A New Extracting Rule Algorithm from Incomplete Information System by Covering Rough Sets

Dai Dai[1] **Jianpeng Wang**[2]

[1]Dept. of Economic Math., South Western Univ. of Finance and Economics, Chengdu 610074, P.R.China
[2]Dept. of Math., Southwest Jiaotong University, Chengdu 610031, P. R. China

Abstract

In this paper, a new extracting rule algorithm from incomplete information system is proposed. First, we produce a covering on a domain according to attribute value of the objects, and then reducts are made on this covering. Second, we utilize rough sets model based on covering to estimate unknown value, so that an incomplete information system is transformed to a complete information system, and thus rules can be extracted.

Keywords: Covering, Rough sets, Incomplete information system, Rules

1. Introduction

Rough sets, presented firstly by Pawlak in 1982, is a new mathematic tool to deal with uncertain knowledge. Now, it has been developed into the important research tendency of artificial intelligence. It has a potential applied prospect in Data Mining and KDD, and has achieved successful application to various fields, such as machine study, decision analysis, process control, model identification and data mining.

Rough sets have significant application in extracting rules from information system. There have been a lot of methods to extract rules from complete information system in the literature, while it is much more difficult to extract rules from incomplete information system. Chmielewski [1] proposed that incomplete data sets may be transformed into complete data sets before learning programs begin by removing objects with unknown values from data sets. Kryszkiewicz [2] used the indiscernibility relations to characterize incomplete data. In the sequel, some researchers made some modifications on indiscernibility relations, for example, Stefanoski [9] proposed the similar relations, Wang [3] proposed constrained indiscernibility relations and Tzung [4] proposed a algorithm which can simultaneously derive rules from incomplete data sets and estimate the missing values.

The main purpose of this paper is to present a new algorithm to extract rules from incomplete information system. The key point of the new algorithm is to form a covering on a domain according to the attribute value of an arbitrary object. We can estimate the unknown attribute value through the minimal description of objects after reducing the covering. Then rules are extracted by transforming incomplete information system to complete information system. Compared with the algorithm of Hong et al., the new algorithm can be more easily generated, while maintaining the same result.

The rest of this paper is organized as follows. In Section 2 we introduce some basic concepts and notations which will be used throughout this paper. The new algorithm of extracting rules from incomplete information system and a comparison with the known method are presented in Section 3. Finally, Section 4 concludes this paper.

2. Basic conception

2.1. Covering rough sets model

Definition 1 *Let U be a domain, C a family of subsets of U. If none subsets in C is empty, and $\bigcup C = U$, then C is called a covering of U, and the ordered pair $< U, C >$ is called a covering approximation space.*

Definition 2 *Let $< U, C >$ be a covering approximation space, $x \in U$, then the set family*

$$\{K \in C | x \in K \wedge (\forall S \in C \wedge x \in S \wedge S \subseteq K \Rightarrow K = S)\}$$

is called the minimal description of x and denoted by $md(x)$.

Definition 3 *Let U be a finite nonempty domain, C is a covering of U, for any $X \subseteq U$,the covering upper and lower approximation set families of X are respectively defined as*

$$\overline{C}(X) = \{x \in U | \bigcap md(x) \bigcap X \neq \phi\},$$

$$\underline{C}(X) = \{x \in U | \bigcap md(x) \subseteq X\}.$$

2.2. Covering reducts

Definition 4 *Let U be a finite nonempty domain, C is a covering of U, $K \in C$, if K is the union of sets in $C - \{K\}$, then K is called a reducible element of C, otherwise K is called an irreducible element of C.*

Definition 5 *Let U be a finite nonempty domain, C is a covering of U, the covering after removing all the reducible element of C is called the reduct of C, and denoted by $reduct(C)$.*

2.3. Incomplete equivalence classes

For any attribute $a \in AT$, each object is represented as a tuple $(obj, symbol)$. If $f(obj, a) = *$, then symbol is denoted by u (uncertain) or c (certain). If an object obj^i has a certain value V_a^i for $a \in AT$, then (obj^i, c) is put in the equivalence class of attribute V_a^i; otherwise, (obj^i, u) is put in each equivalence class. The object sets formed in this way are called incomplete equivalence classes.

3. New extracting rule algorithm from incomplete information system

Collect all elements in the incomplete equivalence classes generated by all the condition attributes together, then the set formed by above elements is a covering of domain U. For any object x in U, $\bigcap md(x)$ can be obtained from the covering rough sets model. It follows from the condition attribute value that $\bigcap md(x)$ is an undistinguished minimal object set, that is, if put any object not in $\bigcap md(x)$ into $\bigcap md(x)$, then there exists a object in $\bigcap md(x)$, which can be distinguished with the added object. If no more conditions other than the given condition attribute value exist, then the condition attribute value of all the objects in $\bigcap md(x)$ are identical.

In the incomplete information system, there is another important condition should be noted, i.e., the decision attribute value. If the decision attribute value can be determined, the object's unknown value then can be estimated in the following way: If $x, y \in \bigcap md(z)$, and, $f(x, d) = f(y, d)$ for any $d \in D$, then all the condition attribute values of x and y can be transformed to the known value. If $x, y \in \bigcap md(z)$, and there exist a d in D such that $f(x, d) \neq f(y, d)$, then we can deal with the problem in the following two ways. One is that we still believe that x and y have the same condition attribute values, then the decision attribute values of x and y can be obtained by the object which has the same condition attribute values as x and y. Another way is that we think x and y can be distinguished if there exists an estimate of the unknown attribute value, so that the condition attribute value is not identical. Notice that the second method can make the information system more harmony, we always use that method in the sequel. When an unknown attribute value of some object is estimated, the corresponding $*$ is displaced by it. Then the object with the new attribute value is compared with other objects in the common $\bigcap md(x)$. If the object can be distinguished with other objects, then it should be removed from $\bigcap md(x)$. Using the same steps to estimate unknown attribute values and remove objects from $\bigcap md(x)$, until the estimation can not be continued. Therefore we can obtain the decision rules according to the objects and their decision attribute value, and simplify the rules.

In the rest of the section, we will illustrate the new algorithm with a specific example.

Example 1 *the following table is an incomplete information system (SP, DP: condition attribute; BP: decision attribute)*

objects	SP	DP	BP
1	L	N	N
2	H	L	H
3	N	H	N
4	L	L	L
5	$*$	H	H
6	N	H	H
7	L	$*$	L
8	L	H	N
9	$*$	N	H

Table 1: An incomplete information system

Step 1: Simplify objects in the incomplete information system. If there are two objects, obj^1 and obj^2 related to $\forall a \in AT$, $\forall d \in D$, satisfying $f(obj^1, a) = f(obj^2, a)$ and $f(obj^1, d) = f(obj^2, d)$, we can delete one of the two objects from the system to get rid of repeated information.

Step 2: Denote incomplete equivalence classes of all the condition attribute values as follows:
$U/\{SP\} = \{\{(3,c),(6,c),(5,u),(9,u)\},\{(2,c),(5,u),(9,u)\},\{(1,c),(4,c),(7,c),(8,c),(5,u),(9,u)\}\}$
$U/\{DP\} = \{\{(1,c),(9,c),(7,u)\},\{(3,c),(5,c),(6,c),(8,c),(7,u)\},\{(2,c),(4,c),(7,u)\}\}$

Step 3: Put all elements of each condition attribute of incomplete equivalence classes together. Obviously, the sets of these elements are a covering of the domain. We can reduce this covering and the corresponding result of the above example is:
$\{(3,c),(6,c),(5,u),(9,u)\},\{(2,c),(5,u),(9,u)\},$
$\{(1,c),(4,c),(7,c),(8,c),(5,u),(9,u)\}$
$\{(1,c),(9,c),(7,u)\},\{(3,c),(5,c),(6,c),(8,c),(7,u)\},$
$\{(2,c),(4,c),(7,u)\}$
then $\bigcap md(obj^i), i = 1, 2, \cdots, n$ are computed as
$\bigcap md(1) = \{1,(7,u),(9,u)\}$;
$\bigcap md(2) = \{2\}$;
$\bigcap md(3) = \{3,(5,u),6\}$;
$\bigcap md(4) = \{4,(7,u)\}$;
$\bigcap md(5) = \{5\}$;
$\bigcap md(6) = \{3,(5,u),6\}$;
$\bigcap md(7) = \{(7,u)\}$;
$\bigcap md(8) = \{(5,u),(7,u),8\}$;
$\bigcap md(9) = \{(9,u)\}$;

Step 4: Simplify sets $\bigcap md(obj^i)$. Note that $\bigcap md(x)$ is the minimal set of undistinguishable object according to the condition attribute value. In general, if $\bigcap md(j) \subseteq \bigcap md(i)$, then we delete $\bigcap md(j)$. It follows that, for the above example, the simplified $\bigcap md(obj^i)$ can be obtained as
$\bigcap md(1) = \{1,(7,u),(9,u)\}$;
$\bigcap md(2) = \{2\}$;
$\bigcap md(3) = \{3,(5,u),6\}$;
$\bigcap md(4) = \{4,(7,u)\}$;
$\bigcap md(8) = \{(5,u),(7,u),8\}$;

Step 5: Evaluate the value of the unknown attribute according to the corresponding evaluated rule. Since the decision attribute value of the 4^{th} object is consistent with that of the 7^{th} object in $\bigcap md(4) = \{4,(7,u)\}$, we then evaluate $f(7, DP) = L$. Note that the default attribute value of the 7^{th} object has been evaluated, and $\bigcap md(1) = \{1,(7,u),(9,u)\}$, it follows that $(7,u)$ in $\bigcap md(8) = \{(5,u),(7,u),8\}$ is distinct from the other objects in $\bigcap md(x)$, then $(7,u)$ is deleted. Similarly, in $\bigcap md(3) = \{3,(5,u),6\}$, the 5^{th} and 6^{th} objects have the same attribute values, thus $f(5, SP) = N$ is evaluated and then $(5,u)$ in $\bigcap md(8) = \{(5,u),(7,u),8\}$ is deleted. In $\bigcap md(1) = \{1,(9,u)\}$, according to the evaluated rule, we think the 1^{st} and 9^{th} objects have different condition attribute value due to the fact that the two objects have different attribute value, thus $f(9, SP)$ is evaluated to be N or H. Therefore, the unknown attribute values are all evaluated, the following table 2 is then obtained from table 1.

objects	SP	DP	BP
1	L	N	N
2	H	L	H
3	N	H	N
4	L	L	L
5	N	H	H
6	N	H	H
7	L	L	L
8	L	H	N
9	H or N	N	H

Table 2: The corresponding complete information system

Step 6: Extract and simplify the rules. In the case that some condition attribute value of a object maybe take two different values, we divide the object into two objects such that they take different attribute value for the condition attribute, while maintain the other attribute values. For the case of multiple values, we can deal similarly. Therefore, the incomplete information system is transformed into a complete information system. Then, the problem is converted into extracting and simplifying the rules in complete information system. Here, we refer to the method in [4]. For the above example, the rules can be extracted as follows
$(1)(SP, L) \wedge (DP, N) \rightarrow (BP, N)$;
$(2)(SP, H) \wedge (DP, L) \rightarrow (BP, H)$;
$(3)(SP, N) \wedge (DP, H) \rightarrow (BP, N)$;
$(4)(SP, L) \wedge (DP, L) \rightarrow (BP, L)$;
$(5)(SP, N) \wedge (DP, H) \rightarrow (BP, H)$;
$(6)(SP, L) \wedge (DP, H) \rightarrow (BP, N)$;
$(7)(SP, H) \wedge (DP, N) \rightarrow (BP, H)$;
$(8)(SP, N) \wedge (DP, N) \rightarrow (BP, H)$;
In terms of simplifying, we have
$R1 : (SP, L) \wedge (DP, N) \rightarrow (BP, N)$;
$R2 : (SP, H) \rightarrow (BP, H)$;
$R3 : (SP, N) \wedge (DP, H) \rightarrow (BP, N) \vee (BP, H)$;
$R4 : (SP, L) \wedge (DP, L) \rightarrow (BP, L)$;
$R5 : (SP, L) \wedge (DP, H) \rightarrow (BP, N)$;
$R6 : (SP, N) \wedge (DP, N) \rightarrow (BP, H)$;

4. conclusion

This paper put forward a new algorithm to extract rules from incomplete information system based on the covering of rough sets models.

The method here and the method raised by Fzung-Pei Hong need to estimate unknown condition attribute values and then extract rules from the evaluated information system. Both the two methods are same in extracting and simplifying rules from the information system which is estimated , therefore, their differences mainly focus on the way to estimate unknown attribute values. Compared with the algorithm of Hong et al., the new algorithm can be more easily generated, while maintaining the same result.

References

[1] M. R. Chmielewski, J. W. Grzymala-Busse, N. W. Peterson and S. Than, The rule induction system LERS-A version for personal computers. *Found. Comput. Decision Sci.*, 18:181–212, 1993.

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

[3] G. Wang, The extension of Rough Sets theory in incomplete information system. *Computer Research and Development*, 10:1238–1243, 2002.

[4] Tzung-Pei Hong, Li-huei Tseng and Shyue-Liang Wang, Learning rules from incomplete training examples by rough sets. *Expert Systems with Applications*. 22:285–293, 2002.

[5] Willian Zhu and Fei-Yue Wang, Reduction and axiomization of covering generalized rough sets. *Information Sciences*, 152:217–230, 2003.

[6] Tsumoto S., Automated extraction of medical expert system rules from clinical databases based on rough set theory. *Information Sciences*, 112:67–84, 1998.

[7] Pawlak Z., Rough sets. *International Journal of Computer and Information Sciences*, 11:341–356, 1982.

[8] Lin T Y, A rough logic formalism for fuzzy controllers: A hard and soft computing view. *International Journal of Approximate Reasoning*, 15:395-414, 1996.

[9] J. Stefanoski and A Tsoukias, On the extension of rough sets under incomplete information. *In N Zhong, A Skowrion, S Ohsuga editors, proceedings of the 7^{th} Int'l Workshop on New Directions in Rough Sets, Data mining, and Granular Soft computing*, pp. 73–81, Springer-Verlag, 1999.