996, Sampling large databases for association rules, Proceedings of 22th International Conference on Very Large Databases, Sept, 3-6, Bombay, India, pp: 134-145

[7]Brin, S., R, Motwani and C, Silverstein, 1997, Beyond market basket: Generalizing association rules to correlations, Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, May 11-15, Tucson, AZ., pp: 265-276

[8]Hidber, C., 1999, Online association rule mining, ACM SIGMOD Rec., 28: 145-156

[9] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports,", Proceedings of the fifth ACM SIGKDD international conference, San Diego, CA, USA August 15-18, 1999, p.341

[10]. Ezeife, C.I.; Min Chen; Incremental mining of Web sequential patterns using PLWAP tree on tolerance MinSupport, Database Engineering and Applications Symposium, 2004, Issue Date: 7-9 July 2004, On page(s): 465 - 469

[11]Pei, J., J, Han and L,V,S, Lakshmanan, 2001, Mining frequent itemsets with convertible constraints, Proceedings of the 17th International Conference on Data Engineering, April 2-6, Heidelberg, Germany, pp: 433-332

[12]Liu, J., Y, Pan, K, Wang and J, Han, 2002, Mining frequent item sets by opportunistic projection, Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases, July 23-26, Edmonton, Canada, pp: 239-248

[13]Grahne, G, and J, Zhu, 2003, Efficiently using prefix-trees in mining frequent itemsets, Proceedings of the 2003 ICDM International Workshop on Frequent Itemset Mining Implementations, (IWFIMI03), Melbourne, FL., pp: 123-132

[14]Lakshmanan, L,V,S., R, Ng, J, Han and A, Pang, 1999, Optimization of constrained frequent set queries with 2-variable constraints, ACM SIGMOD Rec., 28: 157-168

[15]Grahne, G., L, Lakshmanan and X, Wang, 2000, Efficient mining of constrained correlated sets, Proceedings of the 2000 International Conference on Data Engineering, Feb, 28-March 3, San Diego, CA., pp: 512-521

[17]Bucila, C., J, Gehrke, D, Kifer and W, White, 2003, DualMiner: A dual-pruning algorithm for itemsets with constraints, Data Min, Knowl, Discov., 7: 241-272

[18]Bonchi, F., F, Giannotti, A, Mazzanti and D, Pedreschi, 2003, Exante: Anticipated data reduction in constrained pattern mining, Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Sept, 22-26, Cavtat, Dubrovnik, Croatia, pp: 59-70

[19] Gade, K., J, Wang and G, Karypis, 2004, Efficient closed pattern mining in the presence of tough block constraints, Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug, 22-25, Seattle, WA., pp: 138-147

[20]Bonchi, F, and C, Lucchese, 2004, On closed constrained frequent pattern mining, Proceedings of the 2004 International Conference on Data Mining, Nov, 1-4, Brighton, UK., pp: 35-42

[21]Yun, U, and J, Leggett, 2005, Wfim: Weighted frequent itemset mining with a weight range and a minimum weight,

Proceedings of the 2005 SIAM International Conference on Data Mining, April 21-23, Newport Beach, CA., pp: 636-640

[22] Ya-Han Hu; Fan Wu; Tzu-Wei Yen "Considering RFM-values of frequent patterns in transactional databases", 2nd International Conference on Software Engineering and Data Mining (SEDM), June 2010, pages: 422 - 427

[23] Long, Z.A. Hamdan, A.R. Bakar, A.A; Parameter setting procedure via quick parameter evaluation in frequent pattern mining for outbreak detection, 2nd Conference on Data Mining and Optimization, 2009. DMO '09, Issue Date: 27-28 Oct. 2009, On page(s): 90 - 93

[24] Antunes, C.; Pattern Mining over Star Schemas in the Onto4AR Framework, IEEE International Conference on Data Mining Workshops, 2009, ICDMW '09, Issue Date: 6-6 Dec. 2009, On page(s): 453 - 458

[25] Ya-Han Hu; Fan Wu; Yi-Chun Liao; Sequential pattern mining with multiple minimum supports: A tree based approach, 2nd International Conference on Software Engineering and Data Mining (SEDM), Issue Date: 23-25 June 2010 On page(s): 428 – 433

[26] Chuang-Kai Chiou, Judy C. R. Tseng; Sorted Compressed Tree: An Improve Method of Frequent Patterns Mining without Support Constraint, 2nd International Conference on Software Engineering and Data Mining (SEDM), 2010, Issue Date: 23-25 June 2010, On page(s): 328 - 333

[27]Agarwal, R,C., C, Aggarwal and V,V,V, Prasad, 2001, A tree projection algorithm for generation of frequent item sets, J, Parallel Distributed Comput., 61: 350-371

[28]. Yao, H., Hamilton, H.J., Butz, C.J.: A Foundational Approach to Mining Itemset Utilities from Databases. In: Third SIAM Int. Conf. on Data Mining, pp. 482–486 (2004)

[29]. Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. Data & Knowledge Engineering 59, 603–626 (2006)

[30]. Liu, Y., Liao, W.-K., Choudhary, A.: A Two Phase algorithm for fast discovery of High Utility of Itemsets. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS(LNAI), vol. 3518, pp. 689–695. Springer, Heidelberg (2005)

[31]. Erwin, A., Gopalan, R.P., Achuthan, N.R.: CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach. In: 7th IEEE Int. Conf. on Computer and Information Technology (CIT 2007), pp. 71–76 (2007)

[32]. Li, Y.-C., Yeh, J.-S., Chang, C.-C.: Isolated items discarding strategy for discovering high utility itemsets. Data & Knowledge Engineering 64, 198–217 (2008)

[33]. Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K.: CP-tree: A tree structure for single pass frequent pattern mining. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS(LNAI), vol. 5012, pp. 1022–1027. Springer, Heidelberg (2008)

[34] F. Wu, Y.-S. Lee, and J.-N. Yu, "An adaptive approach for modelselection with high stability," in Proceedings of International JointConference on e-Commerce, e-Administration, e-Society, and e-Education, Bangkok, Thailand, 2008.

[35]Y.-L. Chen, M.-H. Kuo, S.-Y. Wu, and K. Tang, "Discovering recency,frequency, and monetary (RFM) sequential patterns from customers'purchasing data," Electronic Commerce Research and Applications, vol.8, pp. 241-251, 2009.

[36] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns withoutcandidate generation: A frequent-pattern tree approach," Data Miningand Knowledge Discovery, vol. 8, pp. 53-87, 2004.

[37] S. Zhang, et al., "Computing the minimum-support for mining frequent patterns," Knowledge and Information Systems, vol. 15, no. 2, 2008, pp. 233-257

[38] G. Ramesh, W. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: theory and application. In Proceedings ACM PODS'03, pages 284–295, 2003.

[39] L. Lhote, F. Rioult, and A. Soulet. Average number of frequent (closed) patterns in bernouilli and markovian databases. In Proceedings IEEE ICDM'05, pages 713–716, 2005.

[40] F. Geerts, B. Goethals, and J. V. den Bussche. Tight upper bounds on the number of candidate patterns. ACM Trans. on Database Systems, 30(2):333–363, 2005.

[41] U. Keich and P. A. Pevzner. Subtle motifs: defining the limits of motif finding algorithms. Bioinformatics, 18(10):1382–1390, 2002.

[42] J. Besson, et al., "Parameter Tuning for Differential Mining of String Patterns," IEEE Computer Society Washington, DC, USA, 2008, pp. 77-86

[43] M. Boley, et al., "A Randomized Approach for Approximating the Number of Frequent Sets," IEEE Computer Society Washington, DC, USA, 2008, pp. 43-52.

[44]. Valiant, L.G.: The complexity of computing the permanent. Theor. Comput. Sci. 8, 189–201 (1979)

[45]. Gunopulos, D., Khardon, R., Mannila, H., Saluja, S., Toivonen, H., Sharm, R.S.: Discovering all most specific sentences. ACM Trans. Database Syst. 28(2), 140–174 (2003)

[46]. Jerrum, M., Sinclair, A.: Approximating the permanent. SIAM J. Comput. 18(6), 1149–1178 (1989)

[47] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in Proceedings of the 1999 International Conference on Knowledge Discovery and Data Mining, pp. 337-341, 1999.

[48] Y. Lee, T. Hong, and W. Lin, "Mining association rules with multiple minimum supports using maximum constraints," International Journal of Approximate Reasoning, vol. 40, pp.44-54, 2005.

[49] Y. C. Lee, T. P. Hong and W. Y. Lin, "Mining fuzzy association rules with multiple minimum supports using maximum constraints", The Eighth International Conference on Knowledge-Based Intelligent Information and Engineering Systems, 2004, Lecture Notes in Computer Science, Vol. 3214, pp. 1283-1290, 2004.

[50] Y. C. Lee, T. P. Hong and W. Y. Lin, "Mining association rules with multiple minimum supports using maximum constraints," International Journal ofApproximate Reasoning, Vol. 40, No. 1, pp. 44-54, 2005.

[51] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the 19th ACM SIGMOD International Conference on Management of Data, 2000, pp. 1–12.

[52] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, D. Yang, H-mine: hyper-structure mining of frequent patterns in large database, in: Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, 2001, pp. 441–448.

[53] M. EI-Hajj, O.R. Zaiane, Inverted matrix: efficient discovery of frequent items in large datasets in the context of interactive mining, in: The ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 109–118.

[54] G. Liu, H. Lu, J.X. Yu, CFP-tree: a compact disk-based structure for storing and querying frequent itemsets, Information Sciences 32 (2007) 295–319.

[55] T. Hu, S.Y. Sung, H. Xiong, Q. Fu, Discovery of maximum length frequent itemsets, Information Sciences 178 (2008) 68–87.

[56] W. Cheung and O.R. Zaiane, “Incremental mining of frequent patterns without candidate generation or support constraint,” Citeseer, 2003, pp. 111-116.

[57] S.Y. Liu, “An Efficiency Incremental Mining with Grouping Compress Tree,” Unpublished master’s thesis, National Central University Taoyuan Country, Taiwan, 2004.

Ms.K.Vinuthna is a Research Scholar of Jawaharlal Nehru Technological University, Hyderabad. She had received Master of Technology in Computer Science and Engineering from University college of Engineering , Osmania University, Hyderabad. Presently She is working as Associate professor in Keshav Memorial Institute of Technology, Hyderabad. Her main research interests are Data Mining and Information Retrieval Systems.

Dr. P.V.S. Srinivas is presently serving as a Professor & Head, Department of Computer Science and Engineering, at Geethanjali College of Engineering and Technology, Hyderabad. He received Masters followed by Ph.D. in computer Science and Engineering in the area of Computer Networks from JNTU Hyderabad in the year 2003 and 2009 respectively. His main research interests are Wireless Communication, Mobile Ad hoc Networks.He is also serving as a Chief Panel Consultant in the area of wireless communications for a company by name a SCADA METER SOLUTIONS Pvt Ltd, Hyderabad. He has published around forty research papers in different peer reviewed and refereed International Journals and conferences in India as well as abroad. He is also serving as an Editor-in-Chief for an International Journal IJWNC and also a peer reviewer for 3 International journals.