Rank Entropy-Based Decision Trees for Monotonic Classification

Qinghua Hu, *Member*, *IEEE*, Xunjian Che, Lei Zhang, *Member*, *IEEE*, David Zhang, *Fellow*, *IEEE*, Maozu Guo, and Daren Yu

Abstract—In many decision making tasks, values of features and decision are ordinal. Moreover, there is a monotonic constraint that the objects with better feature values should not be assigned to a worse decision class. Such problems are called ordinal classification with monotonicity constraint. Some learning algorithms have been developed to handle this kind of tasks in recent years. However, experiments show that these algorithms are sensitive to noisy samples and do not work well in real-world applications. In this work, we introduce a new measure of feature quality, called rank mutual information (RMI), which combines the advantage of robustness of Shannon's entropy with the ability of dominance rough sets in extracting ordinal structures from monotonic data sets. Then, we design a decision tree algorithm (REMT) based on rank mutual information. The theoretic and experimental analysis shows that the proposed algorithm can get monotonically consistent decision trees, if training samples are monotonically consistent. Its performance is still good when data are contaminated with noise.

Index Terms—Monotonic classification, rank entropy, rank mutual information, decision tree

1 Introduction

Ordinal classification is a kind of important tasks in management decision, evaluation, and assessment, where the classes of objects are discretely ordinal, instead of numerical or nominal. In these tasks, if there exists a constraint that the dependent variable should be a monotone function of the independent variables [5], [19], namely, given two objects x and x', if $x \leq x'$, then we have $f(x) \leq f(x')$, we call this kind of tasks monotonic classification, also call it ordinal classification with monotonicity constraints [2], [3], [5], multicriteria decision making [1], [4].

Monotonic classification tasks widely exist in real-world life and work. We select commodities in a market according to the 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 offers according to students' performances. Editors make a decision on a manuscript according to its quality. If we collect a set of samples of a monotonic classification task, we can extract decision rules from the data for understanding the decisions and building an automatic decision model.

Compared with general classification problems, much less attention has been paid to monotonic classification these years [6], [7]. In some literature, a monotonic classification task was transformed from a k-class ordinal problem to $k-1$ binary class problems [9]. Then a learning algorithm for general classification tasks was employed on the derived data. In fact, this technique did not consider the monotonicity constraints in modeling.

Rule extraction from monotonic data attracts some attention from the domains of machine learning and decision analysis. Decision tree induction is an efficient, effective, and understandable technique for rule learning and classification modeling [10], [11], [12], where a function is required for evaluating and selecting features to partition samples into finer subsets in each node. Several measures, such as Gini, chi-square, and Gain ratio, were introduced into decision tree construction algorithms [10], [12]. It was concluded that Shannon's entropy outperforms other evaluation functions in most tasks [13]. Information entropy-based decision tree algorithms have been widely discussed and used in machine learning and data mining. Unfortunately, Shannon's information entropy cannot reflect the ordinal structure in monotonic classification. Even given a monotone data set, the learned rules might not be monotonic. This does not agree with the underlying assumption of these tasks and limits the application of these algorithms to monotonic classification.

In order to deal with monotonic classification, entropy-based algorithms were extended in [14], where both the error and monotonicity were taken into account while building decision trees; a nonmonotonicity index was introduced and used in selecting attributes. In [15], an order-preserving tree-generation algorithm was developed and a technique for repairing nonmonotonic decision trees was provided for multiattribute classification problems with several linearly ordered classes. In 2003, a collection of pruning techniques were proposed to make a nonmonotone tree monotone by pruning [33]. Also in 2003, Cao-Van and Baets constructed a tree-based algorithm to avoid violating the monotonicity of data [16], [41]. In [32], Kamp

- *Q. Hu, X. Che, M. Guo, and D. Yu are with Harbin Institute of Technology, Harbin 150001, China. E-mail: {huqinghua, mzguo, yudaren}@hit.edu.cn, 67139712@qq.com.*
- *L. Zhang and D. Zhang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: {cslzhang, csdzhang}@comp.polyu.edu.hk.*

Manuscript received 9 Mar. 2011; revised 22 May 2011; accepted 31 May 2011; published online 23 June 2011.
Recommended for acceptance by G. Karypis.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-03-0118.
Digital Object Identifier no. 10.1109/TKDE.2011.149.

et al. proposed a new algorithm for learning isotonic classification trees (ICT). In 2008, Xia et al. extended the Gini impurity used in CART to ordinal classification, and called it ranking impurity [17]. Although the algorithms mentioned above enhance the capability of extracting ordinal information, they cannot ensure a monotone tree is built if the training data are monotone.

In 2009, Kotlowski and Slowinski proposed an algorithm for rule learning with monotonicity constraints [18]. They first monotonized data sets using a nonparametric classification procedure and then generated rule ensemble consistent with the monotonized data sets. In 2010, Feelders also discussed the effectiveness of relabeling in the monotonic classification [40]. If there is a small quantity of inconsistent samples, this technique may be effective. However, if a lot of inconsistent samples exist, these techniques have to revise the labels of the inconsistent samples, as shown in [18]. This maybe lead to information loss. In 2010, Jimnez et al. introduced a new algorithm, POTMiner, to identify frequent patterns in partially ordered trees [19]. This algorithm did not consider the monotonicity constraints.

In addition, by replacing equivalence relations with dominance relations, Greco et al. generalized the classical rough sets to dominance rough sets for analyzing multicriteria decision making problems in [20]. The model of dominance rough sets can be used to extract rules for monotonic classification [36], [39]. Since then a collection of research works have been reported to generalize or employ this model in monotonic classification [20], [21], [22], [23], [24], [35]. It was reported that dominance rough sets produced large classification boundary on some real-world tasks [28], which made the algorithm ineffective as no or few consistent rules could be extracted from data.

Noise has great influence on modeling monotonic classification tasks [25]. Intuitively, algorithms considering monotonicity constraints should offer significant benefit over nonmonotonic ones because the monotonic algorithms extract the natural properties of the classification tasks. Correspondingly, the learned models should be simpler and more effective. However, numerical experiments showed that the ordinal classifiers were statistically indistinguishable from nonordinal counterparts [25]. Sometimes ordinal classifiers performed even worse than nonordinal ones. What are the problems with these ordinal algorithms? The authors attributed this unexpected phenomenon to the high levels of nonmonotonic noise in data sets. In fact, both the experimental setup and noisy samples lead to the incapability of these ordinal algorithms. In order to determine whether applying monotonicity constraints improves performance of algorithms, one should make a monotone version of this algorithm that resembles the original algorithm as much as possible and then to see whether performance improves.

More than 15 years ago, robustness was considered as an important topic in decision analysis [26]. In monotonic classification, decisions are usually given by different decision makers. The attitudes of these decision makers may vary from time to time. Therefore, a lot of inconsistent samples exist in gathered data sets. If the measures used to evaluate quality of attributes in monotonic classification are sensitive to noisy samples, the performance of the trained models would degrade. An effective and robust measure of feature quality is required in this domain [38].

The objective of this work is to design a robust and understandable algorithm for monotonic classification tasks. There usually are a lot of inconsistent samples in the noisy context. We introduce a robust measure of feature quality, called rank entropy, to compute the uncertainty in monotonic classification. As we know, Shannon's information entropy is robust in evaluating features for decision tree induction [10], [13], [34]. Rank entropy inherits the advantage of robustness of Shannon's entropy [27]. Moreover, this measure is able to reflect the ordinal structures in monotonic classification. Then, we design a decision tree algorithm based on the measure. Some numerical tests are conducted on artificial and real-world data sets. The results show the effectiveness of the algorithm. The contributions of this work is threefolds. First, we discuss the properties of rank entropy and rank mutual information (RMI). Second, we propose a decision tree algorithm based on rank mutual information. Finally, systemical experimental analysis is presented to show the performance of the related algorithms.

The rest of the paper is organized as follows: Section 2 introduces the preliminaries on monotonic classification. Section 3 gives the measure of ordinal entropy and discusses its properties. The new decision tree algorithm is introduced in Section 4. Numerical experiments are shown in Section 5. Finally, conclusions and future work are given in Section 6.

2 Preliminaries on Monotonic Classification and Dominance Rough Sets

Let $U = \{x_1, \ldots, x_n\}$ be a set of objects and A be a set of attributes to describe the objects; D is a finite ordinal set of decisions. The value of x_i in attributes $a \in A$ or D is denoted by $v(x_i, a)$ or $v(x_i, D)$, respectively. The ordinal relations between objects in terms of attribute a or D is denoted by $\leq$. We say x_j is no worse than x_i in terms of a or D if $v(x_i, a) \leq v(x_j, a)$ or $v(x_i, D) \leq v(x_j, D)$, denoted by $x_i \leq_a x_j$ and $x_i \leq_D x_j$, respectively. Correspondingly, we can also define $x_i \geq_a x_j$ and $x_i \geq_D x_j$. Given $B \subseteq A$, we say $x_i \leq_B x_j$ if for $\forall a \in B$, we have $v(x_i, a) \leq v(x_j, a)$. Given $B \subseteq A$, we associate an ordinal relation on the universe defined as

$$R_B^{\leq} = \{(x_i, x_j) \in U \times U | x_i \leq_B x_j\}.$$

A predicting rule is a function

$$f : U \to D,$$

which assigns a decision in D to each object in U. A monotonically ordinal classification function should satisfy the following constraint:

$$x_i \leq x_j \Rightarrow f(x_i) \leq f(x_j), \quad \forall x_i,\ x_j \in U.$$

As to classical-classification tasks, we know that the constraint is

$$x_i = x_j \Rightarrow f(x_i) = f(x_j), \forall x_i, x_j \in U.$$

Definition 1. *Let $DT = <U, A, D>$ be a decision table, $B \subseteq A$. We say DT is B-consistent if $\forall a \in B$ and $x_i, x_j \in U$, $v(x_i, a) = v(x_j, a) \Rightarrow v(x_i, D) = v(x_j, D)$.*

Definition 2. *Let $DT = <U, A, D>$ be a decision table, $B \subseteq A$. We say DT is B-monotonically consistent if $\forall x_i, x_j \in U$, $x_i \preceq_B x_j$, we have $x_i \preceq_D x_j$.*

It is easy to show that a decision table is B-consistent if it is B-monotonically consistent [5].

In real-world applications, data sets are usually neither consistent nor monotonically consistent due to noise and uncertainty. We have to extract useful decision rules from these contaminated data sets. Dominance rough sets were introduced to extract decision rules from data sets. We present some notations to be used throughout this paper.

Definition 3. *Given $DT = <U, A, D>, B \subseteq A$, we define the following sets*

1. $[x_i]_B^{\preceq} = \{x_j \in U | x_i \preceq_B x_j\}$;
2. $[x_i]_D^{\preceq} = \{x_j \in U | x_i \preceq_D x_j\}$.

We can easily obtain the following properties:

1. *If $C \subseteq B \subseteq A$, we have $[x_i]_C^{\preceq} \supseteq [x_i]_B^{\preceq}$;*
2. *If $x_i \preceq_B x_j$, we have $x_j \in [x_i]_B^{\preceq}$ and $[x_j]_B^{\preceq} \subseteq [x_i]_B^{\preceq}$;*
3. $[x_i]_B^{\preceq} = \cup\{[x_j]_B^{\preceq} | x_j \in [x_i]_B^{\preceq}\}$;
4. $\cup\{[x_i]_B^{\preceq} | x_i \in U\} = U$.

The minimal and maximal elements of decision D are denoted by $d_{\min}$ and $d_{\max}$, respectively, and $d_i^{\preceq} = \{x_j \in U | d_i \leq v(x_j, D)\}$. Then $d_{\min}^{\preceq} = U$ and $d_{\max}^{\preceq} = d_{\max}$.

Definition 4. *Given $DT = <U, A, D>$, $B \subseteq A$, d_i is a decision value of D. As to monotonic classification, the upward lower and upper approximations of $d_i^{\preceq}$ are defined as*

$$\underline{R_B^{\preceq}} d_i^{\preceq} = \{x_j \in U | [x_j]_B^{\preceq} \subseteq d_i^{\preceq}\},$$

$$\overline{R_B^{\preceq}} d_i^{\preceq} = \{x_j \in U | [x_j]_B^{\succeq} \cap d_i^{\preceq} \neq \emptyset\},$$

The model defined above is called dominance rough sets, introduced by Greco et al. [20]. This model was widely discussed and applied in recent years [21], [23], [24].

We know that $d_i^{\preceq}$ is a subset of objects whose decisions are equal to or better than d_i, and $\underline{R_B^{\preceq}} d_i^{\preceq}$ is a subset of objects whose decisions are no worse than d_i if their attribute values are better than x_i, whereas $\overline{R_B^{\preceq}} d_i^{\preceq}$ is a subset of objects whose decisions might be better than d_i. Thus $\underline{R_B^{\preceq}} d_i^{\preceq}$ is the pattern consistently equal to or better than d_i.

It is easy to show that $\underline{R_B^{\preceq}} d_i^{\preceq} \subseteq d_i^{\preceq} \subseteq \overline{R_B^{\preceq}} d_i^{\preceq}$ and the subset of objects

$$BND_B(d_i^{\preceq}) = \overline{R_B^{\preceq}} d_i^{\preceq} - \underline{R_B^{\preceq}} d_i^{\preceq}$$

is called the upward boundary region of d_i in terms of attribute set B.

Analogically, we can also give the definitions of downward lower and upper approximations, and boundary region $\underline{R_B^{\succeq}} d_i^{\succeq}$, $\overline{R_B^{\succeq}} d_i^{\succeq}$, $BND_B(d_i^{\succeq})$. We can obtain that

$$BND_B(d_i^{\succeq}) = BND_B(d_{i-1}^{\preceq})$$

for $\forall i > 1$. Finally, the monotonic dependency of D on B is computed as

$$\gamma_B(D) = \frac{|U - \cup_{i=1}^{N} BND_B(d_i)|}{|U|}.$$

TABLE 1
A Toy Monotonic Classification Task

Sample	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
A	2.12	3.44	5.32	4.53	7.33	6.51	7.56	8.63	9.11	9.56
D	1	1	1	1	2	2	2	3	3	3

If decision table DT is monotonically consistent in terms of B, we have $BND_B(d_i) = \emptyset$. In addition, as $d_{\max}^{\preceq} = U$ and $d_{\min}^{\succeq} = U$ thus $\underline{R_B^{\preceq}} d_{\max}^{\preceq} = U$ and $\underline{R_B^{\succeq}} d_{\min}^{\succeq} = U$.

Dominance rough sets give us a formal framework for analyzing consistency in monotonic classification tasks. However, it was pointed out that decision boundary regions in real-world tasks are so large that we cannot build an effective decision model based on dominance rough sets because too many inconsistent samples exist in data sets according to the above definition [28]. Dominance rough sets are heavily sensitive to noisy samples. Several mislabeled samples might completely change the trained decision models. Here, we give a toy example to show the influence of noisy data.

Example 1. Table 1 gives a toy example of monotonic classification. There are 10 samples described with an attribute A. These samples are divided into three grades according to decision D. We can see that the decisions of these samples are consistent for the samples with larger attribute values are assigned with the same or better decisions. In this case dependency $\rho_A(D) = 1$.

Now we assume that samples x_1 and x_{10} are mislabeled; and x_1 belongs to d_3, while x_{10} belongs to d_1. As

$$d_1^{\preceq} = \{x_1, \ldots, x_{10}\}, \quad d_2^{\preceq} = \{x_1, x_5, x_6, x_7, x_8, x_9\},$$
$$d_3^{\preceq} = \{x_1, x_8, x_9\},$$

then

$$\underline{R^{\preceq}} d_1^{\preceq} = \{x_1, \ldots, x_{10}\}, \quad \overline{R^{\preceq}} d_1^{\preceq} = \{x_1, \ldots, x_{10}\};$$
$$\underline{R^{\preceq}} d_2^{\preceq} = \emptyset, \quad \overline{R^{\preceq}} d_2^{\preceq} = \{x_1, \ldots, x_{10}\};$$
$$\underline{R^{\preceq}} d_3^{\preceq} = \emptyset, \quad \overline{R^{\preceq}} d_3^{\preceq} = \{x_1, \ldots, x_{10}\}.$$

So $\underline{R_B^{\preceq}} d_2^{\preceq} = \underline{R_B^{\preceq}} d_3^{\preceq}$. We derive $\rho_A(D) = 0$ in this context. From this example, we can see that dominance rough sets are sensitive to noisy data.

The above analysis shows that although the model of dominance rough sets provides a formally theoretic framework for studying monotonic classification, it may not be effective in practice due to noise. We should introduce a robust measure.

3 Rank Entropy and Rank Mutual Information

Information entropy performs well in constructing decision trees. However, it cannot reflect the ordinal structure in monotonic classification. The model of dominance rough sets provides a formal framework for studying monotonic classification; however, it is not robust enough in dealing with real-world tasks. In this section, we introduce a measure, called rank mutual information [27], to evaluate the ordinal consistency between random variables.

Definition 5. *Given $DT = <U, A, D>, B \subseteq A$. The ascending and descending rank entropies of the system with respect to B are defined as*

$$RH_B^{\leq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{|[x_i]_B^{\leq}|}{n}, \tag{1}$$

$$RH_B^{\geq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{|[x_i]_B^{\geq}|}{n}. \tag{2}$$

Example 2 (Continue Example 1).

$$\begin{aligned}
RH_A^{\leq}(U) &= -\frac{1}{10}\sum_{i=1}^{10}\log\frac{|[x_i]_A^{\leq}|}{10} \\
&= -\frac{1}{10}\log\frac{10}{10} - \frac{1}{10}\log\frac{9}{10} - \frac{1}{10}\log\frac{7}{10} - \frac{1}{10}\log\frac{8}{10} \\
&\quad -\frac{1}{10}\log\frac{5}{10} - \frac{1}{10}\log\frac{6}{10} - \frac{1}{10}\log\frac{4}{10} - \frac{1}{10}\log\frac{3}{10} \\
&\quad -\frac{1}{10}\log\frac{2}{10} - \frac{1}{10}\log\frac{1}{10} = 0.7921, \\
RH_D^{\leq}(U) &= -\frac{1}{10}\sum_{i=1}^{10}\log\frac{|[x_i]_D^{\leq}|}{10} \\
&= -\frac{4}{10}\log\frac{10}{10} - \frac{3}{10}\log\frac{6}{10} - \frac{3}{10}\log\frac{3}{10} = 0.5144.
\end{aligned}$$

Since $1 \geq \frac{|[x_i]_B^{\geq}|}{n} \geq 0$, we have $RH_B^{\geq}(U) \geq 0$ and $RH_B^{\leq}(U) \geq 0$. $RH_B^{\leq}(U) = 0$ ($RH_B^{\geq}(U) = 0$) if and only if $\forall x_i \in U$, $[x_i]_B^{\geq} = U$ ($[x_i]_B^{\leq} = U$). Assume $C \subseteq B \subseteq A$. Then $\forall x_i \in U$, we have $[x_i]_B^{\geq} \subseteq [x_i]_C^{\geq}$ ($[x_i]_B^{\leq} \subseteq [x_i]_C^{\leq}$). Accordingly, $RH_B^{\geq}(U) \leq RH_C^{\geq}(U)$ ($RH_B^{\leq}(U) \leq RH_C^{\leq}(U)$).

Definition 6. *Given $<U, A, D>, B \subseteq A, C \subseteq A$, The ascending rank joint entropy of the set U with respect to B and C is defined as*

$$RH_{B\cup C}^{\leq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\leq} \cap [x_i]_C^{\leq}\right|}{n}, \tag{3}$$

and descending rank joint entropy of the set U with respect to B and C is defined as

$$RH_{B\cup C}^{\geq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\geq} \cap [x_i]_C^{\geq}\right|}{n}. \tag{4}$$

Given $DT = <U, A, D>$ and $B \subseteq A$, $C \subseteq A$, we have

$$\begin{aligned}
&RH_{B\cup C}^{\leq}(U) \geq RH_B^{\leq}(U); RH_{B\cup C}^{\leq}(U) \geq RH_C^{\leq}(U); \\
&RH_{B\cup C}^{\geq}(U) \geq RH_B^{\geq}(U); RH_{B\cup C}^{\geq}(U) \geq RH_C^{\geq}(U).
\end{aligned}$$

Given $DT = <U, A, D>, C \subseteq B \subseteq A$. Then we have $RH_{B\cup C}^{\geq}(U) = RH_B^{\geq}(U)$ and $RH_{B\cup C}^{\leq}(U) \leq RH_B^{\leq}(U)$.

Definition 7. *Given $DT = <U, A, D>$, $B \subseteq A$, $C \subseteq A$, If C is known, the ascending rank conditional entropy of the set U with respect to B is defined as*

$$RH_{B|C}^{\leq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\leq} \cap [x_i]_C^{\leq}\right|}{\left|[x_i]_C^{\leq}\right|}, \tag{5}$$

and descending rank conditional entropy of the set U with respect to B is defined as

$$RH_{B|C}^{\geq}(U) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\geq} \cap [x_i]_C^{\geq}\right|}{\left|[x_i]_C^{\geq}\right|}. \tag{6}$$

Given $DT = <U, A, D>, B \subseteq A, C \subseteq A$, we have that

$$RH_{B|C}^{\leq}(U) = RH_{B\cup C}^{\leq}(U) - RH_C^{\leq}(U)$$

and $RH_{B|C}^{\geq}(U) = RH_{B\cup C}^{\geq}(U) - RH_C^{\geq}(U)$.

Given $DT = <U, A, D>, B \subseteq C \subseteq A$, we have that $RH_{B|C}^{\leq}(U) = 0$ and $RH_{B|C}^{\geq}(U) = 0$.

Given $DT = <U, A, D>, B \subseteq A, C \subseteq A$. It is easy to obtain the following conclusions:

1.

$$\begin{aligned}
RH_{B\cup C}^{\leq}(U) &\leq RH_B^{\leq}(U) + RH_C^{\leq}(U), \\
RH_{B\cup C}^{\geq}(U) &\leq RH_B^{\geq}(U) + RH_C^{\geq}(U);
\end{aligned}$$

2.

$$\begin{aligned}
&RH_{B|C}^{\leq}(U) \leq RH_B^{\leq}(U), RH_{B|C}^{\leq}(U) \leq RH_C^{\leq}(U), \\
&RH_{B|C}^{\geq}(U) \leq RH_B^{\geq}(U), RH_{B|C}^{\geq}(U) \leq RH_C^{\geq}(U).
\end{aligned}$$

Definition 8. *Given $DT = <U, A, D>, B \subseteq A, C \subseteq A$. The ascending rank mutual information (ARMI) of the set U between B and C is defined as*

$$RMI^{\leq}(B, C) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\leq}\right| \times \left|[x_i]_C^{\leq}\right|}{n \times \left|[x_i]_B^{\leq} \cap [x_i]_C^{\leq}\right|}, \tag{7}$$

and descending rank mutual information (DRMI) of the set U regarding B and C is defined as

$$RMI^{\geq}(B, C) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\left|[x_i]_B^{\geq}\right| \times \left|[x_i]_C^{\geq}\right|}{n \times \left|[x_i]_B^{\geq} \cap [x_i]_C^{\geq}\right|}. \tag{8}$$

In essence, rank mutual information can be considered as the degree of monotonicity between features B and C. The monotonicity should be kept in monotonic classification. Rank mutual information $RMI^{\leq}(B, D)$ or $RMI^{\geq}(B, D)$ can be used to reflect the monotonicity relevance between features B and decision D. So it is useful for ordinal feature evaluation in monotonic decision tree construction.

Given $DT = <U, A, D>, B \subseteq A, C \subseteq A$, we have that

1.

$$\begin{aligned}
RMI^{\leq}(B, C) &= RH_B^{\leq}(U) - RH_{B|C}^{\leq}(U) \\
&= RH_C^{\leq}(U) - RH_{C|B}^{\leq}(U),
\end{aligned}$$

2.

$$\begin{aligned}
RMI^{\geq}(B, C) &= RH_B^{\geq}(U) - RH_{B|C}^{\geq}(U) \\
&= RH_C^{\geq}(U) - RH_{C|B}^{\geq}(U).
\end{aligned}$$

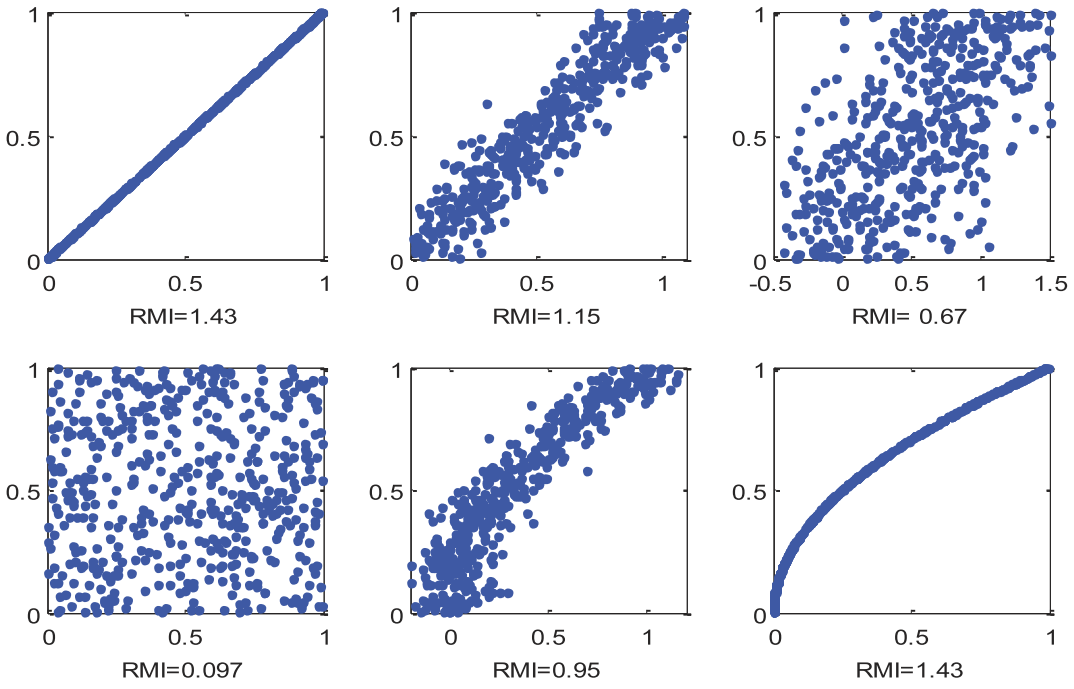


Fig. 1. Scatter plot in different feature spaces and the corresponding rank mutual information of feature pairs.

Given $DT = <U, A, D>, B \subseteq C \subseteq A$, we have that

1. $RMI^{\geq}(B, C) = RH^{\geq}_{B}(U)$;
2. $RMI^{\leq}(B, C) = RH^{\leq}_{B}(U)$.

Given $DT = <U, A, D>$, $B \subseteq A$. $RMI^{\leq}(B, D) \leq RH^{\leq}_{D}(U)$. If $\forall x_i \in U$, we have $[x_i]^{\leq}_{B} \subseteq [x_i]^{\leq}_{D}$, then we say that the decision D is B-ascending consistent and $RMI^{\leq}(B, D) = RH^{\leq}_{D}(U)$. If $\forall x_i \in U$, we have that $[x_i]^{\geq}_{B} \subseteq [x_i]^{\geq}_{D}$, then we say the decision D is B-descending consistent and $RMI^{\geq}(B, D) = RH^{\geq}_{D}(U)$. In addition, if D is B-ascending consistent, then D is also B-descending consistent.

The above analysis tells us that the maximum of rank mutual information between features and decision equals the rank entropy of decision, and rank mutual information arrives at its maximum if the classification task is monotonically consistent with respect to these features. In this case, addition of any new feature will not increase the rank mutual information. In constructing decision trees, we add features one by one for partition the samples in a node until the mutual information does not increase by adding any new feature [10], [11]. Thus, the algorithm stops there.

Shannon's entropy is widely employed and performs well in learning decision trees. The rank entropy and rank mutual information not only inherits the advantage of Shannon's entropy, but also measure the degree of monotonicity between features. We have the following conclusions.

Given $DT = <U, A, D>, B \subseteq A$ and $C \subseteq A$. If we replace ordinal subsets $[x_i]^{\leq}_{B}$ with equivalence classes $[x_i]_B$, where $[x_i]_B$ is the subset of samples taking the same feature values as x_i in terms of feature set B, then we have

1. $RH^{\leq}(B) = H_B(U)$, $RH^{\geq}(B) = H_B(U)$;
2. $RH^{\leq}_{B|C}(U) = RH_{B|C}(U)$, $RH^{\geq}_{B|C}(U) = RH_{B|C}(U)$;
3. $RMI^{\leq}(B, C) = MI(B, C)$; $RMI^{\geq}(B, C) = MI(B, C)$.

The above properties show that rank entropy, rank conditional entropy, and rank mutual information will degenerate to Shannon's one if we replace $[x_i]^{\leq}_{B}$ with $[x_i]_B$. Thus, we can consider that rank entropy is a natural generalization from Shannon's entropy. As we know Shannon's entropy is robust in measuring relevance between features and decision for classification problems, we desire that rank entropy and rank mutual information are also robust enough in dealing with noisy monotonic tasks.

Fig. 1 shows six scatter plots of 500 samples in different 2D feature spaces, where the relation between the first three feature pairs are linear; and some features are contaminated by noise; the fourth feature pair is completely irrelevant to each other; the final two are nonlinear monotonous. We compute the rank mutual information of these feature pairs. We can see that the first and last pairs return the maximal values of rank mutual information among these feature pairs as these pairs are nearly linear or nonlinear monotonous, while the feature pair in Subplot four gets a very small value as these two features are irrelevant. The example shows that rank mutual information is effective in measuring ordinal consistency.

4 Constructing Decision Trees Based on Rank Entropy

Decision tree is an effective and efficient tool for extracting rules and building classification models from a set of samples [10], [11], [12], [29], [30], [31]. There are two basic issues in developing a greedy algorithm for learning decision trees [37]: feature evaluation and pruning strategies, where feature evaluation plays the central role in constructing decision trees; it determines which attribute should be selected for partition the samples if the samples in a node do not belong to the same class. As to classical classification tasks, Shannon's entropy is very effective. In this section, we substitute Shannon's entropy with rank entropy for monotonic decision trees.

We only consider univariate binary trees in this work. Namely, in each node we just use one attribute to split the samples and the samples are divided into two subsets. Assume U_i the subset of samples in the current node and a_i is selected for splitting U_i in this node. Then the samples are divided into U_{i1} and U_{i2}, where $U_{i1} = \{x \in U_i | v(a_i, x) \leq c\}$ and $U_{i2} = \{x \in U_i | v(a_i, x) > c\}$.

Before introducing the algorithm, the following rules should be considered in advance [15].

1. Splitting rule S: give S to generate partition in each node;
2. Stopping criterion H: determine when to stop growing the tree and generate leaf nodes;
3. Labeling rule L: assign class labels to leaf nodes.

As to splitting rule S, we here use rank mutual information to evaluate the quality of features. Given attributes $A = \{a_1, a_2, \ldots, a_N\}$ and a subset of samples U_i, we select attribute a and parameter c satisfying the following condition:

$$\begin{aligned} a &= \arg\max_{a_j} RMI^{\leq}(a_j, c, D) \\ &= \arg\max_{a_j} -\frac{1}{|U_i|} \sum_{x \in U_j} \log \frac{\left|[x]_{a_j}^{\leq}\right| \times \left|[x]_D^{\leq}\right|}{|U_i| \times \left|[x]_{a_j}^{\leq} \cap [x]_D^{\leq}\right|}, \end{aligned} \tag{9}$$

where c is a number to split the value domain of a_j such that the split yields the largest rank mutual information between a_j and D computed with the samples in the current node. As binary trees are used, we just require one number to split the samples in each node.

Now we consider the stopping criterion. Obviously, if all the samples in a node belong to the same class, the algorithm should stop growing the tree in this node. Moreover, in order to avoid overfitting data, we also stop growing the tree if the rank mutual information produced by the best attribute is smaller than a threshold ε. Moreover, some other prepruning techniques can also be considered here [33].

Regarding labeling rule L, if all the samples in a leaf node come from the same class, this class label is assigned to the leaf node. However, if the samples belong to different classes, we assign the median class of samples to this leaf. Furthermore, if there are two classes having the same number of samples and the current node is a left branch of its parent node, we then assign the worse class to this node; otherwise, we assign the better class to it.

The monotonic decision tree algorithm based on rank mutual information is formulated in Table 2.

Now, we study the properties of monotonic decision trees generated with the above procedure.

Definition 9. *Given ordinal decision tree T, the rule from the root node to a leaf l is denoted by R_l. If two rules R_l and R_m are generated from the same attributes, we say R_l and R_m are comparable; otherwise they are not comparable. In addition, we denote $R_l < R_m$ if the feature value of R_l is less than R_m. R_l and R_m are also called left node and right node, respectively. As to a set of rules R, we call it is monotonically consistent if for any comparable pair of rules R_l and R_m, if $R_l < R_m$, then $D(R_l) < D(R_m)$, where $D(R_l)$ and $D(R_m)$ are the decisions of these rules; otherwise, we say R is not monotonically consistent.*

TABLE 2
Algorithm Description

REMT: Rank Entropy Based Decision Tree

input: criteria: attributes of samples.
decision: decision of samples.
ε: stopping criterion: If the maximal rank mutual information is less than ε, the branch stops growing
output: monotonic decision tree T.
begin:
1 generate the root node.
2 if the number of sample is 1 or all the samples come from the same class, the branch stops growing
3 otherwise,
for each attribute a_i,
for each $c_j \in a_i$,
divide samples into two subsets according to c_j
if $v(a_i, x) \leq c_j$, then $v(a_i, x) = 1$.
else $v(a_i, x) = 2$.
compute $RMI_{c_j} = RMI(a_i, D)$.
end j.
$c_j^* = \arg\max_j RMI_{c_j}$.
end i.
4 select the best feature a and the corresponding splitting point: $c^* = \arg\max_i \max_j RMI(a_i, c_j)$
5 if $RMI(a, D) < \varepsilon$, then stop.
6 build a new node and split samples with a and c^*.
7 recursively produce new splits according to the above procedure until stopping criterion is satisfied.
8 end

Property 1. *Given $<U, A, D>$, if U is monotonically consistent, then the rules derived from the ordinal decision trees are also monotonically consistent.*

Proof. Let R_l and R_m be a pair of comparable rules, and $R_l < R_m$. We prove that $D(R_l) < D(R_m)$. If U is monotonically consistent, then for any $x, y \in U$, $v(D, x) \neq v(D, y)$, we can get an attribute $a_i \in A$, such that $v(a_i, x) \neq v(a_i, y)$. That is, $\exists a_i \in A$, it can separate x and y. So the samples in the node of x belong to the same decision, while the samples in the node of y belong to another decision. Thus, if we want to derive $D(R_l) < D(R_m)$, we just need to show if $\forall x, y \in U_i \cup U_j$, $v(D, x) < v(D, y)$, we have $x \in U_i$, $y \in U_j$. As U is monotonically consistent, for $\forall x, y \in U_i \cup U_j$ and $v(D, x) < v(D, y)$, $\exists a_i \in A$, such that $v(a_i, x) < v(a_i, y)$. Then the parent node of R_l and R_m can get the maximal $RMI(a, D)$. We have $v(a, x) < v(a, y)$, namely, $x \in U_i$ and $y \in U_j$. □

The above analysis shows us that with algorithm REMT we can derive a monotonic decision tree from a monotonically consistent data set.

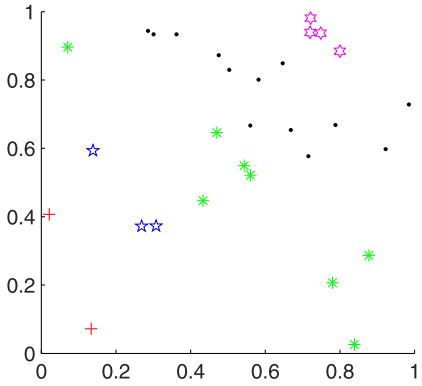


Fig. 2. Artificial data, where 30 samples are divided into 5 classes.

We give a toy example to show this property. We generate a data set of 30 samples described with two features, as shown in Fig. 2. These samples are divided into five classes. Now we employ CART and REMT to train decision trees, respectively. Figs. 3 and 4 give the trained models.

CART returns a tree with eleven leaf nodes, while REMT generates a tree with nine leaf nodes. Most of all, the tree trained with CART is not monotonically consistent for there are two pairs of conflicted rules. From left to right, we can see the fourth rule is not monotonically consistent with the fifth rule; besides, the sixth rule is not monotonically consistent with the seventh rule, where the objects with the better features get the worse decision. However, REMT obtains a consistent decision tree.

5 Experiment Analysis

There are several techniques to learn decision models for monotonic classification. In order to show the effectiveness of the proposed algorithm, we conduct some numerical experiments with artificial or real-world data sets. We compare the proposed algorithm with others on real-world classification tasks.

First, we introduce the following function to generate monotone data sets:

$$f(x_1, x_2) = 1 + x_1 + \frac{1}{2}\left(x_2^2 - x_1^2\right), \tag{10}$$

where x_1 and x_2 are two random variables independently drawn from the uniform distribution over the unit interval.

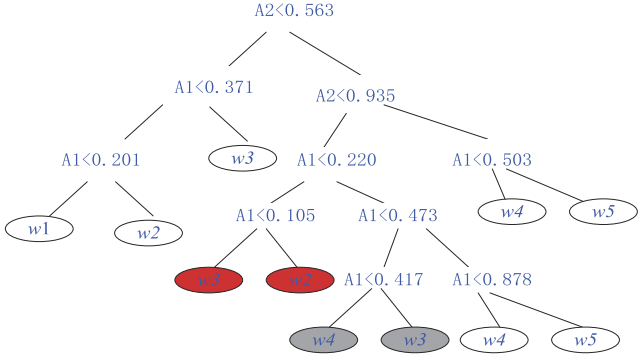


Fig. 3. Nonmonotonic decision tree trained with CART, where two pairs of rules are not monotonically consistent.

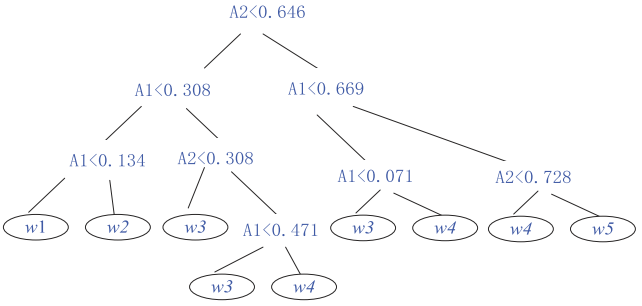


Fig. 4. Monotonic decision tree trained with REMT, where all rules are monotonically consistent.

In order to generate ordered class labels, the resulting numeric values were discretized into k intervals $[0, 1/k]$, $(1/k, 2/k], \ldots, (k-1/k, 1]$. Thus each interval contains approximately the same number of samples. The samples belonging to one of the intervals share the same rank label. Then we form a k-class monotonic classification task. In this experiment, we try $k = 2, 4, 6$, and 8, respectively. The data sets are given in Fig. 5.

We here use the mean absolute error for evaluating the performance of decision algorithms, computed as

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i|, \tag{11}$$

where N is the number of samples in the test set and $\hat{y}_i$ is the output of the algorithm and y_i is the real output of the i'th sample.

We first study the performance of algorithms on different numbers of classes. We generate a set of artificial data sets with 1,000 samples and the class number varies from 2 to 30. Then we employ CART, Rank Tree [17], OLM [6], OSDL [41], and REMT to learn and predict the decisions. OLM is an ordinal learning model introduced by Ben-David et al., while OSDL is ordinal stochastic dominance learner based on associated cumulative distribution [41].

Based on fivefold cross-validation technique, the average performance is computed and given in Table 3. REMT

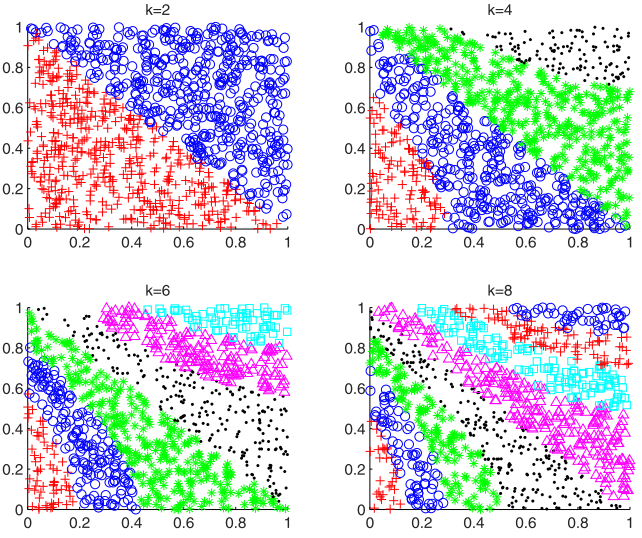


Fig. 5. Synthetic monotone data sets, where the samples are divided into 2, 4, 6, and 8 classes, respectively.

TABLE 3
MAE on Artificial Data

Classes	CART	Rank Tree	REMT	OLM	OSDL
2	**0.0370**	0.8000	0.4200	0.0570	0.0640
4	0.0741	0.1451	**0.0561**	0.1470	0.0910
6	0.1110	0.2959	**0.0770**	0.2180	0.1200
8	0.1479	0.3307	**0.1276**	0.2600	0.1620
10	0.1870	0.4560	**0.1310**	0.3270	0.2010
20	0.3579	1.0029	**0.2569**	0.5060	0.4130
30	0.5146	1.4862	**0.3735**	0.7010	0.5840

yields the least errors in all the cases except the case two classes are considered.

In addition, we also consider the influence of sample numbers on the performance of trained models. We first generate artificial data of 1,000 samples and 4 classes. And then we randomly draw training samples from this set. The size of training samples ranges from 4 to 36. In this process, we guarantee there is at least one representative sample from each class. The rest samples are used in testing for estimating the performance of the trained models. The curves of loss varying with number of training samples are shown in Fig. 6. We can see that REMT is more precise than CART and rank trees, no matter how many training samples are used. In addition, we can also see that the difference between rank tree and REMT gets smaller and smaller as the number of training samples increases.

In order to test how our approach behaves in real-world applications, we collected 12 data sets. Four data sets come from UCI repository: Car, CPU performance, Ljubljana breast cancer, and Boston housing pricing. Bankruptcy comes from the experience of a Greek industrial development bank financing industrial and commercial firms and the other data sets were obtained from weka homepage (http://www.cs.waikato.ac.nz/ml/weka/).

Before training decision trees, we have to preprocess the data sets. Because we use ascending rank mutual information as the splitting rule, we assume that larger rank value should come from larger feature values; we call this positive monotonicity. In practice, we may confront the case that the worse feature value should get the better ranks. This is called negative monotonicity. If we use REMT, we should transform the problem of negative monotonicity to a positive monotonicity task. There are several solution to this objective. We compute reciprocal of feature values if negative monotonicity happens.

For each data set, we randomly drew $n * Ni$ samples as a training set each time, where Ni is the number of classes, $\mathrm{n} = 1, 2, 3$, and so on. At least one sample from each class was drawn in each round when we generate the training set. The remained samples are used as the test set. We compute the mean absolute loss as the performance of the trained model. The experiment was repeated 100 times. The average over all 100 results is output as the performance of the models. The results are given in Fig. 7, where $\varepsilon = 0.01$.

Regarding the curves in Fig. 7, we see REMT is much better than CART and Rank Tree in most cases, except data set Breast and CHNUnvRank. Moreover, we can also see that if the size of training sets is very small, REMT is much better than Rank Tree and CART. As the number of training samples increases, the superiority becomes less and less. This trend shows REMT is more effective than CART and Rank Tree if the size of training samples is small.

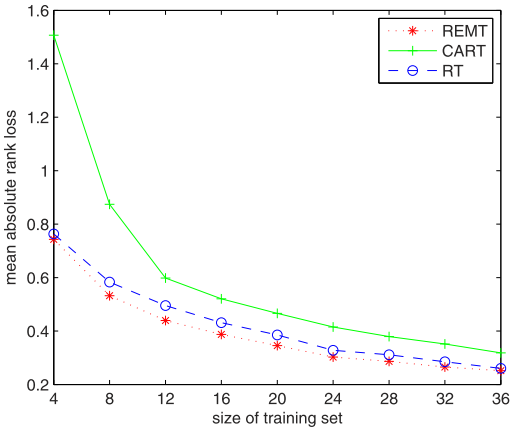


Fig. 6. Loss curves of CART, Rank Tree, and REMT, where the size of training samples gradually increases.

In order to compare the performance when data sets are monotonic, we now relabel the samples so as to generate monotonic training sets. Before training decision trees, we first introduced a monotonization algorithm to revise the labels of some samples and generated monotone training data sets [18]. And then we learned decision trees and predicted labels of test samples. The variation of loss varying with the numbers of training samples is given in Fig. 8. The same conclusion can be derived.

Finally, we test these algorithms on the data sets based on fivefold cross-validation technique. Table 4 presents the mean absolute loss yielded with different learning algorithms, including REMT, CART, Rank Tree, OLM, and OSDL. Among 12 tasks, REMT obtains the best performance on six, while CART, Rank tree, OLM, and OSDL produce 2, 0, 2, and 3 best results, respectively. REMT outperform CART over ten tasks, and it produce better performances than Rank tree, OLM, and OSDL on 9, 10, and 9 tasks, respectively. As a whole, REMT produces the best average performance over 12 tasks, The experimental results show that REMT is better than CART, Rank tree, OLM, and OSDL in most cases.

We also calculate the mean absolute loss of these algorithms on monotonized data sets. The generalization performance based on cross validation is given in Table 5. Comparing the results in Tables 4 and 5, we can see all the mean absolute errors derived from different algorithms decrease if data are monotonized. Furthermore, REMT outperforms CART, Rank tree, OLM, and OSDL on 11, 9, 9, and 10 tasks, respectively. The results show REMT is also more effective than other techniques if training sets are monotonized.

6 Conclusions and Future Work

Monotonic classification is a kind of important tasks in decision making. There is a constraint in these tasks that the features and decision are monotonically consistent. That is, the objects with better feature values should not get worse decisions. Classical learning algorithms cannot extract this monotonous structure from data sets, thus they are not applicable to these tasks. Some monotonic decision

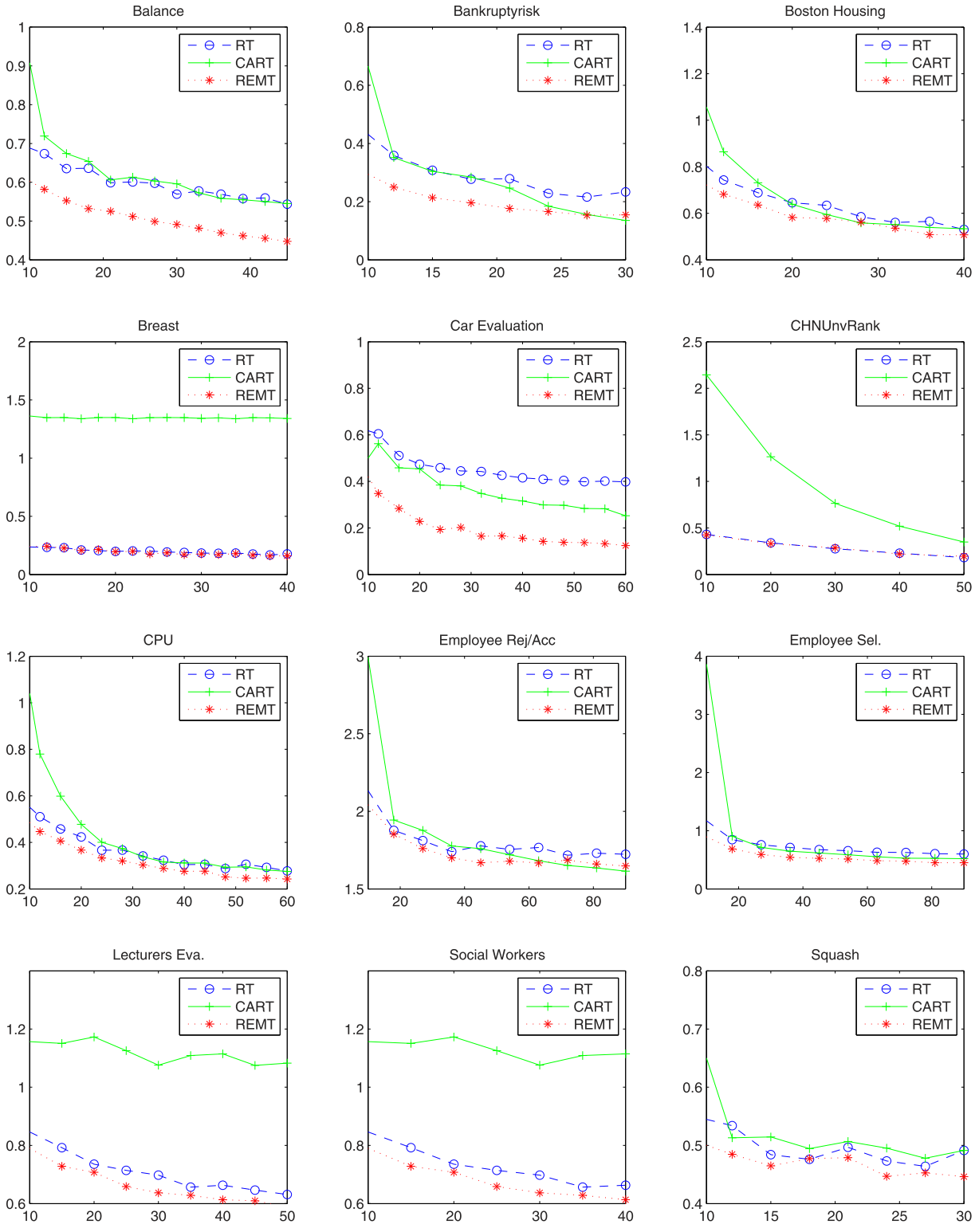


Fig. 7. Average performance of real-world tasks before monotonization, where the x-coordinate is the number of training samples and y-coordinate is mean absolute error.

algorithms have also been developed in these years by integrating monotonicity with indexes of separability, such as Gini, mutual information, dependency, and so on. However, the comparative experiments showed that noisy samples have great impact on these algorithms. In this work, we combine the advantage of robustness of Shannon's entropy with the ability of dominance rough sets in extracting ordinal structures from monotonic data sets by introducing rank entropy. We improve the classical decision tree algorithm with rank mutual information and design a new decision tree technique for monotonic classification. With the theoretic and numerical experiments, the following conclusions are derived.

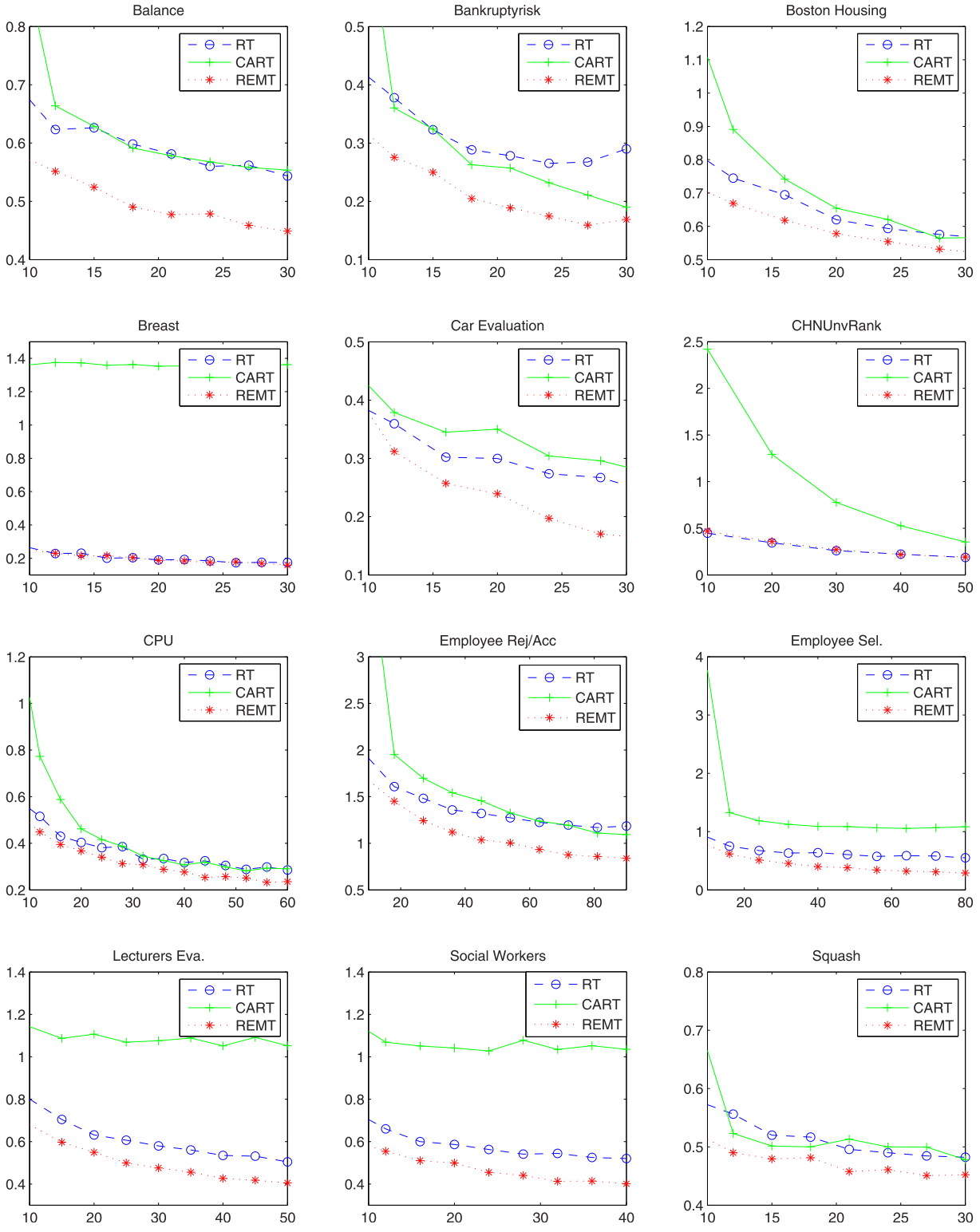


Fig. 8. Average performance curves of real-world tasks after monotonization, where the x-coordinate is the number of training samples and y-coordinate is mean absolute error.

1. Monotonic classification is very sensitive to noise; several noisy samples may completely change the evaluation of feature quality. A robust measure of feature quality is desirable.
2. Mutual information is a robust index of feature quality in classification learning; however it cannot reflect the ordinal structure in monotonic classification. Rank mutual information combines the advantage of information entropy and dominance rough sets. This new measure cannot only measure the monotonous consistency in monotonic classification, but also is robust to noisy samples.
3. Rank entropy-based decision trees can produce monotonically consistent decision trees if the given

TABLE 4
Mean Absolute Loss (Before Monotonization)

Dataset	REMT	CART	Rank Tree	OLM	OSDL
balance	**0.5171**	0.5830	0.7678	0.6176	0.8688
bankrupty	**0.1250**	**0.1250**	0.4679	0.6464	0.3071
breast	0.0916	1.3519	0.1172	0.3320	**0.0859**
car	**0.1696**	0.2436	0.3970	0.4676	0.3102
CHNUnv	0.1900	**0.1800**	0.1900	1.0900	0.5200
CPU	**0.2641**	0.2692	0.3731	0.4159	0.3835
empl. rej/acc	**1.2347**	1.3800	1.3998	1.2920	1.2830
empl. sel.	0.3808	0.5057	0.6389	0.5062	**0.3421**
lect. eva.	**0.3471**	0.3726	0.5015	0.6050	0.4020
workers	0.4451	1.0017	0.5671	0.5630	**0.4270**
housing	0.4464	1.4406	0.4130	**0.4091**	1.0770
squash	0.4430	1.5275	0.4325	**0.3236**	0.5964
Average	**0.3879**	0.5817	0.5222	0.6057	0.5503

TABLE 5
Mean Absolute Loss (After Monotonization)

Dataset	REMT	CART	Rank Tree	OLM	OSDL
balance	0.3408	0.4768	0.5728	0.2304	**0.1504**
bankrupty	**0.1500**	0.1786	0.4821	0.6679	0.3571
breast	0.0744	1.3462	**0.0686**	0.3319	0.0858
car eva.	0.0294	0.0462	0.2287	0.0382	**0.0266**
CHNUnv	0.1900	**0.1800**	0.1900	1.0900	0.5200
CPU	**0.2641**	0.2736	0.3253	0.4159	0.3835
empl. rej/acc	0.5603	0.5685	0.6765	**0.5500**	0.5740
empl. sel.	**0.1269**	1.0145	0.4726	0.2215	0.1311
lect. eva.	**0.0929**	0.1054	0.2516	0.1380	0.1120
workers	**0.1721**	1.0580	0.4671	0.1920	0.1860
housing	0.3595	0.3991	**0.3497**	0.6541	0.4783
squash	0.4430	1.5275	0.5257	**0.3473**	0.5745
Average	**0.2336**	0.5123	0.3842	0.4064	0.2983

training sets are monotonically consistent. When the training data is nonmonotonic, our approach produces a nonmonotonic classifier, but its performance is still good.

It is remarkable that sometimes just a fraction of features satisfies the monotonicity constraint with decision in real-world tasks. Some of features do not satisfy this constraint. In this case, decision rules should be in the form that if Feature 1 is equal to a_1 and Feature 2 is better than a_2, then decision D should be no worse than d_k. This problem is called partial monotonicity. We require some measures to evaluate the quality of two types of features at the same time when we build decision trees from this kind of data sets. We will work on this problem in the future.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to the anonymous reviewers for their constructive comments, which is helpful for improving the manuscript. This work is supported by National Natural Science Foundation of China under Grants 60703013 and 10978011, Key Program of National Natural Science Foundation of China under Grant 60932008, National Science Fund for Distinguished Young Scholars under Grant 50925625 and China Postdoctoral Science Foundation. Dr. Hu is supported by The Hong Kong Polytechnic University (G-YX3B).

REFERENCES

[1] J. Wallenius, J.S. Dyer, P.C. Sishburn, R.E. Steuer, S. Zionts, and K. Deb, "Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead," *Management Science,* vol. 54, no. 7, pp. 1336-1349, 2008.

[2] B. Zhao, F. Wang, and C.S. Zhang, "Block-Quantized Support Vector Ordinal Regression," *IEEE Trans. Neural Networks,* vol. 20, no. 5, pp. 882-890, May 2009.

[3] B.Y. Sun et al., "Kernel Discriminant Learning for Ordinal Regression," *IEEE Trans. Knowledge and Data Engineering,* vol. 22, no. 6, pp. 906-910, June 2010.

[4] C. Zopounidis and M. Doumpos, "Multicriteria Classification and Sorting Methods: A Literature Review," *European J. Operational Research,* vol. 138, pp. 229-246, 2002.

[5] 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.

[6] A. Ben-David, L. Sterling, and Y.H. Pao, "Learning and Classification of Monotonic Ordinal Concepts," *Computational Intelligence,* vol. 5, pp. 45-49, 1989.

[7] A. Ben-David, "Automatic Generation of Symbolic Multiattribute Ordinal Knowledge-Based DSSs: Methodology and Applications," *Decision Sciences,* vol. 23, pp. 1357-1372, 1992.

[8] E. Frank and M. Hall, "A Simple Approach to Ordinal Classification," *Proc. 12th European Conf. Machine Learning,* pp. 145-156, 2001.

[9] J.P. Costa and J.S. Cardoso, "Classification of Ordinal Data Using Neural Networks," *Proc. 16th European Conf. Machine Learning,* pp. 690-697, 2005.

[10] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[11] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[12] L. Breiman et al., *Classification and Regression Trees.* Chapman and Hall, 1993.

[13] J. Mingers, "An Empirical Comparison of Selection Measures for Decision-Tree Induction," *Machine Learning,* vol. 3, no. 4, pp. 319-342, 1989.

[14] A. Ben-David, "Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms," *Machine Learning,* vol. 19, pp. 29-43, 1995.

[15] R. Potharst and J.C. Bioch, "Decision Trees for Ordinal Classification," *Intelligent Data Analysis,* vol. 4, pp. 97-111, 2000.

[16] K. Cao-Van and B.D. Baets, "Growing Decision Trees in an Ordinal Setting," *Int'l J. Intelligent Systems,* vol. 18, pp. 733-750, 2003.

[17] F. Xia, W.S. Zhang, F.X. Li, and Y.W. Yang, "Ranking with Decision Tree," *Knowledge and Information Systems,* vol. 17, pp. 381-395, 2008.

[18] W. Kotlowski and R. Slowinski, "Rule Learning with Monotonicity Constraints," *Proc. 26th Ann. Int'l Conf. Machine Learning,* pp. 537-544, 2009.

[19] A. Jimnez, F. Berzal, and J.-C. Cubero, "POTMiner: Mining Ordered, Unordered, and Partially-Ordered Trees," *Knowledge and Information Systems,* vol. 23, no. 5, pp. 199-224, 2010.

[20] S. Greco, B. Matarazzo, and R. Slowinski, "Rough Approximation of a Preference Relation by Dominance Relations," *European J. Operational Research,* ICS Research Report 16/96, vol. 117, pp. 63-83, 1999.

[21] S. Greco, B. Matarazzo, and R. Slowinski, "Rough Sets Methodology for Sorting Problems in Presence of Multiple Attributes and Criteria," *European J. Operational Research,* vol. 138, pp. 247-259, 2002.

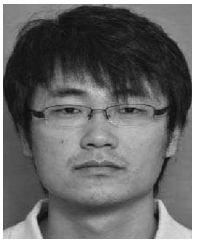
[22] S. Greco, B. Matarazzo, and R. Slowinski, "Rough Approximation by Dominance Relations," *Int'l J. Intelligent Systems,* vol. 17, pp. 153-171, 2002.

[23] J.W.T. Lee and E.C.C. Tsang, "Rough Sets and Ordinal Reducts," *Soft Computing,* vol. 10, pp. 27-33, 2006.

[24] Q.H. Hu, D.R. Yu, and M.Z. Guo, "Fuzzy Preference-Based Rough Sets," *Information Sciences,* vol. 180, no. 10, pp. 2003-2022, 2010.

[25] A. Ben-David, L. Sterling, and T. Tran, "Adding Monotonicity to Learning Algorithms May Impair Their Accuracy," *Expert Systems with Applications,* vol. 36, pp. 6627-6634, 2009.
[26] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius, and S. Zionts, "Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years," *Management Science,* vol. 38, pp. 645-654, 1992.
[27] Q.H. Hu, M.Z. Guo, D.R. Yu, and J.F. Liu, "Information Entropy for Ordinal Classification," *Science in China Series F: Information Sciences,* vol. 53, no. 6, pp. 1188-1200, 2010.
[28] S. Greco, B. Matarazzo, R. Slowinski, and J. Stefanowski, "Variable Consistency Model of Dominance-Based Rough Sets Approach," *Proc. Second Int'l Conf. Rough Sets and Current Trends in Computing (RSCTC '00),* pp. 170-181, 2001.
[29] B. Chandra and P.P. Varghese, "Fuzzy SLIQ Decision Tree Algorithm," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 38, no. 5, pp. 1294-1301, Oct. 2008.
[30] H.W. Hu, Y.L. Chen, and K. Tang, "Dynamic Discretization Approach for Constructing Decision Trees with a Continuous Label," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 11, pp. 1505-1514, Nov. 2009.
[31] D. Hush and R. Porter, "Algorithms for Optimal Dyadic Decision Trees," *Machine Learning,* vol. 80, no. 1, pp. 85-107, 2010.
[32] R.V. Kamp, A.J. Feelders, and N. Barile, "Isotonic Classification Trees," *Proc. Eight Int'l Symp. Intelligent Data Analysis,* pp. 405-416, 2009.
[33] A.J. Feelders and M. Pardoel, "Pruning for Monotone Classification Trees," *Proc. Fifth Int'l Symp. Intelligent Data Analysis,* pp. 1-12, 2003.
[34] D. Cai, "An Information-Theoretic Foundation for the Measurement of Discrimination Information," *IEEE Trans. Knowledge and Data Eng.,* vol. 22, no. 9, pp. 1262-1273, Sept. 2010.
[35] 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.
[36] J.C. Bioch and V. Popova, "Rough Sets and Ordinal Classification," *Proc. 12th Belgian-Dutch Artificial Intelligence Conf. (BNAIC '00),* pp. 85-92, 2000.
[37] J.C. Bioch and V. Popova, "Labelling and Splitting Criteria for Monotone Decision Trees," *Proc. 12th Belgian-Dutch Conf. Machine Learning (BENELEARN '02),* pp. 3-10, 2002.
[38] J.C. Bioch and V. Popova, "Monotone Decision Trees and Noisy Data," *Proc. 14th Belgian-Dutch Conf. Artificial Intelligence (BNAIC '02),* pp. 19-26, 2002.
[39] V. Popova, "Knowledge Discovery and Monotonicity," PhD thesis, Erasmus Univ., 2004.
[40] A. Feelders, "Monotone Relabeling in Ordinal Classification," *Proc. IEEE Int'l Conf. Data Mining (ICDM '10),* pp. 803-808, 2010.
[41] C.-V. Kim, "Supervised Ranking from Semantics to Algorithms," PhD thesis, Ghent Univ., 2003.

Qinghua Hu received the BE, ME, and PhD degrees from Harbin Institute of Technology, China, in 1999, 2002, and 2008, respectively. Currently, he is working as an associate professor with Harbin Institute of Technology and a postdoctoral fellow with the Hong Kong Polytechnic University. His research interests include intelligent modeling, data mining, knowledge discovery for classification, and regression. He is a PC cochair of RSCTC 2010 and severs as referee for a great number of journals and conferences. He has published more than 70 journal and conference papers in the areas of pattern recognition and fault diagnosis. He is a member of the IEEE.



Xunjian Che received the BE degree from Harbin Institute of Technology in 2009. Currently, he is working toward the master's degree from the Harbin Institute of Technology. His research interests include large-margin learning theory, preference learning, and monotonic classification.

Lei Zhang received the BS degree from the Shenyang Institute of Aeronautical Engineering, China, in 1995 and the MS and PhD degrees in electrical and engineering from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to 2006, he was a postdoctoral fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. Since January 2006, he has been an assistant professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include image and video processing, biometrics, pattern recognition, multisensor data fusion, machine learning and optimal estimation theory, etc. He is a member of the IEEE.

David Zhang received the BSc degree in computer science from Peking University, the MSc degree in computer science in 1982, and the PhD degree in 1985 from the Harbin Institute of Technology (HIT). From 1986 to 1988, he was a postdoctoral fellow at Tsinghua University and then an associate professor at the Academia Sinica, Beijing. In 1994, he received the second PhD degree in electrical and computer engineering from the University of Waterloo, Ontario, Canada. Currently, he is working as a head in Department of Computing, and a chair professor at the Hong Kong Polytechnic University where he is the founding director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. He also serves as visiting chair professor in Tsinghua University, and an adjunct professor in Shanghai Jiao Tong University, Peking University, Harbin Institute of Technology, and the University of Waterloo. He is the founder and editor-in-chief in *International Journal of Image and Graphics (IJIG)*, book editor of the *Springer International Series on Biometrics (KISB)*, organizer in the first International Conference on Biometrics Authentication (ICBA), associate editor of more than 10 international journals including *IEEE Transactions and Pattern Recognition*, technical committee chair of IEEE CIS, and the author of more than 10 books and 200 journal papers. He is a croucher senior research fellow, distinguished speaker of the IEEE Computer Society, and a fellow of the IEEE and IAPR.

Maozu Guo received the bachelor and master's degrees from the Department of Computer Sciences, Harbin Engineering University, in 1988 and 1991, respectively, and the PhD degree from the Department of Computer Sciences, Harbin Institute of Technology in 1997. Currently, he is working as a director of the Natural Computation Division, Harbin Institute of Technology, program examining expert of Information Science Division of NSFC, senior member of China Computer Federation (CCF), and member of CCF Artificial Intelligence and Pattern Recognition Society, member of Chinese Association for Artificial Intelligence (CAAI), and standing committee member of Machine Learning Society of CAAI. His research interests include machine learning and data mining, computational biology and bioinformatics, advanced computational models, image process, and computer vision. He has implemented several projects from the Natural Science Foundation in China (NSFC), National 863Hi-tech Projects, the Science Fund for Distinguished Young Scholars of Heilongjiang Province, International Cooperative Project. He has won one second prize of the Province Science and Technology Progress, and one third prize of the Province Natural Science. He has published more than 100 papers in journals and conferences.

Daren Yu received the ME and PhD degrees from Harbin Institute of Technology, China, in 1988 and 1996, respectively. Since 1988, he has been working at the School of Energy Science and Engineering, Harbin Institute of Technology. His main research interests include modeling, simulation, and control of power systems. He has published more than 100 conference and journal papers on power control and fault diagnosis.