International Journal of Hybrid Information Technology
Vol.8, No.9 (2015), pp.339-352
http://dx.doi.org/10.14257/ijhit.2015.8.9.31

A Dual-Reduct Approach to Generate Core Rules

Lv Hanfei and Jiang Congfeng

Department of Information and Management
Zhejiang Police Vocational Academy, Hangzhou, Zhejiang Province, China
lhf_b@sina.com
School of Computer Science and Technology
Hangzhou Dianzi University, Hangzhou, Zhejiang Province, China
cjiang@hdu.edu.cn

Abstract

In this paper, we propose a Dual-Reduct method to generate core rules from original data sets for decision making. We rank the rules by rule usefulness after the step of first reduct. Then we take the useful rules as condition attribute and construct another new decision table. After the step of second reduct we generate core rules from the new constructed decision table. In our approach the generation process is straightforward and objective. At the same time, our approach can significantly reduce the number of rules comparing to the traditional generation approach because we adopt rule usefulness as a measure of core rules. We also provide theoretical proofs and deductions. Our approach is proved to be feasible and effective in a production security system.

Keywords: *Dual-Reduct, Core Rules, Computer Information Security, Prisoners' Reform*

1. Introduction

Pawlak first introduced rough sets theory [1] in 1980s. Rough set theory had been widely applied to attribute selection, knowledge discovery and rule discovery in the databases such as data mining, machine learning and decision make field. The decision table consists of the condition attributes and decision attributes.

A reduct of a decision table is a set of condition attributes which is sufficient to define the decision attributes. A reduct does not contain redundant attributes towards a classification task. It is often used in the attribute selection process to reduce the redundant attributes, and to reduce the computation cost for rule generations. For each decision table there may exist more than one reduct. It has been demonstrated that finding all the reduct sets for a data set is NP-hard. Approximation algorithms are used to obtain reduct sets [10]. The intersection of all the possible reducts is called the core. The core is contained in all the reduct sets, and it is the essential part of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Reduct is one of the most important concepts in rough set theory. A reduct is usually used to represent the whole data set. Reduct is subset of condition attributes. And reduct generation is usually designed to extract core condition attributes from the decision table. The process of generating the core rules will become easier and more effective by considering fewer condition attributes.

However, this method is not diffusely used to get the association rules. We notice one main problem of association rule algorithm. There are usually too many rules generated. It is a huge burden because it will cost plenty of time to process a lot of rules. We can choose some rough set method to find the redundant attributes during the data pre-processing process. After removing the redundant attributes, association rules generation will become faster and more effective. The concept of rule templates was introduced by

ISSN: 1738-9968 IJHIT

Klemettinen [2]. Pre-defined rule templates are helpful to generate association rules. And these rules will be used for auto recommender systems [3-4].

We discuss how the rough sets theory can help to get key association rules. We wonder to know how to apply these rules in decision-making applications. Additionally, we attempt to search such rules which have, in a follow-up part, the decision attributes which are helpful for decision-making applications. We use rule usefulness measure [10] based on rough sets to evaluate the effectiveness of the association rules. This method is helpful not only for decision-making applications but also for auto recommender systems.

We also explain the background of association rules algorithm and the rule usefulness concept in Section 2. In Section 3 we show our Dual-Reduct model to generate core rules. And we describe the experimental study of Dual-Reduct method to artificial car data, prison computer security and prisoners' reform in Section 4. Finally in Section 5 we summarize our contributions and discuss how to expand the model application area in the future work.

2. Dual-Reduct Method

2.1. Motivation

We often analyse a lot of rules generated from the data set during rule generation process. But only some of them are interesting and useful for us. It is a huge challenge to find the interesting and useful rules among all the available rules for many researchers. One method raised by Klemettinen is to rank the rules by "rule interestingness measures" [2]. This method ranks the rules by "rule interestingness measure". Those rules with higher measures are regarded more interesting. The rule interestingness measure from a variety of sources is popular to extract interesting rules. Rank the rules by "rule usefulness measures" [10] is another way. We want to give one method to generate the core rules in an easy and effective way, and the rules are ranked by "rule usefulness measure". Thus we introduce the Dual-Reduct method for the purpose of selecting the core rules quickly in this paper.

2.2. The Concept of Association Rules and Rule Usefulness Measure

An association rule algorithm is often applied to select patterns from business transactions for us. For instance, in market basket analysis, by analysing business transaction records from the market, with the help of association rules algorithm we can find various shopping behaviours. Then we can evaluate the association rules and explain such behaviours. Moreover, by placing related goods in the right place for customers, we can increase the sale of the market.

In conclusion, a rule of the form $\alpha \rightarrow \beta$ is called an association rule [5] . Here α and β are different item sets and they do not share common items. The association rule $\alpha \rightarrow \beta$ holds in the transaction set L with confidence c, $c = \frac{|\alpha \cup \beta|}{|\alpha|}$, if the condition c% of transactions in L that contain α also contain β is true. The rule $\alpha \rightarrow \beta$ has support s, $s = \frac{|\alpha \cup \beta|}{|L|}$, if the condition that s% of transactions in L contain $\alpha \cup \beta$ is true. We define α as antecedent and define β as consequent here.

Here we introduce one parameter named confidence. It means a ratio of the number of transactions that the antecedent and the consequent appear at the same time to the number of transactions the antecedent appears independent. Another parameter is named support. We often use it to calculate how often the antecedent and the consequent appear at the same time in the transaction set. We generated so many rules, to select the core rules in a short time is a huge challenge for association rule algorithm. As everyone knows, it is very difficult to analyse all the rules in a short period. The rule interestingness measures [5] have been introduced to reduce the rule numbers generated.

We adopt the method ***Rule Usefulness Measure*** [10] to evaluate the usefulness of rules in this paper. The rule usefulness definition is defined below. We use RU to replace Rule Usefulness Measure [10] in this paper for convenience.

Definition 1. ***If one rule is generated more frequently from different rule sets, this rule is more useful than any other rules.***

The rule usefulness measure is defined as below [10],

Definition 2. Definition 2.***If one rule appeared k times from different rules sets, and reduct sets number is n.***

$$RU = \frac{k}{n} \quad (1)$$

Suppose for a certain decision table T(C, D), for T(C, D), n is the number of reduct, and m is the number of single rule. RuleSets are rule sets generated based on n reduct. And $ruleset_j \in RuleSets$ $(1 \le j \le n)$ represent single rule sets which include the rules generated from reduction. Here $rule_i$ $(1 \le i \le m)$ represent the single rule of RuleSets.

$$RU_i = \frac{|\{ruleset_j \in RuleSets \mid rule_i \in ruleset_j\}|}{n} \quad (2)$$

For example, a, b, c, d are condition attributes, result is decision attribute. There are 4 reduct used for rule generation. They are {a, b, c}, {b, c, d}, {a, c, d} and {a, b, d}. There are 4 single rules in RuleSets here. They are a1 → result1, b2 → result2, c3 → result1, d4 → result2. Thus n equals to 4, and m equals to 4, too.

Table 1: Rules Usefulness Example

Reduct	Rule Sets
{a, b, c}	{a1 → result1, b2 → result2 …}
{b, c, d}	{b2 → result2, c3 → result1, d4 → result2, …}
{a, b, d}	{a1 → result1, c3 → result1, d4 → result2, …}
{a, c, d}	{a1 → result1, d4 → result2, …}

For reduct1, the rule set generated is a1 → result1, b2 → result2; for reduct2, the rule set generated is b2 → result2, c3 → result1, d4 → result2; for reduct3, a1 → result1, c3 → result1, d4 → result2; for reduct4, a1 → result1, d4 → result2. Rule a1 → result1 is generated from the 3 reducts, and its rule importance is 3/4 =0.75. Rule b2 → result2 is generated from the 2 reducts, and its rule importance is 2/4 =0.5. Rule c3 → result1 is generated from the 2 reducts, and its rule importance is 2/4 =0.5. Rule d4 → result2 is generated from the 3 reducts, and its rule importance is 3/4 =0.75.

We collect the data items and construct the original decision table T = (U, C, D) first. In the table $U = \{u_0, u_1, ..., u_{m-1}\}$ stands for the items set; $C = \{c_0, c_1, ..., c_{p-1}\}$ stands for the condition attributes set; D stands for the decision attributes set. We only consider one decision attribute in decision table in this paper and it is the most common situation in our experimental study. Then we do the first reduct and get a set of rules. We use $RC = \{Rule_0, Rule_1, ..., Rule_{n-1}\}$ to represent it. Then we can construct the new decision table.

We construct one new rule decision table and here we use $A_{m\times(n+1)}$ to represent it . The row consists of each item from the original decision table ($u_0, u_1, ..., u_{m-1}$ is the row), and $Rule_0, Rule_1, ..., Rule_{n-1}$ are the columns of this new decision table. Here the decision attribute of new decision table is inherited from the original decision table. We check every rule and find the relevant decision attribute value. We review the original data items one after another. If the antecedent or the consequent of a rule does not appear, we say one rule is not suitable for the item. If the antecedent and the consequent of a rule appear in pairs, we say one rule is suitable for the item. It means whether a rule can properly explain the item or not.

Here for each Rulej (j ∈ [0, ..., n−1]), cell A[i, j] (i ∈ [0, ...,m−1]) equals to 1 if the rule Rulej can be applied to the record ui. Otherwise cell A[i, j] equals to 0. The decision attribute A[i, n] (i ∈ [0, ...,m−1]) remains the same as the original values of the decision attribute in the original decision table. We show the conditions for the value calculation of the new decision table.

A[i, j] = 1, if j < n and we can apply Rulej to ui

A[i, j] = 0, if j < n and we can not apply Rulej to ui

A[i, j] = di, if j = n and di is the corresponding decision attributes for ui where i ∈ [0, ...,m−1], j ∈ [0, ..., n−1].

The following example shows how to construct the new decision table using the above proposed method. Let us consider a decision table as shown in Table II. c1, c2, c3 are the three condition attributes, and D is the decision attributes.

Table 2: Sample Decision Table

c1	c2	c3	D
1	0	1	1
1	1	0	1
0	0	1	0

There are 2 rules generated based on Table II, and R = {Rule0, Rule1} is the rule set. Rule0 means "if c1 = 1, then D = 1"; Rule1 means "if c2 = 1 and c3 = 0, then D = 1". In this example, m = 3 which represent the number of rows in the original decision table; n = 2 which represent the number of rules in the rule set. Then we can construct a new decision table for ranking the useful rules as $A_{3\times3}$, the condition attributes in the new decision table are Rule0 and Rule1, and decision attribute is D, which inherits from the original decision table. According to the sample, for condition attribute Rule0, A[0, 0] = 1 because Rule0 can properly classify the record in the first row in Table II, here A[1, 0] = 1 because Rule0 can properly classify the record in the second row from Table II; but A[2, 0] = 0 because Rule0 cannot be applied to the record in the third row from Table II since c1 = 0 instead of 1. Therefore, the cells from the first column in Table II are calculated as

Rule0
1
1
0

According to the sample, the cells from the second column in Table II are calculated as

Rule1
0
1
0

The original decision attributes inherit from Table II and the two columns for condition attributes are calculated before, we construct the new decision table $A_{3\times3}$ as shown in the following Table III.

Table 3: New Decision Table $A_{3\times3}$

Rule0	Rule1	D
1	0	1
1	1	1
0	0	0

This new decision table is then used as the input decision table for finding useful core rules.

Definition 3. Dual-Reduct Rule Data Set. ***In decision table we do the first reduct to***

generate the rules, then we use the rules to rebuild the decision table and do the second reduct. After this step, we call the data is Dual-Reduct Rule Data Set.

The rules in the Dual-Reduct Data Set are the core rules and it is convenient for us to rank them by rule usefulness.

3. The Dual-Reduct Method to Generate Core Rules

We show the flow process chart of Dual-Reduct method in Figure 1. And we describe the method by studying how to generate core rules from the original decision table of artificial car data set. Then we calculate the prison computer information security and prisoners' reform data set in detail.

3.1. The Dual-Reduct Method Model

We introduce the general model of our experimental study and interpret how we can generate the core rules progressively.

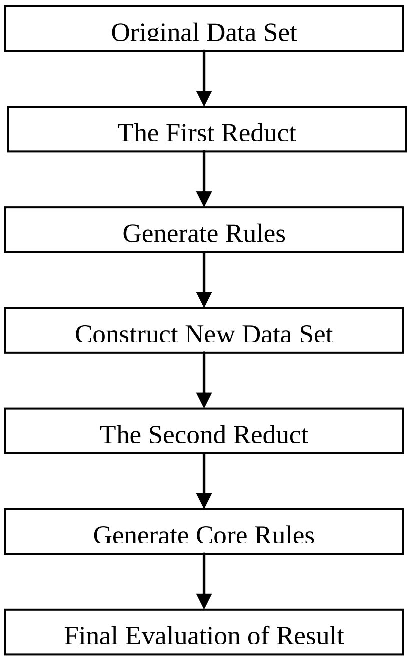


Figure 1: Dual-Reduct Method Model.

3.2. The Method to Generate the Core Rules

Firstly, we do the data pre-processing. The data instances contain the inconsistent data instances and missing attribute values are processed during this step. When two or more data instances contain the same condition attribute values but different decision attribute values in a decision table, we say that inconsistency or corruption exists in this table. These data instances must be cleared from the data set at the beginning. We select some data instances whose condition attribute values are the same but decision attribute values are different. At the beginning such data instances are cleared. In rough set theory some experts raised many discretization algorithms before. For example, equal frequency binning or entropy algorithm [8], can be used during this period. Finally we generate core attributes at the end of data pre-processing step. We must remove the inconsistencies or the corruptions in the data set first to generate the pure core for post-processing.

Secondly, we get various multiple reducts. Experts raised different algorithms which provide generation of multiple reducts. One famous algorithm is ROSETTA's genetic algorithm. It can generate multiple reducts. Another one is RSES [9]. It is one kind of genetic algorithm for user defined number of reducts generation. We usually use it for larger data sets to generate representative reducts.

Thirdly, we should generate the rules. We get multiple reducts first, then we use the condition attributes in the reduct and the decision attributes as the input to generate rules. The rule templates concept was raised by Klemettinen [2]. Pre-defined rule templates are helpful for getting good association rules to be used in auto recommendation systems [3], [4]. Rule templates are used in this step. Depending on all kinds of applications and the anticipated results, rule templates should be defined for desired types of rules and for subsumed rules prior to the rule generation and are used when the rule is in generating step.

For instance, given condition attributes *B1, B2, ... , Bn*, and decision attribute *D*, the following template defines that at the consequent of a rule only decision attributes can appear, and *B1, B2, ..., Bn* lead to a decision of *D*. We can express it in following format:

$$B1;\ B2;\ \ldots\ ;\ Bn \rightarrow D.$$

Definition templates have one advantage. They can remove subsumed rules at the beginning. For instance, given rule *B1; B2 →D*, we can remove the following rules *B1; B2; B3 →D* and *B1; B2; B4 →D*. It is obvious that they are subsumed. We can select the rules only with decision attributes on the consequent part for some decisions. And we usually use these rule templates in our experiment study.

Multiple rule sets are generated for multiple reducts after the rule generations. After the computation of the rule frequencies appearing across all the rule sets, Rule Usefulness Measures are further calculated for every generated rule. We rank the rules with each of their usefulness measures.

Finally, we analyses the results and give an evaluation. Core attributes play an important role for evaluating the ranked rules. Rules with usefulness 1.00 indicate that they contain all of the core attributes. Rules that contain more core attributes are more useful than rules that contain fewer or none core attributes. Among all the condition attributes core attributes are the most representative, and more useful rules contain these more representative attributes, which are the core attributes. Therefore we can evaluate the ranked rules with their rule usefulness by checking for the presence of the core attributes in the rules [10].

3.3. Complex Analyses

We can analyse the time complexity for the proposed Dual-Reduct method of generating core rules. In the original data set we suppose there are P data instances, and there are Q attributes for each data instance. P′ is the number of distinct values in the discernibility matrix [1]. The matrix consists of attributes for calculating the core and the reduct, and t is the number of multiple reducts for the data set. We can compute the time complexity in the worst case.

It costs O(PQ!) [5] to get the apriori association rules, and it costs $O(P'^2)$ [6] to get the first reducts. The core generation takes O(PQ) [7]. In the next step it costs O(PQ!) to get second rule sets for multiple reducts. And it costs O(k log k) to figure up the rule usefulness for the total rules i generated by the multiple rule sets. Therefore in the worst case the time complexity of Dual-Reduct method is limited by $O(P'^2 + PQ + PQ! + k \log k) \approx O(PQ!)$.

4. Experiment of Prison Information Security and Prisoners' Reform

4.1 Example of Artifical Car Data Set

First we use the Artificial Car Data Set as an example to show how our method works. There are 14 effective items for artificial car data set. Then we construct the original decision table. It contains eight condition attributes and one decision attribute. The mileage is the decision attribute. And d stands for it here. For d, {0, 1} means {Mileage is medium, Mileage is high}. And we use $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ to represent {Make Model, Cylinder, Door, Displace, Compress, Power, Transmission, Weight}.

{0, 1} means {the car model is made in USA, The car model is made in Japan} for c_1.
{0, 1} means {There are 6 cylinder in the car, There are 4 cylinder in the car for c_2.
{0, 1} means {There are 2 doors in the car, There are 4 doors in the car} for c_3.
{0, 1} means {Displace of the car is medium, Displace of the car is small} for c_4.
{0, 1} means {Compress of the car is High, Compress of the car is medium} for c_5.
{0, 1, 2} means {Power is High, Power is Medium, Power is Low} for c_6.
{0, 1} means {Auto Transmission, Manual Transmission} for c_7.
{0, 1} means {Weight of the car is Medium, Weight of the car is Light} for c_8.

We show the original data set of Artificial Car Data in Table IV below.

Table 4: Artificial Car Original Data Set

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	d
0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0	0
0	1	0	1	0	1	0	0	0
0	1	0	0	1	1	1	0	0
0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	1	1
1	1	0	1	0	2	1	1	1
1	1	0	0	1	1	1	0	1
1	1	0	1	1	0	1	0	1
1	1	0	1	0	2	1	0	1
1	1	0	1	1	1	1	0	1
0	1	0	1	1	1	1	0	1

We check all of the items in the original data set and find no incomplete or corruption data items here. There are 14 effective data records. Here Johnson's Reduct generation algorithm 2 in ROSETTA [8] is adopted to get reduct on the new decision table.

Table 5: Artificial Car Reduct Set

No	Reduct Set
1	$\{c_1, c_5, c_6, c_7\}$
2	$\{c_1, c_2, c_5, c_7\}$
3	$\{c_1, c_4, c_5, c_7\}$
4	$\{c_1, c_2, c_3, c_4, c_7, c_8\}$

Here in Table V there are 4 reduct data set. Apriori algorithm is adopted to generate 19 rules with *support* = 1% *and confidence* = 100%, and they are shown in Table VI.

Table 6: Rules of Artificial Car

No	Rules
0	$c_1=0 \wedge c_4=0 \wedge c_8=0 \rightarrow d=0$.
1	$c_1=0 \wedge c_5=1 \rightarrow d=0$.

2	$c_1=0 \wedge c_6=2 \rightarrow d=0$.
3	$c_2=0 \rightarrow d=0$.
4	$c_3=1 \rightarrow d=0$.
5	$c_4=0 \wedge c_5=0 \wedge c_8=0 \rightarrow d=0$.
6	$c_4=0 \wedge c_6=2 \rightarrow d=0$.
7	$c_5=1 \wedge c_6=2 \rightarrow d=0$.
8	$c_7=0 \rightarrow d=0$.
9	$c_1=1 \rightarrow d=1$.
10	$c_2=1 \wedge c_4=0 \wedge c_5=0 \rightarrow d=1$.
11	$c_2=1 \wedge c_5=0 \wedge c_6=2 \rightarrow d=1$.
12	$c_4=1 \wedge c_5=1 \rightarrow d=1$.
13	$c_4=1 \wedge c_6=2 \rightarrow d=1$.
14	$c_4=1 \wedge c_7=1 \rightarrow d=1$.
15	$c_4=0 \wedge c_5=0 \wedge c_6=1 \rightarrow d=1$.
16	$c_5=0 \wedge c_7=1 \rightarrow d=1$.
17	$c_6=0 \rightarrow d=1$.
18	$c_8=1 \rightarrow d=1$.

The data is from the traditional rough set example. We use it to show how to get the core rules using our method.

Table 7: Core Rules of Artificial Car

No.	Core Rules Got From Second Reduct	Rule Usefulness
1	$c_1=1 \rightarrow d=1$.	1.0
2	$c_5=0 \wedge c_7=1 \rightarrow d=1$.	0.75

For example, we analyses the rule $c_1=1 \rightarrow d=1$. It means if the car model is made in Japan, the mileage of the car is high. It is easy to understand because the cars made in Japan usually have low fuel consumption. Then we analyses another rule $c_5=0 \wedge c_7=1 \rightarrow d=1$. It means if the compress is high and it is manual transmission, the mileage of the car is high. We know the high compression of the cylinder and manual transmission can save the fuel consumption.

4.2. Prison Computer Information Security Data Set

We got 63 effective items for prison computer information security from one jail. Then we construct the original decision table. It contains eight condition attributes and one decision attribute. The prison computer information security is the decision attribute. And d stands for it here. For d, {0, 1} means {Computer is not safe, Computer is safe}. We use Information Security Evaluation Software Tools to get the information security value of each prison computer. And we use $\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$ to represent {Operation System, CPU, Memory, USB Permission Option, Kill Virus On time Option, Past time after Used, Password Option, Connect to Internet Option }.

{0, 1} means {Window Series(include Window2000, Window NT, Window XP, window 7, etc), Unix Series(include Solaris, AIX, Linux, etc)} for b_1.

{0, 1} means {CPU is AMD series, CPU is Intel series} for b_2.

{0, 1} means {Memory is larger or equal to 1G, Memory is less than 1G} for b_3.

{0, 1} means {USB port is permit to use, USB port is forbidden to use} for b_4.

{0, 1} means {Kill virus on time, Do not kill virus on time} For b_5.

{0, 1, 2} means {Past time after used is less than one year, Past time after used is between one year and three years, Past time after used is more than three years} for b_6.

{0, 1} means {No password or password is weak, password is rough} for b_7.

{0, 1} means {Connect to Internet, Not connect to Internet} for b_8.
We show the original data set of prison computer security in Table VIII below.

Table 8: Prisoner Computer Information Security Original Data Set

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	d
0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0	0
0	1	0	1	0	1	0	0	0
0	1	0	0	1	1	1	0	0
0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	1	1
1	1	0	1	0	2	1	1	1
1	1	0	0	1	1	1	0	1
1	1	0	1	1	0	1	0	1
1	1	0	1	0	2	1	0	1
…	…	…	…	…	…	…	…	…
1	1	0	1	1	1	1	0	1
0	1	0	1	1	1	1	0	1

We check all of the items in the original data set and find no incomplete or corruption data items here. Totally we collect 63 effective data records. Here Johnson's Reduct generation algorithm 2 in ROSETTA [8] is adopted to get reduct on the new decision table.

Table 9: Prison Computer Information Security Product Data Set

No	Reduct Set
1	{ b_1, b_4, b_5, b_7}
2	{ b_1, b_2, b_3, b_4, b_7, b_8}
3	{ b_1, b_5, b_6, b_7}
4	{ b_1, b_2, b_5, b_7}

Here in Table IX there are 4 reduct data set. Apriori algorithm is adopted to generate 23 rules with *support* = 1% *and confidence* = 100%, and they are shown in Table X.

Table 10: Rules of Prison Computer Information Security

No	Rules
0	b_1=1 →d=1.
1	b_1=0 ∧ b_5=1 →d=0.
2	b_1=0 ∧ b_6=2 →d=0.
3	b_2=0 →d=0.
4	b_3=1 →d=0.
5	b_4=0 ∧ b_5=0 ∧ b_8=0 →d=0.
6	b_4=0 ∧ b_6=2 →d=0.
7	b_5=1 ∧ b_6=2 →d=0.
8	b_7=0 →d=0.
9	b_1=0 ∧ b_4=0 ∧ b_8=0 →d =0.
10	b_2=1 ∧ b_4=0 ∧ b_5=0 →d=1.
11	b_2=1 ∧ b_5=0 ∧ b_6=2 →d=1.
12	b_4=1 ∧ b_5=1→d=1.
13	b_4=1 ∧ b_6=2 →d=1.
14	b_4=1 ∧ b_7=1 →d=1.
15	b_4=0 ∧ b_5=0 ∧ b_6=1 →d=1.
1	b_5=0 ∧ b_7=1 →d=1.

6	
.	…
22	b_8=1 →d=1.
23	b_6=0 →d=1.

The prison management policeman informed us that they appreciate our work to generate core rules for prison computer security. The core rules are very useful in improving the prison computer security level.

Table 11: Core Rules of Prison Computer Information Security

No.	Core Rules Got From Second Reduct	Rule Usefulness
1	b_1=1 →d=1.	1.0
2	b_5=0 ∧ b_7=1 →d=1.	0.84

For example, we analyses the first rule b_1=1 → d=1. It means if the prison computer use the operation system of Unix series, the information security of the computer is well. It is easy to understand because virus and attack in Unix series computer is less than in Windows series. Then we analyses the second rule b_5=0 ∧ b_7=1 → d=1. It means if we kill the virus on time and use the rough password, the information security of the computer is well.

4.3. Prisoners' Reform Data Set

We collect 357 items about prisoners' reform information from several jails. There are fourteen condition attributes and one decision attribute here. The decision attributes is reform performance. We use d to describe it. For d, {1, 2, 3} means {reform performance is good, reform performance is ordinary, performance result is bad}. And we use {a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}} to represent {sex, age, education, nationality, native place, health, skill, have criminal record or not, marriage, relationship with family, criminal type, prison term, current stage in prison, have drug taking record or not}.

For a_1, {1, 2} means {male, female}.

For a_2, {1, 2, 3} means {no older than 18 years, older than 18 years and no older than 40, older than 40 years}.

For a_3, {1, 2, 3} means {junior high school or lower education, senior high school or professional school education, college or university education}.

For a_4, {1, 2} means {Han nationality, national minority}.

For a_5, {1, 2} means {native province, other province}.

For a_6, {1, 2} means {healthy, ill-healthy}.

For a_7, {1, 2, 3} means {good, common, bad}.

For a_8, {1, 2} means {have criminal record, not have criminal record}.

For a_9, {1, 2, 3} means {spinsterhood, married, divorce}.

For a_{10}, {1, 2} means {good, bad}.

For a_{11}, {1, 2} means {serious, common, slight}.

For a_{12}, {1, 2, 3} means {longer than 8 years, from 2 years to 8 years, shorter than 2 years}.

For a_{13}, {1, 2, 3} means {early stage in prison, middle stage in prison, later stage in prison}.

For a_{14}, {1, 2} means {have drug taking record, not have drug taking record}.

Original prisoners' reform data set is in Table V below.

Table 12: Original Prisoners' Reform Data Set

a_1	1	2	2	1	1	2	1	…
a_2	2	2	1	2	2	1	3	…

a_3	1	3	2	1	1	3	3	…
a_4	1	1	2	1	2	1	1	…
a_5	1	2	2	1	2	1	2	…
a_6	1	2	2	1	1	2	2	…
a_7	1	2	2	1	2	2	2	…
a_8	1	2	2	1	1	1	2	…
a_9	1	2	2	3	1	3	2	…
a_{10}	2	2	2	3	2	2	3	…
a_{11}	1	2	2	1	2	1	2	…
a_{12}	1	3	1	2	2	3	3	…
a_{13}	1	2	3	1	2	2	2	…
a_{14}	1	2	1	2	1	2	2	…
d	3	1	3	2	3	1	1	…

There are 3 inconsistent data instances in the data set. We remove them in the pre-processing step and we have 344 effective data items. Here we use Johnson's Reduct generation algorithm 2 in ROSETTA [8] on the new decision table to generate reduct.

Table 13: Prisoners' Reform Reduct Data Set

No	Reduct Set
1	{ a_1, a_3, a_4, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}}
2	{ a_1, a_2, a_3, a_5, a_8, a_9, a_{10}, a_{12}, a_{13}, a_{14}}
.	…
12	{ a_2, a_3, a_4, a_5, a_6, a_8, a_{10}, a_{12}, a_{13}, a_{14}}

There are 12 reduct data set in Table XIII and they each has core attribute. We generated 48 rules by apriori algorithm with *support* = 1%*; confidence* = 100%, as shown in Table XIV. These rules are ranked with Rule Usefulness. The first 4 rules' usefulness is 1.00, and they are more useful than other rules here. We should pay more attention to the most useful rules other than the rules which are not so useful for our study.

Table 14: Rules of Prisoners' Reform Data Set

No	Reduct Rules	Rule Usefulness
1	$a_{14} = 1 \rightarrow d = 3$	1.00
2	$a_8 = 2 \wedge a_{13} = 2 \rightarrow d = 1$	1.00
3	$a_3 = 3 \wedge a_{12} = 3 \rightarrow d = 1$	1.00
4	$a_{10} = 2 \rightarrow d = 3$	1.00
.	…	…
47	$a_3 = 2 \wedge a_5 = 1 \wedge a_9 = 2 \rightarrow d = 2$	0.32
48	$a_4 = 2 \wedge a_6 = 2 \rightarrow d = 3$	0.24

Table 15: Core Rules of Prisoners' Reform Data Set

No	Reduct Rules	Rule Usefulness
1	$a_{14} = 1 \rightarrow d = 3$	1.00
2	$a_8 = 2 \wedge a_{13} = 2 \rightarrow d = 1$	1.00
3	$a_3 = 3 \wedge a_{12} = 3 \rightarrow d = 1$	1.00
4	$a_{10} = 2 \rightarrow d = 3$	1.00

We discussed with prison management policeman in different positions. In their opinion, our work can greatly help them to generate some useful rules. Then they can do better in improving the prisoners' reform performance. Before the collaboration with us, they have many years of traditional management experience. But some experiences are not suitable for the new prisoners now. Sometimes their old management experiences are not effective for new prisoners and they do not know how to resolve it.

For example, the first rule, $a_{14} = 1 \rightarrow d = 3$ means if one prisoner have drug taking record, his or her reform result is bad. The drug can cause the body, the spirit and personality distorted and abnormal. When they have no money for drug expenses, they usually steal or rob others. From this rule we can see the best way is to prevent drug of the young.

The 4th rule, $a_{10} = 2 \rightarrow d = 3$ means the prisoners' reform performance is complex. It is not only related with prison but also the family and society. If the person from the family and society do not despise the prisoners and give them the chance, they will have active attitude. Or they will have negative attitude and get poor performance.

4.4. Differences between Dual-Reduct Method and Traditional Reduct Approach

The Dual-Reduct algorithm is a new way to generate the core rules. It is different from the traditional reduct approach in the following ways.

First, the traditional reduct approach first constructs the discern matrix and generate the core. One reduct is enough to generate the rules. However, it generates a large mount of rules and it cannot select all the core rules in one time for most cases. But for our method, after one reduct, we ranked the rules by rule usefulness and select the most useful rules to build the new decision table and do the second reduct. One advantage is that Dual-Reduct method can obviously reduce the calculation and generate the core rules quickly.

Second, adopting rule usefulness as a measure of core rules, our Dual-Reduct method is easy and objective because the Rule Usefulness Measure is an objective measure [10].

However, we found a limitation for the Dual-Reduct method. If there is only one reduct for a data set, the Dual-Reduct method get the rules and their rule usefulness have the same value—1.00 [10]. It means all the rules are core rules. In this case there is no difference between our method and traditional approach. Therefore, if there is only one reduct, the Dual-Reduct method does not differentiate the generated core rules for a given data set.

5. Conclusion

This paper proposed rule evaluation method based on rough set. We review the history and related work on rough set theory especially the rule generation and evaluation. We first introduce the rough set theory because it is the theoretical foundation for the rule evaluation method. At the same time we notice the current rough set based knowledge discovery systems. First we discuss the association rules and related templates. And they are the basis for the rule generation and evaluation method proposed in this paper.

We introduced one Dual-Reduct method based on rough set, and it is a feasible and effective method to generating core rules for prison computer information security and prisoners' reform. After the first reduct and generating the rules, we rank the rules by rule usefulness and look at the useful rules as condition attribute, and then we construct another new decision table. After the process we do the second reduct to generate core rules from the new constructed decision table.

The Dual-Reduct method is based on two rule evaluation approach. One is the Rule Usefulness measure, and another is regarding rules as condition attributes. It is our most import contributions of this paper. Dual-Reduct method provides an automatic, easy and effective way of extracting useful knowledge. They can also be used jointly with other measures to facilitate the rule evaluations. The method can also incorporate domain related information into the rule evaluations. The proposed method in this paper can help user better understand the discovered knowledge from the original data set. They help people automatically select useful and import knowledge from a large amount of data.

The Dual-Reduct method is designed to extract core rules. This approach instead provides a set of core rules, which are from the reduct of a decision table. The decision table is constructed by considering the rules generated from original data as the condition attributes. Empirical studies for this rough set based Dual-Reduct method demonstrate its effectiveness on extracting core rules. This approach can be applied towards various applications.

In the future, we wonder to know how effective these core rules are by performing experiments with prison management policeman who are experts in the domain. Such user satisfaction studies may be conducted for related person's evaluations with proper measures from across a sufficiently large sample of users in the domain. Other human specified method such as interestingness measures and usefulness measures can be combined with Dual-Reduct method to evaluate the core rules. We want to apply our method to the larger data set and analyses the result in the future work.

It is possible to extend the proposed method to more knowledge discovery domains, such as patient survival prediction in the medical research field. We can use it to predict the patients' survival time and status after the doctors' diagnosis. Patients' survival time and status are the two main attributes for the predictions. Survival time means days of survival after a disease is diagnosed. Usually we predict it after generate the rules based on the medical data. The Rule-Usefulness evaluation method discussed in this paper can be applied to evaluate such prediction rules to facilitate the doctors' diagnosis.

Our approach can be also used for other rule generations such as sequential patterns and classification rules. We try to conduct logic analysis on the Dual-Reduct Method to make this method more fundamental and useful.

Acknowledgements

We greatly appreciate the funding supports of this work by the Commonweal Project Funds of Zhejiang Science&Technology Department (Grant No. 2013C31115), the Zhejiang Provincial Natural Science Foundation(Grant No. Y13F020205, LY13F020047) and the Open Project of Zhejiang Provincial Engineering Center(Grant No. 2012E10023-15).

References

[1] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data. Norwell, MA, USA: Kluwer Academic Publishers, **(1992)**.

[2] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I.Verkamo, "Finding interesting rules from large sets of discovered association rules," International Conference on Information and Knowledge Management. ACM Press, **(1994)**, pp. 401–407.

[3] J. Li, B. Tang, and N. Cercone, "Applying association rules for interesting recommendations using rule templates," Advances in Knowledge Discovery and Data Mining, vol. 3056, **(2004)**, pp. 166–170.

[4] W. Lin, S. Alvarez, and C. Ruiz, "Efficient adaptive-support association rule mining for recommender systems," Data Mining and KnowledgeDiscovery, vol. 6, no. 1, **(2002)**, pp. 83 – 105.

[5] Rakesh Agrawal and Ramakrishnan Srikant, "Fast algorithms for mining association rules. " In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proc. 20th Int. Conf. Very Large Data Bases, Morgan Kaufmann, **(1994)**, pp. 487–499.

[6] Xiaohua Hu, T. Y. Lin, and Jianchao Han, "A new rough sets model based on database systems," Fundam. Inf., Vol. 59, no. 2-3, **(2004)**, pp:135–152.

[7] Staal A. Vinterbo and Aleksander Øhrn, "Minimal approximate hitting sets and rule templates," Int. J. Approx. Reasoning, .Vol. 25, no. 2, **(2002)**, pp. 123–143.

[8] Aleksander Øhrn, Discernibility and Rough Sets in Medicine: Tools and Applications. PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim Norway, **(1999)**.

[9] Jan G. Bazan, Marcin S. Szczuka, and Jakub Wr´oblewski, "A new version of rough set exploration system." In RSCTC '02: Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing, Springer-Verlag , London,UK, pp. 397–404, **(2002)**.

[10] Lv Hanfei, "One Method to Get Most Useful Rules and Application in Prison Reform," 2010 2nd International Conference on Multimedia and Computational Intelligence, China, pp.1–5, Sept. **(2010)**.

[11] J. Bazan, H. Nguyen, S. Nguyen, P. Synak, and J. Wroblewski, "Rough set algorithms in classification problem," in Rough set methods and applications: new developments in knowledge discovery in information systems. Heidelberg, Germany: Physica-Verlag GmbH, **(2000)**, pp. 49–88.

[12] Grzymala-Busse, J.W.: Incomplete Data and Generalization of Indiscernibility Relation, Definability and Approximations. In: Proceedings of RSFDGrC **2005**, LNAI 3641 **(2005)** pp. 244–253.

[13] Aleksander Øhrn. Rosetta technical reference manual. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway. May 25, **(2001)**.

[14] Puntip Pattaraintakorn, Nick Cercone, and Kanlaya Naruedomkul. Hybrid intelligent systems: Selecting attributes for soft-computing analysis. In COMPSAC (1), pp. 319–325. IEEE Computer Society, **(2005)**.

[15] Bartlomiej Predki and Szymon Wilk. Rough set based data exploration using rose system. In SMIS '99: Proceedings of the 11th International Symposium on Founda-tions of Intelligent Systems, pp. 172–180, London, UK, **(1999)**. Springer-Verlag.

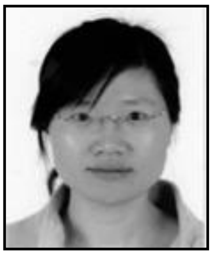
[16] Padhraic Smyth and Rodney M. Goodman. Rule induction using information theory. In knowledge Discovery in Databases, pp. 159–176. **(1999)**.

[17] Marcin S. Szczuka. Rules as attributes in classifier construction. In Ning Zhong, Andrzej Skowron, and Setsuo Ohsuga, editors, RSFDGrC, volume 1711 of Lecture Notes in Computer Science, pp. 492–499. Springer, **(1999)**.

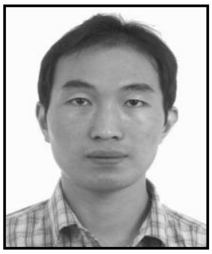
[18] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In KDD, pp. 32–41. ACM, **(2002)**.

[19] K. Kobayashi Y. Yoshida, Y. Ohta and N. Yugami. Mining interesting patterns using estimated frequencies from subpatterns and superpatterns. In G. Grieser et al., editor, DS 2003, pp. 494–501. Springer-Verlag Berlin Heidelberg, **(2003)**.

Authors



Lv Hanfei, she is an associate professor of Department of Information Technology and Management, Zhejiang Police Vocational Academy. Before joining Zhejiang Police Vocational Academy, she was a master student of Zhejiang University from 2002 to 2005. Her current research area include image processing, power aware computing system and its applications in computing systems.



Congfeng Jiang, he is an associate professor of School of Computer Science and Technology, Hangzhou Dianzi University, China. He is with the Grid and Services Computing Lab in Hangzhou Dianzi University.

Before joining Hangzhou Dianzi University, he was a PhD candidate in Huazhong University of Science and Technology from 2002 to 2007. He received his PhD degree in 2007. His current research areas include power aware computing system, virtualization, grid computing, etc. Dr Jiang is a member of ACM, IEEE, and China Computer Federation (CCF).

