ISSN 1009-3095 Journal of Zhejiang University SCIENCE V.3, No.5, P.532-537, Nov.-Dec., 2002
http://www.periodicals.com.cn; http://www.zju.edu.cn/English
http://www.zjupress.com; http://lib.zju.edu.cn/eindex.htm; jzu_s@mail.hz.zj.cn

Meta-information generation in distributed information system*

SU Jian(苏健)†, GAO Ji(高济)

(*Department of Computer Science and Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: jiansue@cs.zju.edu.cn

Received Nov. 3, 2001; revision accepted Jan. 28, 2002

Abstract: The authors discuss the concept of meta-information which is the description of information system or its subsystems, and proposes algorithms for meta-information generation. Meta-information can be generated in parallel mode and network computation can be used to accelerate meta-information generation. Most existing rough set methods assume information system to be centralized and cannot be applied directly in distributed information system. Data integration, which is costly, is necessary for such existing methods. However, meta-information integration will eliminate the need of data integration in many cases, since many rough set operations can be done straightforward based on meta-information, and many existing methods can be modified based on meta-information.

Key words: Meta-information, Distributed information system, Rough set
Document code: A **CLC number:** TP18

INTRODUCTION

Rough set theory has been extensively researched and become a powerful tool for knowledge discovery and intelligence data analysis (Chan, 1998; Komorowski et al., 1999; Slowinski et al., 2000; Grzymala-Busse et al., 2000; Zhong et al., 2001).

In original rough set theory, an information system is represented as a single data table describing a set of objects by a set of attributes (Pawlak, 1991), and information system with this kind of data representation style is called centralized information system. In this paper, data representation of information system is extended to multiple tables; such kind of information system is called distributed information system. Distributed information system consists of a family of distributed sub information systems (or subsystem for short). Each subsystem is represented as a single data table, which is similar to a centralized information system.

Most existing methods assume the information system to be centralized and cannot be applied directly in distributed information system. However, in network computation environment, the information system is typically distributed. In order to use the existing methods, it is necessary to integrate multiple data tables into one data table. But data integration is costly work, especially when the data is heterogeneous and change quickly, and the centralized data table can easily be the bottleneck of processing and slowdown the speed.

In this paper, we present the concept of meta-information, which is the description of information system or its subsystems. Algorithms for meta-information generation are also proposed. We prefer meta-information integration (generating meta-information of information system from that of its subsystems) to data integration, since many rough set operations can be done straightforward based on meta-information and cost of meta-information integration is much less.

In the rest of this paper, we first describe the basic definitions and the algorithms for meta-information generation, and then discuss some rough set operations based on meta-information and finally give the conclusions.

* Project (No. 69773019) supported by National Natural Science Foundation of China

BASIC DEFINITIONS

Definition 1 Information system $S = <U, A, V, f>$, (Guan et al., 1998) where:

1. $U = \{u_1, u_2, \cdots, u_n\}$, $n = |U|$ and $n > 0$, U is called a universe or object space.

2. $A = C \cup D$. A is an attribute set, C is a condition attribute set, and D is a decision attribute set.

3. $V = \bigcup V_p$, where $p \in A$, and V_p is the domain of attribute P.

4. $f: U \times A \rightarrow V$, for $\forall x \in U$, $q \in A$, $f(x, q) \in V_q$

Let $U = U_1 \cup U_2 \cup \cdots \cup U_m$, and $U_i \cap U_j = \phi$ if $i \neq j$. Let $S_i = <U_i, A, V, f>, 1 \leqslant i \leqslant m$. S_i is called sub information system (or subsystem for short) of S. Information system S consists of a family of subsystem S_i.

Definition 2 Indiscernible relation $IND(B)$ on U is defined as

$IND(B) = \{(x, y) \in U^2 \mid (\forall a)(a \in B \rightarrow f(x, a) = f(y, a))\}$.

Obviously, $IND(B)$ is equivalence relation. $IND(B)$ will be written as B for short. Let $B^* = U/B$ denote the family of classes. $X \in C^*$ is called condition class, and $Y \in D^*$ is called decision class.

Definition 3 Let $S = <U, A, V, f>$, $C^* = \{X_1, X_2, \cdots, X_{|C^*|}\}$, $D^* = \{Y_1, Y_2, \cdots, Y_{|D^*|}\}$. $M = (m_{ij})$ is called class matrix of S, where m_{ij} is defined as

$$m_{ij} = |X_i \cap Y_j|.$$

Class matrix describes the intersection of condition class and decision class. $|X_i \cap Y_j|$ requires less storage compared with $X_i \cap Y_j$.

Definition 4 Let $X \in B^*$. $D_X = (x, |X|)$ is called class description of X, where $x \in X$ is characteristic element of X, and $|X|$ is the cardinality of X.

Every element in a class is indiscernible to each other, so that one characteristic element with the cardinality of class is sufficient to describe a class. Using description of class rather than class itself will also save much storage.

Definition 5 Let $S = <U, A, V, f>$. $I = <M, F_C, F_D>$ is meta-information of S, where M is class matrix, $F_C = \{D_X \mid X \in C^*\}$, and $F_D = \{D_Y \mid Y \in D^*\}$. F_C is called condition class description set and F_D is called decision class description set.

In the following section, we will discuss how to generate meta-information of S from its subsystems.

META-INFORMATION GENERATION

1. Algorithms

Meta-information generation includes construction of class matrix, condition class description set, and decision class description set. Information system $S = <U, A, V, f>$. Let $\{U_1, U_2, \cdots, U_n\}$ be a partition of U, thus S is split into n subsystems. Let $S_i = <U_i, A, V, f>$ be the i-th subsystem of S, where $1 \leqslant i \leqslant n$. Let $I = <M, F_C, F_D>$ denote meta-information of S. Let $I_i = <M_i, F_C^i, F_D^i>$ be meta-information of S_i. In the following, we will discuss how to unify $I_1, I_2, \cdots, I_n$ into I.

Let $B^* = U/B$ be a partition of U constructed from B. Let $B_i^* = U_i/B$ be partition of U_i constructed from B. Let $P = \bigcup B_i^*$, $1 \leqslant i \leqslant n$.

Definition 6 For $X \in P$, X-relative subclass set is defined as

$Q(X) = \{Y \mid Y \in P \wedge (\forall x)(\forall y)(x \in X \wedge y \in Y \rightarrow xBy)\}$.

Let $H(X) = \{D_Y \mid Y \in Q(X)\}$ be class description set corresponding to $Q(X)$.

Proposition 1 (1) $\bigcup \{Y : Y \in Q(X)\} \in B^*$; (2) $\forall Z \in B^*$, $\exists X \in P$, $Z = \bigcup \{Y : Y \in Q(X)\}$; (3) $Y \in Q(X) \Leftrightarrow Q(X) = Q(Y)$.

Proof: (1) Let $E = \bigcup \{Y : Y (Q(X)\}$. By definition 6, $\forall x, y \in E$, xBy, $\therefore \exists Z \in B^*: E \subseteq Z$. Meanwhile, $\forall x \in Z$, $\exists X \in P$, $x \in X$, By definition 6, $X \in Q(X)$, $\therefore E \supseteq Z$. $\therefore E = Z$. $\therefore \bigcup \{Y : Y \in Q(X)\} \in B^*$.

(2) and (3) can also be proved by definition 6, and proof is omitted.

By proposition 1, we know that any class $Z \in B^*$ can be computed from some specific X-relative subclass set, i.e., Z is the union of all subclasses in the X-relative subclass set. Let $G = \{D_X \mid X \in P\}$ denote the class description set corresponding to P. Obviously, G and P is one-to-one correspondence. Let $D_X=(x,|X|)$, $D_Y=(y,|Y|)$, where $D_X, D_Y \in G$. Let $Z=\bigcup\{Y: Y\in Q(X)\}$.

Theorem 1 (1) $xBy \Leftrightarrow Y \in Q(X) \Leftrightarrow D_Y \in H(X)$; (2) $D_Z=(y,\sum|Y|)$.

It is easy to prove this theorem by proposition 1. Proof is omitted.

By theorem 1(2), we can compute the class description D_Z of some class $Z\in B^*$, provided that we have already got $H(X)$. Theorem 1(1) indicates that $H(X)$ can be constructed from G directly, and it is not necessary to produce P first and then produce $H(X)$. Thus, we can compute all D_Z from G. G need less storage and processing time than P, since class description generally requires less storage and processing time. In the following, we will discuss how to generate description set.

Algorithm A(description set generation)

Let $H=\{H(X)|X\in P\}$ denote set of $H(X)$. Let $F=\{D_Z|Z\in B^*\}$ denote description set. Given G, algorithm A establishes F and H.

1. $F=\phi$, $H=\phi$。

2. Write G as $\{D_{X_1}, D_{X_2}, \cdots, D_{X_n}\}$, where $n=|G|$. Let $D_{X_1}=(x_1,|X_1|)$ and establish $H(X_1)$ from G.

2a. $i=1$, $H(X_1)=\phi$

2b. Let $D_{X_i}=(x_i,|X|)$, if x_1Bx_i, then $H(X)=H(X)\cup\{D_{X_i}\}$

2c. $i=i+1$, if $i>|G|$, then go to 3, else go to 2b。

3. Establish $D_Z=(z,|Z|)$ from $H(X_1)$. Write $H(X_1)$ as $\{D_{Y_1}, D_{Y_2}, \cdots, D_{Y_m}\}$, where $m=|H(X_1)|$. Let $D_{Y_j}=(y_j,|Y_j|)$.

3a. $j=1$, $|Z|=0$, $z=y_1$

3b. $|Z|=|Z|+|Y_j|$.

3c. $j=j+1$, if $j>m$, then go to 4, else go to 3b.

4. $F=F\cup\{D_Z\}$, $H=H\cup H(X_1)$。

5. $G=G-H(X_1)$, if $G=\phi$, then this algorithm ends, else go to 2.

Algorithm A can be proved by theorem 1 although proof is omitted. Time complexity of algorithm A is $O(|G|^2)$. Since $|G|\leqslant|U|$, time complexity is $O(|U|^2)$ in the worst case.

Let $I_i=<M_i, F_C^i, F_D^i>$ denote meta-information of subsystem $S_i=<U_i, A, V, f>$. Let $G_C=F_C^1\cup F_C^2\cup\cdots\cup F_C^n$. Let $G_D=F_D^1\cup F_D^2\cup\cdots\cup F_D^n$. If all meta-information of subsystem are available, we can get G_C and G_D immediately. Then, with G_C and G_D as input, algorithm A can establish $F_C=\{D_Z|Z\in C^*\}$, H_C and $F_D=\{D_Z|Z\in D^*\}$, H_D.

Let subsystem $S_i=<U_i, A, V, f>$ and Let $I_i=<M_i, F_C^i, F_D^i>$ denote meta-information of S_i. Let C_i^* and D_i^* denote partition of U_i constructed from relation C and D, respectively. Let $P_C=C_1^*\cup C_2^*\cup\cdots\cup C_n^*$, $P_D=D_1^*\cup D_2^*\cup\cdots\cup D_n^*$. Let $S=<U, A, V, f>$ denote information system and $I=<M, F_C, F_D>$ denote meta-information of S, where $M=(m_{ij})$ is the class matrix of information system S. C^* and D^* denote partition of U constructed from relation C and D, respectively. Let $X\in C^*$, $Y\in D^*$. Proposition 1(2) shows that $\exists X_1\in P_C$, $Q(X_1)=\{X_1, X_2, \cdots X_k\}$. Let $X=X_1\cup X_2\cup\cdots\cup X_k$. Similarity, $\exists Y_1\in P_D$, $Q(Y_1)=\{Y_1, Y_2, \cdots Y_l\}$. Let $Y=Y_1\cup Y_2\cup\cdots\cup Y_l$.

Theorem 2 (1) $|X\cap Y|=\sum_{1\leqslant i\leqslant k}\sum_{1\leqslant j\leqslant l}|X_i\cap Y_j|$; (2) if $X_i\in C_a^*$, $Y_j\in D_b^*$, and $a\neq b$, then $|X_i\cap Y_j|=0$; (3) if $X_i\in C_a^*$ is the p-th class of C_a^*, and $Y_j\in D_a^*$ is the q-th class of D_a^*, then $|X_i\cap Y_j|=m_{pq}^a$.

Proof (1) $X\cap Y=(X_1\cup X_2\cup\cdots\cup X_k)\cap(Y_1\cup Y_2\cup\cdots\cup Y_l)=(\bigcup_{1\leqslant i\leqslant k}X_i)\cap(\bigcup_{1\leqslant j\leqslant l}Y_j)=\bigcup_{1\leqslant i\leqslant k}(X_i\cap(\bigcup_{1\leqslant j\leqslant l}Y_j))=\bigcup_{1\leqslant i\leqslant k}(\bigcup_{1\leqslant j\leqslant l}(X_i\cap Y_j))=\bigcup_{1\leqslant i\leqslant k}\bigcup_{1\leqslant j\leqslant l}(X_i\cap Y_j)$

$\therefore |X\cap Y|=\sum_{1\leqslant i\leqslant k}\sum_{1\leqslant j\leqslant l}|X_i\cap Y_j|$

(2) $X_i\in C_a^*\Rightarrow X_i\subseteq U_a$. Similarly, $Y_j\subseteq U_b$. $a\neq b\Rightarrow U_a\cap U_b=\phi\Rightarrow X_i\cap Y_j=\phi\Rightarrow|X_i\cap Y_j|=0$.

(3) can be proved by class matrix definition and proof is omitted.

Theorem 2 shows the relation between M and M_i. By theorem 2(2), we know that $|X_i\cap Y_j|=0$ if X_i, Y_j is from different subsystem. Theorem 2(3) indicates that if $X_i\in C_a^*$ is the p-th class of C_a^* and $Y_j\in D_a^*$ is the q-th class of D_a^*, then $|X_i\cap Y_j|=m_{pq}^a$. Thus, we can establish class matrix M from M_i, which is the class matrix of subsystem.

In the following, algorithm B will construct meta-information of information system from that of its subsystems.

Algorithm B (meta-information generation)

Given $I_i = <M_i, F_C^i, F_D^i>$ where $1 \leqslant k \leqslant n$. Algorithm B establishes meta-information $I = <M, F_C, F_D>$.

1. $G_C = F_C^1 \cup F_C^2 \cup \cdots \cup F_C^n$, $G_D = F_D^1 \cup F_D^2 \cup \cdots \cup F_D^n$.

2. Run algorithm A to produce $F_C = \{D_X \mid X \in C^*\}$, H_C and $F_D = \{D_Y \mid Y \in D^*\}$, H_D.

3. $i=1$, $j=1$. Write F_C as $\{D_{X_1}, D_{X_2}, \cdots, D_{X_{F_C}}\}$, Write F_D as $\{D_{Y_1}, D_{Y_2}, \cdots, D_{Y_{F_D}}\}$. $\not\subset$

4. Get $D_{X_i} \in F_C$ and $H(X_1) \in H_C$.

5. Get $D_{Y_j} \in F_D$ and $H(Y_1) \in H_D$,

6. Compute m_{ij}.

6a. $l=1$, $k=1$, $|X \cap Y| = 0$;

6b. Get $D_{X_l} \in H(X_1)$ and find the position (written as p) of D_{X_l} in F_C^a

6c. Get $D_{Y_k} \in H(Y_1)$, and find the position (written as q) of D_{Y_k} in F_D^b

6d. If $a = b$, then $|X \cap Y| = |X \cap Y| + m_{pq}^a$,

6e. $k = k+1$; if $k \leqslant |H(Y_1)|$, then go to 6c

6f. $l = l+1$; if $l \leqslant |H(X_1)|$, then go to 6b

6g. $m_{ij} = |X \cap Y|$

7. $j = j+1$, if $j \leqslant |F_D|$, then go to 5

8. $i = i+1$, if $i \leqslant |F_C|$, then $j=1$, and go to 4, else this algorithm ends.

Theorem 2 can prove algorithm B, although proof is omitted.

Given all the meta-information of subsystem, algorithm B can merge these meta-information into that of information system. The time complexity of step 2 is $O(\max(|G_C|^2, |G_D|^2))$ and that of step 3 to step 8 is $O(|G_C| \times |G_D|)$. Thus, the time complexity T of algorithm B is $O(\max(|G_C|^2, |G_D|^2, |G_C| \times |G_D|))$. If $|G_C| \geqslant |G_D|$, then $T = O(|G_C|^2)$, and if $|G_C| < |G_D|$, then $T = O(|G_D|^2)$. In the worst case that all the condition classes or decision classes are singletons, i.e., $|G_C| = |U|$ or $|G_D| = |U|$, then $T = O(|U|^2)$. Generally, $|G_C|$ and $|G_D|$ is much less than $|U|$. However, $|G_C|$ and $|G_D|$ may be larger than expected sometimes. In this case, number of classes can be reduced by imposing restrictions to class and discarding all useless classes. In this way, we can control the time complexity of algorithm B. Meta-information I_i needs to be transported through network sometimes, and its main storage cost is represented by the class matrix, space complexity of which is $O(|F_C^i| \times |F_D^i|)$. We can also control the number of classes by setting restrictions to classes, and reduce the storage cost. It may also reduce the storage cost to store the class matrix in the form of sparse matrix.

Algorithm B can unify meta-information of subsystems into that of information system. Obviously, if we want to use algorithm B to establish the meta-information of information system, we must get all the meta-information of subsystem first. A problem is how to establish meta-information of subsystem.

Let $<M, F_C, F_D>$ be meta-information of $<\{x\}, A, V, f>$. Obviously, $F_C = F_D = \{(x, 1)\}$ and M is 1×1 matrix ($m_{11} = 1$). In other words, the meta-information of $<\{x\}, A, V, f>$ can be obtained immediately without having to use algorithm B. Thus, we can generate meta-information I of $S = <U, A, V, f>$ in the following steps: (1) split S into $|U|$ subsystem with single object; (2) get meta-information of these subsystem directly, and use algorithm B to produce I. As $|G_C| = |G_D| = |U|$, the time complexity of this method is $O(|U|^2)$, which meet the worst case of algorithm B. We had proposed another method for meta-information generation of each subsystem (Su et al, 2001), which establish meta-information incrementally and need less time.

2. Strategies to split information system

Distributed information system consists of a family of subsystems. Meta-information generation of these subsystems can be run independently and concurrently, and the meta-information of these subsystems can be unified into that of information system. So we can make use of network computation resources to carry out parallel meta-information generation. We have the following strategies to split information system into subsystems.

(1) Data located in different places should be split into different subsystem to avoid data inte-

gration.

(2) Heterogeneous data is split into different subsystem to avoid data type transformation.

(3) Huge volume of data should be split into many subsystems to increase the degree of parallelism.

Suppose we have N processes, which may run in different computer, for generating meta-information of subsystem. Obviously, the degree of parallelism is up to N. Let L be the number of subsystems. In order to achieve satisfactory degree of parallelism, L should be larger than N. But on the other hand, L cannot be too big since more subsystem means more traffic and more time to unify meta-information. The computation power of the process should also be taken into consideration. More tasks should be assigned to process with higher computation power.

There is no unified solution to split information system. Let V be the size (or object number) of the information system, and Let s be smallest size of subsystem. Based on analysis above, we split information into L subsystems, where

$$L = \min \{k * N, \lceil V/s \rceil\}, \; k \geqslant 1.$$

EXPERIMENT

In our experiment, we generate 13 M random data (about 1M object) and store them in two Oracle 8 databases. Every object has 7 condition attributes and 1 decision attribute. Each attribute has 3 values. We set $k = 4$, $s = 10K$. Experiment was carried out in 10/100M Ethernet network environment. The experiment results of meta-information generation are shown in Table 1.

Table 1 Experiment results

Process number	Subsystem number	Time(s)
1	4	423
2	8	276
3	12	173
4	16	131

Experiment results showed that running time decreased when process number increased. However, speed is not necessarily in proportion to process number. The computation power of these processes is not identical, and process with higher computation power is much faster than others, and so often become idle. In this experiment, information system is split statically and the size of subsystem cannot be changed. In future, a dynamic schedule algorithm will be used to make the processes as busy as possible, and the information system will be split dynamically. If some process becomes idle, the busy process will split its subsystem into some smaller subsystems, and assign task of meta-information generation of smaller subsystem to idle process.

10/100M Ethernet network is fast enough to eliminate the impact of delay in the communication.

DISCUSSION

Meta-information can be stored in database for future reuse. Dynamic increased data can be handled in a quite natural way. New data will be viewed as a new subsystem, meta-information of which can be merged with stored meta-information to produce a new one.

Base on meta-information, many rough set operations can be done straightforward. Due to limitation of paper length, we will just give one example. In the following, we will show how to compute $|POS_B(D)|$ from meta-information, where $B \subseteq C$.

Let $I = \langle M, F_C, F_D \rangle$ be meta-information, where $F_C = \{(x_1, 100), (x_2, 200), (x_3, 300)\}$, $F_D = \{(y_1, 100), (y_2, 500)\}$; M is shown in Table 2. If $x_2 B x_3$, then we have $F_B = \{(x_1, 100), (x_2, 500)\}$. Obviously, $[x_1]_B = [x_1]_C$, so that $|[x_1]_B \cap [y_1]_D| = 0$, and $|[x_1]_B \cap [y_2]_D| = 100$. $[x_2]_B = [x_2]_C \cup [x_3]_C$, so that $|[x_2]_B \cap [y_1]_D| = 50 + 50$ and $|[x_2]_B \cap [y_2]_D| = 150 + 250$. Thus, we get the matrix in Table 3. Obviously, $|POS_B(D)| = 100$. We can see that computation of $|POS_B(D)|$ is just a few addition operations.

Operation on meta-information is always simpler and cost less than on data table directly. In future, we will modify the existing rough set methods to make them work on meta-information. Since parallel computation can accelerate rough set operation, such as parallel reduction

(Susmaga, 1998), our future work will also be aimed to design parallel rough set algorithm based on meta-information.

Table 2 Class matrix 1

	$[y_1]_D$	$[y_2]_D$
$[x_1]_C$	0	100
$[x_2]_C$	50	150
$[x_3]_C$	50	250

Table 3 Class matrix 2

	$[y_1]_D$	$[y_2]_D$
$[x_1]_B$	0	100
$[x_2]_B$	50 + 50	150 + 250

CONCLUSIONS

In this paper, we discuss meta-information of information system and propose algorithms for meta-information generation. Since meta-information can be generated in parallel mode, we can utilize network computation to accelerate meta-information generation. In distributed information system, we prefer meta-information integration (generating meta-information of information system from that of its subsystems) to data integration, since many rough set operations can be done straightforward based on meta-information and cost of meta-information integration is much less.

We can use the obtained results in this paper to analyze distributed databases.

ACKNOWLEDGEMENTS

We thank the reviewers of this paper for their valuable comments and suggestions

References

Chan, C. C., 1998. A rough set approach to attribute generalization in data mining. *Journal of Information Sciences*. **107**: 169 – 176.

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

Grzymala-Busse, J. W., Ziarko, W., 2000. Data mining and rough set theory. *Communications of the ACM*, **43**, 108 – 109

Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A., 1999. Rough sets: A Tutorial. In: Pal, S. K., Skowron A. ed., Rough fuzzy hybridization. A new trend in decision-making. Springer, p. 3 – 98.

Pawlak, Z., 1991. Rough set: Theoretical Aspects of Reasoning About Data. Kluwer Academic, Dordrecht, The Netherlands.

Slowinski, R., Vanderpooten, D., 2000. A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge and Data Engineering*. **12**(2): 331 – 336.

Susmaga, R., 1998. Parallel Computation of Reducts. *In*: Polkowski, L. and Skowron, A., (Eds), Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science, vol. 1424. Springer-Verlag, Berlin, p. 450 – 457.

Su, J., Gao, J., 2001. A meta-information based method for incremental rough set rule generation. *Pattern Recognition and Artificial Intelligence*, **14**(4): 428 – 433 (in Chinese).

Zhong, N., Dong, J. Z., Ohsuga, S., 2001. Using rough sets with heuristics for feature selection. *Journal of Intelligent Information Systems*, **16**: 199 – 214.