A Proposal of Machine Learning by Rule Generation from Tables with Non-deterministic Information and Its Prototype System

doi: info:doi/10.1007/978-3-319-60837-2_43

A Proposal of Machine Learning by Rule Generation from Tables with Non-deterministic Information and Its Prototype System

Hiroshi Sakai[1], Michinori Nakata[2], and Junzo Watada[3]

[1] Graduate School of Engineering, Kyushu Institute of Technology,
Tobata, Kitakyushu 804-8550, Japan
sakai@mns.kyutech.ac.jp
[2] Faculty of Management and Information Science,
Josai International University,
Gumyo, Togane, Chiba 283-0002, Japan
nakatam@ieee.org
[3] Department of Computer & Information Sciences, Universiti Teknologi
PETRONAS, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia
junzo.watada@gmail.com

Abstract. A logical framework on *Machine Learning by Rule Generation* (MLRG) from tables with non-deterministic information is proposed, and its prototype system in SQL is implemented. In MLRG, the certain rules defined in *Rough Non-deterministic Information Analysis* (RNIA) are obtained at first, and each uncertain attribute value is estimated so as to cause the certain rules as many as possible, because the certain rules show us the most reliable information. This strategy is similar to the maximum likelihood estimation in statistics. By repeating this process, a standard table and the rules in its table are learned (or estimated) from a given table with non-deterministic information. Even though it will be hard to know the actual unknown values, MLRG will give a plausible estimation value.

Keywords: Machine learning by rule generation, Uncertainty, NIS-Apriori algorithm, SQL, Prototype.

1 Introduction

The management of information incompleteness in tables [3, 5, 7–13, 20] is still a very important issue in rough sets, data mining, machine learning, and soft computing. We followed *Nondeterministic Information Systems* (NISs) [9, 10] and the missing values [5], and proposed the framework of *Rough Non-deterministic Information Analysis* (RNIA) [12, 13]. Table 1 is an exemplary NIS Φ_{salary}, where each attribute value is given as a set or a missing value ?. We see that there is an actual value in each set but we do not know which is the actual value. We have characterized the rules in such NISs, and we are applying them to the several issues connected with information incompleteness.

Table 1. An exemplary NIS Φ_{salary}.

object	*age*	*depart(ment)*	*smoke*	*salary*
$x1$	$\{young\}$	$\{first\}$	$\{yes\}$	$\{low\}$
$x2$	$\{young, senior\}$	$\{first, second, third\}$	$\{yes\}$	$\{low\}$
$x3$	$\{senior\}$	$\{second\}$	$\{yes, no\}$	$\{high\}$
$x4$	$\{young, senior\}$	$\{second\}$	$\{no\}$	$\{high\}$
$x5$	$\{young\}$	?	$\{yes, no\}$	$\{high\}$
$x6$	$\{senior\}$	$\{third\}$	$\{no\}$	$\{high\}$

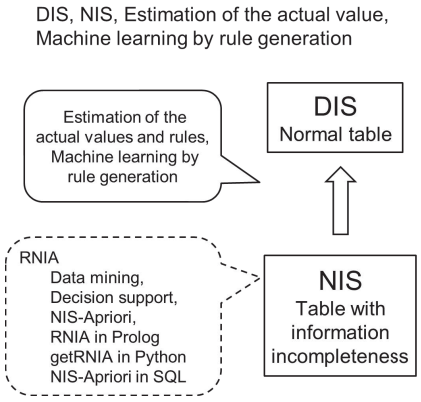

Fig. 1. A research map with respect to RNIA.

In RNIA, the *Apriori* algorithm [1] is extended to the *NIS-Apriori* algorithm [12, 13], and it generates the certain rules and the possible rules. These rules with modality are defined by using all possible tables derived from NIS [12, 13], and there may be a huge number of possible tables. For example, there are more than 10^{100} possible tables in the Mammographic data set in UCI machine learning repository [4]. Even though the definition of the certain rules and the possible rules is natural, it seemed hard to realize a rule generator for these rules due to the huge number of possible tables.

However, the NIS-Apriori algorithm affords a solution to this problem. Since it employs the mathematical property shown in [12, 13], it does not depend upon the number of all possible tables. Furthermore, the NIS-Apriori algorithm is *sound* and *complete* [14] for the certain rules and the possible rules. Recently, we are considering a software tool in SQL [16] in order to handle the large size data sets. Some actual execution logs including the Mammographic data set are in the web page [17].

Figure 1 shows the research map, where the block with the broken lines shows previous research and the block with the solid line does the purpose in this paper. We are applying the NIS-Apriori algorithm to machine learning (or estimating

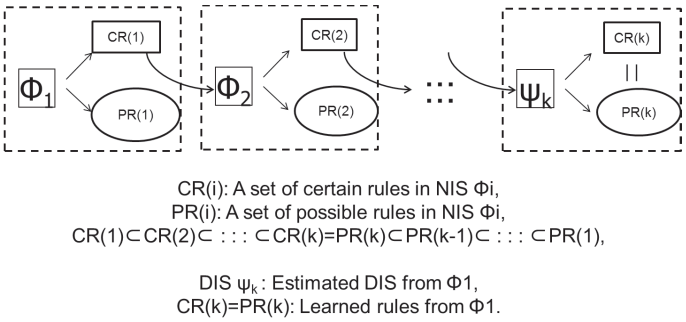

Fig. 2. A chart on machine learning by rule generation [15].

the actual attribute values) from NIS. The idea is the following: We obtain the certain rules in NIS Φ_i by using the NIS-Apriori algorithm, and we change Φ_i to NIS Φ_{i+1} so as to cause the obtained certain rules as many as possible [15]. By repeating this procedure, we finally obtain a standard table *Deterministic Information System* (DIS). Figure 2 shows a chart on *Machine Learning by Rule Generation* (MLRG).

As far as we know, the system based on the NIS-Apriori algorithm is unique, so a software tool on MLRG is also unique. Of course, it will be hard to know the unknown actual values, but MLRG will give a plausible estimation value. This paper is organized as follows: Section 2 surveys the framework of RNIA, and Section 3 proposes MLRG. Section 4 presents an experimental example, and Section 5 investigates some procedures in SQL. Section 6 concludes this paper.

2 Background of Rules in NISs and NIS-Apriori Based Rule Generation

This section briefly reviews RNIA, and describes how NIS-Apriori algorithm solves the computational problem for handling non-deterministic information.

2.1 RNIA and Rule Generation

At first, we clarify the rules in DIS. A pair $[A, val_A]$ of an attribute A and an attribute value val_A is called a *descriptor*. For a fixed decision attribute Dec and a set CON of attributes, an implication $\tau : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val]$ is (a candidate of) a *rule*, if τ satisfies the next two constraints for two given threshold values $0 < \alpha, \beta \leq 1.0$.
(1) $support(\tau)(= N(\tau)/|OB|) \geq \alpha$,
(2) $accuracy(\tau)(= N(\tau)/N(\wedge_{A \in CON}[A, val_A])) \geq \beta$.
Here, $N(*)$ means the number of the objects satisfying the formula $*$, and OB means a set of all objects.

In NIS Φ, we replace each non-deterministic information or a missing value ? with a possible value, and we obtain one DIS. We named it a *derived DIS* from NIS. Let $DD(\Phi)$ be a set of all derived DISs from Φ. We see an actual DIS ψ^{actual} exists in $DD(\Phi)$. For Φ_{salary}, $DD(\Phi_{salary})$ consists of 144 ($=3^2 \times 2^4$) derived DISs. Based on $DD(\Phi)$, we proposed the certain and the possible rules below:

Definition 1. *[12, 13]*
(1) τ is a certain rule, if τ is a rule in each $\psi \in DD(\Phi)$,
(2) τ is a possible rule, if τ is a rule at least one $\psi \in DD(\Phi)$.

The above two types of rules follow the modal concepts [8] by Lipski. Since a certain rule τ is also a rule in an actual DIS ψ^{actual}, this τ is the most reliable. Every certain rule is not influenced by the information incompleteness. On the other hand, a possible rule may be a rule in an actual DIS ψ^{actual}. These two types of rules will be one example of three way decision [20] by Yao.

Even though the definition of rules seems natural, we need to handle a huge number of DISs. For this computational problem, we defined two sets for a descriptor $[A, val]$ below:

$inf([A, val])$=$\{x : object \mid$ the value of x for A is a singleton set $\{val\}\}$,
$sup([A, val])$=$\{x : object \mid$ the value of x for A is a set including $val\}$,
$inf(\wedge_{A \in CON}[A, val_A])$=$\cap_{A \in CON} inf([A, val_A])$,
$sup(\wedge_{A \in CON}[A, val_A])$=$\cap_{A \in CON} sup([A, val_A])$.

For example, $inf([age, young])$=$\{x1, x5\}$ and $sup([age, young])$=$\{x1, x2, x4, x5\}$ hold in Φ_{salary}. The actual equivalence class is between two sets. For $minsupp(\tau)$ ($=min_{\psi \in DD(\Phi)}\{support(\tau)$ by $\psi\}$) and $minacc(\tau)$ ($=min_{\psi \in DD(\Phi)}\{accuracy(\tau)$ by $\psi\}$), we have the following which do not depend upon the number of $DD(\Phi)$.

$$\begin{array}{l} \tau : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val], \\ minsupp(\tau) = |inf(\wedge_{A \in CON}[A, val_A]) \cap inf([Dec, val])|/|OB|, \\ minacc(\tau) = \frac{|inf(\wedge_{A \in CON}[A, val_A]) \cap inf([Dec, val])|}{|inf(\wedge_{A \in CON}[A, val_A])| + |OUTACC|}, \\ OUTACC = \{sup(\wedge_{A \in CON}[A, val_A]) \setminus inf(\wedge_{A \in CON}[A, val_A])\} \\ \quad \setminus inf([Dec, val]). \end{array} \tag{1}$$

The $OUTACC$ means a set of objects, from which we can obtain an implication $\tau' : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val']$ ($val \neq val'$, the same condition part and the different decision). Similarly, we can calculate $maxsupp(\tau)$ and $maxacc(\tau)$. We can also prove that there exists $\psi_{min} \in DD(\Phi)$ which makes both $support(\tau)$ and $accuracy(\tau)$ the minimum. There exists $\psi_{max} \in DD(\Phi)$ which makes both $support(\tau)$ and $accuracy(\tau)$ the maximum. Based on these results, we have the chart in Figure 3 and Theorem 1.

Theorem 1. *[12, 13] For an implication τ, we have the following.*
(1) τ is a certain rule, if and only if $minsupp(\tau) \geq \alpha$ and $minacc(\tau) \geq \beta$.
(2) τ is a possible rule, if and only if $maxsupp(\tau) \geq \alpha$ and $maxacc(\tau) \geq \beta$.

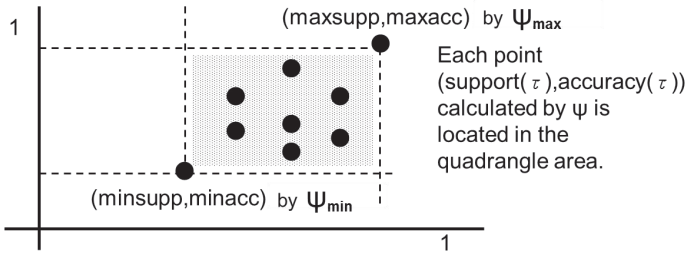

Fig. 3. For every implication τ, each point $(support(\tau), accuracy(\tau))$ by $\psi \in DD(\Phi)$ is located in a rectangle area.

Even though the certain rules and the possible rules depend upon $DD(\Phi)$, it is enough to examine two points ψ_{min} and ψ_{max}. Based on Theorem 1, we can escape from the exponential order problem. Without Theorem 1, it will be hard to handle rules in NISs like Mammographic data set, which has more than 10^{100} derived DISs.

2.2 NIS-Apriori Algorithm and Its Implementation

In order to handle the certain rules and the possible rules in NISs, we adjusted *Apriori* algorithm [1] to NISs, and named it *NIS-Apriori* algorithm [13]. NIS-Apriori algorithm consists of two phases, namely the certain rule generation phase and the possible rule generation phase. We employ *minsupp* and *minacc* values in certain rule generation, and we do *maxsupp* and *maxacc* values in possible rule generation. Since we can calculate *minsupp*, *minacc*, *maxsupp*, and *maxacc* by using inf and *sup* information, the NIS-Apriori algorithm is independent from the number of derived DISs.

Recently, we implemented the NIS-Apriori algorithm in SQL [16], and opened the execution logs [17], for example Lenses, Car Evaluation, Mammographic, Credit Card Approval, Congressional Voting data sets in UCI machine learning repository [4].

The analysis on the computational complexity of the NIS-Apriori algorithm is still in progress. This algorithm consists of two phases, and the Apriori algorithm is applied to each phase. Therefore, we figure out that the computational complexity of the NIS-Apriori algorithm is more than twice the complexity of the Apriori algorithm.

3 Machine Learning by Rule Generation in NISs

This section proposes the framework of MLRG including two strategies for learning by rule generation, and applies RNIA to realizing the MLRG process.

3.1 Motivation and Purpose

The chart of the proposing MLRG process is in Figure 2. Since the environment for NIS-Apriori based rule generation is getting better, we can easily obtain the sets $CR(i)$ (i=1, 2, 3, $\cdots$) of the certain rules in Figure 2. If we recognize them as reliable information, it seems natural that we fix a value so as to cause the reliable rules as many as possible. We think that this concept is similar to the maximum likelihood estimation [2] and MLRG will be a new approach for estimating one DIS from NIS. In this paper, we propose this framework and realize a software tool for MLRG.

3.2 Some Properties on CR(i) and PR(i)

Let us consider Figure 2, then we easily have the properties in the following.

Proposition 1. *In Figure 2, $CR(i) \subset PR(i)$ holds in every Φ_i.*

Proof. In Φ_i, $\tau \in CR(i)$ is a rule in each $\psi \in DD(\Phi_i)$. So, τ satisfies the condition of the possible rule, namely $\tau \in PR(i)$.

Proposition 2. *In Figure 2, $CR(i) \subset CR(i+1)$ holds.*

Proof. In Φ_i and Φ_{i+1} in Figure 2, $DD(\Phi_{i+1}) \subset DD(\Phi_i)$ holds, because some unfixed attribute values in $DD(\Phi_i)$ are fixed in $DD(\Phi_{i+1})$. Let $minsupp(\tau, i)$ be $minsupp(\tau)$ and $minacc(\tau, i)$ be $minacc(\tau)$ in Φ_i. Since $minsupp(\tau, i)$ is the minimum value in $DD(\Phi_i)$ and $minsupp(\tau, i+1)$ is the minimum value in $DD(\Phi_{i+1})$, clearly $minsupp(\tau, i) \leq minsupp(\tau, i+1)$ and $minacc(\tau, i) \leq minacc(\tau, i+1)$ hold. So, if τ is a certain rule in Φ_i, we have $\alpha \leq minsupp(\tau, i) \leq minsupp(\tau, i+1)$ and $\beta \leq minacc(\tau, i) \leq minacc(\tau, i+1)$. This means τ is also a certain rule in Φ_{i+1}, namely $CR(i) \subset CR(i+1)$.

Proposition 3. *In Figure 2, $PR(i+1) \subset PR(i)$ holds.*

Proof. In Φ_i and Φ_{i+1} in Figure 2, $DD(\Phi_{i+1}) \subset DD(\Phi_i)$ holds. Let $maxsupp(\tau, i)$ and $maxacc(\tau, i)$ be $maxsupp(\tau)$ and $maxacc(\tau)$ in Φ_i, respectively. Then, clearly $maxsupp(\tau, i+1) \leq maxsupp(\tau, i)$ and $maxacc(\tau, i+1) \leq maxacc(\tau, i)$ hold. Therefore, if τ is a possible rule in Φ_{i+1}, we have $\alpha \leq maxsupp(\tau, i+1) \leq maxsupp(\tau, i)$ and $\beta \leq maxacc(\tau, i+1) \leq maxacc(\tau, i)$. This means τ is also a possible rule in Φ_i, namely $PR(i+1) \subset PR(i)$.

Therefore, for the fixed threshold values α and β, we have the following inclusion relation in Figure 2. The uncertainty is sequentially reduced, and finally we have one DIS ψ_k.

$$CR(1) \subset CR(2) \subset \cdots \subset CR(k) = PR(k) \subset \cdots \subset PR(2) \subset PR(1). \quad (2)$$

3.3 The Framework of MLRG and Two Strategies

In NIS Φ_i, we fix some attribute values, and have a new NIS Φ_{i+1}. Since there is the inclusion relations in formula (2), we finally have the rules $CR(k)(= PR(k))$ in DIS ψ_k. We named this process MLRG. So, the most important issue on MLRG is how we fix some attribute values.

For this issue, we employ the certain rules in Φ_i. Let τ_1, τ_2, $\cdots$, τ_m be the certain rules in $CR(i)$. The order of each τ_i is defined such that the first priority is *minacc* (descending order) and the second priority is *minsupp* (descending order). By using the ordered certain rules, we propose the next two strategies.

(Strategy 1) (Positive Unification) *In an object x, a value is assigned to the unfixed value so as to cause a higher ordered certain rule.*
(Strategy 2) (Contradiction Prevention) *In an object x, a value is assigned so as not to contradict a higher ordered certain rule.*

Two strategies try to support the obtained certain rules in Φ_i much more. We may see this strategy as that we locally find a functional dependency between attributes and we reinforce its dependency much more. These strategies will also take the similar role of the maximum likelihood estimation in statistics. Each parameter is estimated so as to cause the likelihood function to the maximum in statistics, and each value is estimated so as to support the higher ordered certain rules in MLRG.

Remark 1. In the prototype system based on two strategies, we use only the certain rules with one condition, namely the certain rules in the form of $[A, val_A] \Rightarrow [Dec, val]$ for simplicity. We do not consider the certain rules in the form of $p_1 \wedge p_2 \Rightarrow q$ and $p_1 \wedge p_2 \wedge p_3 \Rightarrow q$. (The current system by NIS-Apriori algorithm generates rules with maximally three conditions.)

Remark 2. Two strategies employ the certain rules with one condition in Φ_i. Without the background of certain rule generation in RNIA, we can consider neither two strategies nor MLRG.

4 An Example of MLRG

For simplicity, we present an example of MLRG, and we describe the details of the prototype system in the next section.

We employ NIS Φ_{salary} in Figure 4 in this section. This Φ_{salary} consists of 6 objects, 4 attributes, age: {young,senior}, depart(ment): {first,second,third}, smoke: {yes,no}, and salary: {low,high}. Non-deterministic information is expressed by a list like $\{young, senior\}$. The decision attribute is 'salary'. There are 144 $(=2^4 \times 3^2)$ derived DISs in Φ_{salary}.

Figure 5 shows $CR(1)$ ($support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$) in Figure 2, and Figure 6 does $PR(1)$ ($support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$) in Figure 2.

```
mysql> select * from `table 1`;
+--------+----------------+----------------------+----------+--------+
| object | age            | depart               | smoke    | salary |
+--------+----------------+----------------------+----------+--------+
|      1 | young          | first                | yes      | low    |
|      2 | [young,senior] | [first,second,third] | yes      | low    |
|      3 | senior         | second               | [yes,no] | high   |
|      4 | [young,senior] | second               | no       | high   |
|      5 | young          | ?                    | [yes,no] | high   |
|      6 | senior         | third                | no       | high   |
+--------+----------------+----------------------+----------+--------+
6 rows in set (0.00 sec)
```

Fig. 4. The original data set of Φ_{salary}.

```
mysql> select * from c1_rule;
+-----------+------------+--------+--------+------------+---------+--------+
| Crule_num | att1       | val1   | deci   | deci_value | minsupp | minacc |
+-----------+------------+--------+--------+------------+---------+--------+
| c1        | age        | senior | salary | high       |   0.333 |  0.667 |
| c2        | depart     | second | salary | high       |   0.333 |  0.667 |
| c3        | smoke      | no     | salary | high       |   0.333 |  1.000 |
| 0         | end_attrib | NULL   | NULL   | NULL       |    NULL |   NULL |
+-----------+------------+--------+--------+------------+---------+--------+
4 rows in set (0.00 sec)
```

Fig. 5. $CR(1)$: Three certain rules satisfying $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in each of 144 derived DISs.

```
mysql> select * from p1_rule;
+-----------+------------+--------+--------+------------+---------+--------+
| Prule_num | att1       | val1   | deci   | deci_value | maxsupp | maxacc |
+-----------+------------+--------+--------+------------+---------+--------+
| p1        | age        | senior | salary | high       |   0.500 |  1.000 |
| p2        | age        | young  | salary | high       |   0.333 |  0.667 |
| p3        | age        | young  | salary | low        |   0.333 |  0.667 |
| p4        | depart     | first  | salary | low        |   0.333 |  1.000 |
| p5        | depart     | second | salary | high       |   0.500 |  1.000 |
| p6        | depart     | third  | salary | high       |   0.333 |  1.000 |
| p7        | smoke      | no     | salary | high       |   0.667 |  1.000 |
| p8        | smoke      | yes    | salary | low        |   0.333 |  1.000 |
| 0         | end_attrib | NULL   | NULL   | NULL       |    NULL |   NULL |
+-----------+------------+--------+--------+------------+---------+--------+
9 rows in set (0.00 sec)
```

Fig. 6. $PR(1)$: Eight possible rules satisfying $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in at least one derived DIS.

In this case, each rule occasionally consists of one condition, however generally NIS-Apriori algorithm generates the rules with maximally three conditions.

The set of rules in DIS ψ^{actual} is a superset of $CR(1)$ and a subset of $PR(1)$. By using MLRG, we estimate ψ^{actual} and the rules in ψ^{actual}. Let us see Figure 7. The step1('salary',6,0.3,0.6) command (the decision attribute is 'salary', the

```
mysql> call step1('salary',6,0.3,0.6);
Query OK, 0 rows affected (0.37 sec)

mysql> call step2('salary',6,0.1,0.3);
Query OK, 0 rows affected (0.33 sec)

mysql> select * from estimated_dis;
+--------+--------+--------+-------+--------+
| object | age    | depart | smoke | salary |
+--------+--------+--------+-------+--------+
|      1 | young  | first  | yes   | low    |
|      2 | young  | first  | yes   | low    |
|      3 | senior | second | no    | high   |
|      4 | senior | second | no    | high   |
|      5 | young  | second | no    | high   |
|      6 | senior | third  | no    | high   |
+--------+--------+--------+-------+--------+
6 rows in set (0.00 sec)
```

Fig. 7. An execution of machine learning by rule generation and an estimated DIS.

```
mysql> select * from rule1;
+------------+--------+--------+------------+---------+----------+
| att1       | val1   | deci   | deci_value | support | accuracy |
+------------+--------+--------+------------+---------+----------+
| age        | senior | salary | high       |   0.500 |    1.000 |
| age        | young  | salary | low        |   0.333 |    0.667 |
| depart     | first  | salary | low        |   0.333 |    1.000 |
| depart     | second | salary | high       |   0.500 |    1.000 |
| smoke      | no     | salary | high       |   0.667 |    1.000 |
| smoke      | yes    | salary | low        |   0.333 |    1.000 |
| end_attrib | NULL   | NULL   | NULL       |    NULL |     NULL |
+------------+--------+--------+------------+---------+----------+
7 rows in set (0.00 sec)
```

Fig. 8. The estimated rules ($CR(3)$=$PR(3)$) satisfying $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in the estimated DIS ψ^{actual}.

number of objects is 6, $minsupp \geq 0.3$, and $minacc \geq 0.6$) generates 3 certain rules in Figure 5, then fixes some attribute values based on two strategies. The step2('salary',6,0.1,0.3) command (the decision attribute is 'salary', the number of objects is 6, $minsupp \geq 0.1$, and $minacc \geq 0.3$) does the similar procedure. In order to find more certain rules, we loosened the constraints to $support(\tau) \geq 0.1$ and $accuracy(\tau) \geq 0.3$. After the step1 and step2 commands, one DIS (a table estimated_dis in Figure 7) is estimated from Φ_{salary} with 144 derived DISs. In this case, the MLRG process terminates in the three steps below, and 6 rules in Figure 8 are estimated.

$$CR(1) \subset CR(2) \subset CR(3) = PR(3) \subset PR(2) \subset PR(1). \quad (3)$$

```
mysql> select * from nrdf1 where object=5;
+--------+--------+--------+------+
| object | attrib | value  | det  |
+--------+--------+--------+------+
|      5 | age    | young  |    1 |
|      5 | depart | first  |    3 |
|      5 | depart | second |    3 |
|      5 | depart | third  |    3 |
|      5 | salary | high   |    1 |
|      5 | smoke  | no     |    2 |
|      5 | smoke  | yes    |    2 |
+--------+--------+--------+------+
7 rows in set (0.00 sec)
```

Fig. 9. The NRDF format of the object 5.

In $CR(1)$, there are three certain rules in Figure 5, and new three rules are learned by the process in Figure 7. On the other hand, in $PR(1)$ there are eight possible rules in Figure 6, and two possible rules are removed by the process in Figure 7.

5 SQL Procedures in MLRG

We have implemented SQL procedures, step1, step2, step3, pstep, apri, and other translation procedures. The arguments in each procedure except the translation are ('decision_attribute', number_of_objects, support, accuracy).

5.1 NRDF Format

In data sets, we usually have the csv format. This is very familiar, however the name of the attribute and the number of all attributes may be different in each data set. For handling various types of data sets uniquely, it is useful to employ another unified format. Otherwise, the program is depending upon the number of the attributes and the name of the attribute.

We employ the NRDF format [18], which is the extended RDF (resource description framework) format. The RDF format may be called as the EAV (entity-attribute-value) format [6, 19]. The NRDF format employs 4 attributes, *object*, *attrib*, *value*, and *det*. Figure 9 shows a part of the NRDF format of Φ_{salary}. In order to specify non-deterministic information, we added the 4th column *det*. The value of *det* means the number of possible values. If *det*=1, this means that the value is deterministic. Otherwise, we know the value is non-deterministic, and see the number of values by *det*.

(Merit 1 of using the NRDF format) *Even though we need to prepare a translation program to each csv file, we can handle any data set uniformly after this translation.*

```
mysql> select * from cll_revise where object=5;
+-----+--------+---------+--------+------+---------+--------+------+
| num | object | attrib1 | value1 | det1 | attrib2 | value2 | det2 |
+-----+--------+---------+--------+------+---------+--------+------+
|   7 |      5 | smoke   | no     |    2 | salary  | high   |    1 |
|   8 |      5 | depart  | second |    3 | salary  | high   |    1 |
+-----+--------+---------+--------+------+---------+--------+------+
2 rows in set (0.00 sec)

mysql> select * from `table 1` where object=5;
+--------+-------+----------------------+----------+--------+
| object | age   | depart               | smoke    | salary |
+--------+-------+----------------------+----------+--------+
|      5 | young | [first,second,third] | [yes,no] | high   |
+--------+-------+----------------------+----------+--------+
1 row in set (0.00 sec)

mysql> select * from nrdf2 where object=5;
+--------+--------+--------+-----+
| object | attrib | value  | det |
+--------+--------+--------+-----+
|      5 | age    | young  |   1 |
|      5 | depart | second |   1 |
|      5 | salary | high   |   1 |
|      5 | smoke  | no     |   1 |
+--------+--------+--------+-----+
4 rows in set (0.00 sec)
```

Fig. 10. Some tables generated by the step1 command and the revised table nrdf2.

This NRDF format is useful for managing MLRG process in Figure 2. Namely, we at first prepare a table nrdf1 in the NRDF format, and we sequentially revise the tables to nrdf2, nrdf3, $\cdots$ in each step. If det=1 holds for each tuple, each value for non-deterministic information is estimated, and we stop the process of MLRG.

(Merit 2 of using the NRDF format) *By using the tables $nrdf_i$ in the NRDF format, we can control the process of MLRG.*

5.2 SQL Procedure step1

The role of step1 is below:
(step1-1) An execution of certain rule generation by using the table nrdf1, and a generation of some data tables.
(step1-2) A generation of the table nrdf2 from the table nrdf1.

After certain rule generation in step1-1, we have some tables. The step1 command copies nrdf1 to nrdf2, and employs two strategies, namely the positive unification strategy and the contradiction prevention strategy, for revising nrdf2.

We focus on the revision on the object 5 in Figure 10. This is an example of Strategy 1, which tries to cause the higher ordered certain rules as many as

```
mysql> select * from nrdf1 where object=2;
+--------+--------+--------+------+
| object | attrib | value  | det  |
+--------+--------+--------+------+
|      2 | age    | senior |    2 |
|      2 | age    | young  |    2 |
|      2 | depart | first  |    3 |
|      2 | depart | second |    3 |
|      2 | depart | third  |    3 |
|      2 | salary | low    |    1 |
|      2 | smoke  | yes    |    1 |
+--------+--------+--------+------+
7 rows in set (0.00 sec)

mysql> select * from nrdf2 where object=2;
+--------+--------+-------+-----+
| object | attrib | value | det |
+--------+--------+-------+-----+
|      2 | age    | young |   1 |
|      2 | depart | first |   2 |
|      2 | depart | third |   2 |
|      2 | salary | low   |   1 |
|      2 | smoke  | yes   |   1 |
+--------+--------+-------+-----+
5 rows in set (0.00 sec)
```

Fig. 11. The revision of the object 2 in the table nrdf2.

possible. There are two certain rules with one condition related to the object 5 (a table c11_revise). As for [smoke,no] in the object 5, the attribute value is not fixed, since det1=2. So, the step1 command removes the tuples (5,smoke,yes,2) and (5,smoke,no,2) from nrdf2, and newly adds (5,smoke,no,1) to nrdf2. As for [depart,second], the step1 command does the same procedure. Each value of the object 5 is fixed in the table nrdf2 in Figure 10.

Then, we show an example of Strategy 2, which tries to reduce the contradiction to the higher ordered certain rules. This Strategy 2 is applied after the application of the procedure on Strategy 1. In an object x, if the condition part in x matches a certain rule τ and det>1, we know the tuple of x contradicts τ, because the revision by Strategy 1 is finished. (If Strategy 1 was applied, every det is changed to det=1.) In this case, we fix the attribute values so as not to contradict τ. In Figure 5 and Figure 11, the certain rule with one condition [depart,second]⇒ [salary,high] in $CR(1)$ contradicts [depart,second]⇒ [salary,low] in the object 2. So, the step1 command removes (2,depart,second,3) from nrdf2, and revises other two tuples to (2,depart,first,2) and (2,depart,third,2). Similarly, since the certain rule [age,senior]⇒ [salary,high] in $CR(1)$ contradicts the implication [age,senior]⇒ [salary,low] in the object 2, the step1 command adds (2,age,young,1) to nrdf2 after removing (2,age,young,2) and (2,age,senior,2).

5.3 SQL Procedure pstep

In MLRG, we sequentially reduce the threshold values for obtaining new certain rules with one condition, and we change the table nrdf_n to the next table nrdf_{n+1}. However, each certain rule is defined as an implication of a definite object [12, 13], so some parts of non-deterministic information may not be changed, even if we employ the lower threshold values. For solving this problem, we define a procedure pstep. The role of the procedure pstep is below:
(pstep-1) An execution of possible rule generation in RNIA, and a generation of some data tables.
(pstep-2) A generation of the table pnrdf from the current table nrdf_n.

Since a possible rule is defined as an implication of any object [12,13], we usually have the table nrdf_{n+1}, where det=1 for any object, after executing the pstep procedure. For example, the step1 command employs three certain rules in Figure 5 for revising the table nrdf1. On the other hand, the pstep command does eight possible rules in Figure 6 for revising the table nrdf1. Actually, the estimated_dis in Figure 7 was obtained after applying the pstep command in the first step. So, we can intentionally terminate MLRG process by using the procedure pstep. However, the application of the pstep command means the use of possible information from NIS. There will be the volatility risk of the possible rules. We may have inconsistent possible rules like $p \Rightarrow q_1$ and $p \Rightarrow q_2$. Thus, we should consider the application of the procedure pstep after the procedure step2 or step3.

5.4 SQL Procedure apri

The procedure apri simulates the Apriori algorithm in DISs. The following is the overview of the series of the SQL procedures in the implemented procedure apri.

```
delimiter //
create procedure apri
begin
create table condi(); /* Generate a table of the specified conditions,
             decision attribute, objects, α, β */
create table deci(); /* Generate a table of the decision */
create table con1(); /* Generate a table of the condition */
create table rule1(); /* Generate a table of the rules satisfying
             support ≥ α and accuracy ≥ β */
create table rest1(); /* Generate a table of the rules satisfying
             support ≥ α and accuracy < β */
create table con20(),create table con21(),create table con2();
             /* Generate a table of the condition part,
             whose element is p1 ∧ p2 from rest1 */
create table con2_infc0(),create table con2_infc();
             /* Generate a table of inf (=sup) information */
create table rule21(),create table rule2(); /* Generate a table of
```

```
            rule2 satisfying support ≥ α and accuracy ≥ β */
create table rest2(); /* Generate a table of rest2 satisfying
            support ≥ α and accuracy < β */
create table con30(),create table con31(),create table con3();
            /* Generate a table of the condition part,
            whose element is p1 ∧ p2 ∧ p3 from rest2 */
create table con3_inf0(),create table con3_infc();
            /* Generate a table of inf (=sup) information */
create table rule31(),create table rule3(); /* Generate a table of
            rule3 satisfying support ≥ α and accuracy ≥ β */
end //
```

The procedure apri generates rules in the forms of $p_1 \Rightarrow q$, $p_1 \wedge p_2 \Rightarrow q$, and $p_1 \wedge p_2 \wedge p_3 \Rightarrow q$. The details of this apri are in the web page [17]. In Figure 8, the procedure apri generated 6 rules in the form of $p_1 \Rightarrow q$ from the estimated DIS ψ^{actual}.

5.5 Implementation of MLRG Procedures in SQL

Since each procedure is implemented as a stored procedure in SQL, each procedure will be applicable to any SQL system. The text file size of all procedures is about 53KB, and we employed windows desktop PC (3.30GHz).

Figure 12 shows the MLRG process on the Congressional Voting data set in UCI machine learning repository. It consists of 435 objects, the decision attribute 'a1', 16 condition attributes, and 392 missing values ?. The decision attribute value is either democrat or republic. Each attribute value for other attribute is either yes or no, so we replaced each missing values with a set $\{democrat, republic\}$ or a set $\{yes, no\}$, and generated NIS $\Phi_{congress}$. The number of $DD(\Phi_{congress})$ is $2^{392} \fallingdotseq 10^{100}$.

The step1 command at first generated $CR(1)$ satisfying $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in each of about 10^{100} derived DISs. In this step, about a half of the unfixed values are fixed. The number of the unfixed values is 199 (=398/2) in the middle of Figure 12. Then, the step2 command generated certain rules with one condition satisfying $support(\tau) \geq 0.1$ and $accuracy(\tau) \geq 0.3$ in each of all derived DISs. In this step, each missing value is fixed, and one DIS ψ^{actual} is estimated. The details of the execution logs including the logs of Mammographic data set are in [17].

6 Concluding Remarks and Discussion

This paper briefly described the framework of MLGR. In real life, if we recognize the proper and attractive property (namely, certain rules), we will take an action (namely, the recovery of non-deterministic information) to support the recognized proper and attractive property as much as possible. Intuitively, MLRG takes such a strategy, and we see the estimated DIS ψ^{actual} and the rules in ψ^{actual} will be reasonable.

```
mysql> select count(*) from nrdf1 where det>1;
+----------+
| count(*) |
+----------+
|      784 |
+----------+
1 row in set (0.00 sec)

mysql> call step1('a1',435,0.3,0.6);
Query OK, 0 rows affected (1 min 2.98 sec)

mysql> select count(*) from nrdf2 where det>1;
+----------+
| count(*) |
+----------+
|      398 |
+----------+
1 row in set (0.00 sec)

mysql> call step2('a1',435,0.1,0.3);
Query OK, 0 rows affected (1 min 39.94 sec)

mysql> select count(*) from nrdf3 where det>1;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)
```

Fig. 12. The MLRG process of the Congressional Voting data set.

We have also implemented a software tool on NIS-Apriori based rule generation in SQL, and applied it to MLGR. We know data recovery by using the functional dependency in a standard table. In a table with uncertainty, we generate the ordered certain rules by the *minacc* value and the *minsupp* value, and make use of the concept on the maximum likelihood estimation in statistics. Then, the plausible value for non-deterministic information is estimated.

As for this prototype, we have the following consideration.
(1) Since SQL has the high versatility, NIS-Apriori in SQL and MLRG in SQL will offer the useful environment for analyzing tables with uncertainty.
(2) It is necessary to clarify the relation between the threshold values and the estimated DIS. If we specify the higher threshold values in the procedure step1, we have less certain rules with one condition and we may need several steps for terminating MLRG process. On the other hand, if we specify the lower threshold values, we have lots of certain rules with one condition and MLRG process will easily terminate. We need to consider what is the proper threshold values for MLRG process. Furthermore, if we employ the procedure pstep with

the lower threshold values, MLRG process will terminate in the first step. Each non-deterministic information is estimated by using the ordered possible rules. However in possible rule generation, we consider only one DIS from several possible tables, so there is a very big volatility. We may have two contradictory rules like p⇒q1 and p⇒q2. So, there is a tradeoff between the steps of the termination and the quality of the estimated DIS. We have not touched this issue yet.
(3) In this prototype, we faithfully simulated the MLRG process, so the procedures in SQL may have meaningless parts. It is necessary to brush up this software tool.

Acknowledgment: The authors would be grateful to the anonymous referees for their useful comments. This work is supported by JSPS (Japan Society for the Promotion of Science) KAKENHI Grant Number 26330277.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. *Proc. VLDB'94*, Morgan Kaufmann, 487–499 (1994)
2. Aldrich, J.: R. A. Fisher and the making of maximum likelihood 1912–1922. *Statistical Science* 12(3), 162–176 (1997)
3. Clark, P., Grzymala-Busse, J.: Mining incomplete data with many attribute-concept values and "do not care" conditions. *Proc. IEEE Big Data 2015*, 1597–1602 (2015)
4. Frank, A., Asuncion, A.: UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science (2010) `http://mlearn.ics.uci.edu/MLRepository.html`
5. Grzymala-Busse, J.: Data with missing attribute values: Generalization of indiscernibility relation and rule induction. *Transactions on Rough Sets* 1, 78–95 (2004)
6. Kowalski, M., Stawicki, S.: SQL-based heuristics for selected KDD tasks over large data sets. *Proc. FedCSIS 2012*, 303–310 (2012)
7. Kryszkiewicz, M.: Rules in incomplete information systems, *Information Sciences* 113(3-4), 271–292 (1999)
8. Lipski, W.: On databases with incomplete information, *Journal of the ACM* 28(1), 41–70 (1981)
9. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. *Theoretical Computer Science* 29(1-2), 27–39 (1984)
10. Pawlak, Z.: Systemy Informacyjne: Podstawy Teoretyczne (in Polish) WNT (1983)
11. Sahri, Z., Yusof, R., Watada, J.: FINNIM: Iterative imputation of missing values in dissolved gas analysis dataset. *IEEE Transactions on Industrial Informatics* 10(4), 2093–2102 (2014)
12. Sakai, H., et al.: Rules and apriori algorithm in non-deterministic information systems. *Transactions on Rough Sets* 9, 328–350 (2008)
13. Sakai, H., Wu, M., Nakata, M.: Apriori-based rule generation in incomplete information databases and non-deterministic information systems. *Fundamenta Informaticae* 130(3), 343–376 (2014)

14. Sakai, H., Wu M.: The completeness of NIS-Apriori algorithm and a software tool getRNIA, in *Proc. Int'l. Conf. on AAI2014* (Mori, M. ed.), IEEE, 115–121 (2014)
15. Sakai, H., Liu, C.: A consideration on learning by rule generation from tables with missing values, in *Proc. Int'l. Conf. on AAI2015* (Mine, T. ed.), IEEE, 183–188 (2015)
16. Sakai, H., Liu, C., Zhu, X., Nakata, M.: On NIS-Apriori based data mining in SQL. in *Proc. Int'l. Conf. on IJCRS* (Victor Flores et al. eds.), Springer, LNCS 9920, 514-524 (2016)
17. Sakai, H.: Execution logs by RNIA software tools (2016) `http://www.mns.kyutech.ac.jp/~sakai/RNIA`
18. Ślęzak, D., Sakai, H.: Automatic extraction of decision rules from non-deterministic data systems: Theoretical foundations and SQL-based implementation. *DTA2009* Springer CCIS Vol.64, 151–162 (2009)
19. Swieboda, W., Nguyen, S.: Rough set methods for large and spare data in EAV format. *Proc. IEEE RIVF 2012*, 1–6 (2012)
20. Yao, Y. Y.: Three-way decisions with probabilistic rough sets, *Information Sciences* 180, 314–353 (2010)