World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:1, 2015

Pruning Algorithm for the Minimum Rule Reduct Generation

Şahin Emrah Amrahov, Fatih Aybar, Serhat Doğan

***Abstract*—**In this paper we consider the rule reduct generation problem. Rule Reduct Generation (RG) and Modified Rule Generation (MRG) algorithms, that are used to solve this problem, are well-known. Alternative to these algorithms, we develop Pruning Rule Generation (PRG) algorithm. We compare the PRG algorithm with RG and MRG.

***Keywords*—**Rough sets, Decision rules, Rule induction, Classification.

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

I. Introduction

NOWADAYS as working area and specialization increase, amount of information also increases comparatively. Lately it becomes necessary to interpret information sets and to get results from them. In this topic Rough Sets Theory is used as an important tool for discovering information from large data sets. Rough Sets Theory is developed by Pawlak [1] and it is applied in many areas. Some of these areas are medical diagnosis [2], [3], artificial intelligence [4], finance [5], conflict resolution [6], image analysis [7], pattern recognition [8], [9], control theory [10], feature extraction [11], [12], classification and rule reduction [13], support vector machines [14]. One of the areas in which Rough Sets Theory is used is classification and rule reduction. The first algorithm Rule Reduct Generation (RG) is proposed by Pawlak [15]. RG algorithm includes important deficiency. The algorithm examines all the situations and it considers all rules that it found as rule reduction. The second algorithm Modified Rule Generation (MRG) is developed by Guo and Chankong [16] as modified of RG. This algorithm fills the deficiency of RG, but in order to achieve this, information system has to be reorganized before each examination. In this paper, study on Pruning Algorithm of Generation A Minimal Set of Rule Reducts, or briefly Pruning Rule Generation (PRG) algorithm is explained. Different from MRG, this algorithm uses tree structured data type. In the first two sections of this study, in which rule reduct algorithms used in Rough Sets Theory are explained, RG and MRG algorithms are given. In the third section, PRG algorithm which is developed in this paper alternatively to these two methods is declared. Using a sample decision table as an example, these algorithms are compared in the fourth section. The experimental results are given in fifth section. In conclusion part, difference of PRG from two other methods and its benefits are explained.

Ş. Emrah Amrahov is with the Computer Engineering Department of Ankara University, Turkey (corresponding author to provide phone: 90-312-203-3300/1771; e-mail: emrah@eng.ankara.edu.tr).

F. Aybar was with Ankara University, Turkey. He is now with Gazi University, Ankara, Turkey.

S. Doğan was with the Bilkent University, Ankara, Turkey. He is now with Carnegie Mellon University, USA.

II. An Overview of the Rough Set Theory

Unlike the classical set theory, in which a set is defined only by its elements, in Rough Set Theory additional information about the elements of the set is given. In Rough Set Theory it is necessary to have some information about the elements of universe first. If some objects are characterized with the same information, then they assumed to be the same or indistinguishable. This relation of indistinguishability forms the base of Rough Set Theory. The main problems that can be solved by Rough Set approach define the objects of the sets according to the feature values, determining the dependence or partial dependence between features, reducing features, presenting the importance of features and setting up the decision rules. Moreover, Rough Set Theory can be used, for reducing data, discovering the dependencies, estimating the importance of data, setting up by decision algorithms from data, classifying data, discovering patterns in data, finding similarity and difference between data and determining cause effect relations [15].

A. Information System

Data for Rough Set analysis is represented in a feature-value table form in which each row shows an object or a sample and each column shows a feature that qualifies an object. Feature values belonging to objects are obtained by either measurement or human experiences. That kind of table is called Information System. An information system S is defined as $S=(U,A)$. U is non empty finite set of objects which is called S's universal set. A, is non empty finite set of features. Any $a \in A$ feature is defined by function $f_a : U \to V_a$. Set V_a is called range set of a. The information systems that include decision information are called decision tables. A decision table is formed by adding decision information to existing information system. In this way, besides the features of objects, the decisions on these objects are considered. In order to make this situation clearer, an example of information system and decision table can be examined. This example of decision table was formed by [17] (Tables I and II).

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

TABLE I
AN INFORMATION SYSTEM THE TOPIC OF WHICH IS PEOPLE WHO APPLIED JOBS

Person	Diploma	Experience	French	Reference	Decision
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSC	High	Yes	Neutral	Accept
x_5	MSC	Medium	Yes	Neutral	Reject
x_6	MSC	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

TABLE II
NUMERICAL FORM OF TABLE I

Person	F1	F2	F3	F4	Decision
x_1	1	2	1	3	1
x_2	1	1	1	1	0
x_3	2	1	1	2	0
x_4	3	3	1	1	1
x_5	3	2	1	1	0
x_6	3	3	1	3	1
x_7	1	3	2	2	1
x_8	2	1	2	3	0

We can show the relation between U universe, A features, d decision data and number values that belong to objects as below.

$$U = \{x_1, x_2, ..., x_8\}$$

$A = \{F1, F2, F3, F4\}$ = {Diploma, Experience, French, Reference}

$d =$ Decision

Range set that belongs to features:

$F1 = \{1,2,3\};\ 1 =$ MBA, $2 =$ MCE, $3 =$ MSC

$F2 = \{1,2,3\};\ 1 =$ Low, $2 =$ Medium, $3 =$ High

$F3 = \{1,2,\};\ 1 =$ Yes, $2 =$ No

$F4 = \{1,2,3\};\ 1 =$ Neutral, $2 =$ Good, $3 =$ Excellent

$d = \{0,1\};\quad 0 =$ Reject, $1 =$ Accept

B. Indiscernibility

A decision table clarifies all information about information system. This table may be very large. Same or indiscernible objects may be shown more than one feature or some features may be redundant.

Let $S = (U, A)$ be an information system and a set $B \subseteq A$ be a subset of features. For different two objects x and y from universe U, the equivalence relation $IND_S(B)$ defined in below is called B -indiscernibility relation.

$$IND_S(B) = \{(x, y) \in U^2 \mid \forall a \in B, a(x) = a(y)\}$$

In indiscernibility relation S index is omitted when it is clear that which information system is referred. If $(x, y) \in IND_S(B)$ then the objects x and y are indiscernible according to B. The objects are x and y indiscernible because both of them have the same feature values and the decision cannot be estimated. Before finding rule reducts in a decision table, it should be searched whether it has any indiscernible relations. Let Table III is handled after analyzing an information system. At first Table IV is checked for indiscernible relations. It can be seen the objects x_1, x_3 and x_4, x_5 are indiscernible between each other. The reorganized decision table is shown in Table IV.

TABLE III
A SAMPLE DECISION TABLE

Object	F1	F2	F3	F4	Decision
x_1	1	2	1	3	1
x_2	1	1	1	1	0
x_3	1	2	1	3	1
x_4	3	3	1	1	1
x_5	3	3	1	1	1

TABLE IV
THE REORGANIZED DECISION TABLE THAT HAS NO INDISCERNIBLE RELATION

Object	F1	F2	F3	F4	Decision
$x_1,\ x_3$	1	2	1	3	1
x_2	1	1	1	1	0
$x_4,\ x_5$	3	3	1	1	1

C. Rule Reduct Generation

Decision table is composed of objects, features of objects and decisions, as it is shown in Table III. For example, there is information of 699 patients in Wisconsin breast cancer database and there are 9 features for each patient. The decision in the database reveals if the tumor is malignant (ill-natured) or benign (good-natured). As the size of the database increase, it will be more difficult to examine all the features and take a decision. Indeed, when we examine any dataset, we always wonder if these data indicates a specific rule or not. That is, we try to find a rule with the data we are studying on. As a matter of course, validity of these rules depends on the sample size and accuracy of the dataset. What do we understand from the word `rule'? Let us try to explain it on Table II. There are only two objects for which feature $F1$ is equal to 2: x_3 and x_8. The decision is zero for both of them. Therefore, we can consider that the decision is zero if $F1 = 2$, without taking into account the value of other features. Also, from Table II, we see that if $F1 = 3$ and $F2 = 3$ then the decision is $d = 1$. In other words, we can think that there exists a rule which reveals "if $F1 = 3$ and $F2 = 3$ then $d = 1$"

In general, we have two kinds of rules in decision table. Suppose that, we handle features $F_{k_1}, F_{k_2}, ...F_{k_t}$. We denote the values of the features for the object m by $F_{k_1}(m), F_{k_2}(m), ...F_{k_t}(m)$. Suppose that we want to find rules

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

for the object i and the features take the values $F_{k_1}(i) = a_{k_1}, F_{k_2}(i) = a_{k_2}, ...F_{k_t}(i) = a_{k_t}$ for this object. Let us denote decision for i by d_i.

First kind rule: If for all objects j different than i, there is an index l such that the inequality

$$F_{k_l}(i) \neq a_{k_l}$$

holds, then we can say that "if $F_{k_1} = a_{k_1}, F_{k_2} = a_{k_2}, ...F_{k_t} = a_{k_t}$ then $d = d_i$" is a rule.

This kind of rule means that for all objects other than i at least one of the equalities

$$F_{k_1} = a_{k_1}, F_{k_2} = a_{k_2}, ...F_{k_t} = a_{k_t}$$

does not hold. For example, in Table II for all objects rather than x_1, at least one of the equalities $F1 = 1, F2 = 2$ is not true. Therefore, "if $F1 = 1$ and $F2 = 2$ then $d = 1$" is a rule.

Second kind rule: If for all objects j which are different than i and provide the equality

$$F_{k_1}(j) = a_{k_1}, F_{k_2}(j) = a_{k_2}, ...F_{k_t}(j) = a_{k_t}$$

We have the equality $d_j = d_i$ then we can say that "If $F_{k_1} = a_{k_1}, F_{k_2} = a_{k_2}, ...F_{k_t} = a_{k_t}$ then $d = d_i$" is a rule.

We should note that if we have a rule, then each statement comprising this rule as a subset will be a rule too. For example, in Table II, we found that "If $F1 = 2$ then $d = 0$". In this case, for instance, "If $F1 = 2$ and $F2 = 1$ then $d = 0$" is also a rule too. In other words, we can mention the minimum rule. There might be many rules in a database. The aim is to find all a minimal set of rules in the possible shortest time. This problem is called as minimum rule reduct generation or rule reduct generation problem. But why the rule reduction is so important? One can build expert systems by using all rules in the database and for this we can use the several of models, for example, Takagi and Sugeno fuzzy model [18] and solve the fuzzy linear systems by known methods [19]-[21]. We comprise such an expert system for breast cancer database and we can solve the classification problem in a few seconds with 96% rate of accuracy by the PRG algorithm which explained below. For comparison, the applying perception learning algorithm [22], [23] to the breast cancer database does not give classification of the patients.

D. Rule Reduct Generation (RG) Algorithm

This algorithm tries to find all situations that consist of rule reduct. Steps of this algorithm are given below:

Step 0. Set object number $i = 1$ and feature number $j = 1$.

Step 1. Choose $j = 1,...m$ for $k \neq i$. If $a_{ij} \neq a_{kj}$ or $a_{ij} = a_{kj} \wedge d_i = d_k$, then a_{ij} is declared to be r-reduct. If it is tried for all features of object, then go to step 2.

Step 2. Set $i = i + 1$. If it is tried for all objects, then go to step 3, else go to step 1

Step 3. Choose two features and go to step 1, until all $m-1$ feature groups have been considered

E. Modified Rule Generation (MRG) Algorithm

Although RG algorithm detects all rule reducts, it cannot find minimal set of rule reducts. Modified Rule Generation (MRG) algorithm is proposed by Guo and Chankong . Aim of MRG algorithm is to find the minimal set of rule reducts. By this way, unnecessary operations are avoided and time needed to achieve result becomes shorter than RG algorithm. Steps of MRG can be summarized as below:

Step 0. Sort information system according to decision values.

Step 1. Assign object number $i = 1$ and feature number in rule reduct $r = 1$.

Step 2. Scan row $i = 1$ from column $j = 1$. If $a_{ij} \neq$"*" then go to step 3, else go to step 4.

Step 3. For all $k \neq i$, if $a_{ij} \neq a_{kj}$ or $a_{ij} = a_{kj} \wedge d_i = d_k$, then a_{ij} is assigned to be r-reduct. If all columns are scanned for $j = 1,...n$, then go to step 4, else set $j = j + 1$ and go to step 2

Step 4. Set $i = i + 1$ and go to step 2 until all objects have .been considered. When there is no object left, go to step 5.

Step 5. Revise the decision table T according to objects which have same feature value by replacing the value of $a_{ij} \neq$"x" used to form corresponding 1-feature reduct by "*". Then go to step 6.

Step 6. In order to find higher order rule reducts in revised T' table, set $r = r + 1$. If $r = m$ then stop, else set $i = 1$ and go to step 7.

Step 7. By scanning ith row $a_{ij_1},...a_{ij_r}$ values, which belongs to $F_{j1},..., F_{jr}$ features, control whether they fit r-feature reduct or not. If a rule reduct is detected go to step 8, else go step 9 .

Step 8. Mark all r-feature rule reduct feature group $\{a_{ij_1},...a_{ij_r}\}$ with "$*r$" . Return to step 7.

Step 9. Set $i = i + 1$. If i i is greater than the object number in U, then go to step 6 , else go to step 7.

III. Pruning Rule Reduct Generation (PRG) Algorithm

In this section we give our algorithm, which we call pruning algorithm rule reduct generation (PRG).

Step 1. Assign the first object to the current object.

Step 2. Create a tree of the features for the current object

Step 3. Traverse the tree in preorder and if the current node is a rule, prune right sub tree of the current node and right sub tree of the nodes which are in the same depth with the current node including all the features of the current

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

node as a subset.

Step 4. If all objects are finished, then stop. Otherwise assign next object to the current one and go to the step 2.

By PRG algorithm we find all rules for each object by navigating on information system from the first object to the last one. That is, if we mark each object with i, for information system with object number N, i will be taken as $i = 1,2,...N$. Let us explain how to search for rules of the object i. Before searching, tree of features is built for the object i (first step). We build this tree by the way given below. We assign first feature to the root. Then, we assign second feature to the left child and, first and second features to the right child. After that, we are building up the left child of any node in this way: we first delete the feature with biggest number which is written in the node and write the next one instead of it. Then right child is built up by adding the next feature to the feature list of the node. As we are building the tree, we first assign zero to keys of all nodes of the tree. As an example, tree of features for an information system with 4 features is shown in Fig. 1.

In the second step, we start to preorder traversal on the tree. But if the key of the node is equal to 1, then we do not traverse on this node and right sub tree of it. (In other words, we are pruning the right sub tree). In the third step, we do these operations to the node in which we are on:

Let the feature list of the node be $F_{k_1}, F_{k_2}, ...F_{k_t}$.

We compare the feature values of the object i with the feature values of all objects different than i.

If we find a rule in a node, we change the key of it as 1. Apart from that, key of all nodes in the right sub tree of this node will be 1. Besides, we assign 1 to key of all nodes which has the same depth with this node and carry the feature list of this node $F_{k_1}, F_{k_2}, ...F_{k_t}$ as a subset too.

It can be seen from Fig. 1 that all subsets that belong to features are located into tree. While examining rule that belong to an object, in tree first of all, the way which will be followed starts from the root, and then goes to left child and finally the right child:

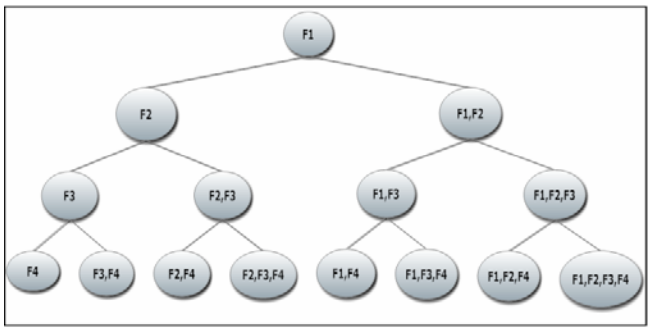


Fig. 1 Tree for the table with four features

Working principle of PRG algorithm can be described in detail as follows

Step 0. Number of object is assigned to be $i = 1$.

Step 1. By building up the tree, all keys in nodes are assigned to be Node.key $= 0$.

Step 2. Node = Root.

Step 3. SEQUENCEOFACTION (Node)

Step 4. Set $i = i + 1$. If all objects are done, go to step 5, if not go to step 1.

Step 5. Finish.

```
SEQUENCEOFACTION (Node)
  If (RULEREDUCTION (Node))
            /* The Rule in node is declared.*/
             Node.key = 1
```

For all nodes connected to right child of the node and all nodes that are in same line with the mentioned node and includes all features that the node has, assign Node.key = 1.

```
            /* This process makes redundant branches of tree pruned. */
  If Node.left ≠ null
     SEQUENCEOFACTION(Node.left)
  If Node.right ≠ null and Node.key = 0
     SEQUENCEOFACTION(Node.right)

RULEREDUCTION (Node)
  If Node.key = 1 return "FALSE"
  If for all j, j ≠ i; [(a_{j,k_1} ≠ a_{i,k_1}) or (a_{j,k_2} ≠ a_{i,k_21}) or
  ... a_{j,k_t} ≠ a_{i,k_t})]
     Print "if F_{k_1} = a_{k_1}, F_{k_2} = a_{k_2}, ...F_{k_t} = a_{k_t} then d = d_i" is rule
  reduct.
     Return "TRUE"
  If for all j, j ≠ i and for which the relation [(a_{j,k_1} = a_{i,k_1}) and
  (a_{j,k_2} = a_{i,k_21}) and... a_{j,k_t} = a_{i,k_t})] hold, we have the equality
  d_j = d_i
     Print "if F_{k_1} = a_{k_1}, F_{k_2} = a_{k_2}, ...F_{k_t} = a_{k_t} then d = d_i" is
  rule reduct.
     Return "TRUE"
  Else Return "FALSE"
```

IV. Comparison of PRG with RG and MRG

To compare PRG with RG and MRG we apply the algorithm to the Table V. This table is taken from Guo and Chankong [16].

TABLE V
A Sample Decision Table

Object	F1	F2	F3	F4	Decision
x_1	0	0	1	3	0
x_2	0	1	1	1	1
x_3	1	2	2	0	1
x_4	0	1	0	2	2
x_5	0	0	0	1	2

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

Application of RG and MRG algorithms is explained by Guo and Chankong [16].

Pruning Rule Reduction (PRG) algorithm also eliminates redundant rule reducts. It finds minimal rule reducts for an information system. The difference between MRG and PRG is the elimination method. MRG needs to revise the decision table but PRG uses a tree structured data type. This tree acts as a map for searching rule reducts. While processing the method, some branches of the tree is pruned, hence it means there is a redundant rule reduct. The branches that are declared as redundant are not searched for rule reduction. Below application of the PRG algorithm is explained for the fourth object from the same Table V. The PRG algorithm is tried to find the rules for this object. The searching methodology is indicated step by step. As it can be remembered PRG uses a tree structured data type. This tree that includes all subsets of the features is like a map for searching. In Fig. 2 PRG tries to find a one-feature rule reduct for x_4. The value of $F1$ is 0 for x_4 and PRG decides there is no rule reduct for $F1$ feature. Searching the rule reducts traces an in-order path on the tree. The second step is done in Fig. 3. The algorithm decides that $F1$ is not a rule reduct for x_4

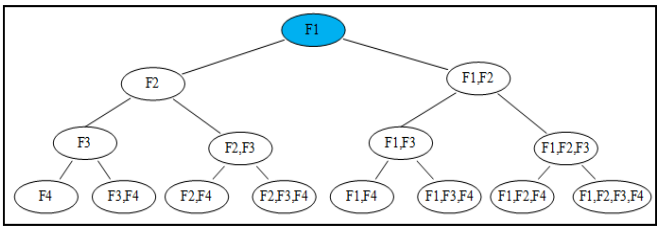


Fig. 2 First step of searching one-feature rule reducts for x_4

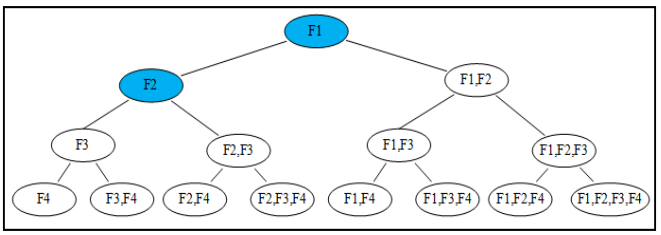


Fig. 3 Second step of searching one-feature rule reducts for x_4

The third step is done for $F3$. The value of $F3$ is 0 and PRG algorithm determines that $F3$ is a one-feature rule reduct for x_4. After finding the rule reduction, the tree branches are pruned. Pruning process is useful to avoid from redundant rule reducts. When a rule reduct is detected, the tree node that shows the feature group is marked. The right child of the node and all brother nodes that include the feature group are pruned.

This pruning process provides fast searching processes, shorter time to terminate and minimal rule reducts. In Fig. 5 the tree node $F3$ is declared as a one-feature rule reduct.

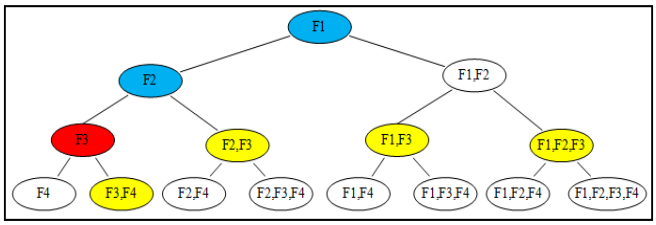


Fig. 4 Finding one-feature rule reduct for x_4 and pruning next related tree branches

After finding the rule reduction the tree nodes $F3F4$, $F2F3$, $F1F3$, $F1F2F3$ are pruned, because they include $F3$ feature and cause redundant rule reducts.

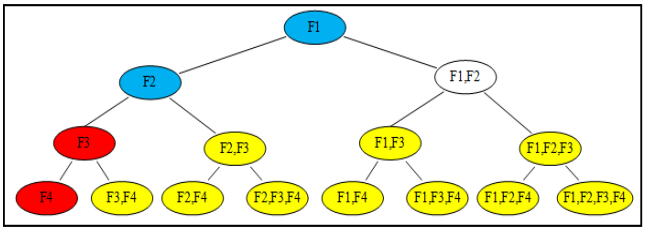


Fig. 5 Next step of finding one-feature rule reducts for x_4 and planning next related tree branches

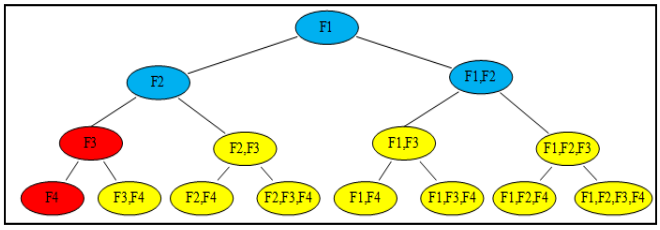


Fig. 6 Searching two features rule-reducts F1F2 for x_4

The next step is done for $F4$ feature. PRG algorithm determines that $F4$ is a one-feature rule reduct for x_4. Then pruning process is begun and the tree nodes $F2F4$, $F2F3F4$, $F1F4$, $F1F3F4$, $F1F2F4$, and $F1F2F3F4$ are pruned. Because these features include $F4$ feature and they are redundant rule reducts. It was noted above that the PRG traces the tree in-order direction. After finding rule reduct for $F4$ feature, it is expected that the algorithm will begin to search for $F3F4$ node. But since $F3$ is a one-feature rule reduct, the algorithm passes the right child. Then the right child of the $F2$, $F2F3$ will be searched, but it is also pruned so this node and its right child $F2F3F4$ are passed. The next node is $F2F4$, but by reason of one-feature rule reduct for $F4$, it has been also pruned and it is passed. Finally in execution queue the node $F1F2$ is searched whether it is a rule reduct or not. It is not rule reduct, hence the algorithm does not make any process for this feature couple. The result of execution is shown in Fig. 6. The other six nodes are not searched by PRG because they have been all pruned.

Open Science Index, Computer and Information Engineering Vol:9, No:1, 2015 publications.waset.org/10000502/pdf

V. Experimental Results

The PRG algorithm is applied to the Wisconsin breast cancer database obtained by Mangasarian and Wolberg [24], to the prostate cancer database [25] and to the Pima Indians Diabetes Dataset by National Institute of Diabetes and Digestive and Kidney Diseases [26] on the computer with features Intel (R) Core (TM) 2 Duo CPU E 8500 @ 3.16 GHz, 3.17 GHz, 4.00 GB of RAM. The Wisconsin breast cancer database includes the medical data of 699 patients. There are 9 features for each patient. Each feature has a value between 1 and 10. The value 10 means the worst case. In the database there are medical data of 241 malignant and 458 benign patients. After searching for rule reducts, 8614 minimal rule reducts were found. Rule reducts usually repeat for more than one patient, these repeating rule reducts were eliminated then 2010 distinct rule reducts were found. Running time for this database was 7 seconds by MRG and 6 seconds by PRG. The prostate cancer database includes the medical data of 458 patients. There are 12 features for each patient. After searching for rule reducts, 29060 minimal rule reducts were found. Running time for this database was 166 seconds by MRG and 139 seconds by PRG. The Pima Indians Diabetes Dataset includes the medical data of 768 patients. There are 8 features for each patient. After searching for rule reducts, 7516 minimal rule reducts were found. Running time for this dataset was 19 seconds by MRG and 20 seconds by PRG (Table VI).

TABLE VI
Experimental Results

		Breast Cancer Database	Prostate Cancer Database	Diabetes Database
Number of patients		699	458	768
Number of discernible patients		463	458	758
Number of features		9	12	7
Number of Rule reducts	RG	212467	1484854	45739
	MRG	8614	29060	7516
	PRG	8614	29060	7516
Running time	RG	84	935	41
	MRG	7	166	19
	PRG	6	139	20

VI. Conclusion

Rough Set Theory presents quite successful solutions for analysis and classifications of big data sets that consist of a large number of features. By rule reduction technique; analyzing data becomes easier. However, detecting rule reduction cases in information system of big data sets is pretty complicated. In this paper, information about methods of rule reduction called RG and MRG is given. Beside their availability, the cases, in which these two methods are deficient, are explained. PRG algorithm, which is developed in this paper, generates minimal set of rule reducts as the MRG algorithm. By avoiding redundant rule reductions, PRG algorithm not only facilitates the analysis of information system, but also solves the problem in short time

References

[1] Z. Pawlak, "Rough Sets", *International Journal of Computer and Information Sciences*, vol. 11, pp. 341-356, 1982
[2] A. Wakulicz-Deja, A and P. Paszek, "Diagnose progressive encephalopathy applying the rough set theory", *International Journal of Medical Informatics*, vol. 46, pp. 119-127,1997
[3] K. Slowinski, J. Stefanowski and D. Swinski, "Application of rule induction and rough sets to verification of magnetic resonance diagnosis", *Fundamenta Informaticae*, vol. 53, pp. 345-363, 2002
[4] P. J. Lingras," Belief and probability based database mining", in *Proc. of the Ninth Florida Artificial Intelligence Symposium*, 1996, pp. 316-320
[5] A. Mrozek and Ekabek , "Rough sets in economic applications." in *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Polkowski and Skowron, Ed. , Germany, Verlag, 1998, pp. 238-271
[6] Z. Pawlak, " On conflicts", *International Journal of Man-Machine Studies,* vol. 21, pp. 127-134, 1984
[7] A. Mrozek, and L. Plonka, "Rough sets in image analysis", *Foundations of Computing Decision Sciences*, vol. 18, pp. 259-273, 1993
[8] G. Griffin and Z. Chen , "Rough set extension of Tcl for data mining", *Knowledge-Based systems*, vol. 11, pp. 249-253, 1998
[9] R. Slowinski and J. Stefanowski, "Rough classification and incomplete information system", *Mathematiical and Computer Modeling*, vol. 12, pp. 1347-1357, 1989
[10] Z. Pawlak, and T. Munakata ," Rough Control, application of rough set theory to control", in *Proc. of the Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT'96*), 1996, pp. 209-218
[11] A. Kusiak and T. L. Tseng, "Modeling approach to data mining", in *Proc. of the Industrial Engineering and Production Management Conference*, Glasgow, Scotland, 1999, pp. 1-13
[12] A. Kusiak, *Computational Intelligence in Design and Manufacturing,* New York ,Wiley, 2000, pp. 498-527
[13] L. P. Khoo, S. B. Tor and L. Y. Zhai, "Rough set-based approach to classification and rule induction", *İnternational Journal of Advanced Manufacturing Technology* , vol. 15, pp. 438-444, 1999
[14] Y. Xu, and C. Liu C (2013) "A rough margin-based one class support vector machine", *Neural Computing and Applications*, vol. 22, pp. 1077-1088, 2013
[15] Z. Pawlak, *RoughSets:Theoretical Aspects of Reasoning About Data* , Boston, Kluver, vol. 9, 1992
[16] J-Y Guo and V. Chankong, V (2002) "Rough set-based approach to rule generation and rule induction", *International Journal of General Systems*, vol. 31, pp. 601-617, 2002
[17] J. Komorowski, L. Polkowski,and A. Skowron, "Rough Sets: A tutorial", In: *Rough- Fuzzy Hybridization: A new method for decision making.*,Springer-Verlag, 1998
[18] T. Takagi, and M. Sugeno, " Fuzzy identification of systems and its application to modeling and control", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 116-132, 1985
[19] Ş. E. Amrahov, and I. N. Askerzade, " Strong Solutions of the Fuzzy Linear Systems", *CMES - Computer Modeling in Engineering & Sciences*, vol. 76, pp. 207-216, 2011
[20] N. Gasilov, Ş. E. Amrahov, A. G. Fatullayev, H. I. Karakaş, and Ö. Akın,"A Geometric Approach to Solve Fuzzy Linear Systems", *CMES - Computer Modeling in Engineering & Sciences*, vol. 75,pp. 189-204, 2011
[21] N. Gasilov, A. G. Fatullayev and Ş. E. Amrahov, "Solution of Non-Square Fuzzy Linear Systems", *Journal of Multiple-Valued Logic and Soft Computing*, vol. 20, pp. 221-237, 2013
[22] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, 1962
[23] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm", *Machine Learning*, vol. 37, pp. 277-296, 1999
[24] O. L. Mangasarian and W. H. Wolberg , Cancer diagnosis via linear programming, *SIAM News, vol.* 23, pp 1-18, 1990
[25] D. F. Andrews and A. M. Herzberg, *A Collection of Problems from Many Fields for the Student and Research Worker*, Springer, 1985
[26] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus", in *Proc. of the Symposium on Computer Applications and Medical Care*, pp. 261—265, 1988