7659, England, UK
Journal of Information and Computing Science
Vol. 2, No. 3, 2007, *pp*. 223-227

A New Reduction Implementation Based on Concept

Tinghuai Ma [1 +], Meili Tang [2]

[1]Computer Dept., Nanjing University of Information & Science Technology, Nanjing 210044, P. R. China
[2]Public Administration Dept., Nanjing Univ. of Information & Science Technology, Nanjing 210044, China

(*Received 24 July 2006, Accepted 5 October 2006*)

Abstract. Rough set is one of the most useful data mining techniques. How to use rough set to extract rule is the basement of rough set's application. This paper discuss an algorithm that be used in attribute reduction. To attribute reduction, generally method is based on discernibility matrix or its improvement. But this series methods usually get one reduction, can't accommodate uncertain information reasoning. We provide a reduction algorithm, which based on reduction pruning. It can calculate all reductions, and suits any uncertain knowledge reasoning. For increase this algorithm's effective, we present two theorems to make algorithm simplified. We calculate reduction through rough reduction (reduction pruning) and backward elimination two steps. The case illustrates we get the reduction effectively through this algorithm.

Keywords: rough set, reduction, pruning, backward elimination

1. Introduction

Rough set is a mathematical tool to depict knowledge's uncertain and not integrated. Rough set theory provides a mathematical method to deal with data's classification[1]. According to rough set, knowledge system is composed of condition and decision attributes. Rough set just analysis the truth hiding behind the data. In rough set analysis, the men's fuzzy won't be brought to knowledge. The rough set is a smart mathematical tool to analysis imprecise system.

As a new data mining algorithm, rough set has it's own characters:

Knowledge granularity Knowledge granularity has been considered as the reason as not exact describe concept. The knowledge granularity's size can be measured by indiscernibility relationship. The knowledge granularity's size is smaller, the concept's description is clearer. The knowledge granularity's size is bigger; the concept's description is more vagueness. Based on indiscernibility relation, lower Approximation and upper Approximation are defined.

New subject relation This is different from traditional set theory and fuzzy set theory. In traditional set theory, an element will be belonging to this group or not. In fuzzy set theory, the element should be assigned membership value. Rough set membership function is resided in [0, 1]. This value is calculated accurately. It just related to known data, not take into account of men's owner views

Concept boundary What's the concept boundary? The concept won't be express clearly for knowledge granularity is concept boundary. We call this vague concept. Vague concept can be depicted by upper and lower approximation. The elements in upper approximation group may be elements in concept; the elements in lower approximation group must be elements in concept. Those elements are concept boundary, which are in lower approximation group but not in upper approximation group. The concept boundary embodies concept's vagueness.

Knowledge reduction Attribution reduction and value reduction are presented in rough set. While not decrease the precision of classification, reduction can simplify data and keep the key characters. Thus, it will get the knowledge's simplest expression. Reduction can also recognize and evaluate data's relationship. It

+ Corresponding author. *Email- address*: thma@nuist.edu.cn

will post simple expression of knowledge. So, rough set can extract rule and reason based on not integrated datum.

2. Related concepts

A classic knowledge system can be represented as: $S =< U, C, D >$, where U is a finite set of object, we call it complete domain, the elements in U are called objects or instances, C is condition attribute set, D is decision attribute set, $A = C \bigcup D$, call A is set of attribute.

Relative redundancy attribute Knowledge system S=< U, A >, $A = C \cup D$, $C \cap D = \phi$, $P \subseteq C$, $Q \subseteq D$, $a \in P$. If $POS_P(Q) = POS_{P-\{a\}}(Q)$, call a is redundancy, which P relative Q. Otherwise, call a is necessary, which P relative Q.

Relative cross attribute set Knowledge system S=< U, A >, $A = C \cup D$, $C \cap D = \phi$, $P \subseteq C$, $Q \subseteq D$, $a \in P$. While arbitrary attribute $a \in P$ is necessary, which P relative Q, call P is cross relative to Q. Otherwise, call P is depend on relative to Q

Relative reduction Knowledge system S=< U, A >, $A = C \cup D$, $C \cap D = \phi$, $P \subseteq C$, $Q \subseteq D$, if satisfied follows:

（1） $POS_P(Q) = POS_C(Q)$;

（2） P is cross relative to Q.

Call P is the reduction of C relative to Q, noted as $RED_Q(C)$.

According to definition of reduction, reduction is an iterative process. If P is reduction of C relative to Q, it should satisfy two condition. First, $POS_P(Q) = POS_C(Q)$. Second, P should be cross relative to Q. It has been proved that computing the reduction of knowledge is a NP problem. If we want get whole reduction sets based on reduction concept, the algorithm's complexity is O(|A|^2|U|^{log}|U|). In which, |A| is the attribute number, |U| is the record number. So, find out all reduction set is a difficult work.

In actual, reduction calculation is not based on reduction concept. The actual algorithm always base on discernibility matrix as follow process: (1). Construct discernibility matrix; (2). Construct discernibility function based on discernibility matrix; (3). Using absorb rule to simplify discernibility function, getting disjunction norm, and then every implication norm is a reduction.

Discernibility matrix's size is |U|$\times$|U|, and great matrix's calculation complexity is a big problem. So, avoiding discernibility matrix calculating, a heuristic algorithm is provided [2, 3]. Especially, heuristic algorithm is based on attribute's importance. Its principal is calculating positive area. The origin of reduction is core. The origin of core may be NULL. Every step, add most important attribute to the core, until the core is the reduction.

3. Reduction Pruning

But as mentioned as before, in discernibility matrix algorithm and heuristic algorithm, only one reduction set can be got. As we know, one system's reduction will be not uniquely. If only get one reduction P, P can represent attribute set A. A new object hasn't integrated attributes A', $P \not\subset A'$. So, this case can't be classified correctly. If this knowledge has another reduction P' ($P' \subset A'$), this case can be classified correctly through reduction P'. So, choosing only one reduction maybe in a dilemma. An algorithm that can get all reduction set is urgently needed.

According to reduction concept, reduction P can be arbitrary combination of condition attribute $c \in C$. Thus, the reduction number can be $C_{|C|}^1 + C_{|C|}^2 + \cdots C_{|C|}^{|C|} = 2^{|C|} - 1$, in which $|C|$ is the number of condition attribute. There are $2^{|C|} - 1$ candidate reductions need to be verified. While $|C|$ is increment, the verification is an impossible work.

So, algorithm that can get all reductions and be effectiveness will generate.

3.1. Rough reduction

Definition 1 (rough reduction). Knowledge system S=< U, A >, $A = C \cup D$, $C \cap D = \phi$, $P \subseteq C$, $Q \subseteq D$, if satisfied $POS_P(Q) = POS_C(Q)$,Call P is the rough reduction of C relative to Q.

From the definition, compared with Relative reduction definition, rough reduction satisfied first condition, not satisfy second condition. Thus, rough reduction includes reduction and reduction's super set. We can use frequent item pruning [4, 5] concept to advisor reduction calculation.

Theorem 1 (reduction pruning). If there is any set has i elements, which is not the rough reduction of C; its subset that has $i-1$ elements should not be rough reduction of C.

It means: if i =3 attributes combination set C^i ={a, b ,c} is not the rough reduction of C, i =2 attributes combination set {a, b}, {b, c}, {a, c} are not the rough reduction of C.

Prove:

1. According to rough set theory, we know, the classification precision is depending on the size of equivalence set. The size of equivalence set is mapping to knowledge granularity.
2. P is the rough reduction of C relative D means P and C have the same knowledge granularity relative D. In the same words, P is the rough reduction of C relative to D, because the knowledge granularity of P is not greater than the knowledge granularity of C.
3. The knowledge granularity of C^{i-1} attribute set is greater than the knowledge granularity of C^i attribute set.
4. C^i attribute set is not the rough reduction of C, the knowledge granularity of C^i attribute set is greater than the knowledge granularity of C.
5. The knowledge granularity of C^{i-1} attribute set is greater than the knowledge granularity of C^i attribute set of course, and greater than the knowledge granularity of C. So, C^{i-1} attribute set is not the rough reduction of condition attribute C.

Theorem 1 implies: if we want to calculate all rough reduction based on rough reduction's concept, we can take C_n^{n-1} attribute set as the origin of rough reduction, in which, n is the number of condition attribute. If attribute set A_i ($A_i \subset C_n^{n-1}$) is not the rough reduction of C, all attribute set C_n^{n-2} that based on A_i ($A_i \subset C_n^{n-1}$) won't be rough reduction of C, and we needn't to test these attribute sets. Thus, we can decrease the candidate reduction. We use *German* dataset to illustrate algorithm effectiveness, which located at *http://www.cs.cmu.edu/afs/cs/project/ai-epository/ai/areas/learning/database/uci_mldb*. This dataset has 20 condition attributes. If we test all candidate reduction sets, we should test ($2^{20}-1$) candidates. Use above theorem, we just test 1687 candidate reduction sets. $(2^{20}-1)/1687 = 621$. We decrease the computing time sharply.

In rough reduction definition, rough reduction's bottleneck is estimation whether $POS_P(Q) = POS_C(Q)$. This need traversing all elements of set $POS_P(Q)$ and $POS_C(Q)$. We find a method to simply this estimation, as followed.

Theorem 2. $B \subset C$, if two set satisfied $POS_B(D) \neq POS_C(D)$, $|POS_B(D)| \neq |POS_C(D)|$.

$|POS_B(D)|$ is the number of set $POS_B(D)$.

Prove: While the number of elements in sets is different, the sets are different absolutely. We need prove while two sets $POS_B(D)$, $POS_C(D)$ are different, the number of elements in these two sets won't be same.

1. As $B \subset C$, $BN_B(D) \geq BN_C(D)$. Knowledge granularity of B is greater than C, so, vagueness of B relative to D is greater than C relative to D.
2. In the same U domain, we know, $U=U_+(B)+U_-(B)+BN_B(D)$. According to step 1, there is $U_+(B)+U_-(B) \leqslant U_+(C)+U_-(C)$, in which $U_+(B)$ represents B's positive domain, $U_-(B)$ represents B's negative domain. Under $B \subset C$, $U_-(B) \geqslant U_-(C)$, so $U_+(B) \leqslant U_+(C)$ is tenable.
3. if satisfy $POS_B(D) \neq POS_C(D)$, in $U_+(B) \leqslant U_+(C)$ instance, it's $|POS_B(D)| \neq |POS_C(D)|$

So, we estimate whether two sets' element number is equal instead of estimate whether two set's element is equal. Thus, the estimate complexity is simplified.

Here, rough reduction calculating begin with sets that has (n-1) elements, every step, sets' element number decrease 1, until there are no rough reduction candidate. Rough reduction generate sequence is from bigger attribute sets to smaller attribute sets.

3.2. Reduction generating

As mention before, the algorithm is focusing on how to generate rough reduction. Generally, rough reduction will be much more than reduction. Especially, while condition attribute number is big, this scenario will be more serious. When we get a great deal of rough reduction, we should find out actually reduction. This step will obey reduction definition's second condition, cross attribute condition.

Search reduction from rough reduction, we use a method called **backward elimination method**.

Its principle is: according to rough reductions' generate process, we know, rough reduction's length that generate later must not greater than that generate former. The rough reduction generated later is close to minimal reduction that its element number is minimal. We concern about this minimal reduction. Those rough reductions that have least elements are absolutely actual reduction. So, we can choose rough reduction that has least elements as minimal reduction. Choosing a minimal reduction set R, search backward, if find a rough reduction $R^{'}$, $R \subseteq R^{'}$, we delete the rough reduction $R^{'}$ from the whole rough reduction sets. We call this backward elimination method. Repeat choosing minimal reduction to backward elimination until all minimal reduction chose. The remnant of rough reduction set is reduction.

3.3. Algorithm of reduction

The algorithm as analyzed former has two steps. First calculating rough reduction and second backward elimination, at last get the reduction.

Reduction pruning algorithm

Input: Knowledge

Output: Reduction set

1. Calculate decision attribute's equivalent set;
2. Calculate condition attribute's equivalent set;
3. Calculate knowledge system's positive domain;
4. Repeat
5. For every candidate (i) in candidate
6. Calculate equivalent set of candidate (i);
7. Calculate positive domain of candidate (i) relative to decision attribute;

a) If (positive domain of candidate (i) = positive domain of knowledge)

   i. TestFore = TestFore + candidate(i);

   ii. Redution = Redution + candidate(i);

b) End if

8. End for
9. candidate=junction of two sets that in TestFore (junction sets' element number = TestFore sets' element number –1);
10. Until candidate is null
11. Rough reduction backward elimination

This algorithm cost most time on testing the rough reduction. The worst case is every attribute is reduction, so total candidate sets is $C_n^{n-1} + C_n^{n-2} + \cdots C_n^1 = 2^n - 2$. The backward elimination cost least time. The worst time complexity is $O(2^n)$. Actually, the test sets is much less than $2^n - 2$.

The rough reduction generating GUI is shown in Fig 1.

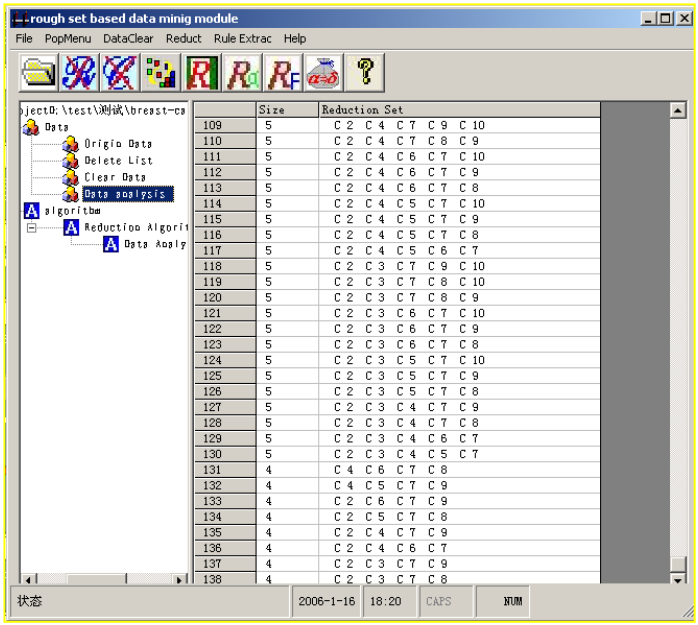

Fig. 1. The result of rough reduction algorithm. The right shows the rough reduction that got by rough reduction algorithm. The number of middle means the candidate rough reduction that was been estimated.

4. Conclusion

In this paper, we presented an algorithm of attribute reduction. This algorithm cost time is tolerable and is efficient. It avoids discernibility methods causing some unintegrated case can't be classified correctly. We presented two theorems to simplify algorithm's complexity. Same to reduction's definition, rough reduction is presented. Reduction pruning is first provided. We divided this algorithm into two steps: one is rough reduction and reduction pruning, second is backward elimination. The ase illustrates our algorithm is effectiveness.

5. Acknowledgements

Our work is partly supported by Science Research Start-up Foundation from Nanjing University of Information & Science Technology. The authors acknowledge all the colleagues for their valuable comments.

6. Reference

[1] Tinghuai Ma. Study on Rough Set Theory based Data Mining Method. Doctoral dissertation, Chinese Academy of Science, China. 2003.

[2] X. Hu. Knowledge Discovery in Databases: an Attribute-oriented Rough Set Approach. Doctoral dissertation, University of Regina, Canada. 1995.

[3] Jing-Kai Liang, Yang Zhang, Yan-Bin Qu. A Heuristic Algorithm of Attribute Reduction in Rough Set. Proceedings of 2005 International Conference on Machine Learning and Cybernetics. 2005, 3140 – 3142.

[4] Songmao Zhang and Georey I. Webb. Further Pruning for E.cient Association Rule Discovery. AI 2001, LNAI 2256, 2001, .605 – 618.

[5] Li Yang. Pruning and Visualizing Generalized Association Rules in Parallel Coordinates. IEEE Transactions on Knowledge and Data Engineering. **17**(1): 60-70.