第 30 卷 第 3 期　　自 动 化 学 报　　Vol. 30, No. 3
2004 年 5 月　　ACTA AUTOMATICA SINICA　　May, 2004

Rough Computational Methods Based on Tolerance Matrix

HUANG Bing[1]　HE Xin[1]　ZHOU Xian-Zhong[2]
[1](*Department of Automation, Nanjing University of Science and Technology, Nanjing* 210094)
[2](*School of Engineering Management, Nanjing University, Nanjing* 210093)
(E-mail: hbhuangbing@sohu.com; hexin_01@sina.com, zhouxizh@public1.ptt.js.cn)

Abstract Rough set theory is a new mathematical tool to deal with vagueness and uncertainty. The classical rough set theory based on equivalence relation has made a great progress, while the equivalence relation is too harsh to meet and is extended to tolerance relation in real world. It is important to investigate rough computational methods for rough set theory, which is one of the bottleneck problems in the development of rough set theory. Matrix computation based on tolerance relation for information systems is discussed, and a one-to-one relationship between tolerance relation and tolerance matrix is constructed in this paper. Two algorithms for attribute reduction in incomplete information system are presented with an example to illustrate their validity.

Key words Rough set, tolerance relation, tolerance matrix, incomplete information system, reduction

1 Introduction

Rough set theory is a mathematical tool to deal with vagueness and uncertainty of imprecise data. The theory introduced by Pawlak in 1982 has been developed and found applications in the field of decision analysis, data analysis, pattern recognition, machine learning, and knowledge discovery in databases.

It is very important but difficult to investigate rough computation in rough set theory, such as finding all reducts or minimal reducts, which is NP-hard. Therefore, many computational methods are proposed by using heuristic algorithm. For example, Guan *et al.*[1] presented a heuristic algorithm for attribute reduction using the importance of attributes. Theresa Beaubouef[2] and Duntsch *et al.*[3] introduced information entropy into rough set theory to depict knowledge roughness. Miao Duo-Qian *et al.*[4] used mutual information as heuristic knowledge and put forward a heuristic algorithm to find one minimal reduct of a decision table. Wang Guo-Yin *et al.*[5] presented a kind of attribute reduction algorithm based on conditional information entropy. Guan *et al.*[6] defined an equivalence matrix based on equivalence relation, thereby made equivalence relation correspond to equivalence matrix, and described rough computational methods with matrix computation. Though intuitively and validly, this method cannot be used in the decision table.

Computation methods mentioned above are based on the equivalent relation of the classical rough set, but equivalent relation is too rigid and even unnecessary in real world, especially in the processing of incomplete information system with null values, where equivalence relation cannot be satisfied. In order to handle incomplete information system, Kryszkiewicz M[7] extended equivalence relation into tolerance relation, which meets reflexive and symmetric laws. Tzung-Pei Hong *et al.*[8] presented an algorithm based on tolerance relation to stuff null value and extracted rules by using the upper and lower approximation. However, this algorithm is invalid for overmuch null value.

A one-to-one relationship between tolerance relation and tolerance matrix is construc-

Received June 11, 2003; in revised form September 8, 2003
收稿日期 2003-06-11; 收修改稿日期 2003-09-08

ted in Section 2. The tolerance matrix of incomplete information system and how it represents the importance of attributes are introduced in Section 3 and Section 4, respectively. On the basis of the above, two new algorithms for attribute reduction are proposed with their time complexity analyzed in Section 5. Section 6 is devoted to example analysis and Section 7 concludes this paper.

2 Tolerance relation and tolerance matrix

$S=(U,A)$ is called an information system, where U is a non-empty finite set called universe denoting the set of all objects, and A is a non-empty finite set denoting the set of all attributes. If the value of an object corresponding to an attribute is unknown, then the information system is called incomplete one.

The binary relation P on U is tolerant, if it is 1) reflexive: uPu for all $u\in U$, 2) symmetric: if uPv, then vPu for all $u,v\in U$.

Definition 1. If Ω is the set of all tolerance relations on $U,P,Q\in\Omega$, the intersection of tolerance relations is defined as follows: $u(P\cap Q)v\Leftrightarrow uPv, uQv$.

Now we introduce a partial ordering relation over Ω: inclusion.

Definition 2. For$P,Q\in\Omega$, if $P\cap Q=P$, namely $uPv\Rightarrow uQv$, we call that P is included in Q (that is to say Q includes P), denoted by $P\subseteq Q$. If $P\subseteq Q$ and $P\neq Q$, we say that P is strictly included in Q (that is to say Q strictly includes P), denoted by $P\subset Q$.

Then we introduce the definition of tolerance matrix as follows, which is corresponding to tolerance relation on the universe $U=\{u_1,u_2,\cdots,u_{|U|}\}$ (where $|U|$ denotes the cardinality of U).

Definition 3. Let $U=\{u_1,u_2,\cdots,u_{|U|}\}$ and P be a tolerance relation on U and M_P be its matrix. M_P is defined as follows: $M_P=(r_{ij})_{|U|\times|U|}$, where $r_{ij}=\begin{cases}1, & \text{if } u_iPu_j\\ 0, & \text{if not}\end{cases}$.

Property 1. 1) $r_{ii}=1 \quad 1\leqslant i\leqslant|U|$; 2) $r_{ij}=r_{ji} \quad 1\leqslant i,j\leqslant|U|$.

The matrix, which satisfies Property 1, is called a tolerance matrix. Obviously, one tolerance matrix can determine one tolerance relation on U. Therefore, the relation between the tolerance relation and tolerance matrix on the universe is one to one.

The intersection operation of tolerance matrix is defined as follows.

Definition 4. For $P,Q\in\Omega$, the corresponding intersection operation of tolerance matrices is defined:

$$M_P\cap M_Q=(r_{ij})_{|U|\times|U|}\cap(r'_{ij})_{|U|\times|U|}=(\min(r_{ij},r'_{ij}))_{|U|\times|U|}$$

Property 2. $M_{P\cap Q}=M_P\cap M_Q$.

Definition 5. The partial order relation "$\leqslant$" of tolerance matrices on the universe U is defined as follows. We say M_P is less than M_Q, if $M_P=(r_{ij})\leqslant M_Q=(r'_{ij})\Leftrightarrow r_{ij}\leqslant r'_{ij}, 1\leqslant i, j\leqslant|U|$. If $M_P\leqslant M_Q$ and $M_P\neq M_Q$, we call that M_P is strictly less than M_Q, denoted by $M_P<M_Q$.

Property 3. $P\subseteq Q\Leftrightarrow M_P\leqslant M_Q$; $P\subset Q\Leftrightarrow M_p<M_Q$.

Definition 6. Let (U,Ω) be a knowledge base. Then two knowledge bases $K_1=(U,\Omega_1)$ and $K_2=(U,\Omega_2)$ are equivalent, if $\bigcap_{P_1\in\Omega_1}P_1=\bigcap_{P_2\in\Omega_2}P_2$, denoted by $K_1\equiv K_2$.

Property 4. $K_1=(U,\Omega_1)\equiv K_2=(U,\Omega_2)\Leftrightarrow\bigcap_{P_1\in\Omega_1}M_{P_1}=\bigcap_{P_2\in\Omega_2}M_{P_2}$.

Definition 7. The knowledge base $K_1=(U,\Omega_1)$ is finer than $K_2=(U,\Omega_2)$, if $\bigcap_{P_1\in\Omega_1}P_1\subseteq\bigcap_{P_2\in\Omega_2}P_2$, denoted by $K_1\leqslant K_2$; $K_1=(U,\Omega_1)$ is strictly finer than $K_2=(U,\Omega_2)$, if $\bigcap_{P_1\in\Omega_1}P_1\subset\bigcap_{P_2\in\Omega_2}P_2$, denoted by$K_1<K_2$.

Property 5. $K_1 \leqslant K_2 \Leftrightarrow \bigcap_{P_1 \in \Omega_1} M_{P_1} \leqslant \bigcap_{P_2 \in \Omega_2} M_{P_2}$; $K_1 < K_2 \Leftrightarrow \bigcap_{P_1 \in \Omega_1} M_{P_1} < \bigcap_{P_2 \in \Omega_2} M_{P_2}$.

3 Information systems and tolerance matrices

3.1 Non-decision tables and tolerance matrices

An information system is denoted by $S=(U,A)$, where U is its universe, A is its attribute set and elements of A are called attributes. For each $a \in A$, we can define a tolerance relation P_a determined by a on U: $uP_av \Leftrightarrow a(u)=a(v) \vee a(u)=^* \vee a(v)=^*$, $u,v \in U$, where $*$ denotes null value. Namely, the object u and the object v are tolerant on the attribute a, if the value of u on the attribute a is equal to the value of v on attribute a, or the value of u on attribute a is unknown, or the value of v on attribute a is unknown. Therefore, one attribute can determine one tolerance relation, and consequently, one tolerance matrix is determined. The tolerance matrix on the attribute a is denoted by M_a.

The intersection of tolerance relations is still a tolerance relation, so the tolerance relation P_B determined by $B \subseteq A$ is defined as follows: $P_B = \bigcap_{b \in B} P_b$, and the corresponding matrix is $M_B = \bigcap_{b \in B} M_b$.

Property 6. For $S=(U,A)$ and $X,Y \subseteq A$, then $M_X \cap M_Y = M_{X \cup Y}$ and $M_{X \cup Y} \leqslant M_X, M_Y$.

Property 7. For $S=(U,A)$ and $X \subseteq Y \subseteq A$, then $M_Y \leqslant M_X$.

Property 8. For $S=(U,A)$ and $X,Y \subseteq A$, if $M_X \leqslant M_Y$ then $\forall Z \subseteq A, M_{X \cup Z} \leqslant M_{Y \cup Z}$, and if $M_X = M_Y$ then $\forall Z \subseteq A, M_{X \cup Z} = M_{Y \cup Z}$.

3.2 Decision tables and tolerance matrices

Let $S=(U,A)$ be an information system, where U is its universe, $A=C \cup D, C \cap D = \phi$, C is its conditional attribute set, D is its decision attribute set, and for every $u \in U$, $d(u) \neq ^*$ $(d \in D)$. We can define tolerance relation P_a of a conditional attribute $a \in C$ as follows:

$$u_iP_au_j \Leftrightarrow d(u_i) = d(u_j) \text{ or } a(u_i) = a(u_j) \vee a(u_i) = ^* \vee a(u_j) = ^* \ \forall u_i, u_j \in U$$

The tolerance matrix of the conditional attribute $a \in C$ is defined as:

$$M_a = (r_{ij})_{|U| \times |U|} = \begin{cases} 1, & d(u_i) = d(u_j) \\ 0, & a(u_i) \neq a(u_j), d(u_i) \neq d(u_j) \\ 0.5, & a(u_i) = a(u_j) \vee a(u_i) = ^* \vee a(u_j) = ^*, d(u_i) \neq d(u_j) \end{cases} \quad 1 \leqslant i,j \leqslant |U|$$

For any subset of the conditional attribute set $B \subseteq C$, its tolerance matrix is $M_B = \bigcap_{b \in B} M_b$.

Property 9. For $S=(U,C \cup D)$ and $X,Y \subseteq C$, then $M_X \cap M_Y = M_{X \cup Y}$ and $M_{X \cup Y} \leqslant M_X$, M_Y.

Property 10. For $S=(U,C \cup D)$ and $X \subseteq Y \subseteq C$, then $M_Y \leqslant M_X$.

Property 11. For $S=(U,C \cup D)$ and $X,Y \subseteq C$, if $M_X \leqslant M_Y$ then $\forall Z \subseteq C, M_{X \cup Z} \leqslant M_{Y \cup Z}$ and if $M_X = M_Y$ then $\forall Z \subseteq C, M_{X \cup Z} = M_{Y \cup Z}$.

3.3 Example

In the following, the incomplete information system in paper [7] is introduced. The information system is shown in Table 1, where the null value is denoted by "$*$", $U=\{1,2,3,4,5,6\}$ is its universe, $C=\{$Price, Mileage, Size, Max-Speed$\}$ is its conditional attribute set, and $D=\{d\}$ is its decision attribute set.

If the decision attribute is not included, the information system is a non-decision table and the tolerance matrices determined by attribute set and some of its subsets are the following ones.

Table 1 Information system about cars

Car	Price	Mileage	Size	Max-speed	D
1	High	High	Full	Low	Good
2	Low	*	Full	Low	Good
3	*	*	Compact	High	Poor
4	High	*	Full	High	Good
5	*	*	Full	High	Excel
6	Low	High	Full	*	Good

$$M_C = M_{C\setminus\{\text{Mileage}\}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_{C\setminus\{\text{Price}\}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad M_{C\setminus\{\text{Size}\}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

If the decision attribute is included, the tolerance matrices determined by attribute set and some of its subsets are described as follows.

$$M_C = M_{C\setminus\{\text{Price}\}} = M_{C\setminus\{\text{Mileage}\}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0.5 & 1 \\ 0 & 0 & 0 & 0.5 & 1 & 0.5 \\ 1 & 1 & 0 & 0 & 0.5 & 1 \end{bmatrix} \quad M_{C\setminus\{\text{Size}\}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0.5 & 0.5 & 0.5 \\ 1 & 1 & 0.5 & 1 & 0.5 & 1 \\ 0 & 0 & 0.5 & 0.5 & 1 & 0.5 \\ 1 & 1 & 0.5 & 1 & 0.5 & 1 \end{bmatrix}$$

4 Matrix presentation of dependency between attribute sets

Definition 8. Let $S=(U,A)$ be an information system, where $A=C\cup D, C\cap D=\phi$, C is the conditional attribute set, and D is the decision attribute set. S is a non-decision table when $D=\phi$(otherwise D is a decision table), and $X,Y\subseteq C, u,v\in U$, if $uP_Xv\Rightarrow uP_Yv$, then we say Y is strictly dependent on X, denoted by $X\to Y$(if $D\neq\phi$, we say Y is dependent on X with respect to D, denoted by $X\overset{D}{\to}Y$).

Property 12. $X\to Y\Leftrightarrow M_X\leqslant M_Y$ ($X\overset{D}{\to}Y\Leftrightarrow M_X\leqslant M_Y$).

Definition 9. If $X\to Y$ ($X\overset{D}{\to}Y$), and $Y\to X$ ($Y\overset{D}{\to}X$), then X and Y are identically dependent, denoted by $X\leftrightarrow Y$ ($X\overset{D}{\leftrightarrow}Y$).

Property 13. $X\leftrightarrow Y$ ($X\overset{D}{\leftrightarrow}Y$) $\Leftrightarrow M_X=M_Y$.

Definition 10. Let $x\in X\subset C$. If $X\leftrightarrow X\setminus\{x\}$ ($X\overset{D}{\leftrightarrow}X\setminus\{x\}$), then x is not important in X (with respect to D), otherwise x is important in X(with respect to D).

Property 14. x is not important in X(with respect to D) if and only if $M_X=M_{X\setminus\{x\}}$; x is important in X(with respect to D) if and only if $M_X<M_{X\setminus\{x\}}$.

Definition 11. For $\phi\subset X\subseteq C$, the set of all important attributes (with respect to D) in X is called the core of X (with respect to D), denoted by $Core(X)$, and $Core(X)=$

$\{x \in X \mid M_X < M_{X\setminus\{x\}}\}$.

Definition 12. $\phi \subset X \subseteq C$ is independent (with respect to D), if each element in X is important (with respect to D), else X is dependent (with respect to D).

Property 15. 1) If X is dependent (with respect to D), then $X \cup \{x_1, x_2, \cdots, x_k\}$ is still dependent (with respect to D), where $x_i \in C \setminus X, 1 \leqslant i \leqslant k$.

2) If X is independent (with respect to D), then $X \setminus \{x_1, x_2, \cdots, x_l\}$ is still independent (with respect to D), where $x_j \in X, 1 \leqslant j \leqslant l$.

Definition 13. The significance of $x \in X \subseteq C$ in X (with respect to D) is denoted by $sig_X(x)$, and defined as follows: $sig_X(x) = |M_{X\setminus\{x\}} - M_X|$, where $|M_Y|$ denotes the number of non-zero elements in tolerance matrix M_Y.

Definition 14. $X_0 \subseteq X \subseteq C$ is defined as a (relative) reduct of X if 1) $X_0 \leftrightarrow X (X_0 \overset{D}{\leftrightarrow} X)$, 2) X_0 is independent (with respect to D).

Property 16. $Core(X) = \bigcap R(X)$, where $R(X)$ denotes all reducts of X.

5 Attribute reduction based on tolerance matrix

We know that acquisition of all reducts of an information system is an NP-hard problem, so it needs heuristic knowledge to reduce the searching space. From the above, the tolerance matrix of an attribute subset should not be changed if an attribute, which is not important (redundant) in an attribute subset, is removed from the attribute subset. So we can obtain a reduct of an information system by removing redundant attributes one by one. The core is the mutual part of all reducts, so we can also increase attributes gradually from the core until the tolerance matrix is equal to the original one, and then we can obtain a reduct. In some cases, the minimal reduct is interesting, so we should select the most important attribute when we increase attributes.

We will present algorithms of the non-decision table and decision table based on tolerance matrix respectively as follows.

5.1 Attribute reduction algorithm for non-decision table

Algorithm 1. Input: non-decision table $S = (U, A)$.

Output: one reduct of $S = (U, A)$, denoted by B.

Step 1. Compute tolerance matrix M_A of $S = (U, A)$;

Step 2. Let $B = A$, repeat:

1) For all $x \in B$, compute $M_{B\setminus\{x\}}$;

2) Let $P = \{x \in B \mid M_B = M_{B\setminus\{x\}}\}$. If $P = \phi$, stop; else select $x \in P, B = B \setminus \{x\}$, go to 1).

Step 3. Output B, which is a reduct of $S = (U, A)$.

We analyze the time complexity of Algorithm 1. Each item of M_A should be computed in Step 1. The values of any two objects on $|A|$ attributes should be compared in the worst case. Because there are $|U|^2$ items in M_A, the time complexity of Step 1 is $O(|A||U|^2)$. In Step 2, the time complexity is $O(|A|^2|U|^2)$ when computing all $M_{B\setminus\{x\}}$ in 1), and the time complexity of 2) is $O(|A||U|^2)$. It needs to cycle $|A|-1$ times in the worst case, so the time complexity of Step 2 is $O(|A|^3|U|^2)$ and the whole time complexity of Algorithm 1 is $O(|A|^3|U|^2)$.

Beginning from the core, we increase non-core attributes gradually until the tolerance matrix is equal to that of the information system, and then we can obtain its reduct. The algorithm is introduced as follows.

Algorithm 2. Input: a non-decision table $S = (U, A)$;

Output: one reduct of $S = (U, A)$, denoted by B.

Step 1. Compute tolerance matrix of $S=(U,A)$, denoted by M_A;

Step 2. Compute the core of $S=(U,A)$, denoted by $Core(A)$;

Step 3. Let $B=Core(A)$, and repeat that:

1) If $M_B=M_A$, then stop; else go to 2);

2) Compute $M_{B\cup\{x\}}$, for all $x\in A\backslash B$, and let $x_0\in\{x|\min_{x\in A\backslash B}|M_{B\cup\{x\}}|\}$, $B=B\cup\{x_0\}$, go to 1);

Step 4. Output B, which is a reduct of $S=(U,A)$.

Similarly, We can conclude that the whole time complexity of Algorithm 2 is $O(|A|^3|U|^2)$.

5.2 Attribute reduction for decision table

As the computational method of tolerance matrix is similar for the decision table and the non-decision table, the computation of matrix is the same. We can generalize the attribute reduction for non-decision table to include the decision table as shown in the following example analysis.

6 Example analysis

We still use Table 1 to verify the validity of the algorithms presented in Section 5.

We use Algorithm 1 to reduce data table without considering of the decision attribute d.

Step 1. Compute M_C(see 3.3).

Step 2. Let $B=C$, compute $M_{C\backslash\{Price\}}$, $M_{C\backslash\{Mileage\}}$, $M_{C\backslash\{Size\}}$ and $M_{C\backslash\{Max\text{-}Speed\}}$, respectively (see 3.3). Because $M_{C\backslash\{Mileage\}}=M_C$, $B=C\backslash\{\text{Mileage}\}=\{\text{Price, Size, Max-Speed}\}$. Compute $M_{B\backslash\{Price\}}$, $M_{B\backslash\{Size\}}$ and $M_{B\backslash\{Max\text{-}Speed\}}$, respectively. None of them is equal to M_C, so stop.

Step 3. Output $B=\{\text{Price,Size,Max-Speed}\}$, which is a reduct of the non-decision table.

We can also obtain the reduct by using Algorithm 2.

For Table 1, we use Algorithm 2 to compute its relative reduction.

Algorithm 2.

Step 1. Compute M_C(see 3.3)

Step 2. Compute $Core(C)=\{\text{Size,Max-Speed}\}$

Step 3. Let $B=Core(C)=\{\text{Size,Max-Speed}\}$, compute M_B. Because $M_B=M_C$, stop

Step 4. Output $B=\{\text{Size,Max-Speed}\}$, which is the unique reduct of the decision table.

7 Conclusion

In the rough set theory, it is very important to examine computational methods for incomplete information systems. In this paper, we define a tolerance matrix based on tolerance relation, construct a one-to-one relationship between tolerance relation and tolerance matrix, and represent rough computational methods by matrix computation for incomplete information systems. Finally, we present two algorithms for attribute reduction based on tolerance matrix with polynomial time complexity for incomplete information systems, whose validity is experimentally verified.

References

1 Guan J W, Bell D A. Rough computational methods for information systems. *Artificial Intelligence*, 1998, **105**:77～103

2 Beauboueg T, Petry F E, Arora G. Information-theoretic measures of uncertainty for rough sets and rough relational databases. *Information Sciences*, 1998, **109**:185～195

3 Duntsch I, Gediga G. Uncertainty measures of rough set prediction. *Artificial Intelligence*, 1998, **106**:109～137

4 Miao Duo-Qian, Hu Gui-Rong. A heuristic algorithm for reduction of knowledge. *Journal of Computer Research & Development*, 1999, **36**(6):681～684(in Chinese)

5 Wang Guo-Yin, Yu Hong, Yang Da-Chun. Decision table reduction based on conditional information entropy. *Chinese Journal of Computers*, 2002, **25**(7): 759～766(in Chinese)

6 Guan J W, Bell D Z, Guan Z. Matrix computation for information systems. *Information Sciences*, 2001, **131**: 129～156

7 Kryszkiewicz M. Rough set approach to incomplete information systems. *Information Sciences*, 1998, **112**: 39～49

8 Hong Tzung-Pei, Tseng Li-Huei, Wang Shyue-Liang. Learning rules from incomplete training examples by rough sets. *Expert Systems with Applications*, 2002, **22**: 285～293

HUANG Bing Received his master degree from Sichuan University in 1999 and now is a Ph. D. candidate at Nanjing University of Science and Technology. His research interests include rough set theory and its applicantions.

HE Xin Ph. D. candidate at Nanjing University of Science and Technology. His research interest includes speech signal processing.

ZHOU Xian-Zhong Received his master degree in 1985 and Ph. D. degree from Nanjing University of Science and Technology in 1996. He is a professor at Nanjing University. His research interests include intelligent information processing and information system theory.

基于相容矩阵的粗计算

黄 兵[1] 何 新[1] 周献中[2]

[1](南京理工大学自动化系1004教研室 南京 210094)

[2](南京大学工程管理学院 南京 210093)

(E-mail: hbhuangbing@sohu.com; hexin_01@sina.com, zhouxizh@public1.ptt.js.cn)

摘 要 基于等价关系的经典粗糙集理论已取得了极大进展.但现实中的等价关系要求过于严格.因此,可将其放宽为相容关系.粗糙集理论中的粗计算方法一直是该理论的重要研究内容.本文在基于相容关系的基础上提出了相容矩阵的概念,建立了相容关系和相容矩阵间的一一对应关系,通过矩阵计算来刻画粗分析中的一系列计算方法;并利用相容矩阵提出了不完备信息系统的属性约简启发式算法,分析了算法的时间复杂度.通过实例说明了该方法是适用而有效的.

关键词 粗糙集,相容关系,相容矩阵,不完备信息系统,约简

中图分类号 TP18