Reduct Generation in Information Systems

G.Ganesan, D.Latha and C.Raghavendra Rao

***Abstract*— In any information system, the reducts are useful in classifying data. Janusz Starzyk developed an algorithm for computing reducts using strong equivalence and the law of expansion on the data. However, implementation of this algorithm is cumbersome for huge volume of data. This paper deals with a technique for obtaining the reduct of the entire system by partitioning it into two with respect to records and obtaining the reducts of the two subsystems and the 'between reducts'. Further, it also deals with a technique for combining the reducts computed at the clients to obtain global reducts.**

Index Terms— reduct, information system, strong equivalence, expansion law, between reducts.

I. Introduction

An information system consists of various data. Sometimes there may be several attributes, which are not necessary for rule discovery. In order to reduce time complexity for rule discovery, these redundant attributes or features have to be eliminated. Hence, there are several methods proposed for selecting the features, which are termed, to be reducts [1,5,6,10].

In the theory of rough sets, Skowron developed the concept of discernibility matrices [7], which helps in computing reducts. The minimal set of attributes, which intersects all the elements of the discernibility matrix, is called a reduct. In general, the reducts thus obtained may not be unique. Hence, sometimes it is difficult to list all reducts. So, Janusz Starzyk [5] developed an algorithm for computing reducts. However, this algorithm is limited and it is necessary to modify for the following situations.

Consider a set of records containing information about the qualification, years of experience, skill set and the performance level in various projects of employees in a software industry. For a particular project, when two committees evaluate employees independently using the above attributes, two sets of decision are made. It is obvious that the decisions taken by them depend on their choice of attributes. Hence, in order to have the collective reduct, it is necessary to derive a tool of getting it using the reducts given by the committees. Here, in section VII, we propose an algorithm for similar situations.

Moreover, for the huge volume of data, computation of the reduct is NP-hard. Hence, in this paper, the system is subdivided into two subsystems with respect to records and the reducts of each subsystem and the 'relational reduct' are found. Using them, in section VIII, we modify the reduct generation algorithm to compute the reduct of the entire system.

First, we shall discuss the elimination method, which is useful in finding the reducts. In this method it is necessary to check all possible combination of data to find the reduct. Hence it is effective only in a system with limited number of records and attributes

II. Elimination method

In this method, first we have to eliminate all duplication of records in the information system. Next it is necessary to check whether the system is non ambiguous. In general, the ambiguity arises when two identical hypothesis give different conclusions. (for example, 1+2=3 and 1+2=8). Whenever such ambiguity arises, both the records are to be eliminated.

If there is no ambiguity, then the set of attributes is called a reduct. This method is called the elimination method.

III. Rough Sets

In 1982, Pawlak introduced the theory of Rough sets [9,11]. This theory was initially developed for a finite universe of discourse in which the knowledge base is a partition, which is obtained by any equivalence relation defined on the universe of discourse.

Let U be any finite universe of discourse. Let R be any equivalence relation defined on U. Clearly, the equivalence relation partitions U. Here, (U, R) which is the collection of all equivalence classes, is called the approximation space. Let $W_1, W_2, \ldots, W_n$ be the elements of the approximation space (U,R). This collection is called as knowledge base. Then for any subset A of U, the lower and upper approximations are defined respectively as follows:

$$\underline{R}A = \cup\{W_i / W_i \subseteq A\}$$

$$\overline{RA} = \cup\{W_i / W_i \cap A \neq \varphi\}$$

The ordered pair $(\underline{R}A, \overline{RA})$ is called a rough set. In general, $\underline{R}A \subseteq A \subseteq \overline{RA}$. If $\underline{R}A = \overline{RA}$ then A is called exact. The lower

G.Ganesan is working as Associate Professor of Mathematics in Jayaprakash Narayan College of Engineering, Mahabubnagar, Andhra Pradesh, India {elgee110201@yahoo.com}

D.Latha is with Adarsh Post Graduate College of Computer Sciences, Venkateshwara Colony, Mahabubnagar, Andhra Pradesh, India {corida_durai@yahoo.com}

C.Raghavendra Rao is working as Reader in Department of Mathematics University of Hyderabad, Hyderabad, Andhra Pradesh, India {crrsm@uohyd.ernet.in}

approximation of A is called the positive region of A and is denoted by POS(A) and the complement of upper approximation of A is called the negative region of A and is denoted by NEG(A). Its boundary is defined as BND(A)= $\overline{RA}$-$\underline{R}$A. Hence, it is trivial that if BND(A)=φ, then A is exact. However, the equivalent definitions of rough sets are dealt in [2].

This approach provides a mathematical tool that can be used to find out all possible reducts. However, this process is NP-hard [3,4], if the number of elements of the universe of discourse is large. As there is a one-to-one correspondence between the knowledge base and the knowledge representation, the theory can be adopted for the decision tables in information systems.

IV. Rough Sets in Information Systems

In the theory of rough sets, the decision table [4,8] of any information system is given by T=(U, A, C, D), where U is the universe of discourse, A is a set of primitive features, C and D are the subsets of A called condition and decision features respectively.

For any subset P of A, a binary relation IND (P), called the indiscernibility relation is defined as IND (P)={(x,y)∈UxU : a(x)=a(y) for all a in P}

Denote the classes obtained by the relation IND (P) by U/IND (P) or U/P. For the indiscernibility relation IND(R), the lower and upper approximations are defined as

$$\underline{R}X = \cup\{Y \in \frac{U}{R} : Y \subseteq X\} \text{ and}$$

$$\overline{R}X = \cup\{Y \in \frac{U}{R} : Y \cap X \neq \Phi\} \text{ respectively.}$$

The classes U/IND(C) and U/IND (D) are called condition and decision classes respectively.

The C-Positive region of D is given by $POS_C(D)= \bigcup_{X \in U/D} \underline{C}X$.

A. Dispensable and indispensable Features

Let c∈C. a feature c is dispensable in T, if $POS_{C-\{c\}}(D)= POS_C(D)$; otherwise the feature c is called indispensable in T. If c is an indispensable feature, deleting it from T makes T to be inconsistent. T is said to be independent if all the features of it are indispensable.

B. Reduct and Core

A set of features R in C is called a reduct, if T'=(U, A, R, D) is independent and $POS_R(D)=POS_C(D)$. In other words, a reduct is the minimal feature subset preserving the above condition.

The set of all features indispensable in C is denoted by CORE(C). In other words, CORE(C) = ∩RED(C) where RED(C) is the set of all reducts of C.

C. Discernibility Matrix

A.Skrowron introduced the representation of the decision table into discernibility matrix to compute reduct. Let T=(U,A,C,D) be a decision table, with $U=\{x_1,x_2,\ldots,x_n\}$. By a discernibility matrix of T, denoted M(T), we will mean n x n matrix defined as

$m_{ij}=\{a\in C:a(x_i)\neq a(x_j)\wedge(d\in D, d(x_i)\neq d(x_j))\}$ for i,j=1,2,…,n

In the decision table, two attributes 'x' and 'y' are said to be strongly equivalent if they appear always together in the elements of the discernibility matrix. Each element can be viewed as the disjunctive expression. i.e., if an element of the discernibility matrix is a,b,c then it can be viewed as a∨b∨c. The discernibility function is given by taking the conjunction of the disjunctive expressions of the discernibility matrix.

Example: Consider the knowledge representation system given below with C={a,b,c,d} and D={E}.

	a	b	c	d	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

The discernibility matrix is given by

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	---					
x_3	b,c,d	b,c				
x_4	b	b,d	c,d			
x_5	a,b,c,d	a,b,c	---	a,b,c,d		
x_6	a,b,c,d	a,b,c	---	a,b,c,d	---	
x_7	---	---	a,b,c,d	a,b	c,d	c,d

Using the discernibility matrix, the reducts of the decision table can be found, which is discussed below.

D. Core and Reducts through Discernibility matrix

The core can be defined as the set of all singleton entries in the discernibility matrix. The reduct is the minimal element in the discernibility matrix, which intersects all the elements of the discernibility matrix. The reducts can be obtained by using the Reduct generation algorithm.

In order to proceed further, it is necessary to know the expansion law, which is used in the algorithm. Here, the elements of the discernibility matrix are viewed in OR form. For example, the element {b,c,d} is viewed as b∨c∨d. Further the entire matrix can be written by using the connective AND.

Here, one may mislead the above, by treating AND and OR with usual conjunction and disjunction. Here, as the attributes are not 0 or 1, they are to be viewed as absent or present accordingly.

V. Expansion Law

(a) find the attribute X that occurs most frequently (at least twice)
(b) Apply AND of X and all other OR form of elements of the discernibility matrix which do not contain X
(c) Apply the connective AND between the OR form of all the elements, in which if the element contains X eliminate X.
(d) Combine the elements obtained from (a) and (b) by AND

Example: Consider the elements of the discernibility matrix be {{a,b,e},{a,b},{a,c},{d}}. The discernibility relation is given by (a∨b∨e)∧ (a∨b) ∧(a∨c)∧d
Here the element 'a' occurs often.
On applying AND `a' with `d', we get {a}∧{d}={a,d}, say as component 1
On applying AND for b∨e, b and c we get (b∨e)∧(b)∧(c)= {b, c} ∧{b, c, e}, say as component 2
The Integrated form is {a, d}, {b,c},{b,c,e}

VI. Reduct Generation Algorithm

Now, we discuss the algorithm proposed by Janusz Starzyk, Dale E.Nelson and Kirk Sturtz for reduct generation [5].

A. Algorithm

Given $f=f_1 \wedge f_2 \wedge \ldots \wedge f_t$ is the discernibility function

Step 1 Apply absorption law to eliminate all disjunctive expressions, which are supersets of another disjunctive expression

Step 2: Replace each set of strongly equivalent attributes by dummy variable

Step 3: Select the attribute, which belongs to the large number of conjunctive sets, numbering at least two, and apply the expansion law.

Step 4: Repeat steps 1 to 3 until the expansion law cannot be applied for each component.

Step 5: Substitute all strongly equivalent classes for their corresponding attributes.

Step 6: Calculate the reducts in each component.

Step 7: Write the Integrated reduct.

The above algorithm is illustrated by the following example.

Example: Consider the discernibility relation $F=\{a \vee b \vee c \vee f\} \wedge \{ b \vee d\} \wedge \{ a \vee d \vee e \vee f\} \wedge \{ a \vee b \vee c \vee d\} \wedge \{ b \vee d \vee e \vee f\} \wedge \{ c \vee d\}$

On applying absorption law, as $\{ b \vee d\} \subseteq \{ a \vee b \vee c \vee d\}$, we have, $\{ b \vee d\} \wedge \{a \vee b \vee c \vee d\}=\{ b \vee d\}$. Similarly, $\{ b \vee d \vee e \vee f\} \wedge\{ b \vee d\} =\{ b \vee d\}$. Hence, the discernibility relation becomes $F=\{a \vee b \vee c \vee f\} \wedge\{ b \vee d\} \wedge \{ a \vee d \vee e \vee f\} \wedge \{ c \vee d\}$.

It is observed that {a,f} are strongly equivalent. Denote a∨f=M. Hence, the discernibility relation becomes $F=\{M \vee b \vee c\} \wedge\{b \vee d\} \wedge \{M \vee d \vee e\} \wedge \{c \vee d\}$.

The attribute 'd' appears most frequently. Using it apply expansion law: $F=[\{d\}\wedge\{M \vee b \vee c\}]\wedge[\{M \vee b \vee c\} \wedge\{b\} \wedge \{M \vee e\} \wedge \{c\}]$. By applying absorption law in component 2, we have, $F=[\{d\}\wedge\{M \vee b \vee c\}]\wedge [\{b\} \wedge \{M \vee e\} \wedge \{c\}]$

Now all the components are in simple form.

On replacing M by a∨f, we have, $F=[\{d\}\wedge\{ a \vee f \vee b \vee c\}]\wedge [\{b\} \wedge \{ a \vee f \vee e\} \wedge \{c\}]$

The reduct of the first component is {a,d},{d,f},{b,d},{b,c} and the reduct of the second component is {a,b,c}{b,c,f},{b,c,e}

Hence, the Integrated reduct is {a,d},{d,f},{b,d},{b,c}, {a,b,c}{b,c,f},{b,c,e}

In the above example, the method of computing reducts was illustrated. As the information system can be a relational database, sometimes there may be a necessity of combining the reducts from two or more clients to get global reduct. For example, about the decision on some important issues, if they are sent to several referees from the server and if we receive the set of all reducts from each referee based on his decision, it is necessary to find a simpler technique of getting the global possible reducts.

VII. INTEGRATED REDUCT FROM THE REDUCTS FROM CLIENTS

The following algorithm gives the procedure of computing the global reduct of the information system T=(U,A,C,D) which has n clients say $T_1,T_2,\ldots,T_n$ with the decision on the issues $E_1,E_2,\ldots,E_n$ respectively. Hence, each client itself can be considered as an information system which can be given as $T_i=(U,(A\text{-}D)\cup\{E_i), C,\{E_i\}\}$.

A. Algorithm

Step 1: Construct discernibility matrix for each client T_i

Step 2: Obtain the Discernibility relation for each T_i

Step 3: Compute reduct of each T_i using Reduct Generation Algorithm

Step 4: Consider a set A=ϕ

Step 5: Use absorption Law in between the reducts of different clients;[for example, if {a,b} and {a,b,c} belong to different client, then consider the absorption {a,b}∨{a,b,c}={a,b,c}] and include the output in A.

Step 6: Repeat step 5 until step 5 cannot be applied further.

Step 7: Include all reducts obtained in step 3 in A and use absorption law in A [for example, if {a,b} and {a,b,c } are in A, then consider the absorption {a,b}∨{a,b,c}={a,b,c}]

Step 8: Write the Integrated reduct.

In the above algorithm, it can be seen that the computation of Integrated reduct is straight if the reducts from all the clients are known. The above algorithm is illustrated by the following example.

Example: Consider the decision table of the first client given below with C={a,b,c,d, e,f} and D={E_1}.

	a	b	c	d	e	f	E_1
x_1	1	0	1	1	1	1	1
x_2	0	1	0	1	1	0	1
x_3	1	1	1	0	1	1	0
x_4	0	1	0	0	1	1	0

The discernibility matrix is given by

	x_1	x_2	x_3
x_2	---		
x_3	b,d	a,c,d,f	
x_4	a,b,c,d	d,f	----

Here, by reduct generation algorithm, the reducts obtained are {d} and {b,f}

Now, consider the decision table of the second client given below with C={a,b,c,d,e,f} and D={E_2}.

	a	b	c	d	e	f	E_1
x_1	1	0	1	1	1	1	0
x_2	0	1	0	1	1	0	1
x_3	1	1	1	0	1	1	1
x_4	0	1	0	0	1	1	0

The discernibility matrix is given by

	x_1	x_2	x_3
x_2	a,b,c,f		
x_3	b,d	-----	
x_4	----	d,f	a,c

Here, by reduct generation algorithm, the reducts obtained are {a,d}, {a,c},{a,b,f} and {b,c,f}

Hence, by the algorithm 7.A, the Integrated reduct of the server is given by {{d},{b,f}}∨{{a,d}, {a,c},{a,b,f}, {b,c,f}}={{a,d}, {a,c},{a,b,f}, {b,c,f}}. It can be verified by constructing the discernibility matrix for the server by using all decisions.

Consider the Integrated decision table of the server given below with C={a,b,c,d, e,f} and D={E_1}.

	a	b	c	d	e	f	E_1	E_2
x_1	1	0	1	1	1	1	1	0
x_2	0	1	0	1	1	0	1	1
x_3	1	1	1	0	1	1	0	1
x_4	0	1	0	0	1	1	0	0

Here, the discernibility matrix is given by

	x_1	x_2	x_3
x_2	a,b,c,f		
x_3	b,d	a,c,d,f	
x_4	a,b,c,d	b,f	a,c

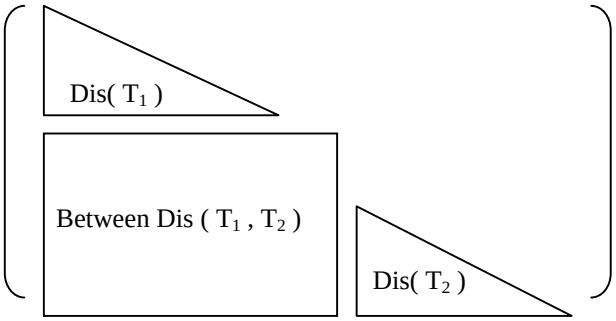
Here, by reduct generation algorithm, the reducts obtained are {a,d}, {a,c},{a,b,f}, {b,c,f}. Thus the algorithm is verified with an example.

From the above example, it can be known that the Integrated reduct of the server is the OR form of all client reducts.

Now, we discuss the process of computing reducts in any decision table of huge size. If the decision table is huge in size with respect to number of records, it may be difficult to form the discernibility matrix and apply reduct generation algorithm. So, it is necessary to introduce a tool to come across such situations. In the forthcoming section, we introduce an algorithm, which makes the job simpler, by dividing the system into two subsystems.

VIII. Reduct from the Decision Table with huge Data

In this section, we consider any information system, which consists of more records. To overcome this case, the usual algorithm is to be extended. Consider the information system T=(U,A,C,D) with n records say $x_1,x_2,\ldots,x_n$. Now, divide T into T_1 and T_2 with respect to records. Let T_1 contains the records $x_1,x_2,\ldots,x_j$ and T_2 contains the records $x_{j+1},x_{j+2},\ldots,x_n$. Then the discernibility matrix of T can be written as



where Dis(T_1) and Dis(T_2) represent the discernibility matrices of T_1 and T_2 respectively, and Between Dis (T_1 , T_2) represents the between discernibility matrix obtained by considering the imparity between the elements of T_1 and T_2 . The reduct which is found in Between Dis (T_1 , T_2) is called as the between reduct and is denoted by Bet_Red(T_1 , T_2).

The integrated reduct of this case, can be obtained using the following algorithm.

A. Algorithm

Step 1: Construct discernibility matrix for T_i , T_2 and between matrix of T_i , T_2

Step 2: Obtain the Discernibility relation for each of the three

Step 3: Compute reduct of each of the three using Reduct Generation Algorithm and denote them as Red(T_i),Red (T_2) and Bet_Red(T_1 , T_2) respectively.

Step 4: Combine Red(T_1) and Red(T_2) by using all possible unions between the elements from different sets, say Red($T_1 \cup T_2$) and apply absorption law in it [for example, if {a,b} and {a,b,c}are in Red($T_1 \cup T_2$) , then consider the absorption $\{a,b\}\wedge\{a,b,c\}=\{a,b\}$].

Step 5: Combine Red($T_1 \cup T_2$) and Bet_Red(T_1 , T_2) by using all possible unions between the elements from different sets and and apply absorption law in it [for example, if {a,b} and {a,b,c} exist, then consider the absorption $\{a,b\}\wedge\{a,b,c\}=\{a,b\}$].

.

Step 6: Write the integrated reduct obtained in step 5.

The above algorithm can be illustrated by the following example.

Example: Consider the discernibility matrix of the records $\{x_1,x_2,\ldots, x_8\}$ with the attributes {a,b,c,d,e,f}

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_2	a,b						
x_3	b,c,d	a,b					
x_4	---	a,b,d	a,c				
x_5	b,d,e	a,e	a,b,e	a,f			
x_6	a,e,f	a,f	---	---	a,e		
x_7	a,e,f	a,f	a,e	a,e,f	---	a,d,f	
x_8	---	---	a,f	a,e	a,d,e	---	a,d,f

If the system is subdivided into two namely $T_1=\{x_1,x_2,x_3,x_4\}$ and $T_2=\{x_5,x_6,x_7,x_8\}$, the Red(T_1)={{a,b},{b,c},{a,c}}; Red(T_2)={{a},{e,f}}and Bet_Red(T_1 , T_2)={{a,b},{a,d},{a,e},{e,f}}. Hence, Red($T_1 \cup T_2$)={{a,b},{a,c},{b,c,e,f}}. The integrated reduct is given by {{a,b},{b,c,e,f},{a,c,d},{a,c,e}}

This algorithm can be further developed for huge databases by using iterative process of the same procedure.

IX. Conclusion

In this paper, we gave an algorithm for obtaining the global reduct from the reducts obtained from different clients. Further, it real problems, when we process with huge data it is difficult to compute the reduct for the entire system. So, an algorithm is proposed to compute reducts by subdividing the discernibility matrix into three. However, this algorithm is limited to size, because, after dividing the discernibility matrix into three, if the sub matrices have huge data, again it is necessary to apply this algorithm for each of the sub matrices.

X. Reference

[1] W. Buszkowski and E. Orlowska, "*On the logic of database dependencies*", Bull. Polish Sci. Math., Vol 34, pp345-354,1986.

[2] G.Ganesan, D.Latha, Raghavendra Rao C, '*Proper Rough Sets*', Proceedings of National Conference on Intelligent Optimization, Allied Publishers, India , 2006 [in print]

[3] Kohavi R. "*Useful feature subsets and Rough set reducts*", Proceedings, Third International Workshop on Rough Set and Soft Computing, 310-317, 1994
[4] Ning Zhong, Juzhen Dong, Setsuo Ohsuga, "*Using Rough Sets with Heuristics for Feature Selection*", Journal of Intelligent Information systems, Vol 16, 199-214, 2001,
[5] Starzyk J, Nelson D.E., Sturtz K, '*Reduct Generation in Information Systems*', Bulletin of International Rough Set Society, 3 (1/2), 1999
[6] A. Skowron and J. Stepaniuk, "*Towards an approximation theory of discrete problems: Part I*", Fundamenta Informaticae 15(2), pp.187-208, 1991.
[7] A. Skowron and C. Rauszer, "*The discernibility matrices and functions in information systems*", Fundamenta Informaticae 15(2), pp.331-362, 1991.
[8] Zdzislaw Pawlak, "*Information systems - theoretical foundations*", Information Systems, Vol. 6, pp.205-218, 1981.
[9] Zdzislaw Pawlak, "*Rough sets*", International Journal of Computer and Information Sciences, 11, 341-356, 1982.
[10] Zdzislaw Pawlak, "*On rough dependency of attributes in information systems*", Bull. Polish Acad. Sci. Tech. Vol. 33, pp.481-485, 1985.
[11] Zdzislaw Pawlak, "*Rough Sets-Theoretical Aspects and Reasoning about Data*", Kluwer Academic Publications, 1991