Fusing complete monotonic decision trees

Hang Xu, Wenjian Wang, and Yuhua Qian, *Member, IEEE*

Abstract—Monotonic classification is a kind of classification task in which a monotonicity constraint exist between features and class, i.e., if sample x_i has a higher value in each feature than sample x_j, it should be assigned to a class with a higher level than the level of x_j's class. Several methods have been proposed, but they have some limits such as with limited kind of data or limited classification accuracy. In our former work, the classification accuracy on monotonic classification has been improved by fusing monotonic decision trees, but it always has a complex classification model. This work aims to find a monotonic classifier to process both nominal and numeric data by fusing complete monotonic decision trees. Through finding the completed feature subsets based on discernibility matrix on ordinal dataset, a set of monotonic decision trees can be obtained directly and automatically, on which the rank is still preserved. Fewer decision trees are needed, which will serve as base classifiers to construct a decision forest fused complete monotonic decision trees. The experiment results on ten datasets demonstrate that the proposed method can reduce the number of base classifiers effectively and then simplify classification model, and obtain good classification performance simultaneously.

Index Terms—Monotonic classification, decision tree, ensemble learning, feature selection, discernibility matrix.

1 Introduction

Classification is one of important research issues in machine learning and data mining. From the viewpoint of constraints among feature values, classification tasks can be regarded as two types: nominal classification and ordinal classification. For an ordinal classification task, the ordinal relationship between different class labels should be taken into account [1], [2]. Monotonic classification is a special ordinal classification task, where the class values are ordinal and discrete, and there is a monotonicity constraint between features and class [3]. The monotonicity constraint indicates that if sample x_i has a higher value in each feature than sample x_j, it should be assigned to a class with a higher level than the level of x_j's class [4]. Monotonic classification is a common task, which has attracted increasing attention from domains of data mining, knowledge discovery, pattern recognition, intelligent decision making, and so on.

There are many monotonic classification tasks in real-life. For example, evaluating a university's comprehensive ability is such a problem. In this problem, scientific research ability, teacher quality and teaching level are three important indicators, and in the scores of these indicators an ordinal relation exists obviously. The evaluation of university's comprehensive ability has three levels–"high, medium, low", among which an ordinal relation exists. There is a monotonicity constraint between the features (scientific research ability, teacher quality and teaching level) and class (the evaluation level of university's comprehensive ability) as follows: If a university A has a higher scores in these three features than another university B, university A will have a higher level in the evaluation level than university B. In addition, there are many problems with the same characteristics as follows: consumers select commodities in a market according to their price and quality; employers select their employees based on their education and experience; investors select stocks or bonds in terms of their probability of appreciation or risk; universities select scholarship students according to students performances; editors make a decision on a manuscript according to its quality; and so on. This kind of problem is monotonic classification.

Typical classification methods, neural networks, support vector machine, decision tree, etc., are not fit for solving monotonic classification problems because they do not consider the monotonicity constraint between features and class [4]. Therefore, special methods for monotonic classification task need to be designed [1], [2]. Monotonic classification problems are widespread in real-life world, but compared with general classification problems, much less attention has been paid to monotonic classification these years. At present, some effective results for monotonic classification have been reported, and they can be roughly classified into two kinds of methods: First, some theoretic frameworks for monotonic classification have been developed, such as rule-based classifiers [5]–[10], set-valued and interval ordered information systems [11], [12] and ordered entropy model [13]. These methods always got few consistent rules because they produced much larger classification boundary on practical works [14]. Second, some algorithms for learning monotonic decision model were designed [15]–[18], like ordinal learning model [19], modified nearest neighbor algorithm [18], ranking impurity [20] and ordinal decision trees [21]–[25]. They can improve the performance of extracting ordinal information, but they can not ensure the monotonicity of a decision tree learned from a training dataset with a monotonicity constraint. These two kinds of methods were reviewed and analyzed in Ref [26] in detail.

Ref [4] presented a rank entropy based monotonic decision tree (REMT) algorithm to reduce the influence of noisy data and obtain decision rules with clear semantics, which can get a monotonic decision tree if training samples are from a monotonic dataset. Although REMT is robust and understandable, its generalization ability is limited. Ref

• *The authors are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi Province, China. E-mail: xuh102@126.com; wjwang@sxu.edu.cn; jinchengqyh@126.com.*

Manuscript received – –, ––; revised – –, ––.

[26] proposed a fusing monotonic decision trees (FREMT) algorithm which combined decision trees with ensemble learning technique. The method had obvious effect on improving the classification performance. However, it may not obtain complete feature subsets (the complete feature subsets mean all the feature subsets of original feature set under a given condition) under one variable precision parameter value due to its adopted heuristic search strategy, which results in a series of variable precision parameter values to construct base decision trees are needed. The performance of FREMT has a large fluctuation along the variation of variable precision parameter values and no single one can make the classification performance good enough. This is because the parameter is introduced into the definition of classes' lower and upper approximation sets under the rough set frame, and the "belong to under a strict sense" in computing lower and upper approximations will relax to the "belong to under the sense of a precision". When the variable precision parameter value is zero, the upper and lower approximations become strict, which will lead to fewer samples in lower approximation set and more samples in upper approximation set. Then the union set of all classes' boundary domain always approaches the whole dataset, which may result in an almost zero significance on most features. Like this, the variable precision value can affect the number of samples in upper and lower approximation set, and further the value of feature significance. So if FREMT only runs under one variable precision value, the number of feature subsets may be too few to get a good classification model. The FREMT usually obtains a large number of base classifiers in multiple variable precision values in order to get a higher accuracy, which results in complicated classification model. So for monotonic classification problem, it is needed to build a simplified classification model with good classification performance through learning from a set of samples with class labels. Therefore, it is significant to obtain a comparable or even better classification performance with fewer base classifiers.

To address this issue, it is necessary that complete feature subsets should be achieved only under one variable precision parameter value. It has four reasons: (1) The completeness of feature subsets is helpful to improve the classification accuracy of monotonic classification. (2) Because the existing methods can not get enough feature subsets on one variable precision parameter value, more feature subsets need to be found under various variable precision parameter values, which may lead to redundancy of feature subsets. (3) Too many feature subsets obtained by the existing methods make their running take up more storage space and spend more computational cost. (4) The redundant feature subsets may lead to the over-fitting of an algorithm and the weakening generalization ability. Besides heuristic search strategy, the discernibility matrix method can also be used to obtain feature subsets [27]–[30]. But the existing feature selection algorithms based on discernibility matrixes can only be used in general data and not consider the ordinal relations. In this work, we define a discernibility matrix on ordinal dataset and then obtained complete feature subsets. Then we propose a method of fusing complete monotonic decision trees, namely FCMT, which omits the process of selecting decision trees and determining the number of decision trees. A set of monotonic decision trees can be obtained directly and automatically, and they will serve as base decision trees to construct a decision forest. Although it includes fewer number of trees, rank is still preserved which can ensure monotonically consistent rules. The FCMT method can reduce the number of base classifiers effectively and then simplify classification model, and obtain good classification performance simultaneously.

The rest of the paper is organized as follows. In Section 2, the preliminaries on discernibility matrix and monotonic decision tree are introduced. In Section 3, we explain how to construct discernibility matrix and the FCMT method in detail. Experimental results and analysis are presented in Section 4. Finally, we give some conclusions about this paper in Section 5.

2 PRELIMINARIES

To illustrate the proposed method clearly, some basic concepts, such as dependency and feature selection on a ordinal dataset, discernibility matrix, discernibility function and REMT algorithm, are introduced briefly in this section.

2.1 Feature selection on ordinal dataset

Let $U = \{x_1, ..., x_n\}$ be a set of samples and C be a set of features to describe the samples; d is a class. For the features and class, if there is a superior sequence relationship between the values of samples, the dataset $OD = \{U, C \cup \{d\}\}$ will be an ordinal dataset.

Definition 1. [31] Given an ordinal dataset $OD = \{U, C \cup \{d\}\}$, $B \subseteq C$, the range of d is $\{d_1, d_2, ..., d_t\}$, where $d_1 \prec d_2 \prec ... \prec d_t$ ($d_1 \prec d_2$ means d_1 is dominated by d_2), the dominance relations on discourse domain U are as follows:

$$R_B^{\geq} = \{(x_i, x_j) \in U \times U | f(x_i, c) \geq f(x_j, c), \forall c \in B\} \quad (1)$$

$$R_B^{\leq} = \{(x_i, x_j) \in U \times U | f(x_i, c) \leq f(x_j, c), \forall c \in B\} \quad (2)$$

$$R_{\{d\}}^{\geq} = \{(x_i, x_j) \in U \times U | f(x_i, d) \geq f(x_j, d)\} \quad (3)$$

$$R_{\{d\}}^{\leq} = \{(x_i, x_j) \in U \times U | f(x_i, d) \leq f(x_j, d)\} \quad (4)$$

where $f(x_i, c)$ is the value of x_i in feature $c (c \in B)$, and $f(x_i, d)$ is the value of x_i in class d.

Definition 2. [4] Let $OD = \{U, C \cup \{d\}\}$ be an ordinal dataset, $B \subseteq C$. If $\forall (x_i, x_j) \in U$, $(x_i, x_j) \in R_B^{\geq} \Rightarrow (x_i, x_j) \in R_{\{d\}}^{\geq}$, we say OD is B-monotonically consistent.

When the ordinal dataset satisfies the monotonically consistency between its feature set and class, it is a monotonic classification.

Denote the set which dominates x and the set dominated by x as follows:

$$[x]_B^{\geq} = \{y \in U | f(y, c) \geq f(x, c), \forall c \in B\} \quad (5)$$

$$[x]_B^{\leq} = \{y \in U | f(y, c) \leq f(x, c), \forall c \in B\} \quad (6)$$

The lower approximation and upper approximation of the set which dominates d_i are defined as follows:

Definition 3. [31] Let $d_i^{\geq}$ be a sample set whose class is no worse than class d_i. The lower approximation and upper approximation are:

$$\underline{R_B^{\geq}} d_i^{\geq} = \{x \in U | [x]_B^{\geq} \subseteq d_i^{\geq}\} \tag{7}$$

$$\overline{R_B^{\geq}} d_i^{\geq} = \{x \in U | [x]_B^{\leq} \cap d_i^{\geq} \neq \varnothing\} \tag{8}$$

The feature dependency on an ordinal dataset is defined as follows.

Definition 4. [32] Given an ordinal dataset $OD = \{U, C \cup \{d\}\}$ and $B \subseteq C$, the monotonic dependency of d respect to B is defined as:

$$\gamma_B(d) = \frac{|U - \bigcup_{i=1}^{t} BND_B(d_i)|}{|U|} \tag{9}$$

Where t is the number of classes, $BND_B(d_i)$ is the class boundary of d_i in terms of feature set B, $BND_B(d_i) = BND_B(d_i^{\geq}) = BND_B(d_i^{\leq}) = \overline{R_B^{\geq}} d_i^{\geq} - \underline{R_B^{\geq}} d_i^{\geq}$.

According to different purposes, feature selections on different senses were defined [33], [34]. Ref [35] defined an feature selection on ordinal dataset based on the feature dependency.

Definition 5. [35] Given an ordinal dataset $OD = \{U, C \cup \{d\}\}$ and $B \subseteq C$, the feature selection on an ordinal dataset is that from feature set C we selected the feature subset B which satisfies two conditions:
(1) Sufficiency condition: $\gamma_B(d) = \gamma_C(d)$;
(2) Necessity condition: $\forall c \in B, \gamma_{B-\{c\}}(d) < \gamma_C(d)$.

2.2 Discernibility matrix

Ref [28] defined the discernibility matrix and discernibility function on information system.

Definition 6. Given an information system $IS = (U, C)$, the discernibility matrix of the information system IS is $n \times n$ matrix, which is defined as $M^I = \{m_{ij}^I\}$, where $m_{ij}^I = \{c \in C | f(x_i, c) \neq f(x_j, c)\}$.

Definition 7. Given an information system $IS = (U, C)$, the discernibility matrix $M^I = \{m_{ij}^I\}$, $m_{ij}^I = \{c \in C | f(x_i, c) \neq f(x_j, c)\}$, then the discernibility function of a discernibility is defined as:

$$f(M)^I = \bigwedge \Big\{ \bigvee (m_{ij}^I) | \forall x_i, x_j \in U, m_{ij}^I \neq \varnothing \Big\} \tag{10}$$

Then Ref [27] extended the definition of discernibility matrix from information system to dataset as follows:

Definition 8. Let $D = (U, C \cup \{d\})$ be a consistent dataset. Then the class-relative discernibility matrix is defined as $M^D = \{m_{ij}^D\}$, where

$$m_{ij}^D = \begin{cases} \{c \in C | f(x_i, c) \neq f(x_j, c)\}, & f(x_i, d) \neq f(x_j, d) \\ m_{ij}^I, & otherwise \end{cases} \tag{11}$$

2.3 REMT

Hu et al. [4] proposed a monotonic decision tree algorithm REMT, which had a good robustness and could solve the conflict between the monotonicity and generalization ability to a certain extent. The method can generate a rule-set which is simple and easy to understand. The REMT algorithm is listed in Algorithm 1.

Algorithm 1 REMT

Require: criteria: features of samples; decision: class labels of samples; ε: stopping criterion;
Ensure: a monotonic decision tree T.
1: generate the root node.
2: if the number of samples is 1 or all samples are from the same class, the branch stops growing.
3: otherwise,
4: **for** each feature a_i, **do**
5: **for** each $c_j \in V_{a_i}$ (V_{a_i} is the domain of value of a_i), **do**
6: divide samples into two subsets according to c_j,
7: **if** $f(a_i, x) \leq c_j$ **then**
8: put x into one subset, and set $f(a_i, x) = 1$,
9: **else**
10: put x into the other subset , and set $f(a_i, x) = 2$.
11: **end if**
12: denote now a_i with respect to c_j by $a_i(c_j)$, compute $RMI_{c_j} = RMI^{\geq}(\{a_i(c_j)\}, \{d\})$.
13: **end for** j
14: $c_j^* = arg \max_j RMI_{c_j}$.
15: **end for** i
16: select the best feature a^* and the corresponding point c^*:
$(a^*, c^*) arg \max_i \max_j RMI^{\geq}(\{a_i(c_j)\}, \{d\})$.
17: if $RMI^{\geq}(\{a^*\}, \{d\}) < \varepsilon$, then stop.
18: build a new node and split samples with a^* and c*.
19: recursively produce new splits according to the above procedure until stopping criterion is satisfied.

3 Fusing Complete Monotonic Decision Trees

In this section, we will explain the proposed FCMT method in detail. A discernibility matrix for the ordinal dataset is generated, and then we get a set of complete feature subsets through the matrix. Based on the subsets, a complete monotonic decision forest for monotonic classification are obtained.

Similar to the general classification problem, the training samples for the monotonic classification problem also have some redundant or unrelated features which may affect performance of classify model and the decision maker's understanding to the essence of problem. So more important information in a dataset could be extracted by obtaining the feature subset on ordinal dataset. The selected feature subset should keep the same approximation ability for classification results as original feature set, that is to say, the dependency of class respecting to feature subset and the dependency of class respecting to original feature set should be identical. Moreover, redundant features should not be include in selected feature subset.

Here a method of feature selection based on discernibility matrix is proposed, which can obtain complete feature subsets under one variable precision parameter value only. Thus the number of feature sets can be reduced greatly and then the classification model could be simplified.

3.1 Discernibility matrix for ordinal dataset

Although some discernibility matrix constructing approaches were proposed, most of them were defined on the general datasets and they can't be used in the ordinal dataset directly. An discernibility matrix for ordinal dataset is proposed in this subsection.

We firstly divide samples in an ordinal dataset into two sets: monotonically consistent set and non-monotonic set.

Definition 9. Let $OD = (U, C \cup \{d\})$ be an ordinal dataset, C the feature set, $\{d\}$ the class. Under the variable precision parameter β, the monotonically consistent set U_M^β is defined as:

$$U_M^\beta = \begin{cases} \{x | \frac{|[x]_C^{\leq} \cap [x]_d^{\leq}|}{|[x]_C^{\leq}|} \geq 1-\beta\}, & if\ |[x]_C^{\geq}| < n_0 \\ \{x | \frac{|[x]_C^{\geq} \cap [x]_d^{\geq}|}{|[x]_C^{\geq}|} \geq 1-\beta\}, & if\ |[x]_C^{\leq}| < n_0 \\ \{x | \frac{|[x]_C^{\geq} \cap [x]_d^{\geq}|}{|[x]_C^{\geq}|} \geq 1-\beta, \frac{|[x]_C^{\leq} \cap [x]_d^{\leq}|}{|[x]_C^{\leq}|} \geq 1-\beta\}, & otherwise \end{cases} \tag{12}$$

Where $0 < \beta < 0.5$, n_0, which is much smaller than the size of dataset, is a constant not less than 1. That is, when the number of samples which dominates x is very small, we judge the monotonic consistency of x by considering the samples dominated by x. In the same way, when the number of sample set dominated by x is very small, we determine the monotonic consistency of x by considering the samples which dominates x. And the non-monotonic set U_{NM}^β is:

$$U_{NM}^\beta = U - U_M^\beta. \tag{13}$$

In consideration of the sample consistency, we define the discernibility matrix on ordinal dataset as follows.

Definition 10. Let $OD = (U, C \cup \{d\})$ be an ordinal dataset, C the feature set, $\{d\}$ the class. The discernibility matrix on ordinal dataset M^O is defined as Eqs. (14) and (15) (see the equations on next page).

In Eq. (15), μ_{ik} and μ_{jk} are as follows:

$$\mu_{ik} = \frac{|[x_i]_{\{c\}}^{\geq} \cap d_k^{\geq}|}{|[x_i]_{\{c\}}^{\geq}|} \tag{16}$$

$$\mu_{jk} = \frac{|[x_j]_{\{c\}}^{\geq} \cap d_k^{\geq}|}{|[x_i]_{\{c\}}^{\geq}|} \tag{17}$$

In Eq. (15), in order to preserve the monotonicity x_j will be replaced by x_j' when $x_i \in U_M^\beta$ and $x_j \in U_{NM}^\beta$. x_j' has the same feature values with x_j, but their class values are different. Define the probability that x_j belongs to the class which dominates d_k as follows:

$$P(d(x_j) \geq d_k) = \frac{|[x_j]_{\{c\}}^{\geq} \cap d_k^{\geq}|}{|[x_j]_{\{c\}}^{\geq}|} \tag{18}$$

Then, the probability that x_j belongs to the class d_k is:

$$\begin{aligned} P(d(x_j) = d_k) &= P(d(x_j) \geq d_k) - P(d(x_j) \geq d_{k+1}) \\ &= \begin{cases} \frac{|[x_j]_{\{c\}}^{\geq} \cap d_k^{\geq}|}{|[x_j]_{\{c\}}^{\geq}|} - \frac{|[x_j]_{\{c\}}^{\geq} \cap d_{k+1}^{\geq}|}{|[x_j]_{\{c\}}^{\geq}|}, & k = 1, 2, ..., t-1 \\ \frac{|[x_j]_{\{c\}}^{\geq} \cap d_k^{\geq}|}{|[x_j]_{\{c\}}^{\geq}|}, & k = t \end{cases} \end{aligned} \tag{19}$$

So the class value of x_j' is:

$$d_k^* = arg \max_k P(d(x_j) = d_k). \tag{20}$$

Note:

1) If the class values of two samples in the monotonically consistent set do not fulfill ordinal relation $R_{\{d\}}^{\leq}$, the features on which the two samples don't fulfill ordinal relation $R_{\{c\}}^{\leq}$ will be put in the discernibility matrix.
2) If x_i belongs to the monotonically consistent set while x_j belongs to the non-monotonic set, we will modify the class value of sample x_j with x_j' according to Eq. (20). When the class values of samples x_i and x_j' do not satisfy the ordinal relation $R_{\{d\}}^{\leq}$, the features on which the two samples don't satisfy the ordinal relation $R_{\{c\}}^{\leq}$ will appear in the discernibility matrix.
3) If both the two samples belong to the non-monotonic set, we will compute the μ values for each class $d_k^{\geq}$. μ_{ik} represents the probability that x_i belongs to the class which dominates d_k. If μ_{ik} is greater than μ_{jk} on each class $d_k^{\geq}$(which means x_i has a higher probability of belonging to a superior class value than x_j), then ordinal relation $R_{\{d\}}^{\leq}$ is not true for x_i and x_j. So discernibility matrix should include the features on which the two samples don't satisfy ordinal relation $R_{\{c\}}^{\leq}$.

By means of the definition of discernibility matrix, the corresponding ordinal discernibility function is defined.

Definition 11. The ordinal discernibility function based on the ordinal discernibility matrix is defined as

$$f(M^O) = \bigwedge \left\{ \bigvee (m_{ij}^O) | \forall x_i, x_j \in U, m_{ij}^O \neq \varnothing \right\} \tag{21}$$

Through Eq. (21), the set of all prime implicants of $f(M^O)$ determines the set of complete feature subsets. To illustrate the proposed idea clearly, a simple case is given as follows.

Example 1. Table 1 is an ordinal dataset including six samples from the dataset "Bankruptyrisk". We selected two samples in each class.

TABLE 1
Ordinal dataset from "Bankruptyrisk"

sample	a	b	c	d	e	f	g	h	i	j	k	l	class
x_1	2	2	2	1	1	4	4	4	4	4	2	4	3
x_2	2	1	3	1	1	3	5	2	4	2	1	3	3
x_3	2	1	1	1	1	3	2	2	4	4	2	3	2
x_4	2	1	2	1	1	2	4	3	3	2	1	2	2
x_5	2	2	1	1	1	1	3	3	3	4	3	4	1
x_6	2	1	1	1	1	1	2	2	3	4	3	4	1

$$M^O = \{m_{ij}^O\} \tag{14}$$

$$m_{ij}^O = \begin{cases} \{c \in C | (x_i, x_j) \notin R_{\{c\}}^{\leq}\}, & if\ (x_i, x_j) \notin R_{\{d\}}^{\leq}\ and\ x_i, x_j \in U_M^{\beta} \\ \{c \in C | (x_i, x_j) \notin R_{\{c\}}^{\leq}\}, & if\ (x_i, x_j') \notin R_{\{d\}}^{\leq}\ and\ x_i \in U_M, x_j \in U_{NM}^{\beta} \\ \{c \in C | (x_i, x_j) \notin R_{\{c\}}^{\leq}\}, & if\ \mu_{ik} > \mu_{jk}\ for\ \forall d_k^{\geq}\ and\ x_i, x_j \in U_{NM}^{\beta} \\ \varnothing, & otherwise \end{cases} \tag{15}$$

$$M^O = \begin{Bmatrix} \varnothing & \varnothing & \{b,c,f,g,h,l\} & \{b,f,h,i,j,k,l\} & \{c,f,g,h,i\} & \{b,c,f,g,h,i\} \\ \varnothing & \varnothing & \{c,g\} & \{c,f,g,i,l\} & \{c,f,g,i\} & \{c,f,g,i\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \{f,i\} & \{f,i\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \{c,f,g\} & \{c,f,g,h\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \end{Bmatrix} \tag{22}$$

According to Eq. (15), the discernibility matrix is Eq. (22). So the discernibility function is:

$$\begin{aligned} f(M_O) = &\{b \vee c \vee f \vee g \vee h \vee l\} \wedge \{b \vee f \vee h \vee i \vee j \vee k \vee l\} \\ &\wedge \{c \vee f \vee g \vee h \vee i\} \wedge \{b \vee c \vee f \vee g \vee h \vee i\} \wedge \{c \vee g\} \\ &\wedge \{c \vee f \vee g \vee i \vee l\} \wedge \{c \vee f \vee g \vee i\} \\ &\wedge \{f \vee i\} \wedge \{c \vee f \vee g\} \wedge \{c \vee f \vee g \vee h\} \\ &= \{c \vee g\} \wedge \{f \vee i\} \\ &= \{c \wedge f\} \vee \{g \wedge f\} \vee \{c \wedge i\} \vee \{g \wedge i\} \end{aligned} \tag{23}$$

In this way, all feature subsets, $\{c, f\}, \{g, f\}, \{c, i\}$ and $\{g, i\}$, could be obtained.

3.2 Fusing complete monotonic decision trees

It is well known that ensemble leaning, which applies multiple learners to solve one problem, can improve generalization performance of a learning system. Decision forest is one kind of ensemble leaning manner, in which decision trees are considered as base learners. Here monotonic classification problem is solved through constructing decision forest. Two aspects should be considered: training multiple decision trees and fusing these classification results of all trees.

For the first problem, we need to generate some different decision trees, which can be achieved by applying different data sets. There are often three kinds of ways: using some different samples of training dataset, choosing some different features from all the features and combining the first and second methods. Because the completed feature subsets are obtained we adopt the second one here. Another difficulty is to determine the number of decision trees. To solve the difficulty, we set the number of decision trees as the number of obtained feature subsets. A decision tree can be generated on a dataset, whose features are all the features of a feature subset obtained by last subsection, so each feature subset can construct a decision trees correspondingly. Then a decision forest can be obtained. Here we construct base decision trees through employing REMT method [4], which can be used to get a monotonic decision trees.

For the second problem, each decision tree gives a class value by its own classification rules for a new sample x, so the final result will be integrated using a weighted voting method. Each tree has a variable weight that is computed by one of its leaf nodes, which gives the classification result of x in this tree by its rules. For the x, the weight of class d_k in ith decision tree $\omega_{d_k}^i$ is computed as follows:

$$\omega_{d_k}^i = \frac{|Leaf_{d_k}^i|}{|Leaf^i|} \tag{24}$$

where $|Leaf_{d_k}^i|$ is the number of samples whose class is d_k on the leaf node of the ith decision tree, and $|Leaf^i|$ is the number of all samples on the leaf node of the ith decision tree.

Based on the above-mentioned constructing method of decision forest, the proposed FCMT is summarized as Algorithm 2.

Algorithm 2 FCMT

Require: a ordinal dataset $OD = (U, C \cup d)$; variable precision: β; stoping criterion of REMT: ε; an sample depicted by A: x.

Ensure: the class of sample x.

1: divide the samples in ordinal dataset into two sets, monotonically consistent set and non-monotonic set, by Eq. (12).
2: produce ordinal discernibility matrix by Eq. (15).
3: compute ordinal discernibility function by Eq. (21) and get the feature subsets $FS = \{fs_1, ..., fs_n\}$.
4: **for** fs_1 to fs_n **do**
5: learn a tree T_i with REMT.
6: for sample x, compute the weight ω_k^i of decision tree T_i voting d_k $(k = 1, 2, ...)$ by Eq. (24).
7: **end for**
8: return the final class: $d_k = arg\max_k(\sum_{i=1}^{n} \omega_k^i)$.

Now, we explain the working process FCMT algorithm. It can be understood through an illustrative example.

Example 2. We generate a dataset containing 12 samples by selecting randomly 6 samples in each class from the dataset "Adult", in which there are 14 features, as shown in Table 2. In this dataset, samples x_1 to x_{10} are treated as the training set of constructing monotonic decision trees, and samples x_{11} to x_{12} are looked forward as the test set of evaluating the performance of this model.

TABLE 2
Ordinal dataset from "Adult"

sample	a	b	c	d	e	f	g	h	i	j	k	l	m	n	class
x_1	2	1	1	1	2	2	5	2	1	2	1	1	2	1	2
x_2	2	4	1	2	1	1	8	2	1	2	1	1	2	1	2
x_3	1	6	1	2	1	1	3	3	5	2	1	1	1	1	2
x_4	2	5	1	4	1	1	11	3	5	2	1	1	2	1	2
x_5	2	2	1	11	2	1	10	3	1	2	1	1	2	1	2
x_6	2	2	1	4	1	1	5	3	1	2	1	1	2	1	1
x_7	2	1	1	11	2	3	6	4	1	1	2	1	2	1	1
x_8	2	1	1	1	2	1	5	3	1	2	2	1	2	1	1
x_9	2	1	1	2	1	1	5	3	5	2	1	1	2	1	1
x_{10}	2	6	1	1	2	1	6	3	2	2	1	1	2	8	1
x_{11}	2	5	1	10	1	1	1	3	1	2	1	1	2	1	2
x_{12}	2	1	1	5	2	1	6	3	1	2	2	1	2	1	1

We set β to 0 and n_0 to 1. According to Eq. (12) all the ten samples are in the monotonic consistent set. Then, the ordinal discernibility matrix is produced as Eq. (25) (see the equation on next page).

And the ordinal discernibility function is:

$f(M_O) = \{f \wedge j \wedge b \wedge d\} \vee \{f \wedge j \wedge b \wedge g \wedge i\}$.

So the feature subsets are $\{f, j, b, d\}$ and $\{f, j, b, g, i\}$.

Next, two training subsets is generated by these two feature subsets, which are shown as Table 3 and Table 4.

A monotonic decision tree can be learned with REMT algorithm on each training subset. The trees T_1 and T_2 learned from these two training subsets and their nodes weight are shown as Fig 1.

According to the trees T_1 and T_2, sample x_{11} in test set is both classified into Class 2, and has two weights in Class 1 and Class 2 respectively, as follows:

$w^1_{d_1} = 0.33$, $w^1_{d_2} = 0.67$;

$w^2_{d_1} = 0.5$, $w^2_{d_2} = 0.5$.

So, the weight sum of d_1 from two trees is 0.83, while the weight sum of d_2 is 1.17. Then sample x_{11} is classified into Class 2 at last.

Sample x_{12} is both classified into Class 1 by two trees and also has two weights in Class 1 and Class 2 respectively, as follows:

$w^1_{d_1} = 0.75$, $w^1_{d_2} = 0.25$;

$w^2_{d_1} = 1$, $w^2_{d_2} = 0$.

So, the weight sum of d_1 and d_2 from two trees are 1.75 and 0.25 respectively. Then sample x_{12} is classified into Class 1 at last.

3.3 Time complexity

The running time of FCMT method is mainly composed of two parts: the time of constructing discernibility matrix and feature subsets; the time of generating decision trees.

Before constructing discernibility matrix, the samples in dataset should be divided into two sets: the monotonically consistent set U_M and the non-monotonic set U_{NM}. To judge the consistency of each sample, we need to traverse the m features of each sample. Then its time complexity is $O(mn^2)$. Next, discernibility matrix will be constructed by pairwise comparison of all the samples: (1) For two samples both in set U_M, we need to compare their m features in turn, and the time complexity of computing discernibility features of these samples is $O(m|U_M|^2)$. (2) For two samples, one of which in set U_M and the other in set U_{NM}, the class label of sample in set U_{NM} should be modified. Supposing that dataset was be divided into t classes, the time complexity of modifying these class labels is $O(tn)$. Then the time complexity of computing discernibility features of these samples is $O(tn|U_{NM}|) + O(m|U_M||U_{NM}|)$. (3) For two samples both in set U_{NM}, μ of sample to each class need to be calculated and its time complexity is $O(tn)$. Then the time complexity of computing discernibility features of these samples is $O(tn|U_{NM}|) + O(tm|U_{NM}|^2)$. Finally, the feature subsets are obtained based on discernibility matrix, and its time complexity is $O(mn^2)$. The sum of the time complexities of above steps is as follows:

$$\begin{aligned} &O(mn^2) + O(m|U_M|^2) + O(tn|U_{NM}|) + O(m|U_M||U_{NM}|) \\ &+ O(tn|U_{NM}|) + O(tm|U_{NM}|^2) + O(mn^2) \\ &\leq O(mn^2) + O(mn^2) + O(tn^2) + O(mn^2) + O(tn^2) \\ &+ O(tmn^2) + O(mn^2) \\ &= O(4mn^2) + O(2tn^2) + O(tmn^2) \end{aligned} \tag{26}$$

So the time complexity of first part is $O(tmn^2)$.

When the decision tree is generated, the time complexities of non-leaf nodes and leaf nodes should be considered separately. In non-leaf nodes, the features (less than m) in feature subsets need to be considered in turn. Taking the v values in feature range as the split points, and rank mutual information will be computed under each split points. We need to traverse all the samples for the rank mutual information of each sample. Then the time complexity of this process is at most $O(mvn^2)$. In leaf nodes, we need to traverse all the samples in this nodes to compute support degree of each class of this node. Then the time complexity in leaf nodes is at most $O(n)$. So supposing that the numbers of non-leaf nodes and leaf nodes in the decision forest are k_1 and k_2 respectively and the number of feature subsets is h, the time complexity of second part is $O(hn+k_1mvn^2+k_2n)$.

Therefore, the time complexity of FCMT method is $O(tmn^2) + O(hn + k_1mvn^2 + k_2n)$.

4 Experimental Results and Analysis

4.1 Experimental data and evaluation

In order to test the performance of our approach, we employed ten datasets, which are same as Ref [26] and are shown in Table 5. In this table, Students Score is a real-world dataset.

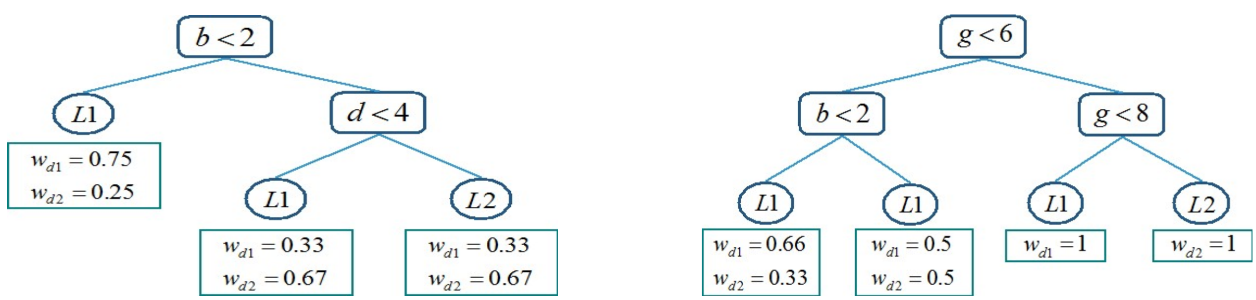
$$M^O = \left\{ \begin{array}{cccccccccc} \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \{e,f\} & \{j\} & \{f\} & \{e,f\} & \{f\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \{b,g\} & \{b,g,j\} & \{b,d,g\} & \{b,g\} & \{d,g\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \{b,i\} & \{b,i,j\} & \{b,d,i\} & \{b\} & \{d,i\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \{b,g,i\} & \{b,g,i,j\} & \{b,d,g,i\} & \{b,d,g\} & \{d,g,i\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \{d,e,g\} & \{b,g,j\} & \{b,d,g\} & \{b,d,e,g\} & \{d,g\} \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \end{array} \right\} \quad (25)$$

TABLE 3
Training subset with features $\{f, j, b, d\}$

sample	b	d	f	j	class
x_1	1	1	2	2	2
x_2	4	2	1	2	2
x_3	6	2	1	2	2
x_4	5	4	1	2	2
x_5	2	11	1	2	2
x_6	2	4	1	2	1
x_7	1	11	3	1	1
x_8	1	1	1	2	1
x_9	1	2	1	2	1
x_{10}	6	1	1	2	1

TABLE 4
Training subset with features $\{f, j, b, g, i\}$

sample	b	f	g	i	j	class
x_1	1	2	5	1	2	2
x_2	4	1	8	1	2	2
x_3	6	1	3	5	2	2
x_4	5	1	11	5	2	2
x_5	2	1	10	1	2	2
x_6	2	1	5	1	2	1
x_7	1	3	6	1	1	1
x_8	1	1	5	1	2	1
x_9	1	1	5	5	2	1
x_{10}	6	1	6	2	2	1



(a) Monotonic decision tree T_1 trained with features $\{f, j, b, d\}$ (b) Monotonic decision tree T_2 trained with features $\{f, j, b, g, i\}$

Fig. 1. Monotonic decision trees trained with two data subsets

TABLE 5
Datasets used in the experiments

Type	Data set	Num. of samples	Num. of features (numeric | nominal)	Num. of classes
UCI or Weka datasets	Adult	500	14 (0 | 14)	2
	Bankruptyrisk	39	12 (0 | 12)	3
	Wine	1599	11 (0 | 11)	2
	Squash	50	24 (22 | 2)	3
	Car	1728	6 (0 | 6)	4
	German	1000	20 (7 | 13)	2
	Australia	690	14 (6 | 8)	2
	Autompg	392	7 (0 | 7)	4
	Swd	3240	10 (0 | 10)	3
Real world dataset	Student Score	512	25 (25 | 0)	3

The classification accuracy (CA) and the mean absolute error (MAE) are used to verify the performance of the proposed approach and reference models.

$$CA = \frac{\sum\limits_{x_i \in U} I(\widehat{y}_i, y_i)}{|U|} \tag{27}$$

$$MAE = \frac{\sum\limits_{x_i \in U} |\widehat{y}_i - y_i|}{|U|} \tag{28}$$

where $I(\widehat{y}_i, y_i) = \begin{cases} 1, & \widehat{y}_i = y_i \\ 0, & \widehat{y}_i \neq y_i \end{cases}$, y_i is the real class of x_i and $\widehat{y}_i$ is the forecasting class of x_i by classifier.

4.2 Effectiveness of feature selection

In this section, we will verify the effectiveness of feature selection based on discernibility matrix by observing the gap between the mean dependency $\widetilde{\gamma}$ on feature subsets and the dependency γ on all features. The dependency of each feature subset is computed at first, and then we compare their mean value $\widetilde{\gamma}$ with the dependency γ on original feature set. If $\widetilde{\gamma}$ is always similar to the dependency of original feature set γ, the feature subsets can be regarded as effective. We get ten pairs of dependencies γ and $\widetilde{\gamma}$ by executing ten experiments which use ten different training datasets from 10-fold cross validation. The experiment results are shown in Fig. 2.

From Fig.2, it can be observed that the difference between the dependency $\widetilde{\gamma}$ and γ is small on all datasets. For Adult, Bankruptyrisk, Car and Australia, the dependencies γ and $\widetilde{\gamma}$ almost have no difference on ten experiments. For Wine, there is only one slight deviation on 3rd experiment. The number of dependency γ with slight deviation is five at most on one dataset, but all the differences are very small (no more than 0.05). Therefore, we conclude that these feature subsets could keep the same approximation ability for classification results as original feature set, which means that the feature subsets are effective.

4.3 Tuning variable precision parameter β

The experiments in this subsection will testify the classification performance of the proposed FCMT with different variable precision parameter β. For each dataset, 10-fold cross validation technique is used, in which 90% data are served as the training data and the remained samples are used as the test data.

For fair comparison, we set $\varepsilon = 0.01$ in experiments like Ref [4], and then observe the influence of β on classification performance. Let β vary from 0 to 0.3 with a step length 0.02. We compare classification accuracy and mean absolute error of FCMT, REMT and FREMT under each β value. The experiment results are shown in Fig.3 and Fig.4 respectively.

In this experiment, REMT has a constant CA and MAE in each different β value. From Fig.3 and Fig.4, we can see that in most β values our method FCMT have higher CA and lower MAE than REMT and FREMT in most datasets. Especially in five datasets (Wine, German, Australia, Autompg and Swd), FCMT have better indicator values under all kinds of β value. In addition, in all datasets FCMT is superior to REMT for two indicators, while FREMT sometimes is inferior to REMT in some β values. This is because FREMT method adopts a heuristic search strategy, which may not obtain all feature subset in one β value. Particularly, When $\beta > 0.16$, FREMT can not find feature subsets, while FCMT can get all the feature subsets under each β value.

TABLE 6
W-T-L summarization table

Method	FCMT-REMT	FCMT-FREMT
CA	144-0-0	81-0-5
MAE	144-0-0	81-0-4

Win-tie-loss (W-T-L) summarization table is shown in Table 6. A win means that the former method is better than the latter method on a criterion, while a loss means that the former method is worse than the latter method. A tie means that both methods have the same performance. From Table 6, It can be seen clearly that FCMT is superior to both the reference methods.

4.4 Performance of the proposed FCMT

In this subsection, we compare the number of trees, classify accuracy and mean absolute error of these three methods under their best parameter values respectively. For FREMT, we integrated all β values (β varied from 0 to 0.16 with a step length 0.02 and $\varepsilon = 0.01$) like Ref [26], which means the feature subsets need to be computed under 9 different parameter values. And 10-fold cross validation technique is also used in each method. The comparisons are listed in Table 7.

From Table 7, it is obvious that FCMT has much fewer trees than FREMT. Especially on the data sets with more features, such as Adult, Squash and German datasets, the dominance is more evident (They have 14, 24 and 20 features, and their numbers of trees have reduced by 88.9%, 91.8% and 83.3%, respectively). On these datasets, the lowest reduced percentage is 50%, while the highest reduced percentage reaches up to 91.8%. Also, FCMT have higher CA and lower MAE than the REMT on all the datasets. The FCMT has better indicator values than the FREMT on all the dataset except German and Australia, in which the difference between the indicators of two methods is also very small (not more than 0.017). The above experiments support that the proposed FCMT is effective for simplifying the model and improving classification performance.

4.5 Verifying on real world dataset

To verify the effectiveness of FCMT in real world, we carry out the experiments described in Sections 4.2 to 4.4 on a real world dataset Student Score, which includes 512 students coming from Software Engineering of grade 2010 in Shanxi University and their scores of 25 courses (features). These students are decided into three groups according to their scores: 122 students with excellent academic achievement, 269 students with ordinary academic achievement and 121 underachievers. The Student Score dataset is a natural monotonic classification problem and its label distribution is shown as Fig.5a.

The three groups of experiments on Student Score are as follows: (1) compare the dependencies γ and $\widetilde{\gamma}$ in ten

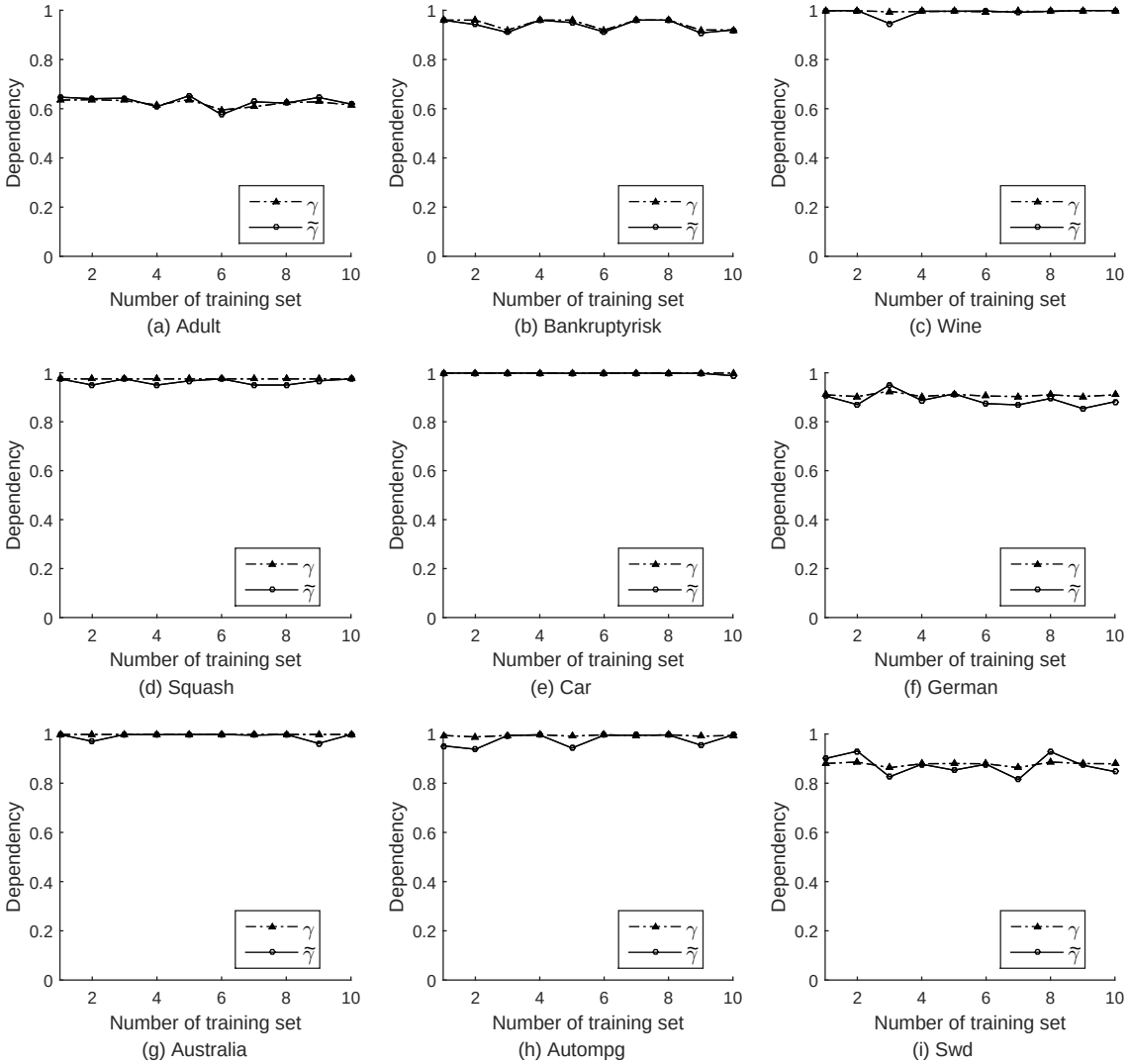


Fig. 2. Dependencies on ten different training sets

TABLE 7
Comparison on number of trees, CA and MAE

Dataset	Num. of trees		CA			MAE		
	FREMT	FCMT	REMT	FREMT	FCMT	REMT	FREMT	FCMT
Adult	36	**4**	0.604 ± 0.126	0.774 ± 0.034	$\mathbf{0.790} \pm 0.049$	0.396 ± 0.126	0.226 ± 0.034	$\mathbf{0.210} \pm 0.049$
Bankruptyrisk	27	**6.4**	0.650 ± 0.139	0.858 ± 0.108	$\mathbf{0.858} \pm 0.072$	0.350 ± 0.139	0.142 ± 0.108	$\mathbf{0.142} \pm 0.072$
Wine	14	**4.3**	0.465 ± 0.063	0.626 ± 0.032	$\mathbf{0.683} \pm 0.053$	0.535 ± 0.063	0.374 ± 0.032	$\mathbf{0.317} \pm 0.053$
Squash	68	**5.6**	0.580 ± 0.145	0.740 ± 0.089	$\mathbf{0.800} \pm 0.092$	0.480 ± 0.147	0.260 ± 0.089	$\mathbf{0.250} \pm 0.092$
Car	12	**6**	0.817 ± 0.031	0.871 ± 0.011	$\mathbf{0.907} \pm 0.025$	0.203 ± 0.031	0.148 ± 0.011	$\mathbf{0.111} \pm 0.032$
German	45	**7.5**	0.529 ± 0.032	$\mathbf{0.711} \pm 0.037$	0.695 ± 0.053	0.471 ± 0.032	$\mathbf{0.289} \pm 0.037$	0.305 ± 0.053
Australia	47	**8.2**	0.586 ± 0.055	$\mathbf{0.735} \pm 0.045$	0.718 ± 0.037	0.414 ± 0.055	$\mathbf{0.265} \pm 0.045$	0.282 ± 0.037
Autompg	27	**6**	0.528 ± 0.054	0.594 ± 0.070	$\mathbf{0.696} \pm 0.066$	0.513 ± 0.054	0.431 ± 0.070	$\mathbf{0.315} \pm 0.068$
Swd	14	**6.3**	0.581 ± 0.031	0.683 ± 0.032	$\mathbf{0.725} \pm 0.041$	0.451 ± 0.031	0.341 ± 0.032	$\mathbf{0.309} \pm 0.042$
Average	32.22	6.03	0.593	0.732	0.764	0.424	0.275	0.249

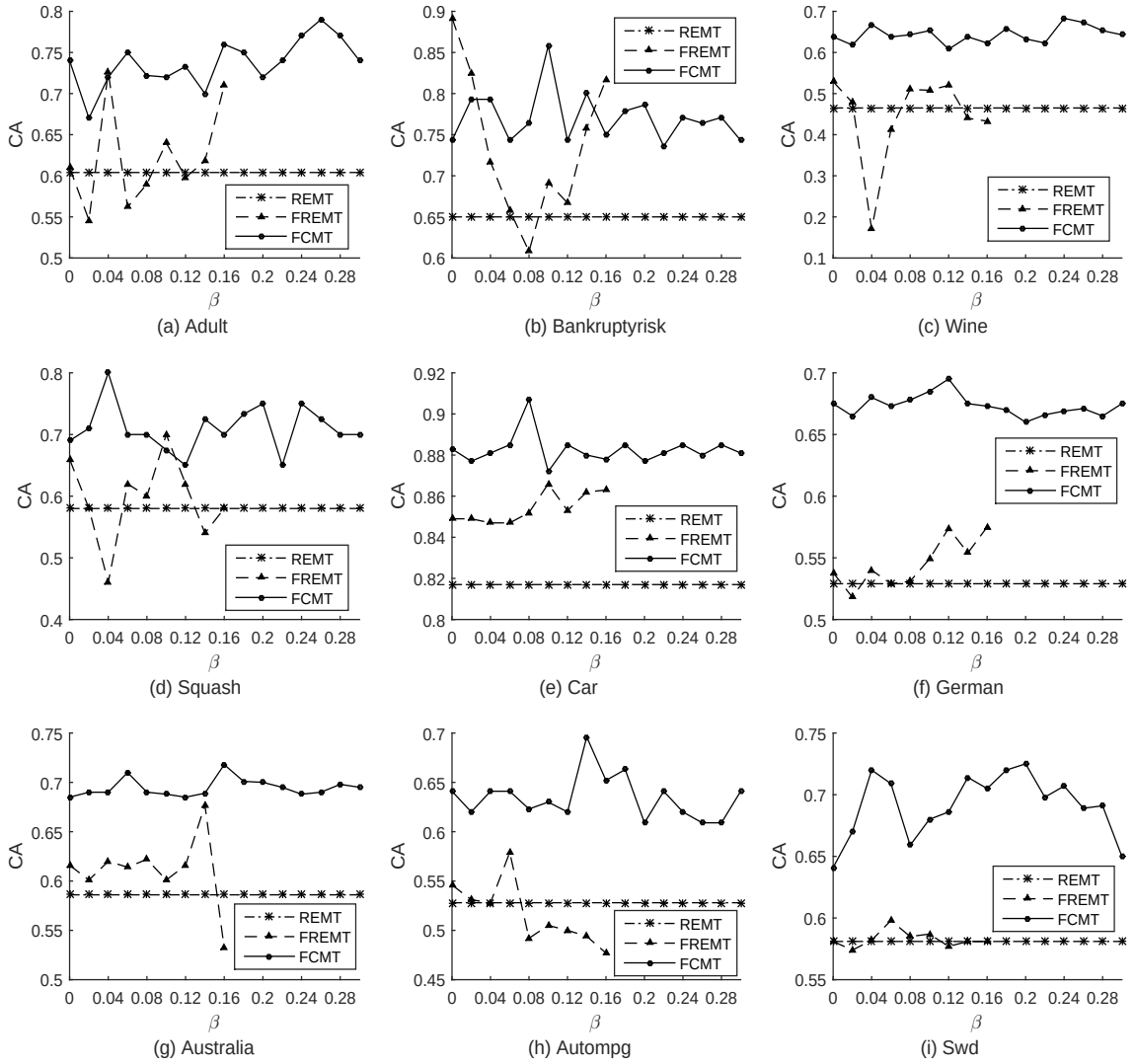


Fig. 3. The average value of classification accuracies of each β value

different training datasets from the 10-fold cross validation; (2) compare classification accuracy and mean absolute error of FCMT, REMT and FREMT under different β values which varies from 0 to 0.3 with a step length 0.02; (3) compare the number of trees, classification accuracy and mean absolute error of FCMT, REMT and FREMT under their best parameters respectively. The experiment results are shown in Fig.5b-d and Table 8.

From Fig.5 and Table 8, we can see that, for Student Score dataset, the dependencies γ and $\widetilde{\gamma}$ almost have no difference on ten experiments. So the feature selection is effective for real world dataset. For all β values except 0.02, FCMT have higher CA and lower MAE than REMT and FREMT. The number of decision tree is determined by the number of feature subsets obtained from dataset. In the existing method FREMT, the feature subsets need to be computed under 9 different variable precision parameter values to construct a good classification model. It results in a large number of feature subsets. But in the proposed FCMT method, the feature subsets are computed only under one parameter value since FCMT can obtain a group of complete feature subsets, whose completeness makes them have a small number and can be used to construct a better classification model. So in the best parameter values, FCMT has much fewer trees than FREMT (The number of trees reduces by 88.1%). In addition, the completeness of feature subsets ensures that decision trees learned from these subsets have good diversity and coverage. Although the small number of parameters make the number of decision trees reduced, the good diversity and coverage make the classification performance of the proposed FCMT method improved compared with earlier work. For FCMT, the percentages of improved CA and MAE are 13.96% and 42.34% respectively compared with REMT, and are 0.71% and 4.03% respectively compared

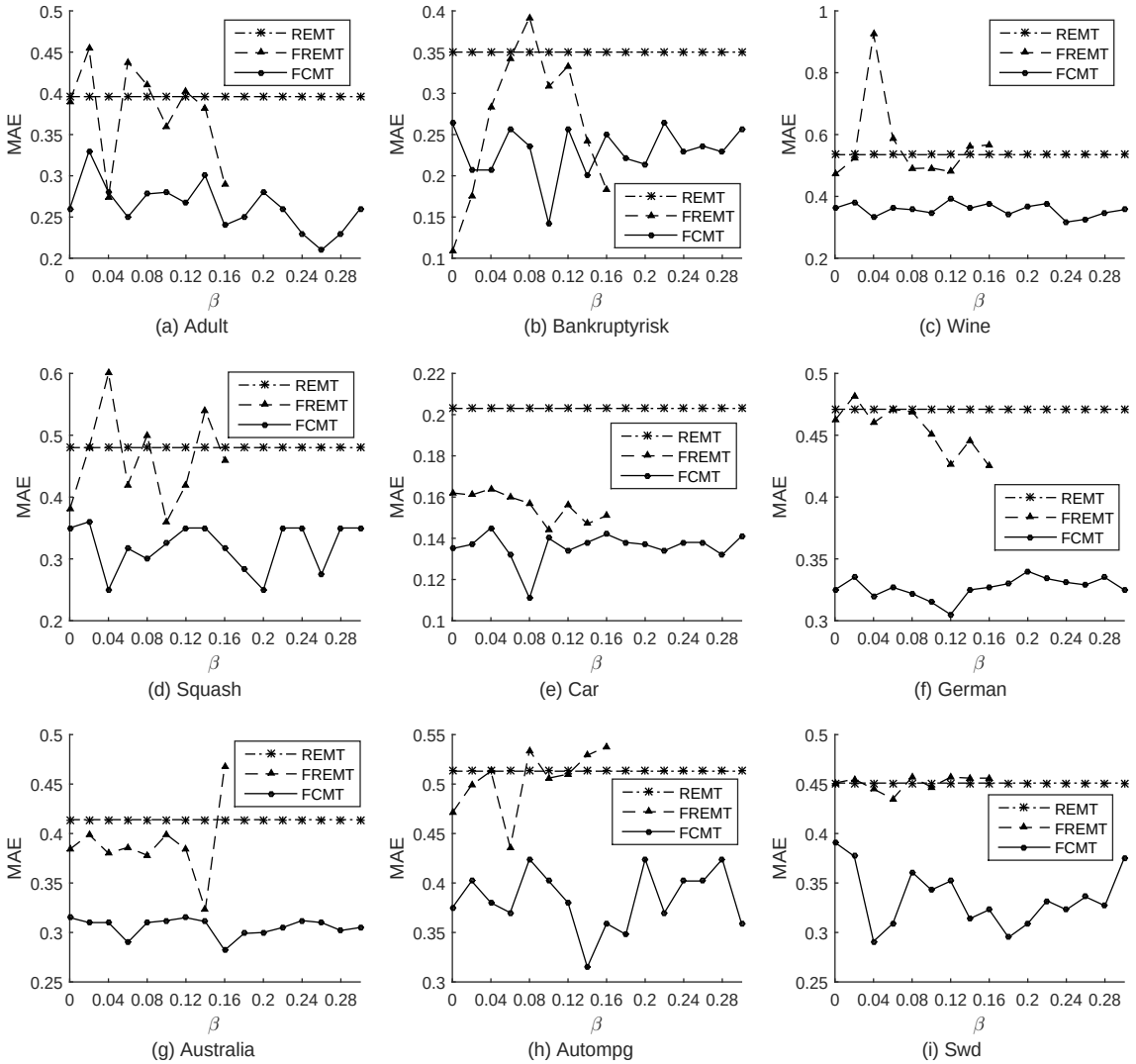
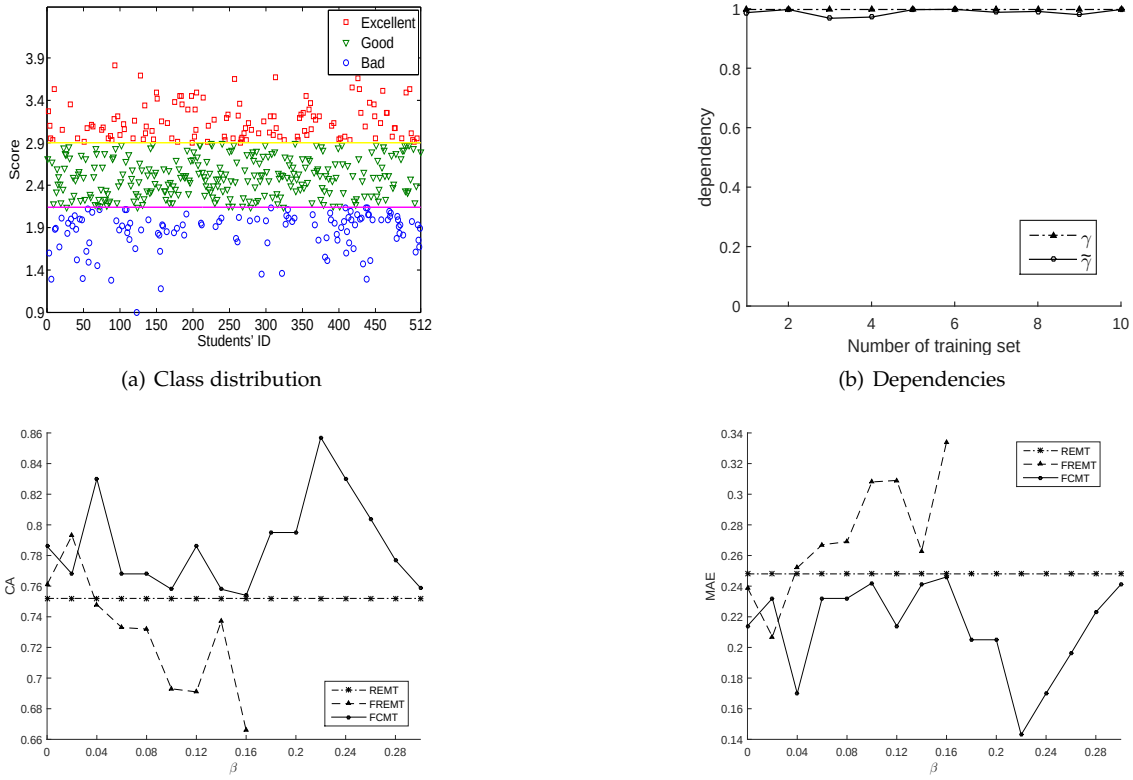


Fig. 4. The average value of mean absolute errors of each β value

TABLE 8
Comparison of number of trees, CA and MAE on Student Score

	Num. of trees	CA	MAE
REMT	–	0.752±0.044	0.248±0.044
FREMT	77	0.851±0.054	0.149± 0.054
FCMT	**9.2**	**0.857**±0.046	**0.143**± 0.046



(a) Class distribution

(b) Dependencies

(c) Classification accuracies

(d) Mean absolute errors

Fig. 5. The class distribution and experiment results on Student Score dataset

with FREMT. Therefore, on real world dataset, FCMT is also able to simplify the classification model and improve classification performance.

5 Conclusions

In this paper, we proposed an approach FCMT to solve monotonic classification problem. We obtained complete feature subsets on an ordinal dataset. Then fusing complete monotonic decision trees method is proposed. Compared with the popular "random forest" and other ensemble methods, the advantages of the proposed FCMT are: (1) The feature selection approach preserves the rank on ordinal dataset. (2) It automatically selects decision trees and automatically determines the number of decision tree. (3) It executes only under a kind of variable precision parameter value, so FCMT can reduce the number of decision trees greatly and obtain good classification performance simultaneously. (4) FCMT method is aimed at solving monotonic classification problem, and it considers ordinal relation and monotonicity constraint in dataset. Although "random forest" and other ensemble methods can be applied to solve this problem, they cannot obtain the classification rules satisfying the monotonicity constraint.

However, there has no guiding method to selected parameter β for FCMT. Besides, the computing cost of ordinal discernibility matrix and discernibility function might be expensive. In the future, we will do some research to solve these two problems. Moreover, our method could be applied to the service selection problem according to the quality of service in future, which is one of problems in service computing and a monotonic classification problem essentially. We plan to design the special algorithm for service selection.

In addition, there are some resemblances between the regression problem and the monotonic classification problem indeed. Both regression problem and monotonic classification problem can deal with the ordinal data. However, the outputs of the training samples in regression problem are quantitative real values. A real-valued function is obtained by learning the training samples, and finally outputs real values, while the monotonic classification in this manuscript outputs class labels as final outcome. For monotonic classification, although there are the ordinal relation among these class labels, the class labels only show the dominance relations among them essentially and do not give specific quantitative values on their dominance quantity. Then, the training samples in monotonic classification problem only have qualitative class labels, and the outputs of the classification model learning from this samples are also class labels. In this work, we focus on whether the estimated classes match the practical classes or not, and do not need to test the exact values of different samples. But it is a very interesting project to solve monotonic classification problem through applying regression technique. First, this problem needs to convert the class labels of training samples into numerical values; Second, for the real-valued outputs of regression technique, a map function between these outputs and each class should be given. Then, two corresponding need to be designed. One is designing the rules by which the class labels of training samples with different feature values

can be converted into numerical values accurately; the other is finding the reasonable map function. So we will consider solving above two questions in the future work, which may provide a new train of thought for monotonic classification problem. Furthermore, in real life there are indeed some monotonic problems which need real-value functions to estimate their specific real values, and it not only focuses on whether the estimated classes match the practical classes or not, but also concerns whether the estimated outputs are accurate or not. This kind of problem needs regression methods to solve, and it also deals with ordinal data and have a monotonicity constraint in their features and outputs. But this paper have not covered this kind of problem, which will be dealt with in our next research.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Nos. 61673249, 61273291), the Research Project Supported by Shanxi Scholarship Council of China (No. 2016-004) and the Innovation Project of Shanxi Graduate Education (No. 2016BY003).

REFERENCES

[1] R. Senge and E. Hullermeier, "Top-down induction of fuzzy pattern trees," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 241-252, 2011.
[2] G. D. Wu, Z. W. Zhu, and P. H. Huang, "A TS-type maximizing discriminability-based recurrent fuzzy network for classification problems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 339-352, 2011.
[3] R. Potharst and A. J. Feelders, "Classification trees for problems with monotonicity constraints," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 1-10, 2002.
[4] Q. H. Hu, X. J. Che, L. Zhang, D. Zhang, M. Z. Guo, and D. R. Yu, "Rank entropy based decision trees for monotonic classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2052-2064, 2012.
[5] D. Chen, T. Li, D. Ruan, J. Lin, and C. Hu, "A rough-set-based incremental approach for updating approximations under dynamic maintenance environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 2, pp. 274-284, 2013.
[6] Q. H. Hu, D. R. Yu, W. Pedrycz, and D. G. Chen, "Kernelized fuzzy rough sets and their applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1649-1667, 2011
[7] Y. H. Qian, J. Y. Liang, W. Z. Wu, and C. Y. Dang, "Information granularity in fuzzy binary GrC model," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 253-264, 2011.
[8] Y. H. Qian, J. Y. Liang, and C. Y. Dang, "Incomplete multigranulation rough set," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 2, pp. 420-431, 2010.
[9] Y. Y. Yao, "Probabilistic rough set approximations," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 255-271, 2008.
[10] S. Y. Zhao, E. C. C. Tsang, and X. Z. Wang, "Building a rule-based classifier–a fuzzy rough set approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 624-638, 2010.
[11] Y. H. Qian, C. Y. Dang, J. Y. Liang, and D. W. Tang, "Set-valued ordered information systems," *Information Sciences*, vol. 179, no. 16, pp. 2809-2832, 2009.
[12] Y. H. Qian, J. Y. Liang, and C. Y. Dang, "Interval ordered information systems," *Computers and Mathematics with Applications*, vol. 56, no. 8, pp. 1994-2009, 2008.
[13] Q. H. Hu, W. W. Pan, L. Zhang, D. Zhang, Y. P. Song, M. Z. Guo, and D. R. Yu, "Feature selection for monotonic classification," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 69-81, 2012.
[14] S. Greco, B. Matarazzo, R. Slowinski, and J. Stefanowski, "Variable consistency model of dominance-based rough sets approach," in *International Conference on Rough Sets and Current Trends in Computing*. Berlin Heidelberg: Springer, 2000, pp. 170-181.
[15] N. Barile and A. Feelders, "Nonparametric monotone classification with MOCA," in *Proceedings IEEE 8th International Conference on Data Mining*, 2008, pp. 731-736.
[16] A. Ben-David, L. Sterling, and Y. H. Pao, "Learning and classification of monotonic ordinal concepts," *Computational Intelligence*, vol. 5, no. 1, pp. 45-49, 1989.
[17] A. Ben-David, "Monotonicity maintenance in information-theoretic machine learning algorithms," *Machine Learning*, vol. 19, no. 1, pp. 29-43, 1995.
[18] W. Duivesteijn and A. Feelders, "Nearest neighbour classification with monotonicity constraints," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Berlin Heidelberg: Springer, 2008, pp. 301-316.
[19] A. Ben-David, "Automatic generation of symbolic multiattribute ordinal knowledge-based DSSs: Methodology and applications," *Decision Sciences*, vol. 23, pp. 1357-1357, 1992.
[20] F. Xia, W. S. Zhang, F. X. Li, and Y. W. Wang, "Ranking with decision tree," *Knowledge and Information Systems*, vol. 17, no. 3, pp. 381-395, 2008.
[21] K. Cao-Van and B. D. Baets, "Growing decision trees in an ordinal setting," *International Journal of Intelligent Systems*, vol. 18, no. 7, pp. 733-750, 2003.
[22] K. Cao-Van, "Supervised ranking from semantics to algorithms," Ph.D. thesis. Ghent University, Belgium, 2003.
[23] R. V. Kamp, "A.J. Feelders, N. Barile, Isotonic Classification Trees," in *International Symposium on Intelligent Data Analysis*. Berlin Heidelberg: Springer, 2009, pp. 405-416.
[24] W. Kotlowski and R. Slowinski, "Rule learning with monotonicity constraints," in *Proceedings of the 26th International Conference on Machine Learning*. Montreal, Quebec, Canada, 2009, pp. 537-544.
[25] R. Potharst and J. C. Bioch, "Decision trees for ordinal classification," in *Intelligent Data Analysis*, vol. 4, no. 2, pp. 97-111, 2000.
[26] Y. H. Qian, H. Xu, J. Y. Liang, B. Liu, and J. T. Wang. "Fusing monotonic decision trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. no. 10, pp. 2717-2728, 2015.
[27] X. H. Hu and N. Cercone, "Learning in relational databases: a rough set approach," *Computational Intelligence*, vol. 11, no. 2, pp. 323-338, 1995.
[28] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," *Intelligent Decision Support*. Netherlands: Springer, pp. 331-362, 1992.
[29] D. Y. Ye and Z. J. Chen, "A new discernibility matrix and the computation of a core," *Acta Electronica Sinica*, vol. 30, no. 7, pp. 1086-1088, 2002.
[30] M. Yang, "An incremental updating algorithm of the computation of a core based on the improved discernibility matrix," *Chinese Journal of Computers*, vol. 29, no. 3, pp. 407-413, 2006.
[31] W. X. Zhang, Y. Liang, and W. Z. Wu, *Information System and Knowledge Discovery*. Beijing, China: Science Press, 2003.
[32] S. Greco, B. Matarazzo, and R. Slowinski, "Rough approximation of a preference relation by dominance relations," *European Journal of Operational Research*, vol.117, no.1, pp. 63-83, 1999.
[33] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Boston: Kluwer Academic Publishers, 1991.
[34] D. Q. Miao and G. D. Li, *Theory, Algorithms and Applications of Rough Sets*. Beijing, China: Tsinghua University press, 2008.
[35] Q. H. Hu and D. R. Yu. *Applied Rough Computing*. Beijing, China: Science Press, 2012.

Hang Xu is a Ph.D candidate at school of Computer and Information Technology, Shanxi University. Before this, she got her M.S. degree from the school of Computer and Information Technology, Shanxi University, in 2014. Her research interest includes data mining and knowledge discovery.

Wenjian Wang is a professor and Ph.D. supervisor of Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, China. She received her Ph.D degree in applied mathematics from Xian Jiaotong University, China, in 2004. Before this, She received the B.S. degree in computer science from Shanxi University, China, in 1990, the M.S. degree in computer science from Hebei Polytechnic University, China, in 1993, She worked as a research assistant at the Department of Building and Construction, The City University of Hong Kong from May 2001 to May 2002. She has been with the school of Computer and Information Technology at Shanxi University since 1993, where she was promoted as Associate Professor in 2000 and as Full Professor in 2004, and now serves as a Ph.D. supervisor in Computer Application Technology and System Engineering. She has published more than 70 academic papers on machine learning, computational intelligence, and data mining. Her current research interests include neural networks, support vector machines, machine learning theory and environmental computations, etc.

Yuhua Qian is a professor and Ph.D. supervisor of Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, China. He received the M.S. degree and the PhD degree in Computers with applications at Shanxi University in 2005 and 2011, respectively. He is best known for multi-granulation rough sets in learning from categorical data and granular computing. He is actively pursuing research in pattern recognition, feature selection, rough set theory, granular computing and artificial intelligence. He has published more than 70 articles on these topics in international journals. On professional services, Qian has served as program chairs or special issue chairs of RSKT, JRS, and ICIC, and PC members of many machine learning, data mining, and granular computing. He also served on the Editorial Board of International Journal of Knowledge-Based Organizations and the Editorial Board of Artificial Intelligence Research.