IOP Conf. Series: Materials Science and Engineering **301** (2018) 012134 doi:10.1088/1757-899X/301/1/012134

Fast Reduction Method in Dominance-Based Information Systems

Yan Li, Qinghua Zhou and Yongchuan Wen

School of Applied Mathematics, Beijing Normal University, Zhuhai, Zhuhai City 519087, Guangdong, China

Email: ly@hbu.edu.cn

Abstract. In real world applications, there are often some data with continuous values or preference-ordered values. Rough sets based on dominance relations can effectively deal with these kinds of data. Attribute reduction can be done in the framework of dominance-relation based approach to better extract decision rules. However, the computational cost of the dominance classes greatly affects the efficiency of attribute reduction and rule extraction. This paper presents an efficient method of computing dominance classes, and further compares it with traditional method with increasing attributes and samples. Experiments on UCI data sets show that the proposed algorithm obviously improves the efficiency of the traditional method, especially for large-scale data.

1. Introduction

Rough set theory [1] is a powerful mathematical tool to deal with imprecise, incomplete and incompatible knowledge. It has been widely used in decision making, data mining and pattern recognition [2, 3], among others. In practical problems, there are often some continuous valued or preference ordered data, such as the attribute "score" can be numerical or can be divided into three attribute values: high, medium and low. This type of attributes is often used to evaluate objects in the universe, for example, to evaluate students by their scores of some subjects. In this case, the ordered information contained in the attribute values should not be ignored for better understanding the data. Since the traditional rough set (TRS) cannot effectively deal with this kind of information, dominance relation based rough set approach was proposed [4, 5] by replacing equivalence relations in TRS with dominance relations. DRSA is very useful to deal with practical problems with continuous-valued partial ordered attributes [6-8]. Ever since DRSA has been proposed, many researches and improvements have been made in the literature [9-12]. Nowadays, efficiently processing of large-scale data with dominance relations has become a main concern [13, 14]. A fast algorithm is developed in [15] to reduce the time efficiency of computing dominance classes through gradually reducing the search space. This method can further improve the computational efficiency of attribute reduction as well as rule extraction. Based on this fast algorithm, we develop a complete method of attribute reduction and compare this method and the traditional dominance relation based method with increasing attributes or sample to show its effectiveness and potential usefulness on large-scale datasets.

2. Preliminaries

In this section, some necessary concepts are given for reference.

Definition 1(Target information system) A 4-tuple $S = (U, A, V, f)$ is called as a target information system, where U is a non-empty set of objects; A is a non-empty set of attributes. $A =$

$C \cup \{d\}$ where C is a set of conditional attributes and d is a decision attribute. V is the set of attribute values, and $f: U \times A \to V$ assigning each attribute of each object with a value in V.

For a given information system S, if there is a partial order relation "$\geq_a$" on the value range of an attribute $a \in A$, we call a as a criterion. $x, y \in U, x \geq_a y$ represents that x is at least as good as y under criterion a, i.e., x is superior than y on a. When all attributes in the information system are criteria, the information system is called ***an ordered information system*** [8].

For the attribute set $B \subseteq A$, $x \geq_B y$ means that x is superior than y on all the criteria in B.

Definition 2 (Dominance/inferior relation) In an ordered information system, for any set of attributes $B \subseteq A$, the dominance relation $R_B^{\leqslant}$ is defined as

$$R_B^{\leqslant} = \{(x, y) \in U^2 \mid f(x, a) \leq_a f(y, a), a \in B\}. \tag{1}$$

Obviously, if x and y satisfy $(x, y) \in R_B^{\leqslant}$, y is called to be superior to x on each attribute in B. In contrast, for any set of attributes $B \subseteq A$, the inferior relation $R_B^{\geqslant}$ is defined as

$$R_B^{\geqslant} = \{(x, y) \in U^2 \mid f(x, a) \geq_a f(y, a), a \in B\}. \tag{2}$$

Definition 3 (Dominance/inferior class) In an ordered information system, for any set of attributes $B \subseteq A$, $[x]_B^{\leqslant}$ is called the dominance class of the object x, defining as $[x]_B^{\leqslant} = \{y \in U \mid f(x, a) \leq_a f(y, a), a \in B\}$; $[x]_B^{\geqslant}$ is called the inferior class of object x, defining as

$$[x]_B^{\geqslant} = \{y \in U \mid f(x, a) \geq_a f(y, a), a \in B\}. \tag{3}$$

Definition 4. (Dominance matrix) Given a target information system $S = (U, A, V, f)$, for $B \sqsubseteq A$, dominance matrix $A^{\leq}$ is defined as

$$A^{\leq}(x_i, x_j) = \begin{cases} 1, if\ f(x_j, a) \geq f(x_i, a) \\ 0, \quad\quad otherwise. \end{cases} \tag{4}$$

where i=1,2,…,n, j=1,2,…,n; $A^{\leq}(x_i, x_j) = 1$ represents that x_j is superior to x_i.

On the basis, the approximation sets of a given target concept and attribute reduction can be defined as follows.

Definition 5. (Lower and upper approximations) Suppose that (U, A, V, f) is a continuous-valued information system. For $X \subseteq U$, define

$$\overline{R}_B^{\leqslant}(X) = \{x \mid [x]_B^{\leqslant} \cap X \neq \phi\}, \underline{R}_B^{\leqslant}(X) = \{x \mid [x]_B^{\leqslant} \subseteq X\}. \tag{5}$$

as $R_B^{\leqslant}$-upper approximation and $R_B^{\leqslant}$-lower approximation of the given concept X.

$\overline{R}_B^{\geqslant}(X) = \{x \mid [x]_B^{\geqslant} \cap X \neq \phi\}$ and $\underline{R}_B^{\geqslant}(X) = \{x \mid [x]_B^{\geqslant} \subseteq X\}$ are called the $R_B^{\geqslant}$-upper approximation and $R_B^{\geqslant}$-lower approximation of X. The union of lower approximations of all decision classes is called the **positive region** of the information system.

Definition 6. (Attribute reduction) *S*uppose that (U, A, V, f) is the continuous-valued information system, and the dominance relation $R_B^{\leqslant}$ and the inferior relation $R_B^{\geqslant}$ are given. If $R_B^{\leqslant} = R_A^{\leqslant}$ or $R_B^{\geqslant} = R_A^{\geqslant}$, B is called the dominance coordination set or inferior coordination set. If B is a dominance/ inferior coordination set, and any true subset of B is not, then B is called as an attribute reduction of attribute set A, referred to as the dominance reduction set or the inferior reduction set.

3. Fast Algorthim for Attribute Reduction in Dominance-based Information Systems

One of the important applications of rough set theory is to reduce attributes in data. The most informative information is preserved while some redundant attributes are removed. Based on the concepts in Section II, computing dominance classes is a necessary step for the computation of approximation sets and then the attribute reduction. In the following, a fast algorithm is presented.

3.1. Computing Dominance Classes

Input: An ordered information system $S=(U,A)$,

$$U=\{x_1,x_2,...,x_n\},\ A=\{a_1,a_2,...,a_m\}. \quad (6)$$

Output: The dominance classes of all objects in U.

Step 1. The current attribute and the dominance class for an arbitrary object x in the universe are initialized as j=1; $a=a_j$; $[x]^{\geqslant}=U$. While $j<m$, performs step 2;

Step 2. Compute the dominance class of object x on current attribute a by comparing the attribute values of x with the attribute values of the $(\left|[x]^{\geqslant}\right|-1)$ remaining objects in the dominance class $[x]^{\geqslant}$ (instead of the entire universe U), and then the dominance class of x is updated as $[x]^{\geqslant}=[x]^{\geqslant}\cap[x]_a^{\geqslant}; j=j+1$.

The dominance class of object x can be obtained until all attribute are added;

Step 3. Steps 1-2 are repeatedly performed for all the n objects.

3.2. Computing the Positive Region Based on Domiance-Equivalence Relations

Based on the obtained dominance classed from Section A, the positive region can be quickly computed as follows.

Step 1. Computing the decision classes according to the given target information system as $U/d=\{D_1,D_2,...,D_j\}$;

Step 2. Computing the dominance matrix $A^{\leq}$ and decision class matrix *D*, and conducting "or" operation and obtain the positive region of each decision class:

$$POS_A^{\leq}(D_j)=\{x_i \mid A(i,n)\vee D(j,n)=D(j,n)\}. \quad (7)$$

Step 3. Output positive region $POS_A^{\leq}(D)=\cup_{j=1}^{k} POS_A^{\leq}(D_j)$.

3.3. Computing Attribute Reduction

According to the obtained positive region, the discernibility matrix based on dominance relations can be computed. Then the attribute core and reduction can be also computed by conducting "and"/ "or" operations. The attribute core is found by detecting the singular element in the discernibility matrix and then other attributes are gradually added into the core until the intersection of the current reduction set and all the elements in the matrix is not empty.

4. Experimental Results

The proposed algorithm is compared with the traditional algorithm with respect to the time efficiency. Here, traditional method refers to the attribute reduction algorithm which computes the dominance classes by scanning all the attributes and sample two times, and does not use the concept of dominance matrix to compute the positive region and attribute reduction.

4.1. Data Sets

Totally, six groups of datasets are selected from the UCI repository [16]. The range of the number of samples is from 198 to 10000; and the number of attributes is from 8 to 60. We divide the experiment into two parts: (1) in the first part, we select three datasets which have comparative more samples and less attributes. The proposed method is compared with the traditional method by adding more attributes each time on the original data. (2) In the second part, we select another three data sets which have comparatively less samples but more attributes. In this part, the two methods are compared with increasing samples. The "Size" of data is denoted as (the number of samples* the number of attributes).

4.2. Running Time of Computing Dominance Classes

In this section, we compare the fast algorithm for computing dominance classes and the traditional algorithm with increasing attributes. Three dataset are used: Clapping data, Running data and the ae-test data. In each round of the experiments, two randomly selected attributes are added to the original dataset. The two algorithms are implemented on these datasets and the running time is recorded and compared. The following tables in TABLE I (a) (b) (c) on three groups of data show the results. In each table, the first line of results indicates those on the original dataset. In the following lines, each time two attributes are added on the original data.

Table 1. Running time of the two algorithms with the increase of attributes

(a) Clapping Dataset

Data	Size	Traditional Method	Fast Method	Reduction Rate
		t_1(/s)	t_2(/s)	Percentage(%)
Clapping0	10000*8	117.955	105.823	10.285
Clapping1	10000*10	143.233	106.460	25.674
Clapping2	10000*12	168.317	109.273	35.079
Clapping3	10000*14	195.411	110.913	43.241
Clapping4	10000*16	219.116	112.506	48.655
Clapping5	10000*18	247.486	115.392	53.374
Average	*10000*13*	*181.920*	*110.061*	*36.051*

Reduction rate=$(t_1-t_2)/t_1\times100\%$ (the same in all the following tables)

(b) Running Dataset

Data	Size	Traditional Method	Fast Method	Reduction Rate
		t_1(/s)	t_2(/s)	Percentage(%)
Running0	9964*8	121.137	104.946	13.366
Running1	9964*10	152.289	115.573	24.109
Running2	9964*12	180.238	124.123	31.134
Running3	9964*14	203.675	126.716	37.785
Running4	9964*16	231.723	128.601	44.502
Running5	9964*18	256.192	133.220	48.000
Average	*9964*13*	*190.876*	*122.197*	*33.149*

(c) ae-test Dataset

Data	Size	Traditional Method	Fast Method	Reduction Rate
		t_1(/s)	t_2(/s)	Percentage(%)
ae-test1	5687*12	51.593	33.047	35.947
ae-test1	5687*14	61.728	37.203	39.731
ae-test2	5687*16	69.243	37.677	45.587
ae-test3	5687*18	80.990	42.047	48.084
ae-test4	5687*20	88.792	44.021	50.422
ae-test5	5687*22	98.325	46.154	53.060
Average	*5687*17*	*75.112*	*40.025*	*45.472*

We can see from these tables that the fast method is much more efficient in computing the dominance classes which can reduce at most over 50% of the running time by using traditional method. Another observation is, with the increasing of attributes, the effect is more obvious.

4.3. Running Time with Increasing Samples

In this section, we increasingly add the number of samples on the original data sets. Three dataset are used: Thyroid data, Sonar data and Wpbc data. In each round of the experiments, one copy of original data is added and the number of samples is proportionally increased. The following tables in TABLE II (a) (b) (c) on the three groups of data show the results. The same as those in TABLE I, here in each table, the first line of results indicates those on the original dataset. In the following lines, each time one copy of the original data is added.

Table 2. Running time of the two algorithms with the increase of samples

(a) Thyroid Dataset

Data	Size	Traditional Method	Fast Method	Reduction Rate
		t_1(/s)	t_2(/s)	Percentage (%)
Thyroid1	970*28	3.843	3.021	21.390
Thyroid2	1940*28	15.886	11.318	28.755
Thyroid3	2910*28	35.246	24.208	31.317
Thyroid4	3880*28	63.351	41.414	34.628
Thyroid5	4850*28	92.103	59.375	35.534
Thyroid6	5820*28	129.110	79.574	38.367
Average	*3395*28*	*56.590*	*36.485*	*31.665*

Reduction rate=$(t_1-t_2)/t_1\times100\%$ (the same in all the following tables)

(b)Sonar Dataset

Data	Size	traditional algorithm	fast algorithm	Reduction rate
		t_1(/s)	t_2(/s)	Percentage (%)
Sonar0	208*60	0.317	0.242	23.659
Sonar1	416*60	1.292	0.894	30.805
Sonar2	624*60	2.954	1.845	37.542
Sonar3	832*60	5.146	3.077	40.206
Sonar4	1040*60	8.199	4.501	45.103
Sonar5	1248*60	11.674	6.150	47.319
Average	*728*60*	*4.930*	*2.785*	*37.439*

(c) Wpbc Dataset

Data	Size	traditional algorithm	fast algorithm	Reduction rate
		t_1(/s)	t_2(/s)	Percentage (%)
Wpbc0	198*33	0.165	0.107	35.152
Wpbc1	396*33	0.634	0.410	35.331
Wpbc2	594*33	1.515	0.803	46.997
Wpbc3	792*33	2.652	1.290	51.357
Wpbc4	990*33	4.165	1.943	53.349
Wpbc5	1188*33	6.086	2.642	56.589
Average	*693*33*	*2.536*	*1.199*	*46.463*

Similar to the results in TABLE I, with the increasing number of samples, the reduction rate also increases greatly, which is at most 56.589% in the Wpbc data. All these results show the potential use of the fast algorithm in large-scale datasets with more attributes and samples.

5. Conclusions
This paper presents a fast attribute reduction method for dominance relation-based information systems. A lot of experiments have been conducted to compare the proposed method and the traditional method, and the results show that the fast method is much more effective with increasing number of samples and attributes.

6. Acknowledgment
This work is supported by NSFC (No. 61473111) and the Natural Research Program Foundation of Hebei University (No. 799207217069).

7. References
[1] Z. Pawlak, "Rough set", International Journal of Computer and Information Science, 1982, 11(5): 341-356.
[2] Y. Wang, D. Q. Miao, and Y. J. Zhou, "A survey of rough set theory and its application", Pattern Recognition and Artificial Intelligence, 1996, 9(4): 337-344.
[3] Z. Pawlak, A. Skowron, "Rough sets: some extensions", Information Sciences, 2000, 14(4): 1-12.
[4] S. Greco, B. Matarazzo, and R. Slowingski, "Rough approximation of a preference relation by dominance relation", European Journal of Operation Research, 1999,117(1): 63-83.
[5] S. Greco, B. Matarazzo, and R. Slowingski, "Rough approximation by dominance relations", International Journal of Intelligent Systems, 2002, 17(2): 153-171.
[6] S. Greco, B. Matarazzo, and R. Slowingski, "A new rough set approach to multicriteria and multiattribute classification", Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, 1998, 1424: 60-67.
[7] S. Greco, B. Matarazzo, and R. Slowingski, "A new rough sets approach to evaluation of bankruptcy risk", in Operational Tools in the Management of Financial Risks, Dordrecht: Kluwer, 1999: 121-136.
[8] S. Greco, B. Matarazzo, and R. Slowingski, "Rough sets theory for multi-criteria decision analysis", European Journal of Operational Research, 2001, 55(1): 1-47.
[9] Y.H. Qian, J.Y. Liang, and C.Y. Dang, "Incomplete multi-granular rough set", IEEE Transactions on Systems, Man and Cybernetics, 2010, 40(2): 420-431.
[10] L. H. Wei, Z. M. Tang, and B. L. Yang, "Rough set based on the dominance relation and knowledge reduction for incomplete fuzzy systems", Computer Science, 2009, 36(6): 192-195.
[11] K. Zaras, J.C. Marin, and B. Boudreau-Trude, "Dominance-based rough set approach in selection of portfolio of sustainable development projects", American Journal of Operations Research, 2012, 2:502-508.
[12] T. Azar, H. H. Inbarani, and K. R. Devi, "Improved dominance rough set-based classification system", Neural Computing and Applications, 2016, 6:1-16.
[13] S. S. Qu, Y. S. Lu, "Research on fast attribute reduction algorithm based on rough set", Computer Engineering, 2011, 181(5): 987-1002.
[14] D.W. Xiao, G.Y. Wang, and F. Hu, "A fast parallel attribute reduction algorithm based on rough set theory", Computer Science, 2009, 36(3): 208-211.
[15] Y. Li, Q. Yu. A Fast Algorithm for Computing Dominance Classes, International Journal of Intelligent Information and Management, vol. 5, no. 6, pp. 45-48, 2016
[16] K. Bache, M. Lichma. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.