Computer Communication & Collaboration (Vol. 4, Issue 4, 2016)
ISSN 2292-1028(Print) 2292-1036(Online)
Submitted on Oct. 19, 2016
DOIC: 2292-1036-2016-04-001-74

Generalized Kernel Discriminant Analysis using Weighting Function with Applications to Feature Extraction

Jing Yang (Correspondence author)
School of Computer Science and Technology
Nanjing University of Science and Technology
Nanjing, Jiangsu, China, 210094
E-mail: yangjing860204@163.com

Liya Fan
Liaocheng University
School of Mathematics Sciences
Liaocheng, Shangdong, China, 252059
E-mail: fanliya63@126.com

Quansen Sun
School of Computer Science and Technology
Nanjing University of Science and Technology
Nanjing, Jiangsu, China, 210094
E-mail: sunquansen@mail.njust.edu.cn

Abstract: Linear discriminant analysis (LDA) is a classical approach for dimensionality reduction. However, LDA has shortcomings in that one of the scatter matrices is required to be nonsingular and the nonlinearly clustered structure is not easily captured, moreover, the adverse effects due to outlier classes also affect the performance of LDA. In order to solve these problems, in this paper, several nonlinear generalizations of LDA using weighting function are presented and called them weighted generalized KDA algorithms. Experiments on three real-world data sets are performed to evaluate the effectiveness of the proposed algorithms and the effect of weights on kernel functions. The results show that the effect of weighted schemes on kernel functions is very significantly.

Keywords: Kernel Discriminant Analysis, Under-sampled Problem, Weighting Function, Kernel Function, Misclassification Rate

1. Introduction

Feature extraction process is an important part of pattern recognition and machine learning, which can results in computation cost decreasing and classification performance increasing. An appropriate representation of data from all features is an important problem in machine learning and data mining problems. All original features can not always beneficial for classification or regression tasks. Some features are irrelevant or redundant in distribution of dataset. These features can decrease the classification performance. In order to increase the classification performance and to reduce computation cost of classifier, the feature selection process should be used in classification or regression problems [1].

Linear discriminant analysis (LDA) [2-4] is one of the most popular linear projection techniques for feature extraction, it aims to maximize between-class scatter and minimize within-class scatter, thus maximize the class discriminant. But due to its limitation of linearly, LDA is difficult to capture nonlinear relationships with a linear mapping. To overcome this problem of LDA, the methods which based on the so-called kernel trick have been proposed over the last few years. The essence of kernel discriminant analysis (KDA) is to perform LDA in an implicit high-dimensional feature space[5-6]. Therefore, the naive KDA-based methods usually encounter two problems. One is the singularity problem caused by the under-sampled problems. We briefly present several KDA extensions which have been developed to deal with this problem, such as, KDA/GSVD [7], pseudo-inverse KDA (PIKDA) and null space KDA (NKDA). KDA/GSVD is one of generalizations of LDA based on kernel functions and GSVD, it overcomes the singularity of the scatter matrices by applying the GSVD to solve the generalized eigenvalue problem in the feature space. The traditional KDA solution is a special case of KDA/GSVD method. In PIKDA, the inverse of the kernel scatter matrix is replaced by the pseudo-inverse. In NKDA, the between-class distance is maximized in the null space of the within-class scatter matrix of the kernel matrix.

Another issue of KDA-based algorithms is the adverse effects due to outlier classes affect the performance of KDA-based methods. A promising solution to this problem is to introduce weighted schemes into the criteria. In this paper, we reformulate the KDA-based methods in the weighted forms, we call them weighted generalized KDA algorithms, we can see from [8-10] that the right weights in the weighting function appropriately to put more emphasis on those classes that are close together, hence are more likely to overcome the adverse effects. We present weighted versions of KDA/GSVD, PIKDA, NKDA and range space KDA with five weighting functions for each weighted scheme, where the K-Nearest neighbors (KNN) method [11] is used for a classifier. A weighting function is generally a monotonically decreasing function because classes that are closer to one another are likely to have a greater confusion and should be given a greater weightage. In this paper, we focus on study the effectiveness of the proposed algorithms and the effect of weights on the kernel functions and dimensional algorithms. From recent research in weighted methods, we can see an appropriate choice of the weight on the criterion plays a crucial role in the performance delivered. To further study the effect of weighting functions, we choice different kernel functions and three real-world data sets in our experiments. Extensive comparisons of different weighting functions on KDA methods are conducted.

The rest of the paper is organized as follows. In Section 2, we briefly review generalized KDA algorithms. Weighted versions of generalized KDA algorithms and weighting functions are introduced in Section 3. Extensive experiments with proposed algorithms have been performed in Section 4, the results demonstrate the effectiveness of the proposed algorithms and the effect of weighting functions and the effect of weights on kernel functions. Conclusion follows in Section 5.

2. Generalized KDA

Linear dimension reduction is conceptually simple and has been used in many application areas. However, it has a limitation for the data which is not linearly separable since it is difficult to capture a nonlinear relationship with a linear mapping. In order to overcome such a limitation, nonlinear extensions of linear dimension reduction methods using kernel methods have been proposed [7, 12-16]. The main idea of the kernel methods is that without knowing the nonlinear feature mapping or the mapped feature space explicitly, we can work on the nonlinearly transformed feature space through kernel functions. It is based on the fact that for any kernel function k satisfying Mercer's condition, there exists a mapping ϕ such that $\langle \phi(a), \phi(b) \rangle = k(a,b)$ where $\langle \cdot, \cdot \rangle$ is an inner product in the feature space transformed by ϕ. In this paper, we consider Gaussian RBF kernel, non-normal Gaussian RBF kernel and polynomial kernel [8]

$$k_{RBF} = \exp(-\|x-y\|^2 / \sigma^2), \sigma \in R,$$
$$k_{NN-RBF} = \exp(-\|x-y\|^d / \sigma^2), \sigma \in R, d \geq 0, d \neq 2,$$
$$k_{poly} = (r_1 x \cdot y^T - r_2)^d, d \in N, d \geq 1.$$

We apply the kernel method to perform LDA in the feature space instead of the original input space, namely kernel discriminant analysis (KDA). The basic idea of KDA is to firstly map the original data space into a feature space $\phi: R^m \to \mathcal{F}$, and then implement linear discriminant analysis in the feature space. Note that the feature space $\mathcal{F}$ could have a much higher, possibly infinite, dimensionality. However, by virtue of the kernel trick, the algorithm can be actually implemented in the input space [14].

Given a data matrix $X = [x_1, x_2, \cdots, x_N,]$ with r classes and N samples. Each class has N_i samples, and X_i represents the sample set of the i class. The between-class scatter matrix S_b^ϕ, within-class scatter matrix S_w^ϕ and total scatter matrix S_t^ϕ in $\mathcal{F}$ can be defined as

$$S_b^\phi = \sum_{i=1}^r N_i (m_i^\phi - m^\phi)(m_i^\phi - m^\phi)^T,$$
$$S_w^\phi = \sum_{i=1}^r \sum_{x_j \in X_i} (\phi(x_j) - m_i^\phi)(\phi(x_j) - m_i^\phi)^T,$$
$$S_t^\phi = S_b^\phi + S_w^\phi,$$

Where $m_i^\phi = (1/N_i) \sum_{x_j \in X_i} \phi(x_j)$ is the class mean of X_i in $\mathcal{F}$ and $m_\phi = (1/N) \sum_{i=1}^N \phi(x_i)$ is the global mean. The optimal projection directions for KDA can be obtained by maximizing the Fisher criterion in $\mathcal{F}$:

$$J(v) = \frac{v^T S_b^\phi v}{v^T S_w^\phi v}. \quad (1)$$

The optimal discriminant vectors with respect to the Fisher criterion are actually the eigenvectors of the following generalized equation

$$S_b^\phi v = \lambda S_w^\phi v. \quad (2)$$

We do not solve this optimization problem directly due to the high or even infinite dimensionality

of $\mathcal{F}$. Fortunately, we can show that the eigenvector v must lie in a space spanned by $\{\phi(x_i)\}_{i=1}^{N}$ in $\mathcal{F}$ and thus it can be expressed in the form of the following linear expansion :

$$v = \sum_{i=1}^{N} w_i \phi(x_i) \tag{3}$$

Substituting Eq. (3) into the numerator and denominator of Eq. (1), we obtain $v^T S_b^{\phi} v = \omega^T K_b \omega$ and $v^T S_w^{\phi} v = \omega^T K_w \omega$, then we can rewrite the Fisher criterion in Eq. (1) as:

$$J(\omega) = \frac{\omega^T K_b \omega}{\omega^T K_w \omega}, \tag{4}$$

Where $\omega = (\omega_1, \omega_2, \cdots, \omega_N)^T, K_b$ and K_w [8] can be seen as the variant scatter matrices based on some manipulation of the kernel matrix $K = [k(x_i, x_j)]_{N \times N}$, we can easy show that $K_t = K_b + K_w$. As a result, the solution to Eq. (1) can be the m leading eigenvectors $\omega_1, \omega_2, \cdots, \omega_m$ of the matrix $K_w^{-1} K_b$.

For any input vector x, its low-dimensional feature representation $y = (y_1, y_2, \cdots, y_m)^T$ can then be obtained as

$$y = (\omega_1, \cdots, \omega_m)^T (k(x_1, x), \cdots, k(x_N, x))^T.$$

Note that the solution above is based on the assumption that the within-class scatter matrix K_w is invertible. However, for many real-world applications, this assumption is almost always invalid due to the under-sampled problems as discussed above. In order to solve this problem, we can apply several generalized LDA algorithms, obtaining nonlinear discriminant analysis methods.

2.1 Nonlinear discriminant analysis based on kernel functions and GSVD

In this subsection, we review a nonlinear extension of LDA based on kernel functions and the GSVD.

Howland et al. [17] proposed LDA/GSVD, and it is one of generalizations of LDA based on GSVD, it overcomes the singularity of the scatter matrices by applying the GSVD to solve the generalized eigenvalue problem. And an efficient algorithm for LDA/GSVD was presented in [18].

Note that K_b, K_w and K_t are both singular and LDA cannot be applied, so in [7], a generalization of KDA based on the GSVD which we called KDA/GSVD was proposed. Similarly to the efficient algorithm for LDA/GSVD, we can present an efficient algorithm for KDA/GSVD as follows.

Algorithm 2.1. KDA/GSVD

(1) Compute the EVD of K_t: $K_t = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}$;

(2) Compute V from the EVD of $K_b = \Sigma_1^{-1/2} U_1^T K_b U_1 \Sigma_1^{-1/2} : K_b = V \Gamma_b^T \Gamma_b V^T$;

(3) Assign the first $r-1$ columns of $U_1 \Sigma_1^{-1/2} V$ to G_h;

(4) For any input vector x, its low-dimensional feature representation z can thus be given by $z = G_h^T (k(x_1, x), \cdots, k(x_N, x))^T$.

2.2 Pseudo-inverse KDA

As discussed the KDA, we know that K_b and K_w are both singular, the classical LDA cannot be applied, one simple method for solving this problem is to use the pseudo-inverse of K_w instead, let us call this method pseudo-inverse KDA (PIKDA).

In [19] proposed the pseudo-inverse LDA which the inverse of a scatter matrix is replaced by the pseudo-inverse, and we know that LDA/GSVD is a special case of pseudo-inverse LDA, so with Algorithm 2.1, we can derive PIKDA algorithm. The algorithm for the pseudo-inverse KDA can be given as follows.

Algorithm 2.2. Pseudo-inverse KDA

(1) Compute the EVD of K_t: $K_t = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}$;

(2) Compute V from the EVD of $K_b = \Sigma_1^{-1/2} U_1^T K_b U_1 \Sigma_1^{-1/2} : K_b = V \Gamma_b^T \Gamma_b V^T$;

(3) Assign the first $r-1$ columns of $U_1 \Sigma_1^{-1/2} V$ to X_δ;

(4) $G = X_\delta M$, where M is any nonsingular matrix;

(5) For any input vector x, its low-dimensional feature representation z can thus be given by $z = G^T (k(x_1, x), \cdots, k(x_N, x))^T$.

2.3 Nonlinear discriminant analysis based on the projection onto null (K_w)

Chen et al. [20] proposed the null space LDA (NLDA) for dimensionality reduction of under-sampled problems, in this subsection, we present a method which the kernel trick is incorporated into LDA in the null space of within-class scatter matrix, we call this method NKDA for short. Let the EVD of $K_w \in R^{N \times N}$ be

$$K_w = U_w \Sigma_w U_w^T \begin{bmatrix} U_{w1} & U_{w2} \end{bmatrix} \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix},$$

where $s_1 = \text{rank}(K_w), U_w$ is orthogonal, Σ_{w1} is a diagonal matrix with nonincreasing positive diagonal elements and U_{w1} contains the first s_1 columns of the orthogonal matrix U_w. We can easily show that $\text{null}(K_w) = \text{span}(U_{w2})$ and the transformation by $U_{w2} U_{w2}^T$ projects the data in the feature space $\mathcal{F}$ to $\text{null}(K_w)$. The between-class scatter matrix K_b in the transformed space is $K_b = U_{w2} U_{w2}^T K_b U_{w2} U_{w2}^T$. Consider the EVD of K_b:

$$K_b = U_b \Sigma_b U_b^T \begin{bmatrix} U_{b1} & U_{b2} \end{bmatrix} \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{b1}^T \\ U_{b2}^T \end{bmatrix},$$

where $s_2 = \text{rank}(K_b)$, $U_{b1} \in R^{N \times s_2}$ and $\Sigma_{b1} \in R^{s_2 \times s_2}$. In NKDA, the optimal transformation matrix G_e is obtained by $G_e = U_{w2} U_{w2}^T U_{b1}$. Hence, for any input vector x, its low-dimensional feature representation z can thus be given by $z = G_e^T (k(x_1, x), \cdots, k(x_N, x))^T$.

3. Weighted versions and weighting functions

Similar to the case of LDA-based methods, the adverse effects due to outlier classes also affect the performance of KDA-based algorithms. In fact, if the mapped data points in $\mathcal{F}$ are more separated from each other, such effects can become even more significant. To remedy this problem, a commonly method is to incorporate a weighting function into the Fisher criterion by using a weighted between-class scatter matrix in place of the ordinary between-class scatter matrix.

As in [21-23], we define the weighted between-class scatter matrix in $\mathcal{F}$ as follows:

$$S_B^\phi = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \frac{N_i N_j}{N} w(d_{ij}) (m_i^\phi - m_j^\phi)(m_i^\phi - m_j^\phi)^T, \tag{5}$$

where the weighting function $w(d_{ij})$ is a monotonically decreasing function of the Euclidean distance $d_{ij} = \left\| m_i^\phi - m_j^\phi \right\|$ with m_i^ϕ and m_j^ϕ being the class means for X_i and X_j in $\mathcal{F}$, respectively. Apparently, the weighted between-class scatter matrix S_B^ϕ degenerates to the conventional between-class scatter matrix S_b^ϕ if the weighting function in (5) gives a constant weight value. In addition, it is clear that d_{ij} in $\mathcal{F}$ can be calculated by the kernel trick as follows:

$$d_{il} = \left\| m_i^\phi - m_j^\phi \right\| = d_1^T d_2 = \sqrt{d_3 + d_4 - d_5 - d_6},$$

where

$$k_{i,j} = k(x_i, x_j) = \left\langle \phi(x_i), \phi(x_j) \right\rangle,$$

$$d_1 = \left(\sum_{x_{i_1} \in X_i} \frac{\phi(x_{i_1})}{N_i} - \sum_{x_{j_1} \in X_j} \frac{\phi(x_{j_1})}{N_j} \right),$$

$$d_2 = \left(\sum_{x_{i_2} \in X_i} \frac{\phi(x_{i_2})}{N_i} - \sum_{x_{j_2} \in X_j} \frac{\phi(x_{j_2})}{N_j} \right),$$

$$d_3 = \sum_{x_{i_1}, x_{i_2} \in X_i} \frac{k_{i_1, i_2}}{N_i^2},$$

$$d_4 = \sum_{x_{j_1}, x_{j_2} \in X_j} \frac{k_{j_1, j_2}}{N_j^2},$$

$$d_5 = \sum_{x_{i_1} \in X_i, x_{j_2} \in X_j} \frac{k_{i_1, j_2}}{N_i N_j},$$

$$d_6 = \sum_{x_{i_2} \in X_i, x_{j_1} \in X_j} \frac{k_{j_1, i_2}}{N_i N_j}.$$

Based on the definition of S_B^{ϕ} in Eq. (5), in $\mathcal{F}$ we have

$$S_B^{\phi} v = \lambda S_w^{\phi} v \tag{6}$$

By the theory of reproducing kernel, we can express the solution $v = \sum_{i=1}^{N} w_i \phi(x_i)$ and hence rewrite the generalized equation in Eq. (6) as [8]: $K_B v = \lambda K_w v$, where $\omega = (\omega_1, \omega_2, \cdots, \omega_N)^T$,

$$K_B = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \frac{N_i N_j}{N} w(d_{ij})(m_i - m_j)(m_i - m_j)^T,$$

$$K_w = \sum_{i=1}^{N} \sum_{x_j \in X_i} (k_j - m_i)(k_j - m_i)^T,$$

with

$$m_i = \left(\frac{1}{N_i} \sum_{h=1}^{N_i} k(x_1, x_h), \cdots, \frac{1}{N_i} \sum_{h=1}^{N_i} k(x_N, x_h)\right)^T,$$

$$m_j = \left(\frac{1}{N_j} \sum_{h=1}^{N_j} k(x_1, x_h), \cdots, \frac{1}{N_j} \sum_{h=1}^{N_j} k(x_N, x_h)\right)^T,$$

$$k_j = (k(x_1, x_j), \cdots, k(x_N, x_j))^T.$$

The solution to Eq. (6) is thus the m leading eigenvectors $\omega_1, \cdots, \omega_m$ of the matrix $K_w^{-1} K_B$.

For any input vector x, its low-dimensional feature representation $z = (z_1, \cdots, z_m)^T$ can then be obtained as $z = (\omega_1, \cdots, \omega_m)^T (k(x_1, x), \cdots, k(x_N, x))^T$.

If the between-class scatter matrix S_b^{ϕ} is replaced by the weighted between-class scatter

matrix S_B^{ϕ}, by means of the algorithms obtained in Section 2, we can get some weighted generalized KDA methods.

3.1 Weighted generalized KDA

With Algorithms2.1 and 2.2, we can derive weighted KDA/GSVD and weighted PIKDA algorithms.

Algorithm 3.1. Weighted KDA/GSVD

(1) Compute the EVD of K_t: $K_t = \begin{bmatrix} U_{t1} & U_{t2} \end{bmatrix} \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix}$;

(2) Compute the weighted between-class scatter matrix K_B;

(3) Compute V from the EVD of $K_B = \Sigma_{t1}^{-1/2} U_{t1}^T K_B U_{t1} \Sigma_{t1}^{-1/2} : K_B = V \Gamma_b^T \Gamma_b V^T$;

(4) Assign the first $r-1$ columns of $U_{t1}\Sigma_{t1}^{-1/2}V$ to G_h;

(5) For any input vector x, its low-dimensional feature representation z can thus be given by $z = G_h^T (k(x_1, x), \cdots, k(x_N, x))^T$.

Algorithm 3.2. Weighted PIKDA

(1) Compute the EVD of K_t: $K_t = \begin{bmatrix} U_{t1} & U_{t2} \end{bmatrix} \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix}$;

(2) Compute the weighted between-class scatter matrix K_B;

(3) Compute V from the EVD of $K_B = \Sigma_{t1}^{-1/2} U_{t1}^T K_B U_{t1} \Sigma_{t1}^{-1/2} : K_B = V \Gamma_b^T \Gamma_b V^T$;

(4) Assign the first $r-1$ columns of $U_{t1}\Sigma_{t1}^{-1/2}V$ to X_{δ};

(5) $G = X_{\delta} M$, where M is any nonsingular matrix;

(6) For any input vector x, its low-dimensional feature representation z can thus be given by $z = G^T (k(x_1, x), \cdots, k(x_N, x))^T$.

From Algorithms 3.1 and 3.2, we can see that weighted KDA/GSVD is a special case of weighted PIKDA with M being the identity matrix.

Similarly, we can obtain weighted NKDA.

Algorithm 3.3. Weighted NKDA

(1) Compute the EVD of K_w: $K_w = \begin{bmatrix} U_{w1} & U_{w2} \end{bmatrix} \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix}$;

(2) Compute the weighted between-class scatter matrix K_B;

(3) Compute the EVD of $K_B = U_{w2}U_{w2}^T K_B U_{w2}U_{w2}^T : K_B = \begin{bmatrix} U_{b1} & U_{b2} \end{bmatrix} \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{b1}^T \\ U_{b2}^T \end{bmatrix}$

(4) $G_e = U_{w2}U_{w2}^T U_{b1}$;

(5) For any input vector x, its low-dimensional feature representation z can thus be given by $z = G_e^T (k(x_1, x), \cdots, k(x_N, x))^T$.

3.2 Weighting functions

We can see from [9-10,24] that weighting functions have close relationships with classification accuracy. Different weighting functions can product different classification error for generalized KDA methods. Selecting suitable weighting function can increase classification accuracy. In this paper, we consider four weighted schemes Algorithms 3.1-3.4 with five weighting functions for each weighted scheme. We apply the Euclidean distance $d_{ij} = \left\| x_i - x_j \right\|$ between the means of X_i and X_j in weighting functions $w(d_{ij})$. A weighting function is generally a monotonically decreasing function because classes that are closer to one another are likely to have a greater confusion and should be given a greater weightage.

According to the FLDA procedure in [9], the weighting function should drop faster than the Euclidean distance between the class means for X_i and X_j in $\mathcal{F}$. We first apply two special cases of the weighting function $w(d_{ij}) = (d_{ij})^{-p}$ proposed by Lotlikar et al. [9] with $p = 1$ and $p = 2$, and then an improved version of weighting function $w(d_{ij}) = \frac{1}{2d_{ij}^2} erf(\frac{d_{ij}}{2\sqrt{2}})$ presented by Loog [10] where the Mahanalobis distance is replaced by the Euclidean distance. In addition, according to the feature of weighting functions mentioned above, we present two new weighting functions. They are listed below:

$$w_1 : w(d_{ij}) = (d_{ij})^{-2},$$

$$w_2 : w(d_{ij}) = \frac{1}{2d_{ij}^2} erf(\frac{d_{ij}}{2\sqrt{2}}),$$

$$w_3 : w(d_{ij}) = (d_{ij})^{-1},$$

$$w_4 : w(d_{ij}) = e^{\frac{1}{d_{ij}}},$$

$$w_5 : w(d_{ij}) = \frac{1}{e^{d_{ij}}}.$$

4. Experiments and analysis

In this section, in order to explain the effective of the proposed methods and illustrate the effect of the weights on kernel functions, we conduct a series of experiments on 3 different data sets from the UCI machine learning repository. The detail description of data sets is shown in Table 1. By randomly splitting the data to the training and test set of equal size and repeating it 10 times, 10 pairs of training and test sets were constructed for each data.

Table 1. The description of data sets

Data set	Classes	Dimension	Number
Dermatology	6	35	358
Wine	3	14	178
Irrs	3	5	150

For data set Dermatology, the Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|^2/10^2)$, non-normal Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|/10^2)$ and polynomial kernel $k_{poly}(x,y)=10^{-9}x^Ty+1$ are used, respectively. For data set Wine, the Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|^2)$, non-normal Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|)$ and polynomial kernel $k_{poly}(x,y)=x^Ty+0.51$ are used. For data set Irrs, the Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|^2/10^8)$, non-normal Gaussian RBF kernel $k(x,y)=\exp(-\|x-y\|/10^8)$ and polynomial kernel $k_{poly}(x,y)=x^Ty+0.51$ are used.

We randomly generate 5 matrices for M and compute the misclassification rates by using the optimal transformation matrices produced in PIKDA. In the following experiments, the KNN algorithm with $K=7$ is used as a classifier for all date sets. For each method, we repeated 10 times to obtain mean prediction misclassification rate. The experiment results are listed in Tables 2 - 4.

4.1 Effect of weighting functions

In this subsection, we study the effect of five weighting functions given in subsection 3.2 to classification accuracy on three data sets. Recalling that w_0 indicates no weighting function is introduced. Thus, from Tables 2 - 4, we see that the weighting scheme in the several weighted generalized KDA methods can change the performance, they also shows that weighted generalized KDA methods can successfully resist the adverse effects due to outlier classes and hence can improve the accuracy significantly if we choose the suitable weighting function.

It is interesting to note that the weighting function w_2 achieves higher accuracies than other weighting functions. In Table 2, the weighting function of w_4 on Gaussian RBF kernel produces the classification accuracy 3.8333% higher than weighting function w_0 for M_2, the weighting function of w_5 on Gaussian RBF kernel produces the classification accuracy 2.6667% lower than weighting function w_0 for M_3. In Table 3, the weighting function of w_1 on polynomial kernel produces the classification accuracy 1.7778% lower than w_0 for NKDA. In Table 4, the weighting function of w_2 on polynomial kernel produces the classification accuracy 5.2% higher than w_0 for M_2. Therefore, the weighting function can change the performances of those weighted generalized KDA methods, among the family of solutions to the generalized KDA, most of them perform quite poorly in comparison to the weighted generalized KDA.

Table 2. Misclassification rate (%) on data set Dermatology

Ker	$w(d_{ij})$	KDA/GSVD	PIKDA					NKDA
			M_1	M_2	M_3	M_4	M_5	
RBF	w_0	5.2222	8.8333	10.3333	7.8889	7.8889	7.2778	5.6667
	w_1	6.9444	7.0556	8.3889	7.0556	8.1111	7.6667	5.8333
	w_2	5.7778	6.1667	7.1667	7.0000	6.5000	7.9444	5.6111
	w_3	5.4444	7.0000	8.1667	7.7222	9.1667	7.5000	5.6667
	w_4	5.7222	6.8889	6.5000	6.2222	7.6111	6.2222	5.7222
	w_5	5.9444	6.7222	8.1667	10.556	7.3889	7.1667	5.8333
NN-RBF	w_0	2.7222	2.4444	2.7778	2.7778	2.6111	2.6111	2.5556
	w_1	2.7778	2.7778	2.5556	2.6111	2.4444	2.6667	2.6111
	w_2	2.7222	2.7778	2.6111	2.6667	2.5556	2.3889	2.5556
	w_3	2.4444	2.7778	2.6111	2.5556	2.6111	2.9444	2.5556
	w_4	-	-	-	-	-	-	-
	w_5	2.8333	2.7222	2.9444	2.7222	2.5000	2.7778	2.5556
Poly	w_0	4.5000	4.3889	5.1667	4.2778	4.8889	4.2778	14.1667
	w_1	4.2778	4.3889	4.1667	4.6111	4.4444	4.8333	14.2778
	w_2	4.3333	4.6111	5.1667	4.7778	4.9444	5.4444	13.1111
	w_3	4.3333	4.3333	4.9444	4.6111	5.3333	5.3333	13.7778
	w_4	-	-	-	-	-	-	-
	w_5	4.5000	4.3889	5.1667	4.2778	4.8889	4.2778	14.6111

Table 3. Misclassification rate (%) on data set Wine

Ker	$w(d_{ij})$	KDA/GSVD	PIKDA					NKDA
			M_1	M_2	M_3	M_4	M_5	
RBF	w_0	2.8889	2.3333	2.4444	2.8889	2.4444	2.3333	2.4444
	w_1	2.6667	3.1111	2.4444	2.6667	2.4444	2.4444	2.4444
	w_2	2.7778	2.2222	2.5556	2.6667	2.6667	2.3333	2.4444
	w_3	2.5556	3.0000	2.4444	2.8889	2.4444	2.3333	2.4444
	w_4	2.5556	2.7778	2.5556	2.4444	2.5556	2.4444	2.3333
	w_5	2.5556	2.1111	2.3333	2.7778	2.6667	2.3333	2.4444
NN-RBF	w_0	1.8889	1.8889	1.8889	2.0000	1.8889	2.1111	1.7778
	w_1	1.7778	1.8889	2.0000	1.8889	2.0000	1.8889	1.7778
	w_2	1.7778	1.7778	1.8889	2.0000	1.6667	1.8889	1.8889
	w_3	1.8889	1.7778	2.0000	1.8889	1.7778	1.8889	1.8889
	w_4	1.6667	1.6667	1.8889	2.0000	1.8889	1.8889	1.7778
	w_5	1.7778	1.7778	2.0000	1.8889	2.0000	1.7778	1.8889
Poly	w_0	3.3333	4.5556	4.6667	4.6667	4.6667	4.7778	6.2222
	w_1	3.1111	3.4444	4.6667	5.1111	4.7778	4.3333	8.0000
	w_2	3.5556	3.7778	4.4444	4.4444	4.4444	4.6667	7.5556
	w_3	3.5556	3.7778	4.5556	4.8889	4.5556	4.3333	7.6667
	w_4	3.5556	3.6667	4.6667	4.5556	4.5556	4.5556	7.4444
	w_5	3.3333	3.6667	4.4444	5.1111	4.5556	4.3333	7.5556

From Tables 2 - 4, we observe that they have a similar trend, that is, most of the accuracies of the generalized KDA are much lower than those of weighted generalized KDA algorithms.

4.2 Effect of weights on kernel functions

In this experiment, we study the effect of five weighting functions on three kernel functions with three data sets. We can see that the performances of the weighted generalized KDA on the different kernel functions for the five weighting functions show a similar trend, as we can see in Table 2-4, the effects of weighting functions on kernel functions are big.

Table 4. Misclassification rate (%) on data set Irrs

Ker	$w(d_{ij})$	KDA/GSVD	PIKDA					NKDA
			M_1	M_2	M_3	M_4	M_5	
RBF	w_0	3.4667	3.0667	4.2667	4.5333	3.2000	4.4000	14.9333
	w_1	3.6000	3.0667	3.8667	4.8000	3.4667	4.5333	14.0000
	w_2	3.4667	3.0667	4.1333	4.4000	3.3333	4.5333	13.0667
	w_3	3.4667	3.0667	4.1333	4.4000	3.3333	4.5333	14.4000
	w_4	-	-	-	-	-	-	-
	w_5	3.4667	3.0667	4.2667	4.5333		4.4000	13.0667
NN-RBF	w_0	4.8000	4.8000	4.9333	4.6667	4.6667	4.9333	4.6667
	w_1	4.9333	4.8000	4.6667	4.6667	4.6667	4.6667	4.6667
	w_2	4.9333	4.6667	4.6667	4.6667	4.6667	4.8000	4.6667
	w_3	4.9333	4.9333	4.6667	4.8000	4.8000	4.6667	4.6667
	w_4	-	-	-	-	-	-	-
	w_5	4.6667	4.6667	4.9333	4.8000	4.8000	4.9333	4.6667
Poly	w_0	7.8667	8.2667	9.7333	10.400	8.0000	10.933	5.2000
	w_1	7.4667	7.8667	6.5333	7.6000	6.1333	8.5333	5.2000
	w_2	7.8667	7.4667	4.5333	7.2000	5.3333	7.0667	5.6000
	w_3	7.7333	7.7333	7.6000	9.2000	6.4000	8.8000	5.0667
	w_4	7.8667	8.4000	9.7333	10.933	8.6667	11.333	5.2000
	w_5	7.8667	7.2000	5.4667	7.6000	5.4667	7.7333	5.4667

In the following experiments, we first compare the different weighting functions using the same kernel. From the experimental results, we can see the weighting function w_2 produces good overall results on Gaussian RBF kernel, it is better than other weighting functions, especially, we get the classification accuracy 1.8666% higher than that of w_0 for NKDA and Irrs data set; the weighting function w_4 cannot effectively boost the performance of these weighting generalized KDA methods, it can't be applied for Irrs data set, but it produces the classification accuracy 3.8333% lower than that of w_0 for M_2 and Dermatology data set. For the non-normal Gaussian RBF kernel, the five weighting functions produce similar overall results, they can improve the performance of the weighting generalized methods but the effects are not obvious, moreover, the weighting functions w_4 almost can't be applied but it produces the best classification accuracy 98.3333% for M_1 and Wine data set. The weighting function w_2 produces good overall results on the polynomial kernel, it is the best one among the five weighting functions, we get the classification accuracy 5.2% higher than that of w_0 for M_2 and Irrs data set, w_1 produces the

worst classification accuracy 1.7778% lower than w_0 for NKDA and Wine data set.

We next compare the same weighting function on the three different kernels. Table 2-4 show that the weighting function w_2 produces good overall results on the five weighting functions, it's better than other weighting functions, especially, for Irrs data set and M_2, the misclassification rate is 5.2% lower than w_0 on the polynomial kernel. Expect for a few cases, the weighting functions w_1, w_3 and w_5 produce the results are near, but the performances cannot effectively improve. For the weighting function w_4, it almost can't be applied, although it can be applied to other two kernel functions, it can't improve the accuracy significantly.

In general, we can see the results that the weighted schemes in weighted generalized KDA can bring about performance improvement over generalized KDA on different kernel functions.

5. Conclusion

In this paper, based on pseudo-inverse KDA, KDA/GSVD, null space KDA and range space KDA, we propose weighted pseudo-inverse KDA, weighted KDA/GSVD, weighted null space KDA and weighted range space KDA. Not only can these methods deal with the singularity problem caused by the under-sampled problems, they can also resist the adverse effects due to outlier classes. In order to explain the effect of the weighting functions and illustrate the effect of weighting functions on kernel functions, we conduct a series of experiments on 3 different data sets from the UCI Machine Learning Repository. Results show that different kernel functions and different weighting functions affect the classification accuracy of the proposed methods. Also, the weighting functions can affect the kernel functions on the classification results, some can increase the classification accuracy 3% - 5.2%, some can decrease 1.7778% - 2.6667%. And the misclassification rates produced by each of the weighted dimension reduction methods for three kernel function on the five weighting functions are different. Hence, for the different kernel functions in the weighted generalized KDA we must choose different weighting functions.

References

[1]. Cao, Shen, Sun, Yang, Chen (2007), "Feature selection in a kernel space", *In International conference on machine learning (ICML) Oregon*, USA, June 20-24, pp. 121-128.

[2]. Zhang S., Zhao X., Lei B. (2012), "Facial Expression Recognition Based on Local Binary Patterns and Local Fisher Discriminant Analysis", *WSEAS transactions on signal processing*, 8:21-31.

[3]. Yang J., Fan L.Y., Sun Q.S. (2014), "Weighted Generalized LDA for Undersampled Problems", *WSEAS transactions on mathematics*, 13: 694-703.

[4]. Wang X.Y., Hu H.F., Gu J.Q. (2016), "Pose Robust Low-resolution Face Recognition via Coupled Kernel-based Enhanced Discriminant Analysis", *IEEE/CAA Journal of Automatica Sinica*, 3(2): 203-212.

[5]. Yang J., Jin Z., Yang J., Zhang D. (2004), "The essence of kernel Fisher discriminant: KPCA plus LDA", *Pattern Recognition*, 37: 2097-2100.

[6]. Yang J., Frangi A.F., Yang J.Y., Zhang D. (2005), "KPCA Plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, 27: 230-244.

[7]. Park C.H., Park H. (2005), "Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition", *SIAM J. Matrix Anal. Appl.*, 27(1): 87-102.

[8]. Dai G., Yeung D.Y., Qian Y.T. (2007), "Face recognition using a kernel fractional-step discriminant analysis algorithm", *Pattern recognition*, 40: 229-243.

[9]. Lotlikar R., Kothari R. (2000), "Fractional-step dimensionality reduction", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(6): 623-627.

[10]. Loog M., Duin R.P.W., Haeb-Umbach R. (2001), "Multiclass linear dimension reduction by weighted pairwise Fisher criteria", *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(7): 762-766

[11]. Duda R.O., Hart P.E., Stork D. (2000), *Pattern Classification*, Wiley, 2000.

[12]. Yang J., Fan L. (2011), "Weighted Generalized Kernel Discriminant Analysis Using Fuzzy memberships", *WSEAS transactions on mathematics*, 10: 346-357.

[13]. Bouveyron C., Fauvel M., Girard S. (2015), "Kernel discriminant analysis and clustering with parsimonious Gaussian process models", *Statistics and Computing*, 25(6): 1143-1162.

[14]. Baudat G., Anouar F. (2000), "Generalized discriminant analysis using a kernel approach", *Neural Comput.*, 12: 2385-2404.

[15]. Roth V., Steinhage V. (2000), "Nonlinear discriminant analysis using kernel functions", *Adv. Neural Inf. Process. Systems*, 12: 568-574.

[16]. Billings S.A., Lee K.L. (2002), "Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm", *Neural Networks*, 15: 263-270.

[17]. Howland P., Jeon M., Park H. (2003), "Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition", *SIAM J. Matrix Anal. Appl.*, 25(1): 165-179.

[18]. Park C.H., Park H. (2008), "A comparison of generalized linear discriminant analysis algorithms", *Pattern Recognition*, 41: 1083-1097.

[19]. Ye, Janardan R., Park C.H., Park H. (2004), "An optimization criterion for generalized discriminant analysis on undersampled problems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8): 982-994.

[20]. Chen L., Liao H.M., Ko M.,Lin J., Yu G. (2000), "A new LDA-based face recognition system which can solve the small sample size problem", *Pattern Recognition*, 33: 1713-1726.

[21]. Li Y.X., Gao Y.Q., Erdogan H. (2000), "Weighted pairwise scatter to improve linear discriminant analysis", *Proceedings of the sixth International Conference on Spoken Language Processing*, 2000.

[22]. Dai G., Qian Y.T., Jia S. (2004), "A kernel fractional-step nonlinear discriminant analysis for pattern recognition", *Proceedings of the 18th International Conference on Pattern Recognition*, 2: 431-434.

[23]. Dai G., Yeung D.Y. (2005), Nonlinear dimensionality reduction for classification using kernel weighted subspace method, *Proceedings of the IEEE International Conference on Image Processing*, September 2005, pp. 838-841.

[24]. Liang Y. et al. (2007), "Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion", *Pattern Recognition*, 40: 3606-3615.