Knowl Inf Syst (2014) 39:409–433
DOI 10.1007/s10115-013-0613-2

REGULAR PAPER

An efficient orientation distance–based discriminative feature extraction method for multi-classification

Bo Liu · Yanshan Xiao · Philip S. Yu · Zhifeng Hao · Longbing Cao

Received: 1 May 2012 / Revised: 21 January 2013 / Accepted: 17 February 2013 /
Published online: 14 March 2013

Abstract Feature extraction is an important step before actual learning. Although many feature extraction methods have been proposed for clustering, classification and regression, very limited work has been done on multi-class classification problems. This paper proposes a novel feature extraction method, called *orientation distance–based discriminative (ODD) feature* extraction, particularly designed for multi-class classification problems. Our proposed method works in two steps. In the first step, we extend the Fisher Discriminant idea to determine an appropriate kernel function and map the input data with all classes into a feature space where the classes of the data are well separated. In the second step, we put forward two variants of ODD features, i.e., one-vs-all-based ODD and one-vs-one-based ODD features. We first construct hyper-plane (SVM) based on one-vs-all scheme or one-vs-one scheme in the feature space; we then extract one-vs-all-based or one-vs-one-based ODD features between a sample and each hyper-plane. These newly extracted ODD features are

B. Liu
Faculty of Automation, Guangdong University of Technology,
Guangzhou, China
e-mail: csbliu@gmail.com

B. Liu · P. S. Yu
Department of Computer Science, University of Illinois at Chicago,
Chicago, IL, USA
e-mail: psyu@uic.edu

Y. Xiao (✉) · Z. Hao
Faculty of Computer Science, Guangdong University of Technology, Guangzhou, China
e-mail: xiaoyanshan@gmail.com; mazfhao@scut.edu.cn

P. S. Yu
Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia

L. Cao
Faculty of Engineering and Information Technology, University of Technology,
Sydney, Australia
e-mail: lbcao@it.uts.edu.au

treated as the representative features and are thereafter used in the subsequent classification phase. Extensive experiments have been conducted to investigate the performance of one-vs-all-based and one-vs-one-based ODD features for multi-class classification. The statistical results show that the classification accuracy based on ODD features outperforms that of the state-of-the-art feature extraction methods.

Keywords Multi-class classification · Feature extraction · Support vector machine · One-against-all scheme · One-against-one scheme

1 Introduction

Feature extraction [8] plays an important role in exploring data by mapping the input data onto a space which reflects the inherent structure of the original data. In the mapped space, distinctive features are extracted from source data to represent the source data. The extracted features are therefore utilized for the subsequent learning phase instead of the source data. In general, feature extraction is always considered as the pre-processing step which offers distinctive features for the following learning. To date, feature extraction has been found in a variety of application domains, ranging from pattern recognition to internet applications, medical applications, visualization time series analysis, etc. [15,26,35,38,42].

Depending on the principle of the optimization models, the previous feature extraction methods can be classified into two broad categories: (1) unsupervised feature extraction [2,4, 22,31,34], in which the representative features are constructed out of the original data without taking the labels of the data into account. For example, kernel principal component analysis (KPCA) first maps the source data into a feature space via a kernel function and extracts the distinctive features by calculating the minimal mean-square error between the representation features and the source data. Since the data label is not considered in the learning, it is not easy to guarantee that the data belonging to different categories are well separated in the mapped feature space, consequently, the extracted features may reduce the performance of the subsequent learning. (2) supervised feature extraction methods [7,17,36,39]: they map the input data onto an ideal sub-space, in which the samples from different classes are considerably separated. In the process, the data labels are considered in the determination of the sub-space to let the samples from different categories to be well separated. In general, the supervised feature extraction method always outperforms unsupervised feature extraction methods.

Despite much progress made in this area, most existing methods of feature extraction are not particularly designed for multi-class classification and may have problem in extracting distinctive features for multi-class classification. For example, in the unsupervised feature extraction methods, the label information is not incorporated into the learning procedure; consequently, data from the different classes may share similar extracted features. This potentially reduces the accuracy of multi-class classification. In the supervised feature extraction methods, although the data labels are considered in the learning procedure, it is still difficult to choose an ideal subspace in which the projections of different classes are distinctive. For example, in the feature space of kernel Fisher discriminant analysis (KFDA) [17,30], KFDA determines a canonical direction for which the data are most separated when it is projected on a line in this direction. However, when the number of classes is large, it is not easy to determine such a canonical direction so that the projections of data are well separated, even in the case that the classes are well separable in the feature space. As illustrated in Fig. 1, the projections of the three classes on the direction 1, 2, 3 or 4 are all not separated; even if

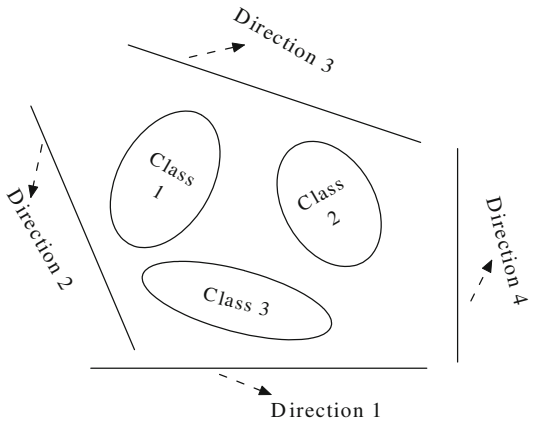

Fig. 1 Possible directions of the projection from higher dimensions to lower dimensions of kernel Fisher discriminant analysis

the three classes are separated themselves in the space, yet there does not exist a direction where the projections of the classes are well separable. This always reduces the performance of the supervised feature extraction methods for multi-class classification problems. Another important observation is that many real-world classification applications fall into the category of multi-class classification problems such as handwriting and image recognition. Therefore, it is worthwhile to explore new feature extraction methods specifically for the multi-class classification problems so that the data represented by the extracted features share less similarity between the different categories. As a result, the extracted features can contribute maximally to classification performance.

This paper proposes a novel supervised feature extraction method, called orientation distance–based discriminative (ODD) feature extraction, for multi-class classification problems. Our proposed method works in two steps. In the first step, we map the source data into a feature space via a kernel function where the data from the different classes are separable. In the second step, we extract ODD features between each class. The main contribution of this paper can be summarized as follows.

1. In the first step, we map the source data into a feature space via a kernel function. In order to guarantee the separability of the classes in the feature space, we propose an extension of the Fisher discriminant method to determine a proper kernel function [22,30] by maximizing the ratio of inter-class similarity and within-class similarity so that the data from the same class share high similarity, while the samples belonging to the different classes have less similarity. By doing this, we guarantee the data from the different classes distinguishable in the feature space.
2. In the second step, we propose two variants of ODD feature extraction method based on one-vs-all scheme and one-vs-one schemem, respectively. We first construct hyper-planes (SVMs) [6] based on the one-vs-all or one-vs-one scheme and then extract orientation features between each sample and each hyper-planes. We then extract one-vs-all-based ODD features and one-vs-one-based ODD features. To the best of our knowledge, this is the first work to extract distinctive features on top of the orientation distance between each sample and each hyper-plane for multi-class classification problem.
3. Finally, we conduct an extensive experiment on real-world UCI datasets to compare the proposed one-vs-all-based ODD features and one-vs-one-based ODD features with kernel principal component analysis (KPCA), kernel Fisher discriminant features (KKFD) and margin maximizing discriminant analysis (MMDA), using three

multi-class classification methods, i.e., decision tree, neural network and SVM. The statistical results show that the one-vs-all-based and one-vs-one-based ODD features outperform those extracted by other methods.

The rest of the paper is organized as follows: Sect. 2 presents the previous work related to our study. Section 3 puts forward the proposed ODD feature extraction method in detail. Extensive experiments are conducted in Sect. 4. Section 5 concludes the paper and presents the possible work in the future.

2 Related work

In this section, we briefly review the previous feature extraction and feature selection methods in this section.

2.1 Feature selection and extraction

Feature selection [44,45] and feature extraction [43,46] have been widely used in machine learning and various application domains, though they attempt to reduce the redundant information from the data and improve the performance of the subsequent learning; however, the principles of the two techniques are based on the different objectives. Feature selection always attempts to select a set of features from the input features, and the selected features are utilized in the subsequence learning; while feature extraction constructs the combinations of the source input features and uses the combined features in the following learning target. We briefly introduce them as follows.

2.1.1 Feature selection

Feature selection [16,19,20,24,40], also known as variable subset selection, is the technique of selecting a subset of relevant features from the original source features before doing the actual learning. Feature selection methods [9,25,32] are always required to find a global NP-hard optimization problem, where greedy approaches—forward selection and backward elimination—are often used to tackle the optimization problem directly. In addition, SVM has been used to generate memberships which are therefore incorporated into feature selection learning [18], and neighborhood-based feature selection has been proposed for SVM [10].

2.1.2 Feature extraction

Feature extraction aims to construct combinations of the source variables while still describing the data with sufficient accuracy. The previous work can be broadly classified into unsupervised feature extraction and supervised feature extraction.

In the unsupervised feature extraction methods, the data labels are not considered in the learning procedure. PCA, KPCA [4,22,33,37], ICA and KICA [2,3,14,34] are well-known feature extraction algorithms. PCA represents the input patterns in a lower-dimensional subspace such that the expected squared reconstruction error is minimized. The transform of PCA is derived from eigenvectors corresponding to the largest eigenvalues of the covariance matrix for data of all classes. PCA seeks to optimally represent the data in terms of minimal mean-square error between the representation and the original data. In recent years, some variances of PCA have been proposed. For instance, Weng et al. [31] propose an incremental

principal component analysis to realize online learning for PCA. Tang et al. [27] use traditional PCA and the non-orthogonal binary feature extraction method to obtain components. In addition, kernel methods have recently been provided to implement PCA in a nonlinear fashion in the form of kernel-PCA [22,33]. ICA has also been proposed as a tool to find interesting projections of the data. The works in [2,3] maximize negentropy (divergence to a Gaussian density function) to find a subspace on which the data has the least Gaussian projection. The criterion corresponds to finding a projection of data that looks maximally clustered. This appears to be a very useful tool for revealing non-Gaussian structures in the data. KICA [14,34] is an extension of ICA in the kernel feature space. In general, KICA has been proposed for nonlinear separable data by kernel function. The limitation of these methods in this category is that they are completely unsupervised methods without taking the data labels into learning; consequently, they lack the ability to enhance class separability, and the extracted features may not be appropriate for classification purpose [13].

In the supervised feature extraction, the data labels are used to supervise the feature extraction procedure. In this category, Fisher discriminant analysis (FDA), also called Linear discriminant analysis (LDA), and its variants are typical [7,13,36,39,41]. FDA searches for directions that allow optimal discrimination between the classes provided that the input patterns are normally distributed for all classes and share the same covariance matrix. Unlike PCA which is not concerned with the class information, FDA takes much consideration of the label information of the data. Moreover, kernel-based LDA (KDA) or kernel-based FDA (KFDA) has been proposed to offer a flexible ability to handle the cases where data is nonlinearly separable in the input space. In such case, the data is then explored in the kernel-induced feature space to find an optimal direction along which the separability of different classes is maximized. In addition, margin maximizing discriminant analysis (MMDA) [13,28], considered as nonparametric extension of LDA, projects input patterns onto the subspace spanned by the normals of a set of pairwise orthogonal margin maximizing hyperplanes which is determined by support vector machine. For this category, it is generally believed that they improve the ability to enhance class separability compared to unsurprised feature extraction methods. However, when the number of classes for multi-class classification problems is large, it is not easy to determine optimal direction where the projections of data are well separated, even if the classes are well separable in feature space. As illustrated in Fig. 1, the projections of the three classes on the direction 1, 2, 3 or 4 are all not separated; even if the three classes are separated themselves in the space, yet there does not exist an direction where the projections of the three classes are well separable. This always reduces the performance of the supervised feature extraction methods for multi-class classification problems.

Our proposed ODD feature extraction method is proposed to extract features for multi-class classification problems. The same as FDA, KFDA and their variants, our method is a supervised feature extraction method since the label information is fully utilized in the feature extraction procedure. The difference from KFDA is that, KFDA determines an optimal direction along which the separability of different classes is maximized in the feature space, whereas our ODD method directly constructs optimal hyper-planes (SVMs) in the feature space via one-vs-all and one-vs-one schemes. This strategy potentially guarantees the quality of ODD features compared to KFDA features because even if the classes are well separable in the feature space after kernel mapping, there may not exist an optimal direction where the projections of data are well separated, as illustrated in Fig. 1. MMDA [13,28], considered as nonparametric extension of FDA, has similar characteristic as FDA that, even if the classes are well separable in space, there may not exist an optimal direction where the projections of data are well separated.

In addition, our proposed ODD method falls into the feature extraction category, since ODD features are not the subset of the source features, but the extracted features between each instance and each constructed hyper-plane. In the experiments, we will explicitly compare the ability of ODD features with other kernel-based feature extraction methods such as KPCA, KFDA and MMDA, since kernel-based methods outperforms the ones in input space.

3 Our proposed algorithm

Feature extraction aims at constructing the combinations of source variables. Due to the fact that most real-world classification applications fall into the category of multi-class classification and previous works have difficulty in extracting distinctive features for multi-class classification, therefore it is worthwhile to explore new feature extraction techniques for multi-class classification problems.

We propose an orientation distance–based discriminative (ODD) feature extraction technique which extracts distinctive features in the kernel space after constructing support vector machines for each decomposed binary classification problem. In all, our proposed method works in two steps.

1. In the first step, we propose the extension of the Fisher discriminant–based method [21] to determine a proper kernel function and map the source data from the input space into a feature space where the classes are found to be distinguishable.
2. In the second step, we put forward two variants of ODD feature extraction, i.e., one-vs-all-based ODD and one-vs-one-based ODD feature extraction. We construct SVMs according to the one-vs-all or one-vs-all schemes to separate the classes into distinctive domains; we then extract one-vs-all-based ODD and one-vs-one-based ODD features by calculating the orientation distance between each sample and each constructed hyper-plane (SVM).

In the following, we exhibit the two steps in detail.

3.1 The extension of Fisher discriminant for Kernel function selection

In the kernel method, the input data is mapped from the input space (R^m) into another higher-dimensional feature space (F) via a nonlinear mapping function ($\phi(.)$) or kernel function ($K(.,.)$). In the feature space, the classes are expected to be much more linearly separable from each other [23]. In addition, the inner products of the two vectors in the feature space can be obtained efficiently and directly from the original data items using a kernel function. RBF as shown in (1) is a typical kernel function in many real-world applications.

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\| \mathbf{x} - \mathbf{x}_i \|_2 / 2\sigma^2) \tag{1}$$

Let us take a toy problem for example as illustrated in Fig. 2: two classes are nonlinearly separable in the input space (left panel (A)), but are linearly separably from each other after mapping them into a three-dimensional feature space via nonlinear mapping. Due to the flexible generation of the kernel method, it has demonstrated its power to enhance the performance of many machine-learning tools such as the support vector machine (SVM) [29], the kernel principal component analysis (KPCA) [33].

In the first step of our proposed feature extraction method, the kernel method is adopted to embed the source multi-class classification data into a feature space where each class is expected to be distinguishable from the other. As for the choice of kernel function, most

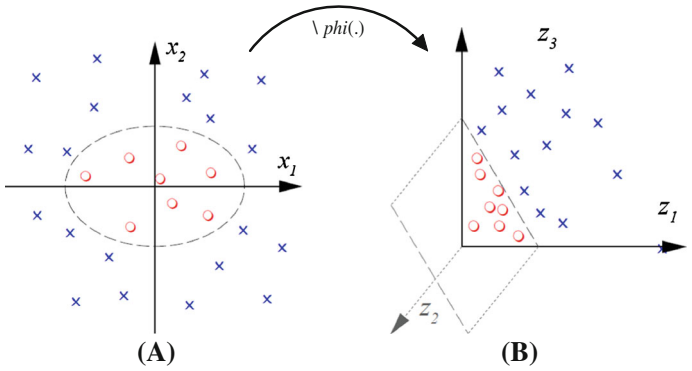

Fig. 2 Illustration of kernel mapping for two-class example. **a** Both classes are nonlinearly separable in the input space. **b** In the feature space, both classes are linearly separable after nonlinear mapping $\phi(.) = (z_1, z_2, z_3) = (x_1^2, x_2^2, \sqrt{2}x_1 \cdot x_2)$ or kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$

previous works have adopted the strategy of employing a type of kernel function first and then determining the proper parameters for the selected type of kernel function [5,6]. As with the previous work, we first employ a specific type of kernel function and then propose the approach to determine proper kernel parameters which maximizes the *between-classes distance* of the classes, while minimizing the *within-classes scatter* of the classes. For example, if RBF kernel function as in (1) is selected as the preferred function, we then need to determine kernel parameter σ.

In order to determine a proper kernel function, we borrow the idea of Fisher Discriminant [17,30] to select a kernel function such that the classes are separable in the feature space. For the Fisher Discriminant, it determines a canonical direction, and the data from different classes are most separated when they are projected on a line in this direction. Fisher Discriminant method attempts to determine a projection and may have problem in determining such projection, since even if the data are separable themselves, there may not exist an projection where the projected data are well separated, as illustrated in Fig. 1. In our method, instead of determining a projection from the space, we utilize the idea of Fisher Discriminant to determine a feature space related with the selected kernel function, where the data of the same class share high similarity, while the data from the different classes share less similarity. First of all, several notations are defined for a training set S with C classes.

Definition 1 Assume the sample size of Class i is l_i, and $\mathbf{x}_{ij} \in Class\, i$, j means the sample index in Class i, the *class center* (m_i) and the *within-class scatter* of Class i (SC_i) in the feature space are defined as

$$m_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \phi(\mathbf{x}_{ij}), \tag{2}$$

$$\begin{aligned} SC_i^2 &= \frac{1}{l_i^2} \sum_{j=1}^{l_i} \|\phi(\mathbf{x}_{ij}) - m_i\|^2 \\ &= \sum_{j=1}^{l_i} K(\mathbf{x}_{ij}, \mathbf{x}_{ij}) - \frac{1}{l_i} \sum_{j=1}^{l_i} \sum_{k=1}^{l_i} K(\mathbf{x}_{ij}, \mathbf{x}_{ik}). \end{aligned} \tag{3}$$

In Definition 1, the class center of each class is implicitly defined; SC_i is denoted as the average distance between each sample and its class center which can be explicitly calculated via kernel function without knowing the actual formulation of the nonlinear mapping function. In general, the smaller value is SC_i, the compacter is class i and more similar are instances of class i.

Definition 2 The *class distance* between Classes i and j is denoted as d_{ij}:

$$\begin{aligned} d_{ij}^2 &= \|m_i - m_j\|^2 \\ &= \frac{1}{l_i^2}\sum_{k=1}^{l_i}\sum_{n=1}^{l_i} K(\mathbf{x}_{ik}, \mathbf{x}_{in}) + \frac{1}{l_j^2}\sum_{k=1}^{l_j}\sum_{n=1}^{l_j} K(\mathbf{x}_{jk}, \mathbf{x}_{jn}) \\ &\quad -\frac{2}{l_i l_j}\sum_{k=1}^{l_i}\sum_{n=1}^{l_j} K(\mathbf{x}_{ik}, \mathbf{x}_{jn}) \end{aligned} \tag{4}$$

In Definition 2, the distance between classes i and j is defined as the distance between their class centers. As with SC_i, the actual value can be explicitly calculated by kernel function. In general, the smaller the value of d_{ij}, the more two classes are likely to overlap.

Definition 3 The *discriminant function* $J_F(\pi)$ is

$$J_F(\pi) = \frac{\sum_{i=1}^{C-1}\sum_{j=i+1}^{C} d_{ij}^2}{\sum_{i=1}^{C} SC_i^2}, \tag{5}$$

where π is the parameter set in a selected type of kernel function. For example, in polynomial kernel function $K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x} \cdot \mathbf{x}_i)^d$, the parameter is d, and in RBF function $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\| \mathbf{x} - \mathbf{x}_i \|_2 / 2\sigma^2)$, the parameter is σ.

In Definition 3, from the intuitive understanding and the essence of the Fisher discriminant idea, the kernel parameter can be chosen by maximizing $J_F(\pi)$ such that the data from the same class share high similarity, while the samples from the different classes have less similarity. This is because, the larger $J_F(\pi)$ is, the larger the distance between each pair of classes (d_{ij}) is, the compacter each class (S_i) is.

In order to maximize $J_F(\pi)$, two typical optimization methods, i.e., the gradient descent method [29] and grid search [11] can be used here. In the former, the initial value of the kernel parameter is set as π_0, and the step moves to where the derivative $\frac{dJ_F(\pi)}{d\pi}$ has maximum value. However, this method always exhibits a local optimal solution. In the latter method, the kernel parameters are chosen in the parameters set to maximize $J_F(\pi)$. More specifically, assume the parameter set π has $|\pi|$ parameters as follows:

$$\pi = \{\pi_1, \pi_2, \ldots, \pi_{|\pi|}\},$$

Suppose π_i has pre-specified n_i choices

$$\{\pi_{i1}, \pi_{i2}, \ldots, \pi_{in_i}\},$$

there are in total $n_{Total} = n_1 \cdot n_2 \cdot \ldots \cdot n_{|\pi|}$ choices. One can compute (5) on each choice in π to explore the parameters with maximum value.

Because the grid search method has been widely used in SVM to tune hyper-parameters [5,6,11] and our ODD feature extraction is an SVM-based technique, we employ the grid search method to explore proper kernel parameters. The pseudo code of the determination

Algorithm 1 Determination of kernel parameters

Input: Training set S // with C classes
a specific type of kernel function. // K(.,.)
π // parameter set $\pi = \{\pi_1, \pi_2, \ldots, \pi_{|\pi|}\}$
preset choices for each π_i // each π_i has n_i choices
Output: kernel parameters

Set set π_{return} to store parameters;
Set $value = 0$ as a temporal variable;
for (each set of choice π^i in π) **do**
  Calculate the class center and within-class scatter of each class according to (2) and (3);
  Compute the class distance between each pair of classes according to (4);
  Calculate the discriminant function according to (5);
  if $J_F(\pi^i) > value$ **then**
    $value = J_F(\pi^i)$;
    $\pi_{return} = \pi^i$;
  end if
end for
Return π_{return}.

of kernel parameters algorithm is outlined in Algorithm 1. After this, the parameters in the selected kernel function are determined, which means the kernel function $K_1(.,.)$ is confirmed.

3.2 ODD feature extraction

After the determination of kernel function $K_1(.,.)$, the classes of the dataset are implicitly mapped into a feature space where the samples in the same class share high similarity, while the data in the different classes have less similarity. In addition, the inner product of two vectors in the feature space can be explicitly calculated by $K_1(.,.)$.

In the second step, we put forward two versions of ODD feature extraction, called one-vs-all-based ODD and one-vs-one-based ODD feature extraction. In the feature space related with kernel function $K_1(.,.)$, we can construct hyper-planes (SVMs) based on one-vs-all scheme or one-vs-one scheme to obtain one-vs-all-based ODD or one-vs-one-based ODD feature extraction. For the one-vs-all scheme, we consider one class as positive class and remaining classes as negative class at each round. For the one-vs-one scheme, we treat one class as positive and another class as negative class at each round. After construct hyper-planes in the feature space, we then extract the orientation distance–based discriminative (ODD) features between each sample and each hyper-plane in the feature space to represent the source data instead of the source input features. In the following, we introduce it in detail.

3.2.1 Hyper-plane construction

(a) One-vs-all-based hyper-plane construction
First, we transfer a multi-class classification into binary problems based on the one-vs-all scheme. More specifically, for a c-class problem, we take class i as positive class and the remaining classes as negative class to compose C binary classification problems. Suppose S_i is the new formed training set for the ith decomposed binary classification problem. The optimal hyper-plane for the ith binary classification problem is trained as follows.

$$
\begin{aligned}
&D_i(\mathbf{x}) = \mathbf{w}_i \cdot \phi(\mathbf{x}) + b_i = 0 \\
&\text{s.t } y_{ij}(\mathbf{w}_i \cdot \phi(\mathbf{x}_{ij}) + b) \geq 1 - \xi_{ij}, \quad j = 1, \ldots, |S_i|, \\
&\xi_{ij} \geq 0, \quad \text{for} \quad j = 1, \ldots, |S_i|.
\end{aligned} \tag{6}
$$

where $\mathbf{w}_i$ is the normal vector for the ith hyper-plane, $\phi(\mathbf{x})$ is a mapping function corresponding to $K_1(.,.)$ and b_i is a bias for the ith hyper-plane.

By introducing the Lagrange function [29], $\mathbf{w}_i$ and b_i can be determined, and hyper-planes $D_i(\mathbf{x}) = 0$ are then obtained. After this step, we obtain C hyper-planes in the feature space, i.e., $D_i(\mathbf{x}) = 0$ $(i = 1, 2, \ldots, C)$ which portion the feature space.

(b) One-vs-one-based hyper-plane construction

We firstly convert a c-class classification into binary classification problems according to the one-vs-one scheme, i.e., taking class i as the positive class and class j as the negative class to compose $C(C-1)/2$ binary classification problems. Suppose S_{ij} is the formed training set for the classification problem between class i and class j. The optimal hyper-plane for the binary classification problem is trained as follows:

$$
\begin{aligned}
&D_{ij}(\mathbf{x}) = \mathbf{w}_{ij} \cdot \phi(\mathbf{x}) + b_{ij} = 0 \\
&\text{s.t } y_k(\mathbf{w}_{ij} \cdot \phi(\mathbf{x}_k) + b_{ij}) \geq 1 - \xi_k, \\
&\xi_k \geq 0, \quad \text{for } \mathbf{x}_k \in S_{ij}, \\
&\quad i = 1, 2, \ldots C-1; \quad j = i+1, \ldots, C.
\end{aligned} \tag{7}
$$

where $\mathbf{w}_{ij}$ is the normal vector, $\phi(\mathbf{x})$ is a mapping function corresponding to $K_1(.,.)$ and b_{ij} is a bias.

By introducing the Lagrange function [29], $\mathbf{w}_{ij}$ and b_{ij} can be determined, and hyper-planes $D_{ij}(\mathbf{x}) = 0$ are then obtained. After this step, we obtain $C(C-1)/2$ hyper-planes in the feature space, i.e., $D_{ij}(\mathbf{x}) = 0$ $(i = 1, 2, \ldots, C-1;\ j = i+1, \ldots, C)$.

3.2.2 ODD-based feature extraction

Second, for each instance in the training set S, we can calculate the orientation distances between the sample $\mathbf{x}_i$ and each hyper-plane. Firstly, we have the following Theorem.

Theorem 1 *For a hyper-plane $D(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b = 0$ in the feature space, the distance between an instance $\phi(\boldsymbol{x})$ and $D(\boldsymbol{x}) = 0$ is calculated as follows.*

$$
d(\phi(\boldsymbol{x}), D(\boldsymbol{x})) = \frac{D(\boldsymbol{x})}{\|\boldsymbol{w}\|}, \tag{8}
$$

where $\|\boldsymbol{w}\| = \sqrt{(\boldsymbol{w} \cdot \boldsymbol{w})}$ represents the Euclidean norm.

Proof The formulation of SVM: $D(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b = 0$ is a hyper-plane in the feature space.

As illustrated in Fig. 3, assume one instance $\mathbf{o}$ resides on the surface of hyper-plane, that is

$$
D(\mathbf{o}) = \mathbf{w} \cdot \phi(\mathbf{o}) + b = 0 \tag{9}
$$

Then, the distance of instance $\mathbf{p}$ and $D(\mathbf{x}) = 0$ can be calculated as follows.

$$
d(p, D(\mathbf{x})) = \frac{\overrightarrow{OP} \cdot \mathbf{w}}{\|\mathbf{w}\|} = \frac{(\mathbf{p} - \mathbf{o}) \cdot \mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{p} \cdot \mathbf{w}}{\|\mathbf{w}\|} - \frac{\mathbf{o} \cdot \mathbf{w}}{\|\mathbf{w}\|} \tag{10}
$$

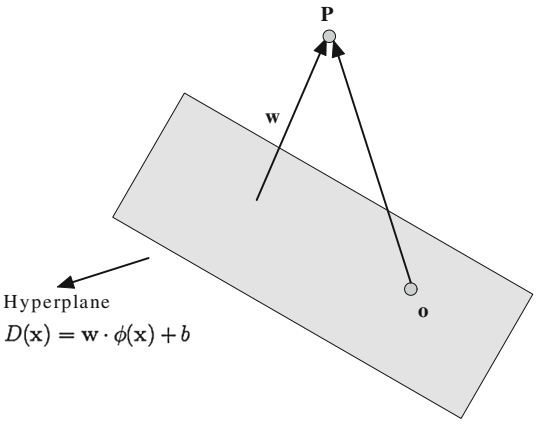

Fig. 3 An illustration of calculating the distance between an instance **p** and hyper-plane $D(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b_i = 0$ in feature space

According to Eq. (9), we have

$$d(p, D(\mathbf{x})) = \frac{\mathbf{p} \cdot \mathbf{w}}{\|\mathbf{w}\|} + b. \tag{11}$$

(a) One-vs-all-based ODD feature extraction
After calculating the orientation distance between each sample $\mathbf{x}_k$ and each hyper-plane, we then obtain the new C dimensional vector in the feature space, i.e.,

$$\frac{D_1(\mathbf{x}_k)}{\|\mathbf{w}_1\|}, \frac{D_2(\mathbf{x}_k)}{\|\mathbf{w}_2\|}, \ldots, \frac{D_C(\mathbf{x}_k)}{\|\mathbf{w}_C\|}$$

It is noted that the value of the orientation distance can be either positive or negative, which depends on the position of the sample and corresponding hyper-plane.

Let us consider the three-class problem as shown in Fig. 4, where each arrowhead denotes the orientation of each hyper-plane, $D_1(\mathbf{x}) = 0$, $D_2(\mathbf{x}) = 0$ *and* $D_3(\mathbf{x}) = 0$ represents the hyper-planes. An input vector is denoted as $(\mathbf{x}, 1)$, where 1 represents the label of sample $\mathbf{x}$. After we construct three hyper-planes based on the one-vs-all scheme and calculate the orientation distances between $\mathbf{x}$ and each hyper-plane, $(\mathbf{x}, 1)$ is transformed into $(\mathbf{x}^{new}, 1)$:

$$(\mathbf{x}, 1) \longrightarrow (\mathbf{x}^{new}, 1) = \left(\left(\frac{D_1(\mathbf{x})}{\|\mathbf{w}_1\|}, \frac{D_2(\mathbf{x})}{\|\mathbf{w}_2\|}, \frac{D_3(\mathbf{x})}{\|\mathbf{w}_3\|} \right), 1 \right). \tag{12}$$

Considering the orientation of each hyper-plane, we have

$$\frac{D_1(\mathbf{x})}{\|\mathbf{w}_1\|} > 0, \frac{D_2(\mathbf{x})}{\|\mathbf{w}_2\|} < 0, \frac{D_3(\mathbf{x})}{\|\mathbf{w}_3\|} < 0.$$

The new training set $S^{OVA-ODD}$ can be transformed from the source training set S as follows:

$$\begin{aligned} S = \begin{pmatrix} \mathbf{x}_1 & y_1 \\ \vdots & \vdots \\ \mathbf{x}_{|S|} & y_{|S|} \end{pmatrix} &= \begin{pmatrix} x_{11} & \cdots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{|S|1} & \cdots & x_{|S|m} & y_{|S|} \end{pmatrix}_{|S|*(m+1)} \\ &\longrightarrow \begin{pmatrix} \frac{D_1(\mathbf{x}_1)}{\|\mathbf{w}_1\|} & \cdots & \frac{D_C(\mathbf{x}_1)}{\|\mathbf{w}_C\|} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{D_1(\mathbf{x}_{|S|})}{\|\mathbf{w}_1\|} & \cdots & \frac{D_C(\mathbf{x}_{|S|})}{\|\mathbf{w}_C\|} & y_{|S|} \end{pmatrix}_{|S|*(C+1)} = S^{OVA-ODD}. \end{aligned} \tag{13}$$

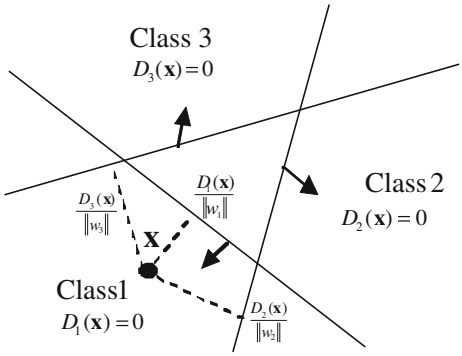

Fig. 4 Orientation distance discriminant (ODD) feature between sample $\mathbf{x}$ and each hyper-plane in terms of one-vs-all scheme

In this way, the original training sample $\mathbf{x}_k$ is represented by using a $C-$ dimensional distance vector:

$$\mathbf{x}_k \rightarrow \mathbf{x}_k^{OVA-ODD} = \left(\frac{D_1(\mathbf{x}_k)}{\|\mathbf{w}_1\|}, \frac{D_2(\mathbf{x}_k)}{\|\mathbf{w}_2\|}, \cdots, \frac{D_{C-1}(\mathbf{x}_k)}{\|\mathbf{w}_{C-1}\|}, \frac{D_C(\mathbf{x}_k)}{\|\mathbf{w}_C\|} \right). \tag{14}$$

For the testing set $S_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_n^t\}, \mathbf{x}_i^t \in R^m$, the orientation distances between a sample and each hyper-plane is also computed, and S_t is transformed into $S_t^{OVA-ODD}$ as follows:

$$\begin{pmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_{|S_t|}^t \end{pmatrix} = \begin{pmatrix} x_{11}^t & \cdots & x_{1m}^t \\ \vdots & \ddots & \vdots \\ x_{|S_t|1}^t & \cdots & x_{|S_t|m}^t \end{pmatrix}_{|S_t|*m}$$

$$\longrightarrow S_t^{OVA-ODD} = \begin{pmatrix} \frac{D_1(\mathbf{x}_1^t)}{\|\mathbf{w}_1\|} & \cdots & \frac{D_C(\mathbf{x}_1^t)}{\|\mathbf{w}_C\|} \\ \vdots & \ddots & \vdots \\ \frac{D_1(\mathbf{x}_{|S_t|}^t)}{\|\mathbf{w}_1\|} & \cdots & \frac{D_C(\mathbf{x}_{|S_t|}^t)}{\|\mathbf{w}_C\|} \end{pmatrix}_{|S_t|*C}. \tag{15}$$

(b) One-vs-one-based ODD feature extraction

Based on Theorem 1, the distance between an instance $\phi(\mathbf{x}_k)$ and $D_{ij}(\mathbf{x}) = 0$ is calculated as follows.

$$d(\phi(\mathbf{x}_k), H_{ij}) = \frac{D_{ij}(\mathbf{x}_k)}{\|\mathbf{w}_{ij}\|},$$
$$\text{for } i = 1, 2, \ldots, C-1; \quad j = i+1, \ldots, C. \tag{16}$$

where $\|\mathbf{w}_{ij}\| = \sqrt{(\mathbf{w}_{ij} \cdot \mathbf{w}_{ij})}$ represents the Euclidean norm, H_{ij} denotes the hyper-plane between class i and class j.

After calculating the orientation distance between each sample $\mathbf{x}_k$ and each hyper-plane, we then obtain the new $C(C-1)/2$ dimensional vector in the feature space, i.e.,

$$\frac{D_{12}(\mathbf{x}_k)}{\|\mathbf{w}_{12}\|}, \frac{D_{13}(\mathbf{x}_k)}{\|\mathbf{w}_{13}\|}, \ldots, \frac{D_{1C}(\mathbf{x}_k)}{\|\mathbf{w}_{1C}\|}, \frac{D_{23}(\mathbf{x}_k)}{\|\mathbf{w}_{23}\|}, \ldots, \frac{D_{C-1C}(\mathbf{x}_k)}{\|\mathbf{w}_{C-1C}\|}$$

We can discover that the orientation distance value can be either minus or plus, which depends on the location of a sample and the corresponding hyper-plane.

Let us consider a three-class problem as shown in Fig. 5, where each arrowhead denotes the orientation of each hyper-plane, $D_{12}(\mathbf{x}) = 0$, $D_{13}(\mathbf{x}) = 0$, $D_{23}(\mathbf{x}) = 0$ represents the

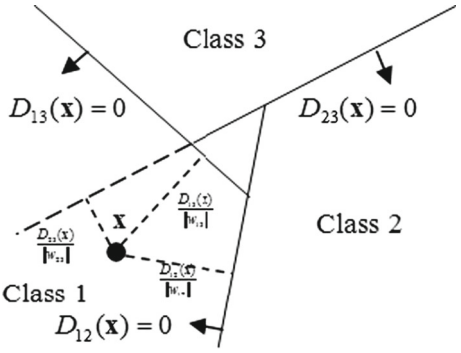

Fig. 5 Orientation distance discriminant (ODD) feature between sample $\mathbf{x}$ and each hyper-plane in terms of one-vs-one scheme

hyper-planes. An input vector is denoted as $(\mathbf{x}, 1)$, where 1 is sample label. After we construct three hyper-planes based on the one-vs-one scheme and calculate the orientation distances between $\mathbf{x}$ and each hyper-plane, $(\mathbf{x}, 1)$ is transformed into $(\mathbf{x}^{new}, 1)$:

$$(\mathbf{x}, 1) \longrightarrow (\mathbf{x}^{new}, 1) = \left(\left(\frac{D_{12}(\mathbf{x})}{\|\mathbf{w}_{12}\|}, \frac{D_{13}(\mathbf{x})}{\|\mathbf{w}_{13}\|}, \frac{D_{23}(\mathbf{x})}{\|\mathbf{w}_{23}\|}\right), 1\right). \tag{17}$$

Considering the orientation of each hyper-plane, we have

$$\frac{D_{12}(\mathbf{x})}{\|\mathbf{w}_{12}\|} > 0, \frac{D_{13}(\mathbf{x})}{\|\mathbf{w}_{13}\|} > 0, \frac{D_{23}(\mathbf{x})}{\|\mathbf{w}_{23}\|} > 0.$$

The new training set $S^{OVO-ODD}$ can be transformed from the source training set S as follows.

$$\begin{aligned} S &= \begin{pmatrix} x_{11} & \cdots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{|S|1} & \cdots & x_{|S|m} & y_{|S|} \end{pmatrix}_{|S|*(m+1)} \\ &\longrightarrow \begin{pmatrix} \frac{D_{12}(\mathbf{x}_1)}{\|\mathbf{w}_{12}\|} & \cdots & \frac{D_{C-1,C}(\mathbf{x}_1)}{\|\mathbf{w}_{C-1,C}\|} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{D_{12}(\mathbf{x}_{|S|})}{\|\mathbf{w}_{12}\|} & \cdots & \frac{D_{C-1,C}(\mathbf{x}_{|S|})}{\|\mathbf{w}_{C-1,C}\|} & y_{|S|} \end{pmatrix}_{|S|*(1+\frac{C(C-1)}{2})} \\ &= S^{OVO-ODD}. \end{aligned} \tag{18}$$

In this way, the original training sample $\mathbf{x}_k$ is represented by using a $C(C-1)/2$ dimensional distance vector in (19)

$$\begin{aligned} \mathbf{x}_k \rightarrow \mathbf{x}_k^{OVO-ODD} &= \left(\frac{D_{12}(\mathbf{x}_k)}{\|\mathbf{w}_{12}\|}, \frac{D_{13}(\mathbf{x}_k)}{\|\mathbf{w}_{13}\|}, \ldots, \right. \\ &\quad \left. \frac{D_{C-2,C}(\mathbf{x}_k)}{\|\mathbf{w}_{C-2,C}\|}, \frac{D_{C-1,C}(\mathbf{x}_k)}{\|\mathbf{w}_{C-1,C}\|}\right). \end{aligned} \tag{19}$$

Table 1 Computational complexity of feature extraction methods

OVA-ODD	OVO-ODD	KPCA	KICA	KFDA
$C \cdot O(l)^{\lambda}$	$\frac{C(C-2)}{2} \cdot O(\frac{l}{C})^{\lambda}$	$O(l)^2$	$O(l)^2$	$O(l)^2$

For the testing set $S_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_n^t\}, \mathbf{x}_i^t \in R^m$, the one-vs-one-based ODD feature can be extracted, and S_t is transformed into $S_t^{OVO-ODD}$ as follows.

$$\begin{pmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_{|S_t|}^t \end{pmatrix} = \begin{pmatrix} x_{11}^t & \cdots & x_{1m}^t \\ \vdots & \ddots & \vdots \\ x_{|S_t|1}^t & \cdots & x_{|S_t|m}^t \end{pmatrix}_{|S_t| * m}$$

$$\longrightarrow S_t^{OVO-ODD} = \begin{pmatrix} \frac{D_{12}(\mathbf{x}_1^t)}{\|\mathbf{w}_{12}\|} & \cdots & \frac{D_{C-1,C}(\mathbf{x}_1^t)}{\|\mathbf{w}_{C-1,C}\|} \\ \vdots & \ddots & \vdots \\ \frac{D_{12}(\mathbf{x}_{|S_t|}^t)}{\|\mathbf{w}_{12}\|} & \cdots & \frac{D_{C-1,C}(\mathbf{x}_{|S_t|}^t)}{\|\mathbf{w}_{C-1,C}\|} \end{pmatrix}_{|S_t| * \frac{(C(C-1)}{2})} . \quad (20)$$

3.3 Computational complexity analysis

In this section, we analyze the computational complexity of one-vs-all-based ODD and one-vs-one-based ODD feature extraction. Assume binary SVM generally suffers from an $O(n)^{\lambda}$ training cost where n represents the training sample size and $\lambda < 2$. For a $C-$class problem, one-vs-all-based ODD method construct C hyper-planes; however, one-vs-one-based ODD method construct $C(C-1)/2$ hyper-planes. Assume each class equally has l/C samples, the computational complexity of one-vs-all-based and one-vs-one-based ODD feature extraction and KPCA, KICA, KFDA is listed in Table 1.

4 Experiments

4.1 Baselines and metrics

In this section, we implement two invariants of our ODD feature extraction, i.e., one-vs-all-based ODD and one-vs-one-based ODD methods to investigate the performance of the ODD features. For comparison, another three classical feature extraction methods are used as baselines.

– The first method is kernel principal component analysis (KPCA) [37,33], which always performs better than principal component analysis (PCA).

The first baseline is unsupervised feature extraction method which does not take the sample label into the learning phase. This baseline is used to investigate our ODD features compared with the unsupervised feature extractions method.

– The second method is kernel linear discriminant analysis (KLDA or KFDA) [39,41], which determines an direction on which the projections of the data are distinguishable.
– The third method is margin maximizing discriminant analysis (MMDA) [13,28] which determines the projection direction by margin maximizing hyper-planes, and therefore is called as nonparametric extension of LDA [13].

Table 2 UCI datasets description

Dataset	# of samples	# of class	# of features
Dermatology	366	6	34
Soybean	683	19	35
Vowel	990	10	11
Vehicle	846	4	18
Optdigits	5,620	10	64
Pageblock	5,473	5	10
Satimage	6,435	6	36
Thyroid	7,200	3	21
Isolet	7,797	26	617
Pendigits	10,992	10	16

The second and third baselines are the supervised feature extraction methods which incorporate the data label into the learning. The two baselines are utilized to show the improvement of our ODD features compared with the supervised feature extraction methods. Since the three baselines, one-vs-all-based and one-vs-one-based ODD method are all kernel-based feature extraction methods, RBF kernel function in (1) is used in the experiments. All the experiments are conducted on a laptop with a Dual 2.8GHz Intel Core2 T9600 PC and 4GB RAM.

In general, the performance of classification is evaluated in terms of accuracy. We utilize this metrics in the experiments.

4.2 Datasets

We used the ten UCI datasets [1], which has been previously studied by other researchers for multi-class classification, in our experiments. The general information of the used datasets is illustrated in Table 2. These UCI datasets are from real-world application problems, such as diseases, handwritten digits and satellite image classification. One may refer to [1] for detail. Thus, the experiments include a various of real-world multi-class classification applications.

At the pre-processing stage, all records in each dataset are normalized to $[-1, 1]$. In the RBF kernel (1), the parameter σ is searched in the range of

$$\{\sigma_0/8, \sigma_0/4, \sigma_0/2, \sigma_0, 2\sigma_0, 4\sigma_0, 8\sigma_0\}, \tag{21}$$

where σ_0 is calculated as $\sigma_0 = \frac{1}{|S|}\sum_{i=1}^{|S|} \mathbf{x}_i$, where $|S|$ is the number of dataset.

4.3 Experiment settings

In order to investigate the performance of the ODD features, the experiments are organized as follows.

We perform ten-folder cross-validation to generate 10 groups of training data and testing data for each dataset. For each group of training and testing sets, the experiments consist of two steps, as illustrated in Fig. 6.

1. In the first step, we extract KPCA, KFDA, MMDA and one-vs-all-based and one-vs-one-based ODD features from the source data.
2. In the second step, we perform decision three, neural network, and support vector machine methods on the extracted features to compare the quality of these extracted features.

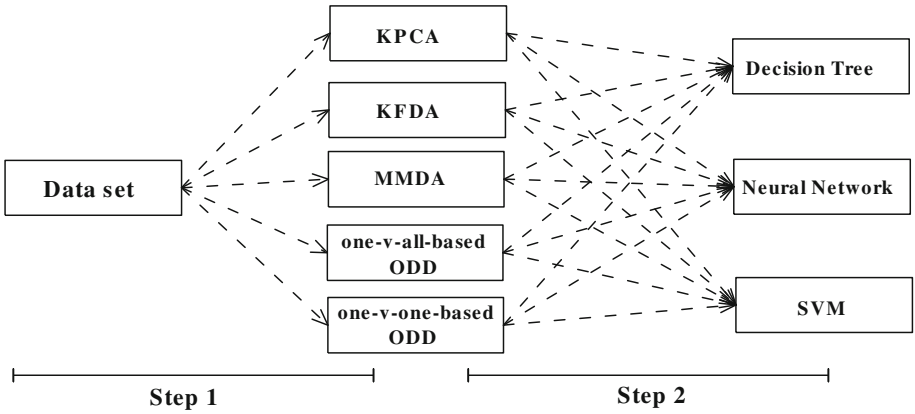

Fig. 6 The process of the experiments

Specifically, in the first step, for one-vs-all-based and one-vs-one-based ODD feature extractions, we perform the extension of the Fisher discriminant method on each choice of (21) to identify the parameter σ in (1) such that the discriminant function J_F in (5) achieves its maximum value. After that, we extract one-vs-all-based and one-vs-one-based ODD features between each sample and each hyper-plane. In this procedure, another parameter γ in SVM is set 1 when constructing binary SVM, since the dataset is normalized and this setting is acceptable in tuning parameters [12].

With respect to KPCA, KFDA and MMDA feature extraction, we keep the extracted features on each choice of (21) because we do not know which σ in (21) leads to a better performance of multi-class classification algorithms on them. Since there exist seven choices for σ in (21), we obtain seven sets of KPCA features, KICA features, KFDA features and MMDA features, respectively.

In the second step, we conduct the three multi-class classification algorithms on one-vs-all-based and one-vs-one-based ODD features and obtain their accuracies. For the seven sets of KPCA features which are obtained in the first step, we conduct the multi-class classification algorithms on each sets of KPCA features. We then obtain seven testing accuracy and retain the most high to represent the accuracy on the KPCA feature. The same operation is performed for the KFDA and MMDA features.

In the end, we report the *average accuracy* of ten groups of datasets to represent the performance of each feature extraction method.

Remark For KFDA, the rank of the between-class matrix is at most C, which is the number of classes, and so the number of features for KFD is always fixed at $C-1$ [17,28]. For MMDA, as suggested in [28], the number of the features is set from $3C$ to $5C$. Since one-vs-one-based ODD features has $C(C-1)/2$ features, the features of MMDA is set from $3C$ to $5C$ and at $C(C-1)/2$. For one-vs-all-based ODD features, the number of features equals to C; while one-vs-one-based ODD methods has $C(C-1)/2$ features. Since each sample is represented by the full ODD features in the feature space, we use them in subsequent multi-class classification algorithms. For KPCA and MMDA, as different number of extracted features are used in the subsequent multi-class classification, the performance will be different. In general, the more extracted features are used, the more information is considered, the more accuracy the algorithm will obtain. This has been widely studied in the previous work [13,28]. In the experiments, we report the most performance when increasing

Table 3 The average accuracy and standard deviation on KPCA, KFDA, MMDA, O-V-A-ODD and O-V-O-ODD features using decision tree method

Dataset	KPCA	KFDA	MMDA	O-V-A-ODD	O-V-O-ODD
Dermatology	90.7 ± 5.2	91.6 ± 4.6	92.4 ± 4.3	93.4 ± 4.1	$\mathbf{93.9 \pm 4}$
Soybean	93.5 ± 7.1	93.8 ± 6.9	94.4 ± 6.6	95.4 ± 6.2	$\mathbf{96.3 \pm 6.1}$
Vowel	96.1 ± 4.3	96.4 ± 4.2	97.1 ± 4.2	$97.9 \pm \mathbf{4.1}$	$\mathbf{98.5 \pm 4.1}$
Vehicle	84.6 ± 4.9	$85.2 \pm \mathbf{4.7}$	$86.2 \pm \mathbf{4.7}$	86.8 ± 4.8	$\mathbf{87.1} \pm 4.8$
Isolet	94.5 ± 4.4	94.8 ± 4.3	95.2 ± 4.3	$95.6 \pm \mathbf{4.1}$	$\mathbf{96.1 \pm 4.1}$
Thyroid	95.7 ± 4.2	96.1 ± 3.8	96.1 ± 3.8	$96.8 \pm \mathbf{3.6}$	$\mathbf{97.1 \pm 3.6}$
Pageblock	96.8 ± 3.2	97.1 ± 2.7	97.4 ± 2.6	$97.9 \pm \mathbf{2.5}$	$\mathbf{98.4 \pm 2.5}$
Pendigits	97.2 ± 3.1	97.4 ± 3.1	97.4 ± 2.8	$97.8 \pm \mathbf{2.7}$	$\mathbf{98.6 \pm 2.7}$
Optdigits	92.1 ± 4.5	93.1 ± 4.5	93.6 ± 4.3	$94.9 \pm \mathbf{4.2}$	$\mathbf{95.4 \pm 4.2}$
Satimage	90.4 ± 2.8	$90.9 \pm \mathbf{2.7}$	91.3 ± 3.2	$92.3 \pm \mathbf{2.7}$	$\mathbf{92.7} \pm 2.7$
Average	93.16 ± 4.37	93.64 ± 4.15	94.11 ± 4.08	94.88 ± 3.92	$\mathbf{95.41 \pm 3.88}$

the number of extracted features in the multi-class classification algorithms, and the number of the features includes $C(C-1)/2$ features if the corresponding feature extraction methods can have.

4.4 Comparison of feature extraction methods

In this section, we compare the performance of the KPCA, KFDA, MMDA and one-vs-all-based ODD and one-vs-one-based ODD feature extraction methods.

4.4.1 Average accuracy and standard deviation

We will report the average accuracy and standard deviation of ten groups of sets from Tables 3, 4, 5. It is clear that the average accuracy of one-vs-all-based and one-vs-one-based ODD features is always higher than that of KPCA, KFDA and MMDA. Since our one-vs-all-based and one-vs-one-based ODD feature extraction methods are supervised methods which take the label information into the learning phase, while KPCA is unsupervised methods; as a results, our ODD method can extract distinctive features compared with those extracted by KPCA. Further, although KFDA and MMDA are supervised feature extraction methods; however, our one-vs-all-based and one-vs-one-based ODD features outperform them.

In addition, we report the Wilcoxon test [48] between KPCA, KFDA and MMDA methods and our one-vs-all-based and one-vs-one-based ODD methods, respectively, using the accuracies by decision tree, neural networks and SVM. The Wilcoxon test values are reported in Table 6. The wilcoxon test is used to test whether there exists significant difference between a pair of data. If the returned value is less than 0.05, it is believed there exists significant difference in statistics. From Table 6, we discover that the Wilcoxon test between KPCA, KFDA, MMDA and one-vs-all-based ODD method is always than 0.05. The same observation is found between KPCA, KFDA, MMDA and one-vs-one-based ODD method. These mean that one-vs-all-based and one-vs-one-based ODD feature extraction methods perform better than KPCA, KFDA and MMDA, respectively.

We further discover that, the performance of one-vs-one-based ODD feature extraction is little higher than that of one-vs-all-based ODD features in most cases, since one-vs-one-

Table 4 The average accuracy and standard deviation on KPCA, KFDA, MMDA, O-V-A-ODD and O-V-O-ODD features using neural network method

Dataset	KPCA	KFDA	MMDA	O-V-A-ODD	O-V-O-ODD
Dermatology	91.5 ± 5.1	92.3 ± 4.7	92.7 ± 4.3	93.6 ± 4.2	**94.2 ± 4.1**
Soybean	93.8 ± 7.1	94.2 ± 6.8	95.2 ± 6.6	95.7 ± 6.2	**96.4 ± 6.1**
Vowel	96.4 ± 4.3	97.1 ± 4.1	97.8 ± 4.1	98.3 ± **3.9**	**98.7** ± 4
Vehicle	85.2 ± 4.8	85.9 ± **3.6**	86.7 ± 4.7	87.1 ± 4.8	**87.6** ± 4.7
Isolet	94.7 ± 4.4	95 ± 4.4	95.4 ± 4.2	96.1 ± **4**	**96.5** ± 4.1
Thyroid	96.2 ± 3.8	96.4 ± 3.4	96.5 ± 3.3	97.2 ± **3.2**	**97.3 ± 3.2**
Pageblock	97.2 ± 3.1	97.4 ± 3	97.5 ± 3.1	98.4 ± 2.7	**98.5 ± 2.5**
Pendigits	97.4 ± 3.1	97.4 ± 3	97.5 ± 2.8	98.2 ± 2.7	**98.7 ± 2.6**
Optdigits	92.3 ± 4.3	93.4 ± 4.3	93.9 ± **4.2**	95.4 ± **4.2**	**95.6 ± 4.2**
Satimage	90.4 ± 2.8	91.4 ± **2.6**	91.7 ± 3.1	92.5 ± 2.7	**92.9 ± 2.6**
Average	93.51 ± 4.28	94.05 ± 3.99	94.49 ± 4.04	95.25 ± 3.86	**95.64 ±3.81**

Table 5 The average accuracy and standard deviation on KPCA, KFDA, MMDA, O-V-A-ODD and O-V-O-ODD features using SVM-based one-vs-one method

Dataset	KPCA	KFDA	MMDA	O-V-A-ODD	O-V-O-ODD
Dermatology	92.3 ± 5	93.5 ± 4.6	93.9 ± 4.2	94.2 ± 4.1	**94.6 ± 4**
Soybean	94.7 ± 7.2	95.5 ± 6.7	95.6 ± 6.5	96.3 ± **6.1**	**96.8 ± 6.1**
Vowel	97.7 ± 4.1	97.8 ± 4.1	98.1 ± 3.9	99.1 ± **3.8**	**99.4 ± 3.8**
Vehicle	85.6 ± 4.8	86.9 ± 3.8	87.5 ± 4.5	87.6 ± **4.4**	**87.9 ± 4.4**
Isolet	95.2 ± 4.2	95.3 ± 4.2	95.7 ± 3.9	96.5 ± 3.9	**96.8 ± 3.8**
Thyroid	96.7 ± 3.7	96.9 ± **3.1**	96.9 ± 3.3	97.5 ± **3.1**	**97.7 ± 3.1**
Pageblock	97.6 ± 3.1	97.7 ± 2.9	97.9 ± 3	98.5 ± 2.7	**98.7 ± 2.4**
Pendigits	97.9 ± 3	98.5 ± 3.1	98.5 ± 2.7	98.7 ± **2.6**	**98.9 ± 2.6**
Optdigits	93.2 ± 4.2	93.5 ± 4.2	94.2 ± 4.1	95.8 ± 4.2	**96.2 ± 3.9**
Satimage	91.8 ± 2.7	92.1 ± 2.7	92.4 ± 2.9	92.8 ± 2.6	**93.2 ± 2.5**
Average	94.27 ± 4.2	94.77 ± 3.94	95.07 ± 3.9	95.7 ± 3.74	**96.02 ± 3.67**

Table 6 The Wilcoxon test value between each method and O-V-A-ODD and O-V-O-ODD methods, respectively

Dataset	O-V-A-ODD	O-V-O-ODD
KPCA	1.7170×10^{-6}	1.7246×10^{-6}
KFDA	1.7030×10^{-6}	1.7181×10^{-6}
MMDA	1.6923×10^{-6}	1.6998×10^{-6}

based ODD method always extracts more features from the source data compared with one-vs-all-based ODD feature extraction. We perform Wilcoxon test to study the performance of one-vs-all-based and one-vs-one-based ODD features. The returned wilcoxon value is 1.6271×10^{-6}, it is <0.05. This shows that one-vs-one-based ODD feature extraction method performs better than one-vs-all-based ODD features in statistics.

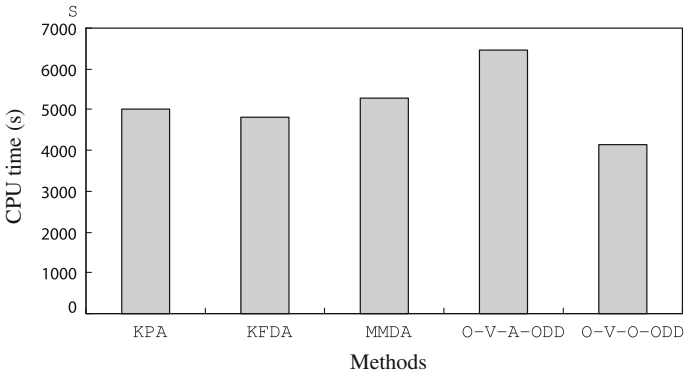

Fig. 7 Average running time of each feature extraction method

In addition, we find KFDA and MMDA outperform KPCA, this is because KPCA is a unsupervised feature extraction method while the former two are supervised methods which can extract more distinctive features from the source data. In addition, the standard deviation of one-vs-all-based ODD and one-vs-one-based ODD feature extraction methods are less than those of other methods for most datasets.

4.4.2 Average running time comparison

We have compared the performance of each feature extraction method and found that one-vs-all-based and one-vs-one-based ODD feature extraction methods are superior to other methods, it is still interesting to compare the running time of each method. For each method, we calculate the average running time of ten datasets and illustrate them in Fig. 7.

We discover that one-vs-all-based ODD feature extraction takes the longest time since it takes all the samples of the datasets to construct binary hyper-plane at each round. However, one-vs-one-based ODD feature takes the least running time, this is because one-vs-one-based ODD feature extraction only takes two of classes from the dataset to construct a binary hyper-plane; thus, it saves the running time compared with other feature extraction methods.

4.5 Comparison with SVM-based multi-class method on input data

In the one-vs-all-based and one-vs-one-based ODD feature extraction methods, we first perform SVM based on one-vs-all and one-vs-one schemes to construct hyper-planes and then conduct SVM-based multi-class classification methods on the extracted ODD features. In this set of experiments, we conduct SVM-based one-vs-all and one-vs-one multi-class classification methods on the input data to obtain the results, respectively, and then compare them with those based on one-vs-all-based and one-vs-one-based ODD features using SVM as classifier Fig. 8. It can be seen that, after we extract the ODD features and conduct SVM-based multi-class methods on them, the obtained performance is higher than those of SVM-based one-vs-all and SVM-based one-vs-one methods on the input data.

4.6 Comparison with feature selection method

In this section, we compare the one-vs-all-based and one-vs-one-based ODD feature extraction method with feature selection method. As discussed in the related work, feature selection

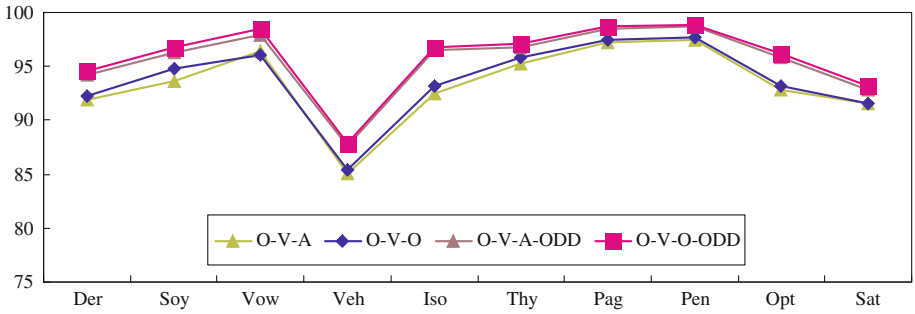

Fig. 8 The comparison with one-vs-all and one-vs-one multi-class classification methods

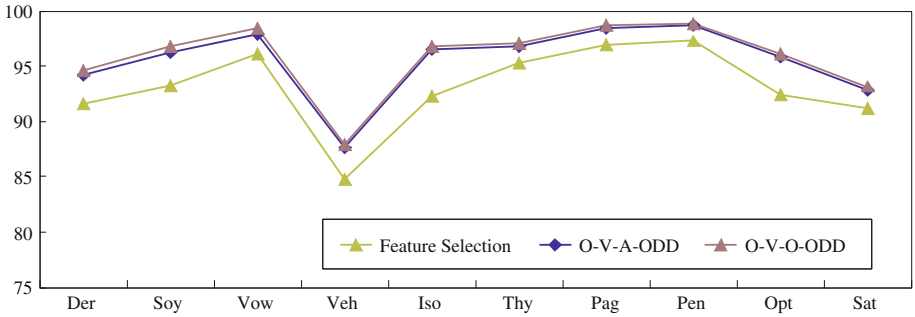

Fig. 9 The comparison with feature selection method and ODD features

aims at selecting a subset of relevant features from the original source features, while feature extraction constructs combinations of the source variables. We perform forward feature selection method [47] to select a subset of relevant features from the input features. In all, the performance of one-vs-all-based and one-vs-one-based ODD features and feature selection is illustrated in Fig. 9. It can be seen that, one-vs-all-based and one-vs-one-based ODD features perform better than the feature selection method.

4.7 ODD features for binary classification problems

ODD feature extraction method is based on one-vs-all and one-vs-one schemes, which are designed for multi-class classification problems. The same as one-vs-all and one-vs-one scheme, ODD feature extraction is designed for the multi-class classification problem. In case of binary classification problems, we can also extract ODD features: construct a hyper-plane to separate the two classes and extract orientation distance–based feature toward the constructed hyper-plane. Since we only construct one hyper-plane, the extracted ODD feature only has one dimension.

In this set of experiment, we study the ODD feature for binary classification problems. The used four binary classification datasets from UCI datasets are illustrated in Table 7. By using ten-folder cross-validation, we illustrate the average accuracy of the results on the input data and the results on the extracted ODD feature in Table 7. It can be seen that, the performance on the ODD feature is only slightly better or the same as the results on the input data. After construct hyper-plane for binary classes, support vector machine uses the sign of each sample to the constructed hyper-plane to make decision, i.e., if the orientation distance toward the

Table 7 UCI binary classes datasets description and the results on input features and on ODD features

Dataset	# of samples	# of class	# of features	Results on input features	Results on ODD features
Ripley	1,250	2	2	**88.9**	**88.9**
Pima	768	2	8	78.2	**78.3**
Waveform	5,000	2	21	90.1	**90.15**
Splice	3,175	2	62	**89.3**	**89.3**

hyper-plane is plus, the sample is assigned to the plus class; otherwise, it is classified into the minus class. Since we utilize the extracted ODD feature as input of following support vector machine classifier, the accuracy should not be less than the results on the input features. However, the extracted ODD feature only has one dimension; the results on the ODD feature is not much better than the results on the input features.

5 Conclusions and future work

While many feature extraction methods have been proposed, they are often not suitable for identifying discriminative features for multi-class classification and potentially result in low classification accuracy. This paper has proposed a novel feature extraction method, to extract orientation distance–based discriminative (ODD) features, specifically designed for multi-class classification problems. The proposed method works well in two steps. In the first step, the kernel function is determined by extending the Fisher discriminant idea. In the second step, one-vs-all-based and one-vs-one-based ODD features are then extracted to generate discriminative features. Substantial experiments on ten UCI datasets have shown that our proposed one-vs-all-based and one-vs-one-based ODD features method outperforms state-of-the-art feature extraction methods.

We are extending our work in several directions. We would like to apply the ODD feature extraction method to the online environment; we also plan to use the proposed method to bioinformatics with multi-class classification data.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by US NSF through grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, Google Mobile 2014 Program, HUAWEI and KAU grants, Natural Science Foundation of China (61070033, 61203280, 61202270), Natural Science Foundation of Guangdong province (9251009001000005, S2011040004187, S2012040007078), Specialized Research Fund for the Doctoral Program of Higher Education (20124420120004), Australian Research Council Discovery Grant (DP1096218, DP130102691) and ARC Linkage Grant (LP100200774 and LP120100566).

References

1. Blake CL, MERZ CJ (1998) UCI Repository of machine learning databases: http://www.ics.uci.edu/mlearn/MLRepository.html
2. Chien J, Chen BC (2003) A new independent component analysis for speech recognition and separation. IEEE Trans Pattern Anal Mach Intell 14(4):1245–1254
3. Dagher I, Nachar R (2006) Face recognition using ipca-ica algorithm. IEEE Trans Pattern Anal Mach Intell 28(6):996–1000
4. Devijver PA, Kittler J (1982) Pattern recognition: a statistical approach. Prentice Hall, London

5. Escalera S, Pujol O, Radeva P (2011) Online error correcting output codes. Pattern Recognit Lett 32(3):458–467
6. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. Pattern Recognit 44(8):1761–1776
7. Girolami M, Cichocki A, Amari SI (1998) A common neural network model for unsupervised exploratory data analysis and independent component analysis. IEEE Trans Neural Netw 9(6):1495–1501
8. Guyon I, Gunn S, Nikravesh M, Zadeh L (2006) Feature extraction foundations and applications. Studies in fuzziness and soft computing. Springer, Germany
9. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422
10. He Q, Xie Z, Hu Q (2011) Neighborhood based sample and feature selection for svm classification learning. Neurocomputing 74(10):1585–1594
11. Hsu C (2011) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13:415–425
12. Keerthi S (2002) Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms. IEEE Trans Neural Netw 5:1225–1229
13. Kocsor A, Kovacs K, Szepesvari C (2004) Margin maximizing discriminant analysis. In: International conference on machine learning, pp 227–238
14. Kuo SC, Lin CJ, Liao JR (2011) 3d reconstruction and face recognition using kernel-based ica and neural networks. Expert Syst Appl 38(5):5406–5415
15. Liu Y, Lita LV, Niculescu RS, Bai K, Mitra P, Giles CL (2008) Real-time data pre-processing technique for efficient feature extraction in large scale datasets. In: ACM international conference on information and knowledge management, ACM, pp 981–990
16. Liu Z, Hsiao W, Cantarel BL (2011) Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data. Bioinformatics 27(23):3242–3249
17. Mika S, Rätsch G, Weston J, Schoölkopf B, Müller KR (1999) Fisher discriminant analysis with kernels. In: Hu YH, Larsen J, Wilson E, and Douglas S, (eds) Neural Networks for Signal Processing IX, Piscataway, NJ:IEEE, pp 41–48
18. Moustakidis SP, Theocharis JB (2010) A novel svm-based feature selection method using a fuzzy complementary criterion. Pattern Recognit 41(11):3712–3729
19. Pan F, Converse T, Ahn D, Salvetti F, Donato G (2009) Feature selection for ranking using boosted trees. In: ACM international conference on information and knowledge management, pp 2025–2028
20. Ren J, Qiu Z, Fan W, Cheng H, Yu PS (2008) Forward semi-supervised feature selection. In: Pacific-Asia conference on knowledge discovery and data mining, pp 970–976
21. Roth V, Steinhage V (2000) Nonlinear discriminant analysis using kernel function. Adv Neural Inf Process Syst 568–574 MIT Press, Cambridge
22. Schölkopf B, Mika S, Burges C, Knirsch P, Müller K, Rütsch G, Smola A (1999) Input space vs. feature space in kernel-based methods. IEEE Trans Neural Netw 10:1000–1017
23. Schölkopf B, Smola A (2011) Learning with kernels. MIT Press, Cambridge
24. Shima K, Todoriki M, Suzuki A (2004) Svm-based feature selection of latent semantic features. Pattern Recognit Lett 25(9):1051–1057
25. Song L, Smola A, Gretton A, Borgwardt K, Bedo J (2007) Supervised feature selection via dependence estimation. In: International conference on machine learning, pp 823–830
26. Sun T, Chen S, Yang J, Shi P (2008) A novel method of combined feature extraction for recognition. In: IEEE international conference on data mining, pp 1550–4786
27. Tang F, Crabb R, Tao H (2007) Representing images using nonorthogonal haar-like bases. IEEE Trans Pattern Anal Mach Intell 29(12):2120–2134
28. Tsang IW, Andras K, Kocsor TK (2006) Effcient kernel feature extraction for massive data sets. In: ACM SIGKDD conference on knowledge discovery and data mining, pp 724–729
29. Vapnik V (1998) Statistical learning theory. Springer, Berlin
30. Wang JH, Li Q, You J (2011) Fast kernel fisher discriminant analysis via approximating the kernel principal component analysis. Neurocomputing 74(17):3313–3322
31. Weng J, Zhang Y, Hwang WS (2003) Candid covariance-free incremental principal component analysis. IEEE Trans Pattern Anal Mach Intell 25(8):1034–1040
32. Weston J, Elisseeff A, Schölkopf B, Tipping M (2003) Use of zero-norm with linear models and kernel method. J Mach Learn Res 3:1439–1461
33. Xiao YQ, He YG (2011) A novel approach for analog fault diagnosis based on neural networks and improved kernel pca. Neurocomputing 74(7):1102–1115

34. Xu B, Jin X, Guo P, Bie F (2006) Kica feature extraction in application to fnn based image registration. In: International joint conference on neural networks, pp 3602–3608
35. Xu Y, Furao S, Zhao J, Hasegawa O (2009) To obtain orthogonal feature extraction using training data selection. In ACM international conference on information and knowledge management, pp 1819–1822
36. Yang J, Frangi AF, Yang JY, Zhang D, Jin Z (2005) Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
37. Zhang F (2004) A polygonal line algorithm based nonlinear feature extraction method. In: International conference on data mining, pp 281–288
38. Zhang J, Gruenwald L (2006) A high-level approach to computer document formatting. In : IEEE opening the black box of feature extraction: incorporating into high-dimensional data mining processes, pp 1550–4786
39. Zhao H, Sun S, Jing Z, Yang J (2006) Local structure based supervised feature extraction. Pattern Recognit 39:1546–1550
40. Zhou JD, Wang XD, Song H (2012) Feature selection with conjunctions of decision stumps and learning from microarray data. IEEE Trans Pattern Anal Mach Intell 34(1):174–186
41. Zhu ZB, Song ZH (2011) A novel fault diagnosis system using pattern classification on kernel fda subspace. Expert Syst Appl 38(6):6895–6905
42. Zuo W, Zhang D, Yang J, Wang K (2006) Bdpca plus lda: a novel fast feature extraction technique for face recognition. IEEE Trans Syst Man Cybern Part B Cybern 36(4):946–953
43. Dhir CS, Lee J, Lee SY (2012) Extraction of independent discriminant features for data with asymmetric distribution. Knowl Inf Syst 30(2):375
44. Zhang Z, Ye N (2011) Locality preserving multimodal discriminative learning for supervised feature selection. Knowl Inf Syst 27(3):473–490
45. Yang S, Hu B (2012) Discriminative feature selection by nonparametric bayes error minimization. IEEE Trans Knowl Data Eng 24(8):1422–1434
46. Quanz B, Huan J, Mishra M (2012) Knowledge transfer with low-quality data: a feature extraction issue. IEEE Trans Knowl Data Eng 24(10):1789–1802
47. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(4):491–502
48. Garcia S, Herrera F (2008) An extension on statistical comparisons of classifiers over multiple data sets for all Pairwise Comparisons. J Mach Learn Res 9:2677–2694

Author Biographies

Bo Liu is with the Department of Automation, Guangdong University of Technology and Department of Computer Science, University of Illinois at Chicago. His research interests include machine learning and data mining. He has published papers on IEEE TNN, IEEE TKDE, KAIS, IJCAI, ICDM, SDM and CIKM.

Yanshan Xiao is in the Faculty of Engineering and Information Technology, University of Technology, Sydney. Her research interests include multi-instance learning, data mining.

Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Professor in the Department of Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. He spent most of his career at IBM Thomas J. Watson Research Center and was manager of the Software Tools and Techniques group. His research interests include data mining, privacy preserving data publishing, data stream, Internet applications and technologies, and database systems. Dr. Yu has published more than 710 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. He is the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data. He is on the steering committee of the IEEE Conference on Data Mining and ACM Conference on Information and Knowledge Management and was a member of the IEEE Data Engineering steering committee. He was the Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering (2001–2004). He had also served as an associate editor of ACM Transactions on the Internet Technology (2000–2010) and Knowledge and Information Systems (1998–2004). In addition to serving as program committee member on various conferences, he was the program chair or co-chairs of the 2009 IEEE Intl. Conf. on Service-Oriented Computing and Applications, the IEEE Workshop of Scalable Stream Processing Systems (SSPS07), the IEEE Workshop on Mining Evolving and Streaming Data (2006), the 2006 joint conferences of the 8th IEEE Conference on E-Commerce Technology (CEC 06) and the 3rd IEEE Conference on Enterprise Computing, E-Commerce and E-Services (EEE 06), the 11th IEEE Intl. Conference on Data Engineering, the 6th Pacific Area Conference on Knowledge Discovery and Data Mining, the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, the 2nd IEEE Intl. Workshop on Research Issues on Data Engineering: Transaction and Query Processing, the PAKDD Workshop on Knowledge Discovery from Advanced Databases, and the 2nd IEEE Intl. Workshop on Advanced Issues of E-Commerce and Web-based Information Systems. He served as the general chair or co-chairs of the 2009 IEEE Intl. Conf. on Data Mining, the 2009 IEEE Intl. Conf. on Data Engineering, the 2006 ACM Conference on Information and Knowledge Management, the 1998 IEEE Intl. Conference on Data Engineering, and the 2nd IEEE Intl. Conference on Data Mining. He had received several IBM honors including 2 IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, 2 Research Division Awards and the 94th plateau of Invention Achievement Awards. He was an IBM Master Inventor. Dr. Yu received a Research Contributions Award from IEEE Intl. Conference on Data Mining in 2003 and also an IEEE Region 1 Award for promoting and perpetuating numerous new electrical engineering concepts in 1999.

Zhifeng Hao is with the Faculty of Computer, Guangdong University of Technology. His current research interests include design and analysis of algorithm, mathematical modeling, and combinatorial optimization.

Longbing Cao is a professor in the Faculty of Engineering and Information Technology, University of Technology, Sydney. His research interests include data mining, multiagent technology, and agent and data mining integration. He is a senior member of the IEEE.