Granules for Association Rules and Decision Support in the getRNIA System

†Hiroshi Sakai,* †Mao Wu, †Naoto Yamaguchi, and ‡Michinori Nakata

†Graduate School of Engineering,
Kyushu Institute of Technology,
Tobata, Kitakyushu 804-8550, Japan

‡Faculty of Management and Information Science,
Josai International University,
Togane, Chiba 283-0002, Japan

Abstract

This paper proposes *granules for association rules* in *Deterministic Information Systems* (*DISs*) and *Non-deterministic Information Systems* (*NISs*). Granules for an association rule are defined for every implication, and give us a new methodology for knowledge discovery and decision support. We see that decision support based on a table under the condition P is to fix the decision Q by using the most proper association rule $P \Rightarrow Q$. We recently implemented a system *getRNIA* powered by granules for association rules. This paper describes how the *getRNIA* system deals with decision support under uncertainty, and shows some results of the experiment.

*Corresponding author. `Tel.:+81-93-884-3258; E-mail:sakai@mns.kyutech.ac.jp.`

Keywords: granules; rough sets; association rules; software; incomplete information; non-deterministic information

1 Introduction

Rough set theory offers a mathematical approach to vagueness and uncertainty, and the rough set-based concepts have been recognized to be very useful [5, 10, 11]. This theory usually handles tables with deterministic information, which we call *Deterministic Information Systems.* Furthermore, *Non-deterministic Information Systems* and *Incomplete Information Systems* (*IISs*) have also been proposed for handling information incompleteness in *DIS.* Several theoretical works have been reported [3, 4, 6, 7, 8, 9, 10].

We followed these previous works, and investigated rough sets in *DIS* and rough sets in *NIS.* We call a series of our works *Rough Non-deterministic Information Analysis* (*RNIA*) [14, 15, 16, 17, 18]. *RNIA* can handle tables with inexact data like non-deterministic information, missing values and interval values. This paper proposes granules for association rules in tables with such inexact data, and adds the functionality of decision making to the *getRNIA* system [2].

In decision making based on association rules, we have the following steps.
(Step 1) We at first generate rules from data sets, and store them. For the condition P, if a rule $P \Rightarrow Q$ is stored, we conclude Q is the decision for P.
(Step 2) If there is no rule with condition P, we generate each implication $P \Rightarrow Q_i$, and calculate its criterion value. By comparing criterion values, we select an implication $P \Rightarrow Q_j$ and the decision Q_j.
If the constraint for rule generation is weak, we have much more rules, and we may apply a stored rule to deciding Q. However, it will be hard to store all implications for any condition P. Therefore, (Step 1) is not enough for the

condition P, and we need to reconsider the above (Step 2) in *DIS* and *NIS*.

Recently, *granular computing* [12] is attracting researchers as the new paradigm of computing. Since the manipulation of granules in *NIS* is one of the interesting topics, we focus on granules for association rules, and we investigate the criterion value and its calculation in *NIS*.

This paper is organized as follows: Section 2 surveys rules in *DIS* and *NIS* as well as the framework of *RNIA*. Section 3 defines granules for association rules, and clarifies their properties. Section 4 applies granules for association rules to decision support in *DIS* and *NIS*, respectively. Section 5 describes the details of the *getRNIA* system [2] and the application to the *Mushroom* data set in UCI machine learning repository [19]. Finally, Section 6 concludes this paper.

2 Rules in DIS and NIS

This section surveys rules in *DIS* and *NIS*, and refers to *RNIA*.

2.1 Rules in DIS

Deterministic Information System (*DIS*) ψ is a quadruplet [10, 11],

$$\psi = (OB, AT, \{VAL_A \mid A \in AT\}, f),$$

where OB is a finite set whose elements are called *objects*, AT is a finite set whose elements are called *attributes*, VAL_A is a finite set whose elements are called *attribute values*, and f is a mapping below:

$$f : OB \times AT \to \cup_{A \in AT} VAL_A.$$

The value $f(x, A)$ means the attribute value for the object x and the attribute A. We usually consider a standard table instead of this quadruplet ψ. We call a pair $[attribute, value]$ a *descriptor*. Let us consider an attribute $Dec \in AT$ which we call the *decision attribute* and $CON \subseteq (AT \setminus \{Dec\})$ which we call (*a set of*) *condition attributes*. For descriptors $[A, value_A]$ $(A \in CON)$ and $[Dec, val]$, we call the following expression an *implication* τ in ψ:

$$\tau : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val].$$

Definition 1. *For τ in ψ, if $f(x, A)$=$value_A$ for every $A \in CON$, we say the object x supports the conjunction $\wedge_{A \in CON}[A, value_A]$. For τ, if $f(x, A)$=$value_A$ for every $A \in CON$ and $f(x, Dec)$=val, we say the object x supports τ. In order to specify the object x supporting τ, we employ the notation τ^x, and we say τ^x is an instance of τ in ψ.*

For examining τ in ψ, we evaluate its instance τ^x. In ψ, a (candidate of) *rule* is an implication τ such that an instance τ^x satisfies an appropriate constraint. In this paper, we employ the following constraint [1, 3, 10, 11] for rules:

For given threshold values α and β $(0 < \alpha, \beta \leq 1.0)$, and any $x \in OBJ(\tau) \neq \emptyset$,
$support(\tau^x) = |OBJ(\tau)|/|OB| \geq \alpha$,
$accuracy(\tau^x) = |OBJ(\tau)|/|OBJ(\wedge_{A \in CON}[A, value_A])| \geq \beta$,
Here, $OBJ(*)$ means the set of objects supporting the formula $*$, and
$|S|$ means the cardinality of the set S. We do not consider any τ with
$OBJ(\tau) = \emptyset$.

In our previous work, both τ and τ^x are not distinguished well, because they caused no problem in ψ. However in *NIS*, we need to pay attention to x in τ^x.

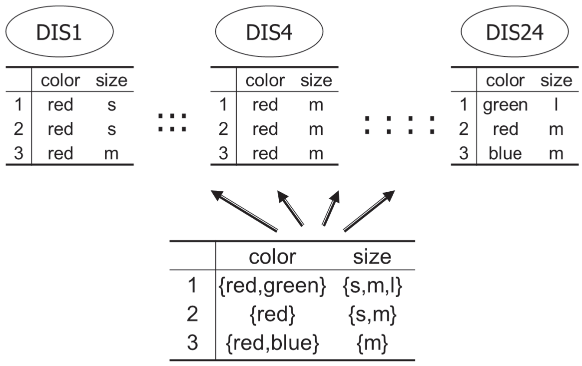

Figure 1: *NIS* Φ_1 and 24 derived *DISs*.

2.2 Rules in NIS

We give the definition of *NIS* Φ [9, 10, 11]:

$$\Phi = (OB, AT, \{VAL_A | A \in AT\}, g),$$
$$OB,\ AT,\ VAL_A \text{ are the same as in } \psi,$$
$$g : OB \times AT \rightarrow P(\cup_{A \in AT} VAL_A) \text{ (a power set)}.$$

In Φ, the attribute value for the object x and the attribute A is given as a set $g(x, A)$, and we see that an actual attribute value exists in the set. If we replace each set $g(x, A)$ with a value $v \in g(x, A)$, we obtain one ψ, which we call a *derived DIS* from Φ. Especially, we see ψ is Φ where every $g(x, A)$ is a singleton set. Figure 1 shows the relation between *NIS* Φ_1 and 24 derived *DISs*. In Φ, we also handle the following implication τ:

$$\tau : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val].$$

Definition 2. *For τ, if $value_A \in g(x, A)$ for every $A \in CON$, we say the object x supports the conjunction $\wedge_{A \in CON}[A, value_A]$ in Φ. If $value_A \in g(x, A)$ for every $A \in CON$ and $val \in g(x, Dec)$, we say the object x supports τ in Φ.*

Remark 1. *Let us consider an implication* $\tau : [color, red] \Rightarrow [size, m]$ *in Figure 1. This* τ *is supported by objects 1, 2, and 3 in* Φ_1*. Namely, we have three instances* τ^1*,* τ^2*, and* τ^3*. However, the first* τ^1 *is supported in 4 (=24/6) derived DISs, and* τ^2 *is supported in 12 (=24/2) derived DISs. Since we define the evaluation of instances in* Φ *over all derived DISs, the evaluation of* τ^1 *and the evaluation of* τ^2 *may not be the same. In* Φ*, if an instance* τ^x *satisfies a given constraint, we see this* τ^x *is an evidence causing the rule* τ*. There may be another instance* τ^y *not satisfying the given constraint. This does not occur in the evaluation in* ψ*, and this remark specifies the difference between rules in* ψ *and rules in* Φ*.*

For Φ, let $DD(\Phi)$ denote a set below:

$$DD(\Phi) = \{\psi \mid \psi \text{ is a derived } DIS \text{ from } \Phi\}.$$

Then, we can consider the following two types of rules with modalities in Φ.

(Certain rule) If an instance τ^x satisfies a given constraint in each $\psi \in DD(\Phi)$, we say τ is a *certain rule* in Φ.

(Possible rule) If an instance τ^x satisfies a given constraint in at least one $\psi \in DD(\Phi)$, we say τ is a *possible rule* in Φ.

2.3 Blocks inf and sup Defined in NIS

In rough set theory, we make use of equivalence classes and descriptors. Here, we give the next definition. This is an enhancement of *blocks* [3, 4]. For Φ with a function $g : OB \times AT \rightarrow P(\cup_{A \in AT} VAL_A)$ and each descriptor $[A, value_A]$ ($A \in AT$), we define two sets *inf* and *sup* [14, 15].

(1) For a descriptor $[A, value_A]$,

$$inf([A, value_A]) = \{x \in OB \mid g(x, A) = \{value_A\}\},$$
$$sup([A, value_A]) = \{x \in OB \mid value_A \in g(x, A)\}.$$

(2) For a conjunction of descriptors $\wedge_i [A_i, value_i]$,

$$inf(\wedge_i [A_i, value_i]) = \cap_i inf([A_i, value_i]),$$
$$sup(\wedge_i [A_i, value_i]) = \cap_i sup([A_i, value_i]).$$

In Φ_1 of Figure 1, we have the following:

$$inf([color, red]) = \{2\},\ sup([color, red]) = \{1, 2, 3\},$$
$$inf([size, s]) = \emptyset,\ sup([size, s]) = \{1, 2\},$$
$$inf([color, red] \wedge [size, s]) = \{2\} \cap \emptyset = \emptyset,$$
$$sup([color, red] \wedge [size, s]) = \{1, 2, 3\} \cap \{1, 2\} = \{1, 2\}.$$

Two sets *inf* and *sup* are the minimum and the maximum sets for the equivalence class defined by a descriptor or a conjunction of descriptors, respectively. We employed these *inf* and *sup* blocks and coped with certain and possible rule generation in Φ [14, 15].

3 Granules for Association Rules

This section proposes granules for association rules in *DIS* and *NIS*. We consider a set of all objects supporting $\tau : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val]$. If an object x supports τ, the object x must support $\wedge_{A \in CON}[A, value_A]$. Namely, we focus on $OBJ(\wedge_{A \in CON}[A, value_A])$ in ψ and $sup(\wedge_{A \in CON}[A, value_A])$ in Φ.

3.1 Granules for Association Rules in DIS

For $\tau : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val]$ in ψ, we may express it by $\tau : P \Rightarrow Q$ for simplicity. For every $x \in OBJ(P)$, clearly either $f(x, Dec)=val$ or $f(x, Dec)=val'$ ($val \neq val'$) holds. Therefore, we divide $OBJ(P)$ by $[Dec, val]$, and we define the following two sets:

$$
\begin{aligned}
① &= \{x \in OBJ(P) \mid x \text{ supports } P \Rightarrow Q\}, \\
② &= \{x \in OBJ(P) \mid x \text{ supports } P \Rightarrow Q'(= [Dec, val'])\}.
\end{aligned}
$$

We name these two sets *granules for τ in ψ*, and we employ the notation $Gr_{\psi}(\{P\}, \{Q\})$=(①, ②). Clearly, the following holds:

$$① \cap ② = \emptyset, \quad ① \cup ② = OBJ(P).$$

Therefore, $Gr_{\psi}(\{P\}, \{Q\})$ defines two equivalence classes over $OBJ(P)$, and stores information about τ. For example, we have the following for every $x \in$ ①:

$$support(\tau^x) = |①|/|OB|, \quad accuracy(\tau^x) = |①|/(|①| + |②|).$$

Furthermore, we can generate *merged granules* $Gr_{\psi}(\{P_1, P_2\}, \{Q\})$=(①, ②) over $OBJ(P_1 \wedge P_2)$ (for an implication $P_1 \wedge P_2 \Rightarrow Q$) from two sets of granules $Gr_{\psi}(\{P_1\}, \{Q\})$=($①_1$, $②_1$) and $Gr_{\psi}(\{P_2\}, \{Q\})$=($①_2$, $②_2$). Then, we have the following easily:

$$① = ①_1 \cap ①_2, \quad ② = OBJ(P_1 \wedge P_2) \setminus ①.$$

Example 1. *Let us consider Table 1, and two implications* $\tau : [A, 1] \Rightarrow [C, 1]$

Table 1: An exemplary *DIS* ψ_1.

OB	A	B	C
x_1	1	2	1
x_2	1	2	1
x_3	1	1	1
x_4	1	3	2
x_5	2	2	1

and $\tau' : [A,1] \wedge [B,2] \Rightarrow [C,1]$.

$$\begin{aligned}
&\textit{For } \tau : [A,1] \Rightarrow [C,1],\ \ OBJ([A,1]) = \{x_1, x_2, x_3, x_4\},\\
&Gr_{\psi_1}(\{[A,1]\}, \{[C,1]\}) = (\{x_1, x_2, x_3\}, \{x_4\}),\\
&support(\tau^{x_1}) = |\{x_1, x_2, x_3\}|/5 = 3/5,\\
&accuracy(\tau^{x_1}) = |\{x_1, x_2, x_3\}|/|OBJ([A,1])| = 3/4.
\end{aligned}$$

Since $Gr_{\psi_1}(\{[B,2]\}, \{[C,1]\}) = (\{x_1, x_2, x_5\}, \emptyset)$*, we obtain the following:*

$$\begin{aligned}
&\textit{For } \tau' : [A,1] \wedge [B,2] \Rightarrow [C,1],\ \ Gr_{\psi_1}(\{[A,1], [B,2]\}, \{[C,1]\}) = (①, ②),\\
&① = \{x_1, x_2, x_3\} \cap \{x_1, x_2, x_5\} = \{x_1, x_2\},\\
&OBJ([A,1] \wedge [B,2]) = \{x_1, x_2\},\ \ ② = OBJ([A,1] \wedge [B,2]) \setminus ① = \emptyset,\\
&support(\tau'^{x_1}) = |\{x_1, x_2\}|/5 = 2/5,\\
&accuracy(\tau'^{x_1}) = |\{x_1, x_2\}|/|OBJ([A,1] \wedge [B,2])| = 1.0.
\end{aligned}$$

The above consideration shows that most of the computation on rule generation in ψ can be executed by a set of granules and the merging process.

3.2 Granules for Association Rules in NIS

We consider $\tau : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val]$ in Φ. For simplicity, we employ the notation $\tau : P \Rightarrow Q$ ($P = \wedge_{A \in CON}[A, value_A]$, Q=$[Dec, val]$), $P \Rightarrow Q'$, $P' \Rightarrow Q$ and $P' \Rightarrow Q'$ ($P \neq P'$, $Q \neq Q'$).

Example 2. *In Figure 1, object 2 supports two implications,* $\tau_{red,s} : [color, red] \Rightarrow [size, s]$ *and* $[color, red] \Rightarrow [size, m]$*. If we focus on* $\tau_{red,s}$*, object 2 supports both* $P \Rightarrow Q$ *and* $P \Rightarrow Q'$*. We express this by the following:*

$$SIMP(2, \tau_{red,s}) = \{P \Rightarrow Q, P \Rightarrow Q'\}.$$

Object 3 supports two implications, and if we focus on $\tau_{blue,m} : [color, blue] \Rightarrow [size, m]$*, object 3 supports* $P \Rightarrow Q$ *and* $P' \Rightarrow Q$*. Similarly, we have the following:*

$$SIMP(3, \tau_{blue,m}) = \{P \Rightarrow Q, P' \Rightarrow Q\}.$$

For τ in Φ, we divide $sup(P)$ by $[Dec, val]$, and we define the following six sets:

$$
\begin{aligned}
&① = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q\}\},\\
&② = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q, P \Rightarrow Q'\}\},\\
&③ = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q'\}\},\\
&④ = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q, P' \Rightarrow Q\}\},\\
&⑤ = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q, P \Rightarrow Q', P' \Rightarrow Q, P' \Rightarrow Q'\}\},\\
&⑥ = \{x \in sup(P) \mid SIMP(x, \tau) = \{P \Rightarrow Q', P' \Rightarrow Q'\}\}.
\end{aligned}
$$

We name these six sets *granules for* τ *in* Φ, and we employ the notation $Gr_{\Phi}(\{P\}, \{Q\})$=(①, ②, ③, ④, ⑤, ⑥). From the above definition, each object supporting τ belongs to either ①, ②, ④ or ⑤. Each object not supporting τ belongs to either ③ or ⑥. In Figure 1, we have the following:

$$
\begin{aligned}
&Gr_{\Phi_1}(\{[color, red]\}, \{[size, s]\}) = (\emptyset, \{2\}, \emptyset, \emptyset, \{1\}, \{3\}),\\
&Gr_{\Phi_1}(\{[color, red]\}, \{[size, m]\}) = (\emptyset, \{2\}, \emptyset, \{3\}, \{1\}, \emptyset).
\end{aligned}
$$

Then, we have the following:

$$① \cap ⓙ = \emptyset \ (i \neq j),$$
$$① \cup ② \cup ③ \cup ④ \cup ⑤ \cup ⑥ = sup(P).$$

Similarly to $Gr_{\psi}(\{P\},\{Q\})$, $Gr_{\Phi}(\{P\},\{Q\})$ stores information about $\tau : P \Rightarrow Q$, and we can also generate *merged granules* $Gr_{\Phi}(\{P_1, P_2\},\{Q\})$=(①, ②, ③, ④, ⑤, ⑥) (for an implication $P_1 \wedge P_2 \Rightarrow Q$) from $Gr_{\Phi}(\{P_1\},\{Q\})$=($①_1$, $②_1$ $③_1$, $④_1$, $⑤_1$, $⑥_1$) and $Gr_{\Phi}(\{P_2\},\{Q\})$=($①_2$, $②_2$, $③_2$, $④_2$, $⑤_2$, $⑥_2$). The following holds:

$$① = ①_1 \cap ①_2, \quad ② = ②_1 \cap ②_2, \quad ③ = ③_1 \cap ③_2,$$
$$④ = (①_1 \cap ④_2) \cup (④_1 \cap ①_2) \cup (④_1 \cap ④_2),$$
$$⑤ = (②_1 \cap ⑤_2) \cup (⑤_1 \cap ②_2) \cup (⑤_1 \cap ⑤_2),$$
$$⑥ = (③_1 \cap ⑥_2) \cup (⑥_1 \cap ③_2) \cup (⑥_1 \cap ⑥_2).$$

The details of this merging algorithm are in [18]. Most of the computation on rule generation in Φ can also be executed by a set of granules and the merging process.

In the previous implementation of rule generation in Prolog and C [13], we employed *inf* and *sup* blocks, which are extensions from the equivalence classes. However, we recently implemented the *getRNIA* system [2] by using the above granules for association rules. The employment of granules for association rules makes *NIS-Apriori* algorithm [15] more simple and more comprehensive.

4 Decision Support in DIS and NIS

This section considers decision support in *DIS* and *NIS*. This is an advancement from [16].

4.1 Decision Support Task in DIS

In *DIS*, we carry out the following task.

(Decision support task in ψ)

(Input) $\wedge_{A \in CON}[A, value_A]$ and the decision attribute *Dec*.

(Output) The set below:

$$\{(val_j, support(\tau_j^x), accuracy(\tau_j^x)) \mid val_j \in VAL_{Dec},\ x \in ①\},$$
$$\tau_j : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val_j].$$

Since τ_j^x is an evidence for supporting the decision val_j, this task helps us to decide the most suitable decision val_j by using $support(\tau_j^x)$ and $accuracy(\tau_j^x)$. We can apply this strategy to any condition $\wedge_{A \in CON}[A, value_A]$.

In Figure 1, let us consider DIS_4. If the condition is $[color, red]$ and $Dec{=}size$, we have the decision $size{=}m$, because there is the unique implication below:

$$\tau^1 : [color, red] \Rightarrow [size, m]\ (support(\tau^1) = 1.0, accuracy(\tau^1) = 1.0).$$

In this case, we definitely conclude the decision $size{=}m$.

4.2 Decision Support Task in NIS

We extend decision support in *DIS* to *NIS*. For each τ^x, it is easy to calculate $support(\tau^x)$ and $accuracy(\tau^x)$ in ψ by using $Gr_\psi(\{P\}, \{Q\})$. However, these values depend upon $\psi \in DD(\Phi)$. Therefore, we define the maximum and the minimum values in Φ below:

$$
\begin{aligned}
&minsupp(\tau^x) = min_{\psi \in DD(\Phi)}\{support(\tau^x) \text{ in } \psi\},\\
&minacc(\tau^x) = min_{\psi \in DD(\Phi)}\{accuracy(\tau^x) \text{ in } \psi\},\\
&maxsupp(\tau^x) = max_{\psi \in DD(\Phi)}\{support(\tau^x) \text{ in } \psi\},\\
&maxacc(\tau^x) = max_{\psi \in DD(\Phi)}\{accuracy(\tau^x) \text{ in } \psi\},\\
&\text{where } support(\tau^x) = accuracy(\tau^x) = 0 \text{ in } \psi,\ \text{if } x \notin OBJ(\tau) \text{ in } \psi.
\end{aligned}
$$

The four values depend upon $|DD(\Phi)|$. However, we can prove the following result, and we escape from the problem on the computational complexity.

Proposition 1. *In Φ, let us consider $\tau : P \Rightarrow Q$ and $Gr_\Phi(\{P\},\{Q\})$= (①, ②, ③, ④, ⑤, ⑥).*

(1) For any $x \in$ ①,

$$
\begin{aligned}
&minsupp(\tau^x) = |①|/|OB|,\\
&minacc(\tau^x) = |①|/(|①| + |②| + |③| + |⑤| + |⑥|).
\end{aligned}
$$

(2) For any $x \in$ (② ∪ ④ ∪ ⑤),

$$minsupp(\tau^x) = 0,\ minacc(\tau^x) = 0.$$

(3) For any $x \in$ (① ∪ ② ∪ ④ ∪ ⑤),

$$
\begin{aligned}
&maxsupp(\tau^x) = (|①| + |②| + |④| + |⑤|)/|OB|.\\
&maxacc(\tau^x) = (|①| + |②| + |④| + |⑤|)/(|①| + |②| + |③| + |④| + |⑤|).
\end{aligned}
$$

Proof. In this proof, we show the procedure to obtain some derived *DISs* from $DD(\Phi)$ by selecting an implication in each granule. Each object x supporting τ belongs to either ①, ②, ④ or ⑤.

(1) Since $accuracy(\tau^x)$=$|OBJ(\tau)|/|OBJ(P)|$, we select implications satisfying

'the same condition and the different decision'. Let $NUME$ and $DENO$ denote the set of objects of the numerator and the denominator.

(A) For ①, we only select $P \Rightarrow Q$, so $NUME$=$DENO$=①.

(B) For ②, we select $P \Rightarrow Q'$, and we obtain $NUME$=①, $DENO$=① ∪ ②.

(C) For ③, we only select $P \Rightarrow Q'$, and we obtain $NUME$=①, $DENO$=① ∪ ② ∪ ③.

(D) For ④, we select $P' \Rightarrow Q$, because the selection of $P \Rightarrow Q$ increases the $accuracy(\tau^x)$ ($N/M \leq (N+1)/(M+1)$ for $0 < N \leq M$), and we obtain $NUME$=①, $DENO$=① ∪ ② ∪ ③.

(E) For ⑤, we select $P \Rightarrow Q'$, and we obtain $NUME$=①, $DENO$=① ∪ ② ∪ ③ ∪ ⑤.

(F) For ⑥, we select $P \Rightarrow Q'$, and we obtain $NUME$=①, $DENO$=① ∪ ② ∪ ③ ∪ ⑤ ∪ ⑥.

By having the above selections, each element in $OBJ(P)$ is fixed. In ψ with such attribute values, $accuracy(\tau^x)$ is clearly the minimum, whose value is $NUME/DENO$=|①|/(|①| + |②| + |③| + |⑤| + |⑥|). This is the formula for $minacc(\tau^x)$. The above selections minimize not only $accuracy(\tau^x)$ but also $support(\tau^x)$, because τ is only supported by the objects in ①. Therefore, we conclude $minsupp(\tau^x)$ is |①|/$|OB|$. We also need to know that there is at least one $\psi_{min} \in DD(\Phi)$, where both $support(\tau^x)$ and $accuracy(\tau^x)$ become the minimum.

(2) For $x \in$ (② ∪ ④ ∪ ⑤), we can select an implication except $P \Rightarrow Q$ from the object x. Therefore, there is at least one ψ where $x \notin OBJ(\tau)$. In this ψ, τ^x does not occur, and we conclude $minsupp(\tau^x)$=$minacc(\tau^x)$=0.

(3) Since $accuracy(\tau^x)$=$|OBJ(\tau)|/|OBJ(P)|$, we select implications satisfying 'the same condition and the same decision'.

(A) For ①, we only select $P \Rightarrow Q$.

(B) For ②, we select $P \Rightarrow Q$.

(C) For ③, we only select $P \Rightarrow Q'$.

(D) For ④, we select $P \Rightarrow Q$.

(E) For ⑤, we select $P \Rightarrow Q$.

(F) For ⑥, we select $P' \Rightarrow Q'$.

In (A) and (C), the selection is unique. In (B), (D), (E), $P \Rightarrow Q$ is selected, respectively. In (F), the selection of $P \Rightarrow Q'$ reduces the $accuracy(\tau^x)$, so $P' \Rightarrow Q'$ is selected. In ψ with the above selections, clearly $accuracy(\tau^x)$=(|①| + |②| + |④| + |⑤|)/(|①| + |②| + |③| + |④| + |⑤|) is the maximum. At the same time in ψ, $P \Rightarrow Q$ is selected in all possible cases. Therefore, $support(\tau^x)$=(|①| + |②| + |④| + |⑤|)/$|OB|$ is also the maximum. We need to know that there is at least one $\psi_{max} \in DD(\Phi)$, where both $support(\tau^x)$ and $accuracy(\tau^x)$ become the maximum. □

By using the above result, we consider the decision support task in *NIS*.

(Decision support task in NIS)

(Input) $\wedge_{A \in CON}[A, value_A]$ and the decision attribute Dec.

(Output) The set below:

$$\{(val_j, [minsupp(\tau_j^x), maxsupp(\tau_j^x)], [minacc(\tau_j^x), maxacc(\tau_j^x)]) \mid$$
$$val_j \in VAL_{Dec}\},\ \tau_j : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val_j].$$

If ① $\neq \emptyset$, we employ an object $x \in$ ①.

Otherwise, we employ an object $x \in$ ② $\cup$ ④ $\cup$ ⑤.

Since τ_j^x is an evidence for supporting the decision val_j, this task helps us to decide the most suitable decision val_j. The $support(\tau^x)$ and $accuracy(\tau^x)$ in ψ are extended to the following intervals in Φ:

$$[minsupp(\tau_j^x), maxsupp(\tau_j^x)] \text{ and } [minacc(\tau_j^x), maxacc(\tau_j^x)].$$

In Φ_1, let us consider the case that the condition is $[color, red]$ and Dec=$size$. There are three implications:

$$(Case\ 1)\ \tau_1^1 : [color, red] \Rightarrow [size, s],$$
$$(Case\ 2)\ \tau_2^1 : [color, red] \Rightarrow [size, m],$$
$$(Case\ 3)\ \tau_3^1 : [color, red] \Rightarrow [size, l].$$

As for τ_3^1, we have the following:

$$Gr_{\Phi_1}(\{[color, red]\}, \{[size, l]\}) = (\emptyset, \emptyset, \{2\}, \emptyset, \{1\}, \{3\}).$$

Since ①=$\emptyset$, $minsupp(\tau_3^1)$=$minacc(\tau_3^1)$=0,

$$maxsupp(\tau_3^1) = (|①| + |②| + |④| + |⑤|)/|OB| = (0+0+0+1)/3 = 1/3,$$
$$maxacc(\tau_3^1) = (|①| + |②| + |④| + |⑤|)/(|①| + |②| + |③| + |④| + |⑤|)$$
$$= (0+0+0+1)/(0+0+1+0+1) = 1/2.$$

The total output is the following:

$$\{([size, s], [0, 2/3], [0, 1.0]), ([size, m], [0, 1.0], [0, 1.0]), ([size, l], [0, 1/3], [0, 1/2])\}.$$

Unfortunately in this example, both intervals seem too wide for decision support. However, these intervals generally give us useful information for decision making under uncertainty. If we take a careful strategy, we will have the decision based on the values $minsupp(\tau_j^x)$ and $minacc(\tau_j^x)$. On the other hand, if we take an optimistic strategy, we will have the decision based on the values $maxsupp(\tau_j^x)$ and $maxacc(\tau_j^x)$.

5 Decision Support in the getRNIA Software

This section describes our *getRNIA* software and its application to decision support on the *Mushroom* data set in UCI machine learning repository [19].

5.1 getRNIA: A Web Version Program

Figure 2 shows the structure of this system, and Figure 3 shows the user interface. This system is open, and we can easily access this site [2]. The main role of the *getRNIA* system is rule generation, but we add the functionality of decision support for much better usage.

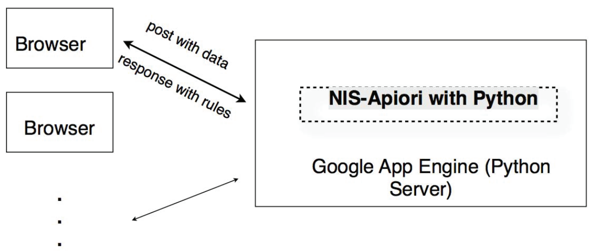

Figure 2: An overview of the principles of getRNIA.

5.2 Decision Support Functionality and an Algorithm

This algorithm is simple, and we employ granules for association rules.

(Algorithm)

(1) Specify the condition $\wedge_{A \in CON}[A, value_A]$ and the decision attribute Dec.

(2) Generate granules for association rules below:

$$\{Gr_{\Phi}(\{[A, value_A]\}, \{[Dec, val_j]\}) \mid val_j \in VAL_{Dec}\}.$$

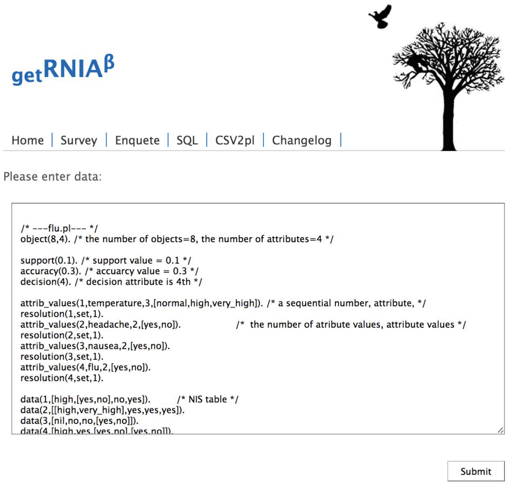

Figure 3: The current user interface of getRNIA.

(3) Apply merging algorithm, and generate granules for association rules below:

$$\{Gr_{\Phi}(\{\wedge_{A \in CON}[A, value_A]\}, \{[Dec, val_j]\}) \mid val_j \in VAL_{Dec}\}.$$

(4) Apply calculation algorithm, and generate additional information, i.e., the minimum and the maximum values.

We may directly generate $Gr_{\Phi}(\{\wedge_{A \in CON}[A, value_A]\}, \{[Dec, val_j]\})$, however if we have each $Gr_{\Phi}(\{[A, value_A]\}, \{[Dec, val_j]\})$, we can flexibly change the condition part.

5.3 An Application of the getRNIA to Mushroom Data Set

Mushroom data set consists of $|OB|$=8124 and $|AT|$=23 [19]. The following is a part of data set in the form of *getRNIA*.

```
object(8124,23)./* number of objects, number of attribute values */
support(0.2)./* support value for rule generation */
accuracy(0.6)./* accuracy value for rule generation */
decision(1)./* decision attribute */
attrib_values(1,dec,2,[e,p])./* 1st attribute values */
attrib_values(2,cap-shape,6,[b,c,x,f,k,s])./* 2nd attribute values */
      :       :       :
attrib_values(12,stalk-root,6,[b,c,u,e,z,r]).
      :       :       :
attrib_values(23,habitat,7,[g,l,m,p,u,w,d])./* 23rd attribute values */
data(1,[p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u])./* data */
data(2,[e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g]).
data(3,[e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m]).
data(4,[p,x,y,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,u]).
      :       :       :
data(8122,[e,f,s,n,f,n,a,c,b,n,e,nil,s,s,o,o,p,o,o,p,b,c,l])./* nil */
data(8123,[p,k,y,n,f,y,f,c,n,b,t,nil,s,k,w,w,p,w,o,e,w,v,l]).
data(8124,[e,x,s,n,f,n,a,c,b,y,e,nil,s,s,o,o,p,o,o,p,o,c,l]).
```

In the above data set, decision attribute is fixed to the 1st attribute, i.e., *Dec* is either *edible*$(=e)$ or *poisonous*$(=p)$. We apply *NIS-Apriori*, and can easily generate rules under this data set. In object 8122, we see a *nil*, which means a missing value. 2480 numbers of *nil* occur only in the 12th attribute *stalk-root*. Therefore we can see *Mushroom* data set is *DIS* except the 12th attribute. Since 12th attribute consists of 6 attribute values, the *getRNIA* system replaces this *nil* with a set $[b, c, u, e, z, r]$, and internally handles as *NIS*. The cardinality of $|DD(\Phi)|$ is 6^{2480}. It will be hard to enumerate all derived *DISs*. In the

Appendix, we show the obtained rules for this data set.

Now, we refer to the decision support in the *getRNIA*. We show examples.

(Case 1)

(One condition) The 14th attribute *stalk-surface-below-ring*.

Four attribute values, f=*fibrous*, y=*scaly*, k=*silky*, s=*smooth*.

The condition is [*stalk-surface-below-ring,s*].

(Decision) The 1st attribute *Dec*.

Two attribute values, e=*edible*, p=*poisonous*.

Since the 12th attribute is not related to (Case 1), we can see this is the decision support under *DIS*, namely $minsupp(\tau_j^x)$=$maxsupp(\tau_j^x)$ and $minacc(\tau_j^x)$= $maxacc(\tau_j^x)$. Probably we will have the decision *Dec*=*edible* from Figure 4. However, if we take a careful strategy, we may have the decision *Dec*=*poisonous*. This depends upon user's strategy.

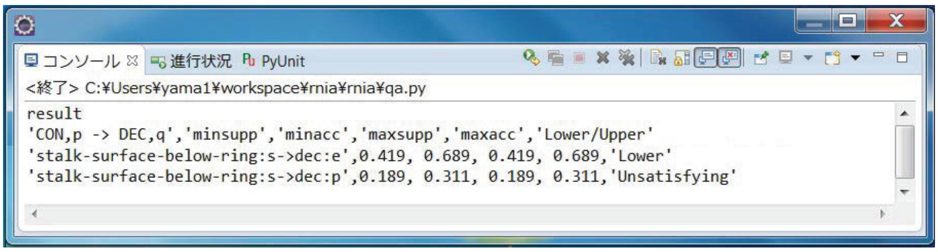

Figure 4: Additional information under [*stalk-surface-below-ring,s*].

(Case 2)

(Two conditions) The 9th attribute *gill-size* and the 11th attribute *stalk-shape*.

Two attribute values for *gill-size*, b=*broad*, n=*narrow*.

Two attribute values for *stalk-shape*, e=*enlarging*, t=*tapering*.

The condition is [*gill-size,b*]$\wedge$ [*stalk-shape,t*].

(Decision) The 6th attribute *odor*.

Nine attributes, a=*almond*, l=*anise*, c=*creosote*, y=*fishy*, f=*foul*, m=*musty*,

n=*none*, p=*pungent*, s=*spicy*.

From Figure 5, we will have the decision $odor$=n. This is also the decision support under *DIS*.

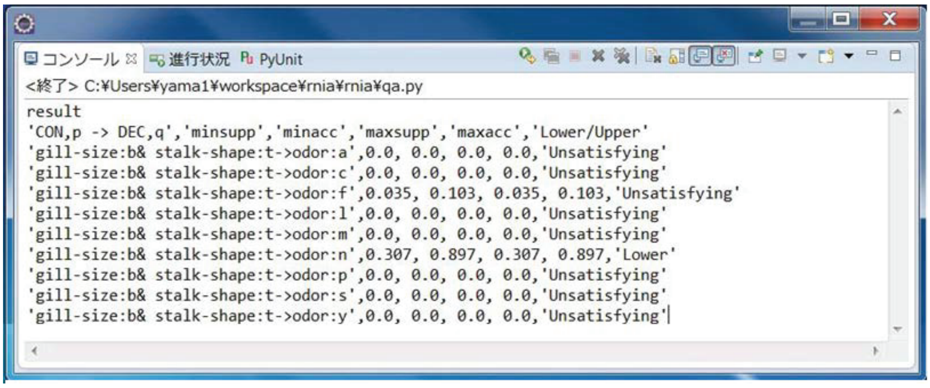

Figure 5: Additional information under [*gill-size*,b]$\wedge$ [*stalk-shape*,t].

(Case 3)

(One condition) The 12th attribute *stalk-root*. Seven attribute values including missing values *nil*, b=*bulbous*, c=*club*, u=*cup*, e=*equal*, z=*rhizomorphs*, r=*rooted*, *nil*=*missing*.

The condition is [*stalk-root*,c].

(Decision) The 7th attribute *gill-attachment*.

Four attribute values for *gill-attachment*, a=*attached*, d=*descending*, f=*free*, n=*notched*.

From Figure 6, we will have the decision *attachment*=f. In this case, each *nil* affects the maximum values.

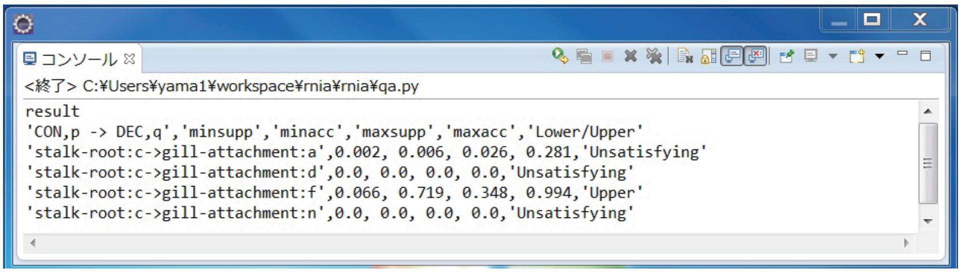

Figure 6: Additional information under [*stalk-root*,*c*].

6 Concluding Remarks

This paper advanced the preliminary work [16] by granules $Gr_\Phi(\{P\},\{Q\})$ for association rules. This will be an attempt to combine rough sets and granular computing. Since granules store information about an implication $\tau : P \Rightarrow Q$, we can see $Gr_\Phi(\{P\},\{Q\})$ as the reduced information about τ from Φ. We see the merging algorithm will be the powerful tool for handling granules, because we can calculate criterion values by a set of granules and the merging process.

RNIA handles non-deterministic information as a kind of incomplete information, and gives the formal definition of rules. We clarified the difference between rules in ψ and rules in Φ in Remark 1. We can consider any $x \in OBJ(P)$ for τ^x in ψ, but τ^x ($x \in \textcircled{1} \subseteq sup(P)$) and τ^x ($x \in \textcircled{2} \cup \textcircled{4} \cup \textcircled{5} \subseteq sup(P)$) take the different criterion values.

We add the functionality of decision support to the *getRNIA* system. In each execution, we obtained the result in real time. For a condition $\wedge_{A\in CON}[A, value_A]$, we decide val_j in ψ by the following additional information.

$$\{(val_j, support(\tau_j^x), accuracy(\tau_j^x)) \mid val_j \in VAL_{Dec}\},$$
$$\tau_j : \wedge_{A\in CON}[A, value_A] \Rightarrow [Dec, val_j].$$

Similarly, we decide val_j in Φ by the following additional information.

$$\{(val_j, [minsupp(\tau_j^x), maxsupp(\tau_j^x)], [minacc(\tau_j^x), maxacc(\tau_j^x)]) \mid$$
$$val_j \in VAL_{Dec}\},\ \ \tau_j : \wedge_{A \in CON}[A, value_A] \Rightarrow [Dec, val_j].$$

These values will help us to have the decision. The proposed decision support depends upon each $\psi \in DD(\Phi)$. If we sequentially pick up each $\psi \in DD(\Phi)$, we face with the problem on the computation. It will be hard to calculate criterion values in each ψ. We have solved this problem by Proposition 1.

The proposed method will be different from neither statistical decision making nor fuzzy decision support, and will be a new framework for decision support under uncertainty.

Acknowledgment: The authors would be grateful to anonymous referees for their useful comments. A part of this work is supported by the Grant-in-Aid for Scientific Research (C) (No.26330277), Japan Society for the Promotion of Science.

References

[1] R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, Proc. VLDB'94, Morgan Kaufmann, 1994, pp. 487–499.

[2] getRNIA software tool:
`http://getrnia.appspot.com/`

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

[4] J.W. Grzymała-Busse and W. Rzisa, A local version of the MLEM2 algorithm for rule induction, *Fundamenta Informaticae* **100** (2010), 99–116.

[5] J. Komorowski, Z. Pawlak, L. Polkowski and A. Skowron, Rough sets: a tutorial, S.K. Pal, A. Skowron (eds.) Rough Fuzzy Hybridization: A New Method for Decision Making, Springer, 1999, pp. 3–98.

[6] M. Kryszkiewicz, Rules in incomplete information systems, *Information Sciences* **113**(3/4) (1999), 271–292.

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

[8] W. Lipski, On databases with incomplete information, *Journal of the ACM* **28**(1) (1981), 41–70.

[9] E. Orłowska and Z. Pawlak, Representation of nondeterministic information, *Theoretical Computer Science* **29**(1/2) (1984), 27–39.

[10] Z. Pawlak, *Systemy Informacyjne: Podstawy teoretyczne*, Wydawnictwa Naukowo-Techniczne Publishers, 1983.

[11] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, 1991.

[12] W. Pedrycz, A.Skowron and V.Kreinovich (eds.): *Handbook of Granular Computing*, Wiley, 2008.

[13] RNIA software tool log:
`http://www.mns.kyutech.ac.jp/~sakai/RNIA`

[14] H. Sakai and A. Okuma, Basic algorithms and tools for rough non-deterministic information analysis, *Transactions on Rough Sets* **1** (2004), 209–231.

[15] H. Sakai, R. Ishibashi, K. Koba and M. Nakata, Rules and apriori algorithm in non-deterministic information systems, *Transactions on Rough Sets* **9** (2008), 328–350.

[16] H. Sakai, M. Wu and M. Nakata, Association rule-based decision making in table data, *International Journal of Reasoning-based Intelligent Systems* **4**(3) (2012), 162–170.

[17] H. Sakai, H. Okuma, M. Wu and M. Nakata, A descriptor-based division chart table in rough non-deterministic information analysis, J. Watada et al. (eds.) Smart Innovation, Systems and Technologies, **15**, 2012, pp. 25–34.

[18] H. Sakai, M. Wu, N. Yamaguchi and M. Nakata, Division charts and their merging algorithm in rough non-deterministic information analysis, Proc. IEEE. GrC2012, 2012, pp. 491–496.

[19] UCI Machine Learning Repository:
`http://mlearn.ics.uci.edu/MLRepository.html`

Appendix.

The *getRNIA* system generates the following visual results in the web page.

Certain and possible Rule Generation(Reducted): back

Support: 0.300000
Accuracy: 0.500000

CON,p -> DEC,q

	CON,p -> DEC,q	minsupp	minacc	maxsupp	maxacc	Lower/Upper
1	bruises:t->dec:e	0.339	0.815	0.339	0.815	Lower
2	odor:n->dec:e	0.419	0.966	0.419	0.966	Lower
3	gill-attachment:f->dec:e	0.494	0.507	0.494	0.507	Lower
4	gill-size:b->dec:e	0.483	0.699	0.483	0.699	Lower
5	stalk-shape:t->dec:e	0.319	0.563	0.319	0.563	Lower
6	stalk-root:b->dec:e	0.236	0.347	0.325	0.587	Upper
7	stalk-surface-above-ring:s->dec:e	0.448	0.703	0.448	0.703	Lower
8	stalk-surface-below-ring:s->dec:e	0.419	0.689	0.419	0.689	Lower
9	stalk-color-above-ring:w->dec:e	0.339	0.616	0.339	0.616	Lower
10	stalk-color-below-ring:w->dec:e	0.333	0.617	0.333	0.617	Lower
11	veil-color:w->dec:e	0.494	0.507	0.494	0.507	Lower
12	ring-type:p->dec:e	0.388	0.794	0.388	0.794	Lower
13	bruises:t& gill-spacing:c->dec:e	0.327	0.812	0.327	0.812	Lower
14	bruises:t& ring-number:o->dec:e	0.311	0.821	0.311	0.821	Lower
15	bruises:f->dec:p	0.405	0.693	0.405	0.693	Lower
16	gill-spacing:c->dec:p	0.468	0.558	0.468	0.558	Lower
17	stalk-root:b->dec:p	0.228	0.413	0.445	0.653	Upper

Figure 7: The rules from *Mushroom* data set generated by the *getRNIA*. Except 12th attribute *stalk-root*, we may see this table as *DIS*. The 6th rule and the 17th rule are related to *stalk-root*, therefore $minsupp(\tau_j^x) < maxsupp(\tau_j^x)$ and $minacc(\tau_j^x) < maxacc(\tau_j^x)$.

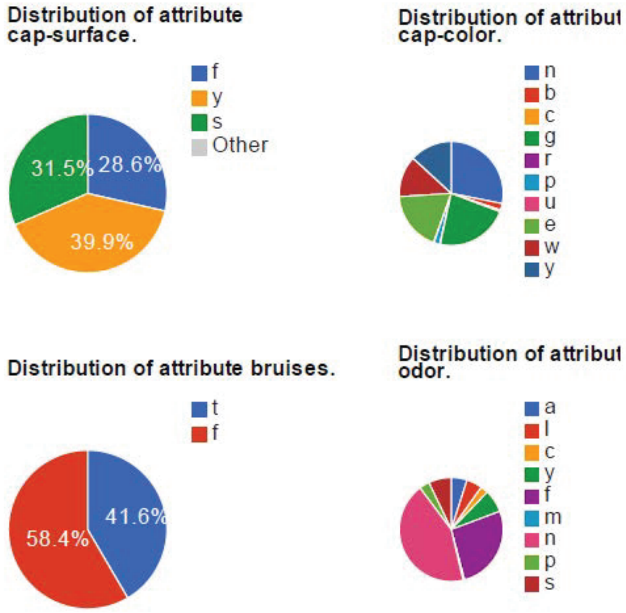

Figure 8: The pie charts for four attributes generated by the *getRNIA*.

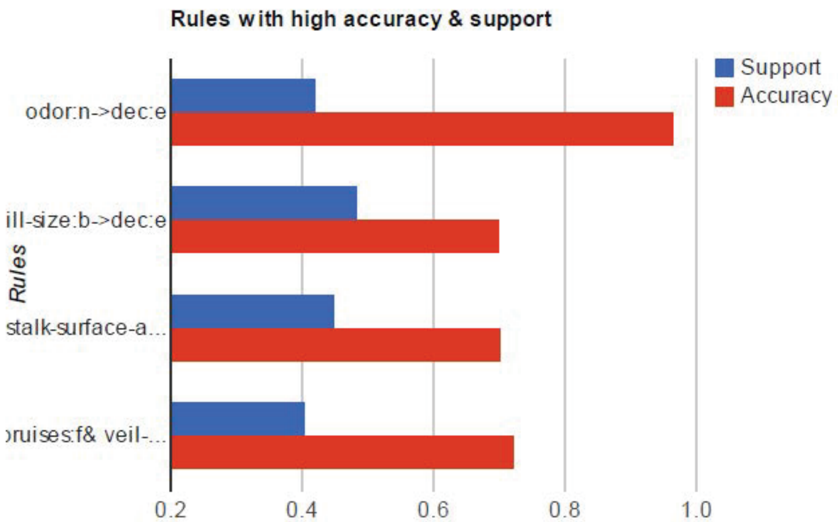

Figure 9: The top four reliable rules generated by the *getRNIA*.