Rough Set Approach to Multivariate Decision Trees Inducing

Dianhong Wang, Xingwen Liu, Liangxiao Jiang, Xiaoting Zhang, Yongguang Zhao
China University of Geosciences, Wuhan, 430074, P. R. China
Email: xwliu1984@yahoo.cn
Email: {dianhwang; ljiang}@cug.edu.cn; xtzhang@163.com; ygzhao@163.com

***Abstract*—Aimed at the problem of huge computation, large tree size and over-fitting of the testing data for multivariate decision tree (MDT) algorithms, we proposed a novel rough-set-based multivariate decision trees (RSMDT) method. In this paper, the positive region degree of condition attributes with respect to decision attributes in rough set theory is used for selecting attributes in multivariate tests. And a new concept of extended generalization of one equivalence relation corresponding to another one is introduced and used for construction of multivariate tests. We experimentally test RSMDT algorithm in terms of classification accuracy, tree size and computing time, using the whole 36 UCI Machine Learning Repository data sets selected by *Weka* platform, and compare it with C4.5, classification and regression trees (CART), classification and regression trees with linear combinations (CART-LC), Oblique Classifier 1 (OC1), Quick Unbiased Efficient Statistical Trees (QUEST). The experimental results indicate that RSMDT algorithm significantly outperforms the comparison classification algorithms with improved classification accuracy, relatively small tree size, and shorter computing time.**

***Index Terms*—decision tree, classification, multivariate decision trees (MDT), rough set, positive region, generalization**

I. INTRODUCTION

Classification is an important part of data mining, and decision trees are widely used methods in this stage [1, 2]. Basically, a decision tree is nothing but a directed acyclic graph containing a root, a set of nodes, and a set of edges. Within the decision tree, an internal node denotes the test of a decision attribute, a branch represents the outcome of this test, and a leaf node is associated with a condition attribute label [3]. Existing studies have identified several advantages to the use of decision trees: no domain knowledge is needed for classification, they are able to handle high dimensional data, they are intuitive and generally easy to comprehend, they are simple and fast, and they have good accuracy [4].

The introduction of a tree-based approach to data mining can be dated back to the 1960s and 1970s [5]. One of the highly influential works on decision trees is the book written by Breiman et al., which a comprehensive technique of decision tree was presented and the classification and regression trees (CART) system [6] was developed. Another significant contribution to decision tree techniques is due to Quinlan [2, 3], who developed the popular ID3 and C4.5 systems. Besides, some scalability algorithms like SLIQ [7], SPRINT [8] and Rainforest algorithms [9] also have wide application.

All of the decision tree techniques stated above involve univariate decision trees, simply UDT [10]. That is, at each node, a split of the data is made based on the value of a single variable. In many real world applications, however, we need more than one condition attribute per record. For example, a bank not only needs to evaluate a customer's credit rating but also needs to predict the likelihood for him/her to apply for a loan in the coming future. In medical diagnosis, the doctor needs to diagnose a great number of diseases based on patient symptoms and test results. In such both cases, it is necessary to predict the values of multiple condition attributes with respect to a given set of decision attributes. Meantime, researchers have mentioned that the UDT leads to the following problems: subtrees are replicated in a UDT; attributes are repeatedly tested more than once along a path in a UDT; and the data increasingly divides into small fragments [10]. In addition, the univariate tests in tree building are sensitive to noise and have minor variations in the data [11].

Multivariate decision trees (MDT) can overcome, to some extent, above-stated problems. MDT generally has higher accuracy and a smaller tree size than UDT. Basically, an MDT is built in a way similar to building a UDT with one difference, i.e. performing univariate or multivariate tests. In inducing a UDT, at each test, we need to select the most promising attribute. Many heuristic measures were proposed, such as impurity [1], information gain [2], and so forth [3]. In building a MDT, however, we should find the most promising hyperplane (a set of attributes is needed). It is known that selecting the best attributes for test at a node is a NP-complete problem [11]. In order to find an optimal hyperplane for a multivariate test, researchers offered many suggestions such as Regression [6], Linear Programming [12, 13], Randomization [14]. For example, testing Boolean combinations of the attributes can handle the replication problem; forming linear combinations of the attributes can deal with the repeated testing problem. In Ref. 12, it

This work was partially supported by National 973 programme of China (No. 2010CB832800), the Natural Science Foundation of Hubei Province, China (No. 2010CDB04203), and by a grant from Three Gorges Research Center for geohazard, Ministry of Education, China (No. TGRC201018).

Corresponding author: Xingwen Liu; Email: xwliu1984@yahoo.cn

doi:10.4304/jcp.7.4.870-879

is shown that these combinations can improve classification accuracy and reduce tree size. Breiman et al. [13] discussed the issue of attributes combinations and incorporated a multivariate option into the CART system presenting a multivariate decision tree system, named OC1. The algorithm combines deterministic hill-climbing with randomized procedures to search for a better split. More recent researches on multivariate decision trees can be found in Refs. 14–17. Comparison experiments of univariate decision trees and multivariate decision trees have been conducted in Refs.10–11 and 15. The results of these studies[15] indicate that a univariate technique does not overfit and the univariate ID3 has better performance than multivariate linear methods while the data set is small and has few classes.

Various existing multivariate tree algorithms, however, e.g. CART-LC [6, 13], OC1 [13], QUEST [31] still follow the sort-and-search approach to finding an appropriate attribute splitting at each node, which is similar to that of univariate trees. Implementing the sort-and-search approach in multi-dimensional space needes much more computation than it is in one-dimensional space [13]. On the other hand, a multivariate tree is more complex to understand and interpret than a univariate tree. These are perhaps two important reasons that the multivariate trees have not achieved great improvement in classification accuracy and tree size over univariate decision trees.

Rough set theory (RST) [18–24] is a mathematical technique used to analyze imprecise, uncertain, or vague information in fields such as data mining, artificial intelligence, and pattern recognition. RST has many advantages, but its main benefit is that it does not require preliminary knowledge or additional information about the data [23]. A variety of methods [25–28] have been proposed to construct decision trees, which can be used to eliminate the unnecessary attributes of an object and thereby create a simplified version of the data. J. Wei et al. [24] proposed a rough-set based decision tree that used lower and upper approximations, while Y. Jiang et al. represented a decision tree that was based on core attributes and entropy.

To overcome the complexity in the tree structure expression and large computation, in this paper, a novel data mining algorithm, named Multivariate Decision Trees based on Rough Set Theory (RSMDT), is proposed. Experimental results show that the RSMDT has simpler tree structures and higher classification accuracy while its computing time only increases linearly with data size.

This paper covers the implementation and the evaluation of the proposed algorithm. The following sections provide a more detailed description of the algorithm. The remainder of this paper is organized as follows. We review some related rough set theory and introduce the two new concepts in section 2. Section 3 detailed states the proposed RSMDT algorithm. Pseudo codes and implementation process are presented. Section 4 contains the experimental results including performance evaluation and discusses the meaning behind the experimental data. Conclusions and suggestion for future work are discussed in section 5.

II. PRELIMINARIES

In this section, we recall some notions and methods related to rough set theory and two new concepts, which will be presented in the proposed algorithm. Detailed description of the rough set theory can be found in Refs. 18–28.

Rough Set Theory (RST) originated by Pawlak in 1980s has been described as a new mathematical tool to deal with inexact, uncertain or vague knowledge [18, 19]. The main idea of RST consists of approximation of a set by a pair of two crisp sets called the lower and upper approximations of the set [20]. The concept of indiscernibility or information granulation is the core of RST. Any set of all indiscernible objects is called an elementary set, and forms a basic granule of knowledge about the universe. Any union of some elementary sets is referred to as crisp set or precise set–otherwise the set is rough. Indiscernibility leads to the concept of boundary-line cases. Obviously crisp sets have no boundary-line elements at all. This means that boundary-line cases cannot be properly classified by employing available knowledge. Therefore, rough set theory expresses vagueness not by means of membership, but by employing a boundary region of a set. The object that belongs to a set with certainty is called lower approximation while upper approximation contains all objects that may possibly belong to the set [21].

Definition 2.1 *An information system [18]is defined as a pair (U,A,V, f) where U is a non-empty finite set of objects, A=C ∪D is a non-empty finite set of attributes, C denotes the set of condition attributes and D denotes the set of decision attributes,* $C \cap D = \varnothing$. $V = \cup_{a \in A} Va$, *Va is the domain (value set) of attribute a, f:U×A→V is an information function,* $\forall a \in A, x \in U, f(x,a) \in V_a$.

Definition 2.2 *(Equivalent Relation or Indiscernibility Relation [18]) Let S = (U, A, V, f) be a decision table and A= C ∪D, the subset* $B \subseteq A$ *is called equivalent relation or indiscernibility relation, denoted by IND(B) , which is defined as*

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

U/IND(B) are the sets of objects that are equivalent with respect to B.

Definition 2.3 *Let S = (U, CUD, V, f) be a decision table, and let* $R \subseteq C \cup D$ *and* $X \subseteq U$. *We can approximate X using the information contained in R by lower approximation R*(X), upper approximation R*(X) and the boundary of X which is called* $BND_R(X)$. *Now we denote R*(X), R*(X) and* $BND_R(X)$ *respectively as follows [18]:*

$$R_*(X) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\} \quad (2)$$

$$R^*(X)=\bigcup_{x\in U}\{R(x):R(x)\cap X\neq\varnothing\} \quad (3)$$

$$BND_R(X)=R^*(X)-R_*(X) \quad (4)$$

Definition 2.4 *(Positive Region [18]) Suppose P and Q are the equivalent relationship in U, then the positive region P of Q can be marked as POSP(Q), and*

$$POS_P(Q)=\bigcup_{X\in U/Q}R_*(x) \quad (5)$$

Definition 2.5 *Let U be a domain. P and Q are two equivalence relation sets defined on U,* $\forall R\in P$. *If Eq. (6) is true, we call the equivalence relation R is Q-unnecessary with respect to P, otherwise Q-necessary.*

$$POS_{IND(P)}(IND(Q))=POS_{IND(P-\{R\})}(IND(Q)) \quad (6)$$

Where IND(P)=∩P is also a equivalence relation, and is called an indiscernibility relation on P. For each $\forall R\in P$, *R is Q-necessary with respect to P, we call P is independent to Q.*

Definition 2.6 *(Relative Reducts and Core [19]) Reduct represents the minimum set of attributes that preserve the indiscernibility relationship. The relative reducts of the attribute set P, P⊂Q, is called the reduct of Q, denoted by REDQ(P), if P is minimal among all subsets of Q. The intersection of all reducts of Q is called the core of Q, and is denoted by CORE(Q). CORE(Q)=∩REDQ(P). If a∈ P and a∈ CORE(Q), the decision performance of the original system will be unchanged if attribute a is deleted from P. Otherwise, the decision performance of the original system will change.*

Definition 2.7 *(Positive Region Degree) Let S=(U, CUD, V, f) be a decision table, and let c∈ C, then the positive region degree on c of D (* $pos_c(D)$ *) denotes as follows:*

$$pos_c(D)=\frac{POS_C(D)-POS_{C-\{c\}}(D)}{POS_C(D)} \quad (7)$$

where $POS_{C-\{c\}}(D)$ implies the postive region in which attribute c is removed from C. $0\le pos_c(D)\le 1$. The greater $pos_c(D)$ is, the more importance condition attribute c has. Therefore, we use $pos_c(D)$ to select splitting attributes to construct multivariate decision trees.

Definition 2.8 *(Extended Generalization) Motivated by Refs.26–28, we introduce a concept of extended generalization based on traditional generalization. Let A and B be two equivalence relation sets on U, then*

$$U/IND(A)=\{X_1,X_2,...,X_m\}$$
$$U/IND(B)=\{Y_1,Y_2,...,Y_n\}$$
$$Z_i=\bigcup_{X_j\in U/IND(A)}\{X_j:X_j\subseteq Y_i\}\quad i=1,2,\dots,n, j=1,2,\dots,m$$

$$Z_k=\bigcup_{X_j\in U/IND(A)}\{X_j:X_j\not\subset Y_i,\forall i\}\quad k=n+1,n+2 \quad (8)$$

$$EGEN_B(A)=\{Z_1,Z_2,\dots,Z_n,Z_{n+1},Z_{n+2}\} \quad (9)$$

where $EGEN_B(A)$ denotes the extended generalization of equivalence relation A corresponding to equivalence relation B on U. Similarly, $EGEN_D(C)$ is the extended generalization of condition attributes C on decision attributes D. We have found that when K takes values of n+1 and n+2, the number of partition of decision attributes D as C is less than that of K takes n+1 so that we can get simpler and more clear construction of multivariate decision trees.

Theorem 2.1 *$EGEN_B(A)=\{Z_1, Z_2, \dots, Z_n, Z_{n+1}, Z_{n+2}\}$ is a partition of the universe set of U.*

Proof. According to the definition of partition, we have the following two properties:

(i) every element of U belongs to some sets of Z_i, $i=1,2,\dots n, n+1, n+2$.

(ii) if i≠j $i, j=1,2,\dots,n,n+1,n+2$, then $Z_i\cap Z_j=\varnothing$.

Now property (i) is obviously true, since

$$\bigcup_{i=1}^{n+2}Z_i=\bigcup_{i=1}^{m}X_i=U \quad (10)$$

Besides, we can easy to know that $Z_i\cap Z_j=\varnothing$, $i=1,2,\dots, n, j=n+1, n+2$.

Next we show by contradiction the property (ii) on the condition of i≠j, $i, j=1,2,\dots,n$.

If $Z_i\cap Z_j\neq\varnothing$, $i\neq j$, $i, j=1,2,\dots,n$, then there is at least an element $x\in U$ such that $x\in Z_i\cap Z_j$, $i, j=1,2,\dots,n$, so $x\in Z_i$ and $x\in Z_j$.

According to Eq. (8), we can get $x\in Y_i$ and $x\in Y_i$, then, $x\in Y_i\cap Y_j$, $Y_i\cap Y_j\neq\varnothing$, $i\neq j$, $i, j=1,2,\dots,n$. This contradicts the definition of partition, so we can get the conclusion that $EGEN_B(A)=\{Z_1, Z_2, \dots, Z_n, Z_{n+1}, Z_{n+2}\}$ is a partition of the universe set of U.

Table 1. The algorithm of RSMDT

Input: The decision table $S = (U, CUD, V, f)$, training data set A
Output: a multivariate decision tree T.
Process:
1: Initialize the tree variable T with the empty tree. Label the root by the set of all objects U and the current condition attributes set C.
2: begin
3: Compute $P=CORE_D(C)=\{a_1, a_2, \ldots, a_i\}$ according to Def. 2.6 where a_i is the relatively core attribute;
4: If $P=\varnothing \mid P=C$ then
5: Select attribute from the set C with highset positive region degree as P via Eq. (7);
6: else
7: Set Q= $a_1 \Lambda a_2 \Lambda \ldots \Lambda a_i$;
8: Compute $EGEN_D(Q)$ according to Def. 2.8;
9: Set $C=C\backslash P$, refine Z_{n+1}, Z_{n+2} from $EGEN_D(Q)$ and treat them as the current datasets;
10: Calculate positive region degree of th e remain of condition attribute C in Z_{n+1} and Z_{n+2};
11: Select the one with the maximum positive region degree as next testing attribute;
12: end
13: end
14: return T

III. The Proposed Algorithm for Attribute Reduction (RSIPSOAR)

In this section, We will discuss how to construct a multivariate decision tree with rough set theoy (RST).

According to the definition of relative core (see Def 2.6), we know that attributes of core in condition attribute set, compared with those in decision attributes set is crucial for making decisions. To select initial testing attributes based on relative core of condition attributes on decision attribute can significantly reduce the computation under mostly-contained condition.

Simple conjunction of the chosen attributes for multivariate test might lead to the problem of over-fitting of the testing data [13], so we define the extended generalization for one equivalence relation corresponding to another one. Because partitions correspond to attributes one by one, this partition defines the new attribute of a universe (U), namely the constructed multivariate test sets $EGEN_D(C)$ (see Def. 2.8). Taking $EGEN_D(C)$ as the root of the decision tree, the subjects in the information system are divided into different subsets according to the values of attributes.

Overall, the suggested algorithm works as follows. Firstly, we utilize discernibility matrix method to get the relative core attributes of condition attribute set according to decision attribute set, and treat them as initial testing attributes. Then, using extended generalization $EGEN_D(C)$ to construct multivariate testing. Next, to refine Z_{n+1} *and* Z_{n+2} from $EGEN_D(C)$ and treat them as current dataset following by computing the positive region degree (see Eq. (7)) of the remain condition attributes in the new datasets Z_{n+1} and Z_{n+2}. The attribute whose positive region degree is the highest will be selected as the splitting node.

The outline of the proposed algorithm is shown in Table 1.

IV. Experimental Results Analysis and Discussion

A. Sample Data Sets

For the purpose of our study, we run the experiments under the framework of *Weka* [29] and utilize all the 36 well-recognized UCI (University of California, Irvine) data sets [30] selected by *Weka* system, which represent a wide range of domains and data characteristics and are described in Table 2. We downloaded these sample data sets in the format of *arff* from the main website of *Weka*.

In order to perform the comparison experiments, the preprocessing stages of the sample data sets were carried out in the *Weka* platform, mainly including three steps:

- Replacing missing attribute values. In the experiments, C4.5, CART, CART-LC, QUEST and OC1 are unable to handle missing data. Therefore, we used the unsupervised filter named *ReplaceMissingValues* in *Weka* system to replace all missing attribute values in each sample data.
- To apply the filter of *Discretize* in *Weka* to discretize numeric attributes. In the following experiments, Equal Interval Width discretizaion method is used in *Weka*.
- Removing useless attributes. Obviously, if the number of values of an attribute is almost equal to the number of examples in a dataset, it rarely contributes to classification. Therefore, we used the unsupervised filter named *Remove* in *Weka* to delete this type of attribute. In these 36 sample data sets, there are only three such attributes: the attribute "*Hospital Number*" in the data set "*colic.ORIG*," the attribute "*instance name*" in the data set "*splice*," and the attribute "*animal*" in the data set "*zoo*."

B. Experiment Setup and Methodology

The suggested algorithm was implemented on a single Intel Pentium 4 processor with a CPU clock speed of 2.75 GHz and 1GB RAM. Our experimental comparisons

were conducted using six decision tree algorithms, namely, C4.5, CART, CART-LC, OC1, QUEST and the proposed algorithm RSMDT. C4.5 [3] is available in *Weka* system, and OC1 [13] and QUEST [31](with its multivariate algorithm) are freely obtained on the Internet. The CART version we used is the one in the IND package [32], which was acquired from NASA Ames Research Center. CART-LC is a multivariate algorithm in CART, which is implemented in OC1. The default settings of each system were used for the experiments.

Experiments were performed using the ten-fold cross validation method. The purpose of running multiple cross validations is to obtain more reliable estimates of the performance measures. A ten-fold cross validation[3] perform for one dataset consists of the following steps:

- Divide the whole dataset into ten equal-sized blocks using a stratified sampling method so that classes are equally distributed in all blocks.
- For each block, build a decision tree based on entire data not belonging to the block (the training dataset) and compute the classification accuracy using the data in the block (the testing dataset).
- Average the ten tests classification accuracy to obtain the final classification accuracy of this cross validation run. Similarly, calculate the average tree size of this operation. Computation time is measured by the total run time required to complete the entire ten-fold cross validation process.

C. Criteria for Evaluation

The following three evaluation criteria were used in our experiments:

- Classification accuracy, which is the main performance measure. This measure refers to the predictive ability of a decision tree with regard to classifying an independent set of test data.
- tree size, i.e. number of leaves, which is a simplicity/complexity measure of tree structure. Occam's Razor is a generally accepted principle, i.e. the fewer terms in a model the better. According to this view, one should attempt to minimize the size of the induced decision tree.
- computing time, which is a measure of computational cost. In our experiments, computing time represents the total time required to complete a ten-fold cross validation operation.

D. Results Analysis on the Three Criteria for Evaluation.

Tables 6, 7, 8 respectively, indicate the compared results of classification accuracy, tree size (in number of leaves) and computation time of all six algorithms on 36 UCI sample data sets, and mean values and standard deviation of the corresponding evaluate criteria on whole datasets are summarized at the bottom of the table. Tables 3, 4, and 5 show the summarized results between each pair of algorithms on three evaluate criteria, in which every entry *w/t/l* means that the algorithm in the corresponding row wins *w* datasets, ties *t* datasets, and loses *l* datasets, compared to the algorithm in the corresponding column.

From our study, we can see that the performance of RSMDT, not only in classification accuracy but also in tree size and computing time, is overall the best among the comparison algorithms in the paper. Now, we summarize several highlights briefly as follows:

- CART slightly outperforms C4.5 in classification accuracy (10 wins and 7 losses) and tree size (17 wins and 8 losses), but performs even worse than C4.5 in computation time (2 wins and 33 losses).
- CART-LC gently outmatches CART as for classification accuracy (11 wins and 9 losses) and tree size (12 wins and 8 losses), performs worse than CART in computing time (11 wins and 20 losses).
- OC1 is superior to CART-LC in classification accuracy (15 wins and 11 losses) , and in tree size (16 wins and 12 losses), performs immensely worse, however, in computing time (2 wins and 32 losses).
- QUEST gently outperforms OC1 in classification accuracy (9 wins and 7 losses) and tree size (10 wins and 7 losses), and significantly outmatches OC1 in computing time (21 wins and 7 losses).
- RSMDT overwhelmingly outperforms QUEST in classification accuracy (12 wins and 5 losses), tree size (18 wins and 7 losses), and computing time (31 wins and 2 losses).

Overall, RSMDT apparently outperforms all the other five algorithms in our experiments not only in classification accuracy but also in tree size. And RSMDT obviously outperforms all the comparison MDT algorithms in computation time.

Table 2. Description of sample datasets used for the experiment

No.	Dataset	Instance	Attributes	Classes	Missing	Numeric
1	anneal	898	39	6	Y	Y
2	anneal.ORIG	898	39	6	Y	Y
3	audiology	226	70	24	Y	N
4	autos	205	26	7	Y	Y
5	balance-scale	625	5	3	N	Y
6	breast-cancer	286	10	2	Y	N
7	breast-w	699	10	2	Y	N
8	colic	368	23	2	Y	Y
9	colic.ORIG	368	28	2	Y	Y
10	credit-a	690	16	2	Y	Y
11	credit-g	1000	21	2	N	Y
12	diabetes	768	9	2	N	Y
13	glass	214	10	7	N	Y
14	heart-c	303	14	5	Y	Y
15	heart-h	294	14	5	Y	Y
16	heart-statlog	270	14	2	N	Y
17	hepatitis	155	20	2	Y	Y
18	hypothyroid	3772	30	4	Y	Y
19	ionosphere	351	35	2	N	Y
20	iris	150	5	3	N	Y
21	kr-vs-kp	3196	37	2	N	N
22	labor	57	17	2	Y	Y
23	letter	20000	17	26	N	Y
24	lymph	148	19	4	N	Y
25	mushroom	8124	23	2	Y	N
26	primary-tumor	339	18	21	Y	N
27	segment	2310	20	7	N	Y
28	sick	3772	30	2	Y	Y
29	sonar	208	61	2	N	Y
30	soybean	683	36	19	Y	N
31	splice	3190	62	3	N	N
32	vehicle	846	19	4	N	Y
33	vote	435	17	2	Y	N
34	vowel	990	14	11	N	Y
35	waveform-5000	5000	41	3	N	Y
36	zoo	101	18	7	N	Y

Table 3. Summary on comparisons of classification accuracy of six decision tree algorithms

	C4.5	CART	CART-LC	OC1	QUEST
CART	10/19/7				
CART-LC	12/15/9	11/16/9			
OC1	16/15/5	13/13/10	15/10/11		
QUEST	19/13/4	14/17/5	10/21/5	9/20/7	
RSMDT	22/7/7	20/9/7	15/16/5	11/21/4	12/19/5

Table 4. Summary on comparisons of tree size (in number of leaves) of six algorithms

	C4.5	CART	CART-LC	OC1	QUEST
CART	17/11/8				
CART-LC	18/13/5	12/16/8			
OC1	15/14/7	18/8/10	16/8/12		
QUEST	13/16/7	14/18/4	11/19/6	10/19/7	
RSMDT	21/10/5	20/11/5	16/11/9	16/12/8	18/11/7

Table 5. Comparisons of computing time of six decision tree algorithms

	C4.5	CART	CART-LC	OC1	QUEST
CART	2/1/33				
CART-LC	1/4/31	11/5/20			
OC1	1/3/32	1/1/34	2/2/32		
QUEST	3/4/29	2/3/31	1/3/32	21/8/7	
RSMDT	6/7/23	19/12/5	22/8/6	28/5/3	31/3/2

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we summarize the existing improved algorithms for MDT and propose a novel approach (RSMDT) based on rough set theoy (RST) by introducing two concepts, i.e. positive region degree and extended generalization. We conducted a systematic experimental study on the classification accuracy, tree size, and computing time of RSMDT. Based on the experimental results, we conclude our study with the following three highlights:

- In terms of classification accuracy, the proposed RSMDT algorithm apparently outperforms all the other compared UDT and MDT algorithms.
- The RSMDT algorithm produces decision trees with relatively small size.
- The RSMDT algorithm performs faster than all the other four MDT algorithms and shows that it scales up well on large datasets.

Considering its simplicity and low computation, RSMDT is a promising MDT algorithm that could be used widely in real world applications.

Currently, the proposed RSMDT algorithm cannot effectively deal with noisy data, improving our algorithm based on variable precision rough set [33] (VPRS) is our main work in the future. VPRS is an extend version of rough set theoy, and has to some extent the ability of fault-tolerance, that is, even though there exists inconsistent data in the decision table, the algorithm may give a relatively satisfying result.

Table 6. Experimental results on classification accuracy (%) and standard deviation

Dataset	C4.5	CART	CART-LC	OC1	QUEST	RSMDT
anneal	97.54±0.82	97.41±0.93	97.82±0.55	95.14±0.38	95.04±0.47	95.16±0.28
anneal.ORIG	91.25±2.13	90.98±2.45	91.24±2.52	92.53±2.85	92.58±2.79	93.5±2.53
audiology	78.13±5.62	77.8±4.72	77.5±4.96	78.84±5.32	78.88±5.27	78.91±5.24
autos	80.2±6.98	80.13±5.24	79.24±5.21	80.92±5.96	78.29±5.8	78.39±5.78
balance-scale	62.03±3.04	62.1±2.8	63.26±3.18	65.87±3.62	65.93±3.56	66.04±3.46
breast-cancer	74.1±4.25	74.19±4.19	73.25±3.7	74.88±4.24	74.96±4.15	75.21±3.85
breast-w	96.02±2.73	95.22±1.78	96.72±2.44	97.26±2.23	97.5±2.27	98.1±2.83
colic	80.75±7.16	80.71±7.2	80.6±7.23	81.48±7.52	81.93±7.02	82.38±7.49
colic.ORIG	82.01±6.25	82.18±6.08	82.10±6.16	82.08±6.19	80.25±7.07	81.74±7.36
credit-a	88.24±3.26	88.19±3.3	88.04±3.47	88.1±3.3	88.63±2.84	90.38±2.47
credit-g	68.05±4.32	71.19±3.83	72.21±3.8	72.24±3.79	75.39±3.3	77.35±3.26
diabetes	76.10±6.81	76.2±6.65	77.39±7.21	77.4±7.17	78.83±7.96	79.95±7.62
glass	79.21±4.3	70.3±5.25	70.3±5.27	70.35±5.23	71.8±5.96	74.6±5.82
heart-c	83.07±0.56	83.21±0.44	83.26±0.3	81.45±0.63	82.73±0.54	82.8±0.26
heart-h	82.65±8.77	82.4±8.93	82.42±9.01	82.48±8.94	80.39±9.26	81.36±8.3
heart-statlog	82.42±7.85	82.39±7.91	82.42±7.87	82.51±7.78	82.74±7.55	84.92±7.24
hepatitis	65.54±10.8	58.17±9.61	56.39±9.25	59.23±9.57	61.46±10.7	64.23±8.3
hypothyroid	75.2±8.75	77.28±8.39	78.32±8.1	76.47±8.32	76.49±8.39	77.84±8.36
ionosphere	88.31±5.68	88.23±5.75	88.14±5.83	88.4±5.52	86.8±5.3	87.25±5.88
iris	95.15±2.43	91.03±2.2	93.27±2.36	94.18±2.81	94.41±2.53	95.20±2.04
kr-vs-kp	99.02±0.17	99.21±0.28	99.38±0.19	98.93±0.26	98.26±0.94	97.34±0.35
labor	65.49±14.6	65.71±14.47	65.32±14.8	65.7±13.5	67.8±12.9	69.3±13.2
letter	82.8±0.36	83.62±0.73	84±0.82	82.04±0.45	80.15±1.26	83.58±0.6
lymph	82.38±7.31	82.34±7.37	83.43±7.2	83.10±8.23	83.53±7.95	85.3±8.02
mushroom	99.19±0.04	99.2±0.11	99.2±0.08	99.31±0.03	99.52±0.05	99.65±0.07
primary-tumor	71.45±1.91	71.29±2.17	70.13±2.35	71.2±1.89	70.01±1.93	70.94±1.37
segment	97.14±0.83	97.55±1.28	97.93±1.42	98.21±1.26	98.28±1.19	98.42±1.35
sick	93.16±4.6	94.5±4.25	95.34±4.38	95.69±4.93	95.8±4.77	96.9±3.5
sonar	70.12±9.68	70.02±9.3	71.88±9.02	72.26±8.75	75.42±9.32	75.88±8.99
soybean	83.25±2.03	83.5±1.87	84.21±2.14	84.4±1.97	82.1±2.26	83.01±1.76
splice	96.72±1.04	97.25±0.72	97.43±0.86	97.6±0.65	97.93±0.35	97.97±0.32
vehicle	82.7±3.47	82.81±3.44	81.5±3.82	81.95±3.96	83.26±3.15	83.35±3.05
vote	97.02±3.18	92.64±3.2	96.3±3.54	96.82±3.83	96.94±3.71	96.9±3.07
vowel	88.24±2.35	88.29±2.38	87.58±2.17	88.49±2.51	86.38±2.6	87.32±2.63
waveform-5000	85.01±1.73	87.14±1.61	84.59±1.32	86.44±1.3	87.2±1.84	87.43±1.69
zoo	89.6±2.45	80.25±2.87	80.48±2.69	81.76±2.47	82.97±2.83	85.32±3.07
Average	83.14±5.29	83.27±5.73	83.95±4.87	84.51±5.42	86.95±4.23	88.47±3.64

Table 7. Experimental results on tree size (in number of leaves) and standard deviation

Dataset	C4.5	CART	CART-LC	OC1	QUEST	RSMDT
anneal	15.4±0.6	13.8±0.9	15.3±0.9	10.4±0.6	8.3±0.5	6.4±0.5
anneal.ORIG	18.3±1.3	15.9±1.3	14.2±1.1	9.3±0.4	6.2±0.3	4.1±0.1
audiology	5.2±0.4	10.7±0.6	16.2±0.9	18.3±1.1	8.4±0.3	8.1±0.2
autos	4.9±0.3	9.6±0.9	14.3±0.6	17.2±0.7	7.9±0.5	6.3±0.3
balance-scale	8.2±0.6	12.6±1.1	23.7±1.4	30.6±1.6	15.7±0.8	11.3±0.6
breast-cancer	5.5±0.4	10.3±0.6	15.2±1.2	17.9±1.6	8.2±0.7	7.2±0.5
breast-w	9.2±0.7	6.4±0.2	6.9±0.7	3.8±0.5	2.5±0.9	12.1±0.8
colic	14.1±0.2	11.6±1.3	17.3±0.8	19.3±1.1	9.3±0.7	8.5±0.7
colic.ORIG	5.8±0.1	10.2±0.5	16.1±0.3	18.3±0.5	27.2±1.6	7.3±0.3
credit-a	61.6±2.5	5.2±0.4	8.6±0.5	7.1±0.4	5.8±0.2	8.2±0.5
credit-g	75.3±3.6	7.9±0.6	10.5±1.4	8.4±0.3	7.2±0.4	29.3±1.6
diabetes	64.4±3.1	5.8±0.8	9.7±0.4	7.8±0.4	6.4±0.3	5.2±0.4
glass	12.1±0.7	9.6±0.5	8.8±0.3	15.2±1.2	16.2±0.7	4.9±0.3
heart-c	15.8±0.6	10.2±0.6	9.3±0.5	16.2±0.9	6.8±0.4	5.1±0.2
heart-h	16.1±0.9	12.5±0.7	11.3±1.2	17.8±0.4	7.3±0.1	8.3±0.8
heart-statlog	25.4±1.4	15.8±0.9	17.9±0.9	15.3±1.1	6.5±0.3	4.6±0.2
hepatitis	5.1±0.7	11.5±0.5	13.2±0.4	10.4±0.7	14.1±0.9	8.7±0.5
hypothyroid	1358.4±65.9	1247.2±56.3	936.2±44.6	752.9±33.5	503.8±22.4	352.4±16.8
ionosphere	6.1±0.7	14.6±1.8	12.5±0.9	18.6±1.1	8.3±0.5	3.8±0.2
iris	4.8±0.4	8.5±0.3	9.1±0.3	10.1±0.4	16.8±1.6	5.1±0.3
kr-vs-kp	1257.7±63.5	1174.2±57.7	894.3±41.2	692.5±33.6	462.2±18.4	325.1±12.2
labor	3.5±0.2	10.4±0.6	17.2±0.8	15.6±0.5	8.4±0.3	9.2±0.4
letter	984.5±48.7	694.3±34.2	479.3±24.8	924.4±65.5	218.2±11.2	153.7±7.3
lymph	4.9±0.3	10.7±0.8	13.7±0.7	11.3±0.6	4.7±0.5	6.3±0.2
mushroom	542.5±10.7	492.6±10.3	416.5±9.9	343.7±6.6	285.1±6.1	195.7±3.8
primary-tumor	8.7±0.3	15.2±0.5	18.8±0.7	14.2±0.5	19.3±1.2	18.9±1.2
segment	949.3±37.5	735.2±26.3	626.1±19.5	537.2±22.3	393.3±16.2	274.5±11.4
sick	1063.2±64.5	935.5±56.6	885.2±40.8	639.3±32.7	437.1±25.3	315.7±13.2
sonar	4.9±0.5	10.4±0.7	9.3±0.9	16.8±0.7	7.4±0.4	8.2±0.7
soybean	13.4±0.7	32.5±1.4	27.9±1.3	35.1±1.5	23.1±1.2	13.6±0.6
splice	647.2±29.3	582±28.5	515.9±23.6	435.2±22.3	385.4±18.9	311.8±14.5
vehicle	99.5±5.5	25.6±1.6	35.9±1.9	31.2±1.4	28.6±1.3	29.9±1.2
vote	44.7±2.6	13.5±1.3	17.4±0.9	15.8±0.8	24.3±1.7	17.5±1.3
vowel	553.1±23.6	374.3±16.8	268.7±14.8	652.3±33.7	105.8±6.1	76.6±3.8
waveform-5000	904.6±46.3	174.9±8.2	88.5±4.6	29.8±1.7	53.6±2.7	18.4±0.9
zoo	7.5±0.7	21.5±1.2	35.7±1.8	31.6±1.5	17.3±0.9	12.6±0.8
Average	258.3±13.9	184.4±8.4	146.2±7.6	152.7±9.2	92.5±4.5	58.1±3.1

Table 8. Experimental results on computing time (in seconds) and standard deviation

Dataset	C4.5	CART	CART-LC	OC1	QUEST	RSMDT(pre-process time)
anneal	61±4	215±11	489±23	2165±102	1047±41	203±9(89±4)
anneal.ORIG	44±5	198±9	412±21	1993±92	931±52	125±7(68±7)
audiology	19	64±4	41±2	159±9	202±13	18(7)
autos	22	73±4	54±3	213±14	293±15	84±19(37±2)
balance-scale	10	39±2	27	203±11	217±19	31±4(15)
breast-cancer	15	48±5	34±6	103±7	254±11	37±2(18)
breast-w	14	59±3	61±9	82±5	326±13	25±2(9)
colic	27±4	23±7	27	98±5	262±9	35(10)
colic.ORIG	19±3	78±4	62±2	116±6	282±16	25±2(4)
credit-a	21	92±4	128±6	942±43	586±31	101±6(55±2)
credit-g	28±6	116±5	173±9	1148±52	623±34	192±7(83±4)
diabetes	30±3	120±8	103±7	1200±47	1004±53	162±9(79±7)
glass	11	67±2	25±3	89±5	103±7	26±3(8)
heart-c	13	51±3	38±2	101±13	96±8	32±5(12)
heart-h	10	47±6	31	81±4	142±8	75±11(21±3)
heart-statlog	21	42±2	20±5	65±4	116±5	43±8(17±8)
hepatitis	14	28±1	20	73±3	193±8	18(7)
hypothyroid	204±11	936±34	3982±215	11242±459	1132±54	638±32(204±17)
ionosphere	21	63±4	22±5	196±11	227±13	44(18)
iris	19	21±2	48±9	104±7	121±5	20(7)
kr-vs-kp	183±8	802±34	3321±152	9814±351	947±42	492±21(165±11)
labor	14	15	19±4	36±3	45±4	75±3(31±4)
letter	351±16	1264±58	7391±341	40031±1237	3317±143	1267±47(427±23)
lymph	18	39±5	42±7	98±6	115±4	15(5)
mushroom	273±14	915±42	6126±263	26215±783	2347±149	1003±51(379±19)
primary-tumor	26	54±2	21±9	137±8	199±9	84±19(37±8)
segment	106±7	773±26	2939±145	8417±316	835±41	561±27(216±11)
sick	194±9	952±31	3517±161	10042±383	1152±67	763±39(324±16)
sonar	21	41±5	28	75±11	158±12	54±6(23±4)
soybean	29±2	137±6	91±5	226±11	351±18	52±9(26)
splice	118±5	447±24	2734±117	8731±316	726±34	309±15(182±8)
vehicle	48±4	249±13	164±9	289±15	416±22	94±7(28±3)
vote	26±6	116±9	72±4	134±10	242±13	47±5(23)
vowel	55±4	276±14	183±9	337±17	42±6	75±4(31)
waveform-5000	274±12	873±35	4639±116	13300±282	1426±57	928±41(392±25)
zoo	12	37±7	15±4	43±2	58±3	65±8(27±3)
Average	78±12	417±62	943±47	3125±126	518±29	283±33(64±15)

REFERENCES

[1] E. Frank, Y. Wang, S. Inglis, G. Holmes, I. H. Witten, "Using model trees for classification," *Machine Learning*, vol. 32, no. 1, pp. 63–76, 1998.

[2] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, no. 1, pp. 81–106, 1986.

[3] J. R. Quinlan, *C4.5: Programs for Machine Learning* Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.

[4] L. Hyafil, R. L. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, 1976.

[5] S. R. Safavin, D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[6] L. Breiman, J. H. Friedman, R. Olshen, C.J. Stone, *Classification and Regression Trees*, Wadsworth Int. Group, 1984.

[7] M. Mehta, R. Agrawal, and J. Rissanen, "SLIQ: A fast scalable classifier for data mining," in *Int. Conf. Extending Database Technology (EDBT'96),* Avignon, France, pp. 18–32, 1996.

[8] J. Shafer, R. Agrawal, and M. Mehta, "SPRINT: A scalable parallel classifier for data mining," in *Int. Conf. Very Large Data Bases (VLDB'96)*, Bombay, India, pp. 544–555, 1996.

[9] J. Gehrke, R. Ramakrishnan, and V. Ganti, "Rainforest: A framework for fast decision tree construction of large datasets," *Data Mining and Knowledge Discovery*, vol. 4, pp. 127–162, 2000.

[10] C. E. Brodley, P. E. Utgo, "Multivariate versus univariate decision trees," *Tech. rep. COINS CR*, Dept. of Computer Science, University of Massachusetts at Amherst, 1992.

[11] C. E. Brodley, P. E. Utgoff, "Multivariate Decision Trees, *Machine Learning*, vol 19, pp. 45–77, 1995.

[12] K. Bennett, "Decision tree construction via linear programming," in *Proc. 4th Midwest Artificial Intelligence Cognitive Science Soc. Conf*., Utica, IL, pp. 97–101, 1992.
[13] S.K. Murthy, S. Kasif , S. Salzberg, "A System for in duction of oblique decision trees," *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 1–33, 1994.
[14] D. Heath, S. Kasif, and S. Salzberg, "Learning oblique decision trees," in *Proc. 13th Int. Joint Conf. Artificial Intelligence*, *R. Bajcsy*, Ed., San Mateo, CA, pp. 1002–1007, 1993.
[15] X. B. Li, "Multivariate Decision Trees for Data Mining," *Ph.D. dissertation*, Univ. South Carolina, Dept. Manage. Sci., Columbia, SC, 1999.
[16] R. Rivest, "Learning decision lists," *Machine Learning*, vol. 2, pp. 229–246, 1987.
[17] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. **7**, pp. 1–30, 2006.
[18] Z. Pawlak, "Rough Sets," *International Journal of Computer and Information Sciences*, vol. 11, pp. 341–356, 1982.
[19] Z. Pawlak and A. Skowron, "Rough sets: Some extensions", *Inf. Sci*., vol. 1, no. **77**, pp. 28–40, 2007.
[20] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Dordrecht, Kluwer, 1991.
[21] Z. Pawlak, "AI and intelligent industrial appliactions: the rough set perspective," *Cysbernetcis and Systems*, vol. 31, pp. 227–252, 2000.
[22] Z. H. Tang, "A novel extension data mining approach based on rough sets pair analysis," *Journal of software*, vol. 5, no. 4, pp. 447–454, April 2010.
[23] J. Wei, D. Huang, S. Wang, Z. Ma, "Rough set based decision tree," In *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 7, pp. 426–430, 2002.
[24] X. P. Li, M. Dong, An algorithm for constructing decision tree based on variable precision rough set model, in *Fourth International Conference on Natural Computation*, pp. 280–283, 2008.
[25] L. J. Huang, M. H. Huang, and B. Guo, A new method for constructing decision tree based on rough set theory, In *2007 IEEE International Conference on Granular Computing*, pp. 241–244, 2007.
[26] D. Q. Miao, "Rough sets based on approach for multivariate decision tree construction," *Journal of Software*, pp. 425–431, 1997. (in Chinese)
[27] C. C. Chan, "A rough set approach to attribute generalization in data mining," *Information Sciences*, vol. 107, pp. 169–176, 1998.
[28] R. Slowinski and D. Vanderpooten, "A generalized definition of rough approximations based on similarity," *IEEE Trans. Knowl. Data Eng*., vol. 12, no. 2, pp. 331–336, 2000.
[29] I. H. Witten and E. Frank, "Data mining: Practical Machine Learning Tools and Techniques, second ed., *Morgan Kaufmann*, http://prdownloads.sourceforge.net /weka/datasets-UCI.jar, 2005.
[30] C. Blake, C. Merz, *UCI Repository of machine learning databases*, available on-line: http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.
[31] W. Y. Loh and Y. S. Shih, "Split selection methods for classification trees," *Statist. Sinica*, vol. 7, pp. 815–840, 1997.
[32] W. Buntine and R. Caruana, *Introduction to IND Version 2.1 and Recursive Partitioning*, NASA Ames Research Center, Moffet Field, CA, 1992.
[33] W. Ziarko, "Variable precision rough set model," *J. Comput. Syst. Sci*., vol. 46, no. 1, pp. 39–59, 1993.

Dianhong Wang was born in 1957. Present he is a professor and doctoral supervisor in Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan, China. His current research areas are data mining, machine learning, rough set and knowledge discoverage.

Xingwen Liu was born in 1984, and is a Ph.D. candidate from 2009, in Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan, China. His current research interests include data mining, rough set, evolutionary computation.

Liangxiao Jiang was born in 1977. He received his PhD degree from China University of Geosciences in June, 2009. Currently, he is an associate professor in Department of Computer Science, China University of Geosciences. His research interests include data mining and machine learning.