A Granular Computing Approach to Inducing Rules in Incomplete Information Systems

Haiyan Yu [1,2] **Daoping Wang** [1]

[1]School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, P. R. China

[2]Computer center, Hebei University of Economics and Business, Shijiazhuang 050061, P. R. China,

Abstract

Generally, two ways are used to handle null value in incomplete information systems. One is transforming an incomplete system to a complete system. The other is process the incomplete information table based on toleration relation. In this paper we propose a method to process the incomplete information table based on granular computing directly. The incomplete information table is divided into granules only use the known value of instances. The rules are generated based on the granules.

Keywords: Granular computing, Rough sets, Incomplete information systems

1. Introduction

The conventional rough set theory is under the assumption that information systems are complete. However, missing data in information systems is common in many real applications. The conventional rough set theory under the indiscernibility relation is limited for analyzing the incomplete information system (IIS). An early extension of rough sets that can directly deal with incomplete data is under a tolerance relation[1].

Now two ways were used to handle null values in incomplete information systems. One is transforming an incomplete system to a complete system, e.g. each object with incomplete descriptor from the source system is represented by a set of quasi-objects in the target system or removing objects with unknown values from the original system. The other is process the incomplete information table based on toleration relation. Because the original data is unknown, no matter which ways are used to process the uncertain value, the results are uncertain. In this paper we propose an algorithm which intends to process the incomplete information table directly based on granular computing only using the certain information of each objects.

Granular Computing (GrC), as defined in the outline of the IEEE-GrC'2006 conference information, is a general computation theory for effectively using granules such as classes, clusters, subsets, groups and intervals to build an efficient computational model for complex applications with huge amounts of data, information and knowledge. It enables us to perceive the real world under various grain sizes, obtain only those useful or interesting things at different granularities, and switch among different granularities to get various levels of knowledge[2]. Today the granular computing has been appeared in many areas of information processing such as machine learning of artificial intelligence, query processes of data mining, processing indistinguishable information of fuzzy and rough set theories, and others [3].

In this paper we divide an incomplete information table into granules at different hierarchies, and rules are generated from these granules. When an incomplete decision table is decomposed to granules, each object is divided to different granules according to the attribute value which is not null.

2. Basic concept

Definition 1[4] An information system is the following tuple: $S = (U, A, V, f)$

Where U is a finite nonempty set of objects, $A = C \cup D$ is a finite nonempty set of attributes, where C is condition attributes and D is decision attributes, $V = \bigcup_{a \in A} V_a$, V_a is a nonempty set of values for $a \in A$, $f : U \times A \to V$ is an information function. If there exists a in C and x in U that satisfy the value $f(x,a)$ is unknown, denoted as *,. S is called incomplete information system.

Definition 2 Given a decision table $S =< U, A, V, f >$, $A = C \cup D$, each subset of attributes $P \subseteq A$ determines a binary indiscernibility relation $IND(P)$:

$$IND(P) = \{(x, y) \in U \times U \mid \forall a \in P, f(x,a) = f(y,a)\}$$

The relation $IND(P)$ is an equivalence relation and constitutes a partition of U . $U \mid IND(C)$ and $U \mid IND(D)$ are partitions of U , divided by C and D respectively.

An information table contains all available information and knowledge. In the decision logic language L [5], an atomic formula is represented by $a = v$, where $a \in A$ and $v \in V_a$. Formulae can be formed by logical negation, conjunction and disjunction. If a formula κ is satisfied by an object x, $x \models_s \kappa$ or in short $x \models \kappa$ is given. If κ is a formula, the set $m(\kappa)$ defined by $m(\kappa) = \{x \in U \mid x \models \kappa\}$ is called the meaning of κ in S. The meaning of a formula κ is the set of all objects having the property expressed by the formula κ [4].

Definition 3 [6] Given a decision table $S = < U, A, V, f >$, $A = C \cup D$, with formula $\kappa = (c_1, v_{c_1}) \wedge ... \wedge (c_i, v_{c_i})$, where $i = 1,...,|C|$, $c_1,...,c_i \in C$, $v_{c_1},...,v_{c_i} \in V$. The meaning of κ, $m(\kappa) = \{x \in U \mid x \models \kappa\}$, is called the granule presented by κ in S, or a granule in short. i is called the length of κ.

According to Definition 3, the granules presented by formulae with the same length have the same granularity. We call these granules are at the same hierarchy. Due to $i = 1,...,|C|$, S is divided into $|C|$ hierarchies at most and the decomposition of S at any hierarchy is a cover of the universe.

Definition 4 [5] Given two formulae κ and φ. A symbol $\Rightarrow$ means that formula κ infers to formula φ in terms of $\kappa \Rightarrow \varphi$. The confidence support of φ provided by κ is defined as follows:

$$AS(\kappa \Rightarrow \varphi) = | m(\kappa) \cap m(\varphi) | / | m(\kappa) |$$

Definition 5 Given a decision table $S = < U, A, V, f >, A = C \cup D$, X_i is defined as $X_i \in U \mid IND(C)$, where, $i = 1,...,|U \mid IND(C)|$, X_j is defined as $X_j \in U \mid IND(D)$, where, $j = 1,...,|U \mid IND(D)|$. X_i and X_j are equivalence classes divided by C and D respectively. They are called granules and represented by $m(\kappa)$ and $m(\varphi)$. Where, κ and φ are formulae.

3. Algorithm

Generally, decision rules are induced with respect to larger granularity. So, at first, the algorithm generates rules from the granules at the 1st hierarchy, an information table is divided into some basic granules with respect to atomic formulae of the decision logic language in term of definition 3, i.e., divided by single condition attributes and decision attribute. The condition attributes $C = \{c_1, c_2, ..., c_m\}$ give atomic formulae like $(c_1, v_{11}), (c_1, v_{12})..., (c_m, v_{mp})$, where $m = |C|$, $p = |U \mid IND(C)|$. The decision attribute $D = \{d\}$ gives atomic formula like $(d, v_1),...,(d, v_n)$, where $n = |U \mid IND(D)|$.

Due to the attribute value "*" in incomplete information system is unknown or uncertain, the rule generated by these value must be uncertain. According to definition 2, when the incomplete decision table is divided into granules, the attributes value $f(x, a)$ and $f(y, a)$ can't be "*" in a binary indiscernibility relation $IND(P)$, where $x, y \in U, a \in P$. If only one of attribute value of object is "*" in $IND(P)$, the object will not be divided into any granules. So the decomposition of S at this hierarchy may be not a cover of the universe. The objects which are not divided into any granules will be divided in next hierarchy according to above method. In another words, we consider these objects are not satisfy the condition of generating rules.

Using the granule whose confidence support is equal to 1 to induce the decision rule. When a rule is generated from a granule, all the objects in this granule will be considered as "covered" by this rule. If not all objects of an information table are contained, the information table should be further decomposed. The rule will generate from the granules continue at the 2nd hierarchy, the 3rd hierarchy, and so on.

Algorithm 1:

Input: An incomplete information table $S = (U, A = (C \cup D))$

Output: Rule set (RS)

Step 1: Set $RS = \{\emptyset\}$, $CS = \{\emptyset\}$, $DS = \{\emptyset\}$, m=1, where, CS is the set of objects already been covered by rules in RS, DS is the set of atomic formulae given by decision attribute $D = \{d\}$, m is the length of formulae κ used to decompose the decision table and also the number of condition attributes used to compute $|U \mid IND(C)|$.

Step 2: Calculate the atomic formulae φ_j given by decision attribute $D = \{d\}$, let $DS = DS \cup \varphi_j$, where. $j = 1,...,|U \mid IND(D)|$.

Step 3: Calculate the decomposition of S at the m th hierarchy.

Divide the information table into granules with reference to m condition attributes. If the one of the m condition attribute values of the object is "*", it will not be divided into any granules. $GS = \{\{m(\kappa_1)\},...,\{m(\kappa_i)\},...,\{m(\kappa_n)\}\}$, where κ_i is a formula, $i = 1,...,n, n$ is the number of granules in GS.

Step 4: For each $\varphi_j \in DS$:

Calculate the certainty support $AS(\kappa_i \Rightarrow \varphi_j)$ in terms of Definition 4.

If $AS(\kappa_i \Rightarrow \varphi_j)$ =1 then

{ $RS = RS \cup \{\kappa_i \Rightarrow \varphi_j\}$

$CS = CS \cup \{m(\kappa_i \Rightarrow \varphi_j)\}$ }

Step 5: If $U - CS = \emptyset$ or there is no object with complete information in the information table, then the algorithm stops, otherwise m=m+1, go to step 3.

4. An example

The following example illustrated the execution process of Algorithm 1.

An incomplete information table is shown in Table 1. It contains 3 condition attributes, 1 decision attribute, and 8 instances. a ,b and c are condition attributes. d is decision attribute.

U	a	b	c	d
x_1	1	2	3	2
x_2	2	3	2	1
x_3	2	*	*	2
x_4	1	*	3	2
x_5	3	2	2	3
x_6	1	3	2	1
x_7	*	1	3	3
x_8	3	1	*	3

Table 1:An Incomplete Information Decision Table

The information in Table 1 is first decomposed with reference to single condition attribute and shown in Table 2.

κ	$m(\phi)$	φ	$m(\varphi)$
(a,1)	$\{x_1, x_4, x_6\}$	(d,1)	$\{x_2, x_6\}$
(a,2)	$\{x_2, x_3\}$	(d,2)	$\{x_1, x_3, x_4\}$
(a,3)	$\{x_5, x_8\}$	(d,3)	$\{x_5, x_7, x_8\}$
(b,1)	$\{x_7, x_8\}$		
(b,2)	$\{x_1, x_5\}$		
(b,3)	$\{x_2, x_6\}$		
(c,2)	$\{x_2, x_5, x_6\}$		
(c,3)	$\{x_1, x_4, x_7\}$		

Table 2:The First Decomposition of Table 1

We give an example to illustrate how to process the null value.

For example, $f(x_7, a) = "*"$, it doesn't satisfy $\kappa = (a,1)$, $\kappa = (a,2)$, $\kappa = (a,3)$, so it isn't divided into $m(a,1)$, $m(a,2)$, $m(a,3)$.

According to Table 2, Calculate the certainty support according to

$$AS(\kappa \Rightarrow \varphi) = | m(\kappa) \cap m(\varphi) | / | m(\kappa) |$$

The result is shown in Table 3.

From Table 3, we can find there are three granules whose certainty support is equal to 1. So the following rules are generated:

$(a,3) \Rightarrow (d,3)$
$(b,1) \Rightarrow (d,3)$
$(b,3) \Rightarrow (d,1)$

$AS(\kappa \Rightarrow \varphi)$			
κ	φ =**(d,1)**	φ =**(d,2)**	φ =**(d,3)**
(a,1)	1/3	2/3	0
(a,2)	1/2	1/2	0
(a,3)	0	0	1
(b,1)	0	0	1
(b,2)	0	1/2	1/2
(b,3)	1	0	0
(c,2)	2/3	0	1/3
(c,3)	0	2/3	1/3

Table 3:The First Certainty Support Table

Since $m(a,3) = \{x_5, x_8\}$, $m(a,3) = \{x_7, x_8\}$, $m(a,3) = \{x_2, x_6\}$,five objects are covered by these rules.

$CS = \{x_2, x_5, x_6, x_7, x_8\}$
$U - CS = \{x_1, x_3, x_4\}$

Because rule granules don't contain all objects of the information table, the information table will be second divided. The objects in $U - CS$ is divided with reference to two condition attributes. The result of the second decomposition is shown in Table 4 The second certainty support is shown in Table 5.

κ	$m(\kappa)$	φ	$m(\varphi)$
(a,1)$\wedge$(b,2)	$\{x_1\}$	(d,1)	$\{x_2, x_6\}$
(a,1)$\wedge$(c,3)	$\{x_1, x_4\}$	(d,2)	$\{x_1, x_3, x_4\}$
(b,2)$\wedge$(c,3)	$\{x_1\}$	(d,3)	$\{x_5, x_7, x_8\}$

Table 4:The Second Decomposition of Table 1

$AS(\kappa \Rightarrow \varphi)$			
κ	φ =**(d,1)**	φ =**(d,2)**	φ =**(d,3)**
(a,1)$\wedge$(b,2)	0	1	0
(a,1)$\wedge$(c,3)	0	1	0
(b,2)$\wedge$(c,3)	0	1	0

Table 5:The Second Certainty Support Table

The following rules are generated:

$(a,1) \wedge (b,2) \Rightarrow (d,2)$
$(a,1) \wedge (c,3) \Rightarrow (d,2)$
$(b,2) \wedge (c,3) \Rightarrow (d,2)$

Two objects are covered by these rules: x_1, x_4

$CS = \{x_1, x_2, x_4, x_5, x_6, x_7, x_8\}$
$U - CS = \{x_3\}$

Next, the information table should be decomposed thirdly using 3 condition attributes. But the condition attributes value of object x_3 is {2,*,*}. According to

Step5, there is no object with complete information in the information table, the algorithm stops.

From the incomplete information table I six rules are generated:

$$(a,3) \Rightarrow (d,3)$$
$$(b,1) \Rightarrow (d,3)$$
$$(b,3) \Rightarrow (d,1)$$
$$(a,1) \wedge (b,2) \Rightarrow (d,2)$$
$$(a,1) \wedge (c,3) \Rightarrow (d,2)$$
$$(b,2) \wedge (c,3) \Rightarrow (d,2)$$

5. Conclusions

Granular computing is a way of thinking and learning, which has been already explored in many fields. It enables us to perceive the world and solve the problem at different granularities. In this paper, a method to decompose incomplete information decision tables is proposed. This method tends to search for rules in larger granules. Based on these granules, rules are generated.

References

[1] T.R. Li, D. Ruan, Wets Geert, J. Song, Yng Xu, A rough set based characteristic relation approach for dynamic attribute generalization in data mining, *Knowledge-Based Systems*, pp. 485-494, 2007.

[2] J. J. An, G. Y. Wang, Y. Wu, Q. Gan, A Rule Generation Algorithm based on Granular Computing, *in Proc. IEEE Int. Conf. on Granular Computing, Beijing, P. R. China,* pp. 102-108, 2005.

[3] Zadeh, L. A., Fuzzy sets and information granularity, *In M. M. Gupta, P. K. Ragade, R. R.Yager, eds, Advances in Fuzzy Set Theory and Applications, North Holland, Amsterdam,* pp. 3-18, 1979.

[4] Y.Y. Yao, On Modeling Data Mining with Granular Computing, *Proceeding of COMPSAC2001*, pp. 638-643, 2001.

[5] Y.Y. Yao, "A Generalized Decision Logic Language for Granular Computing Fuzzy Systems," *Proceedings of the 2002 IEEE International Conference*, 1:773-778, 2002.

[6] Q Gan, G. Y. Wang, J. Hu, A Self-Learning Model based on Granular Computing, *2006 IEEE International Conference on Granular Computing*, pp. 530-533, 2006.

[7] M. Kryszkiewicz, Rough set approach to incomplete information system, *Information Sciences*, 112:39-49, 1998.

[8] Y. Y. Yao, J. T. YAO, Granular Computing as a Basis for Consistent Classification Problems, http://www2.cs.uregina.ca/~yyao/grc_paper/.

[9] Y. Y. Yao, N. Zhong, Potential Applications of Granular Computing in Knowledge Discovery and Data Mining, *in Proceedings of World Multiconference on Systems, Cybernetics and Informatics, Computer Science and Engineering, Orlando,* pp. 573-580, 1999.

[10] T. Y. Lin, Granular Computing (Structures, Representations, and Applications), *in: G. Y. Wang, Q. Liu, Y. Y. Yao, A. Skowron (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Springer-Verlag, Berlin,* pp. 16-24, 2003.

[11] B. Zhang, L. Zhang, *Theory and Applications of Problem Solving,* North-Holland, Amsterdam, 1992.

[12] Kryszkiewicz, M., Rules in incomplete information systems, *Information Sciences,* 113:271-292, 1999.

[13] Kryszkiewicz, M., Rough set approach to incomplete information systems, *Information Sciences ,* 112:39-49, 1998.

[14] B. Zhang, and L. Zhang, The Quotient Space Theory of Problem Solving, *Proceedings of International Conference on Rough Sets, Fuzzy Set, Data Mining and Granular Computing, Lecture Notes in Artificial Intelligence,* pp. 11-15, 2003.