AN EFFICIENT PATTERN MINING ANALYSIS IN HEALTH CARE DATABASE

[1]E.RAMARAJ, [2]N.VENKATESAN

[1]Director and Head, Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu.
Email: dr_ramaraj@yahoo.co.in

[2]Asst.Prof and Head, Dept of IT, Bharathiyar College of Engg and Tech, Karaikal, Pondichery
Email: envenki@sify.com

ABSTRACT

Association Rules are discovered by identifying relationships among sets of items in a transaction database with two measures which quantify the support and confidence of the rule. Finding frequent itemsets is computationally the most expensive step in Association Rule discovery and therefore, it has attracted significant research attention. This paper reviews Apriori related and Eclat algorithms with detailed discussion about various data structures. Computation are made for our own surveyed data sets and compared. The analysis ends with various research issues like types of rules, execution time and space complexity of algorithms.

Key words: *Data Mining, Apriori algorithm, eclat algorithm, Association Rules, data structure*

I. INTRODUCTION

Data mining is an Artificial Intelligence (AI) powered tool that can discover useful information within a database that can then be used to improve the action. ***Data Mining [1],*** also called as data archeology, data dredging, data harvesting, is the process of extracting hidden knowledge from large volumes of raw data and using it to make crucial business decisions. The steps in the knowledge discovery process include pre mining task as data cleaning and data integration, as well as post mining task such as pattern evaluation and knowledge representation. Many types of "interesting patterns" have been identified in the various research literatures and association rule constitute one such type. Data mining tasks to find these various pattern include characterization, discrimination, association analysis, classification and regression, cluster analysis, outlier analysis and evolution analysis.

Association Rule mining [2], one of the most important and well researched techniques of data mining was first introduced in [3]. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in market database, fraudulent data, etc. Implementing Association Rule algorithms to medical database is a new approach to extract hidden knowledge. Various algorithms are emerged to solve the problem of associations.

Most of the previous research works for finding frequent itemsets [3][4][5[7] are concentrated that the problem of discovering association rules is decomposed into two sub problems. First is finding all frequent itemset [12][31] in the database, second construct the association rule using frequent item set.

Association rule mining finds interesting association or correlation relationship among a large set of data items with massive amounts of data continuously being collected and stored, many industries are becoming interested in mining association rules from their databases. Let D be a set of n transactions such that $D=\{T_1, T_2, T_3,..,T_n\}$, where $T_i=I$ and I is a set of items, $I = (i_1, i_2, i_3, .. ,i_m\}$. A subset of I containing k items is called a k-itemset. Let X and Y be two itemsets such that $X \subset I$, $Y\subset I$, and $X \cap Y= \phi$. An association rule is an implication denoted by X=>Y where X is called antecedent and Y is called the consequent. Given an itemset X, support s(X) is defined as the fraction of transactions $T_i \in D$ such that $X\subseteq T_i$. Consider P(X) the probability of appearance of X in D, and P(Y|X) the conditional probability of appearance of Y given X. P(X) can estimated as P(X)=s(X). The support of a rule X=>Y is defined as s(X=>Y) = s(XUY). An association rule X=>Y has a measure of reliability called the confidence, defined as c(X=>Y) = s(X=>Y)/s(X). Confidence can be used to estimated P(Y|X): P(Y|X) = P(XUY)/P(X) = c(X=>Y). Based on this various algorithms are developed for Apriori and éclat principles. This paper provides implementation of these algorithms for medical datasets which is real time one and analyses the various issues.

The organization of the rest of the paper is as follows. Section 2 describes the basic concepts of apriori and éclat algorithms with new medical database. Section 3 analyses the various terminologies for existing algorithms and their comparisons. Section 4 shows implementation results of new dataset. Section 5 provides the Challenges of existing algorithms. Finally, section 6 concludes the paper along with future work.

II. BASIC ALGORITHMS

A. Apriori Algorithm

The Apriori algorithm developed by [] is a great achievement in the history of mining association rules [32]. This technique uses the property that any subset of a large itemset must be a large itemset. Apriori generates the candidate itemsets by joining the large itemsets of the previous pass and deleting those subsets which are small in the previous pass without considering the transactions in the database. An association rule is *valid* if its confidence and support are greater than or equal to corresponding threshold values.

The data structure trie was used in this Apriori algorithm [31]. A trie is a rooted (downward) directed tree like a hash tree. The root is defined to be at depth 0, and a node at depth d can point to nodes at depth d+1. A pointer is also called edge or link which is labeled by a letter. There exists a special letter * which represents an "end" character. If node *u* points to node *v* then well can *u* the parent of *v*, and *v* is a child node of *u*.

Every leaf *l* represents a word which is the concatenation of the letters in the path from the root to *l*. Note that if the first *k* letters are the same in two words, then the first *k* steps on their paths are the same as well. Tries are suitable to store and retrieve not only words, but any finite ordered sets. In this setting a link is labeled by an element of the set, and the trie contains a set if there exists a path where the links are labeled by the elements of the set, in increasing order.

B. Eclat Algorithm

In Eclat algorithm [31][35] implementation the set of transactions as a (sparse) bit matrix and intersects rows to determine the support of item sets. The search space of Eclat algorithm is based on depth first traversal of a prefix tree. A convenient way to represent the transactions for the Eclat Algorithm is a bit matrix, in which each row corresponds to an item, each column to a transaction.. A bit is set in this matrix if the item corresponding to the row is contained in the transaction corresponding to the column, otherwise it is cleared. Eclat searches a prefix tree. The transition of a node to its first child consists in constructing a new bit matrix by intersecting the first row with all following rows. For the second child, the second row is intersected with all following rows and so on.

The item corresponding to the row is intersected with the following rows to form the common prefix of the item sets, processed in the corresponding child node. Of course, rows corresponding to infrequent item sets should be discarded from the constructed matrix, which can be done most conveniently if it stores with each row the corresponding item identifier rather than relying on an implicit coding of this item identifier in the row index. For a sparse representation the column indices for the set bits should be sorted ascending for efficient processing. Then the intersection procedure is similar to the merge step of merge sort. In this case counting the set bits is straightforward.

C. Medical Database

Experimental data in many domains serves as a basis for predicting useful trends. Association rules are generated from one such medical database [36] with real time surveyed records of 20000 patients. The database which we have chosen depicts the complications occurring in diabetes and/or hypertension (increased Blood pressure). All the patients were in the age group of 25 to 70 years, and the sex ratio was almost equal (M:F of **1.1 : 1**). All of them had either diabetes or hypertension or both for duration of 10 years and more. They were sub-categorized based on the extent of control of these diseases namely diabetes and hypertension. Several complications like kidney disease, heart disease and stroke were evaluated in this group of patients.

Database is the storage which holds data. In the above medical database itemsets are stored in the database corresponding to their transaction which is used for future reference. SQL server is used as the database server. Database server holds the data in string format, string processing requires more time and it is difficult too. Conversion of the string into corresponding integer value assigned to the itemsets, which reduces the execution time complexity and space complexity of the process.

The data type is used for the storage of transactional data is structure. It has an *int* type data to store the count of the itemset. Also it has a string type data to store the name of the itemset. It has a float type data to store the support count of itemset. Doubly linked structure is defined to use process memory occupation efficient one.

III. CLASSIFICATION AND COMPARISON OF ALGORITHMS

A. Classification Schemes

The classification scheme provides a framework which can be used to highlight the major differences among algorithms. We identify the features which can be used to classify the algorithms. The approach we take is to categorize the algorithms based on several basic dimensions or features that we feel best differentiate the various algorithms. In our categorization we identify the ways in which the approaches differ. Our classification uses the dimensions which are given in the table 3.1.

Table 3.1 Classification Dimensions

DIMENSION	VALUES
Target	Complete, constrained, qualitative
Type	Regular, generalized, quantitative,
Data type	Database data, text
Data source	Market basket, beyond basket
Technique	Large itemset, strongly collective
Itemset strategy	Complete, Apriori, Dynamic, Hyb
Transaction strat	Complete, sample, partitioned
Itemset data stru	Hash tree, trie, virtual trie, lattice
Transaction data structure	Flat file, TID
Optimization	Memory, skewed, pruning
Architecture	Sequential, parallel
Parallel strategy	None, data, task

B. Comparing Algorithms

We compare various Apriori and Eclat type algorithms based upon several metrics. Space requirements can be estimated by looking at the maximum number of candidates being counted during any scan of the database.

We can estimate the time requirements by counting the maximum number of database scans needed (I/O estimate) and the maximum number of comparison operations (CPU estimate). Since most of the transaction databases are stored on secondary disks and I/O overhead is more important than CPU overhead, we focus on the number of scans in the entire database. The worst case arises when each transaction in the database has all items. Let m be the number of items in each transaction, and L_k the large itemsets with k-itmesets with k-items in a database D. the number of large itemsets is 2^m . In level wise techniques (AIS, SETM, Apriori) all large itemsets in L_1 are obtained during the first scan of the database. Similarly all large itemsets in L_2 are obtained during the second scan and so on. The only itemset in L_m is obtained during the m^{th} scan. All algorithms terminate when no additional entries in the large itemsets are generated, so an extra scan is needed. Therefore, the entire database will be scanned at most (m+1) times. Here it can be recalled that Apriori-TID scans the entire database in the first pass. It uses C_k rather than the entire database in the (k+1)th pass.

The reason is the C_k will contain all of the transactions along with their items during the entire process. On the other hand, the OCD technique [7] scans the entire database only once at the beginning of the algorithmm to obtain large item sets in $L_{1.}$ Afterwards OCD and sampling use only a part of the entire database and the information obtain in the first pass to find the candidate items sets of L_k where $1<k<=m$. In the second scan they compute support of each candidate item set. Therefore, there will be 2 scans in the worst case given enough main memory. The PARTITION technique [5] also reduces the I/O overhead by reducing the number of database scan to 2. Similarly CARMA needs at most 2 database scans.

The goodness of an algorithm depends on the accuracy of the number of “true” candidates it develops. As we have mentioned earlier, all algorithms use large itemsets of previous pass(es) to generate candidate sets. Large items of previous itemsets are needed to be in the main memory to obtain their support counts. Since enough memory may not be available, different algorithms propose different kinds of buffer management and storage structures. AIS proposed by L_{k-1} can be disk-resident if needed. SETM suggested that if Ck is too large to fit into main memory,write it to disk in FIFO manner. The Apriori family recommended to keep L_{k-1} on disk and bring into the main memory one block at a time to find $C_{k.}$ However, C_k should be in main memory to obtain support count in both Apriori-TID and Apriori-Hybrid. On the other hand,all other techniques assume that there is enough memory to handle this problem.However, neither AIS nor SETM proposed any storage structures. Most commercially available implementations to generate association rules rely on the use of the Apriori technique.

Table 3.2: Summarization of Various Association Rule Algorithms

Algorit	Scan	Data Structure	Comments
AIS	M+1	Not Specified	Suitable for low cardinality sparse transaction databa Single consequent
SETM	M+1	Not Specified	SQL compatible
Apriori	M+1	Lk-1: Hash table Ck : Hash tree	Transaction database with moderate cardinality; Outperforms both AIS and SETM; Base algorithm fo parallel algorithms
Apriori-TID	M+1	Lk-1: Hash table Ck:array indexed by TID Ck:sequential structure ID:bitmap	Very slow with large number of Ck; Outperforms Apriori with smaller number of Ck;
Apriori-Hybr	M+1	Lk-1: Hash table 1st phase: Ck : Hash tree 2nd phase: Ck:array indexed by IDs Ck:sequential structure ID:bitmap	Better than Apriori. However, switching from Aprior Apriori-TID is expensive; Very crucial Figure out the transaction point
OCD	2	Not Specified	Applicable in large DB with lower support threshold
Partition	2	Hash table	Suitable for large DB with high cardinality of data; Favors homogenous data distribution.
Sampling	2	Not Specified	Applicable in very large DB with lower support.
DIC	Depends on val Size	Trie	Database viewed as intervals of transactions; Candidates of increased size are generated at the end interval
CARMA	2	Hash Table	Applicable where transaction sequences are read from network; Online, users get continious feedback and c support and/or confidence anytime during process.
CD	M+1	Hash table and tree	Data Parallelism.
PDM	M+1	Hash table and tree	Data Parallelism; with early candidate pruning
DMA	M+1	Hash table and tree	Data Parallelism; with early candidate pruning
CCPD	M+1	Hash table and tree	Data Parallelism; on shared memory machine
DD	M+1	Hash table and tree	Task Parallelism; round-robin partition
IDD	M+1	Hash table and tree	Task Parallelism; partition by the first items
HPA	M+1	Hash table and tree	Task Parallelism; partition by the hash function
SH	M+1	Hash table and tree	Data Parallelism; candidates generated independently each processor.
HD	M+1	Hash table and tree	Hybrid data and task parallelism; grid parallel archite
AprioriTrie	Depends on in size	Trie	Applicable where transaction sequences are read from network; Online, users get continious feedback
ECLAT	2	Sparse Matrix	Applicable for array based data structure

Some algorithms are more suitable to use under specific conditions. AIS do not perform well when the number of items in the database is large. Therefore, AIS is more suitable in transactions databases with low cardinality. As we have mentioned earlier, Apriori needs less exection time than Apriori-TID [21] in earlier passes. On the other hand, Apriori-TID outperforms Apriori in later passes. Therefore, AprioriTrie shows excellent performance with proper switching. However, switching from Apriori to Apriori-TID is very crucial and expensive. Although OCD is an approximate technique, it is very much effective in finding frequent itemsets with lower threshold support. CARMA is an online user interactive feedback oriented technique which is best suited where transaction sequences are read from a network. Eclat algorithm using depth first tree structure for implementation is also suitable faster execution.

Table 3.2 summarizes and provides a means to briefly compare the various algorithms. We include in this table the maximum number of scans, data structures proposed, and specific comments. The classification of the table is based on Algorithm name, number of scan, data using in these algorithms and usage of these algorithms.. Data such that are taken for our investigation and detailed surveyed report are given in the table.

IV. EXPERIMENTAL RESULTS

To carry out performance studies we implemented the most algorithms to mine frequent itemsets, namely, AprioriTrie, DIC, CARMA, partition and Eclat in C++. Size of the database is 20000 transactions with total of 50 item sets. Data sets are numeric data for simplification. For the experiments, Intel Pentium IV dual core processor, Windows XP with 256 MB RAM is used. These five algorithms are experimented with real time surveyed database.

First of all, we generate execution of time of these algorithms for various support level. Experimental results are compared in figure 4.1 and made some investigation about the existing algorithms.

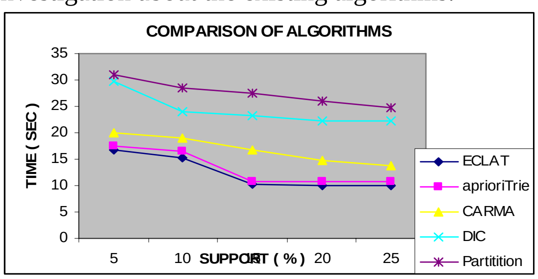

Figure 4.1: Comparison of algorithms execution time

The above diagram comparison gives meaningful information that Eclat and AprioriTrie are efficient one. These algorithms occupy more number of memory space during execution time. Space complexity of the algorithms is the one of the main research issues for our further investigation.

V. CHALLENGES

From the above implementations, we observe the main challenge of computation is to reduce the cost of association rule algorithms i.e. execution time. No algorithm supports the both time and space complexity. The following are used in the above algorithms to reduce the computation time.

- By reducing the number of passes over the database
- By sampling the database
- By adding extra constraints on the structure of patterns
- Through parallelism

To search any number of data sets, all these algorithms support only more than one scan.

The first challenge of this research work in this area is to reduce the database scan to be one. This will create to find the new data structure to the existing algorithmic approaches.

Using of very large data sets is more complicated in these algorithms because of their more database scan during the search time. This implies more process memory occupation.

The second challenge is to reduce the process memory occupation during the item sets search in the database scan.

Generating Association Rules is more complex to the existing algorithms because more number of rules are generated during Association Rule construction step. Some rules are useful, some rules are unwanted and some rules are missed. Interesting informative rules are only needed for us.

The third Challenge is to mine the missing rules which are more useful one.

More number of rules are generated from the existing process. Another research focus is to avoid the unwanted rules.

The fourth challenge is to find the useful rules among all rules.

The medical database is generated by us to create the new investigation in this are and make to find new technique to overcome these challenges. These four major challenges are to be considered for our investigation.

VI. CONCLUSION

In this paper, we surveyed the list of existing association rule mining algorithms and their implementation. Execution results and our database particulars will yield more investigation issues. In order to extract hidden information from the very large database, time and space complexity of the algorithms, types of rules and usefulness of the information are all the factors are considered for new algorithmic approach. This investigation is prepared to our research titled *"An Efficient Algorithm to mine strong Association Rules from very large data sets"* as future enhancement.

REFERENCES

[1]. Mannila, H., Toivonen, H., and Verkamo, A. I. 1995. Discovering frequent episodes I in sequences. In *Interantional Conference on Knowledge Discovery and Data Mining*. IEEE Computer Society Press.

[2]. Agrawal, R., Imielinski, T., and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman and S. Jajodia, Eds. Washington, D.C.,

[3]. Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 487-499.

[4]. Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In *Eleventh International Conference on Data Engineering*, P. S. Yu and A. S. P. Chen, Eds. IEEE Computer Society Press, Taipei, Taiwan, 3-14.

[5]. Bayardo, R., Agrawal, R., and Gunopulos, D. 1999. Constraint-based rule mining in large,dense databases. Berkhin, P. 2002. Survey of clustering data mining techniques. Tech. rep., Accrue Software, San Jose, CA.

[6]. Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. 1997. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, J. Peckham, Ed. ACM Press, 255-264.

[7]. Chen, M.-S., Han, J., and Yu, P. S. 1996. Data mining: an overview from a database perspective. *Ieee Trans. On Knowledge And Data Engineering 8*, 866{883. Cheung, D. W.-L., Han, J., Ng, V., and Wong, C. Y. 1996. Maintenance of discovered association rules in large databases: An incremental updating technique. In *ICDE*. 106-114.

[8]. Cheung, D. W.-L., Lee, S. D., and Kao, B. 1997. A general incremental technique for maintaining discovered association rules. In *Database Systems for Advanced Applications*. 185-194.

[9]. Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM Press, 474-481.

[10]. Duda, R. and Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley&Sons, Inc. Fu, Y. and Han, J. 1995. Meta-rule-guided mining of association rules in relational databases. In *KDOOD/TDOOD*. 39-46.

[11]. Garofalakis, M. N., Rastogi, R., and Shim, K. 1999. SPIRIT: Sequential pattern mining with regular expression constraints. In *The VLDB Journal*. 223-234.

[12]. Han, J. 1995. Mining knowledge at multiple concept levels. In *CIKM*. 19{24. Han, J., Dong, G., and Yin, Y. 1999. Efficient mining of partial periodic patterns in time series database. In *Fifteenth International Conference on Data Engineering*. IEEE Computer Society, Sydney, Australia, 106-115.

[13]. Han, J. and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In *Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), ZÄurich, Switzerland, September 1995*. 420-431.

[14]. Han, J. and Kamber, M. 2000. *Data Mining Concepts and Techniques*. Morgan Kanufmann. Han, J., Koperski, K., and Stefanovic, N. 1997. GeoMiner: a system prototype for spatial data mining. 553-556.

[15]. Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD Explorations Newsletter 2*, 2, 14-20.

[16]. Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In *2000 ACM SIGMOD Intl. Conference on Management of Data*, W. Chen, J. Naughton, and P. A. Bernstein, Eds. ACM Press, 1{12. James, M. 1985. *Classification Algorithms*. Wiley&Sons, Inc.

[17]. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. 1994. Finding interesting rules from large sets of discovered association rules. In *Third International Conference on Information and Knowledge Management (CIKM'94)*, N. R. Adam, B. K. Bhargava, and Y. Yesha, Eds. ACM Press, 401-407.

[18]. Koperski, K. and Han, J. 1995. Discovery of spatial association rules in geographic information databases. In *Proc. 4th Int. Symp. Advances in Spatial Databases, SSD*, M. J. Egenhofer and J. R. Herring, Eds. Vol. 951. Springer-Verlag, 47{66.Koperski, K. and Han, J. 1996. Data mining methods for the analysis of large geographic databases.

[19]. Lee, S. D. and Cheung, D. W.-L. 1997. Maintenance of discovered association rules: When to update? In *Research Issues on Data Mining and Knowledge Discovery*. 0-14.

[20]. Lent, B., Swami, A. N., and Widom, J. 1997. Clustering association rules. In *ICDE*.

220{231. Lu, H., Han, J., and Feng, L. 1998. Stock movement prediction and n-dimensional intertransaction association rules.

[22]. International Conference on machine learning and cybernetics,18-21 AUG 2005.

[23]. Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. 1998. Exploratory mining and pruning optimizations of constrained associations rules. 13-24.

[24]. Park, J. S., Chen, M.-S., and Yu, P. S. 1995. An effective hash based algorithm for mining association rules. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, M. J. Carey and D. A. Schneider, Eds. San Jose, California, 175{186.

[25]. Pei, J. and Han, J. 2000. Can we push more constraints into frequent pattern mining? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, 350-354.

[26]. Psaila, G. and Lanzi, P. L. 2000. Hierarchy-based mining of association rules in data warehouses. In *Proceedings of the 2000 ACM symposium on Applied computing 2000*. ACM Press, 307-312.

[27]. Savesere, A., Omiecinski, E., and Navathe, S. 1995. An efficient algorithm for mining association rules in large databases. In *Proceedings of 20th International Conference on VLDB*.

[28]. Smythe and Goodman. 1992. An information theoretic approach to rule induction from databases. In *IEEE Transactions on Knowledge and Data Engineering*. IEEE Computer Society Press.

[29]. Srikant, R. and Agrawal, R. 1996. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*. ACM Press, 1-12.

[30]. Srikant, R., Vu, Q., and Agrawal, R. 1997. Mining association rules with item constraints. In *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, D. Heckerman, H. Mannila,

[31]. D. Pregibon, and R. Uthurusamy, Eds. AAAI Press, 67{73. Thomas, S., Bodagala, S., Alsabti, K., and Ranka, S. 1997. An efficient algorithm for the incremental updation of association rules in large databases. In *Knowledge Discovery and Data Mining*. 263{266. Wang, W., Yang, J., and Yu, P. 1997. Efficient mining of weighted association rules (war).

[21]. Zhi-Choa Li, Pi-Lian He, Ming Lei, "A High Efficient AprioriTID Algorithm for mining Association rule", Proceedings of 4th

[32]. Mingju Song and Sanguthevar Rajasekaran *A transaction mapping for frequent itemsets mining* IEEE transactions on Knowledge and Data Engineering 18(4):472-480, April 2006.

[33]. Ke Su, Fengsdhan Bai *Mining weighted Association Rules* IEEE transactions on KDE 489-5=495, April 2008

[34]. Balaji Padmanabhan, Alexandar Tuzhilin *On Characterization and Discovery of Minimal unexpected pattern in Rule Discovery* IEEE trans on KDE vol 18 no. 2 Feb '06

[35]. Rajeev Rastogi, Kyuseok Shim *Mining Optimized Association Rules with categorical and numerical attributes* IEEE trans on KDE vol 14 No.1 Jan/Feb '02

[36]. Christian Borglet *Efficient implementation of Aprirori and Eclat* FIMI 2004

[37]. Dr. E. Ramaraj, N. Venkatesan *Discrete Topological Mining Association Rules – An Approach IRIS '06*

Authors

E. Ramaraj is presently working as a Director and Head, Computer Science and Engineering at Alagappa University, Karaikudi. He has 24 years teaching experience and 6 years research experience. He has presented research papers in more than 40 national and international conferences and published more than 30 papers in national and international journals. His research areas include Data mining and Network security.

N.Venkatesan is working as an Assistant Professor & Head, Information Technology Department, Bharathiyar college of Engineering and Technology, Karaikal. Currently pursuing Ph.D in Computer Science at SASTRA University.. He has been member of ISTE. He published 12 papers in National and International conferences. He has authored a book ***Data Mining and Warehousing***.