Granulated Tables with Frequency by Discretization and Their Application

Hiroshi Sakai, Zhiwen Jian

Granulated Tables with Frequency by Discretization and Their Application

Hiroshi Sakai and Zhiwen Jian

Graduate School of Engineering, Kyushu Institute of Technology,
Tobata, Kitakyushu 804-8550, Japan
sakai@mns.kyutech.ac.jp, zhiwen.jian389@mail.kyutech.jp

Abstract. We have coped with rule generation from tables with discrete attribute values and extended the Apriori algorithm to the DIS-Apriori algorithm and the NIS-Apriori algorithm. Two algorithms use table data characteristics, and the NIS-Apriori generates rules from tables with uncertainty. In this paper, we handle tables with continuous attribute values. We usually employ continuous data discretization, and we often had such a property that the different objects came to have the same attribute values. We define a *granulated table with frequency* by discretization and adjust the above two algorithms to granulated tables due to this property. The adjusted algorithms toward big data analysis improved the performance of rule generation. The obtained rules are also applied to rule-based reasoning, which gives one solution to the black-box problem in AI.

Keywords: rule generation, the Apriori algorithm, rule-based reasoning, big data analysis and machine learning

1 Introduction

We are applying rough sets [10, 13, 19] to rule generation from table data sets and are adjusting the Apriori algorithm for transaction data sets [1, 2] to table data sets. We proposed a framework termed "NIS-Apriori" [16–18] based on the combination of equivalence classes in rough sets and the effective enumeration of the candidates of rules in the Apriori algorithm.

We term such a table in Table 1 as a *Deterministic Information System* (DIS). Several rough-set based rule generation methods are proposed [5, 10, 13, 15, 19,

Table 1. An exemplary DIS ψ.

Object	P1	P2	P3	Dec
$x1$	c	1	b	d1
$x2$	b	2	b	d1
$x3$	a	2	b	d2
$x4$	a	3	c	d2
$x5$	c	2	c	d3

Table 2. An exemplary DIS ψ with missing values.

Object	P1	P2	P3	Dec
$x1$	c	?	b	d1
$x2$	b	?	b	d1
$x3$	?	2	b	d2
$x4$	a	3	c	?
$x5$	c	2	?	d3

Table 3. An exemplary NIS Φ. Each ? is changed to a set of all possible values.

Object	P1	P2	P3	Dec
$x1$	c	{1,2,3}	b	d1
$x2$	b	{1,2,3}	b	d1
$x3$	{a,b,c}	2	b	d2
$x4$	a	3	c	{d1,d2,d3}
$x5$	c	2	{a,b,c}	d3

21] in DISs. In Table 1, we have an implication τ: [P1,c]$\Rightarrow$ [Dec,d1] from object $x1$. It occurs 1 time for 5 objects, namely $support(\tau)$ (a ratio of occurrence) $=1/5$. It occurs 1 time for 2 objects with [P1,c], namely $accuracy(\tau)$ (a ratio of consistency)$=1/2$. We term such formulas like [P1,c] and [Dec,d1] *descriptors*. We usually specify constraints $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ for $0 < \alpha, \beta \leq 1$, and we see each implication satisfying the constraints as a rule.

To cope with information incompleteness in DISs, *missing values* '?' [6] (Table 2) and a *Non-deterministic Information System* (NIS) [11, 12] (Table 3) were also investigated. In NIS, some attribute values are given as a set of possible attribute values. In Table 3, we interpret $\{1, 2, 3\}$ in $x2$ as that *one of 1, 2, and 3 is the actual value, but there is not enough information to decide it* due to information incompleteness.

If we replace each set of attributes in NIS Φ with one element of the set, we have one possible DIS. In Table 3, there are 243 $(=3^5)$ possible DISs and let $DD(\Phi)$ denote a set of possible DISs. We considered the certain rules and the possible rules from NIS below:
• An implication τ is a certain rule, if τ is a rule in each $\psi \in DD(\Phi)$.
• An implication τ is a possible rule, if τ is a rule in at least one $\psi \in DD(\Phi)$.
This definition seems natural, but the $|DD(\Phi)|$ increases exponentially. However, we proved some properties and developed the NIS-Apriori algorithm, which does not depend upon the number of $|DD(\Phi)|$ [17, 18].

This paper extends the previous frameworks and considers rule generation from tables with continuous attribute values. Furthermore, this paper applies the obtained rules to decision making. The reasoning based on the obtained rules will recover the black-box problem in AI.

This paper's organization is as follows: Section 2 clarifies the discretization of the continuous attribute values and considers an example of the Iris data set [4]. Section 3 proposes granulated tables Γ with frequency by discretization and extends the previous algorithms for NIS to that for Γ. Section 4 applies the extended algorithms to some data sets and shows the improvement of rule generation performance. An application of the obtained rules to decision making is considered. Section 5 concludes this paper.

2 Tables with Continuous Attribute Values

This section briefly examines the discretization of continuous attribute values and considers the case of the Iris data set [4].

2.1 Rules and Discretization of Continuous Attribute Values

To generate rules from tables with continuous attribute values, we need to discretize tables because there may be too many descriptors. There seem to be several methods for discretization [7]. Most of them focus on the optimal discretization specified by the constraints like minimal entropy, equal-interval width, equal-interval frequency, etc. In these researches on discretization, descriptors seem to be obtained as a side effect.

However, our research purpose is to generate rules by specified descriptors. We at first specify descriptors and their intervals, then generate rules by them. Thus, we can have our rules for our specifications.

2.2 An Example of the Iris Data Set

The Iris data set consists of 150 objects, four condition attributes $\{spl, spw, pel, pew\}$ (each attribute value of them is continuous), one decision attribute $class$ whose attribute value is one of $setosa$, $versicolor$, and $virginica$. Fig. 1 is a part of the Iris data set.

	A	B	C	D	E	F
1	object	spl	spw	pel	pew	class
2	1	5.1	3.5	1.4	0.2	setosa
3	2	4.9	3	1.4	0.2	setosa
100	99	5.1	2.5	3	1.1	versicolor
101	100	5.7	2.8	4.1	1.3	versicolor
150	149	6.2	3.4	5.4	2.3	virginica
151	150	5.9	3	5.1	1.8	virginica

Fig. 1. A part of the Iris data set.

	small	medium	large
spl	5.5<	5.5<= & < 6.7	6.7<=
spw	3<	3<= & < 4	4<=
pel	2.5<	2.5<= & < 5	5<=
pew	1<	1<= & < 2	2<=

Fig. 2. A definition of the discretization of continuous attribute values.

Fig. 2 shows the discretization specification; namely, we want to generate rules using the attribute values $small$, $medium$, and $large$ for each attribute. Here, every object is identified as one element of the Cartesian product $\{small, medium, large\}^4 \times \{setosa, versicolor, virginica\}$, whose number of elements is 243 ($=3^5$). On the other hand, the number of objects is 150. Thus, there exist many such elements of the Cartesian product that do not correspond to objects.

```
[1,(('spl','m'),('spw','s'),('pel','l'),('pew','m'),('class','versicolor')),1]
[2,(('spl','m'),('spw','m'),('pel','l'),('pew','m'),('class','virginica')),3]
[3,(('spl','l'),('spw','s'),('pel','l'),('pew','l'),('class','virginica')),2]
[4,(('spl','s'),('spw','l'),('pel','s'),('pew','s'),('class','setosa')),1]
[5,(('spl','l'),('spw','s'),('pel','l'),('pew','m'),('class','virginica')),3]
[6,(('spl','m'),('spw','l'),('pel','s'),('pew','s'),('class','setosa')),3]
[7,(('spl','m'),('spw','s'),('pel','m'),('pew','m'),('class','versicolor')),27]
[8,(('spl','m'),('spw','m'),('pel','l'),('pew','l'),('class','virginica')),7]
[9,(('spl','m'),('spw','m'),('pel','m'),('pew','m'),('class','virginica')),2]
[10,(('spl','s'),('spw','m'),('pel','s'),('pew','s'),('class','setosa')),42]
[11,(('spl','m'),('spw','s'),('pel','m'),('pew','l'),('class','virginica')),1]
[12,(('spl','m'),('spw','s'),('pel','m'),('pew','m'),('class','virginica')),2]
[13,(('spl','l'),('spw','s'),('pel','m'),('pew','m'),('class','versicolor')),1]
[14,(('spl','s'),('spw','s'),('pel','s'),('pew','s'),('class','setosa')),2]
[15,(('spl','m'),('spw','m'),('pel','m'),('pew','m'),('class','versicolor')),10]
[16,(('spl','l'),('spw','m'),('pel','l'),('pew','m'),('class','virginica')),2]
[17,(('spl','s'),('spw','m'),('pel','m'),('pew','m'),('class','versicolor')),1]
[18,(('spl','l'),('spw','m'),('pel','l'),('pew','l'),('class','virginica')),15]
[19,(('spl','m'),('spw','s'),('pel','l'),('pew','m'),('class','virginica')),8]
[20,(('spl','l'),('spw','m'),('pel','l'),('pew','m'),('class','versicolor')),1]
[21,(('spl','s'),('spw','s'),('pel','m'),('pew','m'),('class','versicolor')),5]
[22,(('spl','m'),('spw','s'),('pel','l'),('pew','l'),('class','virginica')),4]
[23,(('spl','m'),('spw','m'),('pel','s'),('pew','s'),('class','setosa')),2]
[24,(('spl','s'),('spw','s'),('pel','m'),('pew','m'),('class','virginica')),1]
[25,(('spl','l'),('spw','m'),('pel','m'),('pew','m'),('class','versicolor')),4]
```

Fig. 3. Discretized table with frequency.

Table 4. A relationship between data sets and the discretized data sets. Here, type I: tables with continuous values, type II: tables with discrete values, type III: tables with missing values, #object: the number of objects, #con: the number of condition attributes, #Cartesian: the number of elements of the Cartesian product, #object_discre: the number of objects after discretization.

data sets	type	#object	#con	#Cartesian	#object_discre
Iris	I	150	4	243 ($=3^4 \times 3$)	25
Wine quality [4]	I	4898	11	1240029 ($=3^{11} \times 7$)	564
Htru2	I	17898	8	13122 ($=3^8 \times 2$)	134
Phishing [4]	II	1353	9	59049 ($=3^9 \times 3$)	724
Car Evaluation [4]	II	1728	6	27648 ($=4^4 \times 3^3$)	1728
Suspicious Network [3]	II	39427	51	-	39427
Mammographic [4]	III	961	5	3200 ($=4^3 \times 5^2 \times 2$)	301
Congress Voting [4]	III	435	16	131072 ($=2^{17}$)	342

We examined the relationship between the Iris data set and the elements of the Cartesian product. Fig. 3 shows lists of the sequential number, the Cartesian product element, and the duplicated number of objects. For example, the 10th list represents 42 objects for 150 objects. We may say that 42 objects are granulated to one granule represented by the 10th Cartesian product element. Thus, we have a discretized table with re-numbered 25 objects from a table with 150 objects. This phenomenon seems interesting because we can handle re-numbered 25 objects for a total of 150 objects.

We dealt with this phenomenon for other data sets with continuous attribute values. Table 4 shows the results. For example, there are 17898 objects in the

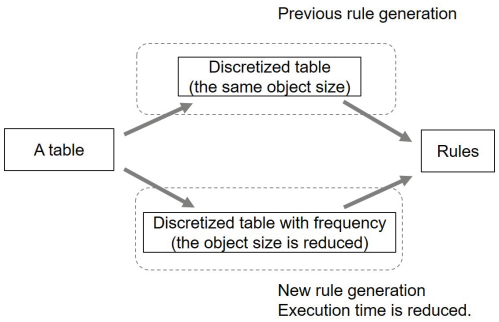

Fig. 4. Previous rule generation and new rule generation.

Htru2 data set [4]. We employed a similar discretization, which divides each attribute into three classes, as Fig. 2. Then, each object is identified with an element of the Cartesian product with 13122 elements. Furthermore, 17898 objects are granulated to 134 elements in the Cartesian product.

Remark 1. In tables with continuous attribute values, we specify descriptors for discretization. If the number of discretized objects is much smaller than that of the original data sets, we will reduce the execution time of rule generation. Of course, we have the same rules as that of the original data set. It will be meaningful to consider the new rule generation in Fig. 4.

The strategy in Fig. 4 will be related to researches on rough sets (coarse classes) [13, 19], the Infobright technology [20], granular computing [14], and the zoom out operation [21]. As for the Car Evaluation and Suspicious Network data sets (type II) in Table 4, each tuple was different, and we cannot reduce the number of objects. Remark 1 will be applied to tables with continuous attribute values, and it may not be useful to tables with discrete attribute values.

3 Rule Generation from Discretized Tables with Frequency

This section defines a granulated table Γ with frequency.

Definition 1. *A discretized table Γ with frequency (from a DIS ψ) consists of the following:*

1. *A finite set AT of attributes,*
2. *A finite set $DESC_A$ of descriptors for $A \in AT$,*
3. *A pair $(LIST, freq)$ ($LIST \in \Pi_{A \in AT} DESC_A$: the Cartesian product of sets of descriptors, and $freq\,(>0)$: the number of objects in ψ satisfying $LIST$),*
4. *We assign a number num to each pair and see a tuple $(num, LIST, freq)$ as an object in Γ.*

Input: Table data set DIS ψ, decision attribute Dec, threshold values α, β.
Output: A set $Rule(\psi)$ of minimal rules.
1: $Rule(\psi) \leftarrow \{\}$; $i \leftarrow 1$;
2: create a set CAN_1 of candidates of rules with 1 condition;
3: **while** ($|CAN_i| \geq 1$) **do**
4: $Rest_i \leftarrow \{\}$; $Rule_i \leftarrow \{\}$;
5: **for all** $\tau_{i,j} \in CAN_i$ **do**
6: **if** $support(\tau_{i,j}) \geq \alpha$ **then**
7: **if** $accuracy(\tau_{i,j}) \geq \beta$ **then** add $\tau_{i,j}$ to $Rule_i$; **else** add $\tau_{i,j}$ to $Rest_i$;
8: **end if**
9: **end if**
10: **end for**
11: $i \leftarrow i + 1$;
12: create CAN_i (candidates of rules with i-th conditions) from $Rest_{i-1}$ and $Rest_1$;
13: **end while**
14: **return** $Rule(\psi) = \cup_{k<i} Rule_k$

Fig. 5. The DIS-Apriori algorithm adjusted to table data set DIS ψ [9].

Fig. 3 is an example of Γ. Now, we consider rule generation from Γ. Previously, we have investigated rules and rule generators [17, 18] in the following.

1. Rules in DIS ψ and the DIS-Apriori rule generator,
2. Certain rules and possible rules in NIS Φ and the NIS-Apriori rule generator,
3. Decision-making tool based on the obtained rules.

We identify each descriptor [attribute,value] as an item and adjusted the Apriori algorithm for transaction data sets to that of table data sets. The overview of the adjusted DIS-Apriori algorithm is in Fig. 5. There is usually one decision attribute in every table, and we can see one itemset defines one implication. Furthermore, in the line 12 in Fig. 5, we have proved that CAN_i can be generated from $Rest_{i-1}$ and $Rest_1$ [9]. Due to these characteristics, we reduced the number of candidates of rules and the execution time of rule generation.

Remark 2. The following properties are related to the DIS-Apriori algorithm.

1. We replace DIS ψ with NIS Φ, *support* and *accuracy* values with *minsupp* and *minacc* values [17, 18], respectively. Then, this algorithm generates all minimal certain rules.
2. We replace DIS ψ with NIS Φ, *support* and *accuracy* values with *maxsupp* and *maxacc* values [17, 18], respectively. Then, this algorithm generates all minimal possible rules.
3. We term the above two algorithms the NIS-Apriori algorithm.
4. Both DIS-Apriori and NIS-Apriori algorithms are logically sound and complete for rules. They generate rules without excess and deficiency.

To handle a discretized table Γ with frequency, we revise the calculation of *support* and *accuracy* values in Fig. 5. To calculate them, we are handling

equivalence classes defined by the concept of rough sets. For example, in Table 1, we have equivalence classes $\{x1, x5\}$ for [P1,c] and $\{x1, x2\}$ for [Dec,d1], respectively. We can easily know that τ: [P1,c]$\Rightarrow$ [Dec,d1] is supported by $\{x1\}$ ($=\{x1, x5\} \cap \{x1, x2\}$). Here, the occurrence of τ is 1, however in Γ, we need to count the frequency $freq(x1)$ of $x1$. Due to this consideration, we have the next Remark 3 for handling Γ.

Remark 3. In a DIS ψ, if an implication τ is supported by an equivalence class $\{x1, x2, \cdots, x_n\}$, τ is supported by n objects. However, in Γ, τ is supported by $freq(x1)+freq(x2)+\cdots+freq(x_n)$ objects (Here, $freq(x_i)$ means the frequency of x_i in Γ). If we replace the number of occurrence (i.e., 1) of one object x_i with $freq(x_i)$, we can have the DIS-Apriori algorithm for handling Γ.

4 An Apriori-based Rule Generator for Γ and Some Experiments

We revised rule generation programs, the DIS-Apriori algorithm for Γ and the NIS-Apriori algorithm for Γ, in Python based on Remarks 2-3. For simplicity, we omit the details of the NIS-Apriori algorithm for Γ and show the execution time in Table 5.

Table 5. A Comparison of the execution time: the original tables and the granulated tables. As for Iris, we coped with four cases. We duplicated the original table by ten times, 100 times, and 1000 times.

Table	support	accuracy	Original		Granulated	
			#object	exec (sec)	#object	exec (sec)
Iris	0.01	0.9	150	0.022	25	0.018
Iris1500	0.01	0.9	1500	0.030	25	0.020
Iris15000	0.01	0.9	15000	0.206	25	0.022
Iris150000	0.01	0.9	150000	2.162	25	0.022
Wine quality	0.001	0.5	4898	13.365	564	10.962
Htru2	0	0.7	17898	3.167	134	0.175
Mammographic						
(certain rule)	0	0.8	961	0.343	299	0.297
(possible rule)	0	0.8	961	0.375	299	0.349
Congress Voting						
(certain rule)	0.1	0.7	435	0.434	342	0.417
(possible rule)	0.1	0.7	435	0.448	342	0.419

Due to Table 5, we know the new rule generator is more effective in three cases of Iris15000, Iris150000, and Htru2. In the granulated tables Γ from Iris15000 and Iris150000, the number of objects is the same, and only the frequency is changed. The rule generation process is the same as that of Iris, and the calculation of *support* and *accuracy* is slightly changed. Thus, the execution time of rule

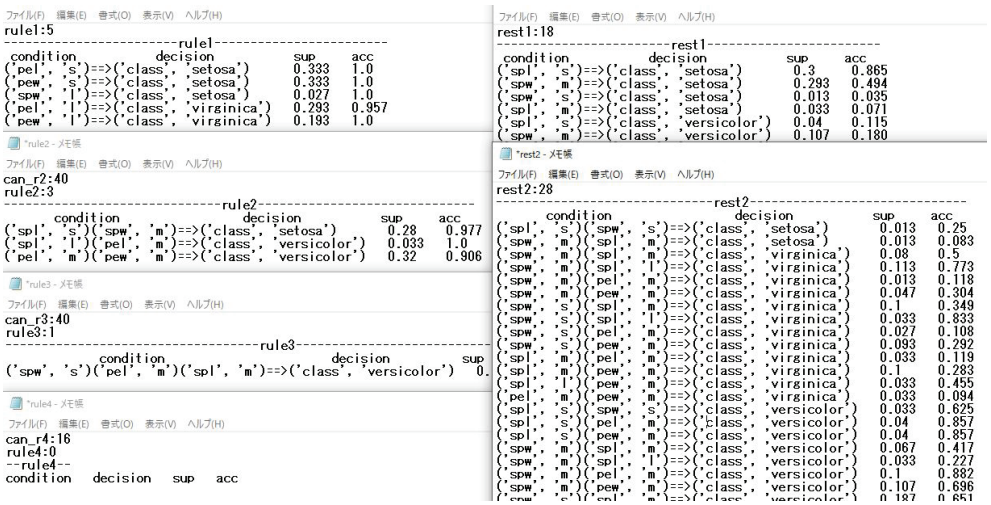

Fig. 6. Obtained rules ($support(\tau) \geq 0.01$, $accuracy(\tau) \geq 0.9$) and $Rest_1$, $Rest_2$ from Iris150000.

generation is almost constant. We think that to consider Γ will be meaningful for big data analysis. Fig. 6 shows the obtained rules (the left hand side) and $Rest_1$, $Rest_2$ (the right hand side). By using $Rest_1$, $Rest_2$, $\cdots$, we can reduce the number of candidates of rules. This reduction causes to reduce the execution time of rule generation.

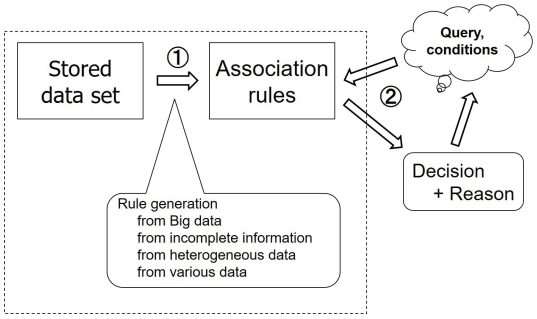

Fig. 7. A chart of rule-based reasoning.

Now, we consider the application of the obtained rules. We quickly think the rule-based decision making in Fig. 7. We have coped with ① and have developed the environment of rule generation. We also need to manage ②, i.e., decision making by the obtained rules. The decision making in Fig. 7 is based on the

applied rules, and the applied rule supports the reasoning. Thus, the reason is apparent. This strategy will recover the black-box problem in AI. We need to clarify the following subjects:

1. How do we select one rule if there are some applicable rules?
2. How do we have a decision if there is no candidate for a rule?
3. How do we think that the different results may be concluded?

For the Suspicious Network data set in Table 4, we employed the $lift$ value for selecting one rule. We employed three-cross validation for 39427 objects and applied the obtained rules. This procedure is based on Fig. 7, and the averaged 94% correct estimation was obtained [8]. The research on Fig. 7 is in progress now.

5 Concluding Remarks

We considered discretization for tables with continuous attribute values and proposed granulated tables with frequency. In some cases, we can reduce the number of objects. This property causes to reduce the execution time of rule generation. We implemented a new rule generator and examined that the adjusted algorithms for big data analysis improved rule generation performance. The obtained rules are also applied to rule-based reasoning, which gives one solution to the black-box problem in AI. We need to improve much more comparative analysis and sensitive analysis comprehensively.

Acknowledgment. The authors would be grateful to the anonymous referees for their useful comments. This work is supported by JSPS (Japan Society for the Promotion of Science) KAKENHI Grant Number JP20K11954.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. Proc. VLDB'94, Morgan Kaufmann, (1994) 487–499
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. Advances in Knowledge Discovery and Data Mining AAAI/MIT Press (1996) 307–328
3. Bigdata Challenge. https://knowledgepit.ml/ [Accessed July 14, 2019]
4. Frank, A., Asuncion, A.: UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science (2010). `http://mlearn.ics.uci.edu/MLRepository.html` [Accessed July 10, 2019]
5. Greco, S., Matarazzo, B., Słowiński, R.: Granular computing and data mining for ordered data: The dominance-based rough set approach. Encyclopedia of Complexity and Systems Science (R.A. Meyers, Ed.), Springer (2009) 4283–4305
6. Grzymała-Busse, J.W., Werbrouck, P.: On the best search method in the LEM1 and LEM2 algorithms. Incomplete Information: Rough Set Analysis, Studies in Fuzziness and Soft Computing, vol. 13, Springer (1998) 75–91

7. Grzymała-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. Int'l. J. Intelligent Systems, vol. 16, Wiley (2001) 29–38
8. Jian, Z., Sakai, H., Watada, J., Roy, A., Hassan, M. B.: An Apriori-based data analysis on suspicious network event recognition. Proc. IEEE Big Data 2019 (2019), 5888–5896
9. Jian, Z., Sakai, H., et. al.: An adjusted Apriori algorithm to itemsets defined by tables and an improved rule generator with three-way decisions. Proc. IJCRS2020, Springer LNCS, vol. 12179 (2020), 95–110
10. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutorial. Rough Fuzzy Hybridization: A New Method for Decision Making (S. K. Pal, A. Skowron, Eds.), Springer (1999) 3–98
11. Lipski, W.: On databases with incomplete information. Journal of the ACM, vol. 28(1) (1981) 41–70
12. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science, vol. 29(1-2) (1984) 27–39
13. Pawlak, Z.: Rough sets. Int'l. Journal of Computer and Information Sciences, vol. 11(5) (1982) 341–356
14. Pedrycz, W.: Granular computing for data analytics: A manifesto of human-centric computing. IEEE/CAA Journal of Automatica Sinica, 5(6) (2018) 1025–1034
15. Riza, L.S., Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Ślęzak, D., Benítez, J.M.: Implementing algorithms of rough set theory and fuzzy rough set theory in the R package RoughSets. Information Sciences, vol. 287(10) (2014) 68–89
16. Sakai, H.: Execution logs by RNIA software tools. http://www.mns.kyutech.ac.jp/~sakai/RNIA [Accessed July 10, 2019]
17. Sakai, H., Nakata, M.: Rough set-based rule generation and Apriori-based rule generation from table data sets: a survey and a combination. CAAI Transactions on Intelligence Technology, vol. 4(4) (2019) 203–213
18. Sakai, H., Nakata, M., Watada, J.: NIS-Apriori-based rule generation with three-way decisions and its application system in SQL. Information Sciences, vol. 507 (2020) 755–771
19. 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 (R. Słowiński, Ed.), Kluwer Academic Publishers (1992) 331–362
20. Ślęzak, D., Eastwood, V.: Data warehouse technology by Infobright. Proc. ACM SIGMOD 2009 (2009) 841–846
21. Yao, Y. Y., Liau, C., Zhong, N.: Granular computing based on rough sets, quotient space theory, and belief functions. Proc. ISMIS 2003, Springer LNCS 2871 (2003) 152–159