Kansei Engineering International Vol.6 No.3 pp.19-24 (2006)

ORIGINAL ARTICLES

A SIMPLE RECOMMENDATION SYSTEM BASED ON ROUGH SET THEORY

Yasuo KUDO*, Shohei AMANO*, Takahiro SEINO** and Tetsuya MURAI***

** Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran-shi 050-8585, Japan*
*** COMTECH 2000 Corporation, 9-5-8 Sakaedori, Shiroishi-ku, Sapporo-shi 003-0021, Japan*
**** Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo-shi 060-0814, Japan*

Abstract: We propose a simple recommendation system based on rough set theory. When the user searches some products by a product retrieval system, if the user does not have enough information of the products, it is very difficult to represent relevant queries. Thus, by data mining and *Kansei* information processing, product recommendation system estimates user's preference from user's queries, and provides information about products that the user may prefer. Our recommendation method constructs decision rules from user's query, and recommends some products by estimating implicit conditions of products based on decision rules. Recommended products do not agree with the query, however, recommended products satisfy estimated implicit conditions, and therefore the user may prefer recommended products. We also propose a criterion to evaluate recommended products based on certainty and coverage of decision rules. Moreover, we evaluate our recommendation method and evaluation method by experiment, and discuss issues for improvement of our methods based on results of the experiment.
Keywords: *Recommendation system, Rough set, Decision rule, Certainty, Coverage*

1. INTRODUCTION

Product recommendation system estimates user's preference from user's queries, and provides information about products that the user may prefer by data mining techniques. For estimating user's preference, there are many studies about combination of *Kansei* engineering and product recommendation system (for example, [1, 5, 10]). Rough set theory [6, 7] is a mathematical theory of approximation of concepts, and various applications of rough set theory in *Kansei* engineering, in particular, applications of extracting decision rules are also widely studied (for detail, see [4]).

In this paper, we propose a simple recommendation system based on rough set theory. The term "simple" intends that our method dose not use any extra information except for product data and user's query. Our recommendation method constructs decision rules from user's query, and recommends some products by estimating implicit conditions of products that the user wants to search based on decision rules. Recommended products do not agree with the query, however, the products satisfy estimated implicit conditions, and therefore the user may prefer the products. Moreover, we propose a criterion to evaluate recommended products based on certainty and coverage of decision rules.

The rest of this paper is organized as follows. In section 2, we review rough set theory briefly. In section 3, we propose a recommendation algorithm using decision rules and a criterion for evaluating the degree of recommendation of products. In section 4, we explain evaluation experiment of proposed methods and results of the experiment. Finally, we summarize this paper in section 5. This paper is a revised and expanded version of two proceeding papers of the authors [2, 3].

2. BACKGROUNDS

We review rough set theory briefly, in particular, decision tables, relative reducts and decision rules. Note that contents of this section are entirely based on [4].

2.1 Decision tables

In applications of rough set theory, data of objects are generally expressed by combination of attributes and its values. The set of such combination about objects is called an *information system*. Formally, an information system is the following quadruple:

$$\langle U, AT, V, \rho \rangle,$$

where U is the set of all objects, and called the universe, AT is the set of attributes, V is the set of values, and $\rho : U \times AT \to V$ is a function that assigns the attribute value $\rho(u,\alpha) \in V$ to the object u at the attribute α.

If the set of attributes AT is divided into the set of condition attributes C and the set of decision attributes D such that $AT = C \cup D$ and $C \cap D$ is empty, the information system is called a *decision table*. The set of decision attributes provides a partition $\mathbf{D}_1, \cdots \mathbf{D}_m$ of the universe, and each $\mathbf{D}_i$ is called a *decision class*.

Example 1. Table 1 illustrates an information system about eight cars c1, ⋯ c8. By dividing AT into the following two parts, we have a decision table: condition attributes C = {Maker, Type, WD, ED} and the decision attribute D = {TM}, where WD, ED and TM are abbreviations of

Received 2006.3.7
Accepted 2006.7.15

Wheel Drive, Engine Displacement, and Transmission, respectively. The decision attribute TM provides two decision classes: the set of automatic transmission cars $\mathbf{D}_1$ = {c1, c3, c4, c5, c6, c7}, and the set of manual transmission cars $\mathbf{D}_2$ = {c2, c8}.

2.2 Relative reducts and decision rules

Relative reducts of the given decision table are minimal sets of condition attributes to classify all objects into decision classes correctly. We use the *discernibility matrix* proposed by Skowron and Rauser [8] to construct all reducts of the set of decision attributes D. The defensibility matrix is an $n \times n$ matrix whose element δ_{ij} at i-th row and j-th column is defined by

$$\delta_{ij} = \begin{cases} \{\beta \in C \mid \rho(x_i, \beta) \neq \rho(x_j, \beta)\}, & \\ & \text{if } \exists \alpha \in D \text{ s.t. } \rho(x_i, \alpha) \neq \rho(x_j, \alpha), \quad (1) \\ \phi, & \text{otherwise,} \end{cases}$$

where n is the number of objects in U.

The set of attributes δ_{ij} illustrates that we can distinguish the objects x_i and x_j by comparing just one attribute $\beta \in \delta_{ij}$. On the other hand, if δ_{ij} is empty, we need not (or can not) distinguish x_i and x_j. In particular, when x_i and x_j are inconsistent, that is, x_i and x_j are in different decision classes even though all values of condition attributes are identical, we can not distinguish x_i and x_j by any condition attributes. Actually, such inconsistent objects are ignored when calculating reducts by the dicsernibility matrix.

All relative reducts $B \subseteq C$ are calculated from the discernibility matrix. Relative reducts $B \subseteq C$ satisfy the following property:

- For any $1 \leq i, j \leq$ n, if δ_{ij} is not empty, then $B \cap \delta_{ij}$ is also not empty.

Each relative reduct $B = \{\beta_1, \cdots, \beta_m\}$ provides *decision rules*. In example 1, we have the following three reducts: {Maker, ED}, {Type, WD} and {ED, WD}. Thus, for example, we have the following four decision rules from the reduct {Type, WD}:

- [Type=Compact] $\wedge$ [WD=2WD]$\Rightarrow$[TM=AT]
- [Type=Compact] $\wedge$ [WD=4WD]$\Rightarrow$[TM=AT]
- [Type=Sedan] $\wedge$ [WD=2WD]$\Rightarrow$[TM=AT]
- [Type=Sedan] $\wedge$ [WD=4WD]$\Rightarrow$[TM=Manual]

Usually, quality of each decision rule $\Delta \Rightarrow \Gamma$ is numerically evaluated by *certainty* $\mathrm{Cer}(\Gamma \mid \Delta)$ and *coverage* $\mathrm{Cov}(\Gamma \mid \Delta)$ as follows:

$$\mathrm{Cer}(\Gamma \mid \Delta) = \frac{\mathrm{Card}\,(\|\Gamma \wedge \Delta\|)}{\mathrm{Card}\,(\|\Delta\|)}, \quad (2)$$

$$\mathrm{Cov}(\Gamma \mid \Delta) = \frac{\mathrm{Card}\,(\|\Gamma \wedge \Delta\|)}{\mathrm{Card}\,(\|\Gamma\|)}, \quad (3)$$

Table 1 : Information system in example 1

	Maker	Type	ED	WD	TM
c1	Toyota	Compact	1500	4WD	AT
c2	Toyota	Sedan	2000	4WD	Manual
c3	Toyota	Sedan	1500	2WD	AT
c4	Nissan	Sedan	2000	2WD	AT
c5	Nissan	Compact	1500	2WD	AT
c6	Nissan	Compact	1500	4WD	AT
c7	Mazda	Compact	1500	4WD	AT
c8	Mazda	Sedan	2000	4WD	Manual

where $\|\Gamma\|$ means the set of objects that agree with the condition Γ, and Card(S) is the number of elements of the set S. The certainty $\mathrm{Cer}(\Gamma \mid \Delta)$ evaluates correctness of the decision rule $\Delta \Rightarrow \Gamma$. On the other hand, the coverage $\mathrm{Cov}(\Gamma \mid \Delta)$ represents the ratio of objects that agree with both Δ and Γ in the set of objects that agree with Γ. $\mathrm{Cov}(\Gamma \mid \Delta)$ evaluates "relevance" of the rule $\Delta \Rightarrow \Gamma$ as an explanation of the decision class that agree with Γ.

3. RECOMMENDATION AND EVALUATION OF PRODUCTS BASED ON ROUGH SET THEORY

In this section, we introduce a recommendation method using decision rules and an evaluation method of recommendation based on certainty and coverage of decision rules. Our recommendation method constructs decision rules from user's query, and recommends some products by estimating implicit conditions of products that the user searches. Recommended products do not agree with the query, however, the products satisfy estimated implicit conditions, and therefore the user may be possible to prefer recommended products. Moreover, using certainty and coverage of decision rules, we evaluate the degree of recommendation of each product recommended by our method.

3.1 A recommendation algorithm using decision rules

We propose a recommendation algorithm using decision rules. The main idea is to estimate user's implicit conditions of searching products from user's query and constructed decision rules. Recommend products satisfy estimated implicit conditions.

A recommendation algorithm using decision rules

Input: A query Q with the following form:

$$[\alpha_1 = x_1] \wedge \cdots \wedge [\alpha_n = x_n]$$

where each α_i is an attribute, and x_i is a value of α_i.

Output: A set of recommended products Rec $\subseteq U$.

Step1. Set Rec = $\emptyset$.

Step2. If $\|Q\| \neq \emptyset$, we set $Q_w = Q$. Otherwise, we reject some conditions $[\alpha_i = x_i]$ in Q as little as possible, and construct a weakened query Q_w until we have $\|Q_w\| \neq \emptyset$.

Step3. Construct the set of decision attributes

$$D_w = \{\gamma_1, \cdots, \gamma_k\},$$

where each γ_i is the attribute that appears in the (weakened) query Q_w. We denote the decision class that agrees with Q_w by $\mathbf{D}^*$.

Step4. Construct all relative reducts $B \subseteq C$ of the decision classes by the discernibility matrix.

Step5. For all reducts $B \subseteq C$ and all products $u \in \mathbf{D}^*$, repeat the following steps:

1. For each relative reduct $B = \{\beta_1, \cdots, \beta_m\}$ and each product $u \in \mathbf{D}^*$, construct the following decision rule:

$$[\beta_1 = y_1] \wedge \cdots \wedge [\beta_m = y_m] \Rightarrow [\gamma_1 = x_1] \wedge \cdots \wedge [\gamma_k = x_k] \quad (4)$$

where $\rho(u, \beta_i) = y_i$ for each β_i. We denote the decision rule (4) by DR(u, Q_w).

2. For each product $v \in \mathbf{D_j}$ ($\mathbf{D_j} \neq \mathbf{D}^*$) that agrees with the antecedents of the decision rule (4), construct the following decision rules:

$$[\beta_1 = y_1] \wedge \cdots \wedge [\beta_m = y_m] \Rightarrow [\gamma_1 = z_1] \wedge \cdots \wedge [\gamma_k = z_k] \quad (5)$$

where $\rho(v, \alpha_i) = z_i$ for each α_i. We denote the conclusion of the decision rule (5) by Q_r, and similar to the decision rule (4), we denote the decision rule (5) by DR(v, Q_r).

3. Add all products $v \in \mathbf{D_j}$ used at 2. to the set of recommended products .

Step6. Output Rec.

Each recommended product $x \in$ Rec does not agree with the (weakened) query Q_w, however, it agrees with the antecedent of a decision rule DR(u, Q_w) of $u \in \mathbf{D}^*$:

$$[\beta_1 = y_1] \wedge \cdots \wedge [\beta_m = y_m].$$

We regard this formula as an *implicit condition IC* of user's query, and assume that *IC* illustrates a hidden condition of user's favorite products. This corresponds to *Abduction* of *IC* from Q_w and DR(u, Q_w). Moreover, by each $x \in$ Rec, we imply the following characteristics Q_r from *IC* and another decision rule DR(u, Q_r):

$$[\gamma_1 = z_1] \wedge \cdots \wedge [\gamma_k = z_k].$$

Note that Q_w and Q_r are different characteristics, that is, there is at least one decision attribute γ_i that has different values in Q_w and Q_r. This reasoning step corresponds to *Deduction* of Q_r from *IC* and DR(x, Q_r).

Note that, as a starting point of recommendation, we need to have at least one product that agrees with user's query. Thus, if there is no product that agrees with user's actual query Q, we need to construct a weakened query Q_w to get some products as a starting point.

Example 2. This example is continuation of Example 1. Suppose that a user searches cars from Table 1 by the following query Q:

$$[\text{Maker} = \text{Mazda}] \wedge [\text{TM} = \text{Manual}].$$

We calculate Rec by the recommendation algorithm as follows:

Step1. We set Rec = $\emptyset$.

Step2. We search products which agree with Q, and find the product c8. Thus, we set $Q_w = Q$.

Step3. We get the set of decision attributes

$$D = \{\text{Maker, TM}\},$$

and the following five decision classes:

- $\mathbf{D}_1 = \{c1, c3\}$ (Toyota, AT),
- $\mathbf{D}_2 = \{c2\}$ (Toyota, Manual),
- $\mathbf{D}_3 = \{c4, c5, c6\}$ (Nissan, AT),
- $\mathbf{D}_4 = \{c7\}$ (Mazda, AT),
- $\mathbf{D}^* = \{c8\}$ (Mazda, Manual).

Step4. We calculatett all reducts, and get one reduct:

$$B = \{\text{Type, ED}\}.$$

Note that the products c1, c6 and c7 are inconsistent because these have the same condition values. Similarly, c2 and c8 are also inconsistent. Therefore, these five products are actually ignored when calculating reducts.

Step5. We calculate recommended products as follows:

1. Using the product c8 $\in \mathbf{D}^*$, we get the following rule DR(c8, Q_w):
$$[\text{Type} = \text{Sedan}] \wedge [\text{ED} = 2000] \Rightarrow [\text{Maker} = \text{Mazda}] \wedge [\text{ED} = \text{Manual}].$$
2. We get two products c2 and c4 that agree with the antecedent of DR(c8, Q_w). Then, we get the following rule DR(c2, Q_{r1}):
$$[\text{Type} = \text{Sedan}] \wedge [\text{ED} = 2000]. \Rightarrow [\text{Maker} = \text{Toyota}] \wedge [\text{TM} = \text{Manual}].$$
Similarly, we also get the following decision rule DR(c4, Q_{r2}):
$$[\text{Type} = \text{Sedan}] \wedge [\text{ED} = 2000]. \Rightarrow [\text{Maker} = \text{Nissan}] \wedge [\text{TM} = \text{AT}].$$
3. We add c2 and c4 to Rec.

Step6. Consequently, we output the following set as recommended products Rec = {c2, c4}.

In this calculation of recommendation, from the user's query, we implied user's implicit condition *IC*:

$$[\text{Type} = \text{Sedan}] \wedge [\text{ED} = 2000]$$

at Step5 by abduction from Q_w and DR(c8, Q_w), and we assumed that the user may also prefer products that agree with this condition. Then, we implied the following two characteristics

Q_{r1} : [Maker = Toyota] $\wedge$ [TM = Manual],

Q_{r2} : [Maker = Nissan] $\wedge$ [TM = AT],

and get c2 and c4 as recommended products that agree with implicit conditions, even though these are not agree with the original query.

3.2 Evaluation of recommended products

We also propose an evaluation method of recommended products by the recommendation algorithm in section 3.1. The main idea of evaluation is, for each x $\in$ Rec that agrees with an implicit condition IC, to evaluate certainty and coverage of decision rules $IC \Rightarrow Q_w$ used for abduction at **Step5**. This is based on the following two observations:

The case that $\mathrm{Cer}(Q_w | IC)$ **is near to 1**: Almost all products that agree with IC also agree with the (weakened) query Q_w. Thus, we can regard that the implicit condition IC almost correctly represents some aspects of user's implicit image about products that the user searches, and therefore products $x \in \mathrm{Rec}$ that agree with IC are rare exceptions that do not agree with user's explicit image about such products.

The case that $\mathrm{Cov}(Q_w | IC)$ **is near to 1**: Almost all objects that agree with Q_w also agree with IC. Thus, we can regard that IC covers almost all images about products.

Thus, if the sum of certainty and coverage of the decision rule $IC \Rightarrow Q_w$ is higher, we treat recommendations of products $x \in \mathrm{Rec}$ that agree with IC as good recommendations. Note that, if $\mathrm{Cov}(Q_w | IC) = 1$, we have $\|IC\| \subseteq Q_w$ and there is no "rare exception" that we regard as recommended products that agree with IC. On the other hand, if both certainty and coverage of the rule $IC \Rightarrow Q_w$ is near to 0, we regard that there is little connection between IC and Q_w, and we think that recommendations of products $x \in \mathrm{Rec}$ that agree with IC are not good recommendations.

We also need to consider the difference between the (weakened) query Q_w and the derived characteristic Q_r. Thus, we introduce a criterion $\mathrm{DAC}(Q_w, Q_r)$ to evaluate this difference as follows:

$$\mathrm{DAC}\ (Q_w, Q_r) = \frac{\mathrm{Card}\ (C_w \cap C_r)}{\mathrm{Card}\ (C_w)}, \tag{6}$$

where $C_w = \{[\gamma_1 = z_1], \cdots, [\gamma_k = z_k]\}$ is the set of conditions that appear in the (weakened) query Q_w, and $C_r = \{[\gamma_1 = v_1], \cdots, [\gamma_k = v_k]\}$ is the set of conditions that appear in the derived characteristics Q_r. We call the criterion $\mathrm{DAC}(Q_w, Q_r)$ the *degree of agreement of characteristics* (for short, DAC).

Combining certainty, coverage and the degree of agreement of characteristics, we introduce a criterion to evaluate the *degree of recommendation* (for short, DoR) of each product $x \in \mathrm{Rec}$ as follows:

$$\mathrm{DoR}(x) = \max_{IC \in \mathbf{IC}(x)} \left\{ \begin{array}{l} \mathrm{Cer}\ (Q_w | IC) + \mathrm{Cov}(Q_w | IC) \\ + \mathrm{DAC}\ (Q_w, Q_r) \end{array} \right\}, \tag{7}$$

where $\mathbf{IC}(x)$ is the set of implicit conditions IC that agree with x such that the decision rule $IC \Rightarrow Q_w$ is identical to $\mathrm{DR}(u, Q_w)$ for some $u \in \|Q_w\|$. For any products $x, y \in \mathrm{Rec}$, if $\mathrm{DoR}(y) < \mathrm{DoR}(x)$, we regard x as better recommendation than y.

Example 3. This example is continuation of Example 2. We evaluate the degree of recommendation of c2 and c4. Both c2 and c4 agree with the implicit condition IC in Example 2, which is derived from $\mathrm{DR}(c8, Q_w)$ by abduction. Moreover, c2 and c4 have the characteristics Q_{r1} and Q_{r2}, respectively. Thus, we calculate DoR(c2) and DoR(c4) as follows:

$$\mathrm{DoR(c2)} = \mathrm{Cer}(Q_w | IC) + \mathrm{Cov}(Q_w | IC) + \mathrm{DAC}(Q_w, Q_{r1}) = \frac{1}{3} + 1 + \frac{1}{2} = \frac{11}{6}.$$

$$\mathrm{DoR(c4)} = \mathrm{Cer}(Q_w | IC) + \mathrm{Cov}(Q_w | IC) + \mathrm{DAC}(Q_w, Q_{r2}) = \frac{1}{3} + 1 + 0 = \frac{4}{3}.$$

We have DoR(c4) < DoR(c2), and therefore we regard c2 as better recommendation than c4.

4. EXPERIMENT

We verify efficiency of our recommendation method and evaluation method by evaluation experiment of product retrieval system based on our method.

4.1 Methods

12 examinees used the product retrieval system. Product data used in this experiment is car data in a magazine about used cars [9]. The number of data is 450, and each datum has 11 attributes. Table 2 illustrates the attributes, the number of values of each attribute, and values of each attribute of the car data.

In this experiment, each examinee entered one query represented by a conjunction of conditions $[\alpha_1 = x_1]$, and evaluated how each recommended car is suitable for the examinee by five grades (1: bad, 2: poor, 3: fair, 4: good, 5: excellent).

4.2 Experimental results

Table 3 illustrates experimental results by 12 examinees. In Table 3, No. is the id number of examinees. Cond and Ans are the number of conditions that the examinee entered to product retrieval system, and the number of products that agree with the (weakened) query. Rule is the number of decision rules used for calculating recommendations, respectively. Total, better and weak in Recommendations are the numbers of all recommended products to the examinee, better recommended products in all recommended products, and weakly recommended products in all recommended products, respectively, where $x \in \mathrm{Rec}$ is called a better recommended products if $\mathrm{DoR}(x)$ is higher than Avg. of DoR. Otherwise we call x a weakly recommended products. Similarly, total, better and weak in Avg. of grades are the averages of all recommended products, better recommended products and weakly recommended products, respectively. Note that there is just one recommendation to the examinee No. 10,

Table 2 : Attributes and values of car data

Attribute	Num	Values
Maker	6	Toyota, Nissan, Honda, Mazda, Mitsubishi, Subaru
Body Type	5	Sedan, Coupe, SUV, Mini-Van, Compact-Car
Price	7	Less 0.5, 0.5~1.0, 1.0~1.5, 1.5~2.0, 2.0~3.0, 3.0~4.0, over 4.0 (Unit: ¥1000000)
Transmission	3	AT, Manual, Both
Color	5	Black, White, Silver, Dark Blue, Others
Engine Displacement	6	1.0~1.5, 1.5~2.0, 2.0~2.5, 2.5~3.0, 3.0~4.0, over 4.0 (Unit: 1000cc)
Wheel Drive	2	2WD, 4WD
Car TV or Navigation	4	Both, Car TV, Navigation, Nothing
Sound Equipment	4	MD/CD, MD, CD, Cassette Player
ABS	2	Yes, No
Air Bags	2	Yes, No

and therefore we do not consider better (weak) recommendations to No.10 in Table 3.

As entire tendency, evaluations for all recommended products by examinees are between poor and fair. Correlation coefficient between Avg. of DoR and total Avg. of grades is 0.22, therefore correlation between these degrees is weak. On the other hand, in Avg. of grades, better is higher than weak on 8 examinees. These suggest tendency that better recommended products are more suitable for examinees than weak ones.

In this experiment, we observed two typical patterns of calculation of recommendations:

1. Many conditions and many recommendations (the case of No. 3, 4, 6): There is just one product that agrees with the (weakened) query, and *IC* agrees with many other recommended products. In this case, DoR depends on DAC completely.
2. A few conditions and a few recommendations (No. 8, 10): Many decision rules concern to the query, however, certainty of almost rules are 1.

From the case1, case2 and the weak tendency in Avg. of grades, we consider characteristics of DoR and issues for improvement our recommendation method as follows:

- We can regard DoR as one index that weakly reflects some aspects of relative suitability about products in each examinee.
- However, because DAC tends to become higher when we have many conditions in the query, comparing DoR among examinees is inappropriate.
- We need to consider how to weaken the original query and implicit conditions to avoid the case1 and 2. This is because rejection of informative and important attributes for the user may cause the case1, and strict treatment of implicit conditions causes a few recommendations in the case2.

5. SUMMARY

We have proposed a simple recommendation system based on rough set theory. First, we have introduced a recommendation algorithm that estimates user's implicit conditions about products using decision rules. Next, we have proposed a criterion to evaluate recommended products based on certainty and coverage. Moreover, to verify efficiency of our methods, we have discussed issues for improvement of our methods based on the experiment. More refinements of recommendation and evaluation methods are future works.

REFERENCES

1. Krose, T., Kajikawa, Y., Nomura. Y.; Music Recommendation System Using *KANSEI* Information, Proc. of DEWS2003, IEICE (2003). (in Japanese)
2. Kudo, Y., Amano, S., Murai, T.; A Note on Evaluation

Table 3 : Experimental results

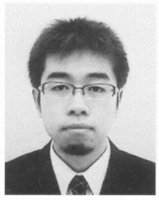
No.	Cond	Ans	Rule	Recommendations			Avg. of Cer	Avg. of Cov	Avg. of DoR	Avg. of grades		
				total	better	weak				total	better	weak
1	5	1	1	2	1	1	0.33	1.0	1.83	4.0	5.0	3.0
2	6	23	12	17	12	5	0.44	0.05	1.01	2.88	3.08	2.4
3	8	1	1	32	19	13	0.03	1.0	1.52	1.50	1.42	1.62
4	8	1	1	24	16	8	0.04	1.0	1.51	2.67	2.63	2.75
5	5	7	7	12	5	7	0.33	0.14	0.97	2.16	2.8	1.71
6	8	1	1	35	22	13	0.03	1.0	1.40	3.11	3.32	2.77
7	6	2	2	43	23	20	0.05	0.5	0.98	3.14	3.35	2.9
8	3	21	5	6	3	3	0.47	0.05	1.10	3.16	3.33	3.0
9	5	30	8	11	8	3	0.46	0.05	1.10	2.72	3.0	2.0
10	2	23	1	1	–	–	0.5	0.04	1.04	3.0	–	–
11	6	2	2	23	10	13	0.10	0.5	0.94	2.48	2.7	2.31
12	5	2	2	9	6	3	0.31	0.5	1.17	3.22	3.17	3.33

of Recommendation Systems Using Decision Rules, Proc. of the 16th Workshop of Aimai and *Kansei*, 5-6, JSKE (2006). (in Japanese)

3. Kudo, Y., Seino, T., Murai, T.; A Recommendation System Using Decision Rules and Upper Approximations, Proc. of the 4th Workshop on Rough Sets and *Kansei* Engineering, 60-61, JSKE and SOFT (2005). (in Japanese)
4. Mori, N., Tanaka, H., Inoue, K. (eds.); Rough Sets and *Kansei*: Knowledge Acquisition and Reasoning from *Kansei* Data. Kaibundo (2004). (in Japanese)
5. Murakami, T. et al.; Product Recommendation System Based on Personal Preference Model Using CAM, Transactions of the Japanese Society for Artificial Intelligence, Vol. 20, No. 5, 346-355, JSAI (2005). (in Japanese)
6. Pawlak. Z.; Rough sets. International Journal of Computer and Information Science, Vol. 11, 341-356 (1982).
7. Pawlak. Z.; Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer (1991).
8. Skowron, A., Rauser, C. M.; The Dicernibility Matrix and Functions in Information Systems, Intelligent Decision Support: Handbook of Application and Advances of the Rough Set Theory, Slowinski, R. (ed), 331-362, Kluwer (1992).
9. Used car information magazine Goo (Hokkaido version), Vol. 175, 2006/03/05 (2006). (in Japanese)
10. Yano, E. et al.; Development of Recommendation System with Anonymous *Kansei* Model, IPSJ Transactions on Databases, Vol. 44, No. SIG8 (TOD18), 46-54, IPSJ (2003). (in Japanese)

Yasuo KUDO

Received the M.E. and D.E. degrees in Systems and Information Engineering from Hokkaido University, in 1997 and 2000, respectively. From 2000 to 2003, he served as a Post Doctoral Fellow at Satellite Venture Business Laboratory, Muroran Institute of Technology, Muroran, Japan. Currently, he is a Research Associate with Department of Computer Science and Systems Engineering, Muroran Institute of Technology, Muroran, Japan. His research interests are rough set, belief change, non-monotonic reasoning, and database. He is a member of Japan Society of *Kansei* Engineering, Japan Society for Fuzzy Theory and Intelligent Informatics, and Japanese Society for Artificial Intelligence.

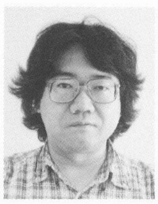


Shoei AMANO

Received the B.E. degree in Computer Science and Systems Engineering from Muroran Institute of Technology, Muroran, Japan, in 2006. Currently, he is a student of Master's Course in Graduate School of Computer Science and Systems Engineering, Muroran Institute of Technology. His research interests are rough set and recommendation system.

Takahiro SEINO

Received the B.E. degree in Computer Science and Systems Engineering from Muroran Institute of Technology, Muroran, Japan, in 2005. Currently, he is with COMTECH2000 Corporation, Sapporo, Japan. His research interests are rough set and recommendation system.



Tetsuya MURAI

Received the M.E. degree in Information Engineering from Hokkaido University in 1985 and the D.E. degree in Information Engineering from Hokkaido University in 1994. Currently, he is an Associate Professor in Hokkaido University. His research interests include a relationship of *Kansei* engineering with rough sets. He is a member of IEEE, ACM, IRSS, JSKE, SOFT, JSAI and so on.