Approach of Rule Extracting Based on Attribute Significance and Decision Classification

Yandong Zhai, Chunguang Zhou and Yingjuan Sun
College of Computer Science and Technology, in Jilin University, Changchun, China
Email:syj_pyf@sohu.com

Dongbing Pu*
College of Computer Science and Information Technology, Northeast Normal University, Changchun, China
Email:pudb@nenu.edu.cn

YingHui Sun*
College of computer Jilin Normal University,Siping , China
Email:sunyh178@163.com

***Abstract*—Propose a novel approach of rule extracting based on attribute significance and decision classification (REBSC). Condition attributes are only discretized with their own features in local discretization theory, and the eventual rule set normally can be achieved after attribute reduction. The REBSC approach given in this paper makes the most of each condition attribute significance and classification object, generates minimum decision rules without attribute reduction. Experiment one fully demonstrates the reasonability of the REBSC algorithm. Experiment two further proves its validity and testifies the significance of breakpoint division.**

***Index Terms*—rough set, attribute significance, discretization, decision rule, attribute reduction**

I. Introduction

Rough set theory, which was originated by a Poland mathematician Z. Pawlak, is a kind of mathematic theory on analyzing imprecise data. Its characteristic is to find out the law of problem with similar fields, ascertained by indiscernible relations and classes, directly from data rather than their characteristics or descriptions given in advance [1] .

Machine learning is to extract knowledge from data. Knowledge discovery, based on rough set, mainly utilizes the information system to represent knowledge and creates the knowledge system with data preprocessing, attribute reduction, rule generation, etc[2]. Generally, each of records in a decision system is regarded as a decision rule. But there is no widely practical meaning for training samples. Therefore, we can find something in common among samples and obtain a few meaningful decision rules nothing but attribute discretization [3].

Generally, discretization algorithms consist of two categories: local discretization algorithms (LDA) and global discretization algorithms (GDA). LDA neglect influences of other attributes in the process of single attribute discretization. As a result, some important relations containing in data are easily destroyed. GDA are often so much the better because of their consideration of interactions between attributes. This paper gives a novel approach of rule extracting based on attribute significance and decision classification (REBSC). The REBSC algorithm belongs to GDA and fully considers the significance and the relationship to decision classification for each condition attribute during generating decision rules [4]. Each of condition attributes is discretized in descending order of significance before ultimate decision rules being generated. The discretized decision table usually requires sequent reduction in order to eliminate redundant rules in rough set theory. The decision table, generated by the REBSC in this paper, does not require reduction any more, but also shows higher recognition accuracy with minimum breakpoints as well as shorter rule.

II. Conceptions

A. Knowledge Representation System and Decision System

A decision knowledge system, also called a decision table, is a knowledge representation system in the form of $S=(U,A,V,f)$ along with condition attributes and decision attributes. Here, U is a universe of non-empty finite set. $A=C\bigcup D$ indicates a non-empty finite set too. $C\bigcap D=\Phi, D\neq\Phi$, where C represents the condition attribute set and D is the decision attribute set. $V=\bigcup_{a\in A}V_a$, where V_a is the value function of attribute a and V is a codomain. Function $f:U\rightarrow V_a$ is a single mapping. For $\forall x\in U$, $f:U\rightarrow V_a$ enable x own a unique value in V_a when x is given the attribute a.At the same time, for $\forall x\in U$, there are two sequences in the

* Corresponding author

doi:10.4304/jcp.7.2.514-520

form of $C(c_1(x),...,c_n(x))$ and $D(d_1(x),...,d_m(x))$. Decision rules can be represent by $c_1(x),...,c_n(x) \to d_1(x),...,d_m(x)$ [5-8].

Definition 1: For $B \subseteq A$ in decision table S, if there are two different objects x, y with the same condition attribute value in the attribute set B and they belong to different classification respectively, then x and y are inconsistent in relation to B, else they are consistent in relation to B.

Definition 2: For any attribute subset $B \subseteq A$, the indiscernibility relation $IND(B)$ is defined as $IND(B) = \{(x,y) \mid (x,y) \in U \times U, \forall_{b \in B} \forall_{x \in U} \forall_{y \in U} (f(x,b) = f(y,b))\}$.

Definition 3: The indiscernibility relation $IND(B)$ divides U into such $X_1, X_2, \cdots, X_t$ as t equivalence classes. The dividing is denoted as $U / IND(B)$. $[x]_B$ is a set in which all of elements exist in U and are equivalent to x by the action of $IND(B)$ [9,10].

B. Description of Discretization

Rough set theory can only process discretized data. We can find out something in common among samples in the process of attribute discretizing. Therefore, it is highly important for rule generation to discretize the continuous attributes in a decision system.

Assuming $V_a = [l_a, r_a]$, l_a and V_a are given by

$$l_a = c_0^a < c_1^a < \cdots c_{k_a}^a < c_{k_a+1}^a = r_a \quad (1)$$

$$V_a = [c_0^a, c_1^a) \cup [c_1^a, c_2^a) \cup \cdots \cup [c_{k_a}^a, c_{k_a+1}^a) \quad (2)$$

For any breakpoint set $\{(a,c_1^a),(a,c_2^a),\cdots,(a,c_k^a)\}$ in V_a, a classification P_a in V_a is defined as

$$P_a = \{[c_0^a, c_1^a), [c_1^a, c_2^a), \cdots, [c_{k_a}^a, c_{k_a+1}^a)\} \quad (3)$$

A new decision table $S^P = (U, A, V^P, f^P)$ can be defined by any $P = \bigcup_{a \in A} P_a$, where $f^P(x_a) = i \Leftrightarrow f(x_a) \in [c^a{}_i, c^a{}_{i+1})$. For $x \in U$ and $i \in \{0, \cdots, k_a\}$, the old decision system will be replaced by a new one after discretization[11,12].

Discretization is essentially to divide the codomain of condition attributes into amount of intervals with selected breakpoints. Each interval is corresponding to a discrete value generally in form of an integer. All values of old attributes which belong to the same interval will be combined into a discrete value. As a result, the process of discretizaiton is the process of breakpoint selection. There are many methods to measure a discretization scheme to the decision table including the precision of classification, the number of breakpoint, the length of rule, the number of rule, the entropy of condition information, etc. [13]

C. Fuzzy C-Mean Cluster Algorithm

The REBSC algorithm in this paper discretizes samples via two stages. It mainly solves the significance of each attribute in one stage and extracts rules in another. The fuzzy C-Mean (FCM) cluster algorithm is organized to discretize samples in 1st stage. The number of cluster center is regarded as the number of decision classification. The cluster analysis is an important method of non-supervision study, its purpose is to divide given data into a certain number of meaningful clusters and to be most similar among intra-cluster objects but most diverse among inter-cluster objects [14-16]. The FCM cluster algorithm, which is designed simply and realizably, is a widely applicable method of attribute fuzzification. FCM has been applied in many areas successfully.

The FCM cluster algorithm ascertains a sample belong to a kind of cluster with the degree of membership. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a sample space. We divide X into m classifications, where m is a positive integer which is bigger than one. X can be denoted by a fuzzy matrix $U = (u_{ij})$, where u_{ij} is the degree of membership for the sample j belonging to the cluster center i. The object function of FCM is defined as

$$\min J_m(\mu, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} \mu_{ij}^m d_{ij}^2 \quad (4)$$

Where

$u_{ij} \in [0,1], i \in [1,c], j \in [1,n]$,

$\sum_{i=1}^{c} \mu_{ij} = 1, \sum_{j=1}^{n} \mu_{ij} \in (0,n)$

We can educe two iterative formulas (5) and (6) for the object function with the lagrangian multiplier method.

$$c_i = \sum_{j=1}^{n} \mu_{ij} x_j \Big/ \sum_{j=1}^{n} \mu_{ij}^m \quad (5)$$

$$\mu_{ij} = 1 \Big/ \sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}} \quad (6)$$

Here c is the number of clusters, n is the number of samples, c_i is the center of fuzzy group i and d_{ij} is the euclidean distance between the cluster center i and the sample j. $m \in [1, \infty)$ is the fuzzy weighted index. We can achieve the optimized fuzzy division U^* with above iterative formulas in FCM algorithm, where

$U^* = [\mu_{ij}^*]_{n \times c}$ [17, 18].

D. *Attribute Significance*

The relevance, between condition attributes and decision attributes, reflects the significance of condition attributes in a decision table. Therefore, the number of potential values of decision attributes indicates the significance of condition attribute relative to a decision attribute when a condition attribute has a value. If potential values of decision attributes are unique when a condition attribute gets a value θ, the value of condition attribute can ascertain the decision attribute uniquely. As a result, we need not take into account other condition attributes whenever condition attributes own the value θ in the process of rule generating.

Definition 4: $M_a = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{l_i}$ is the significance of the attribute a in a decision system, where $a \in A$, n is the radix of $V_a\{V_{a,1}, \cdots, V_{a,n}\}$, l_i is the number of potential values of the decision attribute when a is $V_{a,i}$.

The definition 4 shows that bigger value of M_a is in a decision system, stronger the decision capability of the attribute a is. Whenever a condition attribute in $\{a\}$ is $V_{a,i}$ and the same value for all x and y in U are consistent in relation to $\{a\}$, then we give the maximum value one to M_a. The corresponding condition attribute can be regarded as the unique condition attribute in decision rules.

III. ALGORITHM DESCRIPTIONS

A. *Primary Idea of Algorithm*

In a decision system, decision rules generally are stronger relative to condition attributes that are more significant. It means that attributes with higher significance are stronger in terms of decision capability. Our REBSC algorithm in this paper discretizes samples via two stages. The purpose of the first stage is to discretize continuous attributes in a decision system with FCM in which the number of cluster centers is the number of decision classifications. And then we can ascertain the significance for every condition attribute according to definition 1. Sort all condition attributes in descending order on their significances. In the second stage, each of discretized attribute value is invariant for the condition attribute which owns the highest significance. Other attributes divide intervals again in descending order of significances based on classification objects. Finally, we can achieve a decision table and decision rules after removing repetitive objects.

B. *Descriptions of Algorithm*

Condition attributes are only discretized with their own features in local discretization algorithm. We normally need to reduce attributes further so as to get the eventual rule set. The REBSC algorithm given in this paper belongs to global discretization algorithms. It fully considers the significance for every condition attribute and the core of decision classification in the process of discretization. As a result, the number of generated decision rules decreases without attribute reduction.

Algorithm of rule extracting based on attribute significance and decision classification

Input: training sample set D;

Output: the decision rule table Dtable;

Let $S = (U, A, \{V_a\}, f)$ be a decision knowledge system, n is the number of condition attributes, $\{d\}$ is a decision attribute set and Dtable is empty initially, $S' = (U', A, \{V'_a\}, f')$ is the same as S in structure;

Step 1 Organize the sample set D into the decision system S after preprocessing;

Step 2 $S_0 = S$, $i = 1$, S' is empty and B is empty;

Step 3 Discretize every condition attribute in S with FCM. k is the number of cluster centers which is the number of decision classifications;

Step 4 Calculate the significance for every attribute in S. Sort all condition attributes in descending order on their significances. Suppose the order of condition attributes is $C_1, C_2, \cdots, C_n$;

Step 5 Assign the column of attribute C_1 and decision attribute in S to S';

Step 6 Add C_i to B. Let $X_1, X_2, \cdots, X_t$ be the partition of U' divided by $U'/IND(B)$. Remove all partitions, which have no inconsistent objects, from S' and add all objects in them to Dtable. Suppose the remained partition is $X'\{X'_1, X'_2, \cdots, X'_{t'}\}$. At the same time, find out those objects corresponding to partitions removed from S' and remove them from S_0. Go to 15 if S' is empty;

Step 7 $i = i + 1$, go to 15 when $i > n$;

Step 8 Add the column of C_i in S_0 to S';

Step 9 Let the partition for U divided by $U/IND(\{d\})$ be $XD\{Xd_1, Xd_2, \cdots, Xd_k\}$.

Step 10 Calculate intersections between each subset $X'_j\ (j = 1,2,\cdots,t')$ in X' and that in $XD\{Xd_1, Xd_2, \cdots, Xd_k\}$ .Get the partition $Int_j\{\mathrm{int}_{j1}, \mathrm{int}_{j2}, \cdots, \mathrm{int}_{jk}\}(j = 1,2,\cdots,t')$ .Suppose the partition set to all subsets of X' is $INT\{Int_1, Int_2, \cdots, Int_{t'}\}$;

Step 11 For every

$Int_j\{\text{int}_{j1},\text{int}_{j2},\cdots,\text{int}_{jk}\}(j=1,2,\cdots,t')$ do step 12 to step 15

Step 12 Calculate the maximum value and the minimum value of all objects in $\text{int}_{j1},\text{int}_{j2},\cdots,\text{int}_{jk}$ respectively and then generate partition intervals, where the minimum value is the left end point and the maximum value is the right end point in every partition interval. Suppose the codomain of C_i is divided into $Part_j\{Part_{j1},Part_{j2},\cdots Part_{jk}\}$. We can get k intervals totally;

Step 13 Sort the interval sequence $Part_{j1},Part_{j2},\cdots Part_{jk}$ in ascending order on the position of left end point belonging to each interval.

Step 14 Select two intervals which have intersection but common endpoint and lie foremost in order of sorted intervals. Redivide intervals front-to-rear. That is to say, every two interfacing end points can ascertain an interval. Sort the interval sequence in ascending order on the position of left end point belonging to each interval again.

Step 15 Repeat 14 until any two intervals have no intersection but common endpoint

Step 16 For every $Part_j(j=2,\cdots,t')$ select every interval according on the left endpoint of the interval in ascending order . Insert the selected interval to $Part_1$ with the interval Combination method in C section.

Step 17 If there is no common end point between two neighbor intervals of $Part_1$, then treat the mean value of neighbor end points as the common end point. Replace discretized intervals of C_i with discretized values. Update the value of attribute C_i in S' with a corresponding discretized value according to its interval and go to step 6;

Step 18 Delete all repetitive objects in Dtable.

C. interval Combination method

Combine $Part_j(j=2,\cdots,t')$ to $Part_1$. Select one interval in $Part_j$. Suppose the interval is $Part_{jp}$ .There are four cases for $Part_{jp}$ and $Part_{1k}$ which is conterminous with $Part_{jp}$. How to combine $Part_{jp}$ with $Part_1$ is described as following. There are four different situations (shown in Fig. 1) when descript the relationship between $Part_{jp}$ and a pair of intervals $Part_{1k}$ and $Part_{1,k-1}$ in $Part_1$. Firstly set the label not to be updated for each interval in $Part_1$ before $Part_{jp}$ being combined. Suppose $Part_{1,end}$ to be the last interval of $Part_1$.This is to say $Part_{1,end}$ has the biggest value of left end point in all $Part_1$'s intervals. Each interval is described as $part_{jp}=[l_p,r_p]$, $part_{1,k-1}=[l_{k-1},r_{k-1}]$, $part_{1,k}=[l_k,r_k]$, $part_{1,end}=[l_{end},r_{end}]$. We label every interval with $flag$ (e.g., $Part_{1k}$ is not updated if $flag(part_{1k})=0$, or else $flag(part_{1k})=1$). Four different situations of combining $Part_{jp}$ into $Part_1$ are described as following.

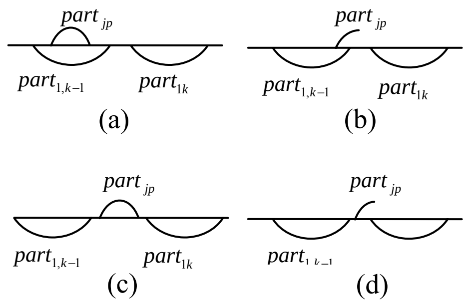

Fig. 1 Situations of Combination $part_{jp}$ into $part_1$

(a) Let $flag(part_{1,k-1})=1$ if $flag(part_{1,k-1})=0$, or else divide $Part_{1,k-1}$ into $[l_{k-1},l_p)$ and $[l_p,r_{k-1}]$, number all intervals of $Part_1$ again sorting by the left end point of each interval in ascending order. The interval $[l_p,r_{k-1}]$ will be the interval k and set $flag(part_{1k})=1$. There isn't $Part_{1,k-1}$ when $k=1$, so update $Part_{1k}$ with $[l_p,r_k]$ and set $flag(part_{1k})=1$

(b) Update $Part_{1,k-1}$ with $[l_{k-1},r_p]$, that is $r_{k-1}=r_p$, if $r_p<=l_k$. At the same time, divide $Part_{1,k-1}$ into $[l_{k-1},l_p)$ and $[l_p,r_{k-1}]$ with the left end point of $Part_{jp}$ if $flag(part_{1,k-1})=1$, number all intervals of $Part_1$ again sorting by the left end point of each interval in ascending order. If $r_p>l_k$ and r_p belongs to some one interval of $Part_1$, label the interval to be updated.

(c) If $flag(part_{1,k-1})=0$, update $Part_{1,k-1}$ with $[l_{k-1},r_p]$, or else update $Part_{1k}$ with $[l_p,r_k]$ and set $flag(part_{1k})=1$.

(d) Update $Part_{1k}$ with $[l_p, r_k]$. If the right end point of $Part_{jp}$ belongs to some one interval of $Part_1$, label the interval to be updated, or else label the left neighbor interval to the right end point of $Part_{jp}$ in $Part_1$ to be updated. If $k-1=end$ (i.e., $Part_{1,k-1}$ is the last interval of $Part_1$), insert $Part_{jp}$ into the last interval position of $Part_1$ directly.

IV. Experiment Results

A. Experiment 1

Our experiment is based on Matlab7.0. Table □ shows some instances on turbine failure diagnosis from [19]. There are totally eleven omens.

TABLE Ⅰ.
INSTANCES OF TURBINE FAILURE DIAGNOSIS

S	Omens											Failure
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	d
1	0.8	0.001	0.1	0.1	1	0.001	1	0.1	0.9	1	0.001	1
2	0.8	0.001	0.1	0.1	0.8	0.001	0.8	0.1	0.8	0.9	0.001	1
3	0.5	0.001	0.1	0.1	0.8	0.001	1	0.1	0.7	0.9	0.001	1
4	0.8	0.001	0.2	0.2	1	0.001	1	0.1	0.9	1	0.001	1
5	0.5	0.001	0.1	0.1	0.8	0.001	0.9	0.1	0.7	0.9	0.001	0
6	0.5	0.9	0.001	0.8	0.001	0.8	0.1	0.9	0.5	0.1	0.8	0
7	0.5	0.9	0.001	0.8	0.001	0.8	0.1	0.5	0.5	0.2	0.9	0
8	0.6	0.7	0.001	0.9	0.001	0.5	1	0.8	0.4	0.1	0.7	0
9	0.4	0.7	0.001	0.7	0.001	0.5	0.1	0.7	0.3	0.1	0.6	1
10	0.3	0.9	0.001	0.9	0.001	1	0.001	0.8	0.1	0.001	0.9	0
11	0.2	0.7	0.001	0.8	0.001	1	0.001	0.8	0.001	0.001	0.8	0
12	0.2	0.6	0.001	0.6	0.001	0.9	0	0.7	0.001	0.001	0.6	0
13	0.4	0.4	0.3	0.6	0.001	0.001	0.1	0.1	0.2	0.001	0.1	0
14	0.4	0.5	0.3	0.7	0.05	0.001	0.001	0.1	0.1	0.001	0.1	0
15	0.4	0.6	0.4	0.9	0.001	0.8	0.001	0.3	0.1	0.001	0.9	0
16	0.3	0.8	0.3	1	0.001	1	0.001	0.1	0.001	0.001	1	0
17	0.3	0.4	0.3	1	0.001	1	0.001	0.1	0.001	0.001	1	0
18	0.6	0.3	0.9	0.3	0.3	0.001	0.001	0.001	0.001	0.001	0.6	0
19	0.7	0.3	0.9	0.3	0.2	0.001	0.001	0.001	0.001	0.001	0.8	0
20	0.7	0.6	0.9	0.6	0.001	0.001	0.2	0.5	0.3	0.001	0.9	1
21	0.7	0.6	0.9	0.7	0.001	0.001	0.3	0.6	0.4	0.001	0.8	1

The decision table, generated by REBSC algorithm, is shown in table II. Experiment results indicate the following conclusions:

The significance is same for each condition attribute of samples. There are only six condition attributes in the decision table in the form of $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ and seven breakpoints (intervals are 13). The whole rules are ten with average lengths to 3.8.The result is shown in table III. Decision attributes drop 40 percent comparing with [19]. At the same time, intervals reduce 60 percent with three decision rules increasing. The REBSC algorithm in this paper can hold something common for samples and its classification is more reasonable.

TABLE II
DECISION RULE TABLE

s_1	s_2	s_3	s_4	s_6	s_7	d
2	1					1
1	1	1	1	1	2	1
1	1	1	1	1	1	0
1	3	1	2	2		0
2	3	1				0
1	3	1	2	1		1
1	3	1	1			0
1	2					0
2	2					0
2	3	2				1

TABLE Ⅲ
CORRESPONDING RELATIONSHIP BETWEEN THE DISCRETE VALUE AND THE DISCRETE INTERVALS FOR EACH CONDITION ATTRIBUTE

	Discrete Value	1	2	3
s_1	interval	(-∞,0.55)	[0.55,∞)	
s_2	interval	(-∞,0.3)	[0.3,0.6)	[0.6,∞)
s_3	interval	(-∞,0.4505)	[0.4505,∞)	
s_4	interval	(-∞,0.7)	[0.7,∞)	
s_6	interval	(-∞,0.65)	[0.65,∞)	
s_7	interval	(-∞,0.95)	[0.95,∞)	

B. Experiment 2

1. Descriptions of Data Set

In order to validate the performance of our algorithm, we have carried out a series of experiments with data samples in UCI which is a standard data set on machine learning. The description of experimental data set is shown in table IV.

TABLE Ⅳ
DESCRIPTION OF EXPERIMENTAL DATA

Data	Number of Condition Attribute	Number of Classification	Number of Sample
diabetes	8	2	768
ionosphere	34	2	351
iris	4	3	150
wine	13	3	178
glass	9	6	214

2. Experiment Analysis

We have completed ten experiments independently and obtained the average value through 10-fold cross validation technology. And we have verified the validity of our REBSC algorithm from five aspects, shown in table 5, including the accuracy, average breakpoints, and the average length of rule, average rules and the standard deviation. Experiment results indicate that the REBSC algorithm can obtain 99.71% recognition rate in ionosphere data set. We have got better result from other data sets and tested the validity of our algorithm.

TABLE V
EXPERIMENT RESULT OF REBSC

Data set	Accuracy	Average Breakpoints	Average Length of Rule	Average Rules	Standard Deviation
iris	0.9553	6.4200	2.8651	14.7500	4.99
wine	0.8829	21.27	3.1584	63.7700	7.23
Ionosphere	0.9971	48.81	17.1401	54.2700	1.03
diabetes	0.8282	111.3600	6.0148	512.1100	4.61
glass	0.9400	68.0400	6.6548	129.3400	5.38

Following are further analyses of experiment results for our algorithm. We have analyzed 100 times' running results for different data set and primarily discussed the relationship between breakpoints and the average recognition rate. Fig.2 (a-f) show relationships between breakpoints and the average recognition rate on six different data sets respectively, including iris, wine, ionosphere, diabetes, glass and breast. Wave forms in Fig. 1 indicate that the recognition rate rises with breakpoints in some intervals whereas it diminishes with breakpoints in some intervals. It further demonstrates the importance of reasonable breakpoint division (discretization of attribute). We would neglect something common among samples if breakpoints were divided too slightly. On the contrary, we might conceal differences among samples if breakpoints were divided too roughly. Therefore, we would achieve higher recognition rate if breakpoints were decreased properly, which is similar to the conclusion of [20] about that the discretized intervals should not hide patterns.

.

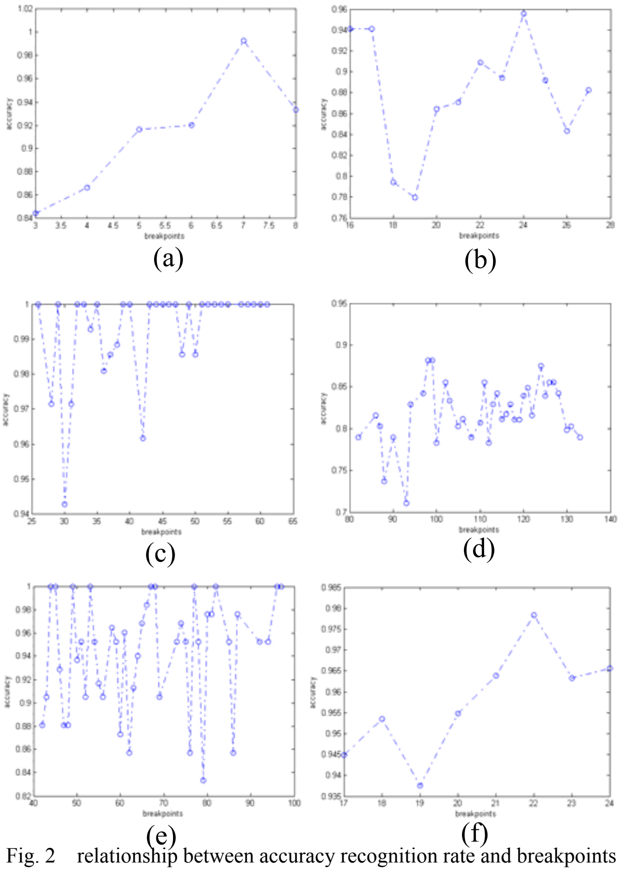

Fig. 2 relationship between accuracy recognition rate and breakpoints on different data set

V. CONCLUSIONS

We propose a novel approach for rule extracting based on attribute significance and decision classification (REBSC). The approach concentrates on the classification object and makes the most of the significance for each condition attribute and classification objects during classification, generates minimum decision rules without attribute reduction but stronger decision ability. We have fully proved the rationality and validity for REBSC in experiment 1 and 2, respectively, and tested on five different UCI data sets. We also demonstrate the importance of breakpoint division.

ACKNOWLEDGEMENT

This paper is supported by (1) the National Natural Science Foundation of China under Grant No. 60873146、60973092、60903097, (2) Program Project of Science and Technology of Jilin Education Ministry of China under Grant No.2007-172 (3) the Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China.

REFERENCES

[1] Li yong-min, Zhu shan-juan, Chen xiang-hui, etc, Data mining model based on rough set theory, J T singhua Univ(Sci&Tech),vol.39, no.1,pp.110-113,1999.
[2] Feng lin, Wang guo-yin and Li tiang-rui, Knowledge Acquisition from Decision Tables Containing Continuous-Valued Attributes, ACTA ELECTRONIC SINICA, vol.37,no.11,pp. 2432-2438,2009.
[3] Wang li-hong and Wu geng-feng, Measurements of Discretization Schemes, PR&AI, vol.21, pp. 494-499,2008..
[4] Zhao jing-xian, Ni chun-peng, Zhan yuan-rui and Du zi-ping. Efficient discretizathion algorithm for continuous attributes, Systems Engineering and Electronics,vol.31,pp.195-199,2009.
[5] Pawlak Z. Rough sets, International Journal of Information and Computer Science, vol.11,pp. 341-356,1982.
[6] Dongbo ZHANG and Yaonan WANG,A new ensemble feature selection and its application to pattern classification, J control Theory Appl,vol.7,pp. 419-426,2009.
[7] Dongbo ZHANG and Yaonan WANG,A new ensemble feature selection and its application to pattern classification, J Control Theory Appl, vol.7,pp.419-426, 2009.
[8] Yang Liu, Guohua Bai, and Boqin Feng, Multi-agent Based

Multi-knowledge Acquisition Method for Rough Set, G.Wang etal.(Eds): RSKT 2008, LNAI 5009,pp.140-147,2008. Springer-Verlag Berlin Heidelberg 2008.
[9] YANG Ming, A Novel Algorithm for Attribute Reduction Based on Consistency Criterion ,CHINESE JOURNAL OF COMPU TERS, vol.33,pp. 231-239, 2010.
[10] JUNHONG WANG, JIYE LIANG and YUHUA QIAN, UNCERTAINTY MEASURE OF ROUGH SETS BASED ON A KNOWLEDGE GRANULATION FOR INCOMPLETE INFORMATION SYSTEMS, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol.16 ,pp. 233-244, 2008.
[11] YUE Hai-liang and YAN De-Qin, New Algorithm for Discretization Based on Information Entropy, Computer Science, vol.37,pp.231-237, 2010.
[12] Zhang jun-ming and Qiu dong, Discretization method based on breakpoint importance in roughness set theory, Meteorological, Hydrological and Marine Instruments, No.1, pp.44-47, 2009.
[13] Wang li-hong and Wu geng-feng, Measurements of Discretization Schemes, PR & AI, vol.21,pp. 494-499, 2008.
[14] Huan Yan-xin, Zhou chun-guang, Yang guo-hui，etc, Identification of teataste signals based on rough set theory, JOURNAL OF JILIN UNIVERSITY(INFORMATION SCIENCE EDITION),vol. 20,pp.73-77, 2002.
[15] XU Xiang, ZHANG Dong-bo, HUANG Hui-xian and LIU Zi-wen, Knowledge acquisition ba sed on improved PSO a lgor ithm and entropy , Journal of Computer Applications,vol.29,pp.2245-2249, 2009.
[16] Liu si-yuan, Jiang wan-lu and Niu hui-feng, Multi-variable decision tree construction method based on principal component analysis and rough set, Systems Engineering and Electronics,vol.30,pp.1133-1137, 2009.
[17] LIU Kun-peng and LUO Ke, Improved fuzzy C-means clustering algorithm Computer Engineering and Applications, vol.45,pp.97-98, 2009.
[18] QI Hong-Yu1, WU Xiao-Jun,WANG Shi-Tong and YANG J ing-Yu, Collaborative FCPM Fuzzy Clustering Algorithm, PR&AI, vol.23,pp.120-126, 2010.
[19] WANG Ke, ZHU Qi-bing and CUI Bao-tong, Method of discretization of continuous attributes of decision table, Computer Engineering and Applications, vol.44, pp.148-149, 2008.
[20] WANG Li-hong, SUN Li-min and MENG Jia-na, Correlation Between Granular Entropy and Classification Accuracy in Discretization, Journal of Chongqing University, vol.31,pp.57-66, 2008.