Value reduction in rough sets based on Apriori algorithm

Yuliang Ma, Zhizeng Luo

School of Automation, Hangzhou Dianzi University, Hangzhou, 310018
E-mail: mayuliang@hdu.edu.cn

Abstract: Aiming at value reduction, a kind of RSVR algorithm was presented based on support in association rules via Apriori algorithm. A more effective reduction table can be obtained by deleting those rules with less support according to least support---minsup. The reduction feasibility of this algorithm was achieved by reducing the given decision table. Testing by UCI machine learning database and comparing this algorithm with least value reduction algorithm indicate the validity of RSVR algorithm.

Key Words: Association rules; Value reduction; Support; Apriori algorithm; Rough sets

1 INTRODUCTION

Rough sets theory (RST) brought forward by Polish scientist Pawlak Z. in 1982 is a new mathematical tool for fuzzy and uncertain knowledge. In this theory knowledge is regarded as partition of the Universe by defining the knowledge from a new angle of view. Knowledge is discussed by equivalence relation in algebra. RST has been successfully applied in such fields as artificial intelligence, knowledge discovery, data mining, pattern recognition and fault diagnose in the recent 20 years [1-4]. RST is very suitable for data analysis because of its intrinsic characteristics.

At present the reduction algorithm always focuses on reducing attributes and aims at obtaining the best attributes reduction, although in practice attributes reduction is not especially important because we only need the satisfactory value reduction [5]. Potential knowledge contained in data is always targeted when we analyze the database. The complexity of the information system can be reduced by attributes reduction, although not all attribute values of each rule are not necessary in the reduced information table, so the value reduction of information or decision table is needed. Value reduction is a process of deleting all redundant values of condition attributes that have no influence on expression rule [6].

It is now proved theoretically that obtaining value reduction of objects of interest is an NP-hard problem and that it is difficult to obtain minimal value reduction by enumeration. In this paper a new rough sets value reduction (RSVR) algorithm is presented via the concept of support in association rules by combining association rules mining with RST based on literatures [7,8]. Perfect reduction results of the given decision table were obtained by this algorithm and the advantages can be seen by comparing this algorithm with least value reduction algorithm.

2 THEORETICAL FOUNDATION

There is a mature theory of rough sets via more than 20 years development and the basic concepts of RST can be consulted in literature [9]. This section mainly introduces concepts such as support, reduction ratio and so on.

Agrawal and Srikant [10] put forward Apriori algorithm, which can compress greatly candidate sets. The concepts can be defined as follows via the support concept in association rules.

Definition 1 In the decision table, t and s are condition and decision attributes respectively. The cardinality card($t \Rightarrow s$) of rule $t \Rightarrow s$ is called support of rule, which is marked as sup($t \Rightarrow s$). The cardinality card(t) of attribute t is called support of t, which is marked as sup(t).

Definition 2 If the support sup($t \Rightarrow s$) of rule $t \Rightarrow s$ satisfies sup($t \Rightarrow s$)=sup(t), then the rule is called determinate rule; if the support of a determinate rule is greater than the least support minsup appointed by user, then the rule is strong determinate rule.

The concepts of reduction ratio here come from literature [11]:

Definition 3 Let the number of condition attributes of the initial database be Na, the number of reduced attributes be Nc, then the attribute reduction ratio is:

$$E_a = (1 - N_c / N_a) * 100\%$$

The attribute reduction ratio denotes decrease of involved factors after data reduction.

Definition 4 Let the initial database's number of rules be Ns, the number of reduced rules be Nr, then the rule reduction ratio is:

$$E_i = (1 - N_r / N_s) * 100\%$$

The rule reduction ratio denotes decrease of rules in a given database.

This work is supported by Natural Science Foundation of Zhejiang Province under Grant Y1080854 and National Hi-Tech Research and Development Program (863) of China under Grant 2008AA04Z212.

Definition 5 Let the data quantity of initial database be N, the reduced data quantity be M, then the data reduction ratio is:

$$E_w = (1 - M/N) * 100\%$$

The data reduction ratio denotes decrease of information in database.

3 ALGORITHM DESIGN

Based on Apriori algorithm, if the rule $t \Rightarrow s$ is not strong, then the extended rule $t \wedge p \Rightarrow s$ is not strong either. The reduction table is obtained by deleting the rules whose support is less than the least support minsup appointed by user.

The description of the algorithm is as follows:

Input: decision table DT, the least support minsup

Output: rules set Rk

Step 1: Attribute reduction for decision table.

Step 2: Set k as 1.

Step 3: Calculate attribute support and rule support of every attribute in candidate set Ck.

Step 4: Delete the rules from Ck if its rule support is less than or equal to the least support minsup, transfer the rule into rule sets Rk if its attribute support is equal to rule support.

Step 5: Expand Ck into Ck+1. Scan Ck first, combine every two items in Ck into candidate item with k+1 attributes and insert the candidate item into Ck+1. Then check every item C in Ck+1, if an item is in k-subset of C but not in Ck, delete C; if C is antipathic, delete C too. Finally obtain Ck+1 and set k as k+1.

Step 6: Repeat steps 3 to 5 until Ck is empty.

Step 7: End.

4 EXAMPLE OF ALGORITHM

In the original decision Table 1, U is the concerned universe, a, b, c, d are condition attributes, e is decision attribute. The least support is *minsup*=1.

Reduce the decision table according to the algorithm presented in the above section:

Attribute reduction. Only attribute a is e-omissible in the original table, so delete attribute a to form a new decision table.

Value reduction. Calculate attribute support and rule support of every attribute to form candidate set C_1 with 1 condition attribute. Check every item in the new table, if rule support of an item is less than or equal to the least support *minsup*, delete the item from C_1; if attribute support of an item is equal to rule support, transfer the item to rule set R as determinate rule.

Form candidate set C_2 with 2 condition attributes. Combine two items that have the same decision attribute in C_1 to form new item with 2 condition attributes by extending C_1. Then deal with them according to the above method.

Form candidate set C_3 with 3 condition attributes.

When candidate set C_4 with 4 condition attributes is empty, stop the algorithm.

At last we get the reduced decision Table 2 yielding the following rules:

(1) $(d,1) \Rightarrow (e,3)$

(2) $(b,2) \wedge (c,2) \Rightarrow (e,3)$

(3) $(b,1) \wedge (c,1) \Rightarrow (e,3)$

(4) $(b,2) \wedge (c,1) \wedge (d,2) \Rightarrow (e,2)$

(5) $(b,1) \wedge (c,2) \wedge (d,2) \Rightarrow (e,1)$

Table 1 The original decision table

U	a	b	c	d	e
1	1	2	1	1	3
2	1	2	2	1	3
3	1	2	1	2	2
4	1	1	1	1	3
5	2	1	2	2	1
6	2	2	2	1	3
7	2	2	2	2	3
8	2	1	2	1	3
9	2	1	1	1	3
10	3	2	2	2	3
11	3	2	1	2	2
12	3	1	1	2	3
13	3	1	1	1	3
14	3	2	2	1	3
15	3	1	2	2	1
16	3	1	2	1	3

Table 2 The reduced decision table

U	b	c	d	e	Rule support
1	—	—	1	3	9
2	2	2	—	3	5
3	1	1	—	3	4
4	2	1	2	2	2
5	1	2	2	1	2

5 COMPARISON OF ALGORITHMS

The 8 discrete datasets in UCI machine learning database are used to test this algorithm and the least value reduction algorithm is used for comparison. Let *minsup*=2 and *minsup*=3 in the two algorithms with the reduction results listed in Table 3 and Table 4, where only rule reduction ratio and data reduction ratio are listed because the attribute reduction of the two algorithms is the same.

Generally speaking, the satisfactory values of E_a, E_i, E_w are E_a>30%, E_i>60%, E_w>85% respectively.

Table 3 Reduction results when minsup=2

Data sets	Number of objects	Number of attributes	RSVR algorithm			Least value reduction algorithm		
			Rule reduction ratio	Data reduction ratio	Runtime (s)	Rule reduction ratio	Data reduction ratio	Runtime (s)
*monk1	124	7	87.9%	93.7%	0.011	86.3%	92.9%	0.01
monk3	122	7	77%	88.5%	0.019	81.1%	90.6%	0.011
mux6	64	7	71.9%	83.5%	0.084	84.4%	93.3%	0.004
*led7	200	8	93.5%	96.1%	0.677	61.5%	68.6%	0.04
*parity5+5	100	11	81%	89.6%	0.072	64%	81.7%	0.018
iris-disc	100	5	88%	93.4%	0.002	94%	96.4%	0.003

Table 4 Reduction results when minsup=3

Data sets	Number of objects	Number of attributes	RSVR algorithm			Least value reduction algorithm		
			Rule reduction ratio	Data reduction ratio	Runtime (s)	Rule reduction ratio	Data reduction ratio	Runtime (s)
*monk1	124	7	89.5%	94.6%	0.009	86.3%	92.9%	0.01
*monk3	122	7	82%	91.3%	0.015	81.1%	90.6%	0.01
mux6	64	7	71.9%	83.5%	0.086	84.4%	93.3%	0.004
*led7	200	8	94%	96.4%	0.458	61.5%	68.6%	0.056
*parity5+5	100	11	88%	93.5%	0.062	64%	81.7%	0.018
iris-disc	100	5	89%	94%	0.002	94%	96.4%	0.003

By comparing the two algorithms, we can see from Table 3 and Table 4 that:

a) The rule reduction ratio and data reduction ratio of 6 datasets in RSVR algorithm are greater than those in least value reduction when *minsup*=3 (denoted by "*" in front of dataset). When *minsup*=2 there are 5 datasets only. It shows that the reduction ratio increases with improvement of *minsup* value. The reduction ratio of RSVR algorithm must be greater than that of least value reduction algorithm if the value of *minsup* is increased.

b) RSVR algorithm reserves all rules being useful for users. The algorithm mainly focuses on applied system instead of on reduction ratio.

c) The runtime of RSVR algorithm is commonly longer than that of least value reduction algorithm, especially when the quantity of objects and attributes is large, because RSVR algorithm adopts times-iterative method and complicated structure database.

6 CONCLUSION

This paper presents an RSVR algorithm based on support in association rules mining via Apriori algorithm. The more effective reduction table can be obtained by deleting those rules with less support according to least support minsup. The reduction feasibility of this algorithm was achieved by reducing the given decision table. Comparing this algorithm with least value reduction algorithm reveals the characteristics and advantages of RSVR. Testing by UCI machine learning database showed the validity and feasibility of this algorithm.

REFERENCES

[1] [1] Pan J.L., Ye X.H. Wang H.X. Node Fault Diagnosis in WSN Based on the Rough set and Bayes Decision-Making. Chinese Journal of Sensors and Actuators, 2009, 22(5): 734-738.

[2] [2] Zhang Z.Y., Yuan R.X., Yang T.Z. Rule Extraction for Power System Fault Diagnosis Based on the Combination of Rough Sets and Niche Genetic Algorithm. Transactions of China Electrotechnical Society, 2009, 24(1): 158-163.

[3] [3] Zhou X.S., Wang Z.M. Application of Rough Set and Neural Network in Data Mining. Computer Engineering and Applications, 2009, 45(7): 146-149.

[4] [4] Wang H. Customer Value Analysis Based on Rough Set and Data Mining Technique. 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008: 1-4.

[5] [5] Jiang W.J., Xu Y.H., Xu Y.S. Research on the Nature of Reduction to Simplifying Reduction Algorithm. Proceedings of the Fourth International Conference on Machine Learning and Cybernatics, Guangzhou, 2005: 1800-1805.

[6] [6] Yang Z.F., Guo J.F., Chang F. A Value Reduction Method Based on Rough Sets. Computer Engineering, 2003, 29(9): 96-97.

[7] [7] Agrawal R., Imielinski T., Swami A. Mining Association Rules between Sets of Items in Large Databases. Proceedings of the ACM SIGMOD Conference on Management of Data. 1993: 207-216.

[8] [8] Lin T.Y. Rough Set Theory in Very Large Databases. Proceedings of CESA'96. Lille, 1996: 936-941.

[9] [9] Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Boston. 1991.

[10] [10] Agrawal R., Srikant, R. Fast Algorithms for Mining Association Rules. Proceedings of the 20th VLDB Conference, Santiago, 1994: 487-499.

[11] [11] Wang J., Wang R., Miao D.Q. Data Condensation Based on Rough Set Theory. Computer Transaction, 1998, 21(5): 393-400.