Discovery of Multidimensional Association Rules Focusing on Instances in Specific Class

Hyontai Sug

***Abstract*—** Conventional association rule finding algorithms as well as multidimensional association rule finding algorithms search association rules based on support, so it is not easy to find association rules of specific class with small support due to computational complexity. In order to overcome the problem of intensive computing time and to avoid the possibility of generating a lot of uninteresting rules, a method that can reduce the intensive computing time and generate smaller number of multidimensional association rules is suggested. By limiting the search for association rules to a specific class in target data set and by selecting instances that have at least one common field value with all instances in the class, the method can reduce the target data set significantly so that computing time can be saved and also smaller number of rules can be generated. Experiments with a real world data set showed a very good result.

***Keywords*—** Class instances, multidimensional association rules, features values.

I. Introduction

Association rules as well as multidimensional association rules are high topic of interest in many fields of data mining tasks. The application areas of the rules are like network analysis [1], software project management [2], a customer resource management in enterprise resource planning [3], DNA pattern recognition [4], book recommendation in library [5], etc. Because association rules are rules that describe association patterns that indicate how likely a set of items occur together, these patterns reveal some information on regularities that exist in a database. For example, suppose we have collected a set of transactional data records of sales in a super market for some period of time. We might find an association rule from the data stating, "bread => jam (70%)," which means "70% of customers who buy bread also buy jam." Such association rules may be helpful in decision making on sales, e.g., designing an appropriate layout for items in the super store.

While association rules can be found from data sets of no attributes, multidimensional association rules are rules that can be found from a table-structured database. The main difference between association rules and multidimensional association rules is that each item in multidimensional association rules has distinct attributes, but there is no such attribute in the items of association rules. In that sense association rules are regarded as one dimensional multidimensional association rules. So, each item in multidimensional association rules is a pair {attribute, a value of the attribute}.

If the attributes of the table-like database can be divided into two kinds of attributes, condition and decision attributes, we can find multidimensional association rules for classification. Differently from conventional multidimensional association rule discovery algorithms in which there are no predefined distinction between the roles of attributes, we want to give attention to the table that has a decision attribute and many conditional attributes, and each conditional attribute is independent on each other, and dependent only on the decision attribute. So, smaller number of rules can be generated than the case in which the association rules are generated by conventional multidimensional association rule algorithms.

Moreover, if we want to find multidimensional association rules for a specific class having specific feature vectors, and if we want to find association rules from all instances that have at least one common attribute value with the instances in the class, we can narrow down the target instances from the target data set. For example, suppose that we want to find multidimensional association rules to decide the sickness of patients in a specific disease from test results that are done as common tests to detect diseases, and the disease is minor compared to other disease, so the instances that indicate this disease belong to a minor class. Therefore, it is highly possible that we may have many uninteresting rules, if we apply conventional multidimensional association rule finding method without doing any action. Therefore, we want to find multidimensional association rules more effectively for such case.

In section 2, we provide the related work to our research, and in sections 3 we present some detailed explanation of the principles of suggested method, and in section 4 we present our method of experiment. Experiments were run to see the effect of the method in section 5. Finally section 6 provides some conclusions.

II. Related Work

As a method of data mining tasks, association rule discovery systems [6, 7, 8] were developed to find association rules that indicate how often a set of items occur together in transaction databases. These rules give information on association patterns that exist in the database. Many good algorithms were suggested to find association rules efficiently. For example, a standard association algorithm, Apriori, large main memory-based algorithm like AprioriTid [6], the hash-table based algorithm,

This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2010.

DHP [9], random sample based algorithms [10], tree structure-based algorithm [11] or even a parallel version of the algorithm [12]. Other related work is multidimensional association rules. Multidimensional association rules are basically an application of general association rule algorithms to table-like databases. The table-like databases may consist of condition attributes and decision attributes. In papers like [13, 14], multidimensional association rules have better accuracy than decision trees for most of example data sets in small size. However, these algorithms generate, potentially, a large number of rules and manifest facts, so rule selection based on interest measure [15, 16] and generalization is important [17, 18]. Hybrid association rules are a generalization technique to discover more interesting association rules. Hybrid-dimension association rules were found by treating attributes as main and subordinate [19]. In order to mine the database having index structures hybrid dimensional association rules are suggested [20]. There was also effort to reduce computing time. MPNAR algorithm [21] divides the datasets into infrequent itemsets and frequent itemsets and discovers multidimensional association rules for infrequent itemsets as well as frequent itemsets. The multidimensional association rules mined from infrequent itemsets are called negative association rules, and the multidimensional association rules mined from frequent itemsets are called positive association rules. So, the algorithm neglects itemsets between being frequent and not being frequent, and negative association rules may not be useful due to the infrequency. Hu and Li [22] tried to find so called weak support association rules efficiently, and they showed that their algorithm is better than conventional association rule finding algorithms like Apriori. In [23] the authors adapted matrix to find association rules for large transactional databases and showed better performance than other conventional association rule finding algorithms. Incremental association rule mining was suggested by Hong and Hwang [24]. They suggested an incremental association rule finding algorithm that can treat the addition and deletion of records. If data are collected in some sequence of observations over intervals of time, the data are called stream sequence data. Because some events can be related over time, stream data are good application area for association rule mining algorithms. In [25] the authors suggested an efficient association rule mining algorithm for the task.

Rough set theory considers dependency between attribute values solely based on data, so it is a god theoretical basis to find minimum-sized rules. Researchers tried to investigate attribute dependency in algebraic aspects [26], or in statistical aspects [27]. ROSETTA [28] and RSES [29] are some examples of data mining tools to find decision rules based on rough set theory. Because the target database for data mining system based on rough set theory and multidimensional association rules is similar, there are some research results that consider both techniques. Yang et al. [30] pre-processed a transaction database based on constraints of profit and frequency to make a decision table which can be supplied to rough set theory-based data mining method.

III. PRINCIPLES OF SUGGESTED METHOD

We can apply any existing association rule finding algorithms with slight modification for our method; for example, we can use Apriori algorithm or AprioriTid algorithm that are well described in literature [31, 32], or FP algorithm for more efficiency [11]. The major difference is that we apply our method to a table structured data set. The table has two distinct kinds of attributes, several condition attributes and a decision attribute. Each cell of the table consists of attribute-value pair.

Unlikely conventional multidimensional association rule discovery algorithms in which there are no predefined distinction between the roles of attributes, we want to give attention to the table that has a decision attribute and many conditional attributes, and each conditional attribute is independent on each other, and dependent only on the decision attribute. Moreover, we want to find multidimensional association rules for a specific class having specific feature vectors of instances, so that we want to find association rules from all instances that have at least one common attribute value with the instances in a class of the decision attribute, we can narrow down the target instances from the target data set.

In order to understand the reason why we need only some part of the original data set for our data mining task, let's assume we have the following data set of instances like in table 1.

Table 1. An example data set

Id.	A	B	C	Class
1	1	2	3	1
2	1	4	5	1
3	1	2	8	2
4	5	6	8	2
5	1	4	5	2
6	9	10	12	3

In table 1 'Id.' means the identifier of each instance, and attributes, A, B, and C are condition attributes, and attribute Class is decision attribute. The decision attribute Class can have class values like 1, 2, or 3.

Let's assume we want to find association rules for class 1 only. As we see in table 1, the two instances of class 1 have attribute value {1} for attribute A, attribute values {2, 4} for attribute B, and attribute values {3, 5} for attribute C. Because instance 4 and instance 6 have no common values with instance 1 and instance 2, they can be eliminated for our multidimensional association rule mining method.

After eliminating the two instances the data set can be reduced like in table 2, so we have smaller data set for data mining.

Table 2. Reduced data set

Id.	A	B	C	Class
1	1	2	3	1

2	1	4	5	1
3	1	2	8	2
4	eliminated			
5	1	4	5	2
6	eliminated			

If we apply our multidimensional association rule finding algorithm, we can find association rules like the following. In order to find rule set, the given confidence limit is 0.5, and minimum support number is 1. In the rules CF means the confidence of rule, and ‘freq. x/y’ means the frequency of condition itemsets is x, and the frequency of condition and decision itemsets is y.

If C=3 Then class=1 (CF: 1, freq.=1/1);
If C=8 Then class=2 (CF: 1, freq.=1/1);
If A=1 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and C=8 Then class=2 (CF: 1, freq.=1/1);
If B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=1 Then class=1 (CF: 0.5, freq.=2/4);
If A=1 Then class=2 (CF: 0.5, freq.=2/4);
If B=2 Then class=1 (CF: 0.5, freq.=1/2);
If B=2 Then class=2 (CF: 0.5, freq.=1/2);
If B=4 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 Then class=2 (CF: 0.5, freq.=1/2);
If C=5 Then class=1 (CF: 0.5, freq.=1/2);
If C=5 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=2 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=2 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=4 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and C=5 Then class=2 (CF: 0.5, freq.=1/2);
If B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 and C=5 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 and C=5 Then class=2 (CF: 0.5, freq.=1/2);

Because we are interested in association rules of class 1 only, we have the following rule set of our interest.

If C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and C=3 Then class=1 (CF: 1, freq.=1/1);
If B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 Then class=1 (CF: 0.5, freq.=2/4);
If B=2 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 Then class=1 (CF: 0.5, freq.=1/2);
If C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=2 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);

Note that we can get more association rules, and, of course, we need more computing time, when we apply the algorithm to the original data set. The following is found rule set. So, we have 13 (= 39 – 26) more rules. Note that if we apply our method to real world data sets, the saving will be far more, since the size of real world data sets is usually very large.

If C=8 Then class=2 (CF: 1, freq.=2/2);
If A=5 Then class=2 (CF: 1, freq.=1/1);
If A=9 Then class=3 (CF: 1, freq.=1/1);
If B=6 Then class=2 (CF: 1, freq.=1/1);
If B=10 Then class=3 (CF: 1, freq.=1/1);
If C=3 Then class=1 (CF: 1, freq.=1/1);
If C=12 Then class=3 (CF: 1, freq.=1/1);
If A=1 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=5 and B=6 Then class=2 (CF: 1, freq.=1/1);
If A=5 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=9 and B=10 Then class=3 (CF: 1, freq.=1/1);
If A=9 and C=12 Then class=3 (CF: 1, freq.=1/1);
If B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If B=6 and C=8 Then class=2 (CF: 1, freq.=1/1);
If B=10 and C=12 Then class=3 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=5 and B=6 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=9 and B=10 and C=12 Then class=3 (CF: 1, freq.=1/1);
If A=1 Then class=1 (CF: 0.5, freq.=2/4);
If A=1 Then class=2 (CF: 0.5, freq.=2/4);
If B=2 Then class=1 (CF: 0.5, freq.=1/2);
If B=2 Then class=2 (CF: 0.5, freq.=1/2);
If B=4 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 Then class=2 (CF: 0.5, freq.=1/2);
If C=5 Then class=1 (CF: 0.5, freq.=1/2);
If C=5 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=2 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=2 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=4 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and C=5 Then class=2 (CF: 0.5, freq.=1/2);
If B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If B=4 and C=5 Then class=2 (CF: 0.5, freq.=1/2);
If A=1 and B=4 and C=5 Then class=1 (CF: 0.5, freq.=1/2);
If A=1 and B=4 and C=5 Then class=2 (CF: 0.5, freq.=1/2);

The correct confidence of the found rules with our method can be adjusted with the whole data. One more notable fact in calculating correct confidence is that unused attributes in the final rule set can be eliminated from the target data set, so that computing time can be saved more. Let’s see how we can reduce useless attributes by considering the following example data in table 3.

Table 3. An example data set

Id.	A	B	C	Class
1	1	2	3	1
2	5	2	5	1
3	4	3	2	1
4	6	2	8	2
5	7	6	8	2
6	8	4	5	2
7	9	10	12	3

As we can see in table 3, the instances of class 1 have the following values in each attribute.

A = {1, 4, 5},
B = {2, 3},
C = {2, 3, 5}.

So, because instance 5 and instance 7 have no common values with instance 1, instance 2, and instance 3 in each attribute, they can be eliminated for our multidimensional association rule mining. But, instance 4 is not eliminated because of B=2, and instance 6 is not eliminated because of B=5.

After eliminating the two instances the data set can be reduced like in table 4, so we have smaller table for data mining.

Table 4. Reduced data set

Id.	A	B	C	Class
1	1	2	3	1
2	5	2	5	1
3	4	3	2	1
4	6	2	8	2
5	eliminated			
6	8	4	5	2
7	eliminated			

If we apply multidimensional association rule finding algorithm, we can find association rules like the following:

If A=1 Then class=1 (CF: 1, freq.=1/1);
If A=4 Then class=1 (CF: 1, freq.=1/1);
If A=5 Then class=1 (CF: 1, freq.=1/1);
If A=6 Then class=2 (CF: 1, freq.=1/1);
If A=8 Then class=2 (CF: 1, freq.=1/1);
If B=3 Then class=1 (CF: 1, freq.=1/1);
If B=4 Then class=2 (CF: 1, freq.=1/1);
If C=2 Then class=1 (CF: 1, freq.=1/1);
If C=3 Then class=1 (CF: 1, freq.=1/1);
If C=8 Then class=2 (CF: 1, freq.=1/1);
If A=1 and B=2 Then class=1 (CF: 1, freq.=1/1);
If A=1 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=4 and B=3 Then class=1 (CF: 1, freq.=1/1);
If A=4 and C=2 Then class=1 (CF: 1, freq.=1/1);
If A=5 and B=2 Then class=1 (CF: 1, freq.=1/1);
If A=5 and C=5 Then class=1 (CF: 1, freq.=1/1);
If A=6 and B=2 Then class=2 (CF: 1, freq.=1/1);
If A=6 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=8 and B=4 Then class=2 (CF: 1, freq.=1/1);
If A=8 and C=5 Then class=2 (CF: 1, freq.=1/1);
If B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If B=2 and C=5 Then class=1 (CF: 1, freq.=1/1);
If B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If B=3 and C=2 Then class=1 (CF: 1, freq.=1/1);
If B=4 and C=5 Then class=2 (CF: 1, freq.=1/1);
If A=1 and B=2 and C=3 Then class=1 (CF: 1, freq.=1/1);
If A=4 and B=3 and C=2 Then class=1 (CF: 1, freq.=1/1);
If A=5 and B=2 and C=5 Then class=1 (CF: 1, freq.=1/1);
If A=6 and B=2 and C=8 Then class=2 (CF: 1, freq.=1/1);
If A=8 and B=4 and C=5 Then class=2 (CF: 1, freq.=1/1);
If B=2 Then class=1 (CF: 0.67, freq.=2/3);
If C=5 Then class=1 (CF: 0.5, freq.=1/2);
If C=5 Then class=2 (CF: 0.5, freq.=1/2);

If we assume that we want to find association rules that have minimum support number of at least two, we can have rules like the followings:

If B=2 Then class=1 (CF: 0.67, freq.=2/3);
If C=5 Then class=1 (CF: 0.5, freq.=1/2);

Because we do not have attribute A in the rule set, we do not need attribute A to calculate actual confidence from the original data set. Table 5 shows the resulting table.

Table 5. An example data set

Id.	B	C	Class
1	2	3	1
2	2	5	1
3	3	2	1
4	2	8	2
5	6	8	2
6	4	5	2
7	10	12	3

As we can see in table 5, we can expect that we can have less computing time compared to the computing time needed to calculate the confidence from the original data. We may have more reduction in computing time as we have less number of attributes in the found association rules from the reduced data set like the one from table 4.

IV. THE METHOD

The following is a formal definition of association rules we are interested. Let I = $\{i_1, i_2, ..., i_m\}$ be a set of items that are in a table, where i_j is an attribute-value pair. Let T be a collection of records, where each record has a set of itemset $X \subseteq I$.

A multidimensional association rule is an implication of the form $Y \rightarrow Z$ where $Y \subset I$, $Z \subset I$, and $Y \cap Z \neq \phi$. The confidence C% of the association rule $Y \rightarrow Z$ implies that among all the records which contain itemset Y, C% of them also contains itemset Z. For an itemset $X \subset I$, support(X) = s is the fraction of the records in T containing X. So, the confidence of a rule $Y \rightarrow$

Z is computed as the ratio, support(Y ∩ Z)/support(Y). Note that Y ∩ Z means itemsets in records. If we consider actual count of itemsets, *support number* can used to represent how many times an itemset occurs in T. The itemsets that occur more frequently than some given support level are called *frequent itemsets*.

We are interested in a more efficient way in discovering multidimensional association rules for specific classes like minor classes. In other words, we want to find multidimensional association rules for a specific class having specific feature vectors. So, because we want to find association rules from all instances that have at least one common attribute value with the instances in the class, we can reduce the number of the target instances from the original data set.

Conventional multidimensional association rule finding algorithms also can find those rules, but all the other multidimensional association rules are found also so that much computing time is needed, and, moreover, possibly a lot of rules are generated, because the size of target data set for data mining is larger than the one with our method.

Because we are interested in finding multidimensional association rules for a specific class, we can limit our search to the records that have itemset Y only, if we want to find multidimensional association rule like Y → Z. So, we can reduce the size of target data set a lot. The following is a brief description of the procedure of the method.

INPUT:
T: a decisional table for data mining,
CF: minimum confidence of rules,
MS: minimum support number,
S: the table of records of items in interesting class.
OUTPUT: a set of multidimensional association rules.
Begin
D := Select all records from T that have any one item in each record of S **;**
Apply multidimensional association rule finding algorithm to D with MS;
Generate multidimensional association rules having greater confidence than or equal to CF ;
Select multidimensional association rules of interesting class;
Calculate the confidence of rules based on T;
Eliminate such rules that have confidence < CF;
If a rule's condition part is shorter and the confidence of the rule is greater than or equal to some other rule
Then
select the rule of shorter condition part;
End If
End;

In the algorithm each record of S consists of items in each record that belongs to the interesting class. We select records that have any identical item in the records of interesting class from the target database T. By doing so, the target database size for data mining can be reduced. As a result, we may have smaller number of multidimensional association rules and save computing time. By eliminating any such rules that have smaller confidence than given CF after calculating the confidence of rules based on T, we can have more accurate confidence of each rule. In the following experiment the given CF value is 65% and minimum support number is two, because the interesting class consists of relatively small number of instances.

We can eliminate such rules that have inferior confidence for the same decision but the condition parts of the rules have additional testing of attribute values. In order to understand the concept let's look at the following example rules.

If A=a Then Class=C1 (CF: 0.7, freq.=7/10) (1)
If A=a and B=b Then Class=C1 (CF: 0.6, freq.=6/10) (2);

Because rule (1) needs shorter condition and confidence is higher, we can eliminate rule (2) in the final rule set. If a multidimensional association rule has larger condition part with the same decision and better frequency value, but the confidence is the same, we may not eliminate the rule.

V. EXPERIMENTATION

Experiments were run using a database in UCI machine learning repository [33] called 'Statlog(shuttle)' [34] to see the effectiveness of the method. The number of instances in the data set for training is 43,500. The data sets were selected, because it is relatively large and consists of many classes. The total number of attributes is ten, named A to J, and there are seven classes. All attributes are continuous attributes for stalog data set. Class distribution is in table 1.

Table 2. Class distribution

Class	Percentage
1	78.41%
2	0.09%
3	0.3%
4	15.51%
5	5.65%
6	0.01%
7	0.03%

We are interested in multidimensional association rules for class 6. Before applying association rule finding algorithm, decritization based on entropy minimization heuristic [35] is performed. So, the records that have any identical item with the records of class 6 are selected. The total number of the selected records is 4,902. So the size of data set for multidimensional association rule finding algorithm is reduced to 11.27%.

A modified Ariori algorithm is was used to generate multidimensional association rules. Note that we can find frequent itemsets more efficiently than the original algorithm, because an item consists of attribute-values pair and there is no duplicate attribute in an itemset.

After applying multidimensional association rule finding algorithm to find rules of minimum support number 2, 23 association rules were found. The next step was to calculate correct confidence from the whole data, and there was no change in confidence and support of the above rules. Among them there are a lot of redundant and useless rules that have longer condition part, but no improvement in confidence. So, we dropped the redundant and useless rules. The following 15 rules are the final result. In attribute value '(' means open bracket, and '[' means closed bracket, and 'freq. x/y' means the frequency of condition itemsets is x, and the frequency of condition and decision itemsets is y.

If A=(54.5~55.5] and B=(586~∞) Then
CLASS=6 (CF: 1, freq.=2/2);
If A=(56.5~68.5] and B=(586~∞) Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and G=(15.5~20.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and H=(43.5~47.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and I=(21~27] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and I=(31~37] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and E=(17~25] and F=(-0.5~0.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and F=(-0.5~0.5] and H=(51.5~57.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and D=(-0.5~0.5] and E=(17~25] and
F=(-0.5~0.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and D=(-0.5~0.5] and F=(-0.5~0.5] and
H=(51.5~57.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If B=(586~∞) and E=(17~25] Then
CLASS=6 (CF: 0.67, freq.=2/3);
If B=(586~∞) and H=(51.5~57.5] Then
CLASS=6 (CF: 0.67, freq.=2/3);
If B=(586~∞) and D=(-0.5~0.5] and E=(17~25] Then
CLASS=6 (CF: 0.67, freq.=2/3);
If B=(586~∞) and D=(-0.5~0.5] and H=(51.5~57.5] Then
CLASS=6 (CF: 0.67, freq.=2/3);
If B=(586~∞) and E=(17~25] and H=(51.5~57.5] Then
CLASS=6 (CF: 0.67, freq.=2/3);

If a rule's condition part is shorter and the confidence of the rule is greater than or equal to some other rules, then the rule with shorter condition part has been selected. For example, there were many redundant rules of the second rule. The followings are some of the rules.

If A=(56.5~68.5] and B=(586~inf) and D=(-0.5~0.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If A=(56.5~68.5] and B=(586~inf) and F=(-0.5~0.5] Then
CLASS=6 (CF: 1, freq.=2/2);
If A=(56.5~68.5] and B=(586~inf) and D=(-0.5~0.5] and
F=(-0.5~0.5] Then CLASS=6 (CF: 1, freq.=2/2);

One more experiment was run to find multidimensional association rules for class 7. So, the records that have any identical item with the records of class 7 are selected. The total number of the selected records is 10,686. So the size is reduced to 24.57%. After applying multidimensional association rule finding algorithm to find rules of minimum support number 3, 233 association rules were found. The next step was to calculate correct confidence from the whole data, and there was no change in confidence and support of the above rules. Among them there are a lot of redundant and useless rules that have longer condition part, but no improvement in confidence. So, we dropped the redundant and useless rules. The following rules are shortest rules of the same confidence with redundant and useless rules.

If B=(-inf~-942.5] and F=(-10.5~-0.5] Then CLASS=7 (CF: 1,
freq.=6/6);
If B=(-inf~-942.5] and H=(71.5~78.5] Then CLASS=7 (CF: 1,
freq.=6/6);
If B=(-inf~-942.5] and I=(3~7] Then CLASS=7 (CF: 1,
freq.=6/6);
If B=(-inf~-942.5] and D=(0.5~inf) Then CLASS=7 (CF: 1,
freq.=4/4);
If B=(-inf~-942.5] and G=(37.5~39.5] Then CLASS=7 (CF: 1,
freq.=4/4);
If B=(-inf~-942.5] and C=(75.5~76.5] Then CLASS=7 (CF: 1,
freq.=3/3);

There were many redundant rules. The followings are some of the redundant rules of the first three rules.

If A=(-inf~37.5] and B=(-inf~-942.5] and F=(-10.5~-0.5]
Then CLASS=7 (CF: 1, freq.=6/6);
If A=(-inf~37.5] and B=(-inf~-942.5] and H=(71.5~78.5]
Then CLASS=7 (CF: 1, freq.=6/6);
If A=(-inf~37.5] and B=(-inf~-942.5] and I=(3~7] Then
CLASS=7 (CF: 1, freq.=6/6);
If A=(-inf~37.5] and B=(-inf~-942.5] and C=(105.5~107.5]
and F=(-10.5~-0.5] Then CLASS=7 (CF: 1, freq.=5/5);
If A=(-inf~37.5] and B=(-inf~-942.5] and C=(105.5~107.5]
and H=(71.5~78.5] Then CLASS=7 (CF: 1, freq.=6/6);
If A=(-inf~37.5] and B=(-inf~-942.5] and C=(105.5~107.5]
and I=(3~7] Then CLASS=7 (CF: 1, freq.=6/6);

VI. CONCLUSIONs

Multidimensional association rules are association rules that can be found from a table-like database. The difference between association rules and multidimensional association rules is that each item in multidimensional association rules has distinct attributes, while there is no such attribute in the items of association rules. Moreover, if the attributes of the table-like database can be divided into two different kinds of attributes, condition and decision attributes, we can find multidimensional association rules for classification task.

Because the target databases of conventional multidimensional association rule algorithms have no distinction in the role of attributes, a lot of rules can be found and the computing time can be enormous. So, we need to limit our search for better result. As a way to solve the problem, we are interested in some efficient way to search for multidimensional association rules for a specific class from the table that has a decision attribute and many conditional attributes.

Moreover, if we find multidimensional association rules for a specific class, it is highly possible that it will become easier to find more interesting rules for the class, because we will have less number of rules. Otherwise we may have many uninteresting rules by the conventional multidimensional association rule finding method.

In order to overcome the problem of intensive computing time and possibility of generating a lot of uninteresting rules, a preprocessing technique that can narrow down the search space is suggested, and the method can generate smaller table for the data mining of multidimensional association rules so that computing time can be saved and also smaller number of rules are generated. Experiments with a real world data set showed a very good result.

REFERENCES

[1] W. He, G. Hu, X. Yao, "Large-scale Communication Network Behavior Analysis and Feature Extraction Using Multiple Motif Pattern Association Rule Mining," *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 5, Volume 8, May 2009, pp. 473-482.

[2] M.N. Moreno, S. Seqrera, V.F. Lopez, M.J. Polo, "A method for mining quantitative association rules," in *Proceedings of the 6th WSEAS International Conference on Simulation, Modeling and Optimization*, Stevens Point, Wisconsin, USA, 2006, pp. 173-178.

[3] A.A.S. Al-Mudimigh, B.F. Saleem, C.Z. Ullah, " Developing an Integrated Data Mining Environment in ERP-CRM Model – A Case Study of MADAR," *INTERNATIONAL JOURNAL OF EDUCATION AND INFORMATION TECHNOLOGIES*, Issue 2, Volume 3, 2009, pp. 135-144.

[4] N. Kerdprasop, K. Kerdprasop, “Recognizing DNA splice sites with the frequent pattern mining technique," *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES*, Issue 1, Volume 5, 2011, pp. 87-94.

[5] R. Chen, Y. Tsai, K.C. Yeh, D.H. Yu, and Y. Bak-Sau, "Using Data Mining to Provide Recommendation Service," *WSEAS TRANSACTION on INFORMATION SCIENCE & APPLICATIONS*, Issue 4, Volume 5, 2008, pp. 459-474.

[6] R. Agrawal, H. Mannila, H., R. Srikant, H. Toivonen, A.I. Verkamo, "Fast Discovery of Association Rules," In *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, R. Uthurusamy ed., AAAI Press/The MIT Press, 1996, pp. 307-328.

[7] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 3, 2000, pp. 372-390.

[8] J. Han, J. Pei, Y. Yin, R. Mao, "Mining frequent patterns without candidate generation," Data Mining and Knowledge Discovery, Vol. 8, 2004, pp. 53-87.

[9] J.S. Park, M. Chen, P.S. Yu, "Using a Hash-Based Method with Transaction Trimming for Mining Association Rules," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 5, 1997, pp. 813-825.

[10] H. Toivonen, *Discovery of Frequent Patterns in Large Data Collections*, phD thesis, Department of Computer Science, University of Helsinki, Finland, 1996.

[11] J. Han, J. Pei, Y. Yin, R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, Vol. 8, 2004, pp. 53-87.

[12] A. Savasere, E. Omiecinski, S. Navathe, *An Efficient Algorithm for Mining Association Rules in Large Databases*, College of Computing, Georgia Institute of Technology, Technical Report No.: GIT-CC-95-04.

[13] W. Li, J. Han, J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," in *Proceedings 2001 Int. Conf. on Data Mining (ICDM'01)*, 2001, pp. 369-376.

[14] B. Liu, W. Hsu, Y. Ma, "Integrating Classification and Association Rule Mining," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, 1998, pp. 80-86.

[15] H. Toivonen, M. Klemettinen, H. Mannila, P. Rokainen, K. Hatonen, "Pruning and Grouping of Discovered Association Rules," In *Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, 1995, pp. 47-52.

[16] M. Dimitrijević, Zita Bošnjak, "Discovering Interesting Association Rules in the Web Log Usage Data," *Interdisciplinary Journal of Information, Knowledge, and Management*, Vol. 5, 2010, pp. 191-207.

[17] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, A.I. Verkamo, "Finding Interesting Rules from Large Set of Discovered Association Rules," In *Proceedings of the Third International Conference on Information and Knowledge Management(CIKM’94)*, 1994, pp. 401-407.

[18] C. Perng, H. Wang, S. Ma, J. Hellerstein, "Discovery in Multi-attribute Data with User-defined Constraints," *ACM SIGKDD Explorations Newsletter, Volume4, Issue 1*, 2002, pp. 56-64.

[19] Y. Xin, S. Ju, "Mining conditional hybrid-dimension association rules on the basis of multi-dimensional transaction database," in *Proceedings of 2003 International Conference on Machine Learning and Cybernetics,* 2003, pp. 216-221.

[20] R. Chithra, S. Nicklas, "A Novel Algorithm for Mining Hybrid-Dimensional Association Rules," *International Journal of Computer Applications*, Vol. 1, No. 16, 2010, pp. 53-58.

[21] H. Jiang, Z. Bai, G. Liu, X. Luan, "An Algorithm for Mining Multidimensional Positive and Negative Association Rules," *Advanced Materials Research*, Vol. 171, 2010, pp. 445-449.

[22] J. Hu, X. Li, "Association Rules Mining Including Weak-Support Modes Using Novel Measures," *WSEAS TRANSACTIONS on COMPUTERS*, Issue 3, Volume 8, March 2009, pp. 559-568.

[23] W. Sun, C. Wang, T. Zhang, Y. Zhang, "Transaction-item Association Matrix-Based Frequent Pattern Network Mining Algorithm in Large-scale Transaction Database," *WSEAS TRANSACTIONS on COMPUTERS*, Issue 8, Volume 8, August 2009, pp. 1327-1336.

[24] T. Hong, T. Huang, "Maintenance of Generalized Association Rules for Record Deletion Based on the Pre-Large Concept," in *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, Greece, February 16-19, 2007, pp. 142-146.

[25] H.A. Aboalsamh, A.M. Hafez, G.M.R. Assassa, "An Efficient Stream Mining Technique," *WSEAS Transactions on Information Science and Applications*, issue 7, Volume 5, July 2008, pp. 1272-1281

[26] I. Düntsch and G. Gediga, "Algebraic aspects of attribute dependencies in information systems," *Fundamenta Informaticae*, Vol. 29, 1997, pp. 119-133.

[27] I. Düntsch and G. Gediga, "Statistical evaluation of rough set dependency analysis," *International Journal of Human-computer Studies,* Vol. 46, 1997, pp. 589-604.

[28] A. Øhrn, *Discernibility and rough sets in medicine: tools and applications*, PhD thesis, Department of computer and information science, Norwegian University of Science and Technology, 1999.

[29] J.G. Bazan, M.S. Szczuka, J. Wroblewski, "A new version of rough set exploration system," *Lecture notes in artificial intelligence,* Vol. 2475, 2002, pp. 397-404.

[30] W. Yang, Y. Li, Y. Xu, H. Liu, "Rough Set Model for Constraint-based Multi-dimensional Association Rule Mining," in *Proceeding of the 2006*

conference on Advances in Intelligent IT: Active Media Technology 2006, pp. 99-105.

[31] P. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2006

[32] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2011.

[33] A. Suncion, D.J. Newman, *UCI Machine Learning Repository* [http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html]. Irvine, CA: University of California, School of Information and Computer Sciences, 2007.

[34] C. Feng, A. Sutherland, S. King, S. Muggleton, R. Henery, "Comparison of Machine Learning Classifiers to Statistics and Neural Networks," in *Proceedings of the Third International Workshop in Artificial Intelligence and Statistics*, 1993, pp. 41-52.

[35] U. M. Fayyad, K. B. Irani, "Multi-interval discretization of continuous valued attributes for classification learning," in *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence*, 1993, pp. 1022-1027.

Hyontai Sug received the B.S. degree in Computer Science and Statistics from Busan National University, Busan, Korea, in 1983, the M.S. degree in Computer Science from Hankuk University of Foreign Studies, Seoul, Korea, in 1986, and the Ph.D. degree in Computer and Information Science & Engineering from University of Florida, Gainesville, FL, in 1998. He is an associate professor of the Division of Computer and Information Engineering of Dongseo University, Busan, Korea from 2001. From 1999 to 2001, he was a full time lecturer of Pusan University of Foreign Studies, Busan, Korea. He was a researcher of Agency for Defense Development, Korea from 1986 to 1992. His areas of research include data mining and database applications.