ASSOCIATION RULE MINING ALGORITHMS FOR HIGH DIMENSIONAL DATA – A REVIEW

K.Prasanna[1] and M.Seetha[2]
[1] JNIAS Research Scholar, Asstt. Prof., Department of CSE, AITS, Rajampet, A.P., India
[2] Professor in CSE, GNITS, Hyderabad, Hyderabad. India

ABSTRACT

In this paper the association rule mining algorithms are discussed and demonstrated. Particularly, the problems of association rule mining on high dimensional data are investigated and comparison of popular association rules algorithms are presented. The paper mainly focusing on the problem of curse of dimensionality associated with data bases and algorithms. To day there are several efficient algorithms that cope with the task of Association rule mining. Actually, these algorithms are less described in terms of dimensionality. In this paper, we described to day's approaches pointing out the common aspect and differences. A comprehensive experimental study against different UCI data sets are evaluated and presented. It turns out that the run time behavior of the algorithms with regards to numerous dimensionalities, derived rules, and processing time similar to be expected. It was ascertained that as increase in the dimensionality of the databases and with varying support values, the run time performance is proven better and faster in CARPENTER, COBBLER and TD-Close algorithms.

KEYWORDS: *Data Mining, Association Rule Mining, High Dimensional Data, Carpenter, Cobbler, TD-CLOSE*

I. INTRODUCTION

1.1. Association Rules

Association Rule Mining has become one of the core data mining tasks, and has motivated tremendous interest among data mining researchers and practitioners [1]. It has an elegantly simple problem statement, that is, to find the set of all subsets of items (called itemsets) that frequently occur in many database records or transactions, and to extract the rules telling us how a subset of items influences the presence of another subset. In other words, Association rule mining finds all rules in the database that satisfies some minimum *support* and minimum *confidence* constraints [2]. There are several algorithms have been developed till today. In a database of transaction D with a set of n binary attributes(items) I , a rule defined as an implication of the form $X \rightarrow Y$ where $X,Y \in I$ and $X \cap Y$. where X and Y are called Antecedent and Consequent of the rule respectively. The support, supp(X), of an item set X is defined as the proportion of transactions in the data set which contain the item set. The confidence of a rule is defined as $Conf(X \rightarrow Y) = supp(XUY)/supp(X)$ [52].

Association rules are defined as an implication rules that informs the user about set of items, which are likely to occur in a transactional data base [1], [2]. They are advantageous to use because they are simple, intuitive and do not make assumptions of any model. Their mining requires satisfaction of user- specified minimum support and confidence from a given data base at the same time. The process of discovering association rules is decomposed into two steps: in step one, generate all the item combinations i.e. frequent itemsets whose support is greater than the user specified minimum support. In step two secondly, use the identified frequent itemsets to generate the rules that satisfy a user

specified confidence. The generation of frequent itemsets requires more effort and the rule generations are discussed.

Association rule mining on high dimensional data, now a days a high topic of research interest in many fields of data mining tasks. There are numerous data repositories which stores the data in different dimensions. Mining association rules on these data bases is a challenging issue. There is a need to find association rules on high dimensional data a more effectively for different applications and decision making. There has been work on quantifying the “usefulness" or “interestingness" of a rule [54]. What is useful or interesting is often application-dependent. The need for a human in the loop and providing tools to allow human guidance of the rule discovery process has been articulated, for example, in [55] [56].

1.2. Outline of the paper

In this paper we deal with algorithmic aspects of association rule mining on high dimensional data. In fact, a broad variety of efficient algorithms to mine association rules have been developed during the last years. We are giving the general review of basic ideas and comparisons behind association rule mining on high dimensional data. Section 2 provides an analysis of high dimensional data. Section 3 represents some detailed analysis of various association rule mining algorithms that are applied on 2-Dimensional, Multi Dimensional and High Dimensional data sets. Section 4 presents brief comparative analyses of assorted algorithms. Section 5 provides conclusions.

II. RELATED WORK

In this paper we mainly restrict ourselves to the classic association rule problem. The mining of all association rules existing in the data base D with respect to certain user interesting measures. D in this case consisting of a data set with varying attributes or dimensions in the average out a total set of 1000-10000 data set. Although the association mining is still topic of further research, during recent years many algorithms for specialised tasks have been developed: First of all, there are the approaches that enhance the association rules itself. E.g. quantitative association rules [19], generalized association rules [31] [32] and to some extent the work on sequential patterns [33][34]. Moreover there are several generalizations of the rule problem [35] [36].

In addition algorithms were developed that mine well defined subsets of the rule set according to specified dimensions or interesting measures etc, e.g. General constraints [37][38], optimized rules[39][40], maximal frequent itemsets[45], and frequent closed itemsets [46][30]. Moreover there are algorithms to mine dense data bases [42] and speed tables. These approaches are supplemented by algorithms for online mining of association rules [43] and incremental algorithms e.g. [44] [41]

III. BASIC PRELIMINARIES

3.1. Problem formulation

Let T be a discretized data table (or data set), composed of a set of rows, S = {r1, r2, …, rn,}, where ri (i = 1, …, n) is called a row ID, or rid in short. Each row corresponds to a sample consisting of k discrete values or intervals. For simplicity, we call each of this kind of values or intervals an item. We call a set of rids a rowset, and a rowset having k rids a k-rowset. Likewise, we call a set of items an itemset. A k-rowset is called large if k is no less than a user-specified threshold which is called minimum size threshold. Let TT be the transposed table of T, in which each row corresponds to an item ij and consists of a set of rids which contain ij in T. For clarity, we call each row of TT a tuple.

Fig 1 shows an example table T with 4 attributes (columns): A, B, C and D. The corresponding transposed table TT is shown in Table 2.2. For simplicity, we use number i (i = 1, 2, …, n) instead of ri to represent each rid.

Originally, we want to find all of the frequent closed itemsets which satisfy the minimum support threshold min_sup from table T. After transposing T to the transposed table TT, the constraint minimum support threshold for itemsets becomes the minimum size threshold for rowsets. Therefore, the mining task becomes finding all of the large closed rowsets which satisfy minimum size threshold min_sup from table TT.

rid	A	B	C	D
1	a1	b1	c1	d1
2	a1	b1	c2	d2
3	a1	b1	c1	d2
4	a2	b1	c2	d2
5	a2	b2	c2	d3

Fig 1: an example table T

3.2. Analysis of High Dimensional Data

The emergence of various new application domains, such as bioinformatics and e-commerce, underscores the need for analyzing high dimensional data. Many organizations have enormous amounts of data containing valuable information for running and building a decision making system. Extracting the value of that data is a challenge. First and foremost is to understand and analyze these large amount data for effective decision making. A study on mining large databases is presented in [3]. Generally, in a gene expression microarray data set, there could be tens or hundreds of dimensions, each of which corresponds to an experimental condition. In a customer purchase behaviour data set, there may be up to hundreds of thousands of merchandizes, each of which is mapped to a dimension. Researchers and practitioners are very eager in analyzing these data sets. However, before analyzing the data mining models, the researcher will analyze the challenges of attribute selection, the curse of dimensionality, the specification of similarity in high dimensional space for analyzing high dimensional data set. Association Rule Mining in high dimensional spaces presents tremendous difficulty in generating the rules, much better than in predictive mining. Attribute selection is a one which reduces the impact of high dimensional data space at the time of generating rules.

Dimensionality curse is a loose way of speaking about lack of data separation in high dimensional space [4], [5], and [6]. The complexity of many existing data mining algorithms is exponential with respect to the number of dimensions [4]. With increasing dimensionality, these algorithms soon become computationally intractable and therefore inapplicable in many real applications.

3.3. Dimensionality Reduction

In general, there are two approaches that are used for dimensionality reduction:
(1) Attribute Transformation
(2) Attribute Decomposition.

Attribute Transformations are simple function of existent attributes. For sales profiles and OLAP-type data, roll ups as sums or averages over time intervals can be used. In multivariate attribute selection can be carried out by using Principle Component Analysis (PCA) [7].

Attribute Decomposition is a process of dividing data into subsets. Using some similarity measures, so that the high dimensional computation happens over smaller data sets [8]. Dimensions stay the same, but the costs are reduced. This approach targets the situation of high dimensions, large data. It was proven that, for any point in a high dimensional space, the expected gap between the Euclidean distance to the closest neighbour and that to the farthest point shrink as the dimensionality grows [6]. This phenomenon may render many data mining tasks ineffective and fragile because the model becomes vulnerable to the presence of noise.

IV. ALGORITHMS OF ASSOCIATION RULE MINING ON HIGH DIMENSIONAL DATA

In this section, we briefly describe and systematize the most common algorithms. We do this by referring to the fundamentals of frequent itemset generation that was defined in the previous section. Our goat is not to go too much into detail but to show the basic principles and differences between the approaches.

The Association rule mining (ARM) research mainly focuses on discovery of patterns and algorithms. In general, most of the discovered huge numbers of patterns are actually obvious, redundant, and useless or uninteresting to the user. Techniques are needed to identify only the useful/interesting patterns and present them to the user. The data items and transactions are organized in high dimensional space; it is natural to extend mining frequent itemsets and their corresponding association rules to high dimensional space. High dimensional association rules involve more number of dimensions or predicate. In many applications, it is difficult to find strong associations among data items at high dimensions of abstraction due to the sparsity of data and may need of commonsense knowledge representation.

The remainder of this chapter let us see several state-of-art algorithms for mining association rules form high-dimensional data. The discussion starts with various algorithms applied on two dimensional data sets followed by algorithms on multidimensional data sets and finally ends with algorithms applied on high dimensional data.

4.1 Apriori Algorithms

The Apriori algorithm is based on the above-mentioned steps of frequent itemsets and rule generation phases [1], [9]. It is applied on 2-Dimensional data set. Frequent itemsets are generated in two steps. In the first step all possible combination of items, called the candidate itemset (Ck) is generated. In the second step, support of each candidate itemset is counted and those itemsets that have support values greater than the user-specified minimum support from the frequent itemset (Fk). In this algorithm the database is scanned multiple times and the number of scans cannot be determined in advance.

The AprioriTID algorithm is a variation to basic Apriori algorithm [1], [9]. It used to determine the candidate itemsets before each pass begins the main interesting feature of this algorithm is that it does not use the database for support counting after the first pass. In this it uses a set with Ck with the elements Tid and a large k-itemsets present in the transactions with the associated identifier Tid. If a transaction does not contain a k-itemsets then the set will not have any entry for that transaction.

The SETM, the Set Oriented Mining [10], [11]. This uses the SQL join operations for candidate generation [53]. It can also be used k-way join operations for candidate generation [12]. The candidate item set with the TID of generating transaction is stored as a sequential structure, which is used for support counting.

Apriori can be combined with BFS and DFS. In BFS it scans the data base once for every candidate set size k [51]. When using DFS the candidate sets consists only of the item sets of on of the nodes in the tree. The simple combination of DFS with counting occurrences is therefore no practical relevance [50].

4.2. Partition Algorithm

The Partition algorithm differs from the Apriori algorithm in terms of the number of database scans [13]. The partition algorithm scans the database at most twice. The algorithm is inherently parallel in nature and can be parallelized with minimal communication and synchronization between the processing nodes the algorithm is divided into 2 phases: i) during the first phase, the database is divided into n non-overlapping partitions and the frequent itemsets for each partition are generated. ii) In the second phase, all the local large itemsets are merged to form the global candidate itemsets and a second scan of the database is made to generate the final counts.

The partition algorithm is designed to generate rules in parallel and utilize the power of a number of processors. It is used to aggregate the power and memory of many processors. The data in the form of a single file is distributed among different processors with each processor generating itemsets for that part of the data in parallel. This would require the passing of intermediate information among processors to generate the global rules. The parallelized partition algorithm, although developed for multiple processors, can be adapted for multiple database scenarios where data is distributed over multiple databases

4.3. Parallel Mining of Association Rules

Although Apriori is a simplest sequential ARM algorithm designed so far but it has some limitations like large number of candidate itemsets were generated that scans database at every step. To

overcome these demerits several parallel algorithms are used to discover the association rules [14]. In spite of the significance of the association rule mining and in particular the generation of frequent itemsets, few advances have been made on parallelizing association rue mining algorithms [49] [48]. Most of these algorithms are based on Shared memory Multiprocessor (SMP)[14] architecture based on Apriori like algorithms.

These algorithms are designed on different platforms i.e. shared memory system (SMS) and distributed memory system (DMS). CCPD (common candidate partition data base) and PCCD (Partitioned Candidate Common Database) proposed on a shared memory system. Three Apriori based parallel algorithms are used to mine association rules in parallel mining approach. They are as follows:

The *Count Distribution* algorithm is a straight-forward parallelization of *Apriori*. Each processor generates the partial support of all candidate itemsets from its local database partition. At the end of the each iteration the global supports are generated by exchanging the partial supports among all the processors.

The *Data Distribution* algorithm partitions the candidates into disjoint sets, which are assigned to different processors. However to generate the global support each processor must scan the entire database (its local partition, and all the remote partitions) in all iterations. It thus suffers from huge communication overhead.

The *Candidate Distribution* algorithm also partitions the candidates, but it selectively replicates the database, so that each processor proceeds independently. The local database partitions are still scanned at each and every iteration.

4.4. Incremental Mining of Association Rules

The Incremental mining algorithm is used to find new frequent itemsets [15], [16], [17]. It requires minimal recompilation when new transactions are added to or deleted from the transaction database. Incremental mining algorithm is combined with negative border approach to find association rules on high dimensional data attributes using sampling on large databases. The negative border consists of all itemsets that were candidates, which did not have the minimum support [18]. During each pass of the Apriori algorithm, the set of candidate itemsets Ck is computed from the frequent itemsets Fk-1 in the join and prune steps of the algorithm. The negative border is the set of all those itemsets that were candidates in the k^{th} pass but did not satisfy the user specified support. The algorithm uses a full scan of the whole database only if the negative border of the frequent itemsets expands. Another variant for incremental mining of association rules proposed in [58]. Incremental clustering can be used along with association rule mining to acquire knowledge which can be further under study.

4.5. Multi Dimensional Association Rules

Multi dimensional association rule mining carried on multi dimensional datasets which stores the data in more than one dimensions or predicates [19]. Multidimensional association rules are used to mine when the data or transactions are located in multidimensional space, such as in a relational database or data warehouse. Multiple dimensional association rule mining is to discovery the correlation between different predicates/attributes [20]. Each attribute/predicate is called a dimension. The data may contain different types of data such as categorical, Boolean, numeric. The attributes are also called as quantitative attributes. Association rule mining on these attributes can be carried both in static and dynamic discretization methods [21].

More research work is carried under Dynamic discretization method. In this the numeric attributes are dynamically discretized during the mining process so as to satisfy the mining criteria. Predicate attributes or values in the dimensions are converted into Boolean values so as to carry the mining process very effectively [22]. It is carried on following steps. 1) Determine the number of partitions for each predicate or attribute. 2) Generation of frequent itemsets. 3) Generating interesting rules using minconf. Fabulous research work is carried under Quantitative association rules ranging from low-level predicates to multi dimensional predicate and used for different range of application.

4.6. CARPENTER Algorithm

A new method for finding closed patterns in high-dimensional biological datasets, called CARPENTER [23]. This integrates the advantages of vertical data formats and pattern growth

methods to discover stored patterns in data sets. By converting data into vertical data format {item: TID_set}, the TID_set can be viewed as rowset and the FP-tree so constructed can be viewed as a row enumeration tree. CARPENTER conducts a depth-first traversal of the row enumeration tree, and checks each rowset corresponding to the node visited to see whether it is frequent and closed.

4.7. COBBLER Algorithm

A COBBLER algorithm is used to find frequent closed itemset by integrating row enumeration with column enumeration [24]. Its efficiency has been demonstrated in experiments on a data set with high dimension and a relatively large number of rows. Mining frequent itemsets in the presence of noise in the large data sets is another issue to consider [25]. Since the data set may contain noisy and inconsistent data, which may affect the performance at the time of mining. Several algorithms and analysis work is presented to mine frequent item.

4.8. TD-Close Algorithm

TD-Close used to find the complete set of frequent closed patterns in high dimensional data [26]. It exploits a new search strategy, top-down mining, by starting from the maximal rowset, integrated with a novel row enumeration tree, which makes full use of the pruning power of the min_sup threshold to cut down the search space. A new algorithm for discovering maximal frequent set [59] is combined along with TD-Close to find efficient frequent set.

4.9. Pattern-Fusion Algorithm

Furthermore, an effective closeness-checking method is also developed that avoids scanning the dataset multiple times. Even with various kinds of enhancements, the above frequent, closed and maximal pattern mining algorithms still encounter challenges at mining rather large called *colossal* patterns [27], since the process will need to generate an explosive number of smaller frequent patterns. Colossal patterns are critical to many applications, especially in domains like bioinformatics
A novel mining approach is investigated, called Pattern-Fusion [28], which efficiently finds a good approximation, to colossal patterns. With Pattern-Fusion, a colossal pattern [27], is discovered by fusing its small fragments in one step, whereas the incremental pattern-growth mining strategies, such as those adopted in *Apriori* and *FP-growth*, have to examine a large number of mid-sized ones. Mining partial periodic patterns in time series databases is another interesting study [57]. This property distinguishes Pattern-Fusion from existing frequent pattern mining approaches and draws a new mining methodology. Further extensions on this methodology are currently under investigation. Long patterns series from huge databases can be mined efficiently using [60].

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section we compare the algorithms and explain the observed differences in performance behaviour.

5.1. Experimental Results

To carry out performance studies we implemented the common algorithms to mine association rules starting with Apriori to multi and high dimensional algorithms like CARPENTER and COBBLER algorithms.
To evaluate the efficiency of the algorithms varying with different dimensionality along with the Apriori algorithm is implemented at the same condition. We use different synthetic data sets and real data set sample Basket Sales contain 3 to 100 dimensions, Car Equipment Data, Lung Cancer(LC), ALL, OC, POS, Thrombin database which contains 3 to 15000 dimensions. The commonly used data sets are show in figure 2 and figure3. We performed our experiments using a Pentium IV 1, 8 Gigahertz CPU with 512MB.
The experiments in figure 4to figure 7 were carried out on a synthetic data sets from [2][47]. These data sets were generated with a data generator [2] that simulates the buying behaviour of customers in retail business' Data set " T10.I4.D100K" means an average transaction size of 10, an average size of the maximal potentially frequent itemsets of 4 and 100,000 generated transactions. The number of patterns was set to 2,000 and the number of items to 1,000.

In addition to the experiments from [2][47], we restricted the maximal length of generated itemsets from 1 up to 9 on the data set "T20.I4.D100K" at min_sup= 0.33%, c.f. figure 8, 9, 10 shows the behaviour of the algorithms on real world applications. The basket data consists of about 70,000 customer transactions with approximately 60,000 different items. The average transaction size is 10.5 items. The car equipment data contains information about 700,000 cars with about 6,000 items. In the average of 20 items are assigned to each car. It is ascertaining that all the algorithms scale linearly with the data base size.

Data set	#item	#row	Avg row length
thrombin	13935	1316	29745
POS	1657	595597	164
synthetic data	100000	15000	1700

Fig 2: Data set for COBBLER,

Data set	#item	#row	No. Of. Dimensions
LC	1000	33	12533
ALL	10000	32	2173
OC	1000	253	532

Fig 3: Data set for CARPENTER, and TD_Close

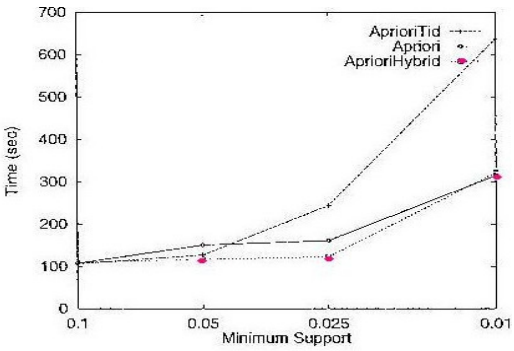

Fig 4: T10.I4.D100K

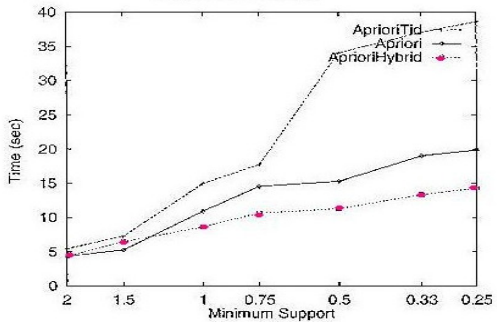

Fig 5: T20.I4.D100K

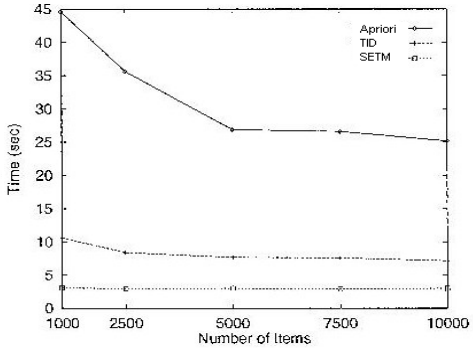

Fig 6: Basket data

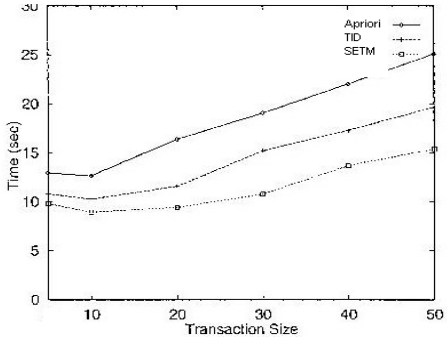

Fig 7: Car Equipment Data

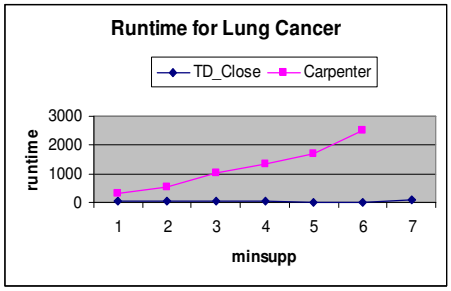

Fig 8: Run time for Lung Cancer

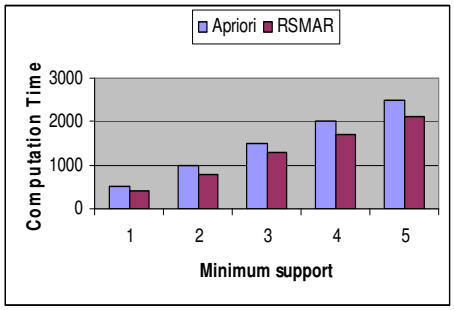

Fig 9: Run time for Sales data with 3 dimensions

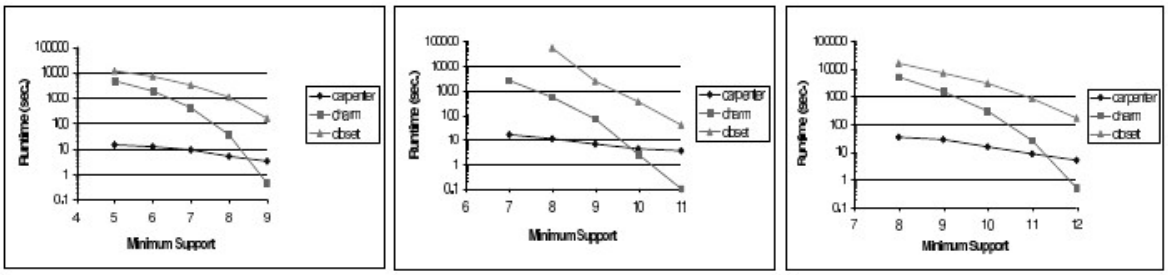

Fig 10: Varying min_sup with l=0.6: a) LC, b) ALL, c) OC

5.2. Comparative Study and Discussion

The several approaches to mine association rules and the need of high dimensional data analysis were discussed on high dimensional data. It represents major issues in analyzing the high dimensional data and their importance. Association rule mining algorithms such as *Apriori* and *AprioriTID* etc on two dimensional data, RSMAR, Fuzzy, and QARM algorithms on Multi dimensional data, CARPERNTER, COBBLER, and TD-CLOSE etc algorithms on high dimensional data are used to find patterns form the databases. They strive to provide approximate solutions on low cost and improve the performance. Their performance categorizations are as follows.

The basic *Apriori, AprioriTID* [1] [9] and *SETM* [10] algorithms ascertain that they are efficient for mining association rules on two dimensional data. But they suffer with multiple scans over the database and as the database size increases it shows effect on performance.

The Partition Algorithm [13] for mining of Association Rules is better than the basic approach in reducing the number scans and increasing the efficiency when the data base size is increase. Six synthetic datasets of varying complexity are generated. These data sets are varying from very few transactions with few frequent itemsets to larger transactions with more number of frequent itemsets. The Incremental Mining of Association rules are proved efficient scalable over the earlier *Apriori* Algorithms [18]. It is observed that the number of passes or scans over the database is reduced with respect to varying number of rows and frequent patterns. Its performance evaluation is carried out on large databases with various dimensionalities and proved efficient in finding frequent patterns.

The performance of Association rule mining on multidimensional dataset was discussed on RSMAR [22], and QARM [29]. It is observed that RSMAR performs better and more rapid than basic traditional *Apriori,* which decreases the number of database scans and reduces the computations.

CARPENTER is much faster that earlier CHARM [3] and CLOSET [30], in finding frequent closed patterns on data sets with small number of rows and large number of features represented as dimensions. CARPENTER outperforms better in running time and faster by varying the minimum support and minimum length ratio [23]. It is observed that it outperforms better run time at higher support level i.e. as *min_sup* is decreased and much faster in finding frequent patterns.

COBBLER is much faster than CARPENTER in finding frequent closed patterns on a data set with large number of rows and features or dimensions [24]. It performs better in runtime by varying

minimum support and row ratio. COBBLER performs better run time and faster when *min_sup* is relatively high and best when it decreased.
The TD-Close algorithm is much faster than CARPENTER after discretization using entropy based method. This is due to the decrease in min_sup as the dimensionality is reduced. This further reduces the runtime of the TD-Close algorithm.

VI. CONCLUSION

The purpose of this article is to present a comprehensive classification of different Association rule mining techniques for high dimensional data. Association rule mining has gained considerable prominence in the data mining community because of its capability of being used as an important tool for knowledge discovery and effective decision making. Association rules are of interest to both database community and data mining users. Since data items and transactions are organized in high dimensional space, it is natural to extend mining frequent itemsets and their corresponding association rules to high dimensional space. Due to the huge accumulation in the data by day to day computations it makes attention to the researcher to study several issues pertaining to rule discovery particularly when data is represented in high dimensional space. As the number of dimensions increase, many Association rule mining techniques begin to suffer from the curse of dimensionality, de-grading the quality of the results.
In this paper, the algorithmic aspects of association rule mining on high dimensional data are presented. From the broad variety of efficient algorithms that were developed are compared the most important one. The algorithms are systemized and analyzed their performance based on both run time experiments and theoretical considerations. . From the above discussion it is observed that the current techniques will suffers with many problems. It is also observed that as the diversity in databases and numerous dimensionalities of the databases, the various algorithms are proven to be efficient in finding frequent patterns in terms of run time as the decrease in *support* values. The COBBLER, CARPENTER and TD-Close are the better and much faster algorithms in finding the frequent patterns over the high dimensional data spaces.

REFERENCES

[1] Agrawal R,. Mannila H,. Srikant R,. Toivonen H, and Inkeri Verkamo A. “Fast discovery of association rules”. *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, Menlo Park, CA, 1995.

[2] Agrawal, R. and Srikant, R. “Fast algorithms for mining association rules”. In: Proceedings of the 1994 *International Conference On Very Large Data Bases* (VLDB’94), Santiago, Chile, pp 487–499, 1994.

[3] M J Zaki and C J Hsiao,” CHARM- an Efficient algorithm for closed itemset mining, in the proceedings of *SDM 2002*, p 457-473., 2002

[4] Aggrawal, C.C., Hinnwburg, A., and Keim, D.A. “On the surprising behavior of distance metrics in high dimensional space”. *IBM Research report*, RC 21739, 2000.

[5] Beyer K., Goldstein, J.Ramakrishnan, R., and Shaft, U. “When is nearest neighbor meaningful?” In Proceedings of the 7th *ICDT,* Jerusalem, Israel. 1999.

[6] Beyer K and Ramakrishnan. “Bottom-up computation of sparse and iceberg cubes”. In: Proceeding of the *ACM-SIGMOD 1999 International Conference on Management of Data (SIGMOD’99)*”, Philadelphia, PA, pp 359–370, 1999.

[7] Mardia.K, Kent, J and Bibby.J.”Multivariate Analysis”. *Academic Press,* 1980.

[8] McCullum. A., Nigam, K., and Ungar, L.H.” Efficient clustering of high dimensional data sets with application to reference matching”. In proceedings of the *6th ACM SIGKDD*, 167-178, Boston., MA, 2000.

[9] Thomas, S. and S. Chakravarthy. “Incremental Mining of Constrained Associations”. In the 7th *International Conference of High Performance Computing* (HiPC). 1998.

[10] Maurice Houtsma., Arun Swami.,” Set-Oriented Data Mining in Relational Databases”, in the Elsevier proceedings *on Data & Knowledge Engineering* , pages 245-262, 1995.

[11] Han, J., “DMQL: A data mining query language for relational database.”. In the proceedings of *ACM SIGMOD workshop on research issues on data mining and knowledge discovery*. Montreal. 1996.

[12] Mishra, P. and S. Chakravarthy,” Evaluation of K-way Join and its variants for Association Rule Mining”. *Information and Technology Lab* at the University of Texas at Arlington, TX. 2002.

[13] Shenoy, P."Turbo-charging Vertical Mining of Large Databases". In *ACM SIGMOD International Conference on Management of Data.* Dallas. 2000.

[14] Zaki MJ, Parthasarathy S, Ogihara M, Li W, "Parallel algorithm for discovery of association rules". *Data mining and knowledge discovery*, 1:343–374., 1997.

[15] Thuraisingham, B., " A Primer for Understanding and Applying Data Mining". *IEEE, 2000*. Vol. 2, No.1: p. 28-31.2000.

[16] Agrawal R." Mining association rules between sets of items in large databases". In: Proceedings of the *ACM-SIGMODinternational conference on management of data* (SIGMOD'93)", Washington, DC, pp 207–216, 1993.

[17] Wei-Guang Teng and Ming-Syan Chen," Incremental Mining on Association Rules".

[18] H.Toivonen, "Sampling large databases for association rules". In: Proceeding of the 1996 *international conference on Very Large Data Bases* (VLDB'96), Bombay, India, pp 134–145, 1996.

[19] Srikant R, Agrawal R "Mining quantitative association rules in large relational table", in the proceedings of *ACM SIGMOD international conference on management of data*, p.1-12 1996.

[20] Thiwari K., Mitra P., Gupta N., Mangal N .," Mining Quantitative Association Rules in Protein Sequences", in the proceedings of *Data Mining, LNAI* 3755, PP 273-281., 2006.

[21] Zhao Q and Bhowmick, S S" Association Rule mining – A Survey",

[22] Anjana P, Kamalraj P,"Rough set model for discovering Multidimensional association rules", in the proceedings of *IJCSN*S, VOL 9, no.6,p 159-164, 2009.

[23] Pan F, Cong G, Tung AKH, Yang J, Zaki M." CARPENTER: finding closed patterns in long biological datasets". In: Proceeding of the *2003 30.ACMSIGKDD international conference on knowledge discovery and data mining* (KDD'03), Washington, DC, pp 637–642, 2003.

[24] Pan F, Tung AKH, Cong G, Xu X,"COBBLER: combining column and row enumeration for closed pattern discovery". In: Proceeding of the *2004 international conference on scientific and statistical database management (SSDBM'04)*, Santorin Island, Greece, pp 21–30, 2004.

[25] Paulsen S, Sun X, Wang W, Nobel A, Prins J." Mining approximate frequent itemsets in the presence of noise: algorithm and analysis". In: Proceeding of the *2006 SIAM international conference on data mining (SDM'06)*, Bethesda, MD, pp 405–416, 2006.

[26] Han, Liu, H. D. Xin and Z. Shao. "Mining frequent patterns on very high dimensional data: A top down row enumeration approach.". Proceedings of the *2006 SIAM International Conference on Data Mining, (ICDM2006)*, Bethesda, MD., pp: 280-291.2006.

[27] Yan X, Han J, Yu PS, Cheng H . "Mining colossal frequent patterns by core pattern fusion". In: Proceeding of the 2007 *International Conference On Data Engineering (Icde'07),* Istanbul, Turkey, 2007.

[28] Zhu .F, Yan.X, Han.J, Philip S Yu. "Mining Colossal Frequent Patterns by core pattern fusion". In proceeding of 2007 *International Conference on Data Engineering*, 2007.

[29] Aumann, Y. Lindell, Y. "A Statistical Theory for Quantitative Association Rules." Proceedings of the Fifth *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 261 -270, 1999.

[30] J Pei., J.Han., R.Mao ," CLOSET: An Efficient Algorithm for mining Frequent closed Itemsets", in proceedings of the 2000ACM SIGMOD internatiional conference on Management of Data, Dallas, Texas, USA p 21-30., 2000.

[31] R.Srikant and R.Agrawal ," Mining generalized assoiction rules", in proceeding of the 21 st conference on very large databases(VLDB'95), Zurich, Switzerland september 1995.

[32] J Hipp, A Myka, R Wirth and U . Guntzer," a new algorithm for faster mining of generalized assocition rules", in proceedings fo the 2nd European Symposium on Priciples of data mining and knowledge discovery (PKDD'98), Nantes, France, September 1998.

[33] R Agrawal land R. Srikant ," Mining Sequential patterns:, in Proceedings of the internation conference on data Engineering (ICDE) Taipei, Taiwan, March 1995.

[34] H Mannila, H Toivonen and I. Verkamo," Discovery of Frequet episodes in event sequences", Daa mining and Knowledge discovery, 1(3), November 1997.

[35] R Motwani, E.Cohen, M. Datar, S.Fujiware, A.Gionix, P.Indyk, J.D. Ullman and C.?Yang, " finding intersitng assoication without support prunning ", in Proceedings of the 16th international conference on data engineering (ICDE), IEEE 2000.

[36] D.Tsur,J.D.Ullman,S.Abitboul, C.Clifton, R.Motwqani, S.Nestorov, and A.Rosenthal, " Query flocks: A generalization of assocition rule mining", in proceedings of 1998 ACM SIGMOD international conference on Management of data, Seattle, Washington, USA, June 1998.

[37] R.Ng, L.S.Lakshmanan, J. Han and A. Pang, "Exploratory mining and pruning optimizations of constrained assoication rules", in proceedings of 1998 ACM SIGMOD international conference on Management of data, Seattle, Washington, USA, June 1998.

[38] R. Srikant , Q.Vu and R.Agrawal," Mining association rules with item constraints", in proceedings fo the 3rd inernational conference on KDD and Data Mining (KDD'97),Newport Beach, California, August 1997.

[39] T. Fukuda, Y. Morimoto, S.Morishta and T.Tokuyama, " Mining optimized assocition rules for numeric attributes", in proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Pricnciples of Database Systems(PODS'96), Montral, Canada, June 1996.

[40] R. Rastogi and K.Shim," Mining optimezed support rules for numeric attribytes", in proceedings of the 15th international conference on Data engineering, Sydney, Australia, March 1999, IEEE Computer society Press.

[41] 4N.F. Ayan, A.U.Tansel and E.Arkun," An efficient algorithm to update large itemets with early pruning", in proceedings of the 5th international conference on knowledge discovery and data mining(KDD'99), San diego, California, USA, August 1999.

[42] 5R.J. BAyardo Jr.,R.Agrawal and D.Gunopulos," Constraint-based rule mining in large, dense databases", in proceedings of the 15th international conference on Data engineering , Sydney, Australia, March 1999.

[43] C Hidber," Online Association Rule mining", in proceedings fo the 1999 ACM SIGMOD Conference on Management of Data, 19899.

[44] S.Thomas, S.Bodagala, K Alsabri and S.Ranka ," An efficient algorithm for the incremental updation of association rules in large databases. In proceedings of the 3rd international conference on KDD and Data mining (KDD'97), Newportbeach, California, August 1997.

[45] M. J.Zaki , S.Parthasarathy ,M.Ogihara, W.Li," New algorithm for fast discovery of assocition rues", in proceedings of the 3rd intenational conference on KDD and Data Mining (KDD'97), Newportbeach, California, August 1997.

[46] N.Pasquier, Y.Bastide,R.Taouil,and L.Lakhal," Discovering frequent closed itemsets for association rules", in proceedings of the 7th inernational conference on database theory (ICDT'99), Jerusalem, Israel, January 1999.

[47] A.Savasere, E.Omiecinski, and S.Navathe. An efficient algorithm for mining associaton rules in large databases", in the proceedings of the 21st conference on very large databases (VLDB'95), Zurich, Switzerland, September 1995.

[48] .R. Agrawal and J. C. Shafer. Parallel mining of associationrules: Design, implementation, and experience. *IEEE Trans.Knowledge and Data Engineering*, 8:962–969, 1996.

[49] D. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In *Proc.1996 Int. Conf. Parallel and Distributed Information Systems*, pages 31–44, Miami Beach, Florida, Dec. 1996.

[50] J.Hipp, U.Guntzer, and G.Nakhaeizadeh," Mining assocition rules: Deriving a superior algorithm by analylzing today's approaches", in the proceedings of the 4th European conference on Priciples and Practices of knoledge Discovery, Lyon, France, September 2000.

[51] S.Brin, R.Motwani, J.D. Ullman, and S.Tsur. " Dynamic itemset counting and impliction rules for market basket data", in the proceedings fo ACM SIGMOD international conference on Management of Data, 1997.

[52] M. Holsheimer and A. Siebes. Data mining: The search for knowledge in databases. Technical Report CS-R9406, CWI, Netherlands, 1994.

[53] M. Houtsma and A. Swami. Set-oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, San Jose, Cali-fornia, October 1993. G.

[54] Piatestsky-Shapiro. Discovery, analy- sis, and presentation of strong rules. In G. Piatestsky-Shapiro, editor, Knowledge Dis- covery in Databases. AAAI/MIT Press, 1991.

[55] R. Brachman et al. Integrated support for data archaeology. In AAAI-93 Workshop on Knowledge Discovery in Databases, July 1993.

[56] R. Krishnamurthy and T. Imielinski. Practitioner problems in need of database research: Research directions in knowledge discovery. SIGMOD RECORD, 20(3):76-78, September 1991.

[57] J.Han, G.Dong, and Y.Yin," Efficient mining of partial periodic patterns in time series database", In ICDE'99 pp.106-115,Sydney, Austrialia, April 1999.

[58] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2(2), 1987.

[59] D-I. Lin and Z. M. Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. In *6thIntl. Conf. Extending Database Technology*, March 1998.

[60] R. J. Bayardo. Efficiently mining long patterns from databases. In *ACM SIGMOD Conf. Management of Data*,June 1998.

Authors

K. Prasanna is B.Tech and M.Tech. in Computer Science and Engineering. He is a JNIAS Research Scholar and pursuing Ph.D. in same theme from JNTUH-Hyderabad, India. Currently he is working as Assistant Professor in the Department of Computer Science and Engineering, AITS, Rajampet, Kadapa, AP, INDIA. He is having 7 years of teaching experience. His research interests are Data mining, Neural Networks, Network Security and Mobile computing, Data Structures and Algorithms.

M. Seetha, is B.Tech in ECE from Nagarjuna University, Guntur, India in 1992, M.S.(Software Systems) from BITS, Pilani, India in 1999., and Ph.D in Computer Science Engineering discipline from JNTU Hyderabad, India in 2007. She has 16 years of teaching experience. Presently, she is Professor in Computer Science and Engineering Department at GNITS, Hyderabad, India. Her areas of interest are Information Processing, Data Mining, NLP, Neural Networks and Computer Architectures. She has published more than 25 research papers in reputed journals and conferences.