A Novel Attribute Reduction Algorithm of Decomposition Based on Rough Sets

Na Jiao
Dept. of Computer Science and Tech.
Tongji University
Shanghai 201804, P.R.China
zdx.jn@163.com

Duoqian Miao
Dept. of Computer Science and Tech.
Tongji University
Shanghai 201804, P.R.China
miaoduoqian@163.com

Hongyun Zhang
Dept. of Computer Science and Tech.
Tongji University
Shanghai 201804, P.R.China
zhangjiaye1972@hotmail.com

***Abstract*—Attribute reduction is a key task for the research of rough sets. However, when dealing with large-scale data, many existing proposals based on rough set theory get worse performance. In this paper, we propose a novel attribute reduction algorithm of decomposition based on rough sets. The idea of decomposition is to break down a complex table into a super-table and several sub-tables that are simpler, more manageable and solvable by using existing induction methods, then joining them together in order to solve the original table. Compared with the traditional methods, experiments with some standard datasets from UCI database are done and experimental results illustrate that the algorithm of this paper improve computational efficiency.**

***Keywords*-rough set theory; attribute reduction; decomposition; super-table; sub-table;**

I. INTRODUCTION

Pawlak has proposed rough sets [1-4] which is a valid mathematical tool to deal with imprecise, uncertain, and vague information. It has been widely applied in many fields such as machine learning [5], data mining [6], and pattern recognition [1], etc. Attribute reduction is the fundamental problem in rough sets. Many researchers propose various attribute reduction approaches [6-9]. These approaches are generally divided into three categories which are methods based on discernibility matrix [7], methods based on positive region [7] and methods based on information entropy [8]. All attribute reduction methods are available for a smaller table. However, the database can be quite large in the information age. We might gain worse performance even get no result when handling large-scale data with traditional attribute reduction methods based on rough sets.

The main motivation of this study is to design a method that can handle massive and complicated real-world problems, we present a decomposition method. The idea of decomposition [10,11] is to break down a large and complex task into several simpler and more manageable sub-tasks that are solvable by using existing induction methods, then joining their solutions together in order to solve the original problem. There are a few works in data mining using decomposition methodology such as decomposition of incomplete information systems [12,13], decomposition in multi-agent systems [14], etc.. However, some decomposition methods may result in the loss of information or distortion of original data and knowledge, and can even render the original data mining system un-minable.

To avoid these decomposition shortcomings in data mining, we should select the appropriate decomposition method. Jiawei Han introduces multirelational data mining [15] using keys to link, furthermore, there are the same expression in database software. There are not any the loss of information or distortion of original data and knowledge when we convert a single table into multirelational tables. Therefore we break down a large-scale decision table into a super-table and several sub-tables. The super-table is composed of a set of decision attributes and several joint attributes which are the key words in sub-tables. The sub-table is made up of random subset in condition attributes. Then we join their solutions together in order to solve the original table. We do numerous experiments comparing with the classical methods using some standard datasets from UCI database. Experimental results show that the algorithm of this paper improve computational efficiency especially to huge database.

II. BASIC NOTIONS

For the convenience of description, some basic definitions and properties are introduced here at first.

A. Basic definitions

We assume that attribute reduction discussed in this paper is performed in a decision table.

Definition 1. A decision table is defined as $T = < U, C \cup D, V, f >$, where U is a non-empty finite set of objects, called universe; C is a set of all condition attributes and D is a set of decision attributes; $V = \bigcup_{a \in C \cup D} V_a$,$V_a$ is a set of attribute values of attribute a; and $f : U \times (C \cup D) \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in C \cup D$.

Definition 2. Given a decision table $T = < U, C \cup D, V, f >$, for each subset $B \subseteq C \cup D$, induces an equivalence (indiscernibility) relation on U as shown: $IND(B) = \{(x, y) \in U \times U | \forall b \in B, b(x) = b(y)\}$. The family of all equivalence classes of $IND(B)$, i.e., the partition induced by B, is denoted as $U/IND(B) = \{[x_i]_B : x_i \in U\}$, where $[x_i]_B$ is the equivalence class containing x_i. All the elements

978-0-7695-3735-1/09 $25.00 © 2009 IEEE
DOI 10.1109/FSKD.2009.97

in $[x_i]_B$ are equivalent (indiscernible) with respect to B. Equivalence classes are elementary sets in rough sets.

Definition 3. Given a decision table $T =< U, C \cup D, V, f >$, for $X \subseteq U$ and $B \subseteq C$, the lower and upper approximations of X with respect to B, denoted by $\underline{B}X$ and $\overline{B}X$, respectively, are defined as: $\underline{B}X = \cup \{[x_i]_B \mid [x_i]_B \subseteq X\}$, $\overline{B}X = \cup \{[x_i]_B \mid [x_i]_B \cap X \neq \emptyset\}$.

Definition 4. Given a decision table $T =< U, C \cup D, V, f >$, for $P \subseteq C$, the positive region of the partition $U/IND(D)$ with respect to P is defined as: $POS_P(D) = \cup_{X \in U/IND(D)} \underline{P}X$.

we break down a decision table into a super-table and several sub-tables. The super-table consists of a set of decision attributes and several joint attributes which are the keywords in sub-tables. The sub-table is composed of random subset in condition attributes.

Definition 5. Given a decision table $T =< U, C \cup D, V, f >$.

- A sub-table is defined as $T^{B_i} =< U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i} >$, where U is a non-empty unique finite set of objects, called universe; $B_i \subseteq C, i = 1, 2, \cdots, m$, $C = \cup_{i=1}^{m} B_i$ and $B_i \cap B_j = \emptyset, i \neq j$. b_i is a joint attribute which join the sub-table to the super-table and it is a keyword in T^{B_i}, $V_{b_i}^{B_i} = b_i^k, k = 1, 2, \cdots, p$; $V^{B_i} = \bigcup_{a \in B_i} V_a^{B_i}$, $V_a^{B_i}$ is a set of attribute values of attribute a; and $f^{B_i} : U^{B_i} \times B_i \to V^{B_i}$ is an information function such that $f^{B_i}(x, a) \in V_a^{B_i}$ for every $x \in U^{B_i}$, $a \in B_i$.
- A super-table is defined as $T^S =< U, S \cup D, V^S, f^S >$, where U is a non-empty finite set of objects, called universe; $S = \cup_{i=1}^{m} \{b_i\}$ is a set of all joint attributes and D is a set of decision attributes; $V^S = \bigcup_{a \in S \cup D} V_a^S$, V_a^S is a set of attribute values of attribute a; and $f^S : U \times (S \cup D) \to V^S$ is an information function such that $f^S(x, a) \in V_a^S$ for every $x \in U$, $a \in S \cup D$.
- A mid-table is defined as $T^{M_i} =< U, M_i \cup D, V^{M_i}, f^{M_i} >$, $i = 1, 2, \cdots, m$, where U is a non-empty finite set of objects, called universe; $M_i = (S \backslash \{b_i\}) \cup B_i$ and D a set of decision attributes; $V^{M_i} = \bigcup_{a \in M_i \cup D} V_a^{M_i}$, $V_a^{M_i}$ is a set of attribute values of attribute a; and $f^{M_i} : U \times (M_i \cup D) \to V^{M_i}$ is an information function such that $f^{M_i}(x, a) \in V_a^{M_i}$ for every $x \in U$, $a \in M_i \cup D$.

Example 1. Table 1 is a decision table, we decompose it into one super-table (Table 2) and two sub-tables (Table 3, Table 4). Combine Table 2 and Table 3 to compose a mid-table Table 5. Similarly, Table 6 comes from Table 2 and Table 4.

B. Basic properties

The following are some properties according to the above definitions. Assume a decision system $T =< U, C \cup D, V, f >$, sub-tables $T^{B_i} =< U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i} >$, $i = 1, 2, \cdots, m$, a super-table $T^S =< U, S \cup D, V^S, f^S >$,

Table I
A ORIGINAL DECISION TABLE

U	a_1	a_2	a_3	a_4	d
1	1	1	1	0	1
2	1	0	1	1	1
3	0	0	0	1	0
4	1	0	1	0	1

Table II
A SUPER-TABLE

U	b_1	b_2	d
1	b_1^1	b_2^1	1
2	b_1^2	b_2^2	1
3	b_1^3	b_2^3	0
4	b_1^2	b_2^1	1

Table III
THE FIRST SUB-TABLE

b_1	a_1	a_2
b_1^1	1	1
b_1^2	1	0
b_1^3	0	0

Table IV
THE SECOND SUB-TABLE

b_2	a_3	a_4
b_2^1	1	0
b_2^2	1	1
b_2^3	0	1

Table V
THE FIRST MID-TABLE

U	a_1	a_2	b_2	d
1	1	1	b_2^1	1
2	1	0	b_2^2	1
3	0	0	b_2^3	0
4	1	0	b_2^1	1

Table VI
THE SECOND MID-TABLE

U	b_1	a_3	a_4	d
1	b_1^1	1	0	1
2	b_1^2	1	1	1
3	b_1^3	0	1	0
4	b_1^2	1	0	1

mid-tables $T^{M_i} =< U, M_i \cup D, V^{M_i}, f^{M_i} >$, $i = 1, 2, \cdots, m$. Some properties are described as follows.

Property 1. The positive region in the super-table T^S is equivalent to the positive region in the original decision table T, that is $POS_{(S)}(D) = POS_{(C)}(D)$.

Corollary 1. The positive region in the mid-table T^{M_i} is equivalent to the positive region in the original decision table T, that is $POS_{(M_i)}(D) = POS_{(C)}(D)$.

Property 2. The joint attribute b_i in the super-table T^S is dispensable, that is $POS_{(S \backslash \{b_i\})}(D) = POS_{(S)}(D)$, iff the condition attribute set B_i in the original decision table T corresponding to the joint attribute b_i is dispensable, that is $POS_{(C \backslash B_i)}(D) = POS_{(C)}(D)$.

Corollary 2. The condition attribute a in the mid-table T^{M_i} is dispensable, that is $\exists a \in B_i, POS_{(M_i \backslash \{a\})}(D) = POS_{(M_i)}(D)$, iff the condition attribute a in the original decision table T is dispensable, that is $POS_{(C \backslash \{a\})}(D) = POS_{(C)}(D)$.

Corollary 3. If the joint attribute b_i in the super-table T^S is indispensable, that is $POS_{(S \backslash \{b_i\})}(D) \neq POS_{(S)}(D)$, then there is a subset A in the condition attribute set B_i corresponding to the joint attribute b_i is indispensable in the original decision table T, that is $A \subseteq B_i$, $POS_{(C \backslash A)}(D) \neq POS_{(C)}(D)$.

Corollary 4. If the condition attribute a in the mid-table T^{M_i} is indispensable, that is $POS_{(M_i \backslash \{a\})}(D) \neq POS_{(M_i)}(D)$, then the condition attribute a in the original decision table T is indispensable, that is $POS_{(C \backslash \{a\})}(D) \neq POS_{(C)}(D)$.

III. THE ATTRIBUTE REDUCTION ALGORITHM OF DECOMPOSITION BASED ON ROUGH SETS

In this section we introduce the strategy of decomposition at first. Then the attribute reduction algorithm of decomposition based on rough sets is proposed.

A. The strategy of decomposition

In this paper, our strategy of decomposition is to decompose the condition attributes into several subsets and connect these subsets with some new attributes called joint attributes. The joint attributes and the set of decision attributes form the super-table. Every subset and the joint attribute construct the sub-table. In the process of algorithm, we combine a sub-table with the super-table to compose the mid-table. Actually, we only combine condition attributes and can't change data table itself. So our methods can't lead to loss of data or incorrect information after decomposition. In this paper, the condition attributes of the original decision table are divided equally among sub-tables randomly.

B. The attribute reduction algorithm of decomposition based on rough sets

Suppose that the number of sub-tables is k. First, we break a original decision table down into one super-table and k sub-tables. The condition attributes of the original decision table are divided equally among k sub-tables. The joint attribute and subset in condition attributes compose a sub-table. The super-table is made up of a set of decision attributes and k joint attributes that are the key words in sub-tables.

Then if the joint attribute in super-table is dispensable, we can delete the joint attribute in super-table and combine the same objects. Judge the next joint attribute. Otherwise, we combine a sub-table with the super-table to compose a mid-table, if the condition attribute in mid-table is dispensable, we can delete the condition attribute in mid-table and combine the same objects, or else continue the next loop. Finally, a reduction can be find.

We show the attribute reduction algorithm of decomposition based on rough sets in Algorithm 1.

Algorithm 1. Attribute reduction algorithm of decomposition based on rough sets (ARD)

Input: A decision table $T =< U, C \cup D, V, f >$; The number of sub-tables is k.

Output: Attribute reduction $RED_{(D)}(C)$.

Break T down into one super-table $T^S =< U, S \cup D, V^S, f^S >$ and sub-tables $T^{B_i} =< U^{B_i}, B_i \cup \{b_i\}, V^{B_i}, f^{B_i} >$, $i = 1, 2, \cdots, k$.
$RED_{(D)}(C) \leftarrow C$, $i \leftarrow 1$.
While $i <= k$ **do**
Begin
  If $POS_{(S \setminus \{b_i\})}(D) = POS_{(S)}(D)$, **then**
  Begin
    $S \leftarrow S - \{b_i\}$;
    $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - B_i$;
    Combine the same objects;
  End
  Else
  Begin
    Combine T^{M_i} with T^{B_i} and T^S, $j \leftarrow card(B_i)$;
    While $j > 0$ **do**
    Begin
      If $\exists a \in B_i, POS_{(M_i \setminus \{a\})}(D) = POS_{(M_i)}(D)$, **then**
        Begin
          $B_i \leftarrow B_i - \{a\}$;
          $RED_{(D)}(C) \leftarrow RED_{(D)}(C) - \{a\}$;
          Combine the same objects;
        End
        $j \leftarrow j - 1$;
    End
  End
  $i \leftarrow i + 1$
End
Return $RED_{(D)}(C)$.

Actually, the condition attributes of a decision table are divided into several parts. We process every part instead of every condition attribute. Every part is substituted by a joint attribute. In other words, $|C|$ condition attributes of a decision table are compressed to k joint attributes of the super-table. If the joint attribute is dispensable, the condition attribute set corresponding to the joint attribute is dispensable and can be deleted once. Each attribute in this condition attribute set needn't to be checked again. Even though the joint attribute is indispensable, we need to convert a super-table and a sub-table into a mid-table, the scale of mid-table is compressed a lot. Relative to $|C|$, the number of attributes of the mid-table is $(|C|/k) + k$, which is very small. Hence our methods achieved significant saving on computation time.

IV. EXPERIMENTS

In this section, we show that our attribute reduction algorithm of decomposition based on rough sets can reduce the computation significantly. The following experiments were conducted on a PC in our laboratory and its configuration is Dual E2140 1.60G (CPU), 1G (memory) and windows XP (operation system). All algorithms were implemented in C# and the programming used the stored procedure of database SQL Server 2000.

A. A comparative experiment on seven datasets

In order to test the validity of the algorithm, four classical algorithms for attribute reduction are described as follows: General attribute reduction algorithm (General); Attribute reduction algorithm based on positive region (Positive) (computing core firstly and appending the most important attribute according to significance of attributes until achieving reduction); Attribute reduction algorithm based on information entropy (Entropy); Attribute reduction algorithm based on discernibility matrix (Matrix). According to ARD algorithm of this paper, we suppose the number of sub-tables is four. We perform the experiments on publicly

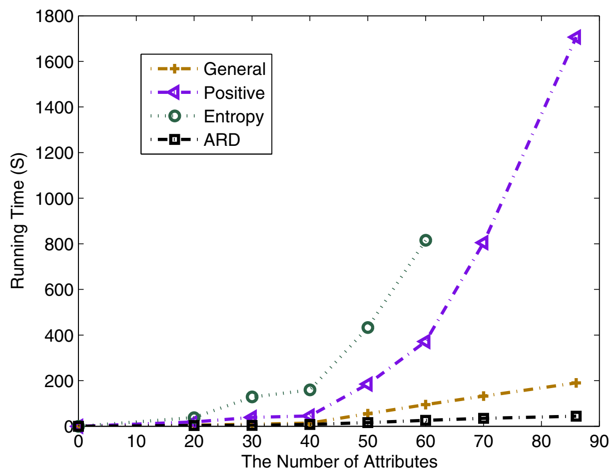

Figure 1. The comparison of performances of different attributes on Insurance-Company-Benchmark dataset

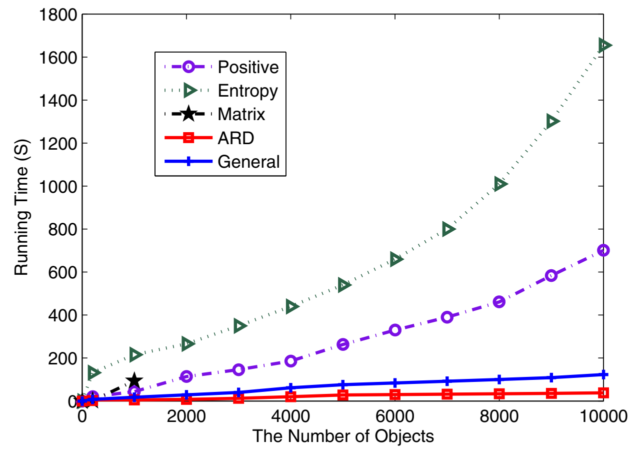

Figure 2. The comparison of performances of different objects on Connect-4 dataset

available datasets from UCI database (These datasets can be downloaded at http://www.ics.uci.edu). The experiment results are shown in Table 7. A brief description is below the datasets. (C: condition attribute; D: decision attribute; O: object). The rest columns present running time of different algorithms (its unit is second and the abbreviation is S) which is average of repeating 10 times experiments. ∞ means that the running time is more than 43200 seconds (12 hours).

When there are missing values in datasets, these values are filled with mean values for continuous attributes and majority values for nominal attributes [16]. If the datasets are numerical, all continuous attributes are discretized using Equal Frequency per Interval [17].

As listed in Tables 7, General outperforms other three classic attribute reduction methods. The performance of Positive is less than that of General. The performance of Entropy is worst. Matrix is less time consuming for small dataset while this algorithm gain worse performance even get no result for large-scale dataset. ARD has been shown to be superior to other methods.

B. A experiment on Insurance-Company-Benchmark dataset with different attributes

The second experiment is performed on Insurance-Company-Benchmark dataset which has 86 attributes and 9822 objects. We select bottom 20, 30, 40, 50, 60, 70 and 86 attributes from this dataset respectively. We use four classical approaches and the attribute reduction algorithm given in this paper to test these data. According to our method we break the datasets down into one super-table and four sub-tables.

From Figure 1, we can see the comparison of efficiencies of various methods as attributes are increased gradually. As depicted in Figure 1, ARD outperforms other methods. The running time of our method increases slightly as attributes are increased gradually. However, other methods consume much time. There is not the execution time of Matrix because it is always ∞. The time of Entropy is ∞ when the number of attributes is greater than 70.

C. A experiment on Connect-4 dataset with different objects

We do another experiment on Connect-4 dataset which has 43 attributes and 67557 objects. We select top 200, 2000, 4000, 6000, 8000 and 10000 objects from this dataset. Four classic approaches and our method are used to test these data. The number of sub-tables is the same as the above experiments.

Figure 2 shows the comparison of efficiencies of various algorithms based on different size of objects. As depicted in Figure 2, ARD can achieve better performance than other methods. The execution time of Matrix is ∞ when the number of objects is more than 2000.

V. Conclusion

Attribute reduction is an important issue for rough sets. Many proposed attribute reduction methods get worse performance for large datasets. In this paper, we introduce a novel decomposition method for rough set attribute reduction. We break down a complex table into smaller, less complex and more manageable sub-tables that are solvable by using existing methods, then joining them together to solve the initial table. In order to test the validity of the algorithm, we do numerous experiments. Experimental results demonstrate that our method is efficient for various datasets.

There are two directions for ongoing work. The first one is to develop other efficient algorithms based on decomposition for attribute reduction. The second one is to focus on how to make sure the number of sub-tables exactly.

Table VII
COMPARISON OF EFFICIENCIES OF DIFFERENT ATTRIBUTE REDUCTION ALGORITHMS

Dataset	General	Positive	Entropy	Matrix	ARD
Audiology(Standardized)(69C,1D,226O)	12S	44S	286S	14S	9S
Breast Cancer Wisconsin(Diagnostic)(31C,1D,569O)	3S	6S	45S	200S	2S
Connect-4(42C,1D,67557O)	287S	1651S	∞	∞	111S
Insurance Company Benchmark(COIL 2000)(85C,1D,9822O)	190S	1706S	∞	∞	44S
Madelon(499C,1D,4400O)	∞	∞	∞	∞	41S
Optical Recognition of Handwritten Digits(64C,1D,1796O)	15S	44S	∞	∞	4S
SPECT Heart(44C,1D,267O)	6S	15S	109S	6S	1S

ACKNOWLEDGMENT

This paper is supported by The National Natural Science Foundation of P.R.China (no. 60475019, 60775036) and The Research Fund for the Doctoral Program of Higher Education (no. 20060247039).

REFERENCES

[1] Z. Pawlak, Rough sets, Int. J. Inf. Comput. Sci. 11 (5) (1982) 341-356.

[2] S. Greco, M. Inuiguchi, R. Slowinski, Fuzzy rough sets and multiple-premise gradual decision rules, International Journal of Approximate Reasoning 41(2) (2006) 179-211.

[3] M. Yang, P. Yang, A novel condensing tree structure for rough set feature selection, Neurocomputing 71 (2008) 1092-1100.

[4] Q. H. Hu, J. F. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowledge-Based Systems 21 (2008) 294-304.

[5] R. W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (2003) 833-849.

[6] H. Liu, H. Motoda, Attribute Selection for Knowledge Discovery and Data Mining, Kluwer, Boston, 1998.

[7] G. Y. Wang, Rough Set Theory and Knowledge Acquisition, Xi'an Jiaotong University Press, Xi'an, 2001 (in Chinese).

[8] D. Q. Miao, J. Wang, Information-based algorithm for reduction of knowledge, in: IEEE International Conference on Intelligent Processing Systems, 1997, pp. 1155-1158.

[9] I. S. Oh, J. S. Lee, B. R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1424-1437.

[10] L. Rokach, Decomposition methodology for classification tasks: a meta decomposer framework, Pattern Analysis and Applications 9 (2006) 257-271.

[11] O. Maimon, L. Rokach, Decomposition methodology for knowledge discovery and data mining: theory and applications, World Scientific, 2005.

[12] J. G. Bazan, R. L. atkowski, M. Szczuka, Missing template decomposition method and its implementation in rough set exploration system. in: Proceedings of the Fifth International Conference on Rough Sets and Current Trends in Computing, Kobe, Japan, 2006, pp. 254-263.

[13] Q. Z. Zhang, An Approach to Rough Set Decomposition of Incomplete Information Systems, in: IEEE Conference on Industrial Electronics and Applications, Harbin, China, 2007, pp. 2455-2460.

[14] H. S. Nguyen, S. H. Nguyen, A. Skowron, Decomposition of task specification problems, in: Proceedings of the 11th International Symposium on Foundations of Intelligent Systems, Warsaw, Poland, 1999, pp. 310 - 318.

[15] J. W. Han, M. Kamber, Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann, 2006.

[16] J. W. Grzymala-Busse, W. J. Grzymala-Busse, Handling missing attribute values. In Handbook of Data Mining and Knowledge Discovery, ed. by O. Maimon and L. Rokach, 2005, 37-57.

[17] J. W. Grzymala-Busse, Discretization of numerical attributes, In Handbook of Data Mining and Knowledge Discovery, ed. by W. Kl?sgen and J. Zytkow, Oxford University Press, 2002, 218-225.