International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2023, Vol. 7, No. 1, IJSKM692

# Rule Generation from Several Types of Table Data Sets and Its Application: Decision-Making with Transparency and an Improved Execution Environment

Hiroshi Sakai *, Zhiwen Jian *

## Abstract

This paper copes with rule generation from table data sets and applies the obtained rules to decision support. Here, two types of table data sets are considered. One type of them is specified as a Deterministic Information System (DIS). The other type is specified as a Non-deterministic Information System (NIS) for dealing with incomplete information. Two rule generation algorithms are refined and newly implemented in Python. Every obtained rule is applied as evidence of decision-making. Therefore, the reasoning process preserves its transparency, which will be an essential characteristic for Explainable AI. The decision support environment is strengthened due to some described improvements and is also brushed up in Python. Some running videos of Python are available on the web page. This framework applies to almost any table data sets, and we can generate rules from them. This framework based on discrete data will complement statistical data analysis based on numerical data.

*Keywords:* apriori algorithm, data mining, decision support, rough sets

## 1 Introduction

This paper discusses Apriori-based rule generation and applies the obtained rules to decision support. We will report the improved situation and further examine the reasoning functionality for decision-making by using our background. Here, we handle several types of table data sets with the following subjects.

- Discrete values or numerical values,
- Deterministic values or non-deterministic values (incomplete information),
- Big data and heterogeneous data.

In our framework, it is possible to generate rules from the above tables in the same way. We can uniformly handle rules from several types of table data sets. Every obtained rule is applied as evidence of decision-making. Therefore, the reasoning process preserves its transparency, which will be an essential characteristic for Explainable AI [5]. We need

* Kyushu Institute of Technology, Kitakyushu, Japan

a system that maintains not only ‘conclusion’ but also ‘conclusion+reasoning.’ The decision support environment is strengthened due to some described improvements and is also brushed up in Python.

This paper consists of the following. In Section 2, the rule generation framework in DIS is reviewed, and some parts in the DIS-Apriori algorithm [17] are developed. The functionality of the DIS-Apriori algorithm is also improved to the FDIS-Apriori algorithm for discretization and big data. The obtained rules are applied to decision-making with transparency. In Section 3, the rule generation framework in NIS is reviewed. We employ the NIS-Apriori algorithm [18] [19] and define two types of decision-making. The one is certainty-first decision-making, and the other is possibility-first decision-making. The functionality of two kinds of decision-making is implemented in Python. In Section 4, DIS and NIS’s perspective is considered, and a framework of Machine Learning by Rule Generation (MLRG) is proposed. Section 5 concludes this paper.

## 2 Rules and Decision Support in DISs

In this section, we clarify rules and the rule generation algorithm. Then, we enumerate the improved functionalities and describe the decision-making method in DISs.

### 2.1 Rules in DISs and Rule Generation

Table 1 is a standard table. Such a table is called a *Deterministic Information System* (DIS) [10] [14] [21].

Table 1: An exemplary Deterministic Information System (DIS) $\psi$.

| Object | A | B | C | D | Dec |
|---|---|---|---|---|---|
| $x1$ | 1 | 2 | 3 | 3 | p |
| $x2$ | 3 | 3 | 3 | 2 | p |
| $x3$ | 2 | 3 | 3 | 2 | q |
| $x4$ | 2 | 1 | 1 | 3 | q |
| $x5$ | 1 | 3 | 1 | 2 | r |

In Table 1, we consider implications like $[A,1] \Rightarrow [Dec,p]$ from $x1$ and $[C,3] \wedge [D,2] \Rightarrow [Dec,q]$ from $x3$. If an implication $\tau$ satisfies a given constraint, we see $\tau$ is a *rule* from DIS $\psi$. A pair of an attribute and an attribute value, like $[A,1]$, is termed a *descriptor*. The following is a standard definition of rules [1] [2] [10] [14] [21], and we newly add the concept of redundancy to it.

**Definition 1.** *We define the following.*
*(A rule from DIS)*
*A rule is an implication* $\tau : \wedge_i [A_i, val_i] \Rightarrow [Dec, val]$ *satisfying (1) and (2).*
*(1)* $support(\tau) \geq \alpha$ *and* $accuracy(\tau) \geq \beta$ $(0 < \alpha,\ \beta \leq 1.0)$ *for given threshold values* $\alpha$ *and* $\beta$*.*
*(2) The condition part of* $\tau$ *is minimal.*
*(Rule generation from DIS)*
*A set RULE of all rules from DIS is fixed, if the* $\alpha$ *and* $\beta$ *values are given. Rule generation is to obtain this set RULE.*

To reduce the size of RULE, we handle only implications with minimal condition part and do not handle any redundant implication satisfying (1). For example, if we detect $[C,3] \Rightarrow [Dec,q] \in$ RULE, we directly decide $[C,3] \wedge [\_,\_] \Rightarrow [Dec,q] \notin$ RULE ($[\_,\_]$ means any descriptor). Here, an implication $\tau$ is characterized by an occurrence ratio $support(\tau)$ (=(the occurrence number of $\tau$)/(the number of objects)) and a consistency ratio $accuracy(\tau)$ (=(the occurrence number of $\tau$)/(the occurrence number of $\wedge_i[A_i, val_i]$)). Furthermore, $lift(\tau)$ is a ratio $accuracy(\tau)/P([Dec,val])$ [11], here $P([Dec,val])$ is a ratio of the decision part. The lift value is employed for imbalanced data sets [3]. For $\tau : [C,3] \Rightarrow [Dec,q]$, $\tau$ occurs one time for 5 objects. Thus, $support(\tau)$=1/5. The condition part $[C,3]$ occurs 3 times, so $accuracy(\tau)$=1/3. The ratio $P([Dec,q])$=2/5, so $lift(\tau)$=(1/3)/(2/5)=5/6.

## 2.2 The DIS-Apriori Algorithm and Some Improvements

In DIS, each object is identified with a set of descriptors, for example object $x1$ is identified with a set $\{[A,1],[B,2],[C,3],[D,3],[Dec,p]\}$ of descriptors. If we identify a descriptor with an item in transaction data, we may see object $x1$ shows one transaction. Due to this way, we can extend the Apriori algorithm [1] [2] for transaction data to the DIS-Apriori algorithm for DIS in Algorithm 1.

Generally, there is a decision attribute $Dec$ in DIS. We make use of this characteristic and introduce the next sets $\mathrm{IMP}_1$, $\mathrm{IMP}_2$, $\cdots$, $\mathrm{IMP}_n$.

$$
\begin{aligned}
&\mathrm{IMP}_1 = \{[A, val_A] \Rightarrow [Dec, val] \text{ for decision attribute } Dec \text{ and for every } A, val_A, val\},\\
&\mathrm{IMP}_2 = \{[A, val_A] \wedge [B, val_B] \Rightarrow [Dec, val]\},\\
&\mathrm{IMP}_3 = \{[A, val_A] \wedge [B, val_B] \wedge [C, val_C] \Rightarrow [Dec, val]\},\\
&\qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots
\end{aligned}
$$

Here, $\mathrm{IMP}_1$ means a set of implications, consisting of one condition attribute. $\mathrm{IMP}_2$ does a set of implications, consisting of two condition attributes. $\mathrm{IMP}_3$ does a set of implications, consisting of three condition attributes, etc. IMP=$\cup_i \mathrm{IMP}_i$ is a set of all implications from

**Algorithm 1** The DIS-Apriori algorithm.

**Input:** DIS $\psi$, decision attribute $Dec$, threshold values $\alpha$, $\beta$.
**Output:** A set $Rule(\psi)$ of rules.

$i \leftarrow 1$;
create $CAN_1$=$\{\tau \in IMP_1 \mid support(\tau) \geq \alpha\}$;
**while** ($|CAN_i| \geq 1$) **do**
    $Rest_i \leftarrow \{\}$; $Rule_i \leftarrow \{\}$;
    **for all** $\tau_{i,j} \in CAN_i$ **do**
        **if** $accuracy(\tau_{i,j}) \geq \beta$ **then** add $\tau_{i,j}$ to $Rule_i$;
            **else** add $\tau_{i,j}$ to $Rest_i$;
        **end if**
    **end for**
    remove redundant implications from $Rule_i$;
    create $CAN_{i+1}(\subset IMP_{i+1})$ from $Rest_1$ and $Rest_i$;
    $i \leftarrow i+1$;
**end while**
**return** $Rule(\psi)$=$\cup_{k<i} Rule_k$

DIS, and RULE is a subset of IMP. Of course, we can simply examine the constraints (1) and (2) in Definition 1 for each $\tau \in$IMP. However, the Apriori algorithm was proposed to avoid this simple method. To reduce the size of this paper, we enumerate the properties of the DIS-Apriori algorithm and improvements.

- The DIS-Apriori algorithm basically enumerates all candidates of rules then examines the constraint. Some propositions for line 11 in Algorithm 1 were proved in [8], and we reduced the number of all candidates by using IMP's characteristics. This result also caused a reduction in the execution time.

- The DIS-Apriori algorithm is sound and complete for RULE. Namely, this algorithm does not miss any rule in Definition 1. If we assign larger values like 0.2 or 0.3 to $\alpha$, smaller numbers of implications are stored in $Rest_i$. Therefore, the calculation will not be time-consuming. If we assign smaller values like 0.01 or 0.001 to $\alpha$, larger numbers of implications are stored in $Rest_i$. Therefore, the calculation will be time-consuming. We can control rule generation by changing the value of $\alpha$.

- If we change the criterion values *support* and *accuracy*, the DIS-Apriori can generate different kinds of rules. The NIS-Apriori algorithm in the subsequent section follows this property.

- We recently implemented the DIS-Apriori algorithm in Python. This rule generator is much faster than the previously implemented rule generator in SQL. The DIS-Apriori system consists of two new programs in Figure 1, the first one is a translation program (`trans.py`) from DIS to one common RDF format [22], and the second one is a rule generation program (`disapri.py`) based on the RDF format. In `trans.py`, the different characteristics (the number of attributes, the names of attributes, etc.) of DIS are translated to the common RDF format, so we need to create one translation program for one DIS. However, `disapri.py` can handle any data in the form of the RDF format.

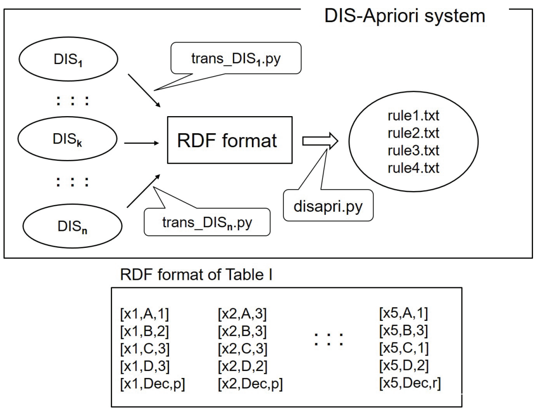


Figure 1: An Overview of the DIS-Apriori System.

The following is the comparison between other frameworks of rule generation.

- The DIS-Apriori algorithm generates the same rules by Pawlak's reduction [14], if we specify $\alpha > 0$ and $\beta$=1.0.

- The DIS-Apriori algorithm also generates the same rules by Skowron's discernibility function [21] for $\alpha > 0$ and $\beta$=1.0.
- The DIS-Apriori algorithm generates the same rules by VPRS [23] by Zialko, if we specify $\alpha > 0$ and $0.5 < \beta < 1.0$.
- Thus, the DIS-Apriori algorithm can simulate previously proposed rule generation. However, to obtain all reducts (all sets of consistent attributes for the decision attribute) is proved to be NP-hard by Skowron [21]. Therefore, to obtain all rules will be NP-hard, and to specify $\alpha \approx 0$ means that this rule generation will be very time-consuming.

## 2.3 The FDIS-Apriori Algorithm for Discretized DISs with Frequency

To handle table data sets with numerical values, we usually employ discretization of numerical values. In this case, we may reduce the number of objects, and this property may help handle big data. We describe an improvement of the DIS-Apriori algorithm to the FDIS-Apriori algorithm using the Iris data set [4].

The Iris data set consists of 150 objects, four condition attributes, 'spl', 'spw', 'pel', and 'pew', one decision attribute 'class'. Each attribute value is numerical. We classified each attribute value to the one of 'small', 'medium', and 'large'. For example, the attribute value of 'spl' is between 4.3 and 7.9, and we define [spl,small]=$\{$object $x \mid$ spl $< 5.5\}$, [spl,medium]=$\{$object $x \mid 5.5 \leq$ spl $< 6.7\}$, and [spl,large]=$\{$object $x \mid$ spl $\geq 6.7\}$. In this discretization, any object belongs to one block of the following Cartesian product

- $\Pi_{att=spl,spw,pel,pew}\{[att,s],[att,m],[att,l]\} \times \{[class,set],[class,ver],[class,vir]\}$,

whose number of blocks is 243 (=$3^5$). Each of the 150 objects belongs to one block. However, 25 blocks were sufficient for expressing 150 objects in Figure 2.

```
[1,((spl,s),(spw,m),(pel,m),(pew,m),(class,ver)),1]
[2,((spl,s),(spw,s),(pel,s),(pew,s),(class,set)),2]
[3,((spl,l),(spw,m),(pel,l),(pew,l),(class,vir)),15]
[4,((spl,l),(spw,s),(pel,l),(pew,l),(class,vir)),2]
[5,((spl,s),(spw,m),(pel,s),(pew,s),(class,set)),42]
[6,((spl,l),(spw,m),(pel,l),(pew,m),(class,ver)),1]
        :          :          :
[15,((spl,s),(spw,l),(pel,s),(pew,s),(class,set)),1]
[16,((spl,l),(spw,s),(pel,m),(pew,m),(class,ver)),1]
[17,((spl,m),(spw,s),(pel,m),(pew,m),(class,ver)),27]
[18,((spl,m),(spw,m),(pel,l),(pew,l),(class,vir)),7]
        :          :          :
[25,((spl,m),(spw,s),(pel,l),(pew,m),(class,ver)),1]
```

Figure 2: Each of the 150 objects is expressed by the 25 blocks with frequency (the last number in the list). Some different objects belong to the same block by discretization.

In Figure 2, block 5 (line 5) expresses 42 objects by this discretization. We are often faced with such cases in discretization. In the HTRU2 data set [4], 17898 objects are classified into only 134 blocks. We term such a discretized DIS with frequency a *FDIS*. We employed the property of FDIS and improved the DIS-Apriori algorithm to the FDIS-Apriori algorithm, where the calculation of *support* and *accuracy* values are slightly changed.

Let us consider $\tau : [C,3] \wedge [D,2] \Rightarrow [Dec,q]$ in Table 1. The DIS-Apriori algorithm internally generates equivalence classes [14] at the beginning, i.e., $eq([C,3])=\{x1,x2,x3\}$, $eq([D,2])=\{x2,x3,x5\}$, and $eq([Dec,q])=\{x3,x4\}$. Since $eq([C,3] \wedge [D,2] \wedge [Dec,q])=\{x3\}$, $support(\tau)= |\{x3\}|/5=1/5=0.2$ and $accuracy(\tau)=|\{x3\}|/|\{x2,x3\}|=1/2=0.5$ are obtained. Since $P([Dec,q])=2/5$, $lift(\tau)=(1/2)/(2/5)=5/4$ is obtained. The frequency of each object is 1. However, we need to consider the frequency of the object in FDIS. The frequency of block 5 is not 1 but 42 in Figure 2. By adjusting the calculation of $support(\tau)$ and $accuracy(\tau)$ w.r.t. such frequencies, we implemented the FDIS-Apriori algorithm for FDISs. Of course, this implemented system generates the same rules as that of the DIS-Apriori for DISs. The execution time of the algorithm for discretized DISs is generally reduced. In the HTRU2 data set, the execution time is reduced to about 1/15 of the DIS-Apriori for DISs. The running video on the HTRU2 data set is on the web page [20].

### 2.4 Decision Support Environment in DISs

In this subsection, we describe the overview of the decision support environment by using the Car Evaluation data set [4]. This data set consists of 1728 objects, 6 condition attributes, one decision attribute *acceptability*. We specify a condition of descriptors and apply each rule $\tau \in$ RULE to decision. We employ the following two cases for selecting one rule.

- In case I, some applicable rules (the first priority is $accuracy(\tau)$ and the second priority is $support(\tau)$) are applied to concluding decision.
- In case II, some applicable rules (the first priority is $lift(\tau)$ and the second priority is $support(\tau)$) are applied to concluding decision.

```
Condition:(('maint','vhigh'),('safety','low'))

Accuracy-first Application----------------------------------------
('acceptability','unacc')
['16',(('safety','low'),('acceptability','unacc')),0.333,1.0,1.43]

Lift-first Application----------------------------------------
('acceptability','unacc')
['16',(('safety','low'),('acceptability','unacc')),0.333,1.0,1.43]
```

Figure 3: An example of decision-making in the Car Evaluation data set.

In Figure 3, the condition is

$\{[buying,\_], [maint, vhigh], [doors,\_], [persons,\_], [lug\_boot,\_], [safety, low]\}$.

Here, two descriptors are specified, and

- Rule 16 ($[safety, low] \Rightarrow [acceptability, unacc]$, $support$=0.33, $accuracy$=1.0, and $lift$=1.43)

is applied, and the decision value *unacc* is obtained. Rule 16 is the evidence of this decision-making. In Figure 4, the decision value is different w.r.t. the selection of *accuracy* or $lift$. The decision value *unacc* is concluded by Rule 13, and the decision value *acc* is concluded by Rule 31. Probably, we need another information management to solve the above case.

```
Condition:(('maint','med'),('doors','2'),('safety','high'),('persons','4'))

Accuracy-first Application----------------------------------------
('acceptability','unacc')
['13',(('doors','2'),('acceptability','unacc')),0.189,0.755,1.08]

Lift-first Application----------------------------------------
('acceptability','acc')
['31',(('safety','high'),('persons','4'),('maint','med'),
('acceptability','acc')),0.017,0.625,2.81]
```

Figure 4: Another example of decision-making in the Car Evaluation data set.

# 3 Rules and Decision Support in NISs

This section extends the framework of the DIS-Apriori algorithm. Table 2 is an exemplary *Non-deterministic Information System* (NIS) [12] [13] [16].

Table 2: An exemplary Non-deterministic Information System (NIS) $\Phi$.

| Object | P | Q | R | Dec |
|---|---|---|---|---|
| $x1$ | a | {1,2,3} | s | 1 |
| $x2$ | a | 2 | t | 1 |
| $x3$ | {a,b,c} | 2 | t | 1 |
| $x4$ | b | 3 | s | 2 |
| $x5$ | c | 3 | {s,t} | 1 |

In NIS, some attribute values are given as a set of attribute values. We see there is one truth attribute value in the set, but we can not decide it due to information incompleteness. NIS was proposed for handling DIS with uncertainty. If we replace every missing value '?' [6] with a set of all possible values, we have one NIS. Therefore, NIS will be a more general framework than that of missing values.

Under incomplete information, we obtain certain rules and possible rules by using the NIS-Apriori algorithm [18], and we realize the framework of 'certainty-first decision support' and the framework of 'possibility-first decision support.'

## 3.1 The NIS-Apriori Algorithm for Rule Generation from NISs

In NIS $\Phi$, if we replace each set of attributes with one attribute value of the set, respectively, we have one derived DIS. Let $DD(\Phi)$ denote a set of all derived DISs. In Table 2, there are 18 (=$3^2 \times 2$) derived DISs.

**Definition 2.** *we define the following.*
*(1) An implication $\tau : \wedge_i[A_i, val_i] \Rightarrow [Dec, val]$ is a certain rule in NIS $\Phi$, if the following (C1) and (C2) hold.*
*(C1) In each $\psi \in DD(\Phi)$, $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ $(0 < \alpha,\ \beta \leq 1.0)$ for given threshold values $\alpha$ and $\beta$.*
*(C2) The condition part of $\tau$ is minimal.*
*(2) An implication $\tau : \wedge_i[A_i, val_i] \Rightarrow [Dec, val]$ is a possible rule in NIS $\Phi$, if the following (P1) and (P2) hold.*
*(P1) In at least one $\psi \in DD(\Phi)$, $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ $(0 < \alpha,\ \beta \leq 1.0)$*

*for given threshold values* $\alpha$ *and* $\beta$.
*(P2) The condition part of* $\tau$ *is minimal.*

This definition based on the possible world semantics seems natural, but the number of $DD(\Phi)$ increases exponentially. For example, the number exceeds $10^{100}$ in the Mammographic data set [4]. It seemed hard to handle rules in Definition 2, however, some properties were proved and the exponential order problem was solved [17] [18] [19]. We briefly enumerate them.

- For each descriptor $[A, val]$, we employ two sets $inf([A, val])$ and $sup([A, val])$ instead of equivalence classes $eq([A, val])$.
  $inf([A, val]) = \{$object $x \mid$ the value $val$ of $A$ is definite$\}$,
  $sup([A, val]) = \{$object $x \mid$ the value $val$ of $A$ is definite or an element of a set$\}$,
  $inf(\wedge_i[A_i, val_i]) = \cap_i inf([A_i, val_i])$, $sup(\wedge_i[A_i, val_i]) = \cap_i sup([A_i, val_i])$.
  For example, in Table 2,
  $inf([P,a] \wedge [Q,2])$=$inf([P,a]) \wedge inf([Q,2])$=$\{x1, x2\} \cap \{x2, x3\}$=$\{x2\}$,
  $sup([P,a] \wedge [Q,2])$=$sup([P,a]) \wedge sup([Q,2])$=$\{x1, x2, x3\} \cap \{x1, x2, x3\}$=$\{x1, x2, x3\}$.

- For each implication $\tau : \wedge_i[A_i, val_i] \Rightarrow [Dec, val]$, there is a derived DIS $\psi_{min} \in DD(\Phi)$ where both $support(\tau)$ and $accuracy(\tau)$ are the minimum, respectively. This $\psi_{min}$ depends on $\tau$. Let $minsupp(\tau)$ and $minacc(\tau)$ be these two values, then we have the following.
  (1) If $inf(\wedge_i[A_i, val_i]) = \emptyset$, $minsupp(\tau) = minacc(\tau) = 0$.
  (2) If $inf(\wedge_i[A_i, val_i]) \neq \emptyset$,

$$support(\tau) \text{ in } \psi_{min} = \frac{|inf(\wedge_i[A_i, val_i]) \cap inf([Dec, val])|}{|OB|} = minsupp(\tau),$$
$$accuracy(\tau) \text{ in } \psi_{min} = \frac{|inf(\wedge_i[A_i, val_i]) \cap inf([Dec, val])|}{|inf(\wedge_i[A_i, val_i])| + |OUT|} = minacc(\tau).$$

  Here, $OUT = \big(sup(\wedge_i[A_i, val_i]) \setminus inf(\wedge_i[A_i, val_i])\big) \setminus inf([Dec, val])$, and $|OB|$ is the number of objects. The above formulas do not depend on the number of elements in $|DD(\Phi)|$.

- For each implication $\tau : \wedge_i[A_i, val_i] \Rightarrow [Dec, val]$, there is a derived DIS $\psi_{max} \in DD(\Phi)$ where both $support(\tau)$ and $accuracy(\tau)$ are the maximum, respectively. This $\psi_{max}$ depends on $\tau$. Let $maxsupp(\tau)$ and $maxacc(\tau)$ be these two values, then we have the following.
  (1) If $sup(\wedge_i[A_i, val_i]) \cap sup([Dec, val]) = \emptyset$, $maxsupp(\tau) = maxacc(\tau) = 0$.
  (2) If $sup(\wedge_i[A_i, val_i]) \cap sup([Dec, val]) \neq \emptyset$,

$$support(\tau) \text{ in } \psi_{max} = \frac{|sup(\wedge_i[A_i, val_i]) \cap sup([Dec, val])|}{|OB|} = maxsupp(\tau),$$
$$accuracy(\tau) \text{ in } \psi_{max} = \frac{|sup(\wedge_i[A_i, val_i]) \cap sup([Dec, val])|}{|inf(\wedge_i[A_i, val_i])| + |IN|} = maxacc(\tau).$$

  Here, $IN = \big(sup(\wedge_i[A_i, val_i]) \setminus inf(\wedge_i[A_i, val_i])\big) \cap sup([Dec, val])$, and $|OB|$ is the number of objects. These formulas do not depend on the number of elements in $|DD(\Phi)|$.

- For each implication $\tau$, threshold values $\alpha$ and $\beta$, we have the following.
  (1) $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ in each $\psi \in DD(\Phi)$ (the constraint (C1)), if and only if $minsupp(\tau) \geq \alpha$ and $minacc(\tau) \geq \beta$.
  (2) $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ in at least one $\psi \in DD(\Phi)$ (the constraint (P1)), if and only if $maxsupp(\tau) \geq \alpha$ and $maxacc(\tau) \geq \beta$.
  Thus, it is sufficient for (C1) and (P1) to consider only $\psi_{min}$ and $\psi_{max}$ instead of all derived DISs in $DD(\Phi)$.

- The DIS-Apriori algorithm sequentially enumerates implications in order from the smaller number of condition part, so the minimalty of the condition part is assured. Therefore, if we replace two criteiron values $support(\tau)$ and $accuracy(\tau)$ in the DIS-Apriori algorithm with $minsupp(\tau)$ and $minacc(\tau)$, all certain rules are generated. If we replace $support(\tau)$ and $accuracy(\tau)$ in the DIS-Apriori algorithm with $maxsupp(\tau)$ and $maxacc(\tau)$, all possible rules are generated.

- The NIS-Apriori algorithm is also sound and complete for all certain rules and all possible rules.

- For NIS, we employ a format termed NRDF [22]. For example, object $x1$ is translated to the set $\{[x1,P,a,1],[x1,Q,1,2],[x1,Q,2,2],[x1,Q,3,2],[x1,R,s,1],[x1,Dec,1,1]\}$. The last element in each list means 1: deterministic and 2: non-deterministic. Like Figure 1, we translate NIS to NRDF format and execute either a certain rule generator or a possible rule generator.

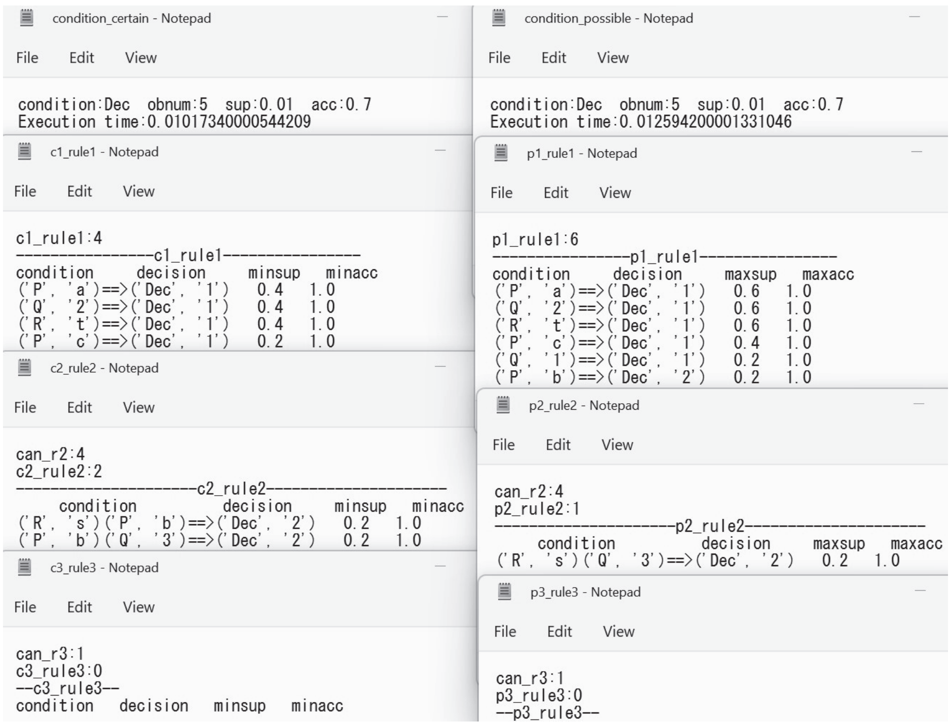


Figure 5: The obtained all certain rules and all possible rules from Table 2. Each condition part is minimal.

Figure 5 shows the stored certain rules and possible rules from Table 2 for the constraint $support(\tau) > 0.01$ and $accuracy(\tau) \geq 0.7$. We refer to the case of the Mammographic data set. This data set consists of 961 objects, 5 condition attributes, one decision attribute, and 179 missing values. The number of derived DISs exceeds $10^{100}$. For the constraint $support(\tau) > 0$ and $accuracy(\tau) \geq 0.8$, 28 certain rules were obtained in 0.252 (sec), and 98 possible rules were obtained in 0.362 (sec). They are all rules due to completeness of the NIS-Apriori algorithm, and there is no missing rule. The NIS-Apriori algorithm can handle both DIS and NIS. If this algorithm handles DIS, the set of certain rules and the set of possible rules become the same. The running video on the Mammographic data set is also available on the web page.

### 3.2 Certainty-first Decision-Making and Possibility-first Decision-Making

We move to decision-making from NIS with uncertainty. In NIS, we obtained certain rules and possible rules. If we employ certain rules for decision-making, the decision value will be supported by the most strict and certain rule, which is the rule in $\psi_{min} \in DD(\Phi)$. On the other hand, if we employ possible rules for decision-making, the decision value is supported by the most plausible rule, which is the rule in $\psi_{max} \in DD(\Phi)$. We term the former decision-making the 'certainty-first decision-making' and the latter the 'possibility-first decision making.' For selecting one implication from some implications, we employ the strategy that the first priority is $accuracy(\tau)$ and the second priority is $support(\tau)$.

We describe two types of decision-making by using the Congressional Voting data set [4]. This data set deals with the political parties in the US. This data set consists of 435 objects, 16 condition attributes, one decision attribute $a1$, and 288 missing values. The number of derived DISs exceeds $10^{80}$. For the constraint $support(\tau) > 0$ and $accuracy(\tau) \geq 0.8$, 135 certain rules were obtained in 2.202 (sec), and 216 possible rules were obtained in 2.417 (sec).

Figure 6 shows the obtained result for the condition $\{[a2,y],[a3,y],[a4,\_],\cdots,[a16,\_]\}$. In the certainty-first case,

- Rule 17 ($[a2,y] \Rightarrow [a1,dem]$, $minsupp(\tau)$=0.359 and $minacc(\tau)$=0.821)

is applied and is the evidence of decision-making. In the possible-first case,

- Rule 110 ($[a2,y] \Rightarrow [a1,dem]$, $maxsupp(\tau)$=0.379 and $maxacc(\tau)$=0.842)

is applied and is the evidence of decision-making.

```
Condition:(('a2','y'),('a3','y'))

Certainty-first---------------------------------------

('a1','dem')
['17',(('a2','y'),('a1','dem')),0.359,0.821]

Possibility-first---------------------------------------

('a1','dem')
['110',(('a2','y'),('a1','dem')),0.379,0.842]
```

Figure 6: The obtained decision [a1,dem] from the Congressional Voting data set.

Figure 7 shows the obtained result for the condition $\{[a2,\_],\cdots,[a6,y],\cdots,[a11,y],\cdots\}$. In the certainty-first case, there is no rule satisfying the specified condition. In the possible-first case,

- Rule 23 ($[a6,y] \wedge [a11,y] \Rightarrow [a1,rep]$, *maxsupp*=0.205 and *maxacc*=0.802)

is applied and is the evidence of decision-making.

```
Condition:(('a6','y'),('a11','y'))

Certainty-first---------------------------------------

Possibility-first---------------------------------------

('a1','rep')
['23',(('a6','y'),('a11','y'),('a1','rep')),0.205,0.802]
```

Figure 7: The obtained decision [a1,rep] from the Congressional Voting data set.

We described the framework of rule generation and decision support from table data sets. Since almost all table data sets will be classified into one case in Figure 8, we can consider rule generation and decision support from almost all table data sets.

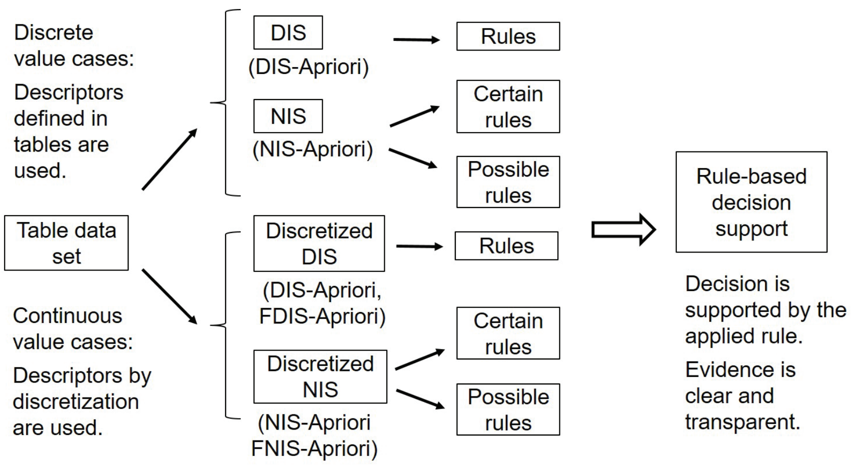


Figure 8: Apriori-based rule generation and decision support [9].

# 4 Perspective on Rule-based Table Data Analysis

This section considers two subjects w.r.t. rule-based table data analysis. The first is rule generation from heterogeneous data sets by the NIS-Apriori algorithm. The second is a new framework termed *Machine Learning by Rule Generation (MLRG)*.

## 4.1 Rule Generation from Heterogeneous Data Sets

This section shows some examples and rule generation by the NIS-Apriori algorithm.

**Example 1.** *(Heterogeneous data I: Additional information from a table)*
*Let us consider Table 3 and Table 4. Table 4 is Table 2 with additional information from Table 3. In this case, information on $x_4$ and $x_5$ is not given in Table 3. Thus, we employ non-deterministic information and can consider additional information. The NIS-Apriori algorithm generates certain and possible rules from Table 4.*

Table 3: An example of additional information.

| $Object$ | $T$ |
|---|---|
| $x_1$ | 1 |
| $x_2$ | 1 |
| $x_3$ | 2 |
| $y_1$ | 2 |
| $y_2$ | 3 |

Table 4: Information in Table 3 is added to Table 2. Here, we suppose $VAL_T=\{1,2,3\}$.

| $Object$ | $P$ | $Q$ | $R$ | $T$ | $Dec$ |
|---|---|---|---|---|---|
| $x_1$ | a | $\{1,2,3\}$ | s | 1 | 1 |
| $x_2$ | a | 2 | t | 1 | 1 |
| $x_3$ | {a,b,c} | 2 | t | 2 | 1 |
| $x_4$ | b | 3 | s | $\{1,2,3\}$ | 2 |
| $x_5$ | c | 3 | {s,t} | $\{1,2,3\}$ | 1 |

**Example 2.** *(Heterogeneous data II: Additional information from clustered uncertain data)*
*Let us consider Figure 9. It shows clustered objects $x_1$ to $x_{10}$. There is information about $x_1$ to $x_5$ in Table 2. We have additional information from Figure 9 to Table 2. Namely,*
$inf([Class,1]=\{x3\}$, $sup([Class,1])=\{x3,x5\}$, $inf([Class,2])=\{x2\}$,
$sup([Class,2])=\{x2,x4,x5\}$, $inf([Class,3])=\{x1\}$, $sup([Class,3])=\{x1,x4\}$.
*Using such information, the NIS-Apriori algorithm generates rules from Table 2 and Figure 9.*

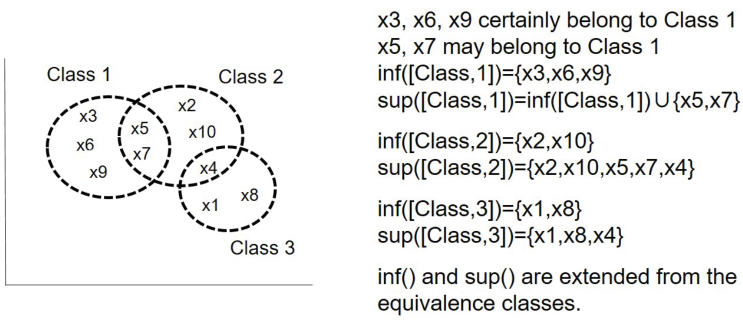


Figure 9: An example of clustered information.

We think the NIS-Apriori algorithm will be applicable to rule generation from heterogeneous data sets. This research is in progress. As for big data, we will be able to the FDIS-Apriori for FDISs. The DIS-Apriori algorithm could handle the Suspicous data set, which consists of 39427 objects, 41 condition attributes [7].

## 4.2 Machine Learning by Rule Generation (MLRG)

We apply decision-making from NIS to estimating an attribute value for incomplete information, and we sequentially have one DIS from NIS. We termed this framework MLRG.

**Example 3.** *We briefly describe the overview of MLRG using Table 2.*
*(Step 1) We fix P as the decision attribute and generate certain rules. Using the obtained rules, we have one decision attribute value 'a' for object* $x3$ *(Figure 10).*

```
['3',('P','?'),('Q','2'),('R','t'),('Dec','1')]
        ['11',(('Dec','1'),('P','a')),0.4,0.5]
        ['12',(('Q','2'),('P','a')),0.2,0.5]
Estimation:('P','a')
```

Figure 10: The estimation of attribute value $a$ from $\{a,b,c\}$. Rules 11 and 12 are applicable to object $x3$. Due to the *support* value, Rule 11 is applied.

*(Step 2) We revise the original NRDF data to a new NRDF data by the result of Step 1. Then, we fix Q as the decision attribute and generate certain rules from the revised NRDF data. By using the obtained rules, we have one decision attribute value '2' for object* $x1$ *(Figure 11).*

```
['1',('P','a'),('Q','?'),('R','s'),('Dec','1')]
        ['11',(('P','a'),('Q','2')),0.4,0.667]
        ['12',(('Dec','1'),('Q','2')),0.4,0.5]
        ['14',(('R','s'),('Q','3')),0.2,0.5]
Estimation:('Q','2')
```

Figure 11: The estimation of attribute value 2 from $\{1,2,3\}$. Rule 11 is employed due to the *accuracy* value.

*(Step 3) We re-revise the current NRDF data to a new NRDF data by using the results of the Step 1 and 2. Then, we fix R is the decision attribute and generate certain rules from the revised NRDF data. Similarly, we have one decision attribute value 't' for object* $x5$ *(Figure 12).*

```
['5',('P','c'),('Q','3'),('R','?'),('Dec','1')]
        ['12',(('Q','3'),('R','s')),0.2,0.5]
        ['15',(('Dec','1'),('R','t')),0.4,0.5]
Estimation:('R','t')
```

Figure 12: The estimation of attribute value $t$ from $\{s,t\}$.

Through the above three steps, we have one DIS from NIS. We implemented a program (`learn.py`), which simulates the above process. In `learn.py`, we specify the set of attributes with non-deterministic information and their order. This program sequentially changes the decision attribute by the specified order of attributes and applies decision making in NIS. This program employs every certain rule whose *accuracy* value is the highest. The process is managed by changing the NRDF format data, namely this program sequentially revises the NRDF data by the estimated attribute values. It took 0.160 (sec) for this execution. After this execution, we had the logged files (Figures 10-12).

**Example 4.** *We applied the implemented program to the Mammographic data set, which we referred in the previous section. There are 179 missing values, which are specified by the symbol '?'. For execution, it took 2.638 (sec) and 172 missing values were estimated. There was no applicable rule for 7 missing values. Figure 13 shows a part of the estimated values (the right side of the table). In the lower part, the logged files on the attributes margin is shown. The missing margin value of object 7 is estimated to 1 using the Rule 15.*

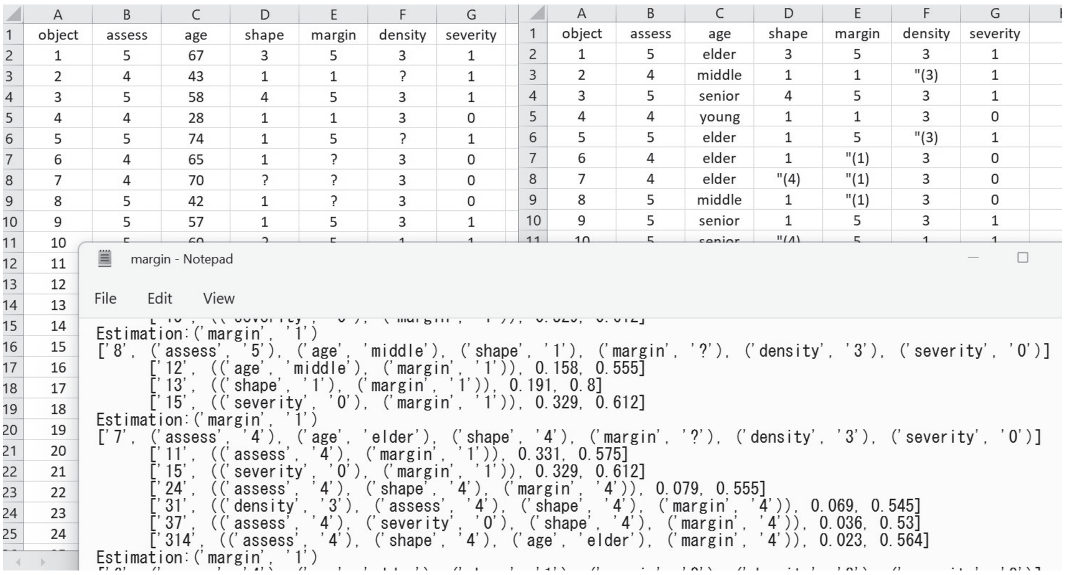


Figure 13: The learning process in the Mammographic data set.

To validate the functionality of MLRG, we are now having some experiments like the cross-validation method below:
(1) For one DIS, we add some missing values and generate one NIS.
(2) We apply the above MLRG steps to each attribute with missing values and estimate missing values.
(3) We calculate a rate such that (correctly estimated number of missing values)/ (estimated number of missing values).
(4) If the correct rate is more than 80% or better percentage, we will be able to see the MLRG will be valid.

We also enumerate the characteristics of this functionality of MLRG for NIS.
(1) This functionality needs no additional information nor statistical information.
(2) This functionality detects certain rules (or functional dependency) for the specified decision attribute and applies them, namely it will be classified as unsupervised learning.
(3) This functionality seems to correspond to the Backpropagation [15] functionality in neural networks.

# 5 Concluding Remarks

We reviewed the framework of rule generation and decision support from table data sets, and we described some improvements of the execution environment. Since almost all table data sets will be classified into one case in Figure 8, we can consider rule generation from them.

We can also control rule generation by the threshold values $\alpha$ and $\beta$. If we employ smaller $\alpha$ value, most implications in the set $\{\tau \in IMP_i \mid support(\tau) \geq \alpha\}$ remain as the candidates of rules. Thus, we will have large numbers of rules, and the execution time will be longer. If we employ larger $\alpha$ value, less implications in the set $\{\tau \in IMP_i \mid support(\tau) \geq \alpha\}$ remain as the candidates. Thus, we will have small numbers of rules. It requires less execution time. By controlling two threshold values, we can have rules from every table data set.

Furthermore, we proposed a framework of MLRG, which advances the previous framework to machine learning. In NIS, we generate certain rules, and we know local knowledge (we may see a rule implies local functional dependency). Then, we revise NIS by using local knowledge. We repeat this process, and we have one DIS from NIS. This strategy, which employs the NIS-Apriori algorithm as the core algorithm, will be unsupervised learning from table data sets with incomplete information.

Most running videos in this paper are opened on the web page [20]. Each program is implemented in Python on Windows PC with Intel Core i7 CPU, 3.60GHz.

## Acknowledgments

This work is supported by JSPS (Japan Society for the Promotion of Science) KAKENHI Grant Number JP20K11954.

## References

[1] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. VLDB'94, Morgan Kaufmann, 1994, pp. 487–499.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, "Fast Discovery of Association Rules," Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996, pp. 307–328.

[3] N.V. Chawla, "Data Mining for Imbalanced Datasets: An overview," Data Mining and Knowledge Discovery Handbook, Springer, 2009, pp. 875–886.

[4] A. Frank, A. Asuncion, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2010. `http://mlearn.ics.uci.edu/MLRepository.html` (Accessed May 27, 2021)

[5] Google Explainable AI, 2020. `https://cloud.google.com/` (Accessed March 14, 2020)

[6] J. W., Grzymała-Busse, "Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction," Transactions on Rough Sets, vol. 1, Springer LNCS 3100, 2004, pp. 78–95.

[7] Z. Jian, H. Sakai, J. Watada, A. Roy, M. B. Hassan, "An Apriori-based Data Analysis on Suspicious Network Event Recognition," Proc. IEEE Big Data 2019, 2019, pp. 5888–5896.

[8] Z. Jian, H. Sakai, T. Ohwa, K. Y. Shen, M. Nakata, "An Adjusted Apriori Algorithm to Itemsets Defined by Tables and an Improved Rule Generator with Three-way Decisions," Proc. IJCRS2020, Springer LNCS 12179, 2020, pp. 95–110.

[9] Z. Jian, H. Sakai, "On Apriori-Based Rule Generation and the Explainable Reasoning Functionality for Decision (Short Notes)," Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, 33(1), 2021, pp. 506-510 (in Japanese).

[10] J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, "Rough Sets: a Tutorial," Rough Fuzzy Hybridization: A New Method for Decision Making, Springer, 1999, pp. 3–98.

[11] P. McNicholas, Y. Zhao, "Association Rules: An Overview," Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction, IDI-Global, 2009, pp. 1–10.

[12] E. Orłowska, Z. Pawlak, "Representation of nondeterministic information," Theoretical Computer Science, vol. 29, no. 1–2, 1984, pp. 27–39.

[13] Z. Pawlak, Systemy Informacyjne: Podstawy Teoretyczne (in Polish) WNT, 1983, p.186.

[14] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991, p. 231.

[15] R. Rojas, "The Backpropagation Algorithm," Neural Networks - A Systematic Introduction, Chapter 7, Springer-Verlag, 1996.

[16] H. Sakai, R. Ishibashi, K. Koba, M. Nakata, "Rules and Apriori Algorithm in Non-deterministic Information Systems," Transactions on Rough Sets, vol. 9, Springer LNCS 5390, 2008, pp. 328–350.

[17] H. Sakai, M. Nakata, W. Z. Wu, D. Miao, G. Wang, "Special Issue: Rough Sets and Data Mining," CAAI Transactions on Intelligence Technology, IET, vol. 4, no. 4, 2019, pp. 201–260.

[18] H. Sakai, M. Nakata, J. Watada, "NIS-Apriori-based Rule Generation with Three-way Decisions and Its Application System in SQL," Information Sciences, vol. 507, 2020, pp. 755–771.

[19] H. Sakai, M. Nakata, J. Watada, "Rough Sets Non-Deterministic Information Analysis and a NIS-Apriori System - A Rule Generation System Based on Possible World Semantics -," Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, vol. 32, no. 4, 2020, pp. 747–758. (in Japanese)

[20] H. Sakai, Z. Jian, "DIS-Apriori, NIS-Apriori, Decision Support, and Machine Learning by Rule Generation," Software Tools for RNIA (Rough Sets Non-deterministic Information Analysis) Web Page, 2021. `http://www.mns.kyutech.ac.jp/~sakai/RNIA/` (Accessed May 25, 2021)

[21] A. Skowron, C. Rauszer, "The Discernibility Matrices and Functions in Information Systems," Intelligent Decision Support - Handbook of Advances and Applications of the Rough Set Theory, Kluwer Academic Publishers, 1992, pp. 331–362.

[22] D. Ślęzak, H. Sakai, "Automatic Extraction of Decision Rules from Non-deterministic Data Systems: Theoretical Foundations and SQL-based Implementation," Proc. DTA2009, Springer CCIS 64, 2009, pp. 151–162.

[23] W. Ziarko, "Variable Precision Rough Set Model," Journal of Computer and System Sciences, vol. 46, no. 1, 1993, pp. 39–59.