Frequent Pattern Mining and Current State of the Art

Kalli Srinivasa Nageswara Prasad
Research Scholar in Computer Science
Sri Venkateswara University, Tirupathi
Andhra Pradesh, India

Prof. S. Ramakrishna
Department of Computer Science
Sri Venkateswara University, Tirupathi
Andhra Pradesh, India

ABSTRACT

Identifying the association rules in large databases play a key role in data mining. The research is mainly aimed at considering prior researches, present working status and to restore the gaps between them with present known information. There are two problems regarding this context, they are identifying all frequent item sets and to generate constraints from them. Here, first problem, as it takes more processing time, is computationally costly. Consequently, many algorithms are proposed to solve this problem. Current study considers such algorithms and their related issues.

General Terms

Data Mining, Market Basket Analysis, Itemset.

Key Words

Data Mining, Association Rule Mining, Frequent Pattern Mining, Apriori Algorithm, SC Tree, CATS Tree, GC tree.

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 specified threshold, it is called a frequent structural pattern.

Identifying frequent patterns is important in mining associations and correlations. It contributed in data indexing, classification and clustering. It is proposed by Agrawal et al. [1] for market basket analysis which explores customer characteristics from the associations between objects in the basket. There are several 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. TAXONOMY

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

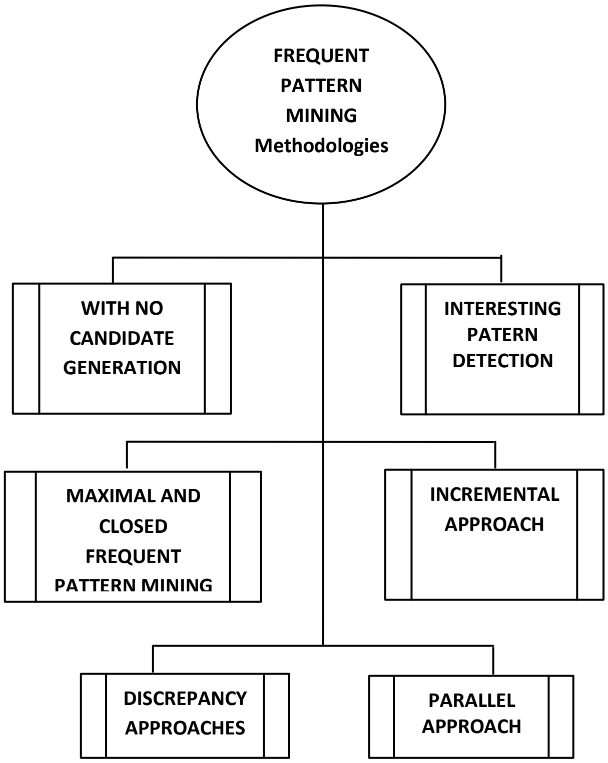
FREQUENT PATTERN MINING Methodologies
WITH NO CANDIDATE GENERATION
INTERESTING PATERN DETECTION
MAXIMAL AND CLOSED FREQUENT PATTERN MINING
INCREMENTAL APPROACH
DISCREPANCY APPROACHES
PARALLEL APPROACH

Figure 1. Frequent Pattern Mining Methodologies

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 large 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, number of database passes also decremented.

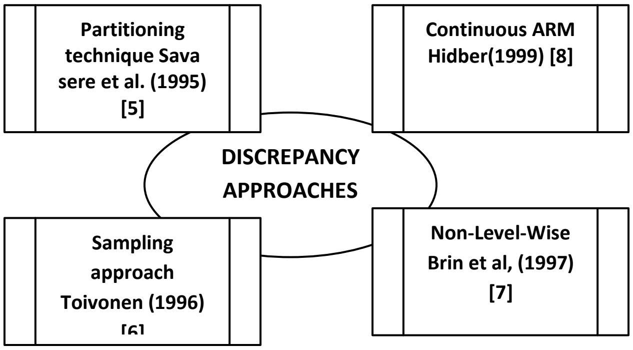
Partitioning technique Sava sere et al. (1995) [5]
Continuous ARM Hidber(1999) [8]
DISCREPANCY APPROACHES
Sampling approach Toivonen (1996) [6]
Non-Level-Wise Brin et al, (1997) [7]

Figure 2. Discrepancy approaches.

2.1 Partitioning technique

It is proposed by Sava sere 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.

2.2 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] 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 efficient than Apriori, but has less memory utilization.

3. MINING FREQUENT ITEM SETS WITHOUT CANDIDATE GENERATION

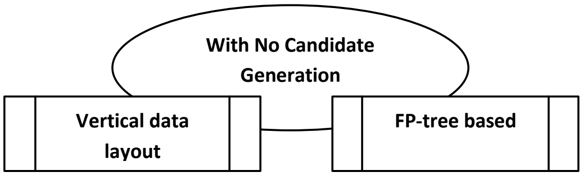

Figure 3. No candidate generation approaches.

3.1 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 develops 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 Agrawal 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].

3.2 Interesting patterns

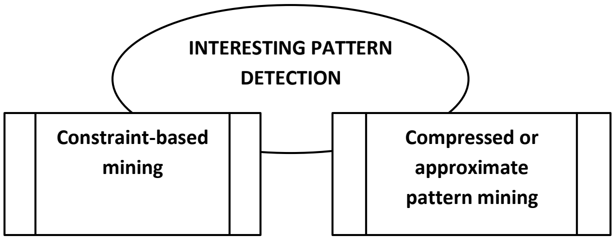
INTERESTING PATTERN DETECTION
Constraint-based mining
Compressed or approximate pattern mining

Figure 4. Interesting pattern detection approaches

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

3.3 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, succint 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.

4. CURRENT STATE OF ART

Authors Ya-Han Hu, Fan Wu, and Tzu-Wei-Yeh[22] have presented an in-detail thesis about Market based Analysis. It is an important data mining application for finding correlation between purchasing items in transactional databases. Apart from considering the constraints affecting the mining process, this study extends the RFM analysis into the mining process to measure the importance of frequent patterns that extend in mining. Here, a customer is to be recognized as valuable if his/her purchasing records are recent, frequent, and having high amount of money. Initially, the RFM patterns are determined. Then, a tree structure is proposed, named RFPM-tree, to compress and store the entire transactional database. The RFPM-growth is developed to discover all RFM-patterns from RFPM-tree. The algorithm can significantly reduce the number of discovered patterns and efficiently find the RFM-patterns.

Association rule mining is one of the prominent aspects of data mining, useful for discovering relationships among items or events in various application domains. The mining of association rules can discover the complete set of frequent patterns. But most of the patterns are not interesting to the users. It is a tedious task to filter the patterns discovered. Instead of filtering the patterns after the discovery process, if the constraints are directly pushed into the process of discovering, then the desired patterns are discovered and the rest is set aside, thereby reducing the manual labor. These constraints are specific algorithms with some regular expressions that possess meaningful information.

In this study, constraints developed from the RFM (Recency, Frequency and Monetary) analysis to examine the interestingness of frequent patterns into the mining algorithm are considered. A frequent pattern is treated as valuable if it occurs frequently in both the whole database and the recent period, and if the pattern can provide higher revenue for business. We call a pattern as a RFM-pattern if its recency score, frequency score and monetary score satisfies the minimum thresholds. Related to the work, Wu et al. discussed the recency problem about the change of data distribution between the past data and the new data. Emerging patterns as introduced by Dong and Li discover significant pattern changes in datasets with different time periods. A major drawback lies in the failure to fully reflect the requirement of RFM in the mining process. RFM Apriori defined the recency constraint of the last transaction that should satisfy the recency threshold. The paper proposed an algorithm for mining patterns satisfying RFM constraints using a fixed time gap. However their approach the different stripe lengths of recency measuring the importance of transaction. This proposed method not only gives a smooth approach but also performs the complete concept of RFM on measuring the importance of patterns. For a set I of items in a database, to measure the scores of RFM of itemsets, the database should contain the more information of a transaction X, like

<tx,(a1 , qa1) (a2 , qa2)………………(am , qam)>,

where tx stands for the transaction time when transaction X occurs, the m pairs, (ai, qai), where $1 \leq i \leq m$ and ai Є I, denotes that there are m different items are purchased in the transaction and for item ai, the purchased quantity is qai. Given an itemset Y ($Y \subseteq I$), Y is said to be contained in X if all items in Y also occur in X.

Definition 1. Assume that an itemset Y is contained in transaction X. Y's transaction recency score gained from X, denoted by trscore(Y, X), is defined as: trscore(Y, X) =

$(1\text{-}\delta)^{tcurrent\text{-}tx}$, where δ is a user-specified decay speed (δЄ[0, 1]), tcurrent denotes the current timestamp.

Definition 2. Following Definition 1, the recency score of itemset Y, denoted by Rscore(Y), is defined as the sum of transaction recency score gained from all of the transactions containing it.

Definition 3. The frequency score of itemset Y in DB, denoted by Fscore(y), is the number of transactions containing itemset Y.

Definition 4. Let p(ai) denotes the unit price of item ai. Assume that an itemset Y is contained in transaction X.

Definition 5. Following Definition 4, the monetary score of itemset Y in DB, denoted by MscoreDB(Y), is the sum of monetary score gained from transactions containing itemset Y.

Definition 6. Following the Definition 4, the transaction amount for transaction X, denoted by ta(X), is the sum of all items' monetary score in transaction Xs.

An RFMP-tree contains RFM-header and RFMP-tree. A RFM-header contains all 1-RFT-patterns that are sorted according to their Fscore in descending order. Each node in RFMP-tree consist item-name, Rscore, Fscore, Tta, parent-link, child-link, and sibling-link as fields, where item-name registers The RFMP-growth algorithm described as two step model. A complete RFMP-tree will be constructed through two database scans (briefly called DB scan). In the first DB scan, all 1-RFT-patterns will be discovered and stored them in the RFMheader to pool up. After that, all

1-RFT-patterns in RFM-header will be arranged according to their Fscore in descending order. In the second DB scan the root node of the tree will be sorted and label it as null, is created and prunes items for every transaction if item not found in RFM-Header. In the first step the pattern growth recursively and find all RFT-patterns from the RFMP-tree is performed. The second step scans database once to compute the Mscore of all RFT-patterns and finally discover all RFM-patterns.

While inserting transactions into RFMP-tree, two possible cases have been observed. If there exists a branch representing the transaction, it is compulsion to update all scores related to the transaction into the corresponding nodes. On the contrary, if there is no corresponding branch, one may have to create all necessary nodes and calculate initial scores for it/them. Once a new node is created, all the links related to this node have to be built up, including parent-link and node-links. The procedure insert node is used to perform the functions describe above.

Observations: The three thresholds used are having good impact on the proposed model. The significance of the values assumed to these thresholds is not generalized. The experiments done on synthetic datasets are justifying the threshold values considered. In my observation 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.

Zalizah Awang Long et al[23] described a design (shown in the above figure) to determine the minimum support required to determine the interesting pattern, minimum support and item set count. It is broadly divided in three stages: 1.Pre-processing

2.frequent item set algorithm and 3.data re-transformation. The pre-processing stage involves Two modules i.e selection and transformation module. 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. In the second stage the minimum support and sequence length 'k' is determined. The output of frequent mining algorithm is used as an important threshold in the outbreak detection tasks. This occurs in the third stage.

An experiment involving three dataset, ZOO, TIC TAC TOE (UCI data repositories) and third one taken from Auto lab for WSARE implementation of outbreak detection was conducted. Synthetic data is used to detect the outbreak detection. The results of this experiment showed that by lowering the minimum support, more interesting pattern can be analyzed .Based on the result, the possible values of minimum support range from 30% to 50% to minimize the uninteresting pattern. As a second task in frequent item set algorithm stage, the item set counts for each generated length. Based on the mean running collected output produce by apriori, positions of high gain are identified. The graph is analyzed based on the state (increasing, decreasing or stable).

Henceforth, from the experiment an appropriate minimum support 40% to 50% is sufficient to generate frequent itemsets. Numbers of item set required for maximum detection rate is identified on the basis of high frequency count. This paper intends to generate hypothesis and identify the potential interesting pattern from the sequence item sets count .Thereby of outbreaks occurring in the pattern are identified.

Observations: The false positive rate of the outbreak detection can be reduced by appropriate numbers of minimum support and length of item set. However approximation of the values are considered so still absolute values are to be verified. Some investigation on the detection of curve is required.

Cláudia Antunes et al[24] proposed a new approach to mine pattern among data stored as a star schema, based in a domain driver framework where available information is reproduced in domain ontology. This new approach is performed by an Aprori based algorithm D2Apriori

Any problem on the Onto4AR framework is centered on the use of an ontology and assumes a modified version of problem where it completely defined in the context of ontology, for every transactional pattern mining, algorithms receive a database composed by a set of items in regards of the individual transaction recorded. Further, these transactions are provided by knowledge base and since, knowledge base is a tuple with ontology, a set of instances and characterized by a set of features which corresponds to traditional items.

In this new context mining can be performed by adapted transactional pattern mining algorithm that is able to incorporate the available knowledge, represented by the ontology. The authors have described each of these procedures and exemplified their application in

Apriori-based algorithm- D2Apriori

The algorithm performs the task in a two step process as:-

a. Receives the dataset as input followed by reception of knowledge base and the constraints applicable.

b. Data file is then read in the context of the knowledge base, creating the dataset with the incorporation of semantics for each read transaction.

As a special feature for reduction of the time spent in the discovery process, measures adopted by the authors are:-

i. Read the set of concepts and its attributes.

ii.Propagate concept properties through the taxonomy.

Thus, this model attempted to resolve the issue of pattern mining and the discovery of information in data stored by the framework Onto4AR, which is achieved by utilizing the knowledge and the hierarchy of concepts, overcoming the results achievable through traditional pattern mainly approaches.

Observations: Though the D2Apriori algorithm fulfils the motive but yet it is unable to complement the real stars, primarily due to the heavy memory consumption resulting from generated candidates as well as the pattern discovered.

Ya-Han Hu et al[25] have laid a detailed emphasis on Sequential pattern Frequent Mining with multiple minimum supports. Frequent pattern mining is a significant data-mining methodology for determining correlations among items or itemsets. Owing to the varied frequencies of various itemsets, specifying a single minimum support cannot exactly discover interesting patterns. The authors argued that the solutions available, which are based on multiple minimum supports(MMS) [9] is not straightforward. So, based on the new definitions of sequential pattern with MMS, an extended version of the PLWAP-tree [10] structure has been proposed, named Preorder Linked Multiple Supports tree (PLMS-tree), to compress and store all necessary information from sequence databases. Then, PLWAP-Mine is further extended to develop an efficient algorithm, named Multiple Supports Candidate Pattern growth (MSCP-growth), for discovering complete set of sequential patterns with MMS. We extend PLWAP-tree and PLWAP-Mine because they have been proven to be a compact data structure and a highly efficient algorithm on sequential pattern mining.

The proposed PLWAP-tree-like algorithm called PLMS tree stores all necessary information from sequence database. After that, proposed new pattern growth algorithm, named MSCP-growth generates the complete set of sequential patterns with MMS based on the PLMS-tree. The PLWAP-tree contains only items with support not less than *minsup* while PLMS-tree retain items with support not less than MIS(*I*). In PLMS-tree we need to add some additional information in to the tree structure for discovering complete set of sequential patterns whereas such is not the case with the later.

Observations: In this thesis, the rare item problem in the mining of sequential patterns is addressed. To generate more interesting patterns, the concept of MMS on sequential pattern mining and propose an algorithm, named MSCP growth is detailed, to discover complete set of sequential pattern with MMS.

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 experimental results also show that the proposed algorithm outperforms Compressed and Arranged Transaction Sequences tree (CATS tree) and Grouping Compressed tree (GC tree) mining algorithms. Several algorithms have been proposed for association rule mining. To find satisfied collection of association rules several minimum supports should be used for repeated mining and to cope up with this drawback, several algorithms were proposed to allow the minimum support to be adjusted without rebuilding the whole data structure for frequent pattern mining. 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 improved 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.

Knowledge behind the huge amounts of data is derived by statistical data and the relationship among products. The relationships among products are not easy to find intuitively, which is the reason for data mining to become an important research topic and among the data mining methodologies "Association Rule Mining" is most widely applied technique. Apriori algorithm and Frequent patterns growth (FP growth) algorithm must configure a minimum support value before mining large item sets and an administrator must adjust the value several times to obtain the satisfied result. 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.

4.1 Frequent Patterns Growth Algorithm

The FP growth algorithm proposed by Han and et al. is the most typical algorithm which is employed for storing compressed information about frequent patterns. The techniques which improved the efficiency of mining are avoiding scanning of the database frequently, avoiding generating candidate item sets and reducing the search space. It is efficient, scalable and faster than Apriori algorithm but the difficulty of minimum support setting still exists.

4.2 Compressed and Arranged Transaction Sequences Tree Algorithm

In CATS tree algorithm proposed by William Cheung and Omar R. Zaine, there is no need for reconstruction when the minimum support is changed and it preserves all elements without reducing. When minimum support threshold is changed, the tree structure is unnecessary to be reconstructed. CATS algorithm constructs without sorting and hence, there is a need to always adjust the tree structure.

4.3 Grouping Compress Tree Algorithm

The GC tree algorithm proposed by Liou further improves the performance of CATS tree. Firstly, the transaction data is pre-processed and then compressed into GC tree data structure. Finally, they use an efficient algorithm to find out the large item set by simplifying the complexity of data construction and mining process.

4.4 Sorted Compress Tree Algorithm

This algorithm is used to simplify the process of tree construction and another is to simplify the rule mining method. The memory requirements for constructing and mining the tree structure of SC tree is also less than the other algorithms 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. SC algorithm mines the frequent item set in three stages:

i. Data processing- It deals with sorting which brings benefit for mining as in binary searching tree. The arrangement of sorted transactions is consistent. Efficiency of tree constructing and rule mining can be improved and searching space can be reduced.

ii. SC tree constructing- In this stage a sorted transaction database is obtained after which the SC tree is constructed.

iii. Rule mining in SC tree- Here nodes (items) in SC tree are sorted, so that the mining direction in SC tree can be in a single direction. In order to satisfy the constraint of minimum support, the mining direction in SC tree is bottom-up. The mining process in SC trees is filter out non-potential items, constructing conditional mining tree, generating frequent item set.

Observations: Under the experiment and evaluation section the performance of SC tree, CATS tree and GC tree algorithm is verified. Focus is on executing performance and memory requirement in different minimum support and transaction size. Under the efficiency evaluation, the experiments focus is on efficiency of tree construction and efficiency of association rule mining. Under the memory requirements of algorithms, the experiments focus on memory requirements for tree construction and memory requirements for association rule mining.

5. CONCLUSION

Here, in this session we are mainly aimed at performance and scalability. Apart from them, we are also focused on flexibility and reusability for making generation of association rules more comfortable. We found from the above discussion that finding frequent item sets is a costly process. Many algorithms are suggested to achieve required goals but most of them are multiple pass algorithms which causes more disk reads creating overhead. Thus, algorithms with single passes are introduced, but even these algorithms don't depict the importance of these patterns.

Though algorithms are proposed for maintaining association rules with every update, such facility is not available for mining incremental principles. Simultaneous algorithms are needed to fasten the computation operation. Further research is required to allow flexibility, reusability and to decrease the number of derived patterns and to retain quality. These requirements acted as the basic reasons for getting interested in this research.

Table 1: Tabular form of the approaches selected from recent literature

Approach	Aim	Description
Sorted Compressed Tree[26]	Tree based Association rule mining algorithm with no support usage	Focused on efficient tree construction for pattern discovery and association rule mining without using support parameter. Empirical studies disclosed that the model is efficient in terms of memory usage for tree construction.
Pattern Mining over Star Schemas in the Onto4AR Framework[24]	Mining patterns among data stored as a star schema, based in a domain driver framework where available information is reproduced in domain ontology.	The D2Apriori algorithm can fulfils the motive with a limitation that it is unable to complement the real stars. It is due to the heavy memory consumption by generated candidates and pattern discovered.
Sequential pattern mining with multiple minimum supports[25]	Proposed a tree based approach that uses multiple minimum supports to discover the frequent patterns that are in sequence	Rare item problem of pattern discovery in sensitive datasets has been addressed. Memory usage is not so impressive but it is negligible when considered when considered this as a solution to rare item problem
Parameter setting procedure via quick parameter evaluation in frequent pattern mining for outbreak detection[23]	Determining the minimum support required to determine the item set count and interesting patterns.	Relied on an experimental observation, which conclude that minimum support between 40 to 50% is sufficient to find frequent patterns. This observation is no evident in standard literature and required an empirical study on standard bench mark datasets. Some investigation on the detection of curve is required.
Considering RFM-values of frequent patterns in transactional databases[22]	Applying RFM analysis into the mining process to measure the importance of frequent patterns that extend in mining.	The three thresholds used are having good impact on the proposed model. The significance of the values assumed to these thresholds is not generalized.

6. 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] Sava sere, 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 aminimum 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]Agrawal, R,C., C, Agrawal and V,V,V, Prasad, 2001, A tree projection algorithm for generation of frequent item sets, J, Parallel Distributed Comput., 61: 350-371

7. AUTHORS PROFILE

Kalli Srinivasa Nageswara Prasad has completed M.Sc (Tech)., M.Sc., M.S (Software Systems)., P.G.D.C.S. He is currently pursuing Ph.D degree in the field of Data Mining at Sri Venkateswara University, Tirupati, Andhra Pradesh State, India. He has published Five papers in International journals.

Prof. S.Ramakrishna is currently working as a professor in the Department of Computer Science, College of Commerce, Management & Computer Sciences in Sri Venkateswara university, Tirupathi, Andhra Pradesh State, India. He has completed M.Sc., M.Phil. Ph.D. M.Tech(IT).He is specialized in Fluid Dynamics & Theoretical Computer Science. His area of Research includes Artificial Intelligence, Data Mining & Computer Networks. He has an experience of 23 years in Teaching field. He has published 35 Research papers in National & International Journals. He has also attended 13 national Conferences and 11 International Conferences. He has guided 14 Ph.D Scholars and 15 M.Phil Scholars.