Mining Association Rule From Large Databases

Sarjon Defit, Mohd Noor Md Sap
Faculty of Computer Science and Information System
University of Technology Malaysia, KB. 791
80990 Johor Bahru, Malaysia
Telp: (07)-5576160, Fax: (07) 5566155
{sarjon_d@hotmail.com},{mohdnoor@fsksm.utm.my}

Abstract

Association rules, introduced by Agrawal, Imielinski and Swami, is one of data mining technique to discover interesting rules or relationships among attributes in databases. It has attracted great attention in database research communities in recent years. In this paper, we propose a Mining Association Rules (MAR) model which integrate intelligent and data analysis techniques. MAR model has been implemented and tested using Jakarta Stock Exchange (JSX) databases. Our study conclude that MAR model can improve the performance ability of generated rules. In this paper, we explain the proposed MAR model, testing and experimental results in looking into the performance of the model and conclusion.

Keywords: **Data Mining, Association Rules, Interesting Rules, Intelligent Technique, Data Analysis Technique.**

Introduction

Today, organizations and enterprises gather and store large amounts of data in their daily activity and operations continuously. Understanding or learning about the implicit or hidden information in the data is important for strategic decision support or technical operations. For instance, how supermarket could utilize patterns or rules discovered in their customers transactions (Mika, K., and Heikki, M., et.al., 1994; Wei, W., 1998; Sarjon, D., Mohd, N., 2001).

This is a great demand for analyzing data and turning them into useful knowledge (Hua, Z., 1998; Sarjon, D., Mohd, N., 2001). Therefore, it is necessary and interesting to examine how to extract hidden information / knowledge from large amounts of data automatically.

Association rules, introduced by Agrawal, Imielinski and Swami (Nimrod, M., Ramakrishnan, S., 1998; Heikki, M., et.al., 1994), is one of data mining technique to discover interesting rules or relationships among attributes in databases. It has attracted great attention in database research communities in recent years. The discovered rules may help marketing, decision making, and business management to make business decisions (David, W.C., Vincent, T.Ng., et.al (1996); Hua, Z., (1998); Juchen, H., Ulrich, G. et.al (2000); Ke, W., Yu, H., et.al (2000); Bing, L., Wynne, H., et.al (1999).

In the past research, a multitude of promising association rule methods have been developed. For example, Hua, Z (1998) proposed and developed an interesting association rule mining approach, called on-line analytical mining of association rules, which integrates the recently developed OLAP (on-line analytical processing) technology with some efficient association mining methods. Several algorithms are developed based on this approach for mining various kinds of association in multi dimensional databases, including intra and inter dimensional association, hybrid association and constraints based association. These algorithms give great advantages over many existing algorithm in term of flexibility and efficiency.

Yongjian, F (1996) examined how a knowledge discovery process may proceed to find different kinds of knowledge at multiple concept level in different kinds of databases, including relational databases and transaction databases. Several variants of the method with different optimization techniques are implemented and tested. The results shown that the method is efficient and effective.

However, these methods still have several weaknesses and need further improvement. Some of weaknesses are as follows:

1) These method did not apply a special data preprocessing techniques to identify which of data in databases are an inaccurate, irrelevant, inconsistent or missing values in the databases (Shan, C., 1998; Wei, W., 1999; Sarjon, D., Mohd, N., 2000).

2) The number of rules grows exponentially with the number of items. The key element that make association rule mining practical is Minimum Support, called MinSup. It is used to prune the search space and to limit the number of rules generated. Fortunately, today's algorithms are able to efficiently prune this immense search space based on minimal threshold for quality measures on the rules (Juchen, H., Ulrich, G., et.al (2000). If the frequencies of items vary a great deal, we will encounter two problems (Bing, L., Wynne, H., et.al (1999):

   a) If MinSUp is set too high, we will not find those rules that involve in frequent items or rare items in the data, and

   b) If we set MinSup very low, this may cause a combinatorial explosition, producing too many rules, because those frequent items will be associated with one another in all possible ways and many of them are meaningless.

3) Interesting rules must be picked from the set of generated rules. This might be quite costly because the generated rule sets normally are quite large, - i.e., more than 100.000 rules are not uncommon- and in contrast the percentage of useful rules is typically only a very small fraction (Bing, L., Wynne, H., et.al (1999).

Based on the above mentioned problems, at the present stage we propose a MAR (Mining Association Rules) model to overcome these problems. This model integrate intelligent and data analysis techniques , i.e., statistical analysis, rough sets and association rules techniques.

The rest of the paper is organized as follows. The proposed Mining Association Rule (MAR) model is given in section 2 and testing MAR model in section 3. The experimental results and conclusion are given in section 4 and 5 respectively.

2. Mining Association Rule (MAR) Model

Figure 2.1 represents the general architecture of MAR model. MAR model consists of two main modules, pre-processing and processing module. The first module, pre-processing, is used to transform data, identify and remove inconsistent data from databases. Next, processing, is executed to generate rules and evaluate the generated rules. Description of each module is given in following sub section.

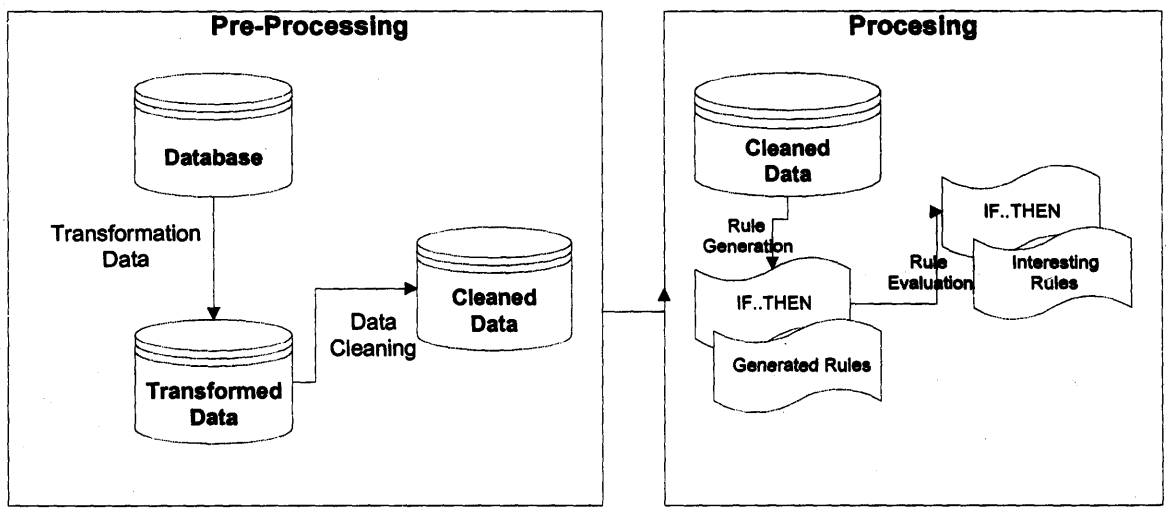


Figure 2.1: The General Architecture of MAR Model

2.1 Pre-Processing Module

This module consists of two main steps, namely data transformation and data cleaning. The explanation of these steps are given in section 2.1.1 and 2.1.2 respectively.

2.1.1 Data Transformation

The raw data in a databases is called at its primitive level and the knowledge is said to be at a primitive level if it is discovered using raw data only. Abstracting raw data to a higher conceptual level and discovering and expressing knowledge at higher abstraction level have superior advantages over data mining at a primitive level.

Discretization is one of the technique that can be used to transform the raw data into higher level concept. The discretization method will be explained in detail in the next paragraph.

2.1.1.1 Basic Concepts

Definition 2.1: In the discretization of a decision table A=(U,A{d}), where $V_a = (v_a, w_a)$ is an interval of reals. We search for a partition P_a of V_a for a $\in$ A. Any partion of V_a is defined by a sequence of so called cuts $V_1 < V_2 < \ldots < V_k$ from V_a. Hence, any familily of partitions $\{P_a\}_{a \in V_a}$ be identified with a set of cuts.

Example 2.1: Let us consider a consistent decision system with two conditional attributes price and index, and seven objects $U_1, U_2, \ldots, U_7$. The values of attributes on these objects and the values of the decision volume are presented in table 2.1.

Table 2.1: The Stock Information System

U	Price	Index	Volume
U_1	0.8	2	1
U_2	1	0.5	0
U_3	1.3	3	0
U_4	1.4	1	1
U_5	1.4	2	0
U_6	1.6	3	1
U_7	1.3	1	1

The sets of values of price and index on objects from U are given by :

$A_{(U)} = \{0.8, 1, 1.3, 1.4, 1.6\}$

$B_{(U)} = \{0.5, 1, 2, 3\}$

Definition 2.2: The discernibility formula $\varphi_{(i,j)}$ for different pairs (V_i, V_j) of discernible from U x U with different decision.

One can choose at least one cut on one of the attributes appearing in the entry (X_i, X_j) of the discernibility matrix of A. For any objects X_i and X_j discernible by conditional attributes which have different decisions. In our example, we have the following form as is shown in table 2.2.

Table 2.2: Discernibility Matrix of Condition Attribute

$\varphi_{(2,1)}$	$P_1a \vee P_1b \vee P_2b$
$\varphi_{(2,4)}$	$P_2a \vee P_3a \vee P_1b$
$\varphi_{(2,6)}$	$P_2a \vee P_3a \vee P_4a \vee P_1b \vee P_2b \vee P_3b$
$\varphi_{(2,7)}$	$P_2a \vee P_1b$
$\varphi_{(3,1)}$	$P_1a \vee P_2a \vee P_3b$
$\varphi_{(3,4)}$	$P_2a \vee P_2b \vee P_3b$
$\varphi_{(3,6)}$	$P_3a \vee P_4a$
$\varphi_{(3,7)}$	$P_2b \vee P_3b$
$\varphi_{(5,1)}$	$P_1a \vee P_2a \vee P_3a$
$\varphi_{(5,4)}$	P_2b
$\varphi_{(5,6)}$	$P_4a \vee P_3b$
$\varphi_{(5,7)}$	$P_3a \vee P_2b$

For example, the formula $\varphi_{(5,6)}$ is true on the set of cuts if there is exists a cut $P_1 = (a, c)$ on V_a in this set such that $c \in (1.4 \sim 1.6)$ or a cut $P_2 = (b, c)$ on V_b that $c \in (2 \sim 3)$.

Rows	P_1a	P_2a	P_3a	P_4a	P_1b	P_2b	P_3b	d
$\varphi_{(2,1)}$	1	0	0	0	1	1	0	1
$\varphi_{(2,4)}$	0	1	1	0	1	0	0	1
$\varphi_{(2,6)}$	0	1	1	1	1	1	1	1
$\varphi_{(2,7)}$	0	1	0	0	1	0	0	1
$\varphi_{(3,1)}$	1	1	0	0	0	0	1	1
$\varphi_{(3,4)}$	0	1	0	0	0	1	1	1

$\varphi_{(3,6)}$	0	0	1	1	0	0	0	1
$\varphi_{(3,7)}$	0	0	0	0	0	1	1	1
$\varphi_{(5,1)}$	1	1	1	0	0	0	0	1
$\varphi_{(5,4)}$	0	0	0	0	0	1	0	1
$\varphi_{(5,6)}$	0	0	0	1	0	0	1	1
$\varphi_{(5,7)}$	0	0	1	0	0	1	0	1

Definition 2.3: Set of Cuts. These are pairs (a,c) where $c \in V_a$. Cuts is defined by the middle points of the above defined intervals.

Example 2.2: In our example, we obtained the following cuts:

(a, 0.9); (a, 1.15); (a, 1.35); (a, 1.5)
(b, 0.75); (b, 1.5); (b, 2.5)

Any cut defines a new conditional attribute with binary values. For example, the attribute corresponding to the cut (a, 1.2) is equal to 0 if $a(x) < 1.2$ otherwise is equal to 1.

To construct a set of cuts with the minimal number of elements, Boolean Reasoning approach. can be used for any attribute a and any interval determined by a the corresponding Boolean variable. In our example the set of Boolean variables defined by A is equal to:

$$VB_{(A)} = \{P_1a, P_2a, P_3a, P_4a, P_1b, P_2b, P_3b\}$$

Where

P_1a = (0.8 ~ 1) of a (i.e., P_1a corresponds to the interval {0.8 ~ 1} of attribute a)

$P_2a = (1 \sim 1.3)$

$P_3a = (1.3 \sim 1.4)$

$P_4a = (1.4 \sim 1.6)$

$P_1b = (0.5 \sim 1)$

$P_2b = (1 \sim 2)$

$P_3b = (2 \sim 3)$

2.1.1.2 Discretization algorithm

Discretization algorithm is given in algorithm 2.1.

Algorithm 2.1: Discretization Algorithm

Input : Initial Databases
Output : Discrete Values
Method :

1) Sort raw data into ascending order based on conditional attribute

2) Define interval of each conditional attribute

3) Generate discernibility matrix, A

4) Copy A to A* to construct a new decision A*

5) For each column, find the total number of occurrence of 1, and sort them into descending order

6) Choose column for A* with the maximal number of occurrence of 1's

7) Delete from A* the column choose in step (6)

8) If row is not empty, go to (5), else define set of cuts to define interval of discrete value.

9) Creation of segments. A set of segments are created based on the value of set of cuts.

2.1.2 Data Cleaning

Organizations accumulate much data that they want to access and analyze as a consolidated whole. However, the data often has inconsistencies in schema, formats and adherence to constraints, due to many factors including data entry errors and emerging from multiple sources. The data must be purged of such discrepancies and transformed into a uniform format before it can be used (Vijayshankar, R., and Joseph, M.H., (2001); Michael, N., (2001).

Data cleaning and transformation are important tasks in many context such as data warehousing and data integration. The current approaches to data cleaning are time consuming and frustating due to long running non interactive operations, poor coupling between analysis and transformation, and complex transformation interface that often require user programming (Vijayshankar, R., and Joseph, M.H., (2001). The problem of data cleaning, which consists of removing inconsistencies and errors from original data sets, is well known in the area of Decision Support System and data warehouses (Helena, G., and Daniela, F., et.al (2001).

The basic concept of data cleaning is given in section 2.1.2.1

2.1.2.1 Basic Concepts

One of the most fundamental concepts in Rough Set theory is indiscernibility which is defined as in definition 2.4.

Definition 2.4: Let A = {U,A} be an Information System. Every subset of attributes $B \subseteq A$ defines an equivalence relation. IND (B), called indiscernibility relation, defined as follows:

$$IND(B) = \{(e_i,e_j) \in U \times U : a(e_i) = a(e_j) \text{ for all } a \in B \qquad (2.1)$$

IND (B) is called the B-indiscernibility relation if $(e_i, e_j) \in IND(B)$, then object e_i and e_j are indiscernibility from each other by attributes from B.

Example 2.3: Let us illustrate how a decision table such as table 2.2 defines an indiscernibility relation. The non empty subsets of the conditional attributes are {Price}, {Index} and {Price, Index}. The relation IND defines three partitions of the Universe.

IND (Price) = $\{U_1\}, \{U_2\}, \{U_3, U_4, U_7, U_8, U_9, U_{10}\}, \{U_5, U_6\}$

IND (Index) = $\{U_1\}, \{U_2, U_3, U_6, U_7, U_8, U_9, U_{10}\}$

IND (Price, Index) = $\{U_1\}, \{U_2\}, \{U_3, U_7, U_8, U_9, U_{10}\}, \{U_4\}, \{U_5\}, \{U_6\}$

Vagueness and uncertainty are approached through the definition of two sets called the Lower and Upper Approximations, which are as in definition 2.5.

Definition 2.5: Given an IS A = {U, A}, let $X \subseteq U$ be a set of objects and $B \subseteq A$ be a selected set of attributes. The B_Lower approximation $\underline{B}X$ and the B_Upper approximation BX of X with reference to attribute B are

$$\underline{B}X = \{x \in U : [x]B \subseteq X\} \quad (2.2)$$

$$BX = \{x \in U : [x]B \cap X \# 0\} \quad (2.3)$$

The objects in $\underline{B}X$ can be with certainty classified as members of X on the basis knowledge in B, while the objects in BX can be only classified as possible members of X on the basis of knowledge in B. The set $BNB(X) = BX - \underline{B}X$ is called the B-Boundary region of X.

2.1.2.2 Data Cleaning Algorithm

The term inconsistency has been used in the literature for several specific cases. One form of inconsistency occurs when two tuples match on all the key attribute values but have conflicting values for some non key attributes. While the above notion of conflicts between tuples with matching key attribute values implies an inconsistency, the lack of conflict between such tuples does not guarantee that the data is consistent. Another from of inconsistency occurs when there is an error in the values of the primary key attributes themselves (Shailesh, A., Arthur, M.K., et.al (1995). In this research, inconsistencies data is as in definition 2.6.

Definition 2.6: An example $e \in U$ is said to be inconsistent if there is an example $e' \in U$ such that $a(e) = a(e')$ for all $a \in A$ and $c(e) \# c(e')$ for some $c \in C$. An example is said to be consistent if it is not inconsistent.

Example 2.4: Table 2.1 shows that examples 3 and 7 are inconsistent since a (e) = a (e'), i.e., Price (e_4) = {Price , Index (e_6)}, while c(e) # c (e'), i.e., {c (e_4) # c(e_6)}.

Data cleaning algorithm is given in algorithm 2.2.

Algorithm 2.2: Data Cleaning Algorithm to Remove Inconsistencies Data

Input : inconsistent raw data

Output : consistent raw data

Method :

1. Sort raw data into ascending order based on conditional attribute

2. Generate indiscernibility of pairs conditional attributes

3. Compare indiscernibility of pairs attributes with decision attribute. If pairs of conditional attributes have different decision attribute, it is called inconsistent else inconsistent.

4. Generate approximation, i.e., lower and upper approximation, to remove inconsistencies in data.

2.2 Processing Module

This module consists of two main steps, called rule generation and rule evaluation. Description of these steps is given in following 2.2.1 and 2.2.2 respectively.

2.2.1 Rule Evaluation

2.2.1.1 Basic Concepts

A formal definition of association rule is given in definition 2.7.

Definiton 2.7: An association rule is a rule in the form of

$$A_1, A_2,\ldots,A_m \rightarrow B_1, B_2,\ldots,B_n$$

where A_i and B_j are predicates or items.

The left hand side of the rule (the part of $A_1, A_2,\ldots,A_m$) is known as the body of the rule, and the righ hand side (the part of $B_1, B_2,\ldots,B_m$) is the head of the rule.

The task of mining association rules is divided into two steps:

a. Find all combinations of items that have transaction support above minimum support
b. Use the frequent itemsets to generate the desired rules.

Definition 2.8: The itemset is said to be frequent itemsets if the support is no less than a minimum support, σ',. A valid association rules is an association rules with support and confidence greater than or equal to the minimum support, σ', and minimum confidence, φ' respectively. The definition indicates that if "X→Y' is valid, then $\sigma (X \cup Y) \geq \sigma'$, and $\varphi (X \rightarrow Y) \geq \varphi'$.

Example 2.5: The following rule is generated from the Stock Exchange database.

Stock (X,"AALI"), Price (X,"P136") → Volume (X,"V111")
[Support = 20%, Confidence = 80%

If the minimum support, σ', is 15% and the minimum confidence, φ', is 75%, the above rule is said valid association rule.

Minimum support is the key element that makes association rule mining practical. It is used to prune the search space and to limit the number of rules generated. If minimum support, σ', is set to high, we will not find those rules that involve infrequent items or rare items in the data, and in order to find rules that involve both frequent and rare items, we have to set minimum support, σ', very low. However, using only a single minimum support, σ', implicitly assumes that all items in

the data are the same nature and/or have similar frequencies in the database. This is often not the case in real applications. In many applications, some items appear very frequently in the data while other rarely appear.

In this paper, the definition of association rules remains the same, while the definition of minimum support is changed as is given in definition 2.9.

Definition 2.9: Let α be a total frequencies of items at level n+1 $\sum_{i=1}^{n} fr(i)$, and β the number of nodes in multi level database at level n+1 . The minimum support, σ', at level n is:

$$\sigma' = \frac{\sum_{i=1}^{n} fr(i)}{\beta} \tag{2.4}$$

$$\sigma' = \frac{\alpha}{\beta} \tag{2.5}$$

Example 2.6: Suppose the hierarchical concepts of Stock Exchange database as is shown in figure 5.1. Suppose the total frequencies α of Main Board at Level-1 is 42 and the number of nodes ,β, is 3. The minimum support ,σ', of this level is 14, while minimum support, σ', of Second Board is 5. It means that, the database can have more than one MinSup.

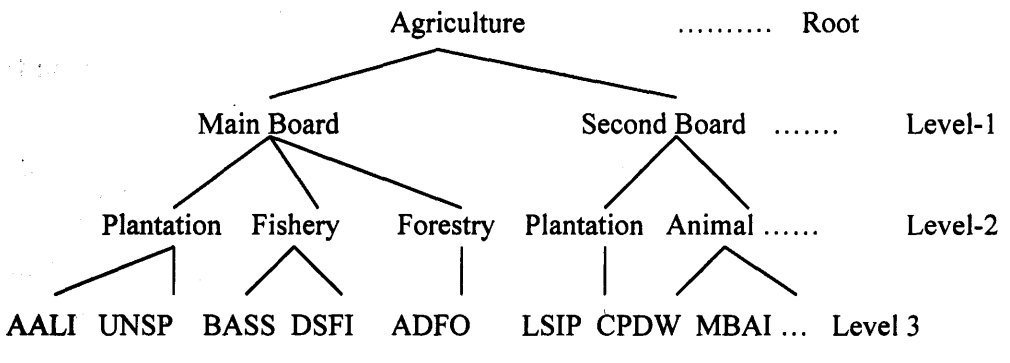


Figure 5.1: The Hierarchical Concepts of Stock Exchange Database

Definition 2.10: Let the minimum support of item I, , σ'(i). The Minimum Support of a rule r is the Lowest Minimum Support among the items in the rule. A rule r: $x_1,x_2,\ldots,x_m \rightarrow x_m+1, x_m+2,\ldots, x_r$, where $x_j \in I$, satisfies its Minimum Support , σ', if the rules actual support in the data is greater than or equal to

$$\text{Min} (\sigma'(x_1), \sigma'(x_2),\ldots, \sigma'(x_r)) \quad (2.6)$$

Example 2.7: The minimum support, σ', values as follows: σ' (AALI) is 70% and σ' (CPDW) is 55%. The following rule does not satisfy its minimum minimum support:

Buy (X,"AALI") → Buy (X,"CPDW") [Support = 50%]

Because Min (σ'(AALI), σ'(CPDW)) is 55%

2.2.1.2 Multiple Level Association Rules

Definition 2.11: Multiple level association rules are rules in which the concepts at multiple levels in conceptual hierarchies.

In general, mining knowledge at multiple levels is divided into categories, namely multiple level single dimensional and multiple level multi dimensional. The definition are given as follows:

Definition 2.12: Multiple level single dimensional association rule is the association within one dimension in which the concepts at the multiple level.

Example 2.8: This is the association within one dimension.

Stock (X,"S111") → Stock (X,"S212") [70%, 80%]

We can see that all items in this rule are from one dimension.

Definition 2.13: Multiple level multi dimensional association rule is the association among a set of dimensions in which the concepts at the multiple level.

Example 2.9: This is the association among a set of dimension

Stock (X,"S111"), Price (X,"P111") → Volume (X"V111") [60%, 75%]

We can see that all items in this rule are from different dimension.
Multiple level multi dimensional can be classified into two types, called multiple level multi dimensional with and without repetitive attributes. The definition of these association rules are given in the following definitions:

Definition 2.14: Multiple level multi dimensional without repetitive attributes is the association among a set of items in multiple level database in which all the items have distinct attribute name.

Example 2.10: The following rule is generated from Stock Exchange database.

Stock (X,"AALI"), Price (X,"P111") → Volume (X,"V112")

[Support = 60%, Confidence = 80%]

Definition 2.15: Multiple level multi dimensional with repetitive attributes is the association among a set of items in multiple level database in which one of attributes is repeated.

Example 2.11: The following rule is generated from Stock Exchange database.

Stock (X,"AALI"), Price (X,"P111") → Stock (X,"ADFO")

[Support = 70%, Confidence = 85%]

2.2.1.3 Multiple Level Multi Dimensional Algorithm

Multiple level multi dimensional algorithm is given in algorithm 2.3.

Algorithm 2.3: Multiple Level Multi Dimensional Algorithm

Input : A hierarchy information encoded and task relevant set of a transaction database in the format (tid, itemset)
Output : Multiple level large itemset

Method : A top down

i) Derive Minimum support, σ', of each items of attributes at each conceptual level, i.e., $\sigma'(A_1)$ is 5, $\sigma'(A_2)$ is 4, ..., $\sigma'(A_n)$ is 2.

ii) Sort the Minimum support into ascending or descending order, i.e., $\sigma'(A_1) > \sigma'(A_2 > \ldots.> \sigma'(A_n)$.

iii) Derive for each level l, start at level 1, the large k itemset, L[l,(i,p)], for each I and p, and the set of large itemset for all i's and p's. For every large itemset, the actual support must greater than or equal to minimum support.

iv) Derive the set of association rules from the every large itemsets for each level based on the minimum confidence at this level, Minconf (l).

2.2.2 Rule Evaluation

After frequent itemsets and discover rules are generated, the next process is to measure the interestingness of the rules. A rule is interesting or not can be identified either subjectively or objectively. Only the user can identify if a desired rule is interesting or not, and this identification may differ from one user to another. To solve this problem , objective interestingness based on the statistics can be used as one step to identify our rules is interesting or not.

Example 2.12: Suppose a rule AALI → ADFO [support = 80%, Confidence = 90%]. We can say that this rule is a strong association rule based on the support and confidence framework. However, this rule is incomplete and misleading since the overall support of ADFO is 95%, even greater than 90%.

To help filter out such incomplete and misleading strong association rules X→Y, we need to study how the two itames X and Y are correlated.

Definition 2.16: Let I be interestingness between X and Y, γ be possibility of item X and Y, γ1 and γ2 be possibility of item X and item Y respectively. The interestingness between X and Y, denote as $I_{X \to Y}$ is defined as follows:

$$I_{X\to Y} = \frac{\gamma}{\gamma 1+\gamma 2} \qquad (2.7)$$

The values of interestingness vary between 0 and 1. Item X and Y is said strong correlation if interestingness, $I_{X\to Y}$, greater than or equal to 1 and negatively correlation if interestingness less than 1.

Example 2.13 : Suppose the rule "AALI → ADFO". The possibility of AALI and ADFO is 0.50, possibility of AALI and ADFO are 0.75 respectively. The interestingness between AALI and ADFO, $I_{AALI\to ADFO}$ is 0.89. It means that buying AALI is negatively associated with buying ADFO and this rule is not interestingness enough to be reported.

In data mining, it would be natural to make use of association rules in defining interestingness between items.

Definition 2.17: We notice that β be support (X∪Y), β1 and β2 be support (X) and Support (B) respectively. The interestingness, IX→Y can be defined as follows:

$$I_{X\to Y} = \frac{\sigma(X \cup Y)}{\sigma(X)\sigma(Y)} \qquad (2.8)$$

$$I_{X\to Y} = \frac{\beta}{(\beta 1)(\beta 2)} \qquad (2.9)$$

Example 2.14: Suppose the rule "BASS→DSFI". The Support (BASS∪DSFI) is 0.50. The Support (BASS) and Support (DSFI) are 0.75 and 0.70 respectively. The Interestingness between AALI and DSFI, denoted as $I_{BASS\to DSFI}$, is 95%.

3. Testing MAR Model

We have tested and studied MAR model using Jakarta Stock Exchange (JSX). The values are organized into a conceptual hierarchy with three level. The data sample of inconsistent and consistent data are given in appendix A and B.

The process of mining association rules, i.e., using inconsistent data, is as follows:

1. Derive Minimum support, σ' from data in appendix A. The minimum support are as follows:

   Minsup (S1) = 12%
   Minsup (S2) = 4%
   Minsup (P1) = 9%
   Minsup (P2) = 9%
   Minsup (V1) = 8%
   Minsup (V2) = 2%

2. Generate the frequent itemsets. The frequent itemsets from data in appendix A is as follows:

Itemsets	Support
S1,P1,V1	23
S1,P1,V2	5
S1,P2,V1	8
S2,P1,V1	7
S2,P1,V2	2

3. Generate rules. Generated rules, i.e., Minconf = 50%, is as follows:

Itemsets	Support	Confidence
S1,P1,V1	23	0.82
S1,P2,V1	8	0.89
S2,P1,V1	7	0.78

4. Evaluate generated rules. The interstingness of generated rules is as follows:

Itemsets	Support	Confidence	Interestingness
S1,P1,V1	23	0.82	0.34
S1,P2,V1	8	0.89	0.16
S2,P1,V1	7	0.78	0.14

4. Experimental Results

Table 4.1and 4.2 show the experimental results of MAR model using inconsistent and consistent data.

Table 4.1: The Experimental Results Using Inconsistent Data

Level	Rules Generated ≥ Minsup	Valid Rules ≥ Minconf	Invalid Rules < Minconf	MDE	RME
1	83%	49.8%	33.2%	0.61	0.74
2	100%	37.5%	62.5%	0.41	0.71

Table 4.2: The Experimental Results Using Consistent Data

Level	Rules Generated ≥ Minsup	Valid Rules ≥ Minconf	Invalid Rules < Minconf	MDE	RME
1	67%	50.25%	16.75%	0.68	0.75
2	100%	50%	50%	0.73	0.84

In the following, we show the performance comparison of MAR model using inconsistent and consistent data.

The performance comparison at level-1 are shown in table 4.3 and graph 4.1.

Table 4.3: The Performance Comparison at Level-1

Data	Generated Ruled	Valid Rules	Invalid Rules
Inconsistent	83	49.8	33.2
Consistent	67	50.25	16.75

Graph 4.1: The Performance Comparison at Level-1

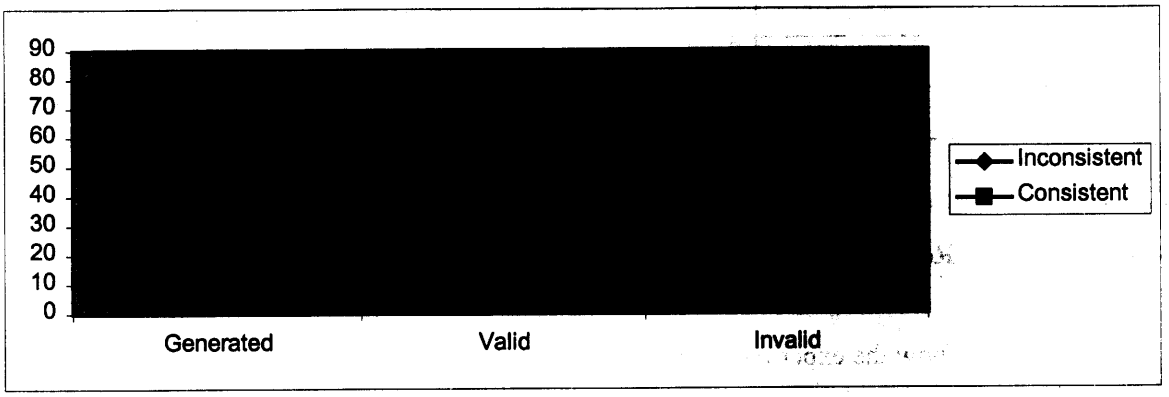
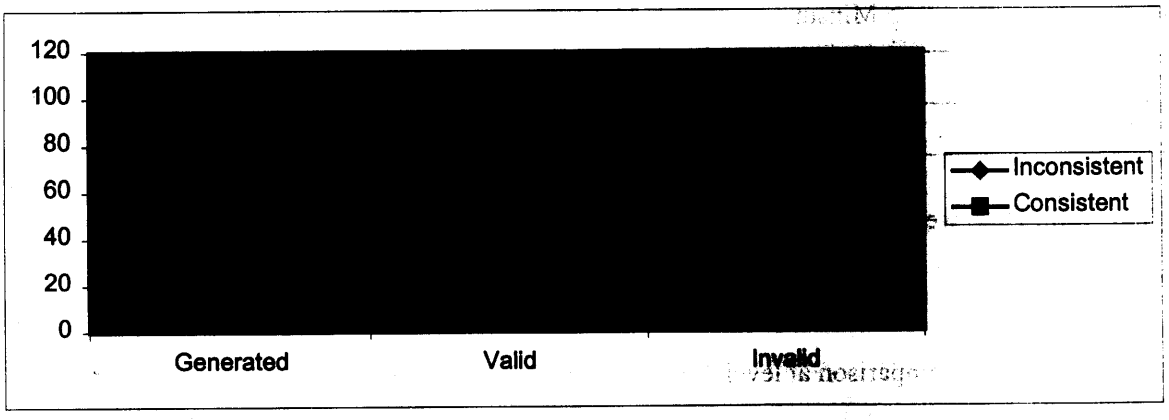


Table 4.4 and graph 4.2 show the performance comparison at level-2.

Table 4.4 The Performance Comparison at Level-2

Data	Generated Ruled	Valid Rules	Invalid Rules
Inconsistent	100	37.5	62.5
Consistent	100	50	50

Graph 4.2: The Performance Comparison at Level-2



From table 4.4, some improvement can be seen from several aspects. First, the valid rules is improved from 37.5% to 50% at level 2, an increment of 16.45%. Second, invalid rules percentage drops from 62.%% to 50%, a decrement of 12.5%. These great improvement prove that consistent data will give us the more better valid rules.

5. Conclusion

A number of mining association rules methods have been developed and studied. However, a single association rules method has not proven appropriate for every domain and data sets. Instead, several techniques may be needed to be integrated into hybrid system – two or more techniques are combined in a manner that overcomes the limitation of the individual technique. In this paper, we propose a MAR (Mining Association Rules) model which integrate intelligent technique and data analysis techniques. This model has been successfully implemented and studied using JSX (Jakarta Stock Exchange). Our results show that MAR model could give us a better rules or knowledge.

References:

Bing, L., Wynne, H., et.al (1999). " Mining Association Rules With Multiple Minimum Support", ACM SIGKDD, Augustust, 1999.

David, W.C., Vincent, T.Ng., et.al (1996). " Efficient Mining of Association Rules in Distributed Databases", IEEE Transaction on Knowledge and Data Engineering, Vol. 8, No. 6, December 1996.

Helena, G., and Daniela, F., et.al (2001). " Declarative Data Cleaning : Language, Model and Algorithms", Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

Hua, Z., (1998). "On-Line Analytical Mining of Association Rules", MSc Thesis, Simon Fraser University,1998.

Juchen, H., Ulrich, G., et.al (2000). "Algorithm for Association Rule Mining : A General Survey and Comparison", ACM SIGODKDD, July 2000.

Ke, W., and Yu, H., et.al (2000). "Mining Frequent Itemsets Using Support Constraints", Proceeding of the 26th VLDB Conference, Cairo, Egypt, 2000.

Michael, N., (2001). " Artificial Intelligence : A Guide to Intelligent Systems", Addison Wesley.

Mika, K., and Heikki, M., et.al (1994). "Finding Interesting Rules From Large Sets of Discovered Association Rules", The 3rd Conference on Information and Knowledge Management, November 29 – December 2, 1994, Gaithersburg, Maryland.

Sarjon, D., Mohd, N., (2001). "A Stock Price Prediction Based on Association Rules and Neural Network", KMICE 2001, Langkawi, 14-15 May 2001.

Shailesh, A., Arthur, M.K., et.al (1995). " Flecible Relation : An approach for Integrating Data From Multiple, Possibly Inconsistent Databases", IEEE Transaction, 1995.

Vijayshankar, R., and Joshep, M.H., (2001). "Potter's Wheel : An Interactive Data Cleaning System", Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

Wei, W., (1999). "Predictive Modeling Based on Classification and Pattern Matching Method", MSc Thesis, Computing Science, Simon Fraser University, Augustus 1999.

Yongjian, F., (1996). " Discovery of Muliple Level Rules From Large Databases", PhD Thesisi, Simon Fraser University, 1996.