DISCOVERY OF MULTI DIMENSIONAL ASSOCIATION RULES FROM LARGE INCONSISTENT DATABASES

Sarjon Defit

Faculty of Computer Science
University Putra Indonesia “YPTK” Padang, West Sumatera
Telp: 62-751-775246, Fax: 62-751-71913, E-mail: sarjond@yahoo.co.uk

Abstract

Association rules is one of data mining method for discovering knowledge from large amounts of data in databases. In this paper, we propose an intelligent method for discovering multi dimensional association rules from large inconsistent databases, called IMAR. IMAR is designed through three main phases, i.e., preprocessing, processing and post processing. It has been experimented using three domain data sets, i.e., Australian Credit Card (ACC), Jakarta Stock Exchange (JSX), and Cleveland Heart Diseases (CLEV) data sets. Our experimental results show that IMAR can (i) discover multi dimensional association rules from large inconsistent databases intelligently and accurately, and (ii) reduce the number of generated interesting association rules without loss information and with higher accuracy.

1. INTRODUCTION

Association rule is one of data mining method for discovering knowledge from large amount of data in databases. It is a rule in the form of \([1, 2, 3, 4, 5] \rightarrow [S,C]\) (1.1)

where \(X_1, X_2, \ldots, X_n \rightarrow Y_1, Y_2, \ldots, Y_m\) are items. The S and C are support and confidence of rules respectively. It has becomes more and more popular since its introduction in 1993. Today, it is still one of the most popular methods in data mining [1, 2, 3, 4, 5, 6, 7].

A number of promising association rules methods have been studied and developed. For instance (i) a close algorithm [7], (ii) a closet algorithm [8], (iii) online analytical mining association rules [4], (iv) mining association rules with multiple Minimum Item Support [6], and (v) discovery of knowledge at multiple level concepts [9]. These methods have given great advantages for user in order to generate rules from large amounts of data. However, association rules methods still have some the following weaknesses and need further improvement. First, accuracy of association rules method [4, 6, 7, 8, 9]. The association rules method should portray the contents of the database accurately. The noise and uncertainty should be cleaned elegantly. It is one of the important tasks in data preparation in order to identify which data in databases are inaccurate or missing values. Second, usefulness of association rules method [4, 6, 7, 8]. The association rules method should be useful for certain application and generate association rules from data which are represented at higher levels concept. The raw data should be transformed from raw data into higher levels concept. It allows users to find association rules deeply and view database contents at different abstraction levels. Third, identification of interesting rules [6, 7, 8, 9]. The association rules method should generate more interesting rules accurately. The generated rules should be identified in order to reduce the number of interesting rules without loss information. Fourth, using of prior and domain knowledge [4, 6, 7, 8, 9]. The association rules method should generate interesting rules intelligently. It allows us to generate association rules from large amount of data in databases intelligently and automatically. Lastly, intelligent hybrid association rules method [4, 6, 7, 8, 9]. The association rules method should generate knowledge from various domains and solve the complex data mining problems. It can generate more interesting rules and reduce the number of rules without loss information.

In order to cater these association rules problems, in this paper, we propose an Intelligent Mining Association Rules, called IMAR.

The rest of this paper is organized as follows. The IMAR general architecture is given in section 2. The data transformation including basic data transformation and extended data transformation
algorithms is given in section 3. The IMAR experimental results and summary of this paper are given in section 4 and 5 respectively.

2. INTELLIGENT MINING ASSOCIATION RULES (IMAR) GENERAL ARCHITECTURE

In this section, we describe the Intelligent Mining Association Rules (IMAR) general architecture as given in appendix. Generally, Intelligent Mining Association Rules (IMAR) consists of three main phases including preprocessing, processing and post processing. The first phase have two processes including data cleaning and data transformation while the main process of processing is rule generation. The first two phases, i.e., preprocessing and processing, have two main steps including training and running steps. The first step is conducted for creating neural network knowledge based of data cleaning, data transformation and association rules. The generation of learned complete data, transformed data and interesting association rules are done in the running step. Next, the generated interesting rules are applied in real world problems in order to create crucial business decision.

Intelligent Mining Association Rules (IMAR) is designed based on combination of several intelligent techniques, i.e, rough set, association rules and neural networks knowledge based. The purpose of combinations these intelligent techniques is to create a method for solving the data mining complex problems, i.e., data cleaning, data transformation and association rules. In order to support IMAR, we also propose data cleaning and data transformation methods. However, this paper is focused on data transformation. The details of proposed data cleaning and association rules methods can be found in [10] and [11].

3. DATA TRANSFORMATION

In this section, we describe the basic and extended data transformation algorithms. The details of these data transformation algorithms are given in the following sub sections 3.1 and 3.2.

3.1 BASIC DATA TRANSFORMATION ALGORITHM

This algorithm is unsupervised data transformation. Number of classes is derived based on number of observed data. Each attributes in databases has the same numbers of classes and each class has the same interval. The basic concept of this algorithm is given in definition 3.1.

Definition 3.1: Let \( \beta_v \) and \( \gamma_v \) be largest and smallest values. The total value range, \( \alpha_v \), is defined as \( \alpha_v = \beta_v - \gamma_v \). Let \( \eta_c \) be the number of classes, defined as \( \eta_c = 1 + 3.3 \log(n) \) where \( n \) equal to number of observations. The Interval, denoted as \( \chi_v \), is defined as \( \chi_v = \frac{\alpha_v}{\eta_c} \), and the cut point, denoted as \( \phi_v \), defined as \( \theta_v = \gamma_v + \chi_v(i) \text{ for } i = 1, 2, 3, \ldots, \eta_c - 1 \).

Based on definition 3.1, the basic data transformation algorithm is given in figure 3.1.

Input : Complete Databases
Output : Transformed Data
Method:
1) Estimation of the total value range.
2) Estimation of the number of classes
3) Derivation of interval value
4) Compute the cut point
5) Code the observed values based on set of cuts, i.e., 0, 1, 2, \ldots, n-1, where n is number of class
6) Generate indiscernibility relation based conditional attribute
7) For each indiscernibility relation, check inconsistent data
8) If inconsistent data is found, replace decision value by modus decision value
9) This process is proceeded until all data are consistent

Figure 3.1: The Basic Data Transformation Algorithm

Figure 3.1 shows a basic data transformation algorithm. It is used for generating a transformed data from complete data in training step and target output of transformed data in running step. The input and output of this algorithm are complete and transformed data respectively.

3.2 EXTENDED DATA TRANSFORMATION ALGORITHM

In this section, we give an extended data transformation algorithm. It is used for generating
learned transformed data which is supported by neural network knowledge based. In this algorithm, the creation of neural network knowledge based of data transformation and target output of transformed complete testing data are supported by basic data transformation algorithm. The illustration of extended data transformation algorithm is given in the following figure 3.2.

**Step 1: Training**

**Input**: The complete training data which is generated at training step of data cleaning  
**Output**: Knowledge of data transformation  
**Methods**:  
1) Transform complete training data, A, using basic data transformation algorithm, and then saved as transformed training data, B.  
2) Merge transformed training data, B, and complete training data, A, into merged data, C.  
3) Split the merged data, C, into training and testing data sets, D and E, respectively.  
4) Train these data sets, D and E, using neural network.  
5) Merge the condition part of merged data and trained data in order to create knowledge of data transformation, N.

**Step 2: Running**

**Input**: Learned complete testing data which is generated at running step of data cleaning  
**Output**: Learned transformed testing data  
**Methods**:  
1) Transform learned complete testing data, I, using basic data transformation algorithm and then saved as transformed testing data, O.  
2) Merge the learned complete testing data and transformed testing data, and I and O, into merged data, P.  
3) Suppose the merged data, P, and knowledge of data transformation which is generated at training step, N, as testing and training data sets.  
4) Learn training and testing data sets, P and N, using neural network.  
5) Merge the condition part of merged data with the learned data in order to create new knowledge of data transformation, Q.  
6) Compare the merged data with the new knowledge of data transformation. When the condition parts are match, then replace the action part of merged data with the new knowledge of data transformation. The results is saved as new data transformation, R.  
7) Add new knowledge into previous knowledge of data transformation.

**Figure 3.2: The Extended Data Transformation Algorithm**

Figure 3.2 shows the extended data transformation algorithm. It consists of two main steps including training and running steps. The training step is conducted for generation neural network knowledge based of data transformation while the running step is used for generating learned transformed testing data. The input and output of the training step are complete training data which is obtained in the first step of data cleaning and neural network knowledge based of data transformation. Compared to the first step, the input of running step is the learned complete testing data which is obtained in the second step of data cleaning while the output is learned transformed testing data.

4 IMAR EXPERIMENTAL RESULTS

We have tested IMAR using three differences domain data sets. It includes Australian Credit Card (ACC), Jakarta Stock Exchange (JSX) and Cleveland Heart Diseases (CLEV). In the following, we describe the IMAR performances in both preprocessing and processing phases.

4.1 PREPROCESSING OF IMAR EXPERIMENTAL RESULTS

In this sub section, we discuss the preprocessing of IMAR performances. It includes IMAR data cleaning and data transformation methods.

4.1.1 IMAR DATA CLEANING EXPERIMENTAL RESULTS

We have studied and analyzed the IMAR data cleaning experimental results. In this research, the IMAR data cleaning performance is measured based on three differences error measurements, called Mean Difference Error (MDE), Mean Relative Error (MRE) and MSE (Mean Square Error) which are defined as follows

$$MDE = \frac{1}{n} \sum_{i=1}^{n} (v_{ai} - v_{pi})$$  

(4.1)

$$MRE = \frac{1}{n} \sum_{i=1}^{n} \frac{(v_{ai} - v_{pi})}{v_{pi}}$$  

(4.2)

$$MSE = \left[ \frac{1}{n} \sum_{i=1}^{n} (v_{ai} - v_{pi})^2 \right]^{1/2}$$  

(4.3)

where  

- $v_{ai}$ and $v_{pi}$ = the observed and predicted values of output respectively  
- $n$ = the number of observation
The performance of IMAR data cleaning compared with mean substitution data cleaning method is given in the following table 4.1 and figure 9.1.

**Table 4.1: The IMAR Data Cleaning Performance Compared With Mean Substitution Data Cleaning Method**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>IMAR Data Cleaning</th>
<th>Mean Substitution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MDE</td>
<td>MRE</td>
</tr>
<tr>
<td>ACC</td>
<td>0.004</td>
<td>0.0013</td>
</tr>
<tr>
<td>JSX</td>
<td>0.260</td>
<td>0.0012</td>
</tr>
</tbody>
</table>

**Mean Substitution**

<table>
<thead>
<tr>
<th></th>
<th>MDE</th>
<th>MRE</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0127</td>
<td>0.0013</td>
<td>0.4033</td>
</tr>
<tr>
<td></td>
<td>0.108</td>
<td>0.0032</td>
<td>10.987</td>
</tr>
</tbody>
</table>

**The Performance of Data Cleaning Method**

Figure 4.1 shows the performance of IMAR data cleaning compared with mean substitution data cleaning method. For example, JSX data set, the Mean Difference error (MDE) of IMAR data cleaning and mean substitution are equal to 0.2600 and 0.1080 respectively. While the Mean Relative Error (MRE) are equal to 0.0012 and 0.0032 respectively. The Mean Square Error (MSE) of these methods is equal to 7.000 and 10.987 respectively. Based on these experimental results, we concluded that these methods give a better results for handling uncertainty and missing data. However, IMAR data cleaning offers several advantages as given in table 4.2.

Table 4.2 shows the IMAR data cleaning advantages and disadvantages. This method can (i) handle the missing and uncertainty data in databases, (ii) it can fill the missing values with the accurate values, and (iii) it can decrease the error measurements including mean differences error (MDE), Mean Relative Error (MRE) and Mean Square Error (MSE). Although IMAR data cleaning have great advantages, this method is still limited for handling the missing values form numerical data. It is still needed further improvement for handling uncertainty and missing data from another data type, i.e., categorical and text data.

**4.1.2 IMAR DATA TRANSFORMATION EXPERIMENTAL RESULTS**

In this research, the performance of IMAR data transformation is evaluated based number of generated classes of transformed data. The performance of IMAR data transformation compared with two previous data transformation, i.e., yongjian Fu’s and equal binning frequency data transformation methods is given in table 9.3 and figure 9.2.

**Table 9.3: The IMAR Data Transformation Performance Compared with YongjianFu’s and Equal Binning Frequency Data Transformation Methods**

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>The Generated Number of Classes (Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y-DT</td>
</tr>
<tr>
<td>JSX</td>
<td>14</td>
</tr>
<tr>
<td>ACC</td>
<td>14</td>
</tr>
<tr>
<td>CLEV</td>
<td>14</td>
</tr>
</tbody>
</table>
Table 9.3 and figure 9.2 show the IMAR data transformation performance compared with Yongjian Fu’s and equal binning frequency data transformation methods. We show that, difference methods generate differences number of classes. For instance, Yongjian Fu’s data transformation method, generate number of classes depend on number of classes threshold, i.e., 2, 3, ..., n. While equal binning frequency generate each attributes in databases into static number of classes. In other words, all attributes in databases could have the same number of classes. For instance, attributes in JSX, ACC and CLEV data sets are transformed into 3 (three) classes. Compared to Yongjian Fu’s and equal binning frequency data transformation, the IMAR data transformation could transform raw data in databases into difference number of classes. For instance, “Price” and CapVal” attributes in JSX data set are transformed into 9 and 3 classes respectively.

Based on these experimental results, we conclude that IMAR data transformation method offers several advantages as given in table 9.4.

### Table 9.4: The Advantages and Disadvantages of IMAR Data Transformation

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) It can transform raw data into transformed data accurately and intelligently.</td>
<td>i) It is limited for transforming raw data from numerical data.</td>
</tr>
<tr>
<td>ii) It can transform raw data into various number of classes of transformed data</td>
<td>ii) It cannot transform raw data into more higher levels.</td>
</tr>
</tbody>
</table>

In this research, the processing of Intelligent Mining Association Rules (IMAR) performance is measured based on number and accuracy of generated interesting rules as given in table 9.5, figure 9.3a and 9.3b.

### Table 9.5: The Processing of IMAR Performance Compared With Association Rules Without Preprocessing and Basic Association Rules Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>ACC</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NGR</td>
<td>NIR</td>
</tr>
<tr>
<td>AR-Without Pre Processing</td>
<td>1181</td>
<td>1004</td>
</tr>
<tr>
<td>BRG</td>
<td>1240</td>
<td>1236</td>
</tr>
<tr>
<td>IMAR-AR</td>
<td>202</td>
<td>189</td>
</tr>
</tbody>
</table>
Table 9.5, figure 9.3a and 9.3 b show the performance of IMAR association rules compared with association rules without pre processing, basic rule generation association rules methods. For example, generated rules form JSX data set, using association rules without pre processing, basic rule generation and IMAR association rules are equal to 1181, 1240 and 202 while the number of interesting rules are equal to 1004, 1236, and 189 rules respectively. The accuracy of generated interesting rules are equal to 90.3, 99.5 and 91.4 %. Another example, generated rules from ACC data set, the number of generated rules using association rules without pre processing step, basic rule generation and IMAR association rules are equal to 7120, 1879 and 232 respectively while the generated interesting rules are equal to 4941, 1758 and 231 respectively. The accuracy of generated interesting rules are equal to 78.71, 96.7 and 99.6 respectively.

Based on these experimental results, we show that IMAR offers several advantages as given in table 9.6.
Table 9.6: The Advantages and Disadvantages of IMAR Method

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) it can mine multi dimensional association rules from large databases accurately and intelligently</td>
<td>i) It cannot mine more complex association rules</td>
</tr>
<tr>
<td>ii) It can reduce the number of generated interesting rules without loss information</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.6 shows the advantages and disadvantages of IMAR method. This method can (i) mine multi dimensional association rules from large inconsistent data intelligently and accurately, and (ii) it can reduce the number of generated interesting rules without loss information. Although IMAR association rules could reduce the number of generated interesting rules with higher accuracy, this method should be extended for generating association rules from another types of rules since it is limited for generating rules from first predicates calculus form.

5 Summary

We have proposed an intelligent method for discovering multi dimensional association rules from large inconsistent databases, called IMAR. IMAR is designed through three main phases, i.e., preprocessing, processing and post processing. IMAR has been experimented using three domain data sets, i.e., Australian Credit Card (ACC), Jakarta Stock Exchange (JSX), and Cleveland Heart Diseases (CLEV) data sets. Our experimental results show that IMAR can (i) discover multi dimensional association rules from large inconsistent databases intelligently and accurately, and (ii) reduce the number of generated interesting association rules without loss information and with higher accuracy.

References:

Appendix: The IMAR General Architecture

1. **Database**
   - **Training Data**
     - Data Cleaning
       - Complete Training Data
         - Rough Set
           - Data Transformation
             - Transformed Complete Training Data
               - Rule Generation
                 - Generated Rules
                   - Train
                     - Knowledge of Data Cleaning
                       - Neural Network
2. **Testing Data**
   - Data Cleaning
     - Complete Testing Data
       - Learn
         - Learned Complete Testing Data
           - Rough Set
             - Data Transformation
               - Transformed Learned Complete Testing Data
                 - Learn
                   - Learned Interesting Rules
                     - Application
                       - Business Decision
3. **Rule Generation**
   - Generated Rules
     - Learn
       - Learned Interesting Rules
         - Application
           - Business Decision