http://www.scientific-journals.org

Exposition and Execution of a Scaled Aprori in Multidimensional Dataset

M Afshar Alam[1], Sapna Jain[2] , Ranjit Biswas[3]

Jamia Hamdard University, New Delhi
[1] mailtoafshar@rediffmail.com, [2] hellosap@sify.com, [3] ranjitbiswas@yahoo.com

ABSTRACT

This research work proposes discovery of quantitative association rule in spatial data that is related only to objects what enables data to be stored in relational database management system .The traditional Apriori algorithm is a method that helps to minimize the number of candidate sets while generating association rules by evaluating quantitative information associated with each item .We have proposed an algorithm for *Aprori-UB which uses multidimensional access method*, UB-tree to generate better association rules with high support and confidence. In multidimensional databases, objects are indexed according to several or many independent attributes. However, this task cannot be effectively realized using many standalone indices and thus special indexing structures have been developed is last two decades. Common to all this structures is that they index vectors of values instead of indexing single values. The UB-tree represents one of the promising multidimensional index structures. Indexing and querying high-dimensional databases is a challenge for current research since high-dimensional indexing is significantly influenced phenomenon called curse of dimensionality .The proposed method in the paper reduces the number of item sets generated and also improves the execution time of the algorithm. Any valued attribute will be treated as quantitative and will be used to derive the quantitative association rule which usually helps in making the rules efficient to handle all types of data.

Keywords: *Data Mining, Association Rule, Quantitative multidimensional theorem, Apriori algorithm, Frequent Itemset, UB-tree.*

1. INTRODUCTION

Scaling is considered an important aspect in Data Mining. The problem of scalability in Data mining is not only how to process such large sets of data, but how to do it within a useful timeframe. Scalability means that as a system gets larger, its performance improves correspondingly. The main purpose of data mining techniques is to find hidden information and unknown relations within an amount of data. The size of the data increases vastly , but useful information extracted from the data seems to be decreasing. In order to comply with future demands, new data mining algorithms and concepts have to be developed that can handle the growing data sets and extract more sophisticated information. Existing software packages should be extended, by adding robustness and scalability, in order to successfully handle the large data sets.

Association rules are highly popular data mining tool. However, most of the approaches are designed for "market basket analysis" and operate on qualitative data. It focuses on common types of data based on numeric values. Special forms of association rules are quantitative attributes that are the area of research for many researchers. There are only few algorithms and methodologies to deal with quantitative associations [8, 4, 5].

Data-driven algorithms are expected to be competitive to those based on discretization. An example of such algorithm is Window algorithm proposed in [4] for their new form of a quantitative rule. In Window, the boundaries of ranges in the antecedent of an association rule are determined by attribute values for specific tuples. A set of these ranges, called a profile, selects a subset of tuples. The antecedent consists of a statistical measure usually the mean, which is based on values of another numeric attribute.The measure for the subset is compared with the same measure for the whole relation. The rule is significant if the difference between these two measures is high. For a set of quantitative values the best description of its behaviour is its distribution. For numerical values, the standard measures for describing a distribution are the *mean* and *variance*. Range is a weak measure, and may be distorted and therefore misleading. The mean and variance is a more satisfactory description of numerical values.

In [4] only rules with single numeric attribute in the antecedent are presented. This paper describes a generalization of this solution to multiple attributes. The main task of this methodology is the automatic discovery of or hyper-cuboids sub-spaces that have significantly different qualities from the whole space. It may be useful

for intelligent analysis of maps, continuous processes or even multimedia. Consequently, this paper discusses one aspect of such multidimensional quantitative database.

The association rules are most popular technique used for database research. In the given set of transactions, where each transaction is a set of items, an association rule is an expression of the form A + B, where A and B are sets of items. This method is used to find all association rules that satisfy user-specified minimum support and minimum confidence constraints. Conceptually, this problem can be viewed as finding associations between the "l" values in a relational table where all the attributes are Boolean. The table has an attribute corresponding to each item and a record corresponding to each transaction. The value of an attribute for a given record is "1"if the item corresponding to the attribute is present in the transaction corresponding to the record, "0" else.

Relational tables in most business and scientific domains have richer attribute types. Attributes can be quantitative (e.g. age, income) or categorical (e.g. zip code, make of car). Boolean attributes can be considered a special case of categorical attributes. This research work defines the problem of mining association rules over quantitative attribute in large relational tables and techniques for discovering such rules. This is referred as the Quantitative Association Rules problem [1].

The original problem of mining association rules was formulated as how to find rules of the form set1 →set2. This rule is supposed to denote affinity or correlation among the two sets containing nominal or ordinal data items. More specifically, such an association rule should translate the following meaning: customers that buy the products in *set1* also buy the products in *set2*. Statistical basis is represented in the form of minimum support and confidence measures of these rules with respect to the set of customer transactions [2].

This research work is the extension of the previous work where we have proposed an algorithm for Discovery of Scalable Association Rule from large set of multidimensional quantitative datasets using k-means clustering method based on the range of the attributes in the rules and Equi-depth partitioning using scale k-means for obtaining better association rules with high support and confidence.

This paper has the following sections. Section 2 represents the previous work done in the same field .Section 3 gives the conceptual details used in the proposed algorithm. Section 4 highlights the proposed Aprori-UB method .Section 5 gives the implementation details. Section 6 gives rule comparison details. Section 7 and section 8 discusses the conclusion and future scope.

2. RELATEDWORK

The problem of mining association rules is to find all rules that satisfy a user-specified minimum support and minimum confidence.

Given a set of transactions, where each transaction is a set of items, an association rule is an expression X → Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions in the database which contain the items in X tend to also contain the items in Y. An example of such a rule might be that 98% of customers that purchase tires and auto accessories also buy some automotive services; here 98% is called the confidence of the rule. The support of the rule X → Y is the percentage of transactions that contain both X and Y .

Usually association analysis is not given decision attributes so that we can find association and dependence between attributes to the best of our abilities. But the aimless analysis may take much time and space. Decision attributes determined can reduce the amount of candidate sets and searching space, and then improve the efficiency of algorithms to some extent [27].In addition, users are not interested in all association rules, but they are just concerned about the associations among condition attributes and decision attributes. If mining association rules from continuous attributes data, the continuous attributes have to be discretized first. The essence of discretization is to use the selected cut-points to divide the values of the continuous attributes into intervals. The methods of dividing determine the quality of association rules [25].

Educational data sets are normally very small if we compare them with databases used in other data mining fields – typical sizes are the size of one class, which are often only 50-100 exemplars. In very few cases, we get data from 200-300 students.

Since Apriori algorithm was first introduced and as experience was accumulated, there have been many attempts to devise more efficient algorithms of frequent itemset mining. Many of them share the same idea with Apriori in that they generate candidates. These include hash-based technique, partitioning, sampling and using vertical data format. Hash-based technique can reduce the size of candidate itemsets. Each itemset is hashed into a corresponding bucket by using an appropriate hash function. Since a bucket can contain different itemsets, if its count is less than a minimum support, these itemsets in the bucket can be removed from the candidate sets [19].

This process is repeated until no more large itemsets are found. Apriori is more efficient during the candidate generation process [35]. Apriori uses pruning techniques to avoid measuring certain itemsets, while guaranteeing completeness. These are the itemsets that the algorithm can prove will not turn out to be large.

The AIS algorithm was the first algorithm proposed for mining association rule [34].In this algorithm only one item consequent association rules are generated, which means that the consequent of those rules only contain one item, for example we only generate rules like X ∩ Y⇒Z but not those rules as X⇒Y ∩ Z. The main drawback of the AIS algorithm is too many candidate itemsets that finally turned out to be small regenerated, which requires more space and wastes much effort that turned out to be useless. At the same time this algorithm requires too many passes over the whole database.

In the previous work we introduce the problem of mining association rules in large relational tables containing both quantitative and categorical attributes. We used using k-means clustering method based on the range of the attributes in the rules and Equi-depth partitioning using scale k-means for obtaining better association rules with high support and confidence.The discretization process is used to create intervals of values for everyone of the attributes in order to generate the association rules.The result of the algorithm discover association rules with high confidence and support in representing relevant patterns between project attributes using the scalable k-means.[18]

3. CONCEPTS USED

Multi-dimensional (mean based) quantitative association rule is a rule of the form:

P_{rX})→(PrX) (M(D))

where:

- J≠ X
- $M(P_{rx})$-M(D) >= mindif
- |(PrX)| <= minsup

The antecedent of the rule is a profile that defines a subpopulation of tuples that is significantly different from the whole D with regard to the attribute J. It is assured by the second condition (a difference condition) that holds if there is a minimal difference mindif between the measure for D and for the P_{rX}. In [4] standard methods for statistical hypothesis testing were then applied to check the significance of the difference. The third condition is a standard support requirement for an association rule. Constants mindif and minsup are user-defined parameters. There is no confidence parameter of the rule. The rule has the difference parameter dif = M(PrX) - M(D) instead, to indicate its strength. Let us here specify minimal M for a rule by μ = M(D) + mindif. The dimensionality of the rule is equal to the number of attributes in its profile.

3.1 THEOREMS: Let us present two theorems that describe properties of quantitative rules and are essential for discovering them.

Theorem 21: If the quantitative association rules $P_{m\ N}$ →M(P_{rX}) is irreducible[12], then V_1.Z= l^$V_{2..}$ Z=b^V_1

Proof. This theorem states that on every profile boundary of irreducible rule is a tuple (called V tuple), that has J value above V. Let us assume that, on the co n-trary, there is a plain, below average tuple that is closer to profile boundary than a μ tuple. Then we can draw a division line between the tuple and the rest of the profile along the boundary. As a result the part with this single tuple is below average, so the whole profile cannot be irreducible rule. The practical consequence of this theorem is that μ -tuples with maximal and minimal P_{rx} attribute values define the profile area of the rule[12].

Theorem 22: There are minimum 2, maximum 2k V-tuples to define a profile of the irreducible rule [12].

Proof. The profile of the rule is a hyper-cuboid with 2k faces and 2k vertexes. A single V tuple defines maximum of k faces, if is in one of vertexes. If the V tuple is neither a vertex nor an edge, it defines only 1 face. Hence, there is needed minimum V2 -tuples (in opposite vertexes) and maximum 2k V tuples - one in each face of the hyper-cuboid.

For example, a profile in two dimensions is defined by 2,3 or 4 V -tuples (Fig.1).

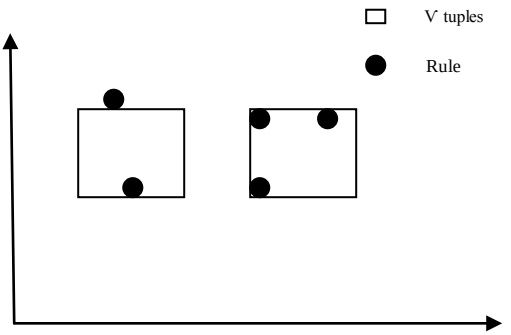

Figure 1: V tuples that define rules.

3.2 Basic Concept of the UB-Tree

The Universal B-tree (UB-tree) was introduced in [39] for indexing multidimensional data. Its main characteristics reside in an elegant combination of the wellknown B+-tree and the Z-ordering. The power of UB-tree lies in linear ordering of vectors, similarly like an ordering of simple values is indexed by the B+-tree.

In the UB-tree we require to establish such ordering on a multidimensional vector space and thus linearize the space onto a single-dimensional interval which is usually realized using space filling curves [6]. A space filling curve orders all the points within a n-dimensional vector space. UB-tree was designed to be used with the

Z-ordering generated using the Z-curve. Points (tuples) in the space are ordered according to their Z-addresses.

An interval [γ : β] (γ is the lower bound, β is the upper bound) on the Zcurve forms a region in the space which is called Z-region. An example of Z-curve and several Z-regions is presented in Figure 2a.

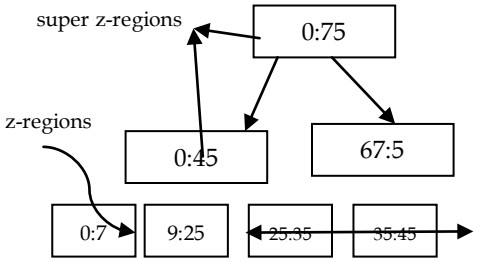

Figure.2 a) The 2-dimensional space 8×8 filled with the Z-curve. The numbers in the grid are the Z-addresses.

The space is partitioned with four Z-regions.

2	4	6	8	10	12	14	16
5	8	13	19	23	27	30	33
7	15	17	20	22	28	31	34
9	16	18	21	24	29	32	35
10	39	61	62	63	64	65	66
11	40	70	71	72	73	74	75
41	51	81	86	91	96	32	21

Figure 2 b) The UB-tree nodes correspond to the Z-regions and super Z-regions.

Each Z-region is then mapped into a single page within the underlying B+-tree. The UB-tree leafs represent the Z-regions containing indexed objects themselves while the inner nodes represent the super Z-regions. A super Z-region contains all the (super) Z-regions lying entirely inside the super Z-region. Hence, the UB-tree structure is determined by a nested Z-region hierarchy. An indexed vector space and its appropriate UB-tree is depicted in Figure 2.

The basic idea of the UB-Tree [13] is to use a space filling curve to map a multidimensional universe to one-dimensional space. Using the Z-Curve Figure 2(a) for preserving multidimensional clustering as good as possible it is a variant of the zkd-B-Tree [14]. The UB-Tree is a multidimensional clustering index, which inherits all good properties of B-Tree [15]. Logarithmic performance guarantees are given for the basic operations of insertion, deletion and point query, and a page utilization of 50% is guaranteed.

The *Z-Address* a = Z(*x*) is the ordinal number of the key attributes of a tuple *x* on the Z-Curve, which can be efficiently computed by bit-interleaving. A standard B-Tree is used to index the tuples taking the Z-Address of the tuples as keys. The fundamental innovation of UB-Trees is the concept of *Z-Regions* to create a disjunctive partitioning of the multidimensional space. This allows for very efficient processing of multidimensional range queries. Z-Region [a : b] is the space covered by an interval on the Z-Curve and is defined by two Z-Addresses a and b. We call b the *region address* of [a : b]. Each ZRegion maps exactly onto one page on secondary storage, i.e., to one leaf page of the B-Tree . For Figure 3(a) shows the regions. Figure 3(b) shows the partitioning. Figure 3(c) corresponding Z-addresses.

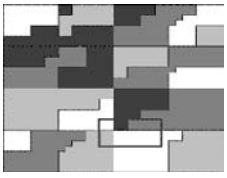

Figure 3(a): Regions

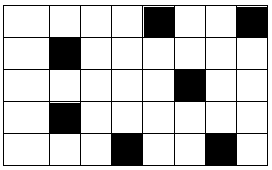

Figure 3(b): Partitioning

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34

Figure 3(c): Z addresses calculation.

3.3 Address Representation and Z-value Computation

An important question for the implementation of the UBTree inside the database kernel is how to represent the Zvalues[15].All algorithms for the UB-Tree basically rely on Z-values in the format of variable length bitstrings (trailing zeros are omitted to reduce storage requirements). The operations on Z-values manipulate single bits and copy parts of the bit string. The UBKEY function can be efficiently implemented, as it requires only reading the specified index attributes bitwise and writing the bits at the corresponding positions in the resulting Z-value. As input the UBKEY function requires a bitstring representation of the attribute values. The natural order □□ of the attribute values in the original domain A has to correspond to the bit-lexicographical order <= bitstr □□ on bitstrings, i.e.,ai □□a j □bitstr(ai) □bitstr bitstr(a j) , where bitstr A□□b: { b|b□ □0,1□□} □generates the corresponding bitstring. For example, in case of unsigned integers and strings bitstr: =□dentity while for signed integers the bitstr function has to take care of the sign bit. To compute the f-value of a tuple, we interleave the bits of the bitstring representation of the key attributes as

shown in Figure 4(a).

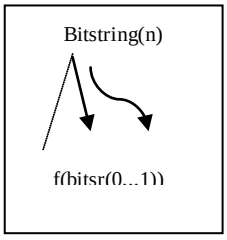

Figure 4(a): Calculation of f-values

4. PROPOSED WORK

The algorithm for generating quantitative association rules starts by counting the item ranges in the database, in order to determine the frequent ones. These frequent item ranges are the basis for generating higher order item ranges using an algorithm similar to Apriori, taking into account the size of a transaction as the number of items that it comprises.

a) Define an item set 'm' as a set of items of size 'm'
b) Specify frequent (large) item sets by 'Fm'
c) Specify candidate item sets (possibly frequent) by 'Lm'.

A 'n' range set is a set of n- item ranges, and each m-item set has a n-range set that stores the quantitative rules of the item set. During each iteration of the algorithm, the system uses the frequent sets from the previous iteration to generate the candidate sets and check whether their support is above the threshold. The set of candidate sets found is pruned by a strategy that discards sets which contain infrequent subsets. The algorithm ends when there are no more candidates' sets to be verified.

Aprori-UB Algorithm

a. Find all frequent item sets (i.e., satisfy minimum support).
b. Generate scaled association rules from the frequent item sets using theorem 21 and 22.
c. Identify the quantitative elements
d. Sorting the item sets based on the frequency and quantitative elements.
e. Merge the more associated rules of item pairs
f. Use UB-tree function to map a multidimensional universe to one-dimensional space.
f. Discard the infrequent item value pairs
g. Iterate the steps c to f till the required mining results are achieved.

Let I = {i1, i2 … i n items} be a set of items, and T a set of transactions, each a subset of I. An association rule is an implication of the form A=>B, where A and B are non-intersecting. The support of A=>B is the percentage of the transactions that contain both A and B. The confidence of A=>B is the percentage of transactions containing A that also contain B (interpret as P (B|A)). The occurrence frequency of an item set is the number of transactions that contain the item set.

5. EXPERIMENTATION

We used java programming language to implement the Aprori-UB algorithm.

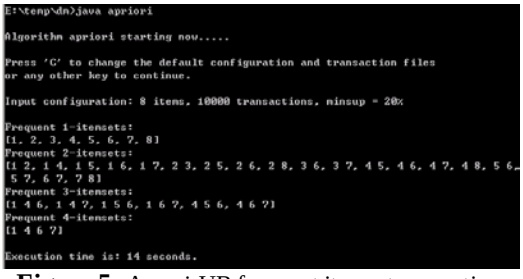

Figure 5: Aprori-UB frequent itemset generation.

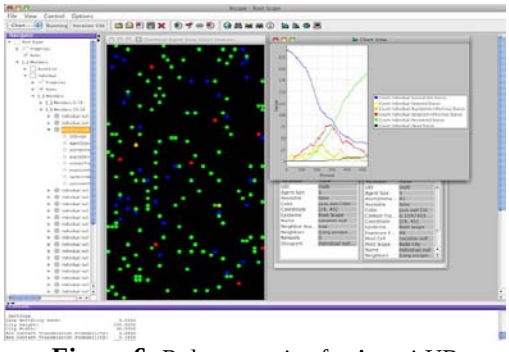

Figure 6: Rule generation for Aprori-UB

We use the data values given in Figure 7(a) for calculation of the Z value as given in Figure 7(b). After the Z -value address calculation we generate the rules with confidence limit as described in Figure 8.

Tuple value
5
10
78
34
68

Figure 7 (a): t values address

Z value
56
123
245
179
234

Figure 7(b): Z value

Code for UBKEY Function

```
Z-value UBKEY (Tuple t) {
int i,s;
int bp; //the bit position in the Z-value
Z-value addr; //the result Z-value
Bitstring bs[dimno]; // bitstring representation of the
//attribute values
//Transformation of the key attributes
```

```
for (i=0; i < d; i++) {
// transformation of the attribute to a bitstring depends on the
// attribute type bs[i] = Transform Attribute(t[i]);
} //Bit-interleaving – Calculation of the Z-value
bp=0; //starting with the first bit of the Z-value
//looping first over dimensions then over steps realizes the
//bit-interleaving
for (s=0;s < steplength; s++) {
for(i=0; i < d; i++) {
// the bpth bit of the Z-value is
// set to the sth bit of the ith bitstring
addr[bp]=bs[i][s];
bp++; //advance to next bit of Z-value
}}
return addr;}
```

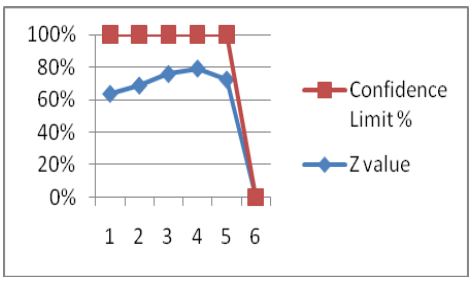

Figure 8: Confidence Limit

We have used proposed Aprori-UB to generate the scaled association rules from the frequent item sets.The results show that the rules generated are more scaled and efficient as depicted in figure 5.The figure 7 show the confidence limit of the rule generated.

6. RESULTS AND DISCUSSION

In order to show the performance of the proposed algorithm, we applied the algorithm to Diabetes Data Set which was obtained from UCI Machine Learning Repository [16]. This dataset is Multivariate, Time-Series and has 20 attributes. After discovering rules, they have to be presented in understandable form to the user.

In multi-dimensional dataset, say for n - 10, the range query efforts rapidly increase. This fact is caused by the curse of dimensionality described later in this section. In practice, the disk access costs and the number of computations grow with the increasing dimensionality.

UB-tree characteristics

card(D)	232	dimensions	2–30
tuples	524,288–7,864,320	tree height	4
nodes	22,400–321,885	Z-regions	21,475–321,885
node capacity	35	utilization	69.7–69.8%
node size	580–4612B	index file	12.4MB–1.44GB

Crucial problems with multi-dimensional quantitative associations are the need to determine the most significant rules and to distinguish between groups of rules with very similar profiles [4]. This problem may be solved by the application of specific quality measures in a rule management system [10].

One kind of measures determines the significance of the rule. Such measure may be support, difference (diff (P)), volume (V (P)), rule density (_(P)) or differential density (_diff (P)), defined as follows:

$$diff(P) = M(P) - M(R) \quad (1)$$

$$V(P) = \pi(v_i - u_i) \quad (2)$$

$$p(P) = \frac{supp(P)}{V(P)} \quad (3)$$

$$pdiff(P) = \frac{supp(P)diff(P)}{V(P)} \quad (4)$$

$$cons(P) = 2^k$$

We have shown a comparison of density calculation in Figure 9 . The rule density that we have proposed have high and are able to generate scaled rules with high support . As it was noticed in Section 3, irreducibility is not always enough to determine intuitively homogeneous rules. Figure 2 shows some examples of not uniform distribution of µ-tuples in irreducible 2D rules.

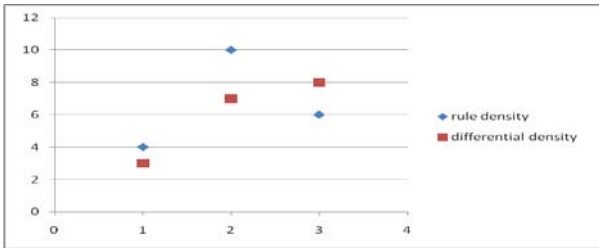

Figure 9. Density calculation in Diabetes dataset

There are various formulas possible for measuring consistence of the rule. For instance let us consider that the rule profile is divided into 2k equal hypercuboid parts (Pr1; : : : ; Pr2k) by splitting all attribute ranges in two. Consistence then may be expressed as:

$$cons(P) = \frac{2^k}{\sum 1 <= i,j <= 2^k |M(Pr^i) - M(Pr^j)|} \quad (5)$$

Another kind of measures are used for comparison of rules - to determine if the profiles are close or distant. Such measure may be common support Csupp or common volume CV or mean intra-rule distance L_{ir} between tuples.

$$C_{supp}(P1, P2) = |\{t : t \in Pr(P1) \cap Pr(P2)\}| \quad (6)$$

$$CV(P1,P2) = V(Pr(P1) \cap Pr(P2))$$

(7)

$$L_{ir}(P1, P2) = \frac{\sum(i=1...N1)\sum(j=2...N2)(tP1,tP2)}{N_1N_2} \quad (8)$$

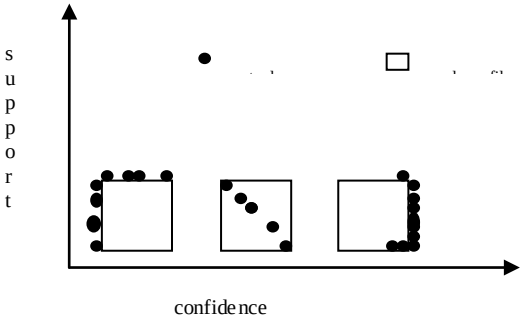

Figure 10. Rules Generated by Aprori-UB

We have represented the rules generated by Aprori-UB in figure 10.

6.2 COMPARISON

The experiment focused on evaluating Scale kmeans,Aprori and Aprori-UB techniques. Since we were interested in seeing the best performance, we used diabetes data set samples. The minsupp , minconfidence level and average rule error was compared in figure 11. The evaluation shows that our proposed Aprori-UB generated strong association rule with less rule generation error.

Algorithm	Minsupp	MinConf	Average rule error
Scale K-means	5%	6%	0.163
Aprori	3%	4%	0.178
Proposed Aprori-UB	6%	6.7%	0.092

Figure 11. Performance of different association rule generation methods

The rule generation error when the algorithm is not able to store the itemset at continues location and it takes time to execute.In our scaled Aprori-UB we use the concept of *Z-Regions* to create a disjunctive partitioning of the multidimensional space. This allows for very efficient processing of multidimensional range queries and generation of Strong association rules.Our proposed Aprori-UB generate scaled and strong association rule with 6%,6.7% and 0.092 average rule error which is much efficient than the previous methods as given the figure 12.The experiments show that the number of rules generated in our proposed algorithm are more in number with less average rule generating error.Figure 10 shows the comparison details of the other algorithm with our proposed scaled Aprori-UB algorithm.

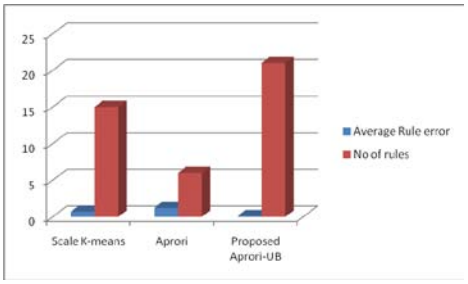

Figure 12. *Average rule error vs. no of rules generated.*

The proposed work has characterized the relationships between the informative rule set and the nonredundant association rule set, and revealed that the informative rule set is a subset of the nonredundant association rule set. The work considers the upward closure properties of informative rule set for omission of uninformative association rules, and presented a direct algorithm to efficiently generate the informative rule set without generating all frequent item sets.

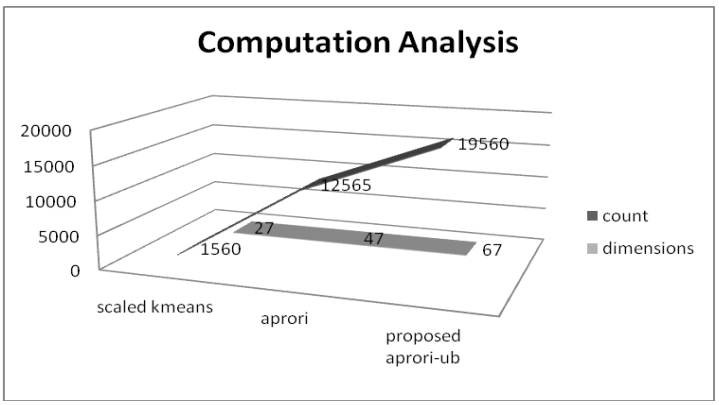

Figure 13: Computation analysis of the algorithms

In Figure 13, the count and the number of computations are presented. We can see that with increasing dimensionality the rules grow. However, the growth for the proposed Aprori-UB algorithm is much than the k-scaled means and aprori algorithm. Thus, the proposed aprori-UB algorithm is more effective for multidimensional dataset. The reason of the aprpri-UB algorithm's success resides in the application of the leaf optimizations.

7. CONCLUSION

The experimental results have shown that the proposed Aprori –UB algorithm makes the UB-tree applicable for effective indexing and querying of multidimensional databases. In this paper we have presented an Aprori-UB, a scaled approach towards generating association whose results have shown that it is the superior

of the other two algorithms. In this paper we have used Breadth-First Search (BFS) strategy to traverse the search space.With BFS the support values of all (k-1) itemsets are determined before counting the support value of k-itemsets. End users of association rule mining tools encounter several well known problems in practice. First, the algorithms do not always return the results in a reasonable time. It is widely recognized that the set of association rules can rapidly grow to be unwieldy, especially as we lower the frequency requirements.

The scaled quantitative association rules in the form presented in this paper are applicable to any form of numeric data and have clear advantages. Data-driven algorithms for rule discovery have polynomial complexity, and are additionally speed up by heuristic strategies. Multidimensional access methods are not widely supported by commercial database management systems despite their performance impacts in various application domains. This is mostly due to the fact that a kernel integration of these sophisticated data structures is considered to be a very costly and complex task. In this paper we have shown that this is not the case for the UBTree, as it heavily relies on the well-known B-Tree, reducing the complexity of the additional algorithms to a minimum. Profile boundaries are determined by the data themselves, without errors indicated by the static discretization. Input data may be sampled even at random. Output rules, especially mean based, are understandable and may be easily visualized because a square or hyper-cuboid is very intuitive in its perception.

8. FUTURE SCOPE

All the algorithms and strategies are currently under rigorous experimental examination that will be described in some follow-up papers. Other future work in this field includes discovery algorithms with dynamic changes of μ level, improved performance strategies and new measures for rule management. The knowledge discovery methodology may be even closer linked to spatial-temporal databases by new pre-processing and visualization techniques. It is also expected that quantitative association rules will be applicable to other forms of numeric data.

REFERENCES

[1] Agarwal R. and V. Prasad :-"A Tree Projection Algorithm for Generation of Frequent Itemsets," Parallel and Distributed Computing, (2000).

[2] S.Prakash and R.M.S.Parvathiv:-" An Enhanced Scaling Apriori for Association Rule Mining Efficiency"European Journal of Scientific Research ISSN 1450-216X Vol.39 No.2 (2010), pp.257-264 .

[3] Bayer, R.:-"The universal B-tree for multidimensional indexing", Institute for Informatic, TUMünchen Technical Report (1996).

[4] Lindell, Y., Aumann, Y-"Theory of Quantitative Association Rules with Statistical Validation", Proceedings of SIGKDD Conference, Boston, (1999).

[5] Miller, R.J., Yang, Y:-"Association Rules over Interval Data", Proceedings of ACM SIGMOD 97 Conference (1997).

[6] Gawrysiak, P., Okoniewski:- "M. Applying data mining methods for cellular radio network planning, Intelligent Information Systems", Springer-Physica Verlag (2000).

[7] Geade, V. Günter, O:-"Multidimensional Access Methods", ACM Computing Surveys, 30(2), (1997) .

[8] Srikant, R., Agrawal, R.:-"Mining Quantitative Association Rules in Large Relational Tables", Proceedings of VLDB-96 Conference (1996).

[9] Markl V. Mistral:-"Processing Relational Queries using a Multidimensional AccessTechinque", PhD Thesis, Institute for Informatice, TUMünchen (1999) .

[10] Okoniewski, M. Discovering Quantitative, Multidimensional Association Rules, Ph.D. Thesis (draft), Warsaw University of Technology (2001).

[11] Skowron, A., Nguyen, S.H:- "Quantization of Real Value Attributes: Rough Set and Boolean Reasoning Approach", Warsaw University of Technolgy Technical Report (1995).

[12] Michal Okoniewski, Ukasz Gancarz, Piotr Gawrysiak:-"Mining Multi-Dimensional Quantitative Associations." Institute of Computer Science, Warsaw University of Technology.

[13] R. Bayer:-"The universal B-Tree for multidimensional Indexing: General Concepts". World-Wide Computing and Its Applications '97 (WWCA '97). Tsukuba, Japan, 10-11, Lecture Notes on Computer Science, Springer Verlag, March, 1997.

[14] J. A. Orenstein and T.H. Merret:-"A Class of Data Structures for Associate Searching". Proc. of ACM SIGMOD-PODS Conf., Portland, Oregon, 1984, pp. 294-305.

[15] Frank Ramsak1, Volker Markl1, Robert Fenk1, Martin Zirkel2, Klaus Elhardt3, and Rudolf Bayer:"Integrating the UB-Tree into a Database System Kernel".

[16] Michael Kahn.:-"UCI Repository of Machine Learning Databases ", 1994. http://archive.ics.uci.edu/ml/datasets/Diabetes.

[17] R. Bayer and E. McCreight:-"Organization and Maintainance of large ordered Indexes". In Acta Informatica 1, pages 173–189, 1972.

[18]Tamanna Siddiqui,M Afshar Alam ,Sapna Jain Discovery of Scaled Association Rules from multidimensional quantitative Datasets.[2011]

[19] Xindong Wu ,Vipin Kumar , J. Ross Quinlan , Joydeep Ghosh Qiang Yang ,Hiroshi Motoda , Geoffrey J. McLachlan , Angus Ng Bing Liu ,Philip S. Yu ,Zhi-Hua

Zhou , Michael Steinbach , David J. Hand , Dan Steinberg :-"Top 10 algorithms in data mining", © Springer-Verlag London Limited 2008, Knowl Inf Syst (2008) 14:1–37,DOI 10.1007/s10115-007-0114-2

[20] V. Venkata, Ramana,M V Rathnamma,A. Rama Mohan Reddy :-" Methods for Mining Cross Level Association Rule In Taxonomy Data Structures" ,International Journal of Computer Applications (0975 – 8887),Volume 7– No.3, September 2010

[21]Mohammad Naderi Dekhordli, Kambiz Badie, Ahmad Khadem,Adeh :- "A new approach for sensitive association rule hiding", International Journal of Rapid manufacturing 2009 Vol 1,No 2 pg(128-129).

[22] Karla Taboada Non-member, Shingo Mabu Member, Eloy Gonzales Non-member, Kaoru Shimada Member, Kotaro Hirasawa –"Mining Fuzzy Association Rules: A General Model Based on Genetic Network Programming and its Applications", IEEJ Transactions on Electrical and Electronic Engineering Volume 5, Issue 3, pages 343–354, May 2010

[23] Rupali Haldulakar,Prof Jitendra Agarwal - "Optimization of Association Rule Mining through Genetic Algorithm", International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397 Vol. 3 No. 3 Mar 2011,pg [1252-1259]

[24] Y.Zhu, H.Zhang and L.D.Kong:-"Research and application of multidimensional association rules mining based on artificial immune system", Computer Science, vol. 36pp.239-242, 2009.

[25] F.Q.Shi, S.Q.Sun, and J.Xu:- "Association rule mining of Dansei knowledge based on rough set," Computer Integrated
Manufacturing Systems, vol.14, pp.407-411, 2008.

[26] W.S.Yao, L.Shang, and Z.Q.Chen :- "A quantization of real-value attributes based on evolution algorithm", Computer Applications and Software, vol. 22, pp.37-39, 2005.

[27] Z.Tong, K.Luo:- "Mining of association rules with decision attributes based on rough set", Computer Engineering and
Applications, vol 42, pp.166-169, 2006.

[28] J. Han and M. Kamber, Data Mining :-" Concepts and Techniques", San Francisco: Morgan Kaufmann Publishers,2001.

[29] Jun Jie Cen, Guo Hong Gao, Ying Jun Wang :-" An Efficient Genetic Simulated Annealing Assiociation Rules Method", Journal of Applied Mechanics and Materials, Jun Jie Cen et al., 2010, Applied Mechanics and Materials, 34-35, 927.

[30] Shichao Zhang, ,Xindong Wu :-"Fundamentals of association rules in data mining and knowledge discovery", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery Volume 1, Issue 2, pages 97–116, March/April 2011.

[31]Tan P. and Kumar V.:- "Interesting Measures for Association Patterns: A Perspectiva, Technical Report", TR00-036. Department of Computer Science. University of Minnnesota, 2000.

[32] Silberschatz, A. and Tuzhilin, A.:-"What makes patterns interesting in Knowledge discovery systems". IEEE Trans. on Knowledge and Data Engineering. 8(6), 970-974, 1996.

[33] Liu B., Wynne H., Shu C. and Yiming M.:- "Analyzing the Subjective Interestingness of Association Rules ". IEEE Intelligent Systems and their Applications. 15:5, 47-55, 2000.

[34] Agrawal R., Imielinski, T., and Swami, A. N.:-- "Mining association rules between sets of items in large databases". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[35] Agrawal, R. and Srikant, R.:- "Fast algorithms for mining association rules". In Proc.20th Int. Conf. Very Large Data Bases, 1994, 487-499.

[36] Sotiris Kotsiantis, Dimitris Kanellopoulos: - "Association Rules Mining: A Recent Overview", GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pg. 71-82

[37] Ashrafi M., Taniar, D., Smith, K.:-"Redundant Association Rules Reduction Techniques", Lecture Notes in Computer Science, Volume 3809, 2005, pp. 254 – 263

[38] M.Anandhavalli, M.K.Ghose, K.Gauthaman:- "Association Rule Mining in Genomics", International Journal of Computer Theory and Engineering, Vol. 2, No. 2 April, 2010 1793-8201

[39] R. Bayer. The Universal B-Tree for multidimensional indexing: General Concepts.In Proceedings of World-Wide Computing and its Applications'97, WWCA'97, Tsukuba, Japan, 1997.

AUTHORS BIOGRAPHY

1. Dr. M Afshar Alam is professor in Department of Computer Science, Jamia Hamdard,New Delhi.He has teaching experience of more than 17 years.He has authored 8 books and guided PhD research works.He has more than 30 publications in international/national/journal/conference proceddings. He has delivered special lectures as a re-source person at various academic institutions and conferences He is a member of expert committees of UGC,AICTE and other national and international bodies.His research areas include software re-engineering,data mining, bioinformatics and fuzzy databases.

2. Sapna Jain is a Phd Fellow in the Jamia Hamdard University who has obtained her MCA (Masters of Computer Application) degree from Maharishi Dayanand University, ndia. Her area of research is Scalability of data mining algorithms.

3. Presently he is the Professor and Head of Computer Science Department ,Jamia Hamdard University ,New Dehi. Dr Ranjit Biswas has taught in IIT Kharagpur NIT Agartala and Calcutta University.He is a member of Editorial Board of many International journals in repute .