Discovery of Multi Dimensional Quantitative Closed Association Rules by Attributes Range Method

Preetham Kumar, Ananthanarayana V S

***Abstract*— In this paper we propose a novel algorithm for discovering multi dimensional association rules with right hand side containing closed frequent itemsets of varying length, by finding the boundaries of attributes domains dynamically. It first builds an abstraction, called Multi-Variate Tree, in a single scan of the database. During this construction the boundaries of domains of quantitative attributes are identified dynamically. These identified attributes boundary values associated with frequent itemsets are then used for finding closed association rules. When a transaction database is very dense and the minimum support is very low, i.e., when the database contains a significant number of large frequent itemsets, mining all frequent itemsets might not be a good idea. The concept of closed frequent itemsets solves this problem.**

***Index Terms*— Boundary values, Closed itemset, MVT, Path, Support.**

I. Introduction

The need to extract the association rules from the pool of varied data, having gained increased momentum in the field of data mining, necessitates the discovery of methods to process multi-dimensional data, and find the quantitative association rules from it by considering all the relevant fields in an efficient manner. Our algorithm first discovers the boundaries of attributes domains dynamically by building an abstraction, called Multi-Variate Tree, in a single scan of the database. During this construction the boundaries of domains of quantitative attributes are identified dynamically. These Identified attributes with boundary values with associated frequent itemsets obtained from MVT are then used for discovering quantitative association rules containing closed frequent itemsets. An itemset is closed if none of its proper supersets has the same support as it has. A frequent itemset is *maximal* if none of its proper supersets is frequent. The total number of maximal frequent itemsets *M* is much smaller than that of frequent itemsets *F*, and we can derive each frequent itemset from *M*. However, *M* does not contain information of the support of each frequent itemset unless it is a maximal frequent itemset. Thus, mining only maximal frequent itemsets loses information. However, when a transaction database is very dense and the minimum support is very low, i.e., when the database contains a significant number of large frequent itemsets, mining all frequent itemsets might not be a good idea. The concept of closed frequent itemsets solves this problem.

The total number of closed frequent itemsets *C* is still much smaller than that of frequent itemsets *F*. Furthermore, we can derive *F* from *C*, because a frequent itemset *I* must be a subset of one (or more) closed frequent itemsets, and *I*'s support is equal to the maximal support of those closed itemsets that contain *I*.

Preetham Kumar is with the Information and Communication Technology Department, Manipal Institute of Technology, Manipal, 576104, Karnataka India (phone : 91 - 9845367692 e-mail:prethk@yahoo.com).

Ananthanarayana V S is Professor and Head in the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India (e-mail: ananthvs1967@gmail.com).

II. Proposed Algorithm

The algorithm is divided into three main phases. The first phase is to build an efficient tree, called Multi-Variate Tree**,** in a single scan of database using the items as well as relevant information in each transaction**.** In the second step, MVT is scanned to obtain the quantitative association rule between frequent item-sets and the corresponding information fields. The third phase uses the result obtained in the above to discover only closed frequent itemsets with associated information.

A. Construction of Multi-Variate Tree, and Finding Boundary Values of Attributes for Item-sets

Structure of a Node in MVT

Each **node** in the MVT has following two parts, as shown in Figure 1:

{ *item part:*
inum: item number
count: support count
clink : link to child nodes
slink :link to sibling nodes
ilink: link to the first relevant information node.
relevant-information part:

Each Relevant-info node has the structure:

{ **range**[2] : // stores min/max values
flink : link to next Relevant-info node .
}
}

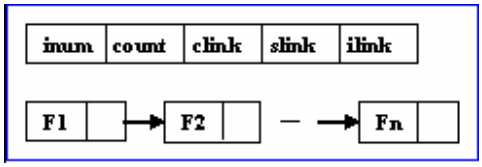

Figure1 : Node structure in a MVT

Algorithm for Construction of MVT
Input: D, a transaction database, in which items are considered in an increasing order of item numbers.
Output: The MVT
Method:
Create a root node of the MVT and label it as "root". Let each transaction be represented by Trans [P| F], where P is the set of items bought, and F is the set of relevant information Fields, with F*i* as its ith field.
Initialize Trancount, the total no. of transactions in the database, to 0.
For each transaction Trans in database begin
Increment Trancount by 1.
Create a relevant info node R with all the fields setting its value*R[Fi].min-value* and *R.[Fi].max- value* to *Trans[Fi].value* for every i
Let p be the first item in the set P,
F*i* be the ith field in the set F, and T denote the current node (starts with root node).
Call insert_tree(Trans)
end

insert_tree(Trans)
{
if T has a child N such that
N.item-number=p.item-number, then increment N's count by 1.
//updating the information of N
For each Fi in F begin
if N[Fi].min-value > Trans.[Fi].value
N[Fi].min-value = Trans.[Fi].value
if N[Fi].max- value< Trans.[Fi].value
N[Fi].max-value =Trans.[Fi].value
end
move to next item in P. i.e consider p as next item in P
If P is nonempty
T=N
insert_tree(Trans)
else
Do until P is non-empty begin
Create a new node N, and initialize its count to 1 and is connected as a child(using clink) or one of the children (using slink)of T .
Set T=N and increment p to point to next item in the item-set P.
end
connect p(last item) of the transaction to relevant information node R
}

B. Finding n-dimensional quantitative association rules with frequent itemsets in the right hand side.

Input: MVT.
min_sup: the minimum support count
Output: The set of frequent itemsets, and the range of values of information fields, i.e. attributes, associated with them, depending on the min_sup. i.e n-dimensional association rules with varying itemsets in the right hand side.
Method:
Traverse MVT in preorder
For every path p with common relevant information field begin
If count(p)> min_sup
Add p to P.
end

C: Discovering multi dimensional quantitative association rules containing closed frequent itemsets as right hand side.

Input: min_conf: user defined minimum confidence**.**
Output: Closed association rules
Method:
// Determining the total count of p in P by ignoring the relevant information.
For every path p in P begin
Traverse MVT to get the total count of p in P
end
// Filtering non closed frequent path p in P.
For every path p in P begin
If p is equal to any other path q in P
Remove p from P
If p is contained in any other q in P with same count
Remove p from P
End
//Recovering closed frequent subset of p in P.
For every subset s of p in P begin
Scan MVT to get count of s
If there exists no proper superset containing s with same count as s
Then add s to P
End

//Discovering association rules between closed frequent itemsets with respect to the relevant information
for every closed itemset c in P begin
for every subset s of c begin

Output a rule of the form **s=>c-s** if count(c)/count(s) is >= min_conf.
End
End

III. ILLUSTRATION

We now apply the above algorithm to a sample database,

shown in Table 1, where we have the information about the age, basic salary and HRA of each customer, and the various items bought by the customer. The corresponding tree structures, in different stages are shown in the Figure 2 through Figure 4.
Figure 4 shows the complete tree structure after processing the 10th transaction.

The nodes in the figures showing MVT are shown numbered, for the purpose of illustration, and only the item number in the nodes are shown. Other information fields have not been shown individually in each of the nodes of MVT, in order to keep the figures manageable and neat. Instead, the range of values of attributes common to an itemset in the tree has been shown by a separate ellipse, as shown in Figure 2 and Figure 3.

Table 1: Sample database

Age	Basic	Hra	Items
43	4886	2777	1,2,4,5,6,7,8,9
28	6567	2429	1,2,3,4,5,6,7,8,9
44	12537	7198	2,3,4
51	12980	5956	6
30	10996	7281	1,3,4,6,7,8,9
23	9857	4124	1
55	12814	7367	6,7,8
18	11276	3178	1,2,3,4,7,9
49	4012	2226	4
49	4795	6570	2,4,9

In the above Table the age is in years, basic salary in rupees and HRA in rupees. Age, basic and HRA form the relevant information part of the node, where as the item numbers and their count comprise the item part of the node.

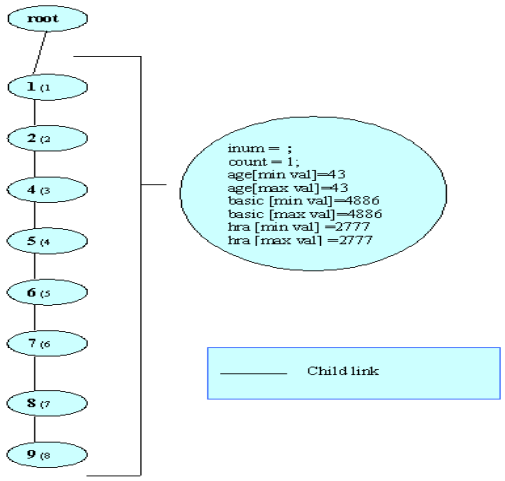

Figure 2. Tree structure after 1st transaction

The tree structure after processing of the first transaction (age=43; basic=4886; HRA=2777; items: 1,2,4,5,6,7,8,9) is shown in Figure 2. The MVT starts with a root node, and the nodes, bearing the item number, are created in the order of increasing item number. Since each of the nodes, bearing the respective item number, are newly created, the minimum and the maximum value for each of the associated information fields age, basic, HRA are the same. In this case, each of the nodes numbered from 1 to 8, has the following values in their information fields.
age[min]=age[max]=43,
basic[min]=basic[max]=4886,
HRA[min]=HRA[max]=2777,
and the count value of each of the items is 1.

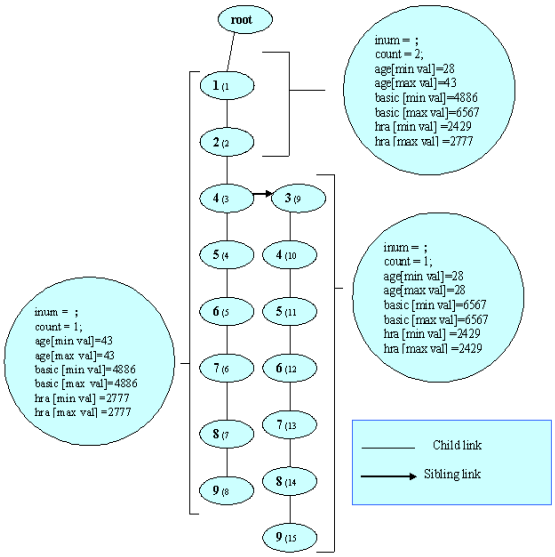

Figure 3. Tree structure after 2nd transaction

After the processing of the second transaction (age=28; basic=6567; HRA=2429; items:1,2,3,4,5,6,7,8,9), the MVT has the structure as shown in Figure 3. Here, since item number 1 was already present under the root, we increment its count value by 1, and update its information field as follows. As 28 is not greater than the stored age[max] value of 43, we leave age[max] value unchanged. However, 28 is less than the stored age[min] value of 43. so, age[min] value is updated to a new minimum value of 28. Next, the basic salary of rupees 6567 is greater than the basic[max] value , but not less than the basic[min] value. Hence, basic[max] value is updated to 6567, where as basic[min] remains unchanged at 4886. Similarly HRA[min] is updated to 2429. Again, the node containing item number 2 already exists as a child of node1. Therefore, its count is incremented by 1, and its associated information fields are updated in a similar manner. Thus, the values contained in the relevant information fields of node 1 and node 2, after the 2nd transaction, are as follows:
age[min]=28, age[max]=43;
basic[min]=4886, basic[max]=6567;
HRA[min]=2429, HRA[max]=2777;

The next item in the transaction is 3. As node number 2 has no child with item number 3, a new node with item number 3 is created under node number 2, as a sibling node of node number 3 having item number 4, with count=1, and other information field values as:
age[min]=age[max]=28;
basic[min]=age[max]=6567;
HRA[min]=HRA[max]=2429;

Similarly, other nodes are created for the rest of the items in the 2nd transaction. After processing the 2nd transaction, the structure of MVT, and the values contained in its nodes are shown in Figure 3.

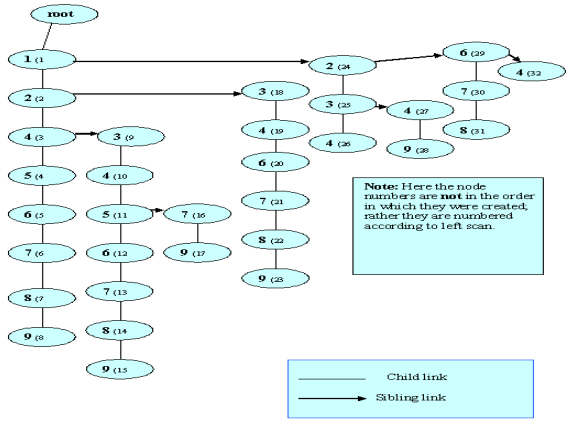

Figure 4. Tree structure after 10th transaction

The complete structure of MVT, after the 10th transaction, is shown in Figure 4. The relevant information corresponding to all the nodes after the 10th transaction are shown in Table.2

Table2: Values contained in the nodes of the MVT shown in Figure 4

Node no.	inum	count	Age-range	Basic-range	Hra-range
1	**1**	5	18-43	4886-11276	2429-7281
2	**2**	3	18-43	4886-11276	2429-3178
3	**4**	1	43	4886	2777
4	**5**	1	43	4886	2777
5	**6**	1	43	4886	2777
6	**7**	1	43	4886	2777
7	**8**	1	43	4886	2777
8	**9**	1	43	4886	2777
9	**3**	2	18-28	6567-11276	2429-3178
10	**4**	2	18-28	6567-11276	2429-3178
11	**5**	1	28	6567	2429
12	**6**	1	28	6567	2429
13	**7**	1	28	6567	2429
14	**8**	1	28	6567	2429
15	**9**	1	28	6567	2429
16	**7**	1	18	11276	3178
17	**9**	1	18	11276	3178
18	**3**	1	30	10996	7281
19	**4**	1	30	10996	7281
20	**6**	1	30	10996	7281
21	**7**	1	30	10996	7281
22	**8**	1	30	10996	7281
23	**9**	1	30	10996	7281
24	**2**	2	44-49	4795-12537	6570-7198
25	**3**	1	44	12537	7198
26	**4**	1	44	12537	7198
27	**4**	1	49	4795	6570
28	**9**	1	49	4795	6570
29	**6**	2	51-55	12814-12980	5956-7367
30	**7**	1	55	12814	7367
31	**8**	1	55	12814	7367
32	**4**	1	49	4012	2226

Now, let us find the item-sets and the boundary values of attributes associated with the item-sets, applying the above algorithm to MVT depicted in Figure 4.
Corresponding to the transaction head {1} some of the Paths associated with relevant information are
(i) {1, 2, 3, 4, 5, 6, 7,8,9 }:1
Age-range=28;
Basic-range=6567;
HRA-range=2429;

(ii), {1,2,4,5,6,7,8,9}:1
Age-range= 43;
Basic-range=4886;
HRA-range=2777;
(iii){1,2}:3
Age-range= 18-43;
Basic-range=4886-11276;
HRA-range=2429-3178;
(iv){1}:5
Age-range= 18-43;
Basic-range=4886-11276;
HRA-range=2429-7281;

All remaining paths with associated information, count and boundary values obtained are shown table 3. This also gives the multi dimensional quantitative association rules.

Table3: Itemsets and the boundary values of attributes associated with them.

itemset	count	Age-range	Basic-range	Hra-range
4	1	49-49	4012-4012	2226-2226
6,7,8	1	55-55	12814-12814	7367-7367
6	2	51-55	12814-12980	5956-7367
2,4,9	1	49-49	4795-4795	6570-6570
2,3,4	1	44-44	12537-12537	7198-7198
2	2	44-49	4795-12537	6570-7198
1,3,4,6,7,8,9	1	30-30	10996-10996	7281-7281
1,2,3,4,7,9	1	18	11276-11276	3178-3178
1,2,3,4,5,6,7,8,9	1	28-28	6567-6567	2429-2429
1,2,3,4	1	28-28	6567-11276	2429-3178
1,2,3	1	28-28	6567-11276	2429-3178
1,2,4,5,6,7,8,9	1	43-43	4886-4886	2777-2777
1,2	3	18-43	4886-11276	2429-3178
1	5	18-43	4886-11276	2429-7281

The exact count of each path given in the above table is shown in the table 4.

Table 4: Count of paths in Table3.

Itemset	Count
4	6
6,7,8	4
6	5
2,4,9	3
2,3,4	3
2	5
1,3,4,6,7,8,9	2
1,2,3,4,7,9	2
1,2,3,4,5,6,7,8,9	1
1, 2, 3,4	2
1,2,3	2
1,2,4,5,6,7,8,9	2
1,2	3
1	5

Now applying above algorithm for the Table 4, some of the closed frequent itemsets with their count are listed in the following table5

Table 5: Closed itemsets

Closed itemset	count
4	6
6,7,8	4
6	5
2,4,9	3
4,9	5
2,4	5
3,4	4
2	5
1,3,4,6,7,8,9	2
1,3	3
1,4	4

IV. EXPERIMENTAL RESULTS

We tested this methodology against the standard IBM synthetic data[7], that contained a wide range of item numbers (more than 900 items) in a transaction, to find only multi dimensional quantitative association rules. To this data, we appended the three attribute fields namely age, basic salary and HRA, using data generating programs based on randomization functions, in order to ensure that the final test data remained unbiased. The data contained 49,100 transactions. The number of rules obtained under different values of constraints, such as minimum support and minimum size of item-sets, is shown in the following table. By increasing the support threshold and minimum size of item-sets, we can filter out the desired rules to get the result, showing the boundary values of attributes for only those item-sets that satisfy the minimum threshold criteria.

Table 6:Number of itemsets obtained with IBM data[7] for different values of support threshold and minimum item-set size

Minimum support count %	Minimum size of itemset	Number of Rules obtained
0.002	1	53573
0.004	1	11291
0.006	1	6428
0.008	1	4437
0.01	1	3403
0.012	1	2782
0.004	2	10947
0.004	3	6012
0.004	4	2228
0.004	5	1087
0.004	6	680
0.004	7	518
0.01	5	286

Some of the rules obtained for different threshold shown in the table 7 .

Table 7: A Sample of results obtained with Different threshold

Item numbers of itemsets	Count %	Age-range	Basic-range	Hra-range
19,556,770,811,868,879	0.016	24-59	577913216	2363-6276
8,73,118,467,569	0.06	12-59	4183-13374	2052-7928
8,328,379,380,680	0.226	10-59	4046-1398	2002-7892
8.328,379,380,608,717	0.208	10-59	4046-13978	2002-7892
8.328,379,380,608,717,733	0.2	10-59	4046-13978	2002-7892
8.328,379,380,608,717,809	0.18	10-59	4046-13978	2002-7892
8.328,379,380,608,717,733,809,965	0.174	10-59	4046-13978	2002-7892
8.328,379,380,608,717,733,965	0.012	14-48	5871-12002	3878-7248
8,328,379,608,717	0.012	12-48	4692-11093	2121-7760
8,109,129,319,939	0.024	20-59	4141-12811	3083-7508

V. CONCLUSION

In this paper we have discussed a novel method for discovering multi dimensional quantitative association rules with frequent itemsets or closed frequent itemsets of varying length in the right hand side of the rule, by using an abstraction called MVT. This approach uses a single scan of the database. While mining an extremely large transaction database, we should discourage the use of algorithms that require multiple scans of the massive database, or create huge data structures that discover frequent itemsets. What is required is a method for developing compact data structure, in minimum number of scans of database, that can store the entire relevant information in a compact form to obtain desired association rule. After having obtained promising results from the IBM synthetic data, next we are going to obtain live data from a Cancer Hospital, and use this

algorithm to obtain quantitative rules pertaining to various diagnostic parameters, that can help in determining the chances of cancer, as well as those pertaining to the treatment process. Also, we have implemented the algorithm to discover closed itemsets for small databases using a single scan of database. We would like to apply our algorithm to very large databases and compare the results with the existing closed frequent itemsets algorithms as a future enhancement

REFERENCES

[1] R. Srikant and R. Agarwal. Mining generalized association rules, *Proceedings of International Conference on Very Large Data Bases 1995*, pages *407-419*.

[2] B.Lent, A.Swami, and Jwidom, Clustering Association Rules, *Proceedings of International conference on Data Engineering* pages *220-231*, Birmingham, England, April1997.

[3] Jiawei Han Micheline Kamber, Data Mining : Concepts and Techniques .Morgan Kaufman, San Francisco, CA, 2001

[4] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, Mining Association Rules between Sets of Items in Large databases, *Proceedings of the 1993 ACM SIGMOD Conference* Washington DC, USA, May 1993

[5] Rakesh Agarwal, Ramakrishnan Srikant, Fast algorithms for mining Association Rules, *Proceedings of the 20th VLDB Conference* Santiago, Chile, 1994

[6] S.Y.Wur and Y.Leu, An effective Boolean Algorithm for mining Association Rules in large databases, *The 6th International conference on Database systems for Advanced Applications, 1999, pp 179-186.*

[7]A Savasere, E. Omiecinski, and S. Navathe An efficient algorithm for mining association rules in large databases. *In Proc. Of the VLDB Conference, Zurich, Switzerland.*

[8] J.Han, J.Pei, and Y.Yin. Mining frequent patterns without candidate generation. *In ACM-SiGMOD, Dallas, 2000.*