Scientiae. Mathematicae. Japonicae

RULE BASED DECISION SUPPORT IN TABLE DATA SETS WITH UNCERTAINTY AND ITS EXECUTION ENVIRONMENT

H. Sakai, K.Y. Shen, G.H. Tzeng, M. Nakata

Received

Abstract. A framework of decision support in table data sets with uncertainty is considered, and the prototype of its software tool is implemented in SQL. We follow the framework of the possible world semantics for table data sets with uncertainty, and two kinds of rules, i.e., the certain rules and the possible rules, are defined. This definition is simple and natural, but we are faced with the fact that the number of the possible worlds may exceed 10^{100}. Even in such huge number of possible worlds, the NIS-Apriori algorithm generates two kinds of rules, because this algorithm is independent from the number of the possible worlds due to the proved properties. The prototype system takes three phases for decision support, i.e.,
(i) the rule generation phase for knowing the general tendency of data sets,
(ii) the aggregation phase for decision support from the obtained rules,
(iii) the aggregation phase for decision support from data sets.
It is possible to employ (ii), if user's condition matches the condition in the obtained rules. Otherwise, it is necessary to employ (iii). The prototype system is applied to the Car Evaluation data set (a table data set without uncertainty) and the Congressional Voting data set (a table data set with uncertainty) in UCI machine learning repository. Since this prototype is implemented in SQL procedure, it will easily be applicable to any table data set on PC with SQL.

1 Introduction The data mining techniques afford to survey the instances in table data sets, and we can know the tendency and the property of data sets. Rule based decision support connected with such data mining techniques seems to be a very active research area now. Actually, we obtain more than 7700 papers for the keywords 'rule based decision support' in Scopus, whose composition ratio is 35% for computer science, 24% for engineering, 13% for medicine, 11% for mathematics, 5% for decision science, 5% for social science, 4% for business and management, 3% for biological science, etc. In these papers, fuzzy sets and rough sets seem very important. Some fuzzy frameworks are proposed in [6, 18], and the rough sets based framework named *Dominance based Rough Set Approach* (DRSA) is proposed in [4]. The authors in this paper also employ the rough sets and fuzzy sets based frameworks. The first and the fourth authors cope with rule generation, which they name *Rough Non-deterministic Information Analysis* (RNIA) [11, 12]. The second and the third authors cope with fuzzy sets and DRSA [15, 16]. This paper focuses on rule based decision support and its execution environment in SQL.

Even though there are a lot of frameworks on rule based decision support, our framework of RNIA preserves the logical aspect. Namely, the core rule generation algorithm named *NIS-Apriori* [12] is *sound* and *complete* for the rules based on the possible world semantics [13]. Therefore, the NIS-Apriori algorithm does not miss any rule for decision support. Generally, the number of the possible worlds becomes very huge, for example there are

2000 *Mathematics Subject Classification.* Primary 68T37, 03BXX; Secondary 68T15, 68P15.

Key words and phrases. decision support, association rules, NIS-Apriori algorithm, prototype in SQL, Uncertainty.

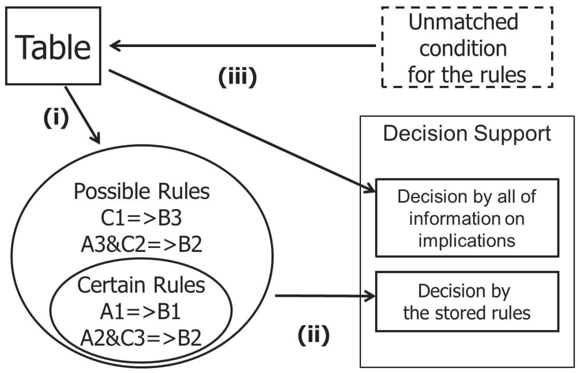

Figure 1: A chart of three phases for decision support environment in table data sets with uncertainty.

more than 10^{100} possible tables in the Mammographic data set in UCI machine learning repository [2]. Even though the definition of certain rules and possible rules is natural, it seemed hard to realize a rule generator for them. However, the NIS-Apriori algorithm affords a solution to this problem, namely this algorithm is independent from the number of the possible worlds [11, 12]. Without such property, it will be hard to address rules defined by the possible world semantics.

The main issue in this paper is to propose three phases (i), (ii), and (iii) in Figure 1.
(i) *The rule generation phase*: For two threshold values α and β, the prototype system generates rules. We will know the tendency and the character of data sets. This phase handling certain rules and possible rules based on the possible world semantics is first realized by the NIS-Apriori algorithm.
(ii) *The search phase for the obtained rules*: For the users' specified condition part $\wedge_i Con_i$, the obtained rules $\tau_k : \wedge_j Con_j \Rightarrow Dec_k$ ($\{Con_j\} \subseteq \{Con_i\}$) are examined, and triplets (Dec_k, $support(\tau_k)$, $accuracy(\tau_k)$) are generated. Users decide one decision Dec_k from the generated triplets by using $support(\tau_k)$ and $accuracy(\tau_k)$ ($support(\tau_k)$ and $accuracy(\tau_k)$ are given in the subsequent section).
(iii) *The search phase for the data set*: If there is no rule with the same condition part, all implications with the specified condition part are searched in the data set. The prototype system similarly generates triplets (Dec_k, $support(\tau_k)$, $accuracy(\tau_k)$), and users decide one decision Dec_k.

Remark 1 *In decision support, we see that the validity of the implication τ_k is measured by two values $support(\tau_k)$ and $accuracy(\tau_k)$. So, our environment tries to afford all of information about implications $\tau_k : \wedge_i Con_i \Rightarrow Dec_k$, i.e., $support(\tau_k)$ and $accuracy(\tau_k)$. We do not strongly touch about what is the final decision, which should be fixed by users.*

Remark 2 *If the phases (ii) is applicable to the specified condition part, the execution is much faster than the execution in the phase (iii). So, the application of the phase (ii) will be useful, however there may not be any rule matching the specified condition part. Thus, it is necessary to prepare the phase (iii). Even though the phase (iii) may take much execution time, this phase responds all implications with the specified condition part.*

Remark 3 *Let us consider the following three cases in Figure 1.*
(1) Let us suppose we need to have one decision under the condition $A1$*. Then, we employ the implication* $\tau : A1 \Rightarrow B1$ *(certain rule, reliable), and have the decision* $B1$*. The validity of* $B1$ *depends upon the validity of* τ*. This is an example of the phase (ii).*
(2) Let us suppose we need to have one decision under the condition $A1\&C3$*. Then, there is no rule with the condition* $A1\&C3$*. However, we have the following equation,*

$$(A1 \wedge C3 \Rightarrow Dec) = (\neg(A1 \wedge C3) \vee Dec) = (\neg A1 \vee \neg C3 \vee Dec) =$$
$$((\neg A1 \vee Dec) \vee (\neg C3 \vee Dec)) = ((A1 \Rightarrow Dec) \vee (C3 \Rightarrow Dec)).$$

Since we can conclude $A1 \wedge C3 \Rightarrow B1$ *from* $A1 \Rightarrow B1$*, we will have the decision* $B1$*. We usually say that* $A1 \wedge C3 \Rightarrow B1$ *is a redundant implication for* $A1 \Rightarrow B1$*. This is also an example of the phase (ii).*
(3) Since the phase (i) takes much execution time, we should not employ the phase (i) frequently. For the Chess data set (3196 instances, 36 attributes) in UCI machine learning repository [2], we obtained 6 rules for support ≥ 0.25 *and accuracy* ≥ 0.6 *by the implemented procedure apri, but it took more than 1 hour. So, in the phase (i), we preliminary employ the weak condition for rule generation, i.e., we employ the lower values of* α *and* β*. Even though we may have a large number of rules, the phase (ii) is effectively applied.*

This paper is organized as follows: Section 2 describes rule based decision support in table data sets without uncertainty and that in table data sets with uncertainty. Section 3 investigates some procedures in SQL, and Section 4 concludes this paper.

2 Rule Based Decision Support in Table Data Sets This section focuses on decision support in table data sets without uncertainty and decision support in table data sets with uncertainty.

2.1 Rules from the Table Data Sets without Uncertainty In order to consider rules from table data sets without uncertainty, we employ the Car Evaluation data set in UCI machine learning repository [2].

```
mysql> select * from `table 1` where object<5;
+--------+--------+-------+-------+---------+---------+--------+---------------+
| object | buying | maint | doors | persons | lugboot | safety | acceptability |
+--------+--------+-------+-------+---------+---------+--------+---------------+
|      1 | vhigh  | vhigh | 2     | 2       | small   | low    | unacc         |
|      2 | vhigh  | vhigh | 2     | 2       | small   | med    | unacc         |
|      3 | vhigh  | vhigh | 2     | 2       | small   | high   | unacc         |
|      4 | vhigh  | vhigh | 2     | 2       | med     | low    | unacc         |
+--------+--------+-------+-------+---------+---------+--------+---------------+
4 rows in set (0.02 sec)

mysql> select * from `table 1` where object<240 and acceptability='acc';
+--------+--------+-------+-------+---------+---------+--------+---------------+
| object | buying | maint | doors | persons | lugboot | safety | acceptability |
+--------+--------+-------+-------+---------+---------+--------+---------------+
|    228 | vhigh  | med   | 2     | 4       | small   | high   | acc           |
|    231 | vhigh  | med   | 2     | 4       | med     | high   | acc           |
|    233 | vhigh  | med   | 2     | 4       | big     | med    | acc           |
|    234 | vhigh  | med   | 2     | 4       | big     | high   | acc           |
+--------+--------+-------+-------+---------+---------+--------+---------------+
4 rows in set (0.00 sec)
```

Figure 2: Some parts of the Car Evaluation data set.

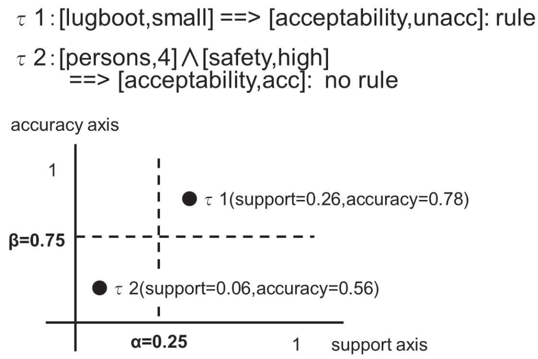

Figure 3: Rules plotted in the plane by the condition support≥0.25 and accuracy≥0.75.

This table data set consists of 1728 *objects* (instances), 6 *attributes*: *buying*, *maint(enance)*, *doors*, *persons*, *lugboot*, *safety*, 3 or 4 *attribute values* for each attribute, one decision attribute *acceptability* with 4 attribute values, *unacc*, *acc*, *good*, *vgood* in Figure 2. Each attribute value can be seen as a categorized value, and it may be hard to consider means nor variance in statistics. In such table data sets, we consider rule based decision support.

A pair $[A, val_A]$ of an attribute A and its attribute value val_A is called a *descriptor*. For a decision attribute *Dec* and a set *CON* of the attributes, we see 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 criterion values [10].

(1) For two threshold values $0 < \alpha, \beta \leq 1.0$,
$support(\tau)(= N(\wedge_{A\in CON}[A, val_A] \wedge [Dec, val])/|OB|) \geq \alpha$,
$accuracy(\tau)(= N(\wedge_{A\in CON}[A, val_A] \wedge [Dec, val])/N(\wedge_{A\in CON}[A, val_A])) \geq \beta$,
Here, $N(*)$ means the number of the objects satisfying the formula $*$, and OB means a set of all objects. We define $support(\tau) = accuracy(\tau) = 0$, if $N(\wedge_{A\in CON}[A, val_A]) = 0$.

For an implication $\tau_1 : [lugboot, small] \Rightarrow [acceptability, unacc]$ in Figure 3,

$$
\begin{aligned}
&N(\tau_1) = 450,\ N([lugboot, small]) = 576,\\
&support(\tau_1) = 450/1728 \fallingdotseq 0.26,\ accuracy(\tau_1) = 450/576 \fallingdotseq 0.78.
\end{aligned} \tag{2}
$$

Similarly, for an implication $\tau_2 : [persons, 4] \wedge [safety, high] \Rightarrow [acceptability, acc]$,

$$
\begin{aligned}
&N(\tau_2) = 108,\ N([persons, 4] \wedge [safety, high]) = 192,\\
&support(\tau_2) = 108/1728 \fallingdotseq 0.06,\ accuracy(\tau_2) = 108/192 \fallingdotseq 0.56.
\end{aligned} \tag{3}
$$

The $support(\tau)$ value means the occurrence ratio of the implication τ. If τ occurs much more time, this τ is much more reliable. On the other hand, the $accuracy(\tau)$ value means the consistency ratio of the implication τ. If the $accuracy(\tau)$ value is higher, this τ is more reliable.

In Figure 3, we see τ_1 and τ_2 are located in the points ($support(\tau)$,$accuracy(\tau)$) by the support and the accuracy axises. We usually fix two threshold values α and β for defining rules in each table data set. In Figure 3, we give α=0.25 and β=0.75, and we see τ_1 is a rule, and τ_2 is not a rule.

2.2 Decision Support in Table Data Sets without Uncertainty If we need to have a decision for the condition $[lugboot, small]$ in the Car Evaluation data set, we make use of the rule τ_1 and have a triplet $([acceptability, unacc], support = 0.26, accuracy = 0.78)$. Thus, we will conclude this car is unacceptable. This inference takes the phases (i) and (ii) in Figure 1.

On the other hand, we consider the condition $[lugboot, medium]$. In this case, we do not have any rule matching this condition and take the phase (iii) in Figure 1. Actually, we have Figure 4 for the condition $[lugboot, medium]$. Probably, we will conclude that this car is also *unacc(eptable)* due to the third implication in Figure 4. In Figure 4, the implemented command $srdf\_con1$ searches the Car Evaluation data set, and it took 0.33 (sec).

```
mysql> call srdf_con1('acceptability',1728,'lugboot','med');
Query OK, 0 rows affected (0.33 sec)

mysql> select * from srdf_con1;
+---------+------+---------------+------------+---------+----------+
| att1    | val1 | deci          | deci_value | support | accuracy |
+---------+------+---------------+------------+---------+----------+
| lugboot | med  | acceptability | acc        |   0.078 |    0.234 |
| lugboot | med  | acceptability | good       |   0.014 |    0.042 |
| lugboot | med  | acceptability | unacc      |   0.227 |    0.681 |
| lugboot | med  | acceptability | vgood      |   0.014 |    0.043 |
+---------+------+---------------+------------+---------+----------+
4 rows in set (0.00 sec)
```

Figure 4: All possible implications with the condition $[lugboot, med]$.

Like this, the prototype system responds all of information w.r.t. $\tau_k : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val_k]$.

2.3 Rules from the Table Data Sets with Uncertainty In order to consider rules from table data sets with uncertainty, we employ the Congressional Voting data set in UCI machine learning repository [2].

```
mysql> select a1,a2,a3,a4,a5,a6,a7,a12,a16,a17 from `table 1` where object < 6;
+------+------+------+------+------+------+------+------+------+------+
| a1   | a2   | a3   | a4   | a5   | a6   | a7   | a12  | a16  | a17  |
+------+------+------+------+------+------+------+------+------+------+
| rep  | n    | y    | n    | y    | y    | y    | ?    | n    | y    |
| rep  | n    | y    | n    | y    | y    | y    | n    | n    | ?    |
| dem  | ?    | y    | y    | ?    | y    | y    | y    | n    | n    |
| dem  | n    | y    | y    | n    | ?    | y    | y    | n    | y    |
| dem  | y    | y    | y    | n    | y    | y    | y    | y    | y    |
+------+------+------+------+------+------+------+------+------+------+
5 rows in set (0.00 sec)
```

Figure 5: Some parts of the Congressional Voting data set.

This table data set consists of 435 objects (instances), 16 attributes: a_2, a_3, $\cdots$, a_{17}, two attribute values *y(es)* or *n(o)* for each attribute, one decision attribute a_1 with two attribute values, *rep(ublic)* or *dem(octat)* in Figure 5. In the Congressional Voting data set, there are 329 missing values expressed by the ? symbol. Of course, rules depend upon the missing values, and it is necessary for handling rules in such table data sets [7, 8, 9]. We have dealt with this problem in RNIA.

We briefly review RNIA. In a table with missing values, we usually apply the discretization procedure, and we handle a finite number of the possible values. By replacing each ? symbol with a possible value, we have a table data set without uncertainty, which we name a *derived DIS* (*DIS: Deterministic Information System*). Let $DD(\Phi)$ denote the set of all derived DISs from Φ with missing values, and we may say Φ is a *NIS: Non-deterministic Information System*. In rule generation, we employ the usual definition of a rule in DIS [10], and extend it to a certain rule and a possible rule in NIS below [11, 12]:

(A certain rule in NIS) An implication τ is a *certain rule*, if τ is a rule in each derived DIS for given α and β.
(A possible rule in NIS) An implication τ is a *possible rule*, if τ is a rule in at least one derived DIS for given α and β.

If τ is a certain rule, we can conclude τ is also a rule in the unknown actual DIS ψ^{actual}. (We see there is one derived DIS $\psi^{actual} \in DD(\Phi)$ which contains the actual values.) This property is also described in Lipski's incomplete information databases [5]. In DIS, the same set of rules are obtained by two definitions, so two definitions will be a natural extension from rules in DIS. However, the number of $DD(\Phi)$ increases exponentially, and there are more than 10^{100} derived DISs for the Congressional Voting data set. It will be hard to examine the certain rules and the possible rules by checking each derived DIS sequentially. For this problem, we afford a solution by showing some properties on rules [11, 12].

(4)

(Property 1) For NIS Φ and any implication τ, there is a derived DIS $\psi_{min} \in DD(\Phi)$ such that
$minsupp(\tau)$(defined by $support(\tau)$ in ψ_{min}) $= \min_{\psi \in DD(\Phi)}\{support(\tau)$ in $\psi\}$,
$minacc(\tau)$(defined by $accuracy(\tau)$ in ψ_{min}) $= \min_{\psi \in DD(\Phi)}\{accuracy(\tau)$ in $\psi\}$.

(Property 2) For NIS Φ and any implication τ, there is a derived DIS $\psi_{max} \in DD(\Phi)$ such that
$maxsupp(\tau)$(defined by $support(\tau)$ in ψ_{max}) $= \max_{\psi \in DD(\Phi)}\{support(\tau)$ in $\psi\}$,
$maxacc(\tau)$(defined by $accuracy(\tau)$ in ψ_{max}) $= \max_{\psi \in DD(\Phi)}\{accuracy(\tau)$ in $\psi\}$.

(Property 3) There is a calculation method of $support(\tau)$ and $accuracy(\tau)$, and this method is independent from the number of $DD(\Phi)$. The details are in [12].

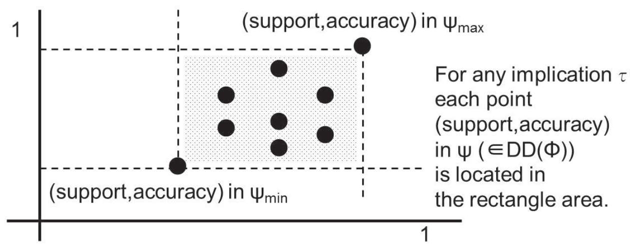

Figure 6: Each point for an implication τ is located in the rectangle area.

```
mysql> select * from c1_rule where att1>'a2' and att1<'a7';
+------+------+------+------------+---------+--------+
| att1 | val1 | deci | deci_value | minsupp | minacc |
+------+------+------+------------+---------+--------+
| a4   | n    | a1   | rep        |   0.326 |  0.798 |
| a4   | y    | a1   | dem        |   0.531 |  0.899 |
| a5   | n    | a1   | dem        |   0.563 |  0.980 |
| a5   | y    | a1   | rep        |   0.375 |  0.881 |
| a6   | n    | a1   | dem        |   0.460 |  0.948 |
| a6   | y    | a1   | rep        |   0.361 |  0.701 |
+------+------+------+------------+---------+--------+
6 rows in set (0.00 sec)
```

Figure 7: A part of the obtained certain rules satisfying $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in the Congressional Voting data set.

Based on the above properties, we have the chart in Figure 6. In Figure 3, the point ($support(\tau)$,$accuracy(\tau)$) in DIS is unique, but each point in $\psi \in DD(\Phi)$ is located in the rectangle area in Figure 6. There are more than 10^{100} points in the rectangle area, however we can have two points by ψ_{min} and ψ_{max} independently from the number of $DD(\Phi)$. Furthermore, we have the next properties for the certain rules and the possible rules [11, 12].

(5)

(Property 4) For NIS Φ and any implication τ, τ is a certain rule if and only if $minsupp(\tau) \geq \alpha$ and $minacc(\tau) \geq \beta$.

(Property 5) For NIS Φ and any implication τ, τ is a possible rule if and only if $maxsuppt(\tau) \geq \alpha$ and $maxacc(\tau) \geq \beta$.

We added the above two properties to the *Apriori* algorithm [1], which is the representative algorithm in data mining, and proposed the *NIS-Apriori* algorithm [11, 12]. We refer to the prototype system in SQL powered by the NIS-Apriori algorithm in the next section.

2.4 Decision Support in Table Data Sets with Uncertainty In the Congressional Voting data set, we had 22 certain rules (with one descriptor in the condition part) for α=0.3 and β=0.6 in Figure 7. They satisfy $support(\tau) \geq 0.3$ and $accuracy(\tau) \geq 0.6$ in each of more than 10^{100} derived DISs. Especially, two certain rules $[a5, n] \Rightarrow [a1, dem(ocrat)]$ and $[a5, y] \Rightarrow [a1, rep(ublic)]$ are very strong. If we have a person's answer to the attribute $a5$, we will easily conclude his supporting party. This inference takes the phases (i) and (ii) in Figure 1. We also had 26 possible rules (with one descriptor in the condition part) and one possible rule (with two descriptors in the condition part) in Figure 8. If the condition does not match any certain rule, we may apply possible rules. Furthermore, if the condition does not much any rule, we have the phase (iii) in Figure 1.

For the implications $\tau : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val]$ and $\tau' : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val']$, if $maxsupp(\tau) \leq minsupp(\tau')$ and $maxacc(\tau) \leq minsupp(\tau')$ hold, we have $support(\tau) \leq support(\tau')$ and $accuracy(\tau) \leq accuracy(\tau')$ for any DIS $\psi \in DD(\Phi)$ (Figure 9). So, we will certainly have the decision $[Dec, val']$ under the table data set with uncertainty. The concept in Figure 9 will be the extension from the concepts in Figure 3 and Figure 6.

3 Rule Based Decision Support System in SQL This section describes each phase in the prototype system. Each program is implemented in the SQL procedure.

```
mysql> select * from p2_rule;
+------------+------+------+------+------+------------+---------+--------+
| att1       | val1 | att2 | val2 | deci | deci_value | maxsupp | maxacc |
+------------+------+------+------+------+------------+---------+--------+
| a12        | n    | a7   | y    | a1   | rep        |   0.301 |  0.753 |
| end_attrib | NULL | NULL | NULL | NULL | NULL       |    NULL |   NULL |
+------------+------+------+------+------+------------+---------+--------+
2 rows in set (0.00 sec)
```

Figure 8: One possible rule with two descriptors in the condition part.

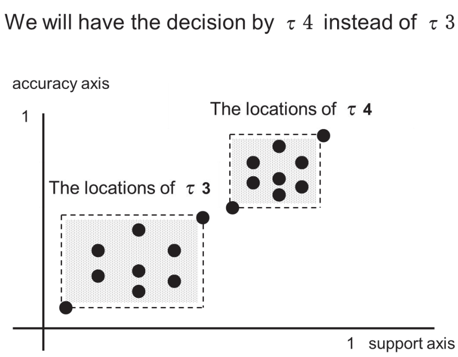

Figure 9: The locations of the implications plotted in the plane.

```
mysql> call car_rdf;
Query OK, 0 rows affected (1.58 sec)

mysql> select * from rdf where object=2;
+--------+---------------+-------+
| object | attrib        | value |
+--------+---------------+-------+
|      2 | acceptability | unacc |
|      2 | buying        | vhigh |
|      2 | doors         | 2     |
|      2 | lugboot       | small |
|      2 | maint         | vhigh |
|      2 | persons       | 2     |
|      2 | safety        | med   |
+--------+---------------+-------+
7 rows in set (0.00 sec)
```

Figure 10: The execution of car_rdf command and the generated rdf file from the Car Evaluation data set.

3.1 The Rule Generation Phase (i) in Figure 1: The Case of DISs In table data sets without uncertainty, we at first translate each csv file to the rdf format [17], and employ the Apriori algorithm for rule generation. In DISs, we implemented the following procedures in SQL.
(1) The procedure `File_name_rdf`: It translates a csv file to the rdf format file. (In Figure 10, `car_rdf` is executed.)

(2) The procedure `apri`: It generates tables rule1 (rules with one condition), rule2 (rules with two conditions), rule3 (rules with three conditions). (For the constraint *support* ≥ 0.25 and *accuracy* ≥ 0.7, the procedure `apri` generated three tables in 9.99 (sec) for the Car Evaluation data set, whose execution logs are in [14].)

In the rdf format, each table data is translated to a table of descriptors. In each table data set, the number of attributes and its attribute values are different, but we can uniformly handle any data set if the data set is in the rdf format. Without this property, we need to make a set of the SQL procedures for each table data set.

3.2 The Search Phase (ii) and (iii) in Figure 1: The Case of DISs Let us consider the case that we need to have a decision for a given condition. The procedures `srule_con1`, `srule_con2`, and `srule_con3` are implemented for searching lots of rules stored in tables. They are the commands for the phase (ii) in Figure 1. Figure 11 shows the execution of `srule_con1`.

```
mysql> call srule_con1('acceptability','persons','2');
Query OK, 1 row affected (0.14 sec)

mysql> select * from srule_con1;
+---------+------+---------------+-------+---------+----------+
| att1    | val1 | deci          | val   | support | accuracy |
+---------+------+---------------+-------+---------+----------+
| persons | 2    | acceptability | -     | 999.000 |  999.000 |
| persons | 2    | acceptability | unacc |   0.333 |    1.000 |
+---------+------+---------------+-------+---------+----------+
2 rows in set (0.00 sec)
```

Figure 11: The all searched rules from obtained rules for the condition $[persons, 2]$. The first line means the query and the number 999 is meaningless value. The second line is picked up from the obtained rules.

Based on Figure 11, we know all kind of information for the condition $[persons, 2]$. This search is restricted to the obtained table data, so it takes less execution time. However, if the condition does not match the obtained rules, we have no information for the condition. In order to handle such case, we consider the phase (iii) in Figure 1. Figure 4 shows the execution about the condition $[lugboot, medium]$. Even though this condition is not in the obtained rules, we will have a decision *unacc(eptable)* from Figure 4. This will be useful for decision support.

3.3 The Rule Generation Phase (i) in Figure 1: The Case of NISs In table data sets with uncertainty, we at first translate each csv file to the nrdf format [17], and employ the NIS-Apriori algorithm for rule generation. In NISs, we implemented the following procedures in SQL.

(1) The procedure `File_name_nrdf`: It translates the csv file with ? symbol and non-deterministic values to the nrdf format file.

(2) The procedure `step1`: It generates tables c1_rule (certain rules with one condition) and p1_rule (possible rules with one condition).

(3) The procedures `step2`, `step3`: They generate tables c2_rule (certain rules with two conditions), p2_rule (possible rules with two conditions), c3_rule (certain rules with three conditions), and p3_rule (possible rules with three conditions).

The execution logs of the Congressional Voting data set are in [14].

3.4 The Search Phase (ii) in Figure 1 for the Obtained Rules: The Case of NISs Let us consider the case that we need to have a decision for a given condition. The procedures `srule_con1`, `srule_con2`, and `srule_con3` are implemented for searching lots of rules stored in tables. Figure 12 shows the execution of `srule_con2`.

```
mysql> call snrule_con2('a1','a5','y','a9','n');
Query OK, 0 rows affected (0.25 sec)

mysql> select * from snrule_con2;
+-----------+------+------+------+------+------+------+---------+---------+---------+---------+
| type      | att1 | val1 | att2 | val2 | deci | val  | minsupp | minacc  | maxsupp | maxacc  |
+-----------+------+------+------+------+------+------+---------+---------+---------+---------+
| Condition | a5   | y    | a9   | n    | a1   | -    | 999.000 | 999.000 | 999.000 | 999.000 |
| Certain   | a5   | y    | NULL | NULL | a1   | rep  |   0.375 |   0.881 | 999.000 | 999.000 |
| Certain   | a9   | n    | NULL | NULL | a1   | rep  |   0.306 |   0.731 | 999.000 | 999.000 |
| Possible  | a5   | y    | NULL | NULL | a1   | rep  | 999.000 | 999.000 |   0.382 |   0.922 |
| Possible  | a9   | n    | NULL | NULL | a1   | rep  | 999.000 | 999.000 |   0.331 |   0.762 |
+-----------+------+------+------+------+------+------+---------+---------+---------+---------+
5 rows in set (0.00 sec)
```

Figure 12: The all searched rules from obtained rules for the condition $[a5, y] \wedge [a9, n]$. The number 999 is meaningless value.

Based on Figure 12, we know all of information for the condition $[a5, y] \wedge [a9, n]$. The implication $[a5, y] \wedge [a9, n] \Rightarrow [a1, rep]$ is redundant for two certain rules $[a5, y] \Rightarrow [a1, rep]$ and $[a9, n] \Rightarrow [a1, rep]$. In both cases, $[a5, y]$ and $[a9, n]$ conclude $[a1, rep]$. We will probably have the decision value *rep(ublic)* in Figure 12. This search is restricted to the obtained table data, so it takes less execution time. However, if the condition does not match the obtained rules, we have no information for the condition.

3.5 The Search Phase (iii) in Figure 1 for Data Sets: The Case of NISs Let us consider the case that we need to have a decision for a given condition. The procedures `snrdf_con1`, `snrdf_con2`, and `snrdf_con3` are implemented for searching tables with uncertainty. In this case, we employ the same condition $[a5, y] \wedge [a9, n]$ in Figure 12. Figure 13 shows the execution of `snrdf_con2`.

```
mysql> call snrdf_con2('a1',435,'a5','y','a9','n');
Query OK, 0 rows affected (4.94 sec)

mysql> select * from snrdf_con2;
+------+------+------+------+------+------+------+---------+---------+--------+--------+
| pkey | att1 | val1 | att2 | val2 | deci | val  | minsupp | maxsupp | minacc | maxacc |
+------+------+------+------+------+------+------+---------+---------+--------+--------+
|    1 | a5   | y    | a9   | n    | a1   | dem  |   0.025 |   0.032 |  0.071 |  0.096 |
|    2 | a5   | y    | a9   | n    | a1   | rep  |   0.303 |   0.329 |  0.904 |  0.929 |
+------+------+------+------+------+------+------+---------+---------+--------+--------+
2 rows in set (0.00 sec)
```

Figure 13: The all searched rules with the condition part $[a5, y] \wedge [a9, n]$ for the Congressional Voting data set.

Based on Figure 13, we know all of information for the condition $[a5, y] \wedge [a9, n]$. In this case, the procedure `snrdf_con2` searches the table nrdf, and it took 4.94 (sec). The execution time is about 20 times longer than that of `snrule_con2`. For two implications $\tau : [a5, y] \wedge [a9, n] \Rightarrow [a1, dem]$ and $\tau' : [a5, y] \wedge [a9, n] \Rightarrow [a1, rep]$, $maxsupp(\tau) \leq minsupp(\tau')$

and $maxacc(\tau) \leq minacc(\tau')$ hold. This is corresponding to the case in Figure 9, and we will easily have the decision value *rep(ublic)*.

3.6 The Validity of the Implementation We have previously implemented the NIS-Apriori algorithm in C and Prolog. This time, we employed SQL, because it will be difficult to use Prolog for the large size data sets. So, we had two independent systems, and we had the same results by the two systems. The execution logs are in [14].

4 Concluding Remarks and Discussion

This paper clarified rule based decision support on RNIA, and reported its prototype system. The definition of the certain rules and the possible rules seems natural, however there is less software tool for handling them, because the rules are defined by all derived DISs whose number may exceed 10^{100}. Without effective property, it will be hard to obtain rules. The NIS-Apriori algorithm affords a solution to this problem, and we implemented the prototype by NIS-Apriori in SQL. This algorithm takes the core part for handling the uncertainty, and we applied it to decision support environment.

Now, let us consider each phase of (i), (ii), and (iii). The phase (i) generates all certain rules and possible rules, which have the characteristic properties. However, it is time-consuming, so the frequent usage of the phase (i) will not be appropriate, and we need to employ the lower values of α and β. In this situation, we need the phase (ii) much more. If we have the large number of rules, the method to find the rules matching the condition may not be easy, and we realized some procedures in the phase (ii). The phase (iii) will be necessary to cope with the case that any rule does not match the condition. In table data sets, the implications are located in the plane like Figure 3. On the other hand in the tables with uncertainty, the implications are located in the plane like Figure 6 and Figure 9. The extension from Figure 3 to Figure 6 and Figure 9 is the key concept for considering decision support for the tables with uncertainty.

However, there may be the cases like Figure 14 and Figure 15, where it is difficult to have a decision even by using the phase (iii). In such cases, we will need other criteria like the type I error and the type II error in the statistical hypothesis tests instead of the support and accuracy values. Furthermore, it is important to have the theoretical property of the distribution of points (implications) with the same conditions and the different decision. Even though we consider that Figure 14 and Figure 15 express the rare cases, the next new challenges are open for them.

Acknowledgment: The authors would be grateful to the anonymous referees and Professor Junzo Watada (Waseda University and Universiti Teknologi PETRONAS, Malaysia) 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] 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`

[3] Grzymała-Busse, J.: Data with missing attribute values: Generalization of indiscernibility relation and rule induction. Transactions on Rough Sets 1, 78–95 (2004)

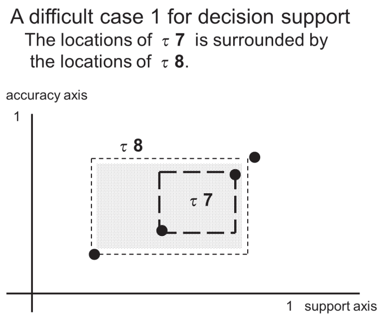

Figure 14: A difficult case 1 for having one decision from the implications with the same conditions and the different decision.

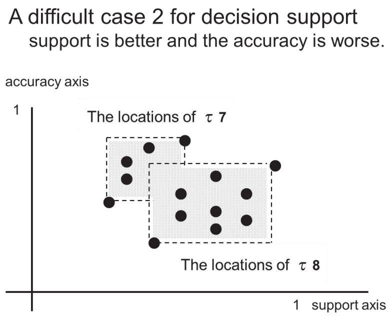

Figure 15: A difficult case 2 for having one decision from the implications with the same conditions and the different decision.

[4] Kadziński, M., Słowiński, R., Szeląg, M.: Dominance-based rough set approach to multiple criteria ranking with sorting-specific preference information. Studies in Computational Intelligence 606, 155-171 (2016)

[5] Lipski, W.: On semantic issues connected with incomplete information databases. ACM Transactions on Database Systems 4(3), 262–296 (1979)

[6] Minutolo, A., Esposito, M., De Pietro, G.: A fuzzy framework for encoding uncertainty in clinical decision-making. Knowledge-Based Systems 98, 95–116 (2016)

[7] Nakata, M., Sakai, H.: Twofold rough approximations under incomplete information. Int'l. J. General Systems 42(6), 546–571 (2013)

[8] Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science 29(1-2), 27–39 (1984)

[9] Pawlak, Z.: Systemy Informacyjne: Podstawy Teoretyczne (in Polish) WNT (1983)

[10] Pawlak, Z.: Rough Sets: Theoretical aspects of reasoning about data. Kluwer Academic Publishers (1991)

[11] Sakai, H., et al.: Rules and apriori algorithm in non-deterministic information systems. Transactions on Rough Sets 9, 328–350 (2008)

[12] 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)

[13] Sakai, H., Wu, M.: The completeness of NIS-Apriori algorithm and a software tool getRNIA. In: Proc. Int'l. Conf. on AAI2014, IEEE, 115–121 (2014).

[14] Sakai, H.: Software Tools for RNIA (Rough Non-deterministic Information Analysis) Web Page (2016) http://www.mns.kyutech.ac.jp/~sakai/RNIA/

[15] Shen, K.Y., Tzeng, G.H.: Contextual improvement planning by fuzzy-rough machine learning: A novel bipolar approach for business analytics. International Journal of Fuzzy Systems 18(6), 940–955 (2016)

[16] Shen, K.Y., Tzeng, G.H.: A novel bipolar MCDM model using rough sets and three-way decisions for decision aids. In: Proc. SCIS-ISIS, IEEE, 53–58 (2016)

[17] 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)

[18] Zarikas, V., Papageorgiou, E., Regner, P.: Bayesian network construction using a fuzzy rule based approach for medical decision support. Expert Systems 32(3), 344-369 (2016)