Large-Scale Association Rule Discovery from Heterogeneous Databases with Missing Values using Genetic Network Programming.

Eloy Gonzales*, Takafumi Nakanishi* and Koji Zettsu*
* *Information Services Platform Laboratory*
Universal Communication Research Institute
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
Tel: +81-774-98-6866, Fax: +81-774-98-6960
e-mail: {egonzales, takafumi, zettsu}@nict.go.jp

***Abstract*—Association Rule Mining is an important data mining task and it has been studied from different perspectives. Recently multi-relational rule mining algorithms have been developed due to many real-world applications. However, current work has generally assumed that all the needed data to build an accurate model resides in a single database. Many practical settings, however, require the combination of tuples from multiple databases to obtain enough information to build appropriate models for extracting association rules. Such databases are often autonomous and heterogeneous in their schemes and data. In this paper, a method for association rule mining from large, heterogeneous and incomplete databases is proposed using an evolutionary method named Genetic Network Programming (GNP). Some other association rule mining methods can not handle incomplete data directly. GNP uses direct graph structure and is able to extract rules without generating frequent itemsets. The performance of the method is evaluated using real scientific heterogeneous databases with a high rate of missing data.**

Keywords-Association rule mining; heterogeneous databases; missing values; evolutionary computation.

I. INTRODUCTION

Data mining has emerged as an important area mainly due to the rapid growth of the size and number of databases in a variety of scientific and commercial domains. It had generated a great need for discovering knowledge hidden in large and heterogeneous databases. Thus, recently, data mining techniques focus on finding novel and useful patterns or rules from this kind of databases. Traditionally, data mining algorithms have focused on relational databases and assumed that all relevant information for building a model is present within a single database. Moreover, it is also assumed that the records in the databases are always complete. However, in today's real scenarios, the sources of information for effective data mining algorithms rely on a large number of diverse, heterogeneous, incomplete but interrelated data sources. That implies the combination of records from multiple databases to obtain enough information to build an accurate data mining model. One of the most important tasks in data mining is association rule mining, which is the process of identifying frequent patterns from a dataset that usually require some minimum support and minimum confidence. Then, they allow the construction of association rules which portray the patterns as predictive relationships between particular attribute values. During the last decade, many promising techniques for association rule mining [1][2] have been proposed which achieved effective performances. However, none of them handle incomplete databases. Most of the techniques either eliminate the missing values or replace them with an average or mean value. Nevertheless, it is not possible for all the types of datasets to fill with mean values or frequency, such as the combination of several heterogeneous and diverse databases. Therefore, new algorithms for extraction of interesting association rules directly from incomplete databases are necessary.

In this paper, a method for extracting general association rules from databases with missing values is proposed using an evolutionary optimization technique named Genetic Network Programming (GNP). The *missing completely at random* is the missing data induction mechanism considered because the missing data in the attributes of databases are independent on either the observed or the missing data. [3]. There have been some proposals of association rule mining using GNP [4][5]. Class association rules from incomplete datasets using GNP have been proposed [6][7], however these approaches are only effective in mining class association rules whose consequent parts are restricted within a class label. In this work, an extended method for mining general association rules from incomplete datasets is presented, which uses the *cosine measure* to evaluate the correlation of rules.

The following sections of this paper are organized as follows: In Section II, the concepts and explanations of general association rules are presented, the explanation of incomplete databases is introduced in Section III, the outline of GNP is briefly reviewed in Section IV where also the method for rule extraction from incomplete databases is presented. Simulation results are described in Section V, and finally, conclusion and future work are given in Section VI.

 ISBN: 978-1-61208-162-5

II. Association Rules

In this section, the definition and properties of association rules are briefly reviewed. The following is a formal statement of the problem of mining association rules [8]. Let $I = \{A_1, A_2, \ldots A_l\}$ be a set of attributes. Let G be a set of transactions,where each transaction T is a set of attributes such that $T \subseteq I$. Associated with each transaction is a unique identifier whose set is called TID. A transaction T contains X, a set of some attributes in I, if $X \subseteq I$. An association rule is an implication of the form of $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. X is called antecedent and Y is called consequent of the rule. Both are called **itemsets**. In general, an itemset is a non-empty subset of I.

Each itemset has an associated measure of statistical significance called support. If the fraction of transactions containing X in G equals *t*, then $support(X) = t$. The rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of $support(X \cup Y)/support(X)$. This measure indicates the relative frequency of the rule, that is, the frequency with which the consequent is also fulfilled when the antecedent is fulfilled .

The support-confidence framework is the most widely used model for mining association rules. The algorithm works in two phases, first searching of frequent itemsets in a database and then extract all association rules meeting user-specified constraints such as minimum support and minimum confidence. However, this framework is not enough for extracting interesting association rules [9], therefore additional correlation measures such as lift, chi-squared, cosine, etc. are very useful and convenient to improve the quality of the extracted rules. In this paper, *cosine correlation measure* is used in addition to support-confidence framework because it ensures that only positive correlation rules are extracted [10].

Given two itemsets X and Y, the cosine measure [10] is defined as:

$$cosine(X,Y) = \frac{P(X \cup Y)}{\sqrt{P(X)\ P(Y)}} = \frac{supp(X \cup Y)}{\sqrt{supp(X)\ supp(Y)}} \quad (1)$$

where,

$P(X \cup Y)$ is the probability of taking X and Y.
$P(X)$ is the probability of taking X.
$P(Y)$ is the probability of taking Y.
$supp(X \cup Y)$ is the support of X and Y.
$supp(X)$ is the support of X.
$supp(Y)$ is the support of Y.

Cosine is a number between 0 and 1. This is due to the fact that both $P(X \cup Y) \leq P(X)$ and $P(X \cup Y) \leq P(Y)$ are satisfied. A value close to 1 indicates a positive correlation between X and Y. The total number of transactions N is not taken into account by the cosine measure. Cosine measure is **null-invariant** because its value is not influenced by **null-transactions**. A **null-transaction** is a transaction that does not contain any of the itemsets being examined. Null-invariance is an important property for measuring correlations in large databases especially in the case of missing values.

III. Association Rules within an incomplete database

Most of the conventional association rule mining algorithms assume that databases are complete. Generally, databases are pre-processed in order to eliminate missing values or to replace them with an average or other statistical measures because the main problem in such kind of datasets is the difficulty for calculation of measures such as $support$, $confidence$ and $cosine$.

Table I
Example of database with missing data

TID	A_1	A_2	A_3	A_4	A_5	A_6
1	1	1	1	1	1	0
2	0	1	m	0	1	1
3	1	1	m	1	1	1
4	0	0	1	m	0	1
5	1	0	0	m	0	1
6	0	0	m	1	1	0
7	1	m	1	1	1	1
8	1	m	0	m	0	1
9	0	m	m	1	1	1
10	0	1	1	0	0	0

Table I is an example of an incomplete database which contains missing values. A_i is an attribute in the database. Missing data is represented as *"m"*, a different value of 1 or 0.

Considering Table I, the measurements for association rules from incomplete databases are calculated as follows: In case of the rule $(A_1) \rightarrow (A_5) \wedge (A_6)$, tuple $TID = 3$ includes A_1, A_5 and A_6, but tuple $TID = 10$ does not include neither A_1, A_5 and A_6. Notice that tuple $TID = 3$ contains missing data, however all records ($N = 10$) in the database are available for calculation of the measurements because it is possible to judge whether each record satisfy the rule or not. Consequently the measurements of the rule are: $support((A_1) \rightarrow (A_5) \wedge (A_6)) = 2/10$ and $confidence((A_1) \rightarrow (A_5) \wedge (A_6)) = 2/5$ as usual.

In the case of rule $(A_2) \wedge (A_5) \rightarrow (A_6)$, it is clear that tuples $TID = 2$ and $TID = 3$ satisfy completely the rule. Tuple $TID = 1$ does not satisfy the rule because it does not include A_6, that is $A_6 = 0$, the same as tuples $TID = 4$, $TID = 5$, $TID = 6$, $TID = 8$ and $TID = 10$ which contain at least one attribute whose value is 0 and therefore they surely do not satisfy the rule. However, these tuples are available for calculating the measurements. On the other hand, it is not possible to judge whether tuples $TID = 7$ and $TID = 9$ satisfy the rule or not because of the missing

 ISBN: 978-1-61208-162-5

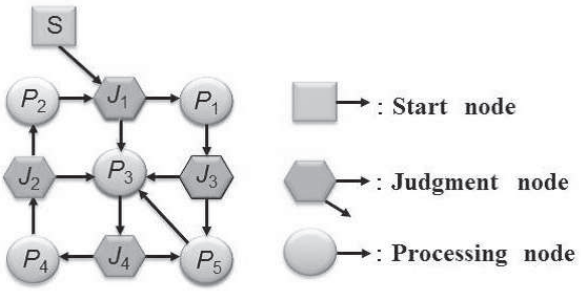


Figure 1. Basic structure of GNP

information of A_2; therefore, these tuples are omitted for the calculation of the measurements. Thus, the measurements of the rule are: $support((A_2) \wedge (A_5) \rightarrow (A_6)) = 2/8$ and $confidence((A_2) \wedge (A_5) \rightarrow (A_6)) = 2/3$.

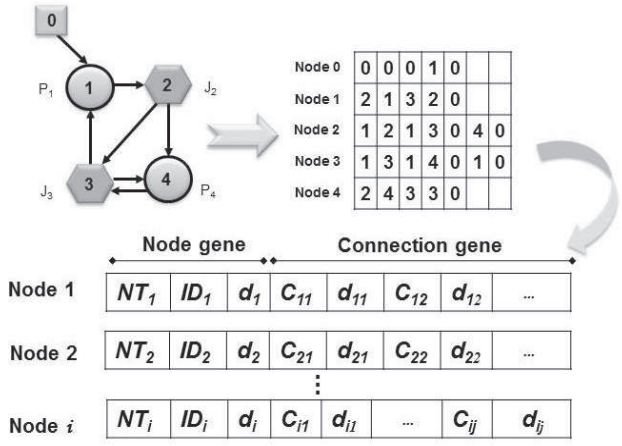
It is clear that the number of records N to be considered for the calculation of the measurements are different rule by rule. The available records for each rule are calculated according to the matching of the rule with the records. In other words, a record is included when it is ensured that it does not satisfy the rule (it contains any attribute with value 0) despite of it may contain missing values. Conversely, a record is excluded when it is not possible to judge if it satisfies the rule or not by missing values. Obviously, in the case of a complete database, i.e., with no missing data, N represent the total number of tuples in the database.

IV. Genetic Network Programming

Genetic Network Programming (GNP) is one of the evolutionary optimization algorithms, which evolves directed graph structures as solutions instead of strings (Genetic Algorithms) or trees (Genetic Programming) [11], [12], [13]. The main aim of developing GNP was to deal with dynamic environments efficiently by using the higher expression ability of graph structures.

The basic structure of GNP is shown in Fig. 1. The graph structure is composed of three types of nodes that are connected on a network structure: a start node, judgment nodes (diamonds), and processing nodes (circles). Judgment nodes are the set of J_1, J_2, ..., J_p, which work as *if-then* conditional decision functions and they return judgment results for assigned inputs and determine the next node to be executed. Processing nodes are the set of P_1, P_2, ..., P_q, which work as action/processing functions. The start node determines the first node to be executed. The nodes transition begins from the start node, however there are no terminal nodes. After the start node is executed, the next node is determined according to the node's connections and judgment results.

The gene structure of GNP (node i) is shown in Fig. 2. The set of these genes represents the genotype of GNP-individuals. NT_i describes the node type, $NT_i = 0$ when node i is the start node, $NT_i = 1$ when node i is a judgment



NT_i : node type (Start node=0; Judgment node=1; Processing node=2)
ID_i : identification number; d_i, d_{ij}: delay time; C_{ij} : connected node

Figure 2. Gene structure of GNP (node i)

node and $NT_i = 2$ when node i is a processing node. ID_i is an identification number, for example, $NT_i = 1$ and $ID_i = 1$ mean node i is J_1. C_{i1}, C_{i2}, ..., denote the nodes, which are connected from node i firstly, secondly, ..., and so on depending on the arguments of node i. d_i and d_{ij} are the delay time, which are the time required to execute the judgment or processing of node i and the delay time of transition from node i to node j, respectively.

In this paper, the execution time delay d_i and the transition time delay d_{ij} are not considered. All GNP-individuals in a population have the same number of nodes.

The characteristics of GNP are described as follows. (1) The judgment and processing nodes are repeatedly used in GNP, therefore the structure becomes compact and an efficient evolution of GNP is obtained. (2) Since the number of nodes is defined in advance, GNP can find the solutions of the problems without bloating, which can be sometimes found in Genetic Programming (GP). (3) Nodes that are not used at the current program execution will be used for future evolution. (4) GNP is able to cope with partially observable Markov processes. (5) The node transition in GNP individual is executed according to its node connections without any terminal nodes.

In the conventional GNP-based mining method, the attributes of the database correspond to the judgment nodes in GNP. Association rules are represented by the connections of nodes. Candidate rules are obtained by genetic operations. Rule extraction using GNP is done without identifying frequent itemsets used in Apriori-like methods [14]. Therefore, this method extracts important rules sufficient enough for user's purpose in a short time. The association rules extracted are stored in a pool through generations. The fundamental difference with other evolutionary methods is that GNP evolves in order to store new interesting rules in the pool, not to obtain the individual with the highest

 ISBN: 978-1-61208-162-5

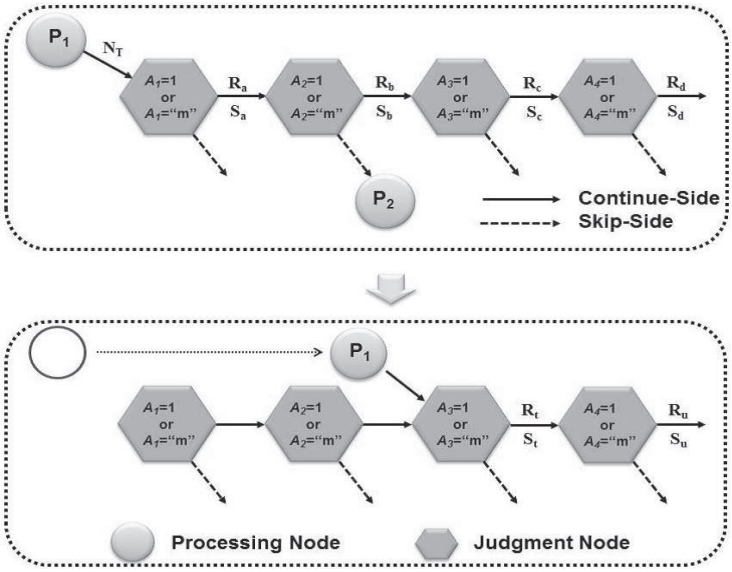


Figure 3. A connection of nodes in GNP for association rule mining with missing values

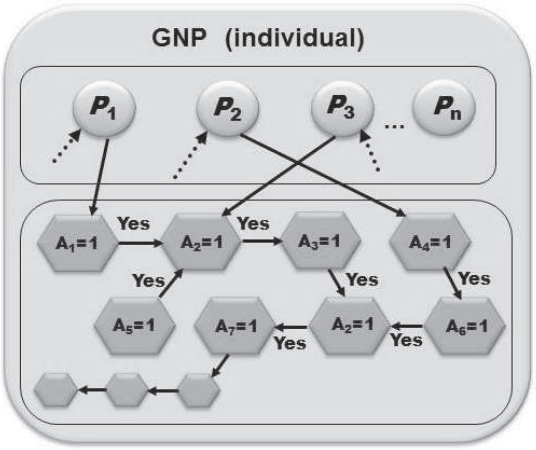


Figure 4. Basic structure of GNP for association rule mining

fitness value. GNP method has also advantages over other evolutionary methods such as Genetic Algorithms (GA) and Genetic Programming (GP). For GA-based methods [15], there are limitations in the number of association rules extracted because they are represented in individuals. In GP-base methods [16], an individual is usually represented by a tree with attribute values in the functions (e.g., logical, relational or mathematical operators) of the internal nodes. An individual's tree can grow in size and shape in a very dynamical way making it very difficult to understand for real applications.

A. GNP for rule extraction in an incomplete database

In this section, a general association rule mining method for incomplete databases is proposed using GNP. Let A_i be an attribute in an incomplete binary database and its value be 1, 0 or "m".

1) Rule Representation: Attributes and its values correspond to the functions of judgment nodes in GNP. Association rules are represented as the connections of nodes .

Fig. 3 shows a sample of the connection of nodes in GNP for association rule mining. P_1 is a processing node and is a starting point of association rules. "$A_1 = 1$", "$A_2 = 1$", "$A_3 = 1$" and "$A_4 = 1$" in Fig. 3 denote the functions of judgment nodes. Association rules are represented by the connections of these nodes, for example, $(A_1 = 1) \Rightarrow (A_2 = 1)$, $(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (A_3 = 1)$, $(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \Rightarrow (A_4 = 1)$ and $(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (A_3 = 1) \wedge (A_4 = 1)$.

Judgment nodes in GNP are used to examine the attribute values of database tuples and processing nodes calculate the measurements of association rules. Judgment nodes determine the next node by a judgment result. Each judgment node has two connections Continue-side and Skip-side. The Continue-side of the judgment node is connected to another judgment node. Skip-side of the judgment node is connected to the next numbered processing node. If the attribute value is 1 or "m", then move to the Continue-side. If the attribute value is 0, then the transition goes for the Skip-side.

For example in Table I tuple $TID = 5$ satisfy $A_1 = 1$ and $A_2 = 0$, therefore a transition from P_1 to P_2 occurs in the upper side of Fig. 3

A basic structure of GNP-individual for association rule mining is shown in Fig. 4. In Fig. 4, the Skip-side of judgment nodes is abbreviated.

Each processing node has an inherent numeric order (P_1, P_2, ..., P_s) and is connected to a judgment node. Start node connects to P_1. For each judgment node, the examinations of attribute values start and in case to move to the Continue-side continuously, the connection is obligatorily transfered to the next processing node using the Skip-node when the maximum number of attributes ($MaxLength$) in the rule is reached.

When the examination of the attribute values of tuple $TID = 1$ from the starting point P_s ends, then GNP examines the next tuple $TID = 2$ from P_1 likewise. Therefore, all tuples in the database are examined.

2) Rule Measurements: In GNP the number of tuples moving to the Continue-side are counted up and they are used for calculation of the measurements In Fig. 3, R_a, R_b, R_c and R_d are the number of tuples moving to the Continue-side at each judgment node when the attribute value is only 1. On the other hand, S_a, S_b, S_c and S_d represent the number of tuples moving to the Continue-side at each judgment node when the attribute value is 1 or "m". Therefore, the number of available records (N_x) for calculation of the rule measurements is given by the following equation:

$$N_x = N_T - (S_x - R_x) \quad (2)$$

where N_T is the total number of tuples in the database. For example N_b is obtained by $N_b = N_T - (S_b - R_b)$.

From Fig. 3, the $support$ and $confidence$ of rule $(A_1 = 1) \Rightarrow (A_2 = 1)$ is calculated as follows:

$$support((A_1 = 1) \rightarrow (A_2 = 1)) = R_b/N_b \quad (3)$$

$$confidence((A_1 = 1) \rightarrow (A_2 = 1)) = \frac{R_b/N_b}{R_a/N_a} \quad (4)$$

Important association rules are defined as the ones satisfying the following:

$$cosine > cosine_{min}, \quad (5)$$

$$support \geq sup_{min}, \quad (6)$$

$$confidence \geq conf_{min}, \quad (7)$$

$$confidence \geq support \quad (8)$$

$cosine_{min}$, sup_{min} and $conf_{min}$ are the minimum cosine, minimum support and minimum confidence values given by users. Table II shows an example of the measurements of some rules generated by node connections of Fig. 3.

Table II
EXAMPLE OF MEASUREMENTS OF ASSOCIATION RULES

Association Rule	*Support*	*Confidence*
$A_1 = 1 \rightarrow A_2 = 1$	$\frac{R_b}{N_b}$	$\frac{R_b/N_b}{R_a/N_a}$
$A_1 = 1 \rightarrow A_2 = 1 \wedge A_3 = 1$	$\frac{R_c}{N_c}$	$\frac{R_c/N_c}{R_a/N_a}$
$A_1 = 1 \rightarrow A_2 = 1 \wedge A_3 = 1 \wedge A_4 = 1$	$\frac{R_d}{N_d}$	$\frac{R_d/N_d}{R_a/N_a}$
$A_1 = 1 \wedge A_2 = 1 \rightarrow A_3 = 1$	$\frac{R_c}{N_c}$	$\frac{R_c/N_c}{R_b/N_b}$
$A_1 = 1 \wedge A_2 = 1 \rightarrow A_3 = 1 \wedge A_4 = 1$	$\frac{R_d}{N_d}$	$\frac{R_d/N_d}{R_b/N_b}$
$A_1 = 1 \wedge A_2 = 1 \wedge A_3 = 1 \rightarrow A_4 = 1$	$\frac{R_d}{N_d}$	$\frac{R_d/N_d}{R_c/N_c}$

The extracted important association rules are stored in a local pool all together through generations. When an important rule is extracted by GNP, the redundancy of the attributes is checked and it is also checked whether the important rule is new or not, that is, whether the rule is already in the local pool or not.

3) Genetic Operations: In order to extract important association rules it is necessary to change the connections of GNP-individuals. For instance, if the connection of P_1 is changed from node $A_1 = 1$ to node $A_3 = 1$ as shown in the lower part of Fig. 3, then, it is possible to calculate the support of $(A_3 = 1)$, $(A_3 = 1 \wedge A_4 = 1)$ and $(A_3 = 1 \wedge A_4 = 1 \wedge A_5 = 1)$ in the next examination.

Changing an attribute to another one or adding some attributes in the rules would be considered as candidates of important rules. These rules can be obtained effectively by GNP genetic operations, because mutation and crossover will change the connections or contents of the nodes.

Three kinds of genetic operators are used for judgment nodes: GNP-crossover, GNP-mutation-1 (change the connections) and GNP-mutation-2 (change the function of nodes).

- GNP-Crossover: uniform crossover is used. Judgment nodes are selected as the crossover nodes with the probability of P_c. Two parents exchange the gene of the corresponding crossover nodes.
- GNP-Mutation-1: Mutation-1 operator affects one individual. The connection of the judgment nodes is changed randomly by mutation rate of P_{m1}.
- GNP-Mutation-2: Mutation-2 operator also affects one individual. This operator changes the functions of the judgment nodes by a given mutation rate P_{m2}.

On the other hand, all the connections of the processing nodes are changed randomly.

At each generation, all GNP-individuals are replaced with the new ones by the following criteria: The GNP-individuals are ranked by their fitness values and the best one-third GNP-individuals are selected. After that, these GNP-individuals are reproduced three times for the next generation using the genetic operators described before.

If the probabilities of crossover (P_c) and mutation (P_{m1}, P_{m2}) are set at small values, then the same rules in the pool may be extracted repeatedly and GNP tends to converge prematurely at an early stage. These parameter values are chosen experimentally.

4) Fitness of GNP: The number of processing nodes and judgment nodes in each GNP-individual is determined based on experimentation depending on the number of attributes processed. The connections of the nodes and the functions of the judgment nodes at an initial generation are determined randomly for each GNP-individual.

Fitness of GNP is defined by:

$$F = \sum_{r \in R} \{cosine(r) + \alpha_{new}(r) + \beta(NA_A(r) - 1) + \beta(NA_C(r) - 1)\} \quad (9)$$

The terms in Eq. (9) are as follows:

R: set of suffixes of extracted important association rules satisfying (5), (6), (7) and (8)

cosine(r): value of *cosine correlation measure* of rule r

$\alpha_{new}(r)$: additional constant defined by

$$\alpha_{new}(r) = \begin{cases} \alpha_{new} & \text{(rule } r \text{ is new)} \\ 0 & \text{(rule } r \text{ has been already extracted)} \end{cases} \quad (10)$$

β: coefficient for the number of attributes.

$NA_A(r)$: the number of attributes in the antecedent of rule r.

$NA_C(r)$: the number of attributes in the consequent of rule r.

Constants in Eq. 9 are defined empirically based on the values of $cosine(r)$. Thus, $\beta = 0.10$ and $\alpha_{new}(r) = 0.3$.

$NA_A(r) \leq MaxLength$ and $NA_C(r) \leq MaxLength$. $MaxLength = 2T + 1$, where T is the number of heterogeneous databases.

Cosine(r), $NA_A(r)$ and $NA_C(r)$, and $\alpha_{new}(r)$ are concerned with the importance, complexity and novelty of rule r, respectively. The fitness represents the potential to extract new rules.

B. Algorithm Summary

The algorithm for discovering general association rules from heterogeneous data with missing values can be summarized as follows:

INPUT: A dataset with n binary attribute values with missing values, a predefined number of generations T, a predefined minimum support (sup_{min}), minimum confidence ($conf_{min}$) and minimum cosine ($cosine_{min}$) thresholds.

OUTPUT: A pool of general association rules with support, confidence and cosine values larger than or equal to the predefined minimum $support$, $confidence$ and $cosine$ thresholds.

$STEP$ 1: Randomly generate a population of GNP individuals with a predefined number of judgment and processing nodes.

$STEP$ 2: Extract general association rules using GNP as follows:

$STEP$ 2.1: Evaluate if an attribute is missing or not using judgment nodes by the following: the transition from one judgment node to another is executed when the value is 1 or "m". Then go to the Continue-side of the judgment node, otherwise, go to the Skip-side of the judgment node.

$STEP$ 2.2: Calculate the rule measurements (support, confidence and cosine) using the number of available records on the Continue-side at each judgment using the processing nodes. That is, N_x, S_x and R_x.

$STEP$ 3: Check whether an important rule is new or not (whether it is already in the pool or not)

$STEP$ 4: Store the new general association rule that satisfy the minimum support, confidence and cosine thresholds.

$STEP$ 5: If the number of generations T reaches, then stop the algorithm, otherwise go to the next step.

$STEP$ 6: Perform the evolution of the GNP individuals as follows:

$STEP$ 6.1: Calculate the fitness of each GNP individual.

$STEP$ 6.2: Select the top $1/3$ GNP individuals according to their fitness values.

$STEP$ 6.3: Execute the genetic operators to the selected GNP individuals in order to create the next population.

$STEP$ 7: Go to $STEP$ 2.

V. Simulation Results

In order to test and validate the effectiveness of the proposed method, two real-time scientific databases from UCI ML Repository [17] and World Data System (WDS) [18] were taken to conduct the experiments, which are frequently used in data mining community. Both of them contains heterogeneous spatial-temporal data and they are suitable for mining general association rules. The first one ("A" dataset) is El Nino dataset and contains oceanographic and surface meteorological readings taken from a series of buoys positioned throughout the equatorial Pacific. The second one ("B" dataset) correspond to the weather information of the Pacific Ocean taken by sensors of World Ocean Circulation Experiment (WOCE).

Table III shows the information of the original datasets, the first column of Table III shows the names of the datasets, the second shows the number of attributes, the third column shows the number of records, the fourth column shows if the dataset contains missing values and the fifth column shows the attribute characteristics of the dataset.

A. Experiment Setting

Both datasets are combined taken into account the date and each attribute is discretized into two corresponding attributes according to their values. For instance, if $Latitude \leq 0$ correspond to the $Latitude = South$. In this experiment, data only from one year (1993) is considered. After the discretization process, one large discretized dataset is generated, which contains 36 attributes and 20610 records. The combined dataset contains missing data, which varies for each attribute and ranges from 0% to 87%.

1) Parameters of GNP: The population size of GNP is 120. The number of processing nodes and judgment nodes in each GNP individual are 10 and 75, respectively. The maximum number of changing the connections of the processing nodes (*MaxLenght*) in each generation is $2(2) + 1 = 5$. The conditions of crossover and mutation are $P_c = 1/5$, $P_{m1} = 1/3$ and $P_{m2} = 1/5$. The termination condition T is 10, 30, 50 and 100 generations.

All algorithms were coded in Java language. Experiments were performed on a 3.2GHz Intel Xeon PC with 12G of main memory, running Windows 7 Ultimate 64bits.

Table IV shows some examples of the rules extracted by GNP. The termination "A" or "B" of each attribute means the correspondence to its dataset. From Table IV, the rules extracted by GNP are simple due to the small number of attributes in the antecedent part, which contribute to their understandability.

Fig. 5 shows the number of extracted rules when minimum confidence is 0.8 for different values of minimum support and number of generations. It can be seen that when the minimum support increases the number of rules extracted decreases for all generations because the constraints become more strict. Fig. 5 also shows that the number of rules increases when more generations in the evolution of GNP are used, especially at earlier generations.

Fig. 6 shows the number of extracted rules when the number of generations is 100 for different values of minimum support and minimum confidence. Fig. 6 shows that the minimum confidence has no great impact in the number of associations rules extracted compared with the minimum support.

 ISBN: 978-1-61208-162-5

Table III
INFORMATION OF THE ORIGINAL DATASETS

Dataset	*No. Attributes*	*No. Records*	Missing values	Attribute characteristics
El Nino	12	178080	Yes	Integer-Real
WOCE	14	71692	Yes	Integer-Real

Table IV
EXAMPLES OF RULES EXTRACTED BY GNP

Association Rules	Cosine
IF Air_Temp = High_A ∧ Longitude = East_B, THEN Longitude = West_A ∧ Speed = Low_B ∧ Temp_T_Air_C = Low_B	0.8327
IF Latitude=North_A ∧ Rel_Hum = High_B, THEN Longitude=West_A ∧ Speed=Low_B ∧ Pressure_Atm = Low_B ∧ Temp_T_Air_C = Low_B	0.8862
IF Sea_Surf_Temp=High_A ∧ Speed=Low_B ∧ Precip=High_B, THEN Pressure_Atm=Low_B ∧ Temp_Air=High_B	0.9179
IF Meridional_Winds=North_A ∧ Longitude=West_B ∧ Pressure_Atm=High_B, THEN Zon_Winds=West_A ∧ Rel_Hum=Low_B	0.9781
IF Latitude=South_A ∧ Temp_Water = High_B, THEN Longitude=West_A ∧ Zon_Winds = West_A ∧ Speed=High_B	0.8729
IF Zon_Winds = West_A ∧ Meridional_Winds = South_A ∧ Speed=Low_B, THEN Temp_T_Air_C = High_B	0.9297

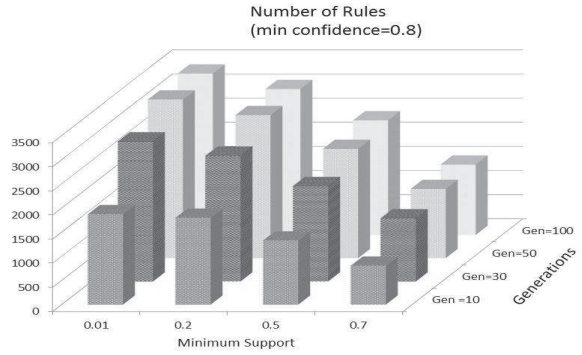


Figure 5. Number of extracted rules (min confidence=0.8)

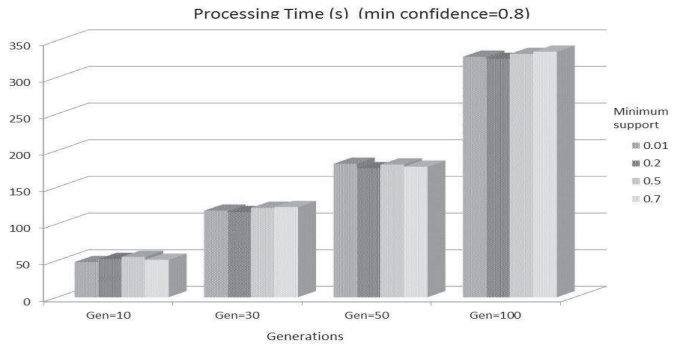


Figure 7. Processing Time (min confidence=0.8)

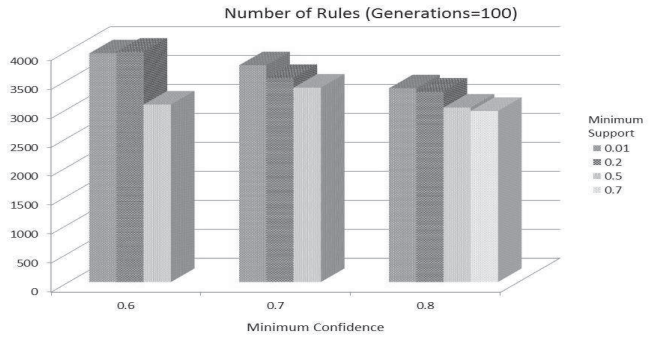


Figure 6. Number of extracted rules (generations=100)

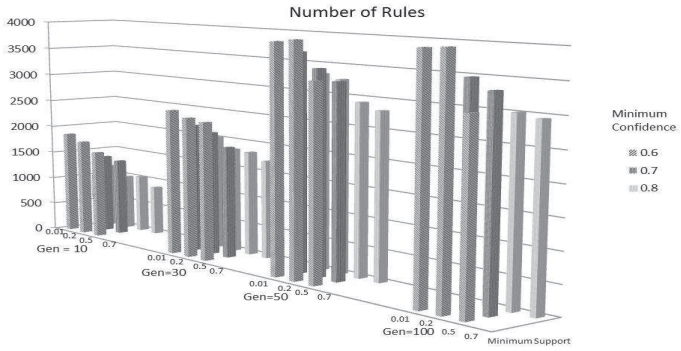


Figure 8. Number of Rules with different min support and min confidence

Fig. 7 shows the processing time for extraction of association rules when minimum confidence is 0.8 for different values of minimum support and number of generations. Fig. 7 shows that the processing time does not vary so much for a given generation as termination condition. On the other hand, the processing time increases when the number of generation increases because in every generation GNP searches and stores new association rules in the rule pool.

Fig. 8 shows the number of rules extracted with different conditions of minimum support, minimum confidence and the number of generations. Fig. 8 shows that although more

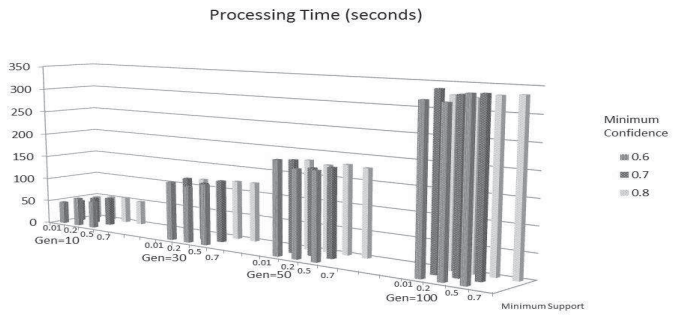


Figure 9. Processing Time with different min support and min confidence

association rules are extracted when used a larger number of generations, i.e., 100 generations; the difference, with the number of rules at 50 generations, is not so much. Therefore, most of the association rules are extracted in the earlier generations, which it is an advantage for the user's purpose.

The processing time increases when the number of generations are larger as shown in Fig. 9, however the number of extracted rules does not increase so much as it has been shown in Fig. 8. Therefore, 50 generations is enough in order to save processing time without risking of losing knowledge.

VI. Conclusion and Future Work

A method for association rule mining from incomplete databases has been proposed using GNP. An incomplete database includes missing data in some tuples, however, the proposed method can extract directly important rules using these tuples and users can define the conditions of important rules flexibly. The performance of the rule extraction has been evaluated using real data sets with a high rate of missing values. The results shows that the proposed method has the potential to realize associations considering heterogeneous databases and may be applied for rule discovery from incomplete databases in several other fields. For future work, the method may be extended to deal with large and heterogeneous databases with continuous values.

References

[1] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining Frequent Patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery, 8:53-87, 2004.

[2] A. K. H. Tung, H. Lu, J. Han, and L. Feng "Efficient Mining of inter-transaction association rules". IEEE Trans. on Knowledge and Data Engineering, 15(1): 43-56, 2003.

[3] A. Farhangfar, L. Kurgan, and J. Dy. "Impact of imputation of missing values on classification error for discrete data", *Journal of Pattern Recognition*, Vol 14, Issue 12, pp. 3692-3705, 2008.

[4] K. Shimada, K. Hirasawa, and T. Furuzuki, "Genetic Network Programming with Acquisition Mechanisms of Association Rules", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 10, No. 1, pp. 102-111, 2006.

[5] E. Gonzales, K. Taboada, K. Shimada, S. Mabu, and K. Hirasawa, "Combination of Two Evolutionary Methods for Mining Association Rules in Large and Dense Databases", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.13, No.5, pp. 561-572, 2009.

[6] K. Shimada, K. Hirasawa, and J.Hu, "Genetic Network Programming with class association rule acquisition mechanism from incomplete databases". In *Proc. of the Society of Instrument and Control Engineers Annual Conference 2007*, pp. 2708-2714, 2007.

[7] K. Shimada and K. Hirasawa, "A method of Association Rule Analysis for Incomplete Database using Genetic Network Programming", in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2010)*, pp.1115-5344, Portland, USA, 2010.

[8] C. Zhang and S. Zhang, *Association Rule Mining: models and algorithms*, Springer, 2002.

[9] C. C. Aggarwal and P.S. Yu . "A New Framework for Item Set Generation". *In: Proceedings of the ACM PODS Symposium on Principles of Database Systems*, pp. 18-24, Seattle, Washington (USA), 1998.

[10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Second Edition, Morgan Kauffman Publishers. USA, 2005.

[11] S. Mabu, K. Hirasawa, and J. Hu, "A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning", Evolutionary Computation, *MIT Press* , Vol 15, No. 3, pp. 369-398, 2007.

[12] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu, J. Hu, and S. Markon, "A Double-deck Elevator Group Supervisory Control System using Genetic Network Programming",*IEEE Trans. on System, Man and Cybernetics, Part C*, Vol.38, No.4, pp. 535-550, 2008.

[13] T. Eguchi, K. Hirasawa, J. Hu, and N. Ota, "A study of Evolutionary Multiagent Models Based on Symbiosis",*IEEE Trans. on System, Man and Cybernetics, Part B*, Vol.36, No.1, pp. 179-193, 2006.

[14] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", in *Proc. of the 20th VLDB Conf.*, pp. 487-499, 1994.

[15] C.Z. Janikow, "A knowledge-intensive genetic algorithm for supervised learning", *Machine Learning 13*, pp. 189-228, 1993.

[16] C.C. Bojarczuk, H.S. Lopes, and A.A. Freitas, "Genetic programming for knowledge discovery in chest pain diagnosis", *IEEE Trans. on Engineering in Medicine and Biology Magazine*, Vol. 19, No.4, pp. 38-44, 2000.

[17] Frank, A. Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [Last Access: Jun 14th, 2011]

[18] Walden, B; WOCE Surface Meteorology Data, WOCEMET (2006): Continuous meteorological surface measurement during KNORR cruise 316N138_12. Woods Hole Oceanographic Institution, Physical Oceanography Department.

 ISBN: 978-1-61208-162-5