NIS-Apriori Algorithm with a Target Descriptor for Handling Rules Supported by Minor Instances

NIS-Apriori Algorithm with a Target Descriptor for Handling Rules Supported by Minor Instances

Hiroshi Sakai[1], Kao-Yi Shen[2], and Michinori Nakata[3]

[1] Graduate School of Engineering, Kyushu Institute of Technology,
Tobata, Kitakyushu 804-8550, Japan
sakai@mns.kyutech.ac.jp
[2] Department of Banking and Finance,
Chinese Culture University (SCE), Taipei, Taiwan
kyshen@sce.pccu.edu.tw
[3] Faculty of Management and Information Science,
Josai International University,
Gumyo, Togane, Chiba 283-0002, Japan
nakatam@ieee.org

Abstract. For each implication τ : *Condition_part* $\Rightarrow$ *Decision_part* defined in table data sets, we see τ is a *rule* if τ satisfies appropriate constraints, i.e., $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ for two threshold values α and β $(0 < \alpha, \beta \leq 1)$. If τ is a rule for relatively high α, we say τ is supported by major instances. On the other hand, if τ is a rule for lower α, we say τ is supported by minor instances. This paper focuses on rules supported by minor instances, and clarifies some problems. Then, the NIS-Apriori algorithm, which was proposed for handling rules supported by major instances from tables with information incompleteness, is extended to the NIS-Apriori algorithm with a target descriptor. The effectiveness of the new algorithm is examined by some experiments.

Keywords: Rule generation, Uncertainty, Apriori algorithm, NIS-Apriori algorithm, SQL.

1 Introduction

We have been coping with some variations of rule generation related to the Apriori algorithm [1, 16], and proposed the NIS-Apriori algorithm for handling tables with definite information (*Deterministic Information Systems: DISs*) and tables with indefinite information (*Non-deterministic Information Systems: NISs*) [9, 13]. Furthermore, we recently realized a software tool termed *NIS-Apriori in SQL* [10]. Since SQL has high versatility, the environment yielded by NIS-Apriori based rule generation in SQL will be useful for table data analysis with information incompleteness. The execution logs are uploaded to the web page [11]. In [12–14], we are also considering new topics in conjunction with three-way decisions [17].

With such a background, this paper considers two kinds of rules and the problem related to Apriori-based rule generation below:

- An implication τ is a *major rule*, if $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ for relatively high α and β.
- An implication τ is a *minor rule*, if $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ for lower α and relatively high β.
- Problem: NIS-Apriori-based rule generation will be effective for generating major rules, but it may not be effective for generating minor rules. It is necessary to take measures for minor rule generation.

Major rules reflect the tendency over major instances of the data sets. On the other hand, minor rules reflect the tendency, which strongly holds in minor instances of the data sets. For example, the English alphabet will be the major alphabet in the world, and the Japanese Hiragana and Katakana alphabets are the minor alphabets in the world. However, the major part of publications in Japan consists of the Hiragana, Katakana, and Chinese alphabets, not the English alphabet. Less people in the world understand any Japanese newspaper, but most people in Japan understand it easily. The framework termed *imbalanced data set* [3, 4] will be another approach to this issue.

In the application of NIS-Apriori-based rule generation, it will be effective for relatively high α, because the amount of possible implications is usually reduced by using the constraint $support(\tau) \geq \alpha$. However, it is not effective for lower α, because most of implications will satisfy the constraint $support(\tau) \geq \alpha$, and they are still remained as candidates of rules. So, it is very time-consuming for NIS-Apriori-based minor rule generation. For solving this problem, we extend the NIS-Apriori algorithm to that with a target descriptor.

This paper is organized as follows: Section 2 surveys the framework of NIS-Apriori-based rule generation, and clarifies the problem of minor rule generation. Section 3 proposes NIS-Apriori-based rule generation with a target descriptor, and Section 4 describes the experiments by the implemented system. Section 5 concludes this paper.

2 NIS-Apriori-based Rule Generation in NISs

This section surveys DIS-Apriori-based rule generation in DISs and NIS-Apriori-based rule generation in NISs, then clarifies the problem related to minor rules.

2.1 DIS-Apriori-based Rule Generation in DISs

Table 1 is an exemplary DIS ψ_1. We usually predefine a decision attribute Dec. In ψ_1, Dec=$price$, and CON is a subset of $\{color, size, weight\}$. In DIS ψ, we term a pair $[A, val_A]$ (an attribute A, an attribute value val_A) a *descriptor*. A *rule* is an implication $\tau : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val]$ satisfying below: [1, 8, 15].

Table 1. An exemplary DIS ψ_1 for suitcases. OB (a set of *instances*), AT (a set of *attributes*), VAL_{color} (a set of *attribute values* of *color*) is $\{red, blue, green\}$, VAL_{price}=$\{high, low\}$.

OB	*color*	*size*	*weight*	*price*
x_1	*red*	*small*	*light*	*low*
x_2	*red*	*medium*	*light*	*high*
x_3	*blue*	*medium*	*light*	*high*
x_4	*red*	*medium*	*heavy*	*low*
x_5	*red*	*large*	*heavy*	*high*
x_6	*blue*	*large*	*heavy*	*high*

$$\begin{array}{l}\text{For two threshold values } 0 < \alpha, \beta \leq 1.0,\\ support(\tau)(= N(\tau)/N(OB)) \geq \alpha,\\ accuracy(\tau)(= N(\tau)/N(\wedge_{A \in CON}[A, val_A])) \geq \beta,\\ \text{Here, } N(*) \text{ means the amount of instances satisfying the formula } *,\\ OB \text{ is a set of all instances. We define } support(\tau) = accuracy(\tau) = 0,\\ \text{if } N(\wedge_{A \in CON}[A, val_A]) = 0.\end{array} \quad (1)$$

The Apriori algorithm is originally defined for the transaction data sets, and the manipulation of item sets is proposed [1]. However, if we identify each descriptor $[A, val_A]$ with an item, we can similarly apply the Apriori algorithm to rule generation from table data sets. We see the instance x_1 shows an item set and the table ψ_1 is a set of item sets below:

$$ItemSet(x_1) = \{[color, red], [size, small], [weight, light], [price, low]\},$$
$$Set\_ItemSet(\psi_1) = \{ItemSet(x_1), ItemSet(x_2), \cdots, ItemSet(x_6)\}.$$

We term the algorithm handling the above data structure a *DIS-Apriori algorithm* (Algorithm 1). It has the following properties.
(Property 1) The amount of elements in each $ItemSet(x_i)$ is equal to the number of the attributes.
(Property 2) The decision attribute Dec is usually predefined, and the decision part is an element in the set $\{[Dec, val] \mid val$ is a decision attribute value$\}$.
(Property 3) Except (Property 1) and (Property 2), the DIS-Apriori algorithm is almost the same as the Apriori algorithm for the transaction data sets.

We say $\tau' : (\wedge_{A \in CON}[A, val_A]) \wedge [B, val_B]) \Rightarrow [Dec, val]$ is a *redundant* implication for $\tau : \wedge_{A \in CON}[A, val_A] \Rightarrow [Dec, val]$. If we recognize that τ is a rule, we automatically see τ' is also a rule for reducing the amount of rules, namely we handle only a minimal implication as a rule. We have next two additional properties.
(Property 4) If an implication τ' is redundant for τ, $support(\tau') \leq support(\tau)$ always holds.
(Property 5) If an implication τ' is redundant for τ, $accuracy(\tau') \leq accuracy(\tau)$ may not hold.

Algorithm 1 DIS-Apriori algorithm

Require: DIS ψ, the decision attribute Dec, the threshold values α, β.
Ensure: $Rule(\psi)$.
  $Rule(\psi) \leftarrow \{\}$; $i \leftarrow 1$;
  create $SubIMP_i(\subseteq IMP_i)$, where each $\tau_{i,j} \in SubIMP_i$ satisfies $support(\tau_{i,j}) \geq \alpha$;
  while $(|SubIMP_i| \geq 1)$ **do**
    $Rest \leftarrow \{\}$;
    for all $\tau_{i,j} \in SubIMP_i$ **do**
      if $accuracy(\tau_{i,j}) \geq \beta$ **then** add $\tau_{i,j}$ to $Rule(\psi)$;
      else add $\tau_{i,j}$ to $Rest$;
      end if
    end for
    $i \leftarrow i + 1$;
    generate $SubIMP_i(\subseteq IMP_i)$ by using $Rest$, where $\tau_{i,j} \in SubIMP_i$ satisfies $support(\tau_{i,j}) \geq \alpha$ and $\tau_{i,j}$ is not redundant for any implication in $Rule(\psi)$;
  end while
  return $Rule(\psi)$

We also introduce the next IMP_1, IMP_2, $\cdots$, IMP_n.
IMP_1=$\{\tau : [A, val_A] \Rightarrow [Dec, val]\}$,
  (Any implication with one condition attribute),
IMP_2=$\{\tau : [A, val_A] \wedge [B, val_B] \Rightarrow [Dec, val]\}$,
  (Any implication with two condition attributes),
IMP_3=$\{\tau : [A, val_A] \wedge [B, val_B] \wedge [C, val_C] \Rightarrow [Dec, val]\}$,
  (Any implication with three condition attribute).
  : : : :
Here, the subscript i in IMP_i is the amount of descriptors in the condition part, and $|IMP_i|$ is the amount of implications. The DIS-Apriori algorithm makes use of $SubIMP_i(\subseteq IMP_i)$ and Properties 4-5, and it is *sound* and *complete* for the rules defined in the formula (1) [13]. So, any rule defined in the formula (1) can be obtained by Algorithm 1.

2.2 NIS-Apriori-based Rule Generation in NISs

The table Φ_1 in Table 2 is an exemplary NIS. NISs were proposed by Pawlak [7], Orłowska [6], and Lipski [5] for handling information incompleteness in table data sets. Formerly, information retrieval and question answering were investigated in NISs, and we are recently coping with rule generation from NISs. We replace each non-deterministic information and each ? symbol with a possible value, and we obtain a table with deterministic information. We termed it a *derived DIS* from NIS. Let $DD(\Phi)$ be a set of all derived DISs from Φ. We see (or suppose) an actual DIS ϕ^{actual} exists in $DD(\Phi)$. For Φ_1, $DD(\Phi_1)$ consists of 144 $(=3^2 \times 2^4)$ derived DISs. Based on $DD(\Phi)$, we proposed the certain and the possible rules below:

Definition 1. *[9]*
(1) *We say τ is a certain rule, if τ satisfies $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$*

Table 2. An exemplary NIS Φ_1 for suitcases. VAL_{color} (a set of *attribute values* of *color*) is $\{red, blue, green\}$, VAL_{size}=$\{small, medium, large\}$, VAL_{weight}=$\{light, heavy\}$, VAL_{price}=$\{high, low\}$.

OB	*color*	*size*	*weight*	*price*
x_1	?	*small*	*light*	*low*
x_2	*red*	?	*light*	*high*
x_3	*blue*	*medium*	?	*high*
x_4	*red*	*medium*	*heavy*	*low*
x_5	$\{red, blue\}$	$\{medium, large\}$	*heavy*	*high*
x_6	*blue*	*large*	*heavy*	$\{high, low\}$

in each $\phi \in DD(\Phi)$,
(2) *We say* τ *is a possible rule, if* τ *satisfies* $support(\tau) \geq \alpha$ *and* $accuracy(\tau) \geq \beta$ *in at least one* $\phi \in DD(\Phi)$.

Definition 1 seems natural, but we have the computational complexity problem, because the amount of elements in $DD(\Phi)$ increases exponentially. In Φ_1, the amount is 144, and the amount is more than 10^{100} in the Mammographic data set in UCI machine learning repository [2]. For this computational problem, we defined two sets for a descriptor $[A, val]$ below:

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

By using these sets inf and sup, we have solved the computational complexity problem. With respect to an implication τ, the following holds [9].

(Result 1) There is a derived DIS $\psi_{min} \in DD(\Phi)$ satisfying (i) and (ii).
(i) $minsupp(\tau)(= \min_{\psi \in DD(\Phi)}\{support(\tau) \text{ in } \psi\}) = support(\tau) \text{ in } \psi_{min}$,
(ii) $minacc(\tau)(= \min_{\psi \in DD(\Phi)}\{accuracy(\tau) \text{ in } \psi\} = accuracy(\tau) \text{ in } \psi_{min}$.
Thus, τ is a certain rule, if and only if τ is a rule in $\psi_{min} \in DD(\Phi)$, i.e., $minsupp(\tau) \geq \alpha$ and $minacc(\tau) \geq \beta$.
(Result 2) There is a derived DIS $\psi_{max} \in DD(\Phi)$ satisfying (i) and (ii).
(i) $maxsupp(\tau)(= \max_{\psi \in DD(\Phi)}\{support(\tau) \text{ in } \psi\}) = support(\tau) \text{ in } \psi_{max}$,
(ii) $maxacc(\tau)(= \max_{\psi \in DD(\Phi)}\{accuracy(\tau) \text{ in } \psi\} = accuracy(\tau) \text{ in } \psi_{max}$.
Thus, τ is a possible rule, if and only if τ is a rule in $\psi_{max} \in DD(\Phi)$, i.e., $maxsupp(\tau) \geq \alpha$ and $maxacc(\tau) \geq \beta$.
(Result 3) Each formula of four criterion values, $minsupp(\tau), \cdots, maxacc(\tau)$, is expressed by using inf and sup sets. This calculation does not depend on the amount of $DD(\Phi)$. (We omit the formulas for them. The details are in [9, 13]). Thus, certain rule generation and possible rule generation does not depend upon

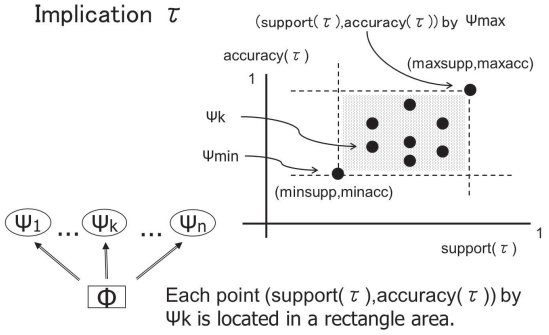

Fig. 1. Each point $(support(\tau), accuracy(\tau))$ by $\psi \in DD(\Phi)$ is located in a rectangle area [13].

the amount of elements in $DD(\Phi)$.

Based on the above results, we have a chart in Figure 1 for each implication τ. We apply the above three results to the DIS-Apriori algorithm in Algorithm 1, and we proposed the NIS-Apriori algorithm. Namely, in certain rule generation $minsupp(\tau)$ and $minacc(\tau)$ are employed instead of $support(\tau)$ and $accuracy(\tau)$ in Algorithm 1. In possible rule generation, $maxsupp(\tau)$ and $maxacc(\tau)$ are employed in Algorithm 1. Therefore, the time complexity of the NIS-Apriori algorithm is more than twice time complexities of the DIS-Apriori algorithm. However, it is possible to calculate criterion four values in polynomial order time, so the NIS-Apriori algorithm does not depend upon the amount of elements in $DD(\Phi)$. The NIS-Apriori algorithm is also sound and complete [13]. Without three results, it will be hard to handle Mammographic data set which has more than 10^{100} derived DISs. Thus, we insist that NIS-Apriori-based rule generation is a significantly new framework supported by possible world semantics.

We recently implemented the NIS-Apriori algorithm in SQL [10], and the execution logs are uploaded to the web page [11]. Figure 2 shows the obtained certain and possible rules (α=0.05 and β=0.8) from the Mammographic data set. There are 960 instances and five attributes *assess*, *age*, *shape*, *margin*, *density* (*assess* was added by physicians).

2.3 Problem on Minor Rule Generation

We clarify the problem on minor rule generation by using the Mammographic data set. We employ five α values (0.25, 0.1, 0.05, 0.01, 0.001) and one β=0.8. We see the cases I (α=0.25 and β=0.8) and II (α=0.1 and β=0.8) define major rules, and we see the cases IV (α=0.01 and β=0.8) and V (α=0.001 and β=0.8) define minor rules.

```
mysql> select * from c1_rule;
+------------+------+----------+------------+---------+--------+
| att1       | val1 | deci     | deci_value | minsupp | minacc |
+------------+------+----------+------------+---------+--------+
| age        | 80   | severity | 1          |   0.056 |  0.806 |
| assess     | 5    | severity | 1          |   0.317 |  0.869 |
| margin     | 1    | severity | 0          |   0.329 |  0.859 |
| end_attrib | NULL | NULL     | NULL       |    NULL |   NULL |
+------------+------+----------+------------+---------+--------+
4 rows in set (0.00 sec)

mysql> select * from p2_rule;
+------------+------+--------+------+----------+------------+---------+--------+
| att1       | val1 | att2   | val2 | deci     | deci_value | maxsupp | maxacc |
+------------+------+--------+------+----------+------------+---------+--------+
| age        | 40   | assess | 4    | severity | 0          |   0.094 |  0.882 |
| age        | 50   | assess | 4    | severity | 0          |   0.115 |  0.821 |
| age        | 70   | margin | 4    | severity | 1          |   0.064 |  0.847 |
| age        | 70   | shape  | 4    | severity | 1          |   0.108 |  0.874 |
| assess     | 4    | margin | 3    | severity | 0          |   0.061 |  0.831 |
| margin     | 3    | shape  | 4    | severity | 1          |   0.069 |  0.805 |
| end_attrib | NULL | NULL   | NULL | NULL     | NULL       |    NULL |   NULL |
+------------+------+--------+------+----------+------------+---------+--------+
7 rows in set (0.00 sec)
```

Fig. 2. The obtained certain rules and possible rules (α=0.05 and β=0.8) from the Mammographic data sets. Each certain rule satisfies $support \geq 0.05$ and $accuracy \geq 0.8$ in every $\psi \in DD(\Phi_{mammo})$, where the amount of elements ψ is more than 10^{100}.

Table 3. The comparison of major rule generation and minor rule generation in the Mammographic data set. $|c1rule|$ is the amount of implications in the table $c1rule$. MIN means the minimum amount of implications so as to be one rule, namely $960 \times \alpha$.

Case	α	MIN	$exec\_time$	$\|c1rule\|$	$\|c1rest\|$	$\|c2rule\|$	$\|c2rest\|$	$\|c3rule\|$	$\|c3rest\|$
				$\|p1rule\|$	$\|p1rest\|$	$\|p2rule\|$	$\|p2rest\|$	$\|p3rule\|$	$\|p3rest\|$
I	0.25	240	19.9 (sec)	2	4	0	2	0	0
	(*major rule*)			2	4	0	2	0	0
II	0.10	96	76.3 (sec)	2	14	4	11	0	1
	(*major rule*)			5	13	2	11	0	2
III	0.05	48	164.9 (sec)	3	20	6	22	0	8
				9	22	6	29	0	12
IV	0.01	10	540.6 (sec)	6	34	4	75	8	66
	(*minor rule*)			15	34	19	112	7	98
V	0.001	1	889.9 (sec)	7	43	2	192	45	378
	(*minor rule*)			18	36	48	190	44	351

Table 3 shows the execution time and the amount of rules, where tables $c1rule$, $c2rule$, and $c3rule$ store certain rules obtained from IMP_1, IMP_2, and IMP_3 in Algorithm 1, respectively. Tables $c1rest$, $c2rest$, and $c3rest$ store implications in $Rest$ in Algorithm 1. Tables $p1rule$, $p2rule$, $p3rule$, $p1rest$, $p2rest$, and $p3rest$ store possible rules and implications. In case V, each implication τ satisfies the constraint $support(\tau) \geq 0.001$, so we need to consider any implication. We cannot reduce the amount of implications by using (Property 4) in Section 2.1.

3 NIS-Apriori Algorithm with a Target Descriptor

In this section, we extend the NIS-Apriori algorithm to the *NIS-Apriori algorithm with a target descriptor* (tNIS-Apriori). Generally in Apriori-based rule generation, the decision attribute is predefined and any attribute value is considered. In the tNIS-Apriori algorithm, we consider predefined descriptor $[Dec, val]$ in Algorithms 2-3.

In Algorithm 1, we consider a set $SubIMP_i(\subseteq IMP_i)$, whose element $\tau_{i,j}$ takes $[Dec, \_]$ ($\_$ means any $val \in VAL_{Dec}$) as its decision part. However, Algorithms 2-3, we fix an attribute value $val \in VAL_{Dec}$ and consider a set $SubIMP_{i,[Dec,val]}(\subseteq SubIMP_i \subseteq IMP_i)$, whose element $\tau_{i,j}$ takes $[Dec, val]$ as its decision part. In tNIS-Apriori based rule generation, we have the following advantage and disadvantage.

(Advantage)

The NIS-Apriori algorithm tries to find all rules whose decision attribute is Dec, but the tNIS-Apriori algorithm tries to find all rules whose decision part is $[Dec, val]$. In the Mammographic data set, the decision attribute values are 0 (*benign*) and 1 (*malignant*). So, we apply the tNIS-Apriori algorithm to two decisions $[Dec, 0]$ and $[Dec, 1]$. The execution time by the NIS-Apriori algorithm is usually more time-consuming than that of the tNIS-Apriori algorithm.

(Disadvantage)

In order to have all rules, we need to repeat the execution for each $[Dec, val]$. So, the user's manipulation may be confused, if the amount of decision attribute values is large.

4 Some Experiments

We employed Windows desktop PC (3.60GHz), and revised the SQL procedures $step1$, $step2$, and $step3$ in the NIS-Apriori algorithm to the SQL procedures $tstep1$, $tstep2$, and $tstep3$ in the tNIS-Apriori algorithm. For example in the Mammographic data set, the next procedure

```
step1('severity',960,0.05,0.8)
```

(Find all implications satisfying $support(\tau) \geq 0.05$, $accuracy(\tau) \geq 0.8$.)

is changed to two procedures below:

```
tstep1('severity','0',960,0.05,0.8), tstep1('severity','1',960,0.05,0.8)
```

(Find all implications with decision value 0) and (Find all implications with decision value 1).

Table 4 shows the comparison of the execution time on the Mammographic data set [2]. Of course, it took less execution time for each of $[severity, 0]$ and $[severity, 1]$ by the tNIS-Apriori algorithm. However, total execution time SUM is worse than that by the NIS-Apriroi algorithm. In this example, the NIS-Apriori algorithm seems better than the tNIS-Apriori algorithm.

Algorithm 2 NIS-Apriori algorithm with a Target Descriptor (Certain rule generation part)

Require: NIS Φ, the descriptor $[Dec, val]$, the threshold values α, β.
Ensure: $Certain\_Rule(\Phi)$.
▷ Each changed part from Algorithm 1 is underlined.

$Certain\_Rule(\Phi) \leftarrow \{\}$; $i \leftarrow 1$;
create $SubIMP_{i,[Dec,val]} (\subseteq IMP_i)$ ($\tau_{i,j} \in SubIMP_{i,[Dec,val]}$ and the decision part of $\tau_{i,j}$ is $[Dec, val]$), and $minsupp(\tau_{i,j}) \geq \alpha$ holds;
while ($|SubIMP_{i,[Dec,val]}| \geq 1$) **do**
  $Rest \leftarrow \{\}$;
  for all $\tau_{i,j} \in SubIMP_{i,[Dec,val]}$ **do**
    if $minacc(\tau_{i,j}) \geq \beta$ **then** add $\tau_{i,j}$ to $Certain\_Rule(\Phi)$;
    else add $\tau_{i,j}$ to $Rest$;
    end if
  end for
  $i \leftarrow i + 1$;
  generate $SubIMP_{i,[Dec,val]} (\subseteq IMP_i)$ by using $Rest$, where
  $\tau_{i,j} \in SubIMP_{i,[Dec,val]}$ satisfies $minsupp(\tau_{i,j}) \geq \alpha$ and $\tau_{i,j}$ is not redundant for any implication in $Certain\_Rule(\Phi)$;
end while
return $Certain\_Rule(\Phi)$

Algorithm 3 NIS-Apriori algorithm with a Target Descriptor (Possible rule generation part)

Require: NIS Φ, the descriptor $[Dec, val]$, the threshold values α, β.
Ensure: $Possible\_Rule(\Phi)$.
▷ In possible rule generation, we replace $minsupp$ and $minacc$ in Algorithm 2 with $maxsupp$ and $maxacc$, respectively. The other part is the same as Algorithm 2.

Table 5 shows the comparison of the execution time on the Congressional Voting data set [2]. This data set consists of 435 instances, 17 attributes, each attribute value is either *yes* or *no*. The decision attribute value is either *democrat* or *republic*. Since there are 392 missing values, $DD(\Phi_{congress})$ consists of about 10^{120} ($\fallingdotseq 2^{392}$) derived DISs. In case IV, the total execution time SUM is slightly larger, but in case V, we ceased the execution by the NIS-Apriori algorithm, because of its too long execution time. In this example, the tNIS-Apriori algorithm is essential. We have to choose the tNIS-Apriori algorithm for handling the case V.

4.1 Discussion

Of course, the execution time of two algorithms depends upon the details of the algorithms and the characteristics of the data sets. The most time-consuming part of Algorithms 1-3 is 'to generate $SubIMP_i$ by using $Rest$'. We generate

Table 4. The execution time (sec) of the tNIS-Apriori algorithm for the Mammographic data set. The column SUM indicates the summation of two cases.

Case	NIS-Apriori	tNIS-Apriori		
	(sec)	SUM (sec)	Dec=0 ($benign$) (sec)	Dec=1 ($malignant$) (sec)
I	19.9	22.8	11.1	11.7
II	76.3	77.8	33.5	44.2
III	164.9	185.0	92.3	92.7
IV	540.6	615.9	280.3	335.6
V	889.9	1112.3	495.6	616.7

Table 5. The execution time (sec) of the tNIS-Apriori algorithm for the Congressional Voting data set. The column SUM indicates the summation of two cases.

Case	NIS-Apriori	tNIS-Apriori		
	(sec)	SUM (sec)	Dec=dem($ocrat$) (sec)	Dec=rep($ublic$) (sec)
I	424.1	308.7	27.3	281.4
II	1620.8	1369.5	444.1	925.4
III	3065.5	2616.3	988.6	1627.7
IV	5281.4	5802.9	1806.9	3996.0
V	ceased	7620.0	1999.5	5620.5

Table 6. The execution time (sec) of the DIS-Apriori algorithm with a target descriptor for the Car Evaluation data set. The column SUM indicates the summation of four cases.

Case	Dec=any	SUM	Dec=$unacc$	Dec=acc	Dec=$good$	Dec=$vgood$
Instances	1728	1728	1210	384	69	65
Ratio	100%	100%	70%	22%	4%	4%
I	8.30	13.7	8.09	2.01	1.79	1.81
II	38.26	33.83	15.48	14.87	1.79	1.69
III	255.23	177.37	15.23	158.57	1.72	1.85
IV	3343.52	2014.33	23.71	1513.74	240.09	213.08
V	6004.56	4043.02	25.09	2103.51	1065.04	849.38

$SubIMP_i$ instead of using IMP_i. This strategy is based on Property 4 in Section 2.1. For major rule generation, the amount of $|SubIMP_i|$ is generally small. Actually, in Mammographic data set, we focus on implications occurring more than 240 times or 96 times. The amount of such implications is small. For minor rule generation, the amount of $|SubIMP_i|$ generally becomes large. In the case V in the Mammographic data set, we need to focus on implications occurring 1 time, namely the *support* constraint is meaningless. We cannot remove any implications satisfying $accuracy(\tau) < \beta$.

In the generation of $SubIMP_i$, we actually pick up all condition descriptors appearing in $Rest$ at first, then we add each of them to implications in $Rest$ and we remove implications which are not in the original table. This manipulation is the most complicated part in the SQL procedure. For example, in certain rule

generation (case V) from the Mammographic data set, $|Rest|$=192 (implications with two condition descriptors) and 20 condition descriptors are picked up. The amount of the candidates of implications is 1068. From these 1068 implications, we generate $SubIMP_{3,[Severity,0]}$ which consists of 987 implications. This seems large amount of implication, however the amount of IMP_3 is huge, because there are about 9600 implications (there are 960 instances and the selection of 3 attributes is ${}_5C_3$=5*4*3/3*2*1=10 cases). Even though there are the same implications in 9600 implications, the amount of IMP_3 is much larger than that of $SubIMP_{3,[Severity,0]}$.

The above manipulation seems to be related to the amount of decision attribute values. In the Pittsburgh Bridges data set [2], there are six decision attribute values, and the execution of the NIS-Apriori algorithm was ceased, because of too long execution time. In case V, $|SubIMP_3|$ in certain rule generation exceeds 10000 implications, and the tNIS-Apriori was essential in this case, too.

In DISs, we also executed the DIS-Apriori algorithm with a target descriptor. Table 6 shows the results of rule generation in DIS, the Car Evaluation data set [2]. In Case I, II, and III, the SUM of four execution times is almost the same as the execution time of Dec=*any*. However in Case IV and V, the summation of four execution times is reduced to about 2/3 of Dec=*any*. In the Balance Scale data set and the Phishing data set [2], we similarly had the same results.

5 Concluding Remarks

This paper proposed the tNIS-Apriori algorithm, which is a NIS-Apriori algorithm with a target descriptor. The merits are the following.
(1) For a fixed decision attribute values, tNIS-Apriori algorithm works much better than NIS-Apriori algorithm.
(2) The tNIS-Apriori algorithm is effective for minor rule generation. Actually in Table 5, the NIS-Apriori algorithm cannot generate rules, but tNIS-Apriori algorithm did them.

The NIS-Apriori is suitable for major rule generation, however it is time-consuming for minor rule generation, because the next properties.
(a) If $support(\tau) < \alpha$ holds, we can decide any redundant implication of τ is not a rule (Property 4 in Section 2.1).
(b) If $support(\tau) \geq \alpha$ and $accuracy(\tau) < \beta$, this τ is not a rule, but some redundant τ' may satisfy $support(\tau') \geq \alpha$ and $accuracy(\tau') \geq \beta$ (Property 5 in Section 2.1).
(c) If we employ the lower threshold value α, most of implications do not satisfy the above (a) and we can not apply the above (a). Furthermore, most of implications satisfy the above (b). Thus, we need to consider large number of redundant implications as candidates of rules.

In order to solve this weak point, we proposed the tNIS-Apriori algorithm. By handling the specified decision descriptor in the tNIS-Apriori algorithm, the candidates of rules are reduced. Thus, we showed the possibility of NIS-Apriori-based minor rule generation. The analysis of the bottlenecks for the execution

time, the improvement of the procedures in SQL, and the evaluation with experiments are still in progress now.

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. Grzymala-Busse, J., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data, *Journal of Intelligent Manufacturing* 16, 565–573 (2005)
4. He, H., Garcia, E.A.: Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering* 21(9), 1263–1284 (2009)
5. Lipski, W.: On databases with incomplete information, *Journal of the ACM* 28(1), 41–70 (1981)
6. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information, *Theoretical Computer Science* 29(1-2), 27–39 (1984)
7. Pawlak, Z.: Systemy Informacyjne: Podstawy Teoretyczne (in Polish), WNT (1983)
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers (1991)
9. 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)
10. Sakai, H., Liu, C., Zhu, X., Nakata, M.: On NIS-Apriori based data mining in SQL, in *Proc. Int'l. Conf. on IJCRS* (Victor Flores et al. eds.), Springer, LNCS 9920, 514–524 (2016)
11. Sakai, H.: Execution logs by RNIA software tools (2016) `http://www.mns.kyutech.ac.jp/~sakai/RNIA`
12. Sakai, H., Nakata, M., Yao, Y.: Pawlak's many valued information system, non-deterministic information system, and a proposal of new topics on information incompleteness toward the actual application, *Studies in Computational Intelligence* 708, 187–204 (2017)
13. Sakai, H., Nakata, M., Watada, J.: NIS-Apriori-based rule generation with three-way decisions and its application system in SQL, *Information Sciences*, (2018) online published https://doi.org/10.1016/j.ins.2018.09.008
14. Shen, K.Y., Sakai, H., Tzeng, G.H.: Comparing two novel hybrid MRDM approaches to consumer credit scoring under uncertainty and fuzzy judgments, *International Journal of Fuzzy Systems*, (2018) online published https://doi.org/10.1007/s40815-018-0525-0
15. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems, *Intelligent Decision Support - Handbook of Advances and Applications of the Rough Set Theory*, 331–362, Kluwer Academic Publishers (1992)
16. Sarawagi, S., Thomas, S., Agrawal, R.: Integrating association rule mining with relational database systems: alternatives and implications, *Data Mining and Knowledge Discovery* 4(2), 89–125 (2000)
17. Yao, Y. Y.: Three-way decisions with probabilistic rough sets, *Information Sciences* 180, 314–353 (2010)