

Incremental Updating Algorithm for Attribute Reduction Based on Ordered Element Pairs

Baohua Liang¹

*Institute of Information Engineering, Chaohu University, Hefei City 238000,
China
liangbh426@126.com*

Abstract

According to the condition of attribute reduction of dynamic decision tables, the concept of indistinguishable element pairs is introduced firstly in this paper. On the basis of the discernibility matrix definition, a large number of element pairs without necessary comparison are eliminated, which reduce the search space of algorithm. Then the paper makes a serial number for each element and links indistinguishable element pairs about different condition attributes, and consequently gets an ordered table. It is easy to receive the attribute reduction of decision tables through the characteristic of the order table. At last, based on the former attribute reduction, a dynamic incremental updating algorithm was proposed. And through examples and experimental comparisons, the effectiveness and feasibility of the algorithm will be showed.

***Keywords:** Rough set, attribute reduction, element pairs, Order list, incremental updating*

1. Introduction

Data Mining could find lots of potential and useful information from mass data. But in terms of objects of mining, its time and space grows explosively, among which lots of useless noise data exists. Therefore, it is necessary to eliminate that information before data mining so as to decrease the consumption of time and space.

Rough set theory [1] is proposed by mathematician Pawlak in 1982. It is a new tool to deal with fuzzy and uncertain information and does not need any priori knowledge. Rough set theory has been successfully applied in the fields like decision support, intelligent control, *etc.* [2-3]. Attribute reduction is one of the highlights of Rough set theory [4-6]. It intends to eliminate redundancy data by using fewer attribute, without affecting the capacity of classification, so as to remove obstacles of data mining preprocess. At present, there exist many attribute reduction algorithms about static information system [7-8], but the information system in real life is constantly changing. In order to adapt this changing system, the increment updating algorithm can be an efficient tool.

The research about increment updating is very few. Hu Feng *et al.* in literature [9] designed a increment updating algorithm based on the positive region and raised the efficiency of updating process. But it didn't make full use of the updating core and the inconsistencies condition about adding some new objects needs to be improved. literature [10] proposed a updating attribute reduction algorithm based on differential matrix and the minimum attributes reduction set can be acquired quickly, but it consumes much time when the algorithm updates the set of discernibility element. Liu Yang in literature [11] proposed a updating algorithm based on discernibility matrix. But it didn't simplify the decision table, which led to the high complexity of time and space.

¹ *Corresponding Author: Baohua Liang

The size of dataset directly affects the efficiency of attribute reduction algorithm.

Therefore, firstly, this paper will use the way proposed by literature [12] to get the simplified decision table, positive and negative regions. Element pair is defined in accordance with improved differential matrix, then order those pairs and an effective incremental updating algorithm is proposed to adapt the dynamic order list. Examples and experimental results show the proposed method is feasible and effective.

2. Preliminary

In order to show the thinking in this paper clearly, the basic concepts on attribute reduction are introduced in this section.

Definition 1. [15] Let $S = (U, A, V, f)$ be a decision system where C and D denote respectively conditional attributes and decision attribute. U is a non-empty finite set. $A = C \cup D$ is a set consisting of attributes of objectives, and subsets C and D are respectively called condition attribute set and decision attribute set, $C \cap D = \emptyset$. A takes value in V. Meanwhile, Let $f : U \times A \rightarrow V$ be an information function, i.e. for any $a \in A$ and $x \in U$ then $f(x, a) \in V_a$.

Definition 2. [15] Given the information system $S = (U, A, V, f)$, for every subset $B \subseteq A$, a indiscernibility relation $IND(B)$ is defined as follows,

$$IND(B) = \{(x, y) \in U \times U \mid \forall a \in B, f(x, a) = f(y, a)\} \quad (1)$$

Obviously, if $IND(B)$ denotes as U / B , U / B is an equivalence relation. We assume U / B includes x, the equivalence relation x is defined as follows,

$$[x]_R = \{x \mid \forall y \in U, f(x, R) = f(y, R), x \neq y\} \quad (2)$$

If U / B includes x and y, then S is a consistent information system where $f(x, C) = f(y, C) \wedge f(x, D) = f(y, D)$, otherwise S is an inconsistent information system.

Definition 3. [15] Given the information system $S = (U, A, V, f)$, for every subset $X \subseteq U$ and indiscernibility relation B, the lower approximation set and the upper approximation set of X can be defined by basic set of B as follows,

$$B^-(X) = \cup \{Y_i \mid (Y_i \in U / IND(B), Y_i \cap X \neq \emptyset)\}, \quad (3)$$

$$B_-(X) = \cup \{Y_i \mid (Y_i \in U / IND(B), Y_i \subseteq X)\}. \quad (4)$$

Definition 4. [15] Given the information system $S = (U, A, V, f)$, U is the domain of objectives. P and Q are two equivalence relations defined on U. The positive region of Q with respect to P is defined as follows,

$$POS_p(Q) = \bigcup_{X \in U / Q} P_-(X) \quad (5)$$

The negative region is defined as $U - POS_p(Q)$

Definition 5. Given the information system $S = (U', A, V, f)$, U' is the domain of simplified decision table objectives and $U' = U'_{POS} \cup U'_{NEG}$, and subsets U'_{POS} and U'_{NEG} are respectively called the positive region and the negative. $A = C \cup D$ is a set consisting of attributes of objectives, and subsets C and D are respectively called condition attribute set and decision attribute set, $C \cap D = \emptyset$. U'_{POS} is the positive region of D with respect to C as

about simplified decision table. U'_{NEG} is the negative region except U'_{POS} in U' . The elements of discernibility matrix are defined as follows,

$$m_{ij} = \begin{cases} \{a \mid a \in C, x, y \in U'_{POS} \wedge f(x, a) \neq f(y, a) \\ \wedge f(x, D) \neq f(y, D) \wedge x \neq y\} \\ \{a \mid a \in C, x \in U'_{POS}, y \in U'_{NEG} \wedge f(x, a) \neq f(y, a)\} \\ \emptyset \text{ otherwise} \end{cases} \quad (6)$$

Definition 6. Given the information system $S=(U',A,V,f)$, U' is the domain of simplified decision table objectives and $U'=U'_{POS} \cup U'_{NEG}$, and subsets U'_{POS} and U'_{NEG} are respectively called the positive region and the negative. The element pairs are defined as follows,

$$Mpair = \{(x, y) \mid (x \in U'_{POS} \wedge y \in U'_{NEG}) \vee (x, y \in U'_{POS} \wedge f(x, D) \neq f(y, D))\} \quad (7)$$

Definition 7. If we give a different number to every element pairs of Mpair, then the indistinguishable order string of attribute set B as follows:

$$DisString_B = \{s \mid \forall b \in B, (x, y) \in Mpair \wedge f(x, b) = f(y, b), s \text{ is a number according element pairs } (x, y)\} \quad (8)$$

Theorem 1. As for $S=(U',C \cup D)$, U' is a simplified decision table. C is condition attribute set. Let B a subset of C and b is a subset of $C-B$, then $DisString_{B \cup b} \subseteq DisString_B$.

Proof: Suppose, any s is a subset of $DisString_{B \cup b}$, and serial number s stands for the corresponding element(x, y) in Mpair. According to the definition 6, $(x, y) \in Mpair$ and $f(x, B \cup b) = f(y, B \cup b)$. We can get the result of $f(x, B) = f(y, B)$ because B is subset of $B \cup b$. S is arbitrary, so $DisString_B$ includes s. In conclusion, $DisString_{B \cup b}$ is a subset of $DisString_B$, in other words $DisString_{B \cup b} \subseteq DisString_B$.

Proposition 1. The shorter the indistinguishable order string $DisString_B$, the stronger its ability to distinguish will be.

Proof: By definition 4, we know that in a given decision table, the corresponding Mpair elements are settled. And now $DisString_{C_i}$ writes down the indistinguishable element pairs of Mpair about conditional attribute C_i . If the shorter the number string of $DisString_{C_i}$ is, the more indistinguishable the less samples is on condition attribute C_i . On the contrary, the more the samples can be distinguished, the stronger its distinguishing ability is.

3. Attribute Reduction Incremental Updating Algorithm Based On Ordered Element Pairs

3.1. Attribute Reduction Algorithm Based On Ordered Element Pairs

Most of the traditional discernibility matrix algorithms use two-dimensional table to store the information. Much of the information is repeated. When the effect is tested by program, a great deal of computer's internal storage is used to store the two-dimensional table. The attributes with highest degree of distinction will be selected in the process of selecting important properties. However, it costs a lot of time because it needs scan the whole target data. Thus, this paper proposes an indistinguishable element pairs algorithm. Which does not need to store all the two-dimensional table. It will preserve the quality of being straightforward and easy to understand and will also reduce effectively consumption of time and space.

Algorithm 1 Attribute reduction algorithm based on ordered element pairs

Input : U'_{POS}, U'_{NEG}, C, D

Output: attribute reduction sets Red

(1) Compute the U'_{POS} and U'_{NEG} through the algorithm of literature [3];
 (2) Combine U'_{POS} and U'_{NEG} an element pairs as U_1 by definition 4, in a same way with the internal elements of U'_{POS} as U_2 , then make serial number for each element pairs, and label U_1 with ‘*’;

(3) Let $C' = C, Red = \emptyset$;

(4) for(i from 1 to $|C'|$) {

(4.1) Divide Data according to different values of conditional attribute C_i , make the samples element pairs in each sub-division and correspond them to the serial number in Mpair;

(4.2) Collect them into one OrderList(C_i);

(4.3) OrderList = min[List(C_i)];

(4.4) $C' = C' - C_{min}$; // C_{min} is the NO of minimum List(C_i)

(4.5) $Red = Red \cup C_{min}$; }

(5) While(OrderList = \emptyset || $C' = \emptyset$) { // Order-List is null or C' is null

(5.1) Let tempList = OrderList ;

(5.2) Let tempList = OrderList ;

(5.3) for(j=1 to $|C'|$) // Write down the overlapping part of OrderList and $List(C_j)$

Let OrderList = min (InsertSet(tempList, $List(C_j)$));

(5.4) Set the minimum of InsertSet(tempList, $List(C_j)$) function to OrderList

(5.5) Set $C' - C_k$ to C' ; // attribute C_k is selected

(5.6) Set $Red \cup C_k$ to Red; }

algorithm 2 $List \cap List(C_j)$ // calculate the intersect of two order lists

InterSet (int[] arr1, int[] arr2)

{ count = 0;

for(i=j=0; i<arr1.length || j<arr2.length;)

if(arr1[i] == arr2[j]) {

temp[count++] = arr1[i];

i++; j++; }

else if(arr1[i] < arr2[j])

i++;

else

j++;

return temp; }

Time complexity of step (1) is $O(|C||U|)$, and time complexity of step (2) is $O(|U'_{POS}||U'|)$, and time of (3) is scarce, and time complexity of (4) is $O(|C||U'_{POS}||U'|C|)$. As for (5), even in the best case, time complexity is $O(|C|\min(|list(C_i)|))$, and the worst case is $O(|C|^2 \min(|list(C_i)| + |list(C_j)|))$. Most of the space consumption is used for storing the order lists with according attribute in Mpair. As the order lists can be taken out one by one in experiment without storing the whole order lists in the storage memory, the length of Order List is unlikely to go beyond $|U'_{POS}||U'|$, and the complexity space is $O(|U'_{POS}||U'| + \max(|list(C_i)|))$.

3.2. Incremental Updating Algorithm

3.2.1. Incremental Analysis

Data of Information System is constantly changing in our real life. How, under the condition of target data's growth, to get the new reduction sets by availing the results of previous reduction and taking only a small amount of calculation? The problem has attracted the attention of most researchers. According to my analysis of discernibility matrix element pairs, assuming that x is an original object and y is a new object, the possible cases as follow:

(1) If exists $x \in U'_{POS}$ and for any $C_i \in C \wedge f(x, C_i) = f(y, C_i)$, the result of Mpair is not changeable where $f(x, D) = f(y, D)$. If $f(x, D) \neq f(y, D)$, then x merges into U'_{NEG} and delete the pairs of x and element of subset U'_{NEG} from Mpair, and label '*' for pairs of x & U'_{POS}

(2) If exists $x \in U'_{NEG}$ and for any $C_i \in C \wedge f(x, C_i) = f(y, C_i)$, the result of Mpair is not changeable

(3) If for any $x \in U'$ and $f(x, C) \neq f(y, C)$ then x merge into U'_{NEG} and Mpair is changeable as following: $Mpair \cup \{(x, y) \mid x \in U'_{POS} \wedge f(x, D) \neq f(y, D) \vee x \in U'_{NEG}\}$

3.2.2. Increment Updating Algorithm

algorithm 3 attribute reduction

Input: simplified decision table $S' = (U', A, V, f)$, y , Mpair // y is new an object

Output: attribute reduction sets Red

(1) if $(\exists x \in U'_{POS} \wedge \forall C_i \in C \wedge f(x, C_i) = f(y, C_i))$ {

(1.1) if $(f(x, D) = f(y, D))$

Red is not changeable ;

(1.2) else { //in condition of $f(x, D) \neq f(y, D)$

Labeled with '*' the element pairs which combine U'_{POS} with x ;

Delete the serial number of the element pair of x and U'_{NEG} ;

Update the order string of List(C_i);

Let $U'_{POS} = U'_{POS} - \{x\}$;

Let $U'_{NEG} = U'_{NEG} \cup \{x\}$;

Remove a redundant attribute from Red sets and delete reversely redundancy attributes until the intersection of attribute order string in Red become null.

Then return Red. }

}

(2) if exists $x \in U'_{NEG}$ and for any $C_i \in C$ where $f(x, C_i) = f(y, C_i)$ then Red is unchangeable;

(3) if exists $C_i \in C$ and for any $x \in U'_{POS}$ where $f(x, C_i) \neq f(y, C_i)$ then

(3.1) Let y and everyone of U'_{POS} combine with Mpair, increases number and updating the List(C_i);

(3.2) Let y and everyone of U'_{NEG} do the same treatment as U'_{POS} and label it with*.

(3.3) if List(Red) is null, Red is not changeable.

else{

while (List(Red) $\neq \emptyset$) {

foreach C_j in $C' - Red$

Take the minimum of InterSet (List(Red), List(C_j)), if there are multiple conditions which can meet the requirement, choose any one .

Let Red =Red - C_k ;
 Until List(Red) is \emptyset }
 }

Complexity analysis: when new object y joins the data set, the worst time of (1.1) is $O(|C||U_{POS}'|U')$; The worst time of (1.2) is

$O(|C||U_{POS}'|) + O(|Mpair|) + Min(O(|Red|^2 length(ListC_i)))$, and $O(|Mpair|)$ is the number of element pairs in definition 4, and it equals $O(|C||U_{POS}'|U')$; The time complexity of (2) is $O(|C||U_{NEG}'|)$.The time of (3.1) and (3.2) is $O(|C||U_{POS}'|U')$ and (3.3) is $O(|C||U_{POS}'|U') + Min(O(|Red|^2 length(ListC_i)))$.As $O(length(ListC_i))$ is no more than $O(|U_{POS}'|U')$, the total time complexity is:

$Max(O(|C||U_{POS}'|U'), O(|Red|^2|U_{POS}'|U'))$ It is clear that the space complexity is $O(|C||U_{POS}'|U')$. If we examine sample data according to the condition attribute one by one, the space complexity is $O(|U_{POS}'|U')$. Both the time and the space complexity are $O(|C|^2|U|^2)$ in literature [13]. But in literature [14], the time complexity is $Max(O(|C||U'), O(|Red|^2|U_{POS}'|U'))$ and the space complexity is $O(|C||U_{POS}'|U')$. The time complexity of literature [14] is slightly lower than algorithm 3. In actual practice, the time of this paper didn't appear to be much different from literature [14] as we select the minimum order string every time, but the space of literature [14] exceeds algorithm 3 greatly. To sum up, the time and space of this paper are better than literature [13] and literature [14].

4. Calculation Example and Experimental Analysis

Suppose, S' is simplified decision table, Let $S'=(U',C,D,V,f)$.The data of S' used in the example is outlined in Table 1.

Table 1. Decision Table

Data	TID	C ₁	C ₂	C ₃	C ₄	D
POS	1	1	1	2	1	0
	2	1	2	1	1	0
	3	2	1	2	2	1
	4	2	2	2	1	1
	5	2	2	1	2	2
NEG	6	1	1	1	2	3

4.1. Example about Attribute Reduction Progress

Mpair can be obtained according to definition 4 and number every element as shown in Table 2.

Table 2. Table of Order Element Pairs

NO.	x	y	NO.	X	y
1	1	3	8*	2	6
2	1	4	9	3	5
3	1	5	10*	3	6
4*	1	6	11	4	5
5	2	3	12*	4	6
6	2	4	13*	5	6
7	2	5			

Divide U' according to condition attribute, then we can get as following:

$$U'/C_1 = \{\{1,2,6\}, \{3,4,5\}\}; U'/C_2 = \{\{1,3,6\}, \{2,4,5\}\};$$

$$U'/C_3 = \{\{1,3,4\}, \{2,5,6\}\}; U'/C_4 = \{\{1,2,4\}, \{3,5,6\}\};$$

The element value of the underlined part in each division is equal. We can conclude according to definition 4 that samples with the same decision values do not need to compare. Combine every two objects in each subdivision into element pairs, and we can get serial numbers based on Mpair. And then form a link table. The results as following:

$$\text{List}(C_1) = (*4, *8, 9, 11); \text{List}(C_2) = (1, *4, *6, 7, *10, 11);$$

$$\text{List}(C_3) = (1, 2, 7, *8, *13); \text{List}(C_4) = (2, 6, 9, *10, *13);$$

List(C1) is shortest. So, Let Red = { C₁ }, meanwhile Let OrderList = (*4, *8, 9, 11). In the same way, we can get that List(Red ∩ C₃) = (*8) or List(Red ∩ C₄) = (9) is shortest. Suppose C₃ is selected, let Red = { C₁, C₃ }. Now, the OrderList(Red) is not empty, when C₂ is been added to Red, OrderList(Red) is null, the algorithm is end. So, Red = { C₁, C₂, C₃ }

4.2. Example about Incremental Updating Progress

(1) Suppose the new object y is {1,1,2,1,0}, as for any $C_i \in C$, exists $x_1 \in U'_{POS}$ and then $f(y, C_i) = f(x_1, C_i)$ Red is not changeable according to setp1.1 of algorithm 3. If new object y is {1,2,2,1,1}, exists $x_2 \in U'_{POS}$ and for any $C_i \in C$ where $f(y, C_i) = f(x_2, C_i)$ and $f(x_2, D) \neq f(y, D)$ According to step1.2, numbers such as 5,6,7 are labeled with '*', and the element *8 deleted Let $U'_{POS} = \{1,3,4,5\}$ and $U'_{NEG} = \{2,6\}$ then remove { C₂ } from Red. Now, the OrderList(Red) is empty, and the algorithm is end. Red is {C₁, C₃}

(2) If the new object y is {1,1,1,2,4}, exists $x_6 \in U'_{NEG}$ for any $C_i \in C$, thus $f(y, C_i) = f(x_6, C_i)$, Red is not changeable according to step2, Red is {C₁, C₃}

(3) If the new object y is {1,1,2,0,1}, exists $C_i \in C$ for any $x \in U'_{POS}$ where $f(y, C_i) \neq f(x, C_i)$, update elements in Mpair according to step3, as in Table 3:

Table 3. Update Order Element Pairs Table

No	x	y	No	x	y
1	1	3	*10	3	6
2	1	4	11	4	5
3	1	5	*12	4	6
*4	1	6	*13	5	6
*5	2	3	14	1	7
*6	2	4	*15	2	7
*7	2	5	16	5	7
9	3	5	*17	6	7

As y is added to the dataset, all condition attributes obtain indistinguishable element pairs, The results of updating part as following:

List(C_1)' is (14,*15,*17),List(C_2)' is (14,*17) and List(C_3)' is (14).

As OrderList(Red) is not empty and the List(C_4) is not updating, Red will be update. Let Red = $\{C_1, C_3, C_4\}$. Remove the redundant attribute according to step3.3, no redundant elements are found. The results of reduction is $\{C_1, C_3, C_4\}$

4.3. Experimental Analysis

In order to further verify the performance of the increment updating algorithm in this paper and other similar methods, we use programming environment of Windows 7, with development tools of visual studio2008, the CPU 2.3 G, and 4 GB memory. In this paper, six data sets were all downloaded from UCI repository of machine learning databases. The data sets are outlined in Table 4. Data sets name is respectively Tae(D_1),Monks(D_2),Pima(D_3),Car(D_4),Chess(D_5) and Mushroom(D_6). $|U|$ is the number of instances, $|U'|$ is the number of simplified decision table instances, $|C|$ is the number of attribute, $|B|$ is the number of basic data, Datatype is data type, $|Cate|$ is the number of Categorical attribute, $|Rea|$ is the number of real attribute and $|Int|$ is the number of Integer attribute. We take out 80% of the data from each data set as the original data, the rest part as incremental.

In experiment, we use the six data sets from UCI and conduct five times, then calculate average. Compare with the similar incremental updating algorithm of literature [13-14] and this paper, the results as shown in Table 5.

Table 4. Test Dataset

Data	$ U $	$ U' $	$ C $	$ B $	DataType		
					$ Cate $	$ Rea $	$ Int $
D_1	151	151	5	120	0	0	5
D_2	432	432	6	345	6	0	0
D_3	768	768	8	590	0	2	6
D_4	1728	972	6	1382	6	0	0
D_5	3196	3196	36	2556	36	0	0
D_6	8124	8124	22	6499	22	0	0

Table 5. Comparison of the Three Incremental Updating Algorithms

Data	Algorithm A			Algorithm B			Algorithm 3		
	T_1	S_1	R_1	T_2	S_2	R_2	T_3	S_3	R_3
D_1	0.51	0.55	2	0.29	0.12	2	0.29	0.03	2
D_2	1.19	6.44	3	0.48	1.11	3	0.39	0.21	3
D_3	2.25	36.21	5	0.61	4.71	6	0.49	0.73	5
D_4	3.12	102.63	5	0.63	5.53	5	0.52	1.02	5
D_5	8.36	138.58	29	2.52	358.58	28	2.05	17.64	29
D_6	Overflow			3.01	1392.22	6	2.76	70.44	6

The Table 5 shows that the above three algorithms are dynamic updating algorithm based on positive region model and the reduction results of the above six data sets are basically identical. As for time, the smaller the data, the less obvious the effect. But, when the dimension and the amount of data expand, its effect changes greatly.

Although time complexity of this paper and literature [14] are similar, algorithm 3 of this paper takes the shortest indistinguishable string each time, and the result is pruned effectively, accelerate the speed of searching important attributes.

Since that the time complexity of literature[10] is $O(|C|^2|U|^2)$, and that of this paper and literature[11] is $\max(O(|C||U'_{pos}||U'|), O(|Red|^2|U'_{pos}||U'|))$, and it thus has a close relation to $|U_1|$. The length of $|U_1|$ affects largely the computation time of the literature [14] and this paper, but has little impact on literature [13].

In data sets above, U' is not equal U only to $Car(D_4)$. The data volume of data set D_4 is much bigger than D_3 . But judged from the computation time, it is obvious that literature [13] has less effects than literature [14] and this paper. So literature [14] and this paper are more suitable to reducing data sets with more repeated samples.

Meanwhile, seen from the time effect, literature [14] has greater improvement than literature [13]. This paper algorithm 3 can save more space resources than literature [14], for it we can compute attribute reduction by taking out attribute of data set one by one, and time complexity is $|U'||U'_{pos}|$. When storing information matrix, it just needs $\frac{1}{|C|}$ of the space in literature [14].

As literature [13] requires space $O(|C|^2|U|^2)$, when computing data set D_6 , the memory overflows and the reduction can not be achieved, while literature[14] and this paper can still be reduced, and the reducing results are accordingly the same.

5. Conclusion

When new data joins the decision table, the attribute reductions will change accordingly. To get a relatively accurate attribute reduction sets needs a large number of repeated calculations and thus leads to low efficiency of computation. Based on positive region, the definition of element pair is proposed. It is used to identify the two objects for comparison. All elements are numbered, indistinguishable degrees of each condition attribute can be based on the length of the number elements. The speed of searching will accelerate as the shortest string is selected every time. But other similar algorithms are based on the longest distinguishable element pairs, and thus can't effectively prune. The next step of the research will be how to decide the data set into n units, and to assign each units calculated by one processor, and thus, to process all units synchronically, so as to adapt to the modern multi-processor system.

Acknowledgments

This work is supported by the Postdoctoral Science Foundation of China 2014M551565, by the national natural science foundation of China 60573174, by the natural science foundation of Anhui Province KJ2013Z231.

References

- [1] Z. Pawlak, "Rough sets", International Journal of Computer and Information Science, vol. 11, no. 5, (1982), pp. 341-356.
- [2] Q. Yuhua, L. Jiye and W. Pedrycz, "An efficient accelerator for attribute reduction from incomplete data in rough set framework", Pattern Recognition, vol. 44, no. 8, (2011), pp. 1658-1670.
- [3] C. Jinkun and L. Jinjin, "An Application of Rough Sets to Graph Theory", Information Sciences, vol. 201, no. 19, (2012), pp. 114-127.
- [4] W. Xizhao, W. Tingting and Z. Junhai, "An attribute reduction algorithm based on instance selection", Journal of Computer Research and Development, vol. 49, no. 11, (2012), pp. 2305-2310.

- [5] X. Weihua, L. Yuan and L. Xiuwu, "Approaches to attribute reductions based on rough set and matrix computation in inconsistent ordered information systems", Knowledge Based System, vol. 27, no. 1, (2012), pp. 78-91.
- [6] H. Qiang, W. Congxin and C. Degang, "Fuzzy rough set based attribute reduction for information system with fuzzy decisions", Knowledge Based System, vol. 24, no. 5, (2011), pp. 689-696.
- [7] Y. Chuanjian, G. Hao and L. Longshu, "Attribute reduction based on logical operation of Boolean discernibility matrix", Journal of Sichuan University (Engineering science edition), vol. 44, no. 2, (2012), pp. 76-82.
- [8] L. Baohua, W. Shiyi and C. Ming, "Heuristic Attribute Reduction algorithm based on order table", Computer Engineering, vol. 38, no. 2, (2012), pp. 51-53.
- [9] H. Feng, W. Guoyin and H. Huang, "Incremental attribute reduction based on elementary sets", Proceedings of the 10th International conference on Rough Sets, Fuzzy sets, Data Mining and Granular Computing Regina, Canada, (2005), pp. 185-193.
- [10] G. Lihe and W. Guoyin, "An incremental updating algorithm for attribute reduction set of decision tables", Journal of Frontiers of Computer Science & Technology, vol. 4, no. 5, (2012), pp. 436-44.
- [11] L. Yang, F. Boqin and Z. Jiangwei, "Complete Algorithm of increment for attribute reduction based on discernibility matrix", Journal of Xian Jiao tong University, vol. 4, no. 2, (2007), pp. 158-161.
- [12] X. Zhangyan, L. Zuopeng and Y Binru, "A Quick Attribute Reduction Algorithm with Complexity of $Max(O(|C|U), O(|C|^2|U|C|))$ ", Chinese Journal of Computers, vol. 29, no. 3, (2006), pp. 391-399(in Chinese).
- [13] L. Xiaowei, "The matrix approaches of rough sets", Computing Appl., vol. 42, no. 17, (2006), pp. 73.
- [14] Q. Wenbin, Y Binru and X. Zhangyan, "Efficient algorithm for dynamic attribute reduction based on a matrix", Journal of University of Science and Technology Beijing, vol. 35, no. 2, (2013), pp. 249-255.
- [15] W. Guoyin, "Rough Set Theory and Knowledge Acquisition", Xi'an Jiaotong University Press, Xi'an, (2001).

Author



Bao-Hua Liang, he received the Master's degree in Engineering from the GuangXi Normal University, Guilin, China. He is currently an associate professor at the Institute of Information Engineering, Chaohu University.