Knowledge-Based Systems 31 (2012) 8–18

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Large-margin feature selection for monotonic classification

Qinghua Hu *, Weiwei Pan, Yanping Song, Daren Yu

Harbin Institute of Technology, Harbin 150001, Heilongjiang, PR China

ARTICLE INFO

Article history:
Received 2 August 2010
Received in revised form 13 January 2012
Accepted 13 January 2012
Available online 30 January 2012

Keywords:
Monotonic classification
Ordinal classification
Monotonicity constraint
Feature selection
Classification margin

ABSTRACT

Monotonic classification plays an important role in the field of decision analysis, where decision values are ordered and the samples with better feature values should not be classified into a worse class. The monotonic classification tasks seem conceptually simple, but difficult to utilize and explain the order structure in practice. In this work, we discuss the issue of feature selection under the monotonicity constraint based on the principle of large margin. By introducing the monotonicity constraint into existing margin based feature selection algorithms, we design two new evaluation algorithms for monotonic classification. The proposed algorithms are tested with some artificial and real data sets, and the experimental results show its effectiveness.

1. Introduction

Monotonic classification, also called ordinal classification with monotonicity constraints, refers to a class of special classification tasks in some real-world applications, such as multicriteria decision making, risk analysis, customer satisfaction evaluation and information retrieval [9,11,12,14]. In these tasks the instances are assigned with a set of ordered decisions and the instance with better feature values should not be assigned with a worse decision.

Some researchers have discussed monotonic classification tasks. In 1989, Ben-David et al. pointed out that the classical decision tree algorithms did not consider the monotonicity constraint. Even given a consistent data set, these algorithms might produce inconsistent decision trees. They developed several techniques to learn ordinal decision trees [2,3]. In 2000, Potharst et al. developed an order-preserving tree-generation algorithm for ordinal classification and this algorithm can also be used for repairing non-monotonic decision trees [37]. In 2001, Frank and Hall presented a simple method that the standard classification algorithms can make use of ordering information [10]. In 2002, Potharst et al. gave an extensive review on classification trees for problems with monotonicity constraints [38]. In 2011, Hu et al. designed a decision tree algorithm (REMI) based on rank mutual information. It has been proved that if the training samples are monotonically consistent, the proposed algorithm can get monotonically consistent decision trees [19]. In 2008, Barile and Feelders described a monotone classification algorithm by minimizing the mean absolute prediction error for ordinal classification tasks [1]. In addition, Dembczynski et al. introduced a rule induction algorithm for ordinal classification with monotonicity constraint [7], where the experiments showed considering the monotonicity constraint can improve the classification accuracy. Cardoso and Costa proposed a large margin solution to ordinal classification by reducing a multiclass problem to a set of binary problems [4]. Then the binary problems are solved with support vector machines and neural networks. By replacing the indiscernibility relations with dominance relations, Greco et al. proposed the dominance-based rough set approach (DRSA) for dealing with ordinal classification [15]. Then a collection of extension work based on dominance-based rough set approach were proposed [22,23,39,40,44].

Feature selection is considered as one of the fundamental problems in pattern recognition and machine learning [5,17,25]. Numerical experiments show that discarding the irrelevant features can not only reduce system complexity, but also enhance system performance [31,36,42,43,46]. Feature selection has attracted much attention in last decade [13,16,26,45,48]. Roughly speaking, there are two classes of techniques for feature selection: filter and wrapper. The wrapper methods evaluate features with the performance of a learning algorithm, while the filter methods select features with some measures independent of classification algorithms. Distance measures [28], information measures [32], similarity [34] and dependency [35] were discussed. Recently, classification margin is widely discussed in evaluating features. Both theoretical and experimental analysis shows that features yielding large margin can also produce good generalization performance [45].

Compared with feature selection for general classification tasks, little attention has been paid to the issue of feature selection for monotonic classification these years. An algorithm for feature selection in ordinal decision systems was constructed by evaluating features with approximation quality [47]. From the perspective

* Corresponding author.
E-mail address: huqinghua@hit.edu.cn (Q. Hu).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.01.011

of information theory, Hu generalized Shannon's entropy ordinal classification and fuzzy ordinal classification, the proposed indexes are employed to evaluate the monotonicity degree between features and decision [20,21]. Xu et al. discussed the feature selection in ordered information without decision by the concept of plausibility and belief consistent sets [50].

Margin-based methods are considered as one of the most effective algorithms [8,24,33,45,48] in feature selection. Relief based on margin was studied and some extension algorithms were proposed, including ReliefF [41], RReliefF [27], Simba [6] and I-Relief [45]. In this paper, along with the principle of large-margin, we introduce margin-based feature selection algorithms for monotonic classification by incorporating the monotonicity constraint into tackling ordinal tasks. We extend ReliefF and Simba to the context of ordinal classification and introduce two new algorithms. Some ordinal tasks are collected for testing the proposed algorithms. The experiments show their effectiveness.

The paper is organized as follows. Section 2 gives the basic idea of margin based feature selection algorithms. Section 3 extends ReliefF and Simba to the context of ordinal classification by considering the monotonicity constraints. In Section 4, we present the experimental analysis on some data sets. The conclusions are introduced in Section 5.

2. Large margin feature selection methods

In this section we give a brief description on margin based feature selection.

A classification task is described by a set of features, and an object related to this task is written as $\langle x_i, y_i \rangle$, where x_i is a vector consisting of the feature values of the ith object and y_i is the class label. Given a set of such samples, the task is to learn the classification function from these samples. In practice, as we do not know which information is useful for learning, a lot of irrelevant features are provided. It has been shown that the feature subspace producing a large margin is able to build a good classifier.

There is more than one approach to defining the classification margin. From the viewpoint of the nearest neighbor rule, the margin can be computed as follows.

Definition 1. Let S be the instance space and x be an instance and w be a weight vector over the feature set. Then the classification margin of x is defined as

$$\theta_S^w(x) = \frac{1}{2}(\|x - NM(x)\|_w - \|x - NH(x)\|_w) \tag{1}$$

where $NH(x)$ is the nearest sample point from the same class of x and $NM(x)$ is the nearest sample point from different classes of x. They are called the nearest hit (NH) and the nearest miss (NM), respectively. $\|x - NH(x)\|_w$ and $\|x - NM(x)\|_w$ are the distances between two instances. $\|z\|_w = \sqrt{\sum_i w_i^2 z_i^2}$. The value of $\theta_S^w(x)$ reflects how far sample x would be misclassified if it moves to the region of different classes.

The margin of a single object is also defined as the distance from an instance to the decision border. It is a geometric measure for weighting the feature subset with respect to its decision. The definition of margin is illustrated in Fig. 1.

In [13], the relationship between margin and generalization power was discussed.

Theorem 1. *Let D be a distribution over $R^N \times \{\pm 1\}$ which is supported on a ball of radius R in R^N, N is the size of the feature space. Let $\delta > 0$ and S be a sample of size m. With probability $1 - \delta$ over the random choice of S, for any set of features F and any $\gamma \in (0,1]$:*

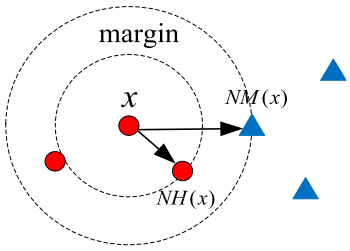

Fig. 1. Illustration for margin.

$$error_D(h) \leqslant \widehat{error}_s^r(h) + \sqrt{\frac{2}{m}\left(d\ln\left(\frac{34em}{d}\right)\log_2(578m) + \ln\left(\frac{8}{\gamma\delta}\right) + (|N|+1)\ln N\right)}$$

where h is the nearest neighbor classification rule when distance is measured only on the features in F and $d = \left(\frac{64R}{r}\right)^{|N|}$.

From Theorem 1 [13], we can obtain that the generalization bound is data dependent. Thus it explains that a good generalization performance can be guaranteed when selecting a small set of features with large margin.

In the rest of this paper, we assume S is the instance space under consideration, I is the number of features, m is the size of instance space S, T is the number of iterations.

In fact, far earlier than 2004, margin was used to evaluate features. In 1992, a feature evaluation technique, called Relief, was proposed, where the margin defined above was introduced [27].

Algorithm 1. Relief

```
initialize the weight vector: w = 0;
for t = 1:T
    randomly select an instance x from S;
    find nearest hit NH(x) and nearest miss NM(x);
    for i = 1:I do
        w_i = w_i − diff (i,x,NH(x))/T + diff (i,x,NM(x))/T;
    end
end
```

Relief (Algorithm 1) is able to deal with both nominal and numerical attributes. For nominal attributes:

$$\mathrm{diff}(i,x,y) = \begin{cases} 0; & \text{if } i(x) = i(y) \\ 1; & \text{otherwise} \end{cases} \tag{2}$$

where $i(x)$ denotes the attribute value of sample x on Feature i. For numerical attributes:

$$\mathrm{diff}(i,x,y) = \frac{|i(x) - i(y)|}{\max(i) - \min(i)} \tag{3}$$

The function diff (i,x,y) calculates the distance between the attribute i for two instances x and y. Under these definitions of distance functions, the value domain of the distance is $[0,1]$. The function diff (i,x,y) calculates the distance between the attribute i for two instances x and y, then the sum of the distances over all attributes is computed as the distance between these two instances. There are different definitions of distance. For example,

$$D(x,y) = \sum_{i=1}^{I} w_i \mathrm{diff}(i,x,y) \tag{4}$$

$$D(x,y) = \left(\sum_{i=1}^{I} \mathrm{diff}(i,x,y)^p\right)^{\frac{1}{p}} \tag{5}$$

Euclidean distance is widely used in practice.

$$D(x,y)=\left(\sum_{i=1}^{I}\mathrm{diff}(i,x,y)^2\right)^{\frac{1}{2}} \tag{6}$$

Each time, an instance x is randomly selected from the instance space and then two nearest neighbors for x are searched: one from the same class (called nearest hit, NH) and the other from the different class (called nearest miss, NM). Relief updates the weight vector depending on the distance of the instance x from NH and NM. If a data set is very large, it is usually time consuming to select the entire instances. We can control the number of iterations T to speed up the computation [41].

Relief is viewed as an efficient and effective feature evaluation method. One major drawback of Relief is that the result may be not exact if noisy and missing data are provided. Some extended algorithms have been developed. We introduce two representative algorithms ReliefF and Simba here.

Similarly to the Relief, ReliefF considers an instance x in each iteration. The main difference between Relief and ReliefF is the number of nearest neighbors used in evaluating features. Instead of selecting the nearest neighbor, ReliefF considers k nearest neighborhoods from the same class and different classes in computing the sample margin for reducing the influence of noise. Experiments conducted by researchers demonstrated that ReliefF can improve the performance [41].

In ReliefF, k is a unique parameter to be set. It is usually high computational complexity if we try all candidate number of the nearest neighbors. The experiments show if the number of neighbors is relatively small, the ReliefF is robust enough. In general parameter k is set to 10 [29]. Hong suggested that $k=\log m$ may also be a good choice in most cases [18].

Algorithm 2. ReliefF

```
initialize the weight vector: w = 0;
for t = 1:T
    randomly select an instance x from S;
    find nearest hit NH_j(x), j = 1,2, ··· ,k;
    for each class C ≠ class(x), (class(x) denote as the class
  label of x), find k nearest misses NM_{j,C}(x), j = 1,2, ··· ,k;
    for i = 1:I do
      w_i = w_i − Σ_{j=1}^{k} diff(i,x,NH_j(x))/(T·k) +
          Σ_{C≠class(x)} [ P(C)/(1−P(class(x))) Σ_{j=1}^{k} diff(i,x,NM_{j,C}(x)) ]/(T·k);
    end
end
```

In ReliefF (Algorithm 2), squared Euclidean distance is employed to compute the distance between two instances. Euclidean distance function can also be taken into account. Robnik performed experiments on some data sets and proved that there is no significant difference in using these two metrics [41]. If the Euclidean distance is used to calculate the difference between two instances, the Relief is extended to Simba.

Simba attempts to search the feature set which maximizes the margin.

Definition 2. Let S be the instance space and w be a weight vector, the evaluation function is defined as:

$$e(w)=\sum_{x\in S}\theta_{S\backslash x}^{w}(x) \tag{7}$$

The evaluation function is the sum of the margin of all the instances, in which the margin of each instance x is computed with all the instances in S excluding x. Simba tries to find the weight vector w that maximizes the evaluation function $e(w)$. It performs a stochastic gradient ascent over the evaluation function e (w) while ignoring the constraint $\|w^2\|_\infty=1$. $e(w)$ is smooth almost everywhere, so the gradient ascent method is used to maximize it. The gradient of $e(w)$ is computed as:

$$(\nabla e(w))_i=\frac{\partial e(w)}{\partial w_i}=\sum_{x\in S}\frac{\partial\theta(x)}{\partial w_i}$$

$$=\frac{1}{2}\sum_{x\in S}\left(\frac{(x_i-NM(x)_i)^2}{\|x-NM(x)\|_w}-\frac{(x_i-NH(x)_i)^2}{\|x-NH(x)\|_w}\right)w_i \tag{8}$$

In each iteration, we only need to calculate $(\nabla e(w))_i$ for each feature and add it to the weight vector w, respectively. By the definition of the evaluation function, for any $a\in R^+$and w, it holds that $e(aw)=|a|e(w)$. Then the constraint is done at the last step.

Algorithm 3. Simba

```
initialize the weight vector: w = (1,1, ··· ,1);
for t = 1:T
    randomly select an instance x from S;
    find nearest hit NH(x) and nearest miss NM(x) from
  S\{x} and calculate the weight vector w;
    for i = 1:I calculate
        Δ_i = 1/2 ( (x_i − NM(x)_i)^2 / ||x − NM(x)||_w − (x_i − NH(x)_i)^2 / ||x − NH(x)||_w ) w_i
    w = w + Δ;
w ← w^2/||w^2||_∞, where (w^2)_i := (w_i)^2.
```

All Relief, ReliefF and Simba take classification margin as the statistic to evaluate features for general classification tasks. Margin cannot reflect the monotonicity constraint in monotonic classification.

3. Feature selection for monotonic classification

In this section, we consider the ordinal classification problem with the monotonicity constraint.

An ordinal decision system is referred as a database $ODS=(S,A,D,V)$, where S is a finite set of instances; A is the set of attributes and $D=\{d_1,d_2,\cdots,d_M\}$ is the set of decisions. V_a is the domain of the attribute a, $V=\bigcup_{a\in A}V_a$. The domain of D is $D=\{d_1,d_2,\cdots,d_M\}$.

In monotonic classification, the domains of the attributes and the decisions have a structure of order. As to decision D, we have $d_1<d_2<\cdots<d_M$. The nested decision structure is given by $d_i^{\geqslant}=\bigcup_{j=i}^{M}d_j$ or $d_i^{\leqslant}=\bigcup_{j=1}^{i}d_j$. For example, consider the problem of customer satisfaction analysis, the decision-maker tries to classify customer preferences with respect to the product quality. If the evaluation on attribute A is better than B, we expect that the comprehensive evaluation on A is no worse than B by customers.

Definition 3. f is said to be a monotone function if for any $x,y\in U$, it satisfies the monotonicity constraint:

$$x\succcurlyeq y\Rightarrow f(x)\succcurlyeq f(y) \tag{9}$$

where $x\succcurlyeq y$ denotes $x_i\geqslant y_i$ for all coordinates x_i and y_i.

Feature selection for monotonic classification is rarely considered in the literature. By combination of large margin and monotonicity constraint, we introduce two large-margin feature selection algorithms for monotonic classification.

Suppose that an monotonic decision system has M ordered classes, labeled from d_1 to d_M, the order between classes can be written as $d_1<d_2<\cdots<d_M$. Given an instance x from d_k, the instances with

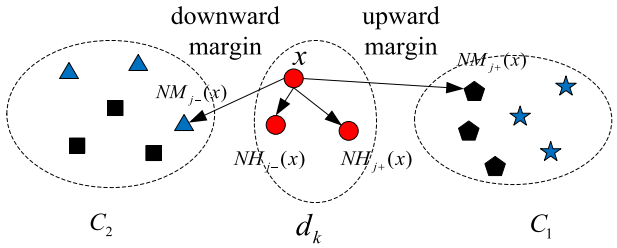

Fig. 2. Illustration the margin for monotonic classification.

decision $d^{\geqslant}_{k+1}$ should be better than x in terms of feature values. In the meanwhile, the instances with decision $d^{\leqslant}_{k-1}$ should be worse than x with respect to attributes. So the monotonic decision system is divided into three subsets. To the instance x, k nearest misses are found from $d^{\geqslant}_{k+1}$ (coined by $NM_{j+}(x)$, $j = 1,2,\cdots,k$) and $d^{\leqslant}_{k-1}$ (denoted by $NM_{j-}(x)$, $j = 1,2,\cdots,k$). Correspondingly, from d_k, we also can find nearest hits which are larger or smaller than x on all features (denoted by $NH_{j+}(x)$ and $NH_{j-}(x)$, $j = 1,2,\cdots,k$). Then we get two margins of instance x according to the dominance relation, called the upward margin and downward margin, respectively. In this process, the sum of upward margin and downward margin compose the margin of an instance x with respect to decision boundary. The idea is illustrated in Fig. 2.

In each iteration, given an instance x, suppose its k nearest hits and nearest misses have been found. In ReliefF, the weight update rule is modified as:

$$w_i = w_i - \sum_{j=1}^{k} diff(i,x,NH_{j-}(x))/(T\cdot k) + \sum_{j=1}^{k} diff(i,x,NM_{j-}(x))/(T\cdot k) - \sum_{j=1}^{k} diff(i,x,NH_{j+}(x))/(T\cdot k) + \sum_{j=1}^{k} diff(i,x,NM_{j+}(x))/(T\cdot k) \quad (10)$$

We extend ReliefF to monotonic classification and name this new algorithm as Ordinal ReliefF, for short we call it O-ReliefF (Algorithm 4).

Algorithm 4. O-ReliefF

```
initialize the weight vector: w = 0;
for t = 1:T
    randomly select an instance x from S;
    find nearest hit NH_{j+}(x), NH_{j-}(x), j = 1,2,...,k from the
  same class to x;
    from C_1 find k nearest misses NM_{j+}(x), j = 1,2,...,k, from
  C_2 find k nearest misses NM_{j-}(x), j = 1,2,...,k;
    for i = 1:I do
      w_i = w_i
  -sum_{j=1}^{k} diff(i,x,NH_{j-}(x))/(T·k) + sum_{j=1}^{k} diff(i,x,NM_{j-}(x))/
  (T·k);
  -sum_{j=1}^{k} diff(i,x,NH_{j+}(x))/(T·k) + sum_{j=1}^{k} diff(i,x,NM_{j+}(x))/
  (T·k)
    end
end
```

We also generalize Simba to monotonic classification based on taking the essence of order relation into account. The evaluation function for monotonic classification represents the margin of all the instances, which contain two parts: upward margin and downward margin. The evaluation function can be rewritten as:

$$e(w) = \sum_{x\in S}\left(\theta^{w}_{(S\backslash x)-}(x) + \theta^{w}_{(S\backslash x)+}(x)\right) \quad (11)$$

where $\theta^{w}_{(S\backslash x)-}(x)$ denotes the downward margin of instance x and $\theta^{w}_{(S\backslash x)+}(x))$ denotes the upward margin of instance x, respectively.

The gradient of evaluation function for monotonic classification is modified as:

$$\begin{aligned}(\nabla e(w))_i &= \frac{\partial e(w)}{\partial w_i} = \sum_{x\in S}\left(\frac{\partial \theta_-(x)}{\partial w_i} + \frac{\partial \theta_+(x)}{\partial w_i}\right)\\ &= \frac{1}{2}\left(\frac{(x_i - NM_-(x)_i)^2}{\|x - NM_-(x)\|_w} - \frac{(x_i - NH_-(x)_i)^2}{\|x - NH_-(x)\|_w}\right)w_i\\ &\quad + \frac{1}{2}\left(\frac{(x_i - NM_+(x)_i)^2}{\|x - NM_+(x)\|_w} - \frac{(x_i - NH_+(x)_i)^2}{\|x - NH_+(x)\|_w}\right)w_i \end{aligned} \quad (12)$$

The weight of ith feature is then updated. We call this algorithm Ordinal Simba (O-Simba, Algorithm 5).

Algorithm 5. O-Simba

```
initialize the weight vector: w = (1,1,...,1);
for t = 1:T
    randomly select an instance x from S;
    find nearest hit NH_+(x) and NH_-(x) from the same class
  to x, find nearest miss NM_+(x) from C_1 and nearest miss
  NM_- from C_2, calculate the weight vector w;
    for i = 1:I calculate
      Δ_i = 1/2 ((x_i - NM_-(x)_i)^2 / ||x - NM_-(x)||_w - (x_i - NH_-(x)_i)^2 / ||x - NH_-(x)||_w) w_i
        + 1/2 ((x_i - NM_+(x)_i)^2 / ||x - NM_+(x)||_w - (x_i - NH_+(x)_i)^2 / ||x - NH_+(x)||_w) w_i
    w = w + Δ;
w ← w^2 / ||w^2||_∞, where (w^2)_i := (w_i)^2.
```

Although the proposed O-ReliefF and O-Simba seem more complicated, the computation complexity of them are O (TmI), just the same as Relief. For an instance x, we have to compute distances between all the instances and x, which takes O (mI) operations. When iterating over the entire instance space, i.e., $T = m$, then the complexity is O (m^2I). The computational complexity shows that the O-ReliefF and O-Simba are efficient algorithms.

4. Experimental analysis

In order to test the proposed algorithms, some ordinal classification tasks are collected. Since Relief is applicable only to binary problems, while ReliefF can deal with multi-class tasks, we compare the performance of the proposed O-ReliefF with ReliefF and O-Simba with Simba, which have been successfully used to feature selection in general classification.

In ReliefF, the number of k nearest neighborhoods need to be set. In this paper, we set $k = 1$ and $k = \log m$. In the iteration, we use all the instances in each data set during evaluation. The experiments on two types of data sets demonstrate the effectiveness of the proposed algorithms. In the first experiment, we generate a synthetic data sets. In the second one, we select ten UCI monotonic tasks for which the monotonicity constraint should not be neglected.

4.1. Experiments on synthetic data

We generate a set of artificial samples of three classes: the first feature has the monotonicity constraint with the decision and the second feature does not. And 50 irrelevant features, which are drawn independently from the uniform distribution on the unit interval. The scatter plot of the data in 2-dimension space of Feature 1 and Feature 2 is shown in Fig. 3.

Here, our objective is to assess the ability of each algorithm to select the relevant features. Since the presence of a large amount irrelevant features, it is reasonable that the first two features have

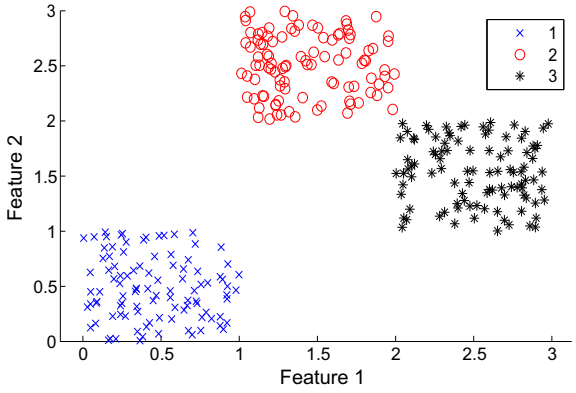

Fig. 3. The class distribution in the 2-dimensionality space.

a larger feature weight. Moreover Feature 1 should produce the largest weight among these features. We try different sizes of data sets, ranging from 10 to 100 samples per class, called Data 1 to Data 10, respectively. The experiments are run 10 times on each data set. The feature weight vector is normalized into [0,1]. Figs. 4 and 5 present the average feature weight based on ReliefF and Simba, respectively, where ReliefF (1) denotes we search only one nearest neighbor in the experiment, while ReliefF (2) denotes $\log m$ nearest neighbors are searched. It is clear that O-ReliefF and O-Simba correctly select Feature 1, which has the monotonicity structure with the decision. The irrelevant features have much smaller weights than ReliefF and Simba on the irrelevant features. This example demonstrates O-ReliefF and O-Simba are able to rank features according to the relevance in monotonic classification problems.

4.2. Experiments on real-world data sets

We selected 10 monotonic data sets from UCI machine learning Repository. The description of the data sets is given in Table 1. We use classification error to evaluate the performance of each feature weighting methods. Ranking tree, proposed by Xia et al., was introduced for validating the selected features [49].

Tables 2 and 3 present the classification error for each data sets using ranking tree. The last column in Table 3 gives the classification error with the original data, where the best methods have been marked with bold. The number of selected features is shown in Table 4.

From the experimental results in Tables 2–4, we can derive that:

1. In most cases, the proposed algorithm O-ReliefF outperforms ReliefF and O-Simba outperforms Simba as to the classification performances, and the selected features by O-ReliefF and O-Simba are less than ReliefF and Simba, respectively.

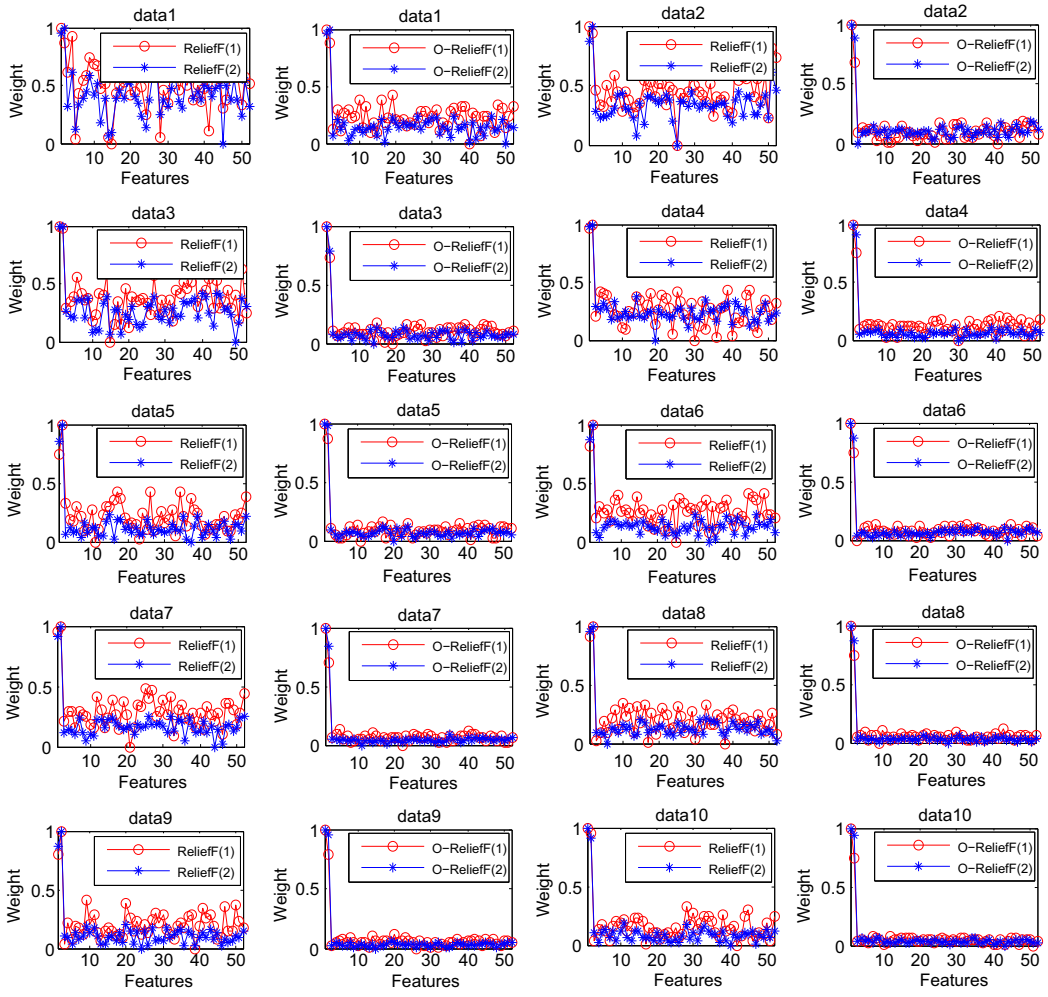

Fig. 4. Comparison of ReliefF and O-ReliefF via feature weight on artificial data sets.

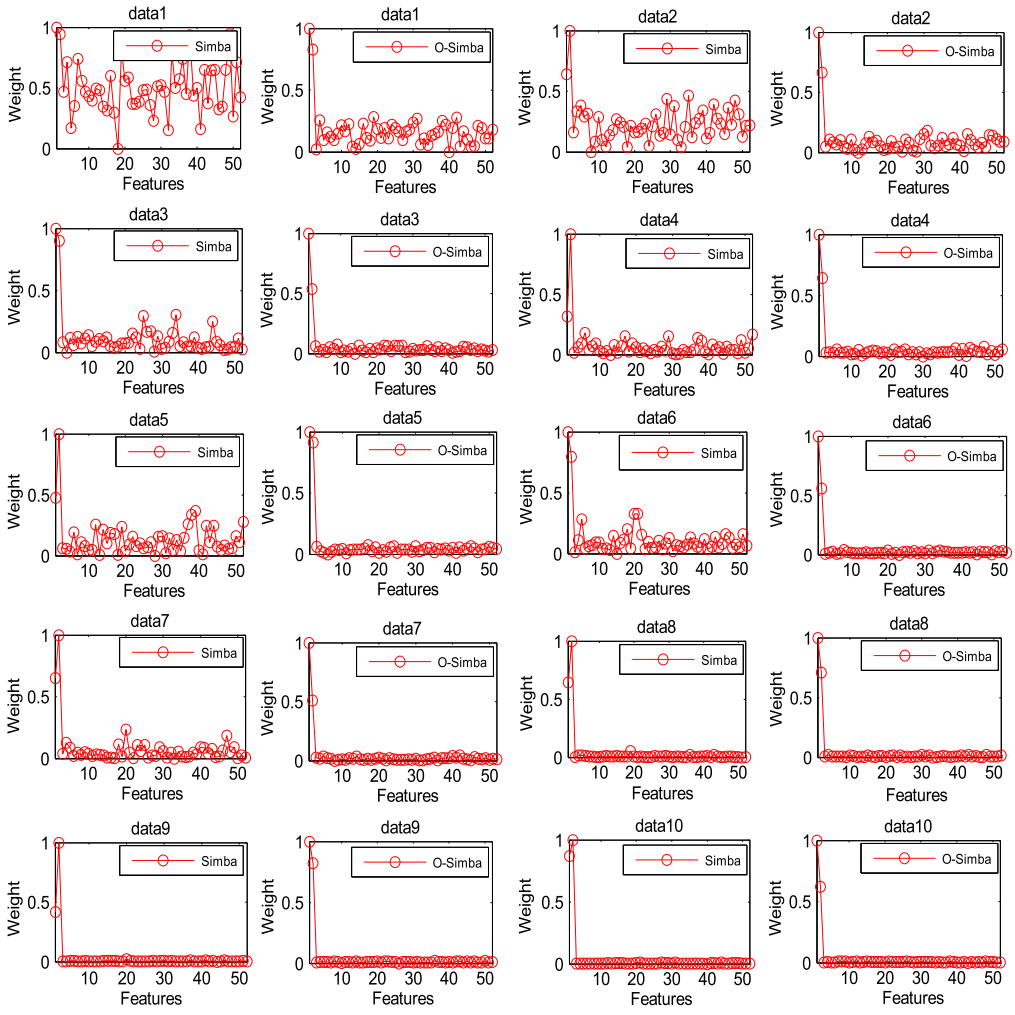

Fig. 5. Comparison of Simba and O-Simba via feature weight on artificial data sets.

Table 1
Description of data sets used in the experiments.

Data set	Instances	Features	Classes
Housing	506	13	4
Pasture	36	22	3
Cpu	209	7	4
Bankruptyrisk	39	12	3
Australian credit	690	15	2
Japan credit	690	14	2
German credit	1000	24	2
Wine quality	1599	11	6
Adult	48842	14	2
Wdbc	569	30	2

Table 2
Classification error (%) based on ranking tree.

Data set	ReliefF (1)	O-ReliefF (1)	ReliefF (2)	O-ReliefF (2)
Housing	35.2 ± 10.9	**34.6 ± 8.5**	35.2 ± 10.9	**33.4 ± 9.7**
Pasture	31.7 ± 17.1	**21.7 ± 11.2**	**25.0 ± 11.8**	26.7 ± 14.9
Cpu	29.2 ± 15.3	29.2 ± 15.3	29.2 ± 15.3	29.2 ± 15.3
Bankruptyrisk	22.5 ± 22.4	**17.5 ± 20.9**	25.0 ± 17.3	**17.5 ± 20.9**
Australian credit	**22.9 ± 2.3**	23.8 ± 2.2	31.0 ± 2.7	**25.7 ± 2.3**
Japan credit	14.5 ± 15.8	14.5 ± 15.8	**14.5 ± 15.8**	16.2 ± 15.0
German credit	28.3 ± 1.7	**28.3 ± 2.4**	29.7 ± 2.4	**27.4 ± 2.4**
Wine quality	43.6 ± 2.2	**43.4 ± 2.0**	43.6 ± 2.2	**43.3 ± 2.0**
Adult	19.8 ± 3.9	**18.2 ± 5.0**	19.80 ± 2.4	**17.8 ± 5.4**
Wdbc	7.9 ± 1.0	**7.0 ± 2.7**	7.7 ± 1.7	**7.0 ± 2.0**
Average	25.6	23.8	26.1	24.4

2. The classification performances are greatly improved compared with the raw data. As a whole, the results show that feature selection improves classification performance.

In monotonic classification, the loss function is usually computed as [1,7,30]:

$$L(y,k) = |y-k|^p, p \geqslant 0 \tag{13}$$

where k and y are the predicted class and the true class, respectively. In the rest of the paper, we take the absolute error loss function $L(y,k) = |y-k|$. Then the average rank loss for Ranking Tree algorithm can be written as $\frac{1}{T}\sum_{t=1}^{T}|y_t - k_t|$, where T is the number of the instances, y_t is the true label of instance t, k_t is the predicted label. We estimate the performance of algorithms with 5-fold cross validation. The mean absolute error and standard deviation are given in Tables 5 and 6. And the last column in Table 6 shows the rank loss with the raw data.

We can see from Table 4 that our new algorithm O-ReliefF performs better than ReliefF, O-Simba performs better than Simba in most of monotonic data sets. For further comparison of each algorithm, we show the curves of the classification error varying with the selected feature subset on four data sets in Figs. 6 and 7. It is

Table 3
Classification error (%) based on ranking tree.

Data set	Simba	O-Simba	Original data
Housing	34.4 ± 9.4	**34.0 ± 9.4**	35.8 ± 10.5
Pasture	26.7 ± 14.9	**21.7 ± 7.5**	40.0 ± 19.0
Cpu	29.2 ± 15.3	29.2 ± 15.3	32.0 ± 10.1
Bankruptyrisk	25.0 ± 17.7	**20.0 ± 19.0**	41.8 ± 19.6
Australian credit	41.0 ± 3.4	**34.6 ± 4.3**	44.5 ± 0.7
Japan credit	14.5 ± 15.8	14.5 ± 15.8	17.7 ± 13.9
German credit	27.9 ± 2.8	**27.4 ± 2.0**	29.1 ± 1.8
Wine quality	**43.5 ± 2.2**	43.8 ± 2.6	43.6 ± 2.2
Adult	19.0 ± 4.3	**17.8 ± 5.4**	20.8 ± 4.6
Wdbc	6.5 ± 2.7	**6.1 ± 1.3**	8.1 ± 1.3
Average	26.8	24.9	31.3

Table 5
Rank loss (%) based on ranking tree.

Data set	ReliefF (1)	O-ReliefF (1)	ReliefF (2)	O-ReliefF (2)
Housing	39.7 ± 12.1	**38.1 ± 10.3**	39.3 ± 12.2	**37.4 ± 12.3**
Pasture	25.0 ± 11.8	**23.3 ± 9.1**	**25.0 ± 11.8**	38.3 ± 24.7
Cpu	31.6 ± 15.9	**31.5 ± 17.1**	31.6 ± 15.9	31.6 ± 15.9
Bankruptyrisk	27.5 ± 31.1	**22.5 ± 31.1**	30.0 ± 27.4	**22.5 ± 31.1**
Australian credit	**22.9 ± 2.3**	23.8 ± 2.2	31.0 ± 2.7	**25.7 ± 2.3**
Japan credit	14.5 ± 15.8	14.5 ± 15.8	**14.5 ± 15.8**	16.2 ± 15.0
German credit	28.3 ± 1.7	**28.3 ± 2.4**	29.7 ± 2.4	**27.4 ± 2.4**
Wine quality	47.0 ± 2.1	47.0 ± 2.1	47.0 ± 2.1	47.0 ± 2.1
Adult	19.8 ± 3.9	**18.2 ± 5.0**	19.80 ± 2.4	**17.8 ± 5.4**
Wdbc	7.9 ± 1.0	**7.0 ± 2.7**	7.7 ± 1.7	**7.0 ± 2.0**

easily noted that the new algorithms achieve the best performances firstly and have higher accuracy than the other algorithms.

For comparing the performance of different algorithm in dealing with a lot of irrelevant features, we add 50 independently random features into the bankruptyrisk data set and German credit data set. There are 12 features and 24 features in the raw data set, respectively. We expect the relevant features can get larger weight, while the random features obtain a small weight. For comparison, ReliefF is also performed, where we set $k = \log m$. Each feature weight vector is normalize into interval $[0,1]$. The results are given in Figs. 10 and 11.

It can be seen from Figs. 8 and 9 that O-ReliefF and O-Simba can identify the relevant features. The weights of irrelevant features are significantly smaller than those of the relevant features. Both ReliefF and Simba are fail to identify the relevant features. Some irrelevant features also get a larger feature weight than the relevant features.

To evaluate the robustness of the proposed algorithms, we add 10 percent class noise to the bankruptyrisk data set and German credit data set with irrelevant features. The feature weights are given in Figs. 10 and 11.

From this experiment, we can arrive that with 10 percent class noise, O-ReliefF and O-Simba can also identify which feature is relevant and which not. The above experiments demonstrate that the new feature selection algorithms are effective for monotonic classification problems.

To evaluate classification performance, we use the traditional *F*-measure, which combines recall and precision, to reflect the performance of the selected features via C4.5 based on 5-fold cross validation.

The *F*-measure for each data sets and the number of selected features are shown in Tables 7 and 8. The last column in Table 8 presents the *F*-measure of the raw data. The results show that the proposed algorithm O-ReliefF and O-Simba get better F score, which shows a higher classification accuracy than other algorithms.

Table 6
Rank loss (%) based on ranking tree.

Data set	Simba	O-Simba	Original data
Housing	38.1 ± 11.4	**38.1 ± 10.3**	40.5 ± 12.6
Pasture	26.7 ± 14.9	**21.7 ± 7.45**	43.3 ± 19.0
Cpu	31.6 ± 15.9	31.6 ± 15.9	33.5 ± 11.9
Bankruptyrisk	30.0 ± 27.4	**25.0 ± 29.3**	46.8 ± 24.7
Australian credit	41.0 ± 3.4	**34.6 ± 4.3**	44.5 ± 0.7
Japan credit	14.5 ± 15.8	14.5 ± 15.8	17.7 ± 13.9
German credit	27.9 ± 2.8	**27.4 ± 2.0**	29.1 ± 1.8
Wine quality	**47.0 ± 2.1**	47.7 ± 3.3	47.0 ± 2.1
Adult	19.0 ± 4.3	**17.8 ± 5.4**	20.8 ± 4.6
Wdbc	6.5 ± 2.7	**6.1 ± 1.3**	8.1 ± 1.3

5. Conclusions

Ordinal classification with monotonicity constraint is an important task in applications. It is necessary to consider the order relation between attributes and decision in these tasks. In this paper, we proposed two feature selection algorithms for monotonic classification based on the large-margin principals, denoted by O-ReliefF and O-Simba.

We conducted a set of experiments on synthetic data sets and real-world tasks. By incorporating the order relation, we have experimentally shown that O-ReliefF performs better than ReliefF, O-Simba performs better than Simba with respect to classification error and ranking loss, also F score. These two algorithms have proved to be effective for monotonic classification tasks. We also show that the proposed O-ReliefF and O-Simba are robust against class noise.

Table 4
Number of the selected features.

Data set	ReliefF (1)	O-ReliefF (1)	ReliefF (2)	O-ReliefF (2)	Simba	O-Simba
Housing	10	4	10	4	5	6
Pasture	17	20	8	18	14	4
Cpu	2	2	2	2	2	2
Bankruptyrisk	4	2	5	2	5	3
Australian credit	6	6	6	5	3	2
Japan credit	1	1	5	8	2	3
German credit	18	12	19	12	11	6
Wine quality	11	10	11	10	11	11
Adult	12	9	3	8	10	7
Wdbc	25	9	29	4	4	5
Average	10.6	7.5	9.8	7.3	6.7	4.9

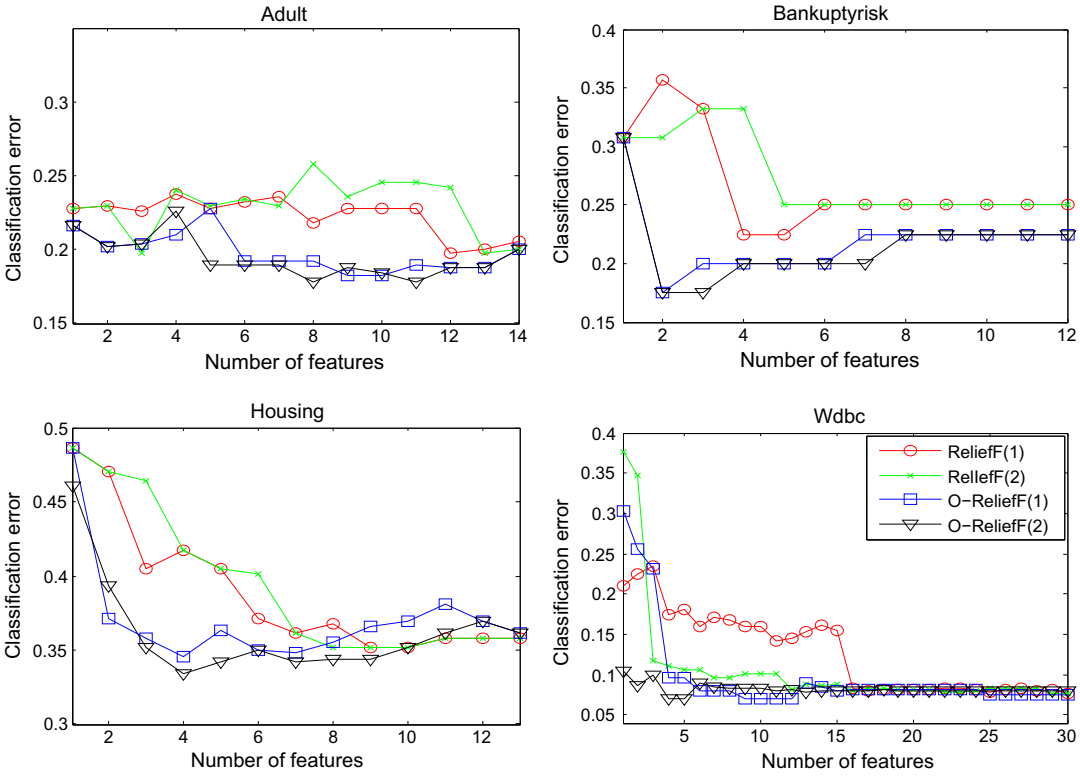

Fig. 6. Comparison of ReliefF and O-ReliefF using classification error.

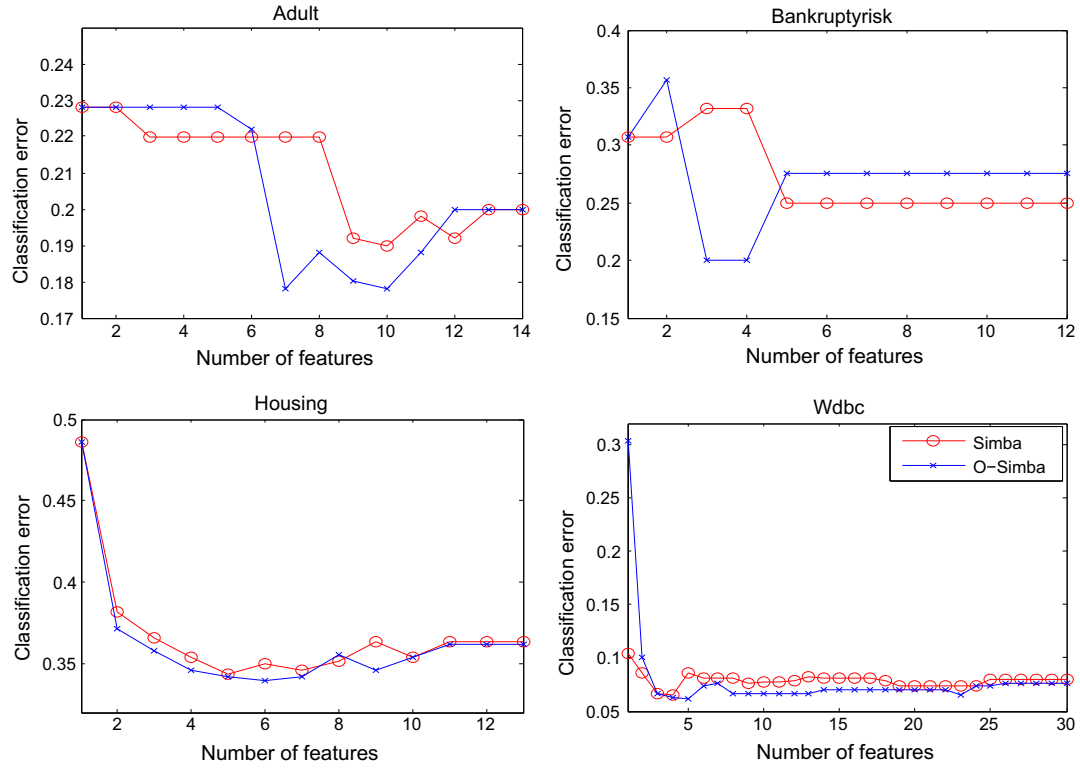

Fig. 7. Comparison of Simba and O-Simba using classification error.

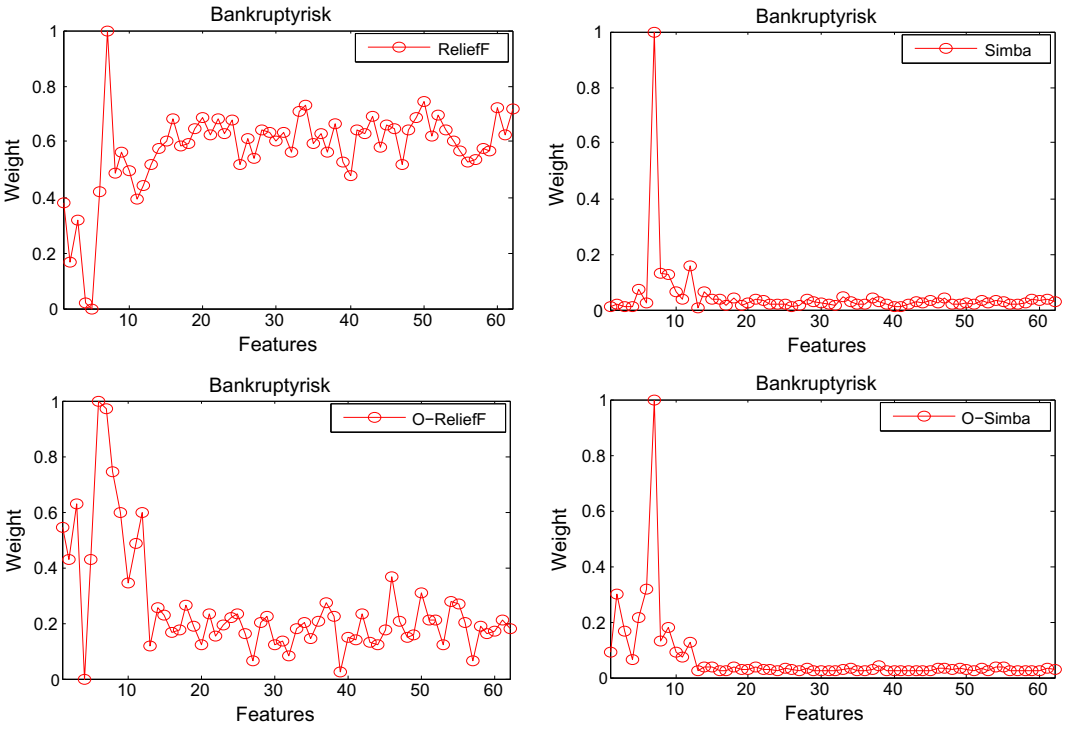

Fig. 8. Feature weights on bankruptcy risk data set (add 50 irrelevant features).

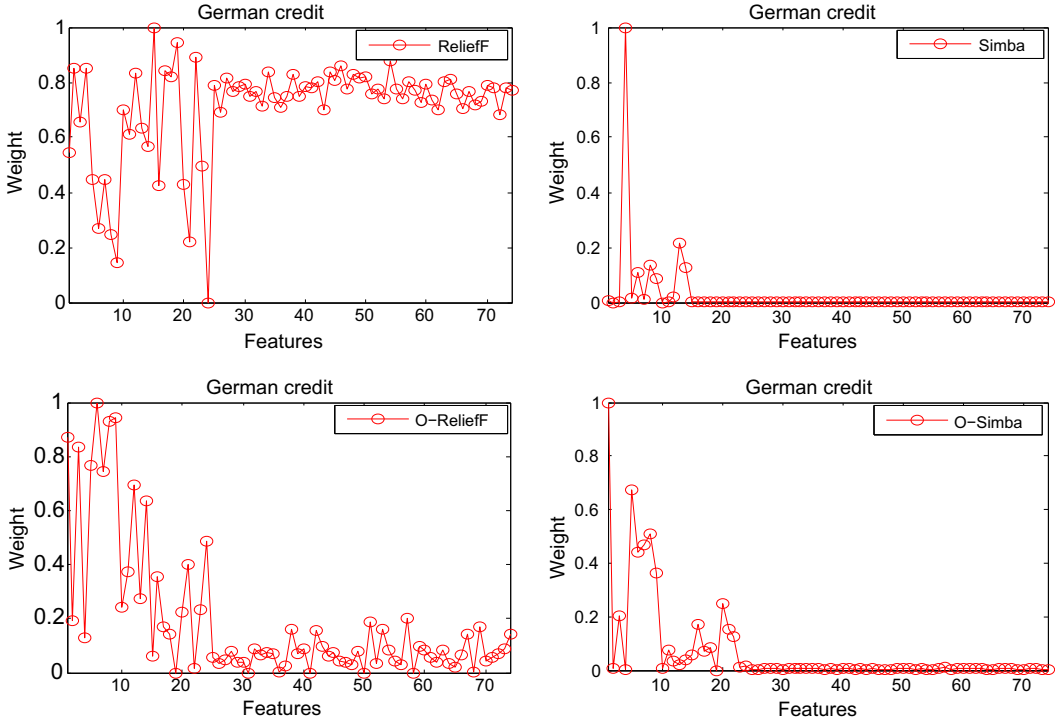

Fig. 9. Feature weights on German credit data set (add 50 irrelevant features).

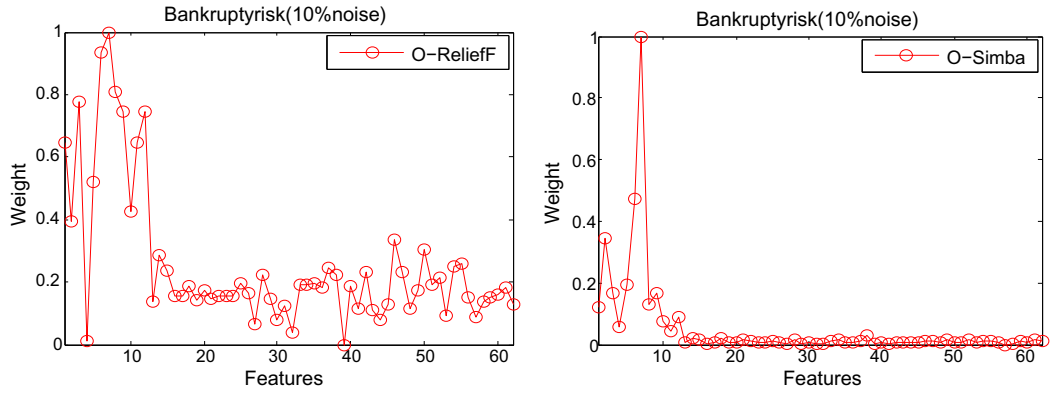

Fig. 10. Feature weights on bankruptcy risk data set (10% class noise).

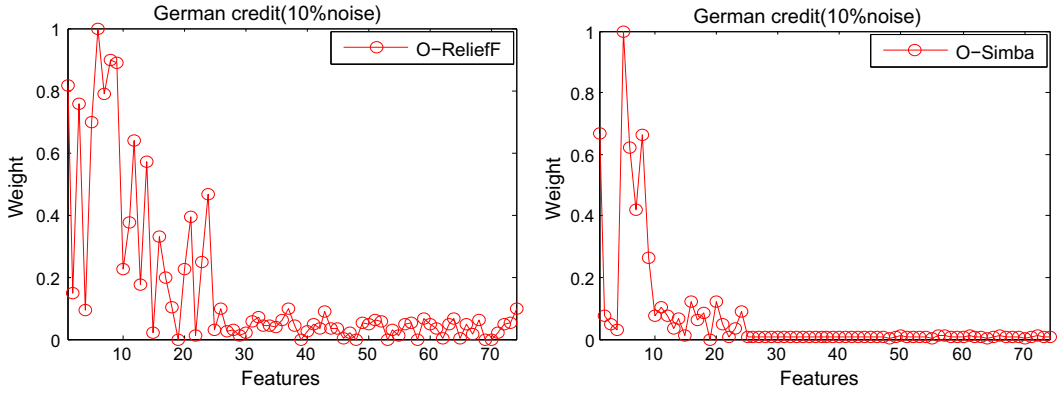

Fig. 11. Feature weights on German credit data set (10% class noise).

Table 7
Comparison of ReliefF and O-ReliefF with *F*-measure.

Data set	ReliefF (1)	O-ReliefF (1)	ReliefF (2)	O-ReliefF (2)
Housing	0.657 (6)	**0.662 (11)**	0.663 (11)	**0.669 (4)**
Pasture	0.694 (18)	0.694 (21)	0.694 (16)	**0.796 (20)**
Cpu	0.890 (6)	**0.923 (3)**	0.923 (3)	0.923 (3)
Bankruptyrisk	0.854 (2)	**0.892 (11)**	0.892 (6)	**0.892 (9)**
Australian credit	0.849 (15)	0.856 (14)	0.855 (14)	**0.855 (9)**
Japan credit	0.858 (15)	**0.859 (12)**	0.858 (15)	**0.858 (13)**
German credit	0.714 (17)	**0.718 (15)**	0.716 (18)	**0.722 (12)**
Wine quality	**0.597 (11)**	0.583 (11)	0.582 (11)	**0.589 (8)**
Adult	0.814 (14)	**0.832 (10)**	0.800 (4)	**0.825 (13)**
Wdbc	0.935 (27)	**0.947 (6)**	0.927 (11)	**0.951 (10)**

Table 8
Comparison of Simba and O-Simba with *F*-measure.

Data set	Simba (1)	O-Simba (1)	Original data
Housing	0.663 (11)	**0.670 (9)**	0.647
Pasture	0.767 (6)	**0.796 (4)**	0.668
Cpu	0.885 (7)	**0.933 (4)**	0.894
Bankruptyrisk	0.888 (7)	**0.892 (8)**	0.892
Australian credit	0.853 (3)	**0.855 (3)**	0.850
Japan credit	0.858 (15)	**0.861 (6)**	0.858
German credit	0.717 (14)	**0.723 (15)**	0.716
Wine quality	0.593 (10)	0.593 (10)	0.582
Adult	0.801 (14)	**0.825 (13)**	0.801
Wdbc	**0.952 (5)**	0.951 (5)	0.931

Acknowledgments

This work is partly supported by National Natural Science Foundation of China under Grants 60703013 and 10978011 and the National Basic Research Program of China (973 Program) under Grant 2012CB215201. Prof. Yu is supported by National Science Fund for Distinguished Young Scholars under Grant 50925625.

References

[1] N. Barile, A. Feelders, Nonparametric monotone classification with MOCA, in: Eighth IEEE International Conference on Data Mining, 2008, ICDM'08, pp. 731–736.
[2] A. Ben-David, Monotonicity maintenance in information-theoretic machine learning algorithms, Machine Learning 19 (1995) 29–43.
[3] A. Ben-David, L. Sterling, Y.H. Pao, Learning and classification of monotonic ordinal concepts, Computational Intelligence 5 (1989) 45–49.
[4] J. Cardoso, J. da Costa, Learning to classify ordinal data: the data replication method, Journal of Machine Learning Research 8 (2007) 1393–1429.
[5] Y. Chen, D. Miao, R. Wang, K. Wu, A rough set approach to feature selection based on power set tree, Knowledge-Based Systems 24 (2011) 275–281.
[6] A. Dalaka, B. Kompare, M. Robnik-ikonja, S. Sgardelis, Modelling the effects of environmental conditions on apparent photosynthesis of Stipa bromides by machine learning tools, Ecological Modelling 129 (2000) 245–257.
[7] K. Dembczyński, W. Kotłowski, R. Słowiński, Ensemble of decision rules for ordinal classification with monotonicity constraints, Rough Sets and Knowledge Technology 5009 (2008) 260–267.
[8] T. Ditterrich, Machine learning research: four current direction, Artificial Intelligence Magzine 4 (1997) 97–136.
[9] M. Doumpos, F. Pasiouras, Developing and testing models for replicating credit ratings: a multicriteria approach, Computational Economics 25 (2005) 327–341.
[10] E. Frank, M. Hall, A simple approach to ordinal classification, Machine Learning: ECML 2001 (2001) 145–156.
[11] Y. Freund, R. Iyer, R. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, The Journal of Machine Learning Research 4 (2003) 933–969.

[12] D. Genest, M. Chein, A content-search information retrieval process based on conceptual graphs, Knowledge and Information Systems 8 (2005) 292–309.
[13] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection-theory and algorithms, in: Proceedings of the 21th International Conference on Machine Learning, ACM, pp. 43–50.
[14] S. Greco, B. Matarazzo, R. Slowinski, A new rough set approach to evaluation of bankruptcy risk, Operational Tools in the Management of Financial Risks (1998) 121–136.
[15] S. Greco, B. Matarazzo, R. Slowinski, Rough approximation by dominance relations, International Journal of Intelligent Systems 17 (2002) 153–171.
[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, The Journal of Machine Learning Research 3 (2003) 1157–1182.
[17] Q. He, C. Wu, D. Chen, S. Zhao, Fuzzy rough set based attribute reduction for information systems with fuzzy decisions, Knowledge-Based Systems 24 (2011) 689–696.
[18] S. Hong, Use of contextual information for feature ranking and discretization, IEEE Transactions on Knowledge and Data Engineering 9 (1997) 718–730.
[19] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, D. Yu, Rank entropy based decision trees for monotonic classification, IEEE Transactions on Knowledge and Data Engineering, in press.
[20] Q. Hu, M. Guo, D. Yu, J. Liu, Information entropy for ordinal classification, Science China Information Sciences 53 (2010) 1188–1200.
[21] Q. Hu, W. Pan, L. Zhang, D. Zhang, Y. Song, M. Guo, D. Yu, Feature selection for monotonic classification, IEEE Transactions on Fuzzy Systems 20 (2012) 69–81.
[22] Q. Hu, D. Yu, M. Guo, Fuzzy preference based rough sets, Information Sciences 180 (2010) 2003–2022.
[23] B. Huang, H. Li, D. Wei, Dominance-based rough set model in intuitionistic fuzzy information systems, Knowledge-Based Systems 28 (2012) 115–123.
[24] Y. Huang, P.J. McCullagh, N.D. Black, An optimization of ReliefF for classification in large datasets, Data & Knowledge Engineering 68 (2009) 1348–1356.
[25] A. Jain, J. Mao, Artificial neural network for nonlinear projection of multivariate data, in: International Joint Conference on Neural Networks, IJCNN, 1992, pp. 335–340.
[26] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 153–158.
[27] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of the Ninth International Workshop on Machine Learning, Morgan Kaufmann Publishers Inc., pp. 249–256.
[28] K. Kira, L. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Proceedings of the National Conference on Artificial Intelligence, John Wiley & Sons Ltd., pp. 129–129.
[29] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: Machine Learning: ECML-94, Springer, pp. 171–182.
[30] W. Kotłowski, R. Słowiński, Rule learning with monotonicity constraints, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp. 537–544.
[31] N. Kwak, C. Choi, Input feature selection for classification problems, IEEE Transactions on Neural Networks 13 (2002) 143–159.
[32] C. Lee, G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Information Processing & Management 42 (2006) 155–165.
[33] Y. Li, B. Lu, Feature selection based on loss-margin of nearest neighbor classification, Pattern Recognition 42 (2009) 1914–1921.
[34] P. Mitra, C. Murthy, S. Pal, Unsupervised feature selection using feature similarity, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 301–312.
[35] M. Modrzejewski, Feature selection using rough sets theory, in: Proceedings of the European Conference on Machine Learning, Springer-Verlag, pp. 213–226.
[36] D. Muni, N. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36 (2006) 106–117.
[37] R. Potharst, J. Bioch, Decision trees for ordinal classification, Intelligent Data Analysis 4 (2000) 97–111.
[38] R. Potharst, A.J. Feelders, Classification trees for problems with monotonicity constraints, SIGKDD Explorations Newsletter 4 (2002) 1–10.
[39] Y. Qian, C. Dang, J. Liang, D. Tang, Set-valued ordered information systems, Information Sciences 179 (2009) 2809–2832.
[40] Y. Qian, J. Liang, P. Song, C. Dang, On dominance relations in disjunctive set-valued ordered information systems, International Journal of Information Technology & Decision Making 9 (2010) 9–33.
[41] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Machine Learning 53 (2003) 23–69.
[42] S. Saha, P. Mitra, S. Sarkar, A comparative study on feature reduction approaches in hindi and bengali named entity recognition, Knowledge-Based Systems (2011).
[43] S. Senthamarai Kannan, N. Ramaraj, A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm, Knowledge-Based Systems 23 (2010) 580–585.
[44] P. Song, J. Liang, Y. Qian, A two-grade approach to ranking interval data, Knowledge-Based Systems 27 (2012) 234–244.
[45] Y. Sun, Iterative RELIEF for feature weighting: algorithms, theories, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1035–1051.
[46] Y. Sun, S. Todorovic, S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1610–1626.
[47] R. Susmaga, R. Slowinski, S. Greco, B. Matarazzo, Generation of reducts and rules in multi-attribute and multi-criteria classification, Control and Cybernetics 29 (2000) 969–988.
[48] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik, Feature selection for SVMs, Advances in Neural Information Processing Systems (2001) 668–674.
[49] F. Xia, W. Zhang, F. Li, Y. Yang, Ranking with decision tree, Knowledge and Information Systems 17 (2008) 381–395.
[50] W. Xu, X. Zhang, J. Zhong, W. Zhang, Attribute reduction in ordered information systems based on evidence theory, Knowledge and Information Systems 25 (2010) 169–184.