Research on Complete Algorithms for Minimal Attribute Reduction

Jie Zhou, Duoqian Miao, Qinrong Feng, and Lijun Sun

Department of Computer Science and Technology, Tongji University
Shanghai, P.R. China, 201804
{jie_jpu,miaoduoqian,fengqr72}@163.com, sunlj1028@yahoo.com.cn

Abstract. Minimal attribute reduction plays an important role in both theory and practice, but it has been proved that finding a minimal reduct of a given decision table is a NP-hard problem. Some scholars have also pointed out that current heuristic algorithms are incomplete for minimal attribute reduction. Based on the decomposition principles of a discernibility function, a complete algorithm *CAMARDF* for finding a minimal reduct is put forward in this paper. Since it depends on logical reasoning, it can be applied for all decision tables after their discernibility functions constructed reasonably. The efficiency of *CAMARDF* is illustrated by experiments with UCI data sets further.

Keywords: decision table, discernibility function, minimal reduct, complete algorithm.

1 Introduction

Rough set theory[1] is a new mathematic tool aimed at data analysis problems involving uncertain or imprecise information. Attribute reduction is one of the most fundamental and important notions in rough set theory. A reduct is a minimal subset of attributes that preserving the same information that is considered as provided by the entire set of attributes[2]. The conciseness, understandability, generality and preciseness of the decision rule sets, which are derived from reducts directly, will be distinct, so we want to find the minimal reducts, i.e., the shortest reducts, such that attributes can be removed as much as possible. Then the storage space for the decision table can be compressed effectively and the properties of the decision rule set will be excellent. Unfortunately, searching for a minimal reduct has been proved to be a NP-hard problem[3].

Some heuristic algorithms have been constructed to get the optimal or approximate reducts. Generally, these heuristic algorithms can be divided into three categories based on fitness functions (or called as heuristic information): algorithms that are based on positive region[4,5], information entropy[6,7] and discernibility matrix[3,8] respectively. Though heuristic algorithms are effective, Miao et al[9] have illustrated that all of these heuristic algorithms are incomplete for finding minimal reducts, in other words, minimal reducts can not be always attained by these heuristic algorithms when decision tables are given and even the result is just a superset of a reduct sometimes[10].

Some properties about discernibility function of a decision table are analyzed critically in this paper. Due to space restrictions, the proofs of theorems are omitted. According to these properties, some search strategies can be added to attribute reduction. A complete algorithm *CAMARDF* for minimal attribute reduction is put forward based on the decomposition principles of a discernibility function. The experiments show that a minimal reduct can be found with *CAMARDF* effectively for UCI data sets after their discernibility functions constructed well.

2 Some Properties about Discernibility Function

A prime implicant of a boolean function is an implicant that can not be covered by a more general implicant. Skowron[3] has proved that all reducts are in one-to-one correspondence with the prime implicants of the discernibility function in a given decision table. The problem of finding minimal reducts is polynomially equivalent to the problem of searching for prime implicants of the discernibility function with the shortest length. A prime implicant with the shortest length means that the number of its variables is minimal.

Some detailed descriptions of rough set theory can be found in [1,3].

Definition 1[3]**.** Decision table $DT = (U, C \cup D, V, \rho)$, $U = \{o_1, o_2, \cdots, o_n\}$, the discernibility matrix can be defined as a $n \times n$ matrix $DM(DT) = (c_{ij})_{n\times n}$, where the element c_{ij} satisfies:

$$c_{ij} = \begin{cases} \{a|a \in C \wedge \rho(o_i, a) \neq \rho(o_j, a)\} & \Omega \\ \emptyset & others \end{cases} \tag{1}$$

where, Ω means $1 \leq j < i \leq n, \rho(o_i, D) \neq \rho(o_j, D)$ and at least one object between o_i and o_j is consistent.

Definition 2[3]**.** Given a decision table $DT = (U, C \cup D, V, \rho)$, the discernibility function of DT is a boolean function that each boolean variable a is identified with attribute $a \in C$ and is defined as follows:

$$DF(DT) = \wedge\{\vee c_{ij} : 1 \leq j < i \leq n, c_{ij} \neq \emptyset\} \tag{2}$$

where $c_{ij} \in DM(DT)$, $\vee c_{ij} = \vee a(a \in c_{ij})$ is the disjunction of all variables a such that $a \in c_{ij}$. Absorption law is often adopted to reduce the discernibility function and the reduced discernibility function is also a conjunctive normal form obviously.

In the sequel, we all use reduced discernibility function to discuss. Suppose a reduced discernibility function $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, we consider $DF = \{f_1, f_2, \cdots, f_s\}$ instead and if $f_i = a_1 \vee a_2 \vee \cdots \vee a_{k_i}$, we consider $f_i = \{a_1, a_2, \cdots, a_{k_i}\}$ instead when no confusion can arise. The set of all variables of DF is denoted as ψ_{DF}.

Theorem 1. Given a decision table $DT = (U, C \cup D, V, \rho)$, its discernibility function is $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$. $B \subseteq C$ is a reduct of DT, then $\forall f_i \in DF (i = 1, 2, \cdots, s)$, $B \cap f_i \neq \emptyset$.

The theorem 1 indicates that a reduct of a decision table must have some common elements with each clause of the discernibility function.

Theorem 2. Given a decision table $DT = (U, C \cup D, V, \rho)$, its discernibility function is $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, $\forall a \in \psi_{DF}$, the set of the shortest implicants of DF that include variable a are denoted as $I(a)$, the set of minimal reducts of DT is denoted as $MR(DT)$, then $MR(DT)$ satisfies:

$$MR(DT) = \cup\{I(a) | a \in \psi_{DF}, for\ \forall b \in \psi_{DF} - \{a\}, \xi \in I(a), \xi' \in I(b), such\ that\ |\xi| \leq |\xi'|\} \tag{3}$$

where $|\xi|$ denotes the length of ξ. The right side in Eq. (3) indicates that the length of each element in $I(a)$ is the shortest compared with other variables in DF. Since prime implicants are special cases of implicants, so the implicants with the shortest length are also the prime implicants with the shortest length. That is to say, if we find an implicant with shortest length, then we find a minimal reduct. So it is no need to restrict search space only for prime implicants.

Why we focus on the implicants, not the prime implicants? The reason is that $I(a)$ is not always the set of prime implicants that include a. For example, a discernibility function is given as $DF = (a \vee c) \wedge (a \vee b \vee d) \wedge (a \vee b \vee h) \wedge (c \vee g) \wedge (c \vee f) \wedge (b \vee e)$. Where $a \wedge b \wedge c \in I(a)$, but it is not a prime implicant of DF(due to the prime implicant $b \wedge c$, so it is not a reduct. It is no need to cost much to check whether the elements in $I(a)$ are prime implicants.

Theorem 3 (expansion law[11]). Suppose a discernibility function is $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, $\forall a \in \psi_{DF}$, DF can be decomposed with respect to a as follows:

$$DF = DF_1 \vee DF_2 \tag{4}$$

where:

$$DF_1 = \wedge\{f_i | f_i \in DF \wedge a \notin f_i\} \wedge a, \tag{5}$$

$$DF_2 = \wedge\{f_i | f_i \in DF \wedge a \notin f_i\} \wedge \{(f_i - \{a\}) | f_i \in DF \wedge a \in f_i\}. \tag{6}$$

Under expansion law, the implicants of discernibility function DF can be divided into two groups according to each variable $a \in \psi_{DF}$. One includes variable a, can be derived from DF_1. The other does not include variable a, can be derived from DF_2. So the shortest implicants including variable a can only be derived from DF_1. Further, according to theorem 1, it just need to find the shortest implicants of $\wedge\{f_i | f_i \in DF \wedge a \notin f_i\}$ in DF_1, thus theorem 2 and theorem 3 can be applied in this process repeatedly.

We call this iterative process as *decomposition principles* of the discernibility function and the variable a is called a *decomposition variable*. If all variables in ψ_{DF} are considered, we will get the shortest implicants.

Theorem 4. Suppose two conjunctive normal forms $F = f_1 \wedge f_2 \wedge \cdots \wedge f_s$ and $F' = F \wedge f$, f is a new clause that is composed of the disjunction of some variables. ξ and ξ' are the shortest prime implicant of F and F' respectively, then $|\xi'| \geq |\xi|$.

The theorem 4 shows that the length of the shortest prime implicants of conjunctive normal form will increase monotonically as the number of clauses of this conjunctive normal form increased.

Theorem 5. Given a decision table $DT = (U, C \cup D, V, \rho)$, its discernibility function is $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, $\forall a \in \psi_{DF}$ and $P_{DF}(a) = 1$, then:

(1)if $\exists f \in DF$, such that $a \in f$ and $|f| = 1$, then $I(a) = MR(DT)$;

(2)if $\exists f \in DF$, such that $a \in f$ and $|f| > 1$. For $\forall b \in (f - \{a\})$, if $P_{DF}(b) = 1$, then $\forall \xi_a \in I(a)$, $\forall \xi_b \in I(b)$ and $\forall \xi \in MR(DT)$, $|\xi_a| = |\xi_b| = |\xi|$;

(3)if $\exists f \in DF$, such that $a \in f$ and $|f| > 1$, if $\exists b \in (f - \{a\})$ and $P_{DF}(b) > 1$, then $\forall \xi_a \in I(a)$ and $\forall \xi_b \in I(b)$, $|\xi_a| \geq |\xi_b|$.

where $P_{DF}(a)$ denotes the frequency of a in discernibility function DF.

The theorem 5 shows that the search paths, which begin from the variables that their frequencies in the discernibility function are equal to one, needn't be considered in the global search process. Because these variables must be included in a minimal reduct (as (1), (2) in theorem 5) or the search paths that begin from other variables can be considered instead (as (3) in theorem 5). So superfluous search works can be avoided and the efficiency of global search can be improved.

3 Minimal Attribute Reduction Based on Discernibility Function

Heuristic algorithms are often applied to search approximate optimal results for NP-hard problems in artificial intelligence. However, the global optimization can not always be guaranteed, i.e., the search path will not always along the optimal path. The local optimum must be modified gradually for finding global optimization.

In order to find a minimal reduct of a decision table, an iterative algorithm can be constructed by applying theorem 2 and theorem 3 repeatedly. Based on theorem 2 to theorem 5, some search strategies can be added to minimal attribute reduction based on depth-first search method.

Depth search strategy 1: We choose the decomposition variable according to their significance from maximal to minimal, because choosing the attribute with higher significance will reduce search space faster. In a depth search path, if attribute a has been chosen as a decomposition variable for boolean function DF_k in the kth step, then we can only deal with $DF_{k+1} = DF_k \backslash \{a\} = DF_k - \{f_i | a \in f_i, f_i \in DF_k\}$ in the $(k+1)$th step according to decomposition principles.

Depth search strategy II: If the order of variables is constructed at the first time according to their significance and this order will not be changed in the sequel decomposition procedures, then the order is called as static variable order. On the contrary, if attribute significance are changed dynamically based on different boolean function in the sequel decomposition and the relevant order of attributes is also changed simultaneously, then the order is called as dynamic variable order. In the implementation of the algorithm, the later is applied.

Depth search strategy III: If the length of current variable sequence in a depth search path is equal to the length of candidate minimal reduct, then the current depth search will be terminated, and the path turns back to the upper layer for width search continually.

Width search strategy: Suppose $\psi_{DF} = \{a_1, a_2, \cdots, a_t\}$ and the variable order is $a_1 > a_2 > \cdots > a_t$ based on their significance. According to depth search strategy I, a_k is preferential to a_{k+1}. After the search path beginning from a_k terminated, a shortest implicant that includes a_k has been found. For the search path beginning from a_{k+1}, we can only deal with boolean function $\wedge\{f_i | f_i \in DF \wedge a_k \notin f_i\} \wedge \{(f_i - \{a_k\}) | f_i \in DF \wedge a_k \in f_i\}$ using theorem 2 and theorem 3 iteratively. If one clause is empty after removing some variables during decomposition procedures, then the algorithm turns back to the upper layer.

The complete algorithm for minimal attribute reduction based on discernibility function (*CAMARDF*) can be described as follows:

Algorithm: *CAMARDF*

```
Input: Decision table $S = (U, C \cup D, V, \rho)$;
Output: a minimal reduct of $S$.
Initialization: $Reduct.length$=0, $MinReduct.length$=$|C|$ and reduced
discernibility function $DF$ has been constructed.
  CAMARDF( $DF$ )
  {
      ComputeSIG( $a$ , $a \in \psi_{DF}$ );
      SortSIG( $sig(a)$ , $a \in \psi_{DF}$ );
      $i$=0;
      do{
          $Reduct.length$++;
          if( $Reduct.length$ = $MinReduct.length$ ){
              $Reduct.length--$;
              return;
          }//end if

          if( $i$>0 ){
              $DF = DF \backslash \{Attribute[i-1]\}$;
              if( $\exists f_i \in DF$, $f_i = \emptyset$ ){
                  $Reduct.length--$;
                  return;
              }//end if
          }//end if

          $Reduct$=$Reduct \cup Attribute[i]$;
          $DF' = DF - \{f_j | f_j \in DF \wedge Attribute[i] \in f_j\}$;
          if( $DF' = \emptyset$ ){
              if( $MinReduct.length$ > $Reduct.length$ )
                  $MinReduct$=$Reduct$
          }//end if
          else
24                  CAMARDF( $DF'$ );
25              $Reduct$=$Reduct$ - $Attribute[i]$;
26              $Reduct.length--$;
27              $i$++;
28      }while( $sig(Attribute[i]) > 1 \wedge i < |C|$ );
29  }//end CAMARDF
```

where $DF \backslash \{Attribute[i-1]\}$ denotes $\forall f_i \in DF$, if $Attribute[i-1] \in f_i$, then $f_i = f_i - \{Attribute[i-1]\}$.

Reduct and *MinReduct* are global variables in the algorithm. The variable sequence in current depth search path is saved in *Reduct*, the current candidate minimal reduct is saved in *MinReduct* and the last *MinReduct* is the optimal result that we want to find. The operation ComputeSIG in line 1 computes the significance for each variable of discernibility function DF. The variable significance in *CAMARDF* is measured by their frequencies in boolean functions during decomposition procedures. The operation SortSIG in line 2 sorts variables from high to low based on their significance, and the variable order is saved in array *Attribute*. These two steps are corresponding to the depth search strategy II.

The completeness of the algorithm *CAMARDF* for minimal attribute reduction can be guaranteed by theorem 2 and theorem 3. In the implementation, line 5 to line 9, line10 to line 16, line 18 to line 24 are corresponding to the depth search strategy III, the width search strategy, the depth search strategy I respectively and the terminal constrains of do-while sentence reflect theorem 5.

4 Experimental Analyses

The algorithm *CAMARDF* is tested on a personal computer with Intel Pentium Dual-Core E2140 1.6GHz processor and 1Gb memory. The operation system is Windows XP and the programs are implemented on VC 6.0. Ten UCI data sets[12] are chosen for experiments. In order to illustrate the reduct, which is found by *CAMARDF*, is a minimal reduct, all reducts of each data set are computed by using attribute reduction algorithm in [11]. The discernibility function and the reducts of each data set are shown in detail in table 1.

Table 1. The discernibility functions and reducts of some UCI data sets

data sets	No. of objects	No. of attributes	clauses				reducts			
			Num	Max	Min	Avg	No. of reducts	No. of MR	MaxL	MinL
zoo	101	17	14	6	1	3	33	7	7	5
breast	699	10	19	5	1	3	20	8	5	4
mushroom	8124	23	30	12	2	6	292	13	8	4
chess	3196	37	29	2	1	1	4	4	29	29
tic-tac-toe	958	10	36	2	2	2	9	9	8	8
soy	47	36	99	14	6	9	756	4	8	2
audiology	200	70	202	10	1	5	113329	4	31	12
connect	10000	43	440	2	2	2	32	9	36	25
led24	200	25	2458	12	3	8	66800	95	15	11
DNA	200	61	11760	53	30	44	—	—	—	5

In table 1, *Num*, *Max*, *Min* and *Avg* denote the number of clauses, the maximal length of clauses, the minimal length of clauses and the average length of clauses respectively. MR means minimal reducts of the data sets. MaxL and MinL denotes the maximal length and the minimal length of reducts respectively. There are too many reducts of DNA data set (StatLog version) and the memory is overflowed during the implementation, the total number of reducts for DNA data set can not be tested.

The result, which is found by *CAMARDF* for each data set, is a real minimal reduct according to all reducts of this data set. For DNA data set, we can't get the all reducts, but we can get a minimal reduct by *CAMARDF*, it is very important for application. From the data presented in table 1, the length of a minimal reduct is very short compared with the number of condition attributes. That means we can use only several attributes to describe original data sets without losing any information. The time for searching a minimal reduct of a given data set with *CAMARDF* is presented in table 2.

Table 2. The time for searching a minimal reduct of a given UCI data set

data sets	No. of objects	No. of attributes	No. of clauses	time for searching a minimal reduc		
				DF	*CAMARDF*	*TOTAL*
zoo	101	17	14	0.015	0	0.015
breast	699	10	19	0.256	0	0.256
mushroom	8124	23	30	130.062	0	130.062
chess	3196	37	29	7.593	0	7.593
tic-tac-toe	958	10	36	0.531	0	0.531
soy	47	36	99	0.015	0	0.015
audiology	200	70	202	0.468	2.094	2.562
connect	10000	43	440	343.984	0.047	344.031
led24	200	25	2458	6.281	4.922	11.203
DNA	200	61	11760	70.890	25.594	96.484

Remarks: The unit of time is sec. and 0 denotes that time is littler than 0.001s.

As shown in table 2, the minimal reduct can be attained quickly based on *CAMARDF* after *DF* constructed well. The time of *CAMARDF* is only equal to zero for many data sets in table 2. It can be found that the main time for searching a minimal reduct is spent on constructing reduced discernibility function *DF*. However, the time for constructing reduced discernibility function is related to the number of objects and the number of attributes of data sets directly, namely, data sets themselves. How to obtain the reduced discernibility function of a given decision table effectively is our future works.

5 Conclusion

Efficient and complete algorithms for minimal attribute reduction play an important role both in theory and practice. According to the properties of discernibility function, some search strategies are put forward. A complete algorithm *CAMARDF* for searching a minimal reduct of a given data set is constructed and its efficiency can be illustrated by UCI data sets. The algorithm *CAMARDF* is based on logic reasoning, so it can be applied for any information systems, no matter they include decision attributes or do not include decision attributes, they are consistent decision tables or inconsistent decision tables. A minimal reduct will be obtained efficiently as long as their discernibility functions are constructed reasonably.

Acknowledgements

The research in this paper is supported by National Natural Science Foundation of China (Serial No. 60475019, 60775036) and the Ph.D. programs Foundation of Ministry of Education of China (Serial No. 20060247039).

References

1. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
2. Zhao, Y., et al.: A general definition of an attribute reduct. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS, vol. 4481, pp. 101–108. Springer, Heidelberg (2007)
3. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R., et al. (eds.) Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1991)
4. Hoa, N.S., Son, N.H.: Some efficient algorithms for rough set methods. In: Pro. of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 1996), vol. 2, pp. 1451–1456 (1996)
5. Xu, Z.Y., et al.: A quick attribute reduction algorithm with complexity of $max(O(|C||U|, O(|C|^2|U/C|))$. Journal of Computers 29(3), 391–399 (2006)
6. Miao, D.Q., Wang, J.: Information-based algorithm for reduction of knowledge. In: Pro. of the 1997 IEEE International Conference on Intelligent Processing Systems (ICIPS 1997), vol. 2, pp. 1155–1158 (1997)
7. Wang, G.Y., et al.: A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fundamenta Informaticae 68(3), 289–301 (2005)
8. Hu, X.H., Cercone, N.: Learning in relational databases: A rough set approach. International Journal of Computational Intelligence 11(2), 323–338 (1995)
9. Wang, J., Miao, D.Q.: Analysis on attribute reduction strategies of rough set. Journal of Computer Science and Technology 13(2), 189–193 (1998)
10. Yao, Y.Y., Zhao, Y., Wang, J.: On reduct construction algorithms. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS, vol. 4062, pp. 297–304. Springer, Heidelberg (2006)
11. Starzyk, J.A., Nelson, D.E., Sturtz, K.: A mathematical foundation for improved reduct generation in information systems. Knowledge and Information Systems 2, 131–146 (2000)
12. Asuncion, A., Newman, D.U.: Repository of machine learning databases. University of California, Irvine (2007), `http://archive.ics.uci.edu/ml/index.html`