International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-2 Issue-4 June-2012

An Efficient Range Partitioning Method for Finding Frequent Patterns from Huge Database

Ruchita Gupta[1], C.S.Satsangi[2]
M.Tech Scholar, Dept of Information Technology[1]
H.O.D, Dept of Computer Science and Information Technology[2]
Medicaps Institute of Technology and Management[1,2]

Abstract

Data mining is finding increasing acceptance in science and business areas that need to analyze large amounts of data to discover trends that they could not otherwise find. Different applications may require different data mining techniques. The kinds of knowledge that could be discovered from a database are categorized into association rules mining, sequential patterns mining, classification, and clustering. In this paper we present an efficient range partitioning method for finding frequent pattern from huge database. It is based on key based division for finding the local frequent pattern (LFP). After finding the partition frequent pattern from the subdivided local database, we then find the global frequent pattern from the local database and perform the pruning from the whole database.

Keywords

KDD, LFP, Partitioning, Association Rule

1. Introduction

By the increased data collection power, massive data is easily obtained by companies, government agencies and healthcare organizations. Although many data mining algorithms show their effects under normal data, most of them face difficulties in handling data that are substantially larger than available main memory on a single processor. Thus, with the rapid development of computers and communication networks, data mining in distributed environment is becoming a heated problem. Distributed data mining (DDM) is expected to relieve current mining methods from the sequential bottleneck, and provide the ability to scale to massive data sets and improve the response time [1].

Data mining has been a powerful technique in analyzing and utilizing data in today's information-rich society. However, privacy is nowadays a major concern in data mining applications, which has led to a new research area, privacy preserving data mining. A large amount of research work has been devoted to this area, and resulted in such techniques as k-anonymity [2], data perturbation [3], [4], [5], [6], and privacy preserving distributed data mining [7], [8].

Depending on the class of knowledge derived, the data Mining approaches may be classified as finding association rules, classification rules, clustering rules, and sequential patterns [9], among others. Among them, finding association rules in transaction databases is most commonly seen in data mining [10][11][12][13][14].

In the past, many algorithms for mining association rules from transactions were proposed, most of which were based on the Apriori algorithm [15], which generated and tested candidate item sets level by level. Han et al. then proposed the Frequent Pattern-tree (FP-tree) structure for efficiently mining association rules without generation of candidate item sets [16].

The partitioning algorithms divide the transactional dataset D into n non-overlapping partitions, D1, D2…Dn. The algorithms reduce the number of dataset scans to two. During the first scan, the algorithm finds all item sets in each partition. Those local frequent item sets are collected into the global candidate item sets. During the second scan, these global item sets are counted to determine if they are large across the entire dataset. The partitioning algorithms improve the performance of finding frequent item sets and also provide several advantages. Small partitions might be fit into main memory than large one. Because the size of each partition is small, the algorithms might reduce the size of candidate item sets. In addition, the algorithms require only two scans on the dataset. However, the partition algorithms reduce the size of

the data set; they might still need to deal with the large size of item sets.

Many algorithms apply the partitioning technique for association rule mining in parallel. The algorithms mainly partition datasets or candidate item sets for parallelism. Partitioning datasets for parallel association mining (count distribution algorithms) divides a dataset into small partitions. Partitions are distributed to processors where each processor creates its local candidate item sets against its own dataset partition. The processors are then exchanging their local dataset partitions and candidate item sets for the global candidate item sets. Each processor removes its infrequent item sets from its local candidate item sets against the global one. The resulting candidate item sets become the frequent item sets for the next candidate item sets [17]. The extraction of the frequent item sets is done on all the processors in parallel. However, the extraction can also be done on one master processor. By doing so, the master processor need to broadcast the result to the other processors. Partitioning datasets for parallelism has a weakness that it requires much synchronization among processors. In this paper we also proposed an efficient key based partitioning method.

The remaining of this paper is organized as follows. We discuss Data Mining Task in Section 2. In Section 3 we discuss about Association Rule. In section 4 we discuss about Recent Scenario. In section 5 we discuss about the proposed approach. Conclusions are given in Section 6. Finally references are given.

2. Data Mining Task

Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed. A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives. These primitives allow the user to inter actively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths. The data mining primitives specify the following, as illustrated in Figure 1.

The set of task-relevant data to be mined: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).

The kind of knowledge to be mined: This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.

The background knowledge to be used in the discovery process: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.

An example of a concept hierarchy for the attribute (or dimension) age is shown in Figure 1. User beliefs regarding relationships in the data are another form of background knowledge. The interestingness measures and thresholds for pattern evaluation: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting. The expected representation for visualizing the discovered patterns: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact with data mining systems. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built. This facilitates a data mining system's communication with other information systems and its integration with the overall information processing environment. Designing a comprehensive data mining language is challenging because data mining covers a wide spectrum of tasks, from data characterization to evolution analysis. Each task has different requirements. The design of an effective

data mining query language requires a deep understanding of the power, limitation, and underlying mechanisms of the various kinds of data mining tasks.

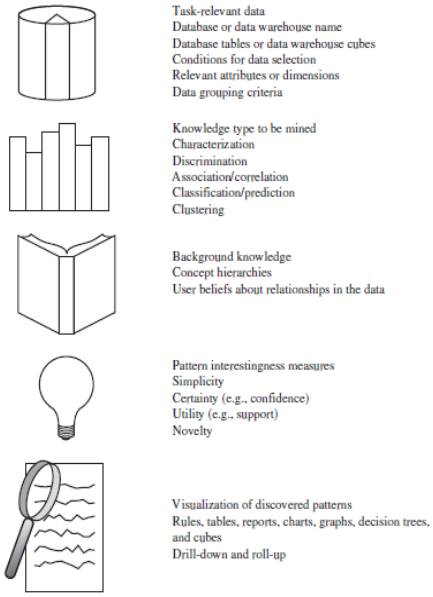

Figure 1: Primitives for specifying the data mining task

3. Association Rule

The market-basket problem assumes we have some large number of items, e.g., \bread," \milk." Customers fill their market baskets with some subset of the items, and we get to know what items people buy together, even if we don't know who they are. Marketers use this information to position items, and control the way a typical customer traverses the store. In addition to the marketing application, the same sort of question has the following uses:
1. Baskets = documents; items = words. Words appearing frequently together in documents may represent phrases or linked concepts. Can be used for intelligence gathering. 2. Baskets = sentences, items = documents. Two documents with many of the same entences could represent plagiarism or mirror sites on the Web.
Goals of market Basket analysis:
Association rules are statements of the form
{X1,X2….Xn}⟼ Y meaning that if we find all of X1,X2,..Xn in the market basket, then we have a good chance of finding Y. The probability of finding Y for us to accept this rule is called the confidence of the rule. We normally would search only for rules that had confidence above a certain threshold. We may also ask that the confidence be significantly higher than it would be if items were placed at random into baskets. For example, we might find a rule like {milk,butter} bread. Simply because a lot of people buy bread.

Causality: Ideally, we would like to know that in an association rule the presence of X1,X2…..Xn actually causes Y to be bought. However, causality is an elusive concept. Nevertheless, for market-basket data, the following test suggests what causality means. If we lower the price of diapers and raise the price of beer, we can lure diaper buyers, who are more likely to pick up beer while in the store, thus covering our losses on the diapers. That strategy works because \diapers causes beer." However, working it the other way round, running a sale on item1 and raising the price of item2, will not result in item1 buyers buying item2 in any great numbers, and we lose money.

Frequent item sets: In many (but not all) situations, we only care about association rules or causalities involving sets of items that appear frequently in baskets. For example, we cannot run a good marketing strategy involving items that no one buys anyway. Thus, much data mining starts with the assumption that we only care about sets of items with high support; i.e., they appear together in many baskets. We then find association rules or causalities only involving a high-support set of items (i.e., fX1; : : :;Xn; Y g must appear in at least a certain percent of the baskets, called the support threshold.

Apriori [18][19] is the most popular and effective algorithm to find all the frequent item sets in dataset. It is proposed by Agrawal and Srikant in 1994. Let I = {I1, I2……, Ik} be a set of k distinct attributes, also called literals. Ai = s is an item, where s is a domain value is attributing, Ai in a relation, R (A1… An). A is an item set if it is a subset of I. DT =. {ti, ti+1… tn) is a set of transactions, called the transaction (tid, item set).

A transaction t contains an item set A if and only if, for all items IA, i is in t-item set. An itemset A in a transaction database DT has a support, denoted as Supp (A) (we also use p(A) to stand for Supp (A)),

that is the ratio of transactions in DT contain A. Supp (A) = |A(t)| / |DT|, Where A(t) = {t in DT/t contains A}. An item set A in a transaction database DT is called a large (frequent) item set if its support is equal to, or greater than, a threshold of minimal support (minsupp), which is given by users or experts. An association rule is an expression of the form IF A THEN B (or A -> B), A∩B = Ø, where A and B are sets of items. The meaning of this expression is that transactions of the databases, which contain A, tend to contain B. Each association rule has two quality measurements: support and confidence, defined as:

(1) The support of a rule A->B is the support of A∪B, where A ∪B means both A and B occur at the same time in same transaction.
(2) The confidence or predictive accuracy [2] of a rule A->B is conf (A->B) as the ratio:| (A ∪B)(t)| / |A(t) or Supp(A ∪B) / Supp(A).

That is, support = frequencies of occurring patterns; confidence = strength of implication. Support-confidence framework [5][11]: Let I be the set of items in database D, A, B I be item set, A ∩B = f, p(A) is not zero and p(B) is not zero. Minimal support minsupp) and minimal confidence (minconf) are given by users or experts. Then A->B is a valid rule if
1. Supp (A∪B) is greater or equal to minsupp,
2. Conf (A->B) is greater or equal to minconf.

Mining association rules can be broken down into the following two sub-problems [5]:
1. Generating all item sets that have support greater than, or equal to, the user specified Minimal support. That is, generating all large item sets.
2. Generating all the rules that have minimum confidence.

Negative Association Rules
The negation of an item set A is represented by ¬A, which means the absence of the item set A. We call a rule of the form A=> B a positive association rule, and rules of the other forms (A=> ¬B, ¬A=>B and ¬A=> ¬B) negative association rules. The support and confidence of the negative association rules can make use of those of the positive association rules [10].In this paper work we have create a meaning for these type rule like:

Positive Rule (PR) = A=> B Consequent Negative Rule (CNR) = A=> ¬B Antecedent Negative Rule (ANR) = ¬A=>B
Antecedent and Consequent Negative (ACNR) = ¬A=> ¬B

The support and Confidence for CNR, ANR and ACNR rule is given by the following formulas: The support and Confidence for CNR, ANR. and ACNR rule is given by the following formulas:

Consequent Negative Rule (CNR)
Supp (A=>¬B) =supp (A)-supp (AuB)
Conf (A=>¬B) =supp (A)-supp (A∪B)/Supp (A)

Antecedent Negative Rule (ANR)
Supp (¬A=>B) =supp (B) - supp (A∪B)
Conf (¬A=>B) =supp (B)-supp (A∪B)/1-supp (A)

To distinguish the strong association rules among all possible rules, the confidence of each possible rule will be calculated.
Confidence ($A \rightarrow B$) =Support (AB)/Support (A)
Confidence ($AB \rightarrow C$) =Support (ABC)/Support (AB)

Association rules have been one of the most developed areas in data mining. Most of research has been done in positive implications, which is when occurrence of an item implies the occurrence of another item. Negative rules also consider negative implications, when occurrence of an item implies absence of another item.

4. Recent Scenario

In 2006, Takahiko Shintani et al. [20] propose methods of reducing processing workload: estimating the upper bound of global support using local supports, reutilizing part of the constructed tree structure, and merging redundant database partitions. Their performance study shows that our algorithm is efficient and can always find all association rules.

In 2008, Dan Hu et al. [21] discuss the relation between the reducts of partitioned data and global data. A useful proposition is obtained, which shows that every reduct of global data determinedly has subsets as the elements in reducts of partitioned data. DMR and PPDMR are proposed for distributed mining of reducts on horizontally partitioned data. DMR concerns the reduction of time complexity while PPDMR focuses on privacy preserving.

Experiments and propositions show the excellent function of DMR and PPDMR through practical and academic ways.

In 2009, Zhenmin Lin et al. [22] generalize the idea to the situation where the data matrix is assumed to be vertically partitioned into several sub-matrices and held by different owners. Each data holder can choose a rotation matrix randomly and independently to perturb their individual data. Then they all send the transformed data to a third party, who collects all of them and forms a whole data set for data mining or other analysis purposes. They show that under such a scheme the geometric properties of the data set is also preserved and thus it can maintain the accuracy of many classifiers and clustering techniques applied on the transformed data as on the original data. This method enables us to develop efficient centralized data mining algorithms instead of distributed algorithms to preserve privacy. Experiments on real data sets show that such generalization is effective for vertically partitioned data sets.

In 2010, Asha Khatri et al. [23] propose architecture for privacy preserving in data mining by combining horizontal data distribution and vertical data distribution for breast cancer data set.

In 2010, Chia-Chu Chiang et al. [24] proposed a distributed association mining algorithm in finding frequent itemsets. The work is different from many existing distributed algorithms where most of existing algorithms center on the reduction of the size of the dataset. Our distributed algorithm focuses on the reduction of the size of candidate item sets. The work of candidate k-item sets generation is evenly distributed to the nodes for workload balancing among processors. The complexity analysis of the distributed algorithm is also presented by the author.

5. Partitioning methods

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of output desired, the known performance of method with particular types of data, the hardware and software facilities available and the size of the dataset. In general, clustering methods may be divided into two categories based on the cluster structure which they produce. The non-hierarchical methods divide a dataset of N objects into M clusters, with or without overlap.

These methods are sometimes divided into partitioning methods, in which the classes are mutually exclusive, and the less common clumping method, in which overlap is allowed. Each object is a member of the cluster with which it is most similar; however the threshold of similarity has to be defined. The hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains. The hierarchical methods can be further divided into agglomerative or divisive methods. In agglomerative methods, the hierarchy is build up in a series of N-1 agglomerations, or Fusion, of pairs of objects, beginning with the un-clustered dataset. The less common divisive methods begin with all objects in a single cluster and at each of N-1 steps divides some clusters into two smaller clusters, until each object resides in its own cluster.

The partitioning methods generally result in a set of M clusters, each object belonging to one cluster. Each cluster may be represented by a centroid or a cluster representative; this is some sort of summary description of all the objects contained in a cluster. The precise form of this description will depend on the type of the object which is being clustered. In case where real-valued data is available, the arithmetic mean of the attribute vectors for all objects within a cluster provides an appropriate representative; alternative types of centroid may be required in other cases, e.g., a cluster of documents can be represented by a list of those keywords that occur in some minimum number of documents within a cluster. If the number of the clusters is large, the centroids can be further clustered to produces hierarchy within a dataset.

Single Pass: A very simple partition method, the single pass method creates a partitioned dataset as follows:

1. Make the first object the centroid for the first cluster.
2. For the next object, calculate the similarity, S, with each existing cluster centroid, using some similarity coefficient.
3. If the highest calculated S is greater than some specified threshold value, add the object to the corresponding cluster and re determine the centroid; otherwise, use the

object to initiate a new cluster. If any objects remain to be clustered, return to step 2.

As its name implies, this method requires only one pass through the dataset; the time requirements are typically of order O(NlogN) for order O(logN) clusters. This makes it a very efficient clustering method for a serial processor. A disadvantage is that the resulting clusters are not independent of the order in which the documents are processed, with the first clusters formed usually being larger than those created later in the clustering run.

A partitioning technique can be used that requires just two database scans to mine the frequent Item sets (Figure 3). It consists of two phases. In phase I, the algorithm subdivides the transactions of D into n non overlapping partitions. If the minimum support threshold for transactions in D is min-sup, then the minimum support count for a partition is min-sup the number of transactions in that partition. For each partition, all frequent item sets within the partition are found. These are referred to as local frequent item sets. The procedure employs a special data structure that, for each item sets, records the TIDs of the transactions containing the items in the item sets. This allow is to find all of the local frequent k-item sets, for k=1,2…., in just one scan of the database.

A local frequent item set may or may not be frequent with respect to the entire database D. Any item set that is potentially frequent with respect to D must occur as a frequent item set in at least one of the partitions. Therefore, all local frequent item sets are candidate item sets with respect to D. The collection of frequent item sets from all partitions forms the global candidate item sets with respect to D. In phase II, a second scan of D is conducted in which the actual support of each candidate is assesses in order to determine the global frequent Item sets. Partition size and the number of partitions are set so that each partition can fit into main memory and therefore be read only once in each phase. We can apply any partitioning techniques like Oracle Partitioning offers three fundamental data distribution methods as basic partitioning strategies that control how data is placed into individual partitions:

- Range
- Hash
- List

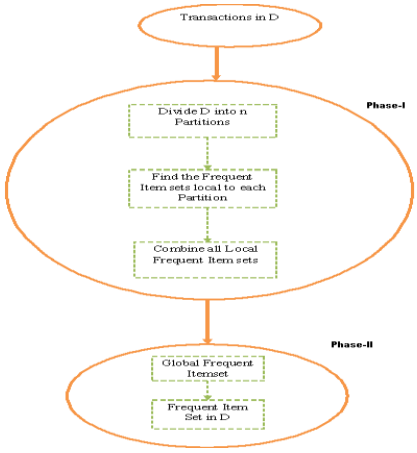

Figure 2: Mining by partitioning the data

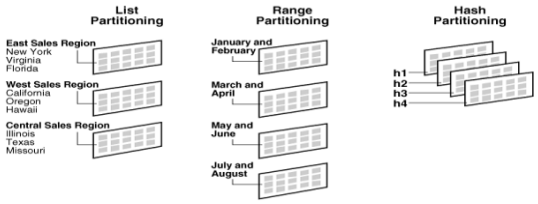

Figure 3: List, Range and Hash Partitioning

In our algorithm we apply range partitioning algorithm.

Range Partitioning

Range partitioning maps data to partitions based on ranges of values of the partitioning key that you establish for each partition. It is the most common type of partitioning and is often used with dates. For a table with a date column as the partitioning key, the January-2010 partition would contain rows with partitioning key values from 01-Jan-2010 to 31-Jan-2010 which is shown in Figure 3.

Each partition has a VALUES LESS THAN clause that specifies a non-inclusive upper bound for the partitions. Any values of the partitioning key equal to or higher than this literal are added to the next higher partition. All partitions, except the first, have an implicit lower bound specified by the VALUES LESS THAN clause of the previous partition. A MAXVALUE literal can be defined for the highest partition. MAXVALUE represents a virtual infinite value that sorts higher than any other possible value for the partitioning key, including the NULL value.

Algorithm: Partition (I1, I2….In)

Input: D=dataset
K= the number of centers
C=initial centroids
Output: Set of k representing a good partitioning of D database and produce the frequent pattern.
1: Select the initial data set
2: for all data point d_i € D do
3: assignedCenter =$d_{i.center}$
4: assignedPartiton= $d_{i.partition}$
5: for all center c_i€C do
6: apply on the Item set I_i€I
7: X_n,n=1,2,3……N

8: A partition P of an interval I is a set of M blocks,

9: P(I)={B_m,m€M},M={1,2…M}
10: where the blocks are sets of data cells defined by index sets $N_{m::}$ B_m={X_n,n€N_m}
11: enter key (n) for partition.
12: Find count (item set)
13: if(count (item set) isEven())
{If key (even)
{Partition in n/2 sets
}Else
{Partition in n+/2 sets

}14: if(count(itemset) isOdd())

{If key(even)

{Partition in n+/2 sets

}Else{

Partition in n /2 sets}

15: find frequent pattern for the local partition until all local partition finishes

16: Combine all local partition and find the global partition.

17: Finish

The above algorithm is shown by the flowchart [Figure 4].

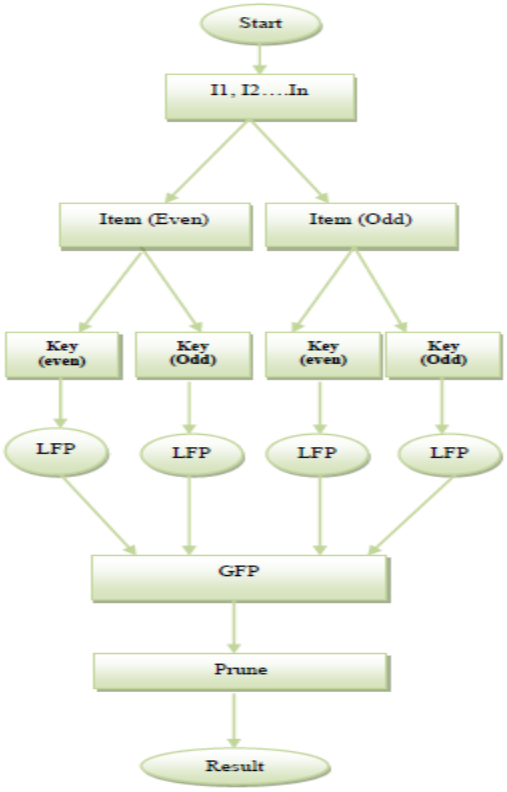

Figure 4: Flowchart

6. Conclusion

Data mining deals with huge volume data for knowledge discovery. Association rule discovery is one of the areas in data mining. The Apriori algorithm was developed in the early years of data mining. However, the Apriori algorithm has a weakness in performance. Various variations are proposed to improve the performance of the Apriori algorithm. Among the existing algorithms, two major algorithms including data distribution and task distribution are using the power of distributed computing. In this paper, we present an improve way for finding frequent pattern.

References

[1] J.Z.Mohammed.Parallel and distributed data mining: an introduction. M.J.Zaki, C.-T.Ho (Eds.) Large-scale parallel data mining, Lecture Notes in Artificial Intelligence, 1759:1-23, 2000.
[2] Sweeney, L. k-anonymity: a model for protecting privacy. International Journal on Uncertainty,

Fuzziness and Knowledge-based Systems, 10(5):557-570, 2002.
[3] Agrawal, R. and Srikand R. Privacy preserving data mining. In Proc. Of ACM SIGMOD Conference, pp. 439-450, 2000.
[4] Chen, K. and Liu. L. A random rotation perturbation approach to privacy data classification. In Proc of IEEE Intl. Conf. on Data Mining (ICDM), pp. 589-592, 2005.
[5] Xu, S., Zhang, J., Han, D. and Wang J. Singular value decomposition based data distortion strategy for privacy distortion. Knowledge and Information System, 10(3):383-397, 2006.
[6] Mukherjeee, S., Chen, Z. and Gangopadhyay, A. A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier related transforms. Journal of VLDB, 15(4):293-315, 2006.
[7] Vaidya, J. and Clifton, C. Privacy preserving k-means clustering over vertically partitioned data. In Prof. of ACM SIGKDD Conference, pp.206-215, 2003.
[8] Vaidya, J., Yu, H. and Jiang, X. Privacy preserving SVM classification. Knowledge and Information Systems, 14:161-178, 2007.
[9] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: a frequent-pattern tree approach, Data Min. Knowl. Discovery 8 (2004) 53–87.
[10] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," The International Conference on Very Large Data Bases, pp.487-499, 1994.
[11] R. Agrawal, R. Srikant and Q. Vu, "Mining association rules with item constraints," The Third International Conference on Knowledge Discovery in Databases and Data Mining, pp. 67-73, 1997.
[12] T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, "Mining optimized association rules for numeric attributes," The ACM SIGACTSIGMOD- SIGART Symposium on Principles of Database Systems, pp. 182-191, 1996.
[13] J. Han and Y. Fu, "Discovery of multiple-level association rules from large database," The Twenty-first International Conference on Very Large Data Bases, pp. 420-431, 1995.
[14] H. Mannila, H. Toivonen, and A.I. Verkamo, "Efficient algorithm for discovering association rules," The AAAI Workshop on Knowledge Discovery in Databases, pp. 181-192, 1994.
[15] R. Agrawal, T. Imielinksi and A. Swami, "Mining association rules between sets of items in large database, "The ACM SIGMOD Conference, pp. 207-216, 1993.
[16] J. Han, J. Pei, and Y. Yin,"Mining frequent patterns without candidate generation" The 2000 ACM SIGMOD International Conference on Management of Data, 2000.
[17] M. H. Dunham, Y. Xiao, L. Gruenwald, and Z. Hossain, "A Survey of Association Rules," Technical Report, Southern Methodist University, Department of Computer Science, Technical Report TR 00-CSE-8, 2000.
[18] Das Sufal and Saha Banani" Data Quality Mining using Genetic Algorithm**"** International Journal of Computer Science and Security, (IJCSS) Volume (3): Issue (2).
[19] X. Wu, C. Zhang, and S. Zhang, "Efficient Mining of both Positive and Negative Association Rules," ACM Transactions on Information Systems, Vol. 22, No. 3, 2004, pp. 381–405.
[20] Takahiko Shintani,"Mining Association Rules from Data with Missing Values by Database Partitioning and Merging", Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS, 2006 IEEE.
[21] [21] Dan Hu, Xianchuan Yu and Yuanfu Feng , "Distributed Mining Reducts of Attributes on Horizontally Partitioned Data", 2008 IEEE.
[22] Zhenmin Lin, Jie Wang, Lian Liu, Jun Zhang, "Generalized Random Rotation Perturbation for Vertically Partitioned Data Sets",IEEE 2009.
[23] Asha Khatri, Swati Kabra, Shamsher Singh , "Architecture for Preserving Privacy During Data Mining by Hybridization of Partitioning on Medical Data", 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, IEEE.
[24] Chia-Chu Chiang and Shen Lu, "Distributed Association Mining on Message Passing Systems",IEEE 2010.

B.E. in Information Technology completed in 2009 from Sushila Devi Bansal College of Technology, Indore. Pursuing M.E. from Medicaps Institute Of Technology and Management Indore taken admission in 2010 in Information Technology.