Learning the Kernel Matrix by Maximizing a KFD-Based Class Separability Criterion

Dit-Yan Yeung, Hong Chang & Guang Dai
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{dyyeung,hongch,daiguang}@cse.ust.hk

September 22, 2006

Abstract

The advantage of a kernel method often depends critically on a proper choice of the kernel function. A promising approach is to learn the kernel from data automatically. In this paper, we propose a novel method for learning the kernel matrix based on maximizing a class separability criterion that is similar to those used by linear discriminant analysis (LDA) and kernel Fisher discriminant (KFD). It is interesting to note that optimizing this criterion function does not require inverting the possibly singular within-class scatter matrix which is a computational problem encountered by many LDA and KFD methods. We have conducted experiments on both synthetic data and real-world data from UCI and FERET, showing that our method consistently outperforms some previous kernel learning methods.

Keywords: kernel learning, Fisher discriminant criterion, kernel Fisher discriminant, face recognition.

1 Introduction

Kernel methods [21] provide a disciplined approach to the nonlinear generalization of many linear methods. Support vector machine (SVM), kernel principal component analysis (KPCA) and kernel Fisher discriminant (KFD) are just some of the better known kernel methods. However, the advantage of a kernel method often depends critically on a proper choice of the kernel function. Over the past few years, some methods have been proposed to learn the kernel from data automatically. Early work on kernel learning is limited to learning the parameters of some prespecified kernel function form, e.g., [5]. More recent work has gone beyond kernel parameter learning by learning the kernel itself in a more nonparametric manner. In practice, since we work with data sets of finite size, we can learn the kernel matrix corresponding to a given data set instead of learning the kernel function. Different kernel matrix learning methods have been proposed. These include performing classical optimization based on kernel alignment [7], semi-definite programming (SDP) based on alignment or margin [10], faster methods such as gradient descent [4] and quadratically constrained quadratic programming (QCQP) [1, 29] for alignment-based or margin-based optimization, boosting based on exponential loss or logarithmic loss [6], the information-geometric *em* method based on Kullback-Leibler (KL) divergence [24], Markov chain Monte Carlo (MCMC) [28] and expectation-maximization (EM) [22] based on likelihood, and matrix exponentiated gradient update and Bregmann projection based on von Neumann divergence [25]. Using the mathematical programming reformulation of KFD by [15], a quadratic programming approach was proposed by [9] to learn a linear combination of kernels for KFD. Some methods perform optimization over the conic structure of the space of kernels [1, 2, 11, 14, 17, 18]. Most of these methods are for classification, but clustering [24] and regression [22] have also been studied.

Inspired by a recent kernel parameter learning method [27], we propose in this paper a novel kernel matrix learning method based on optimizing a class separability criterion that is similar to those used by linear discriminant analysis (LDA) [8, 20] and KFD [3, 16].

In LDA, finding the optimal linear transformation corresponds to solving a generalized eigenvalue problem, which requires that the pooled within-class scatter matrix be invertible. While this is generally not a problem for large-sample applications, the problem does arise in applications when the dimensionality of the input space is larger than the sample size, such as in face recognition and microarray data analysis applications. Unfortunately, this singularity problem is usually more severe for KFD as it essentially performs LDA in the kernel-induced feature space which is of very high or even infinite dimensionality. To address this problem, we do not attempt to solve a generalized eigenvalue problem. This avoids the need for inverting the within-class scatter matrix which may be singular. Instead, we formulate a different optimization criterion which can give rise to a closed-form solution for the optimization problem without requiring to solve a generalized eigenvalue problem. It turns out that this optimization criterion is related to the *maximum margin criterion* (MMC) [12] proposed recently for LDA and KFD methods. As a result, not only is our method more general than the method of [27] in that it learns the kernel matrix rather than just the parameters of a prespecified kernel function, but our optimization method which leads to a closed-form solution is also more appealing than the gradient method in [27] which requires tuning many parameters in the algorithm.

2 Our Kernel Learning Method

Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ be a training set of l labeled examples and $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$ be a test set of $n-l$ unlabeled examples, where $\mathbf{x}_i$ $(i = 1, \ldots, n)$ are n points in the input space $\mathcal{X}$ and the class labels y_i $(i = 1, \ldots, l)$ are from $\{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$ with c being the number of classes. The objective of the classification problem is to predict the class labels of the unlabeled examples in the test set. We consider the classification problem under the transductive learning setting [26] in which the test set is given in advance before the classifier is learned.

2.1 Spectral Variants of Kernel Matrix

Let $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n}$ denote the kernel matrix formed by the n data points in $\mathcal{X}$ for some chosen kernel function $k(\cdot,\cdot)$, such as the RBF kernel or polynomial kernel. We express the spectral decomposition of $\mathbf{K}$ as

$$\mathbf{K} = \sum_{r=1}^{p} \lambda_r \mathbf{v}_r \mathbf{v}_r^T = \sum_{r=1}^{p} \lambda_r \mathbf{K}_r, \tag{1}$$

where $\lambda_r\,(r = 1,\ldots,p)$ are the p positive eigenvalues of $\mathbf{K}$ sorted in a monotonically decreasing order and $\mathbf{v}_r\,(r = 1,\ldots,p)$ are the corresponding normalized eigenvectors.[1] $\mathbf{K}_r\,(r = 1,\ldots,p)$ are base kernel matrices of rank one. Based on these rank-one base kernel matrices, we define a parameterized family of kernel matrices as

$$\mathbf{K}_{\boldsymbol{\mu}} = \sum_{r=1}^{p} \mu_r^2 \mathbf{v}_r \mathbf{v}_r^T = \sum_{r=1}^{p} \mu_r^2 \mathbf{K}_r, \tag{2}$$

where $\boldsymbol{\mu} = (\mu_1,\ldots,\mu_p)^T$ denotes p coefficients for specifying the spectral variants. This method has also been used in some previous kernel learning work [4, 7, 10, 24, 28]. It is trivial to show that each kernel matrix in this family corresponds to a Mercer kernel [21].

2.2 Class Separability Criterion for Optimization

Similar to common KFD methods, our class separability criterion is based on the objective of maximizing the inter-class variability in the feature space while minimizing the intra-class variability. Let $\phi(\mathbf{x}_i)\,(i = 1,\ldots,n)$ be the n points in the feature space induced by kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$, l_i be the number of training data points that belong to class i, $\mathbf{m}_i = \frac{1}{l_i}\sum_{y_j=\mathcal{C}_i}\phi(\mathbf{x}_j)$ be the mean vector of class i in the feature space, and $\mathbf{m} = \frac{1}{l}\sum_{i=1}^{c} l_i\mathbf{m}_i = \frac{1}{l}\sum_{i=1}^{c}\sum_{y_j=\mathcal{C}_i}\phi(\mathbf{x}_j)$ be the mean vector of all l training data points. The between-class

[1] Instead of expressing $\mathbf{K}$ in terms of all its positive eigenvalues and the corresponding eigenvectors, one may discard the very small eigenvalues as in PCA. In that case, $\mathbf{K} \simeq \sum_{r=1}^{p}\lambda_r\mathbf{v}_r\mathbf{v}_r^T = \sum_{r=1}^{p}\lambda_r\mathbf{K}_r$.

scatter matrix $\mathbf{S}_b$ and within-class scatter matrix $\mathbf{S}_w$ in the feature space are given by

$$\mathbf{S}_b = \frac{1}{l}\sum_{i=1}^{c} l_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \tag{3}$$

$$\mathbf{S}_w = \frac{1}{l}\sum_{i=1}^{c}\sum_{y_j=\mathcal{C}_i} (\phi(\mathbf{x}_j) - \mathbf{m}_i)(\phi(\mathbf{x}_j) - \mathbf{m}_i)^T. \tag{4}$$

If the feature space is infinite-dimensional, we can define scatter operators instead. KFD methods typically maximize the class separability through maximizing the Fisher criterion $(\mathbf{w}^T\mathbf{S}_b\mathbf{w})/(\mathbf{w}^T\mathbf{S}_w\mathbf{w})$ to find the optimal linear transformation $\mathbf{w}$. This can be achieved by solving the generalized eigenvalue problem $\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w}$ for the eigenvectors $\mathbf{w}$ corresponding to the largest eigenvalues. However, this method requires inverting $\mathbf{S}_w$.

Here, we use the trace of a scatter matrix to quantify its scatter. Let $\mathrm{Tr}(\cdot)$ denote the trace of a symmetric matrix. We use $\mathrm{Tr}(\mathbf{S}_b)$ and $\mathrm{Tr}(\mathbf{S}_w)$ to characterize the inter-class variability and intra-class variability, respectively. Note that we do not use $\mathrm{Tr}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})$ and $\mathrm{Tr}(\mathbf{W}^T\mathbf{S}_w\mathbf{W})$, for some linear transformation matrix $\mathbf{W}$, because we do not actually find a low-dimensional embedding of the points. We can make use of the kernel trick to express $\mathrm{Tr}(\mathbf{S}_b)$ and $\mathrm{Tr}(\mathbf{S}_w)$ in terms of $\boldsymbol{\mu}$ and the matrix entries of $\mathbf{K}$ without requiring the nonlinear mapping $\phi(\cdot)$ explicitly. Let the (i,j)th entry of $\mathbf{K}_{\boldsymbol{\mu}}$ be expressed as $(\mathbf{K}_{\boldsymbol{\mu}})_{ij} = \mathbf{b}_i^T\mathbf{K}_{\boldsymbol{\mu}}\mathbf{b}_j$, where $\mathbf{b}_i$ is the ith column of the $n \times n$ identity matrix. We can rewrite $\mathrm{Tr}(\mathbf{S}_b)$

as

$$\begin{aligned}
\mathrm{Tr}(\mathbf{S}_b) &= \frac{1}{l}\sum_{i=1}^{c} l_i(\mathbf{m}_i-\mathbf{m})^T(\mathbf{m}_i-\mathbf{m}) \\
&= \frac{1}{l}\sum_{i=1}^{c} l_i\mathbf{m}_i^T\mathbf{m}_i - \mathbf{m}^T\mathbf{m} \\
&= \frac{1}{l}\left[\sum_{i=1}^{c}\sum_{y_j,y_k=\mathcal{C}_i}\frac{1}{l_i}(\mathbf{K}_{\boldsymbol{\mu}})_{jk} - \sum_{j,k=1}^{l}\frac{1}{l}(\mathbf{K}_{\boldsymbol{\mu}})_{jk}\right] \\
&= \frac{1}{l}\sum_{j,k=1}^{l}\left(\sum_{i=1}^{c}a_{jk}^i - \frac{1}{l}\right)\mathbf{b}_j^T\mathbf{K}_{\boldsymbol{\mu}}\mathbf{b}_k \\
&= \sum_{r=1}^{p}\mu_r^2\left[\frac{1}{l}\sum_{j,k=1}^{l}\left(\sum_{i=1}^{c}a_{jk}^i - \frac{1}{l}\right)\mathbf{b}_j^T\mathbf{K}_r\mathbf{b}_k\right] \\
&= \sum_{r=1}^{p}\mu_r^2 f_r \\
&= \boldsymbol{\mu}^T\mathbf{D}_b\boldsymbol{\mu}, \qquad (5)
\end{aligned}$$

where

$$a_{jk}^i = \begin{cases} \dfrac{1}{l_i} & y_j = y_k = \mathcal{C}_i \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

$$f_r = \frac{1}{l}\sum_{j,k=1}^{l}\left(\sum_{i=1}^{c}a_{jk}^i - \frac{1}{l}\right)\mathbf{b}_j^T\mathbf{K}_r\mathbf{b}_k = \frac{1}{l}\sum_{j,k=1}^{l}\left(\sum_{i=1}^{c}a_{jk}^i - \frac{1}{l}\right)\mathbf{b}_j^T\mathbf{v}_r\mathbf{v}_r^T\mathbf{b}_k \qquad (7)$$

$$\mathbf{D}_b = \mathrm{diag}(f_1,\ldots,f_p). \qquad (8)$$

Similarly, we can rewrite $\mathrm{Tr}(\mathbf{S}_w)$ as

$$\begin{aligned}
\mathrm{Tr}(\mathbf{S}_w) &= \frac{1}{l}\sum_{i=1}^{c}\sum_{y_j=\mathcal{C}_i}(\phi(\mathbf{x}_j)-\mathbf{m}_i)^T(\phi(\mathbf{x}_j)-\mathbf{m}_i) \\
&= \frac{1}{l}\sum_{i=1}^{c}\left(\sum_{y_j=\mathcal{C}_i}\phi(\mathbf{x}_j)^T\phi(\mathbf{x}_j)-l_i\mathbf{m}_i^T\mathbf{m}_i\right) \\
&= \frac{1}{l}\left[\sum_{j=1}^{l}(\mathbf{K}_{\boldsymbol{\mu}})_{jj}-\sum_{i=1}^{c}\sum_{y_j,y_k=\mathcal{C}_i}\frac{1}{l_i}(\mathbf{K}_{\boldsymbol{\mu}})_{jk}\right] \\
&= \frac{1}{l}\sum_{j,k=1}^{l}\left(\delta_{jk}-\sum_{i=1}^{c}a_{jk}^i\right)\mathbf{b}_j^T\mathbf{K}_{\boldsymbol{\mu}}\mathbf{b}_k \\
&= \sum_{r=1}^{p}\mu_r^2\left[\frac{1}{l}\sum_{j,k=1}^{l}\left(\delta_{jk}-\sum_{i=1}^{c}a_{jk}^i\right)\mathbf{b}_j^T\mathbf{K}_r\mathbf{b}_k\right] \\
&= \sum_{r=1}^{p}\mu_r^2 g_r \\
&= \boldsymbol{\mu}^T\mathbf{D}_w\boldsymbol{\mu}, \qquad (9)
\end{aligned}$$

where δ_{jk} is the Kronecker delta and

$$g_r = \frac{1}{l}\sum_{j,k=1}^{l}\left(\delta_{jk}-\sum_{i=1}^{c}a_{jk}^i\right)\mathbf{b}_j^T\mathbf{K}_r\mathbf{b}_k = \frac{1}{l}\sum_{j,k=1}^{l}\left(\delta_{jk}-\sum_{i=1}^{c}a_{jk}^i\right)\mathbf{b}_j^T\mathbf{v}_r\mathbf{v}_r^T\mathbf{b}_k \qquad (10)$$

$$\mathbf{D}_w = \mathrm{diag}(g_1,\ldots,g_p). \qquad (11)$$

One possible optimality criterion for maximization is $\mathrm{Tr}(\mathbf{S}_b)/\mathrm{Tr}(\mathbf{S}_w)$, which is a form of generalized Rayleigh quotient. This seems to argue for solving it as a generalized eigenvalue problem $\mathbf{D}_b\boldsymbol{\mu} = \lambda\mathbf{D}_w\boldsymbol{\mu}$. However, since $\mathbf{D}_w^{-1}\mathbf{D}_b$ is diagonal and its diagonal entries are generally different, the eigenvectors are just vectors with all except one entry equal to 0 and the eigenvalues are the diagonal entries of $\mathbf{D}_w^{-1}\mathbf{D}_b$. The best solution is thus the eigenvector corresponding to the largest eigenvalue. This implies that the spectral variant solution degenerates to having only one base kernel. Apparently this is not what we want.

An alternative approach is to regard the maximization of $\mathrm{Tr}(\mathbf{S}_b)/\mathrm{Tr}(\mathbf{S}_w)$ as a nonlinear fractional programming (FP) problem [23]. Inspired by the parametric methods for solving such FP problems, we define the following class separability criterion function:

$$Q(\boldsymbol{\mu}) = \mathrm{Tr}(\mathbf{S}_b) - \alpha \mathrm{Tr}(\mathbf{S}_w) = \boldsymbol{\mu}^T(\mathbf{D}_b - \alpha \mathbf{D}_w)\boldsymbol{\mu}, \tag{12}$$

where $\alpha > 0$ is a parameter that can be determined, for example, by cross-validation. Note that $Q(\cdot)$ is a function of the parameter vector $\boldsymbol{\mu}$ in the parameterized family of kernel matrices.

2.3 Solving the Optimization Problem

When we maximize the criterion function in (12), we eliminate the scaling factor by enforcing the linear equality constraint $\mathbf{1}^T\boldsymbol{\mu} = c$, where $\mathbf{1}$ is a column vector of ones and c is some positive constant. The optimization problem can be solved using the method of Lagrange multipliers with the Lagrangian

$$\hat{Q}(\boldsymbol{\mu}, \rho) = Q(\boldsymbol{\mu}) + \rho(c - \mathbf{1}^T\boldsymbol{\mu}). \tag{13}$$

Differentiating $\hat{Q}(\boldsymbol{\mu}, \rho)$ with respect to $\boldsymbol{\mu}$ and ρ gives the following partial derivatives:

$$\frac{\partial \hat{Q}}{\partial \boldsymbol{\mu}} = 2(\mathbf{D}_b - \alpha \mathbf{D}_w)\boldsymbol{\mu} - \rho \mathbf{1} \tag{14}$$

$$\frac{\partial \hat{Q}}{\partial \rho} = c - \mathbf{1}^T\boldsymbol{\mu}. \tag{15}$$

Setting the partial derivatives to zero, the optimal value of $\boldsymbol{\mu}$ is given by

$$\boldsymbol{\mu} = \frac{c(\mathbf{D}_b - \alpha \mathbf{D}_w)^{-1}\mathbf{1}}{\mathbf{1}^T(\mathbf{D}_b - \alpha \mathbf{D}_w)^{-1}\mathbf{1}}. \tag{16}$$

Note that $(\mathbf{D}_b - \alpha \mathbf{D}_w)$ is a diagonal matrix which is invertible if no diagonal entries are zero. We set the constant c to $\sum_{r=1}^{p} \sqrt{\lambda_r}$.

The learned kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$ can then be used with any kernel-based classification method for classifying the unlabeled examples in the test set.

3 Experiments

In this section, we present experimental results on several classification problems to compare our kernel matrix learning method with some previous methods.

3.1 Experimental Setting for Comparative Study

We compare our method with a kernel matrix learning method based on kernel alignment [7] and a kernel parameter learning method proposed recently by Xiong et al. [27].[2] Xiong et al.'s method is similar to ours in that it also uses the between-class scatter matrix and within-class scatter matrix to define the class separability criterion. The difference, however, is that it learns the parameters of a prespecified kernel function rather than the kernel matrix itself. Standard RBF kernel serves for baseline comparison. Thus, our comparative study consists of the following four kernel methods (with their short forms shown inside brackets for subsequent use): (1) standard RBF kernel (RBF); (2) kernel matrix learning based on alignment (Alignment); (3) kernel parameter learning by Xiong et al. (Xiong's); and (4) our kernel matrix learning method (Ours).

We use RBF kernel, $k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/\sigma^2)$, to define the initial kernel matrix for different kernel learning methods. The parameter σ^2 is selected from $\{\bar{d^2}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 10^2, 10^3\}$ using leave-one-out (LOO) method, where $\bar{d^2}$ represents the mean squared Euclidean distance between all data points in the input space. For Xiong et al.'s method, the kernel optimization problem is solved in an iterative manner. As reported in [27], the initial learning rate and the number of iterations are set to 0.01 and 200, respectively, for all data sets. For our kernel learning method, the parameter α is set to

[2] We do not include another KFD-based kernel learning method by Fung et al. [9] in our comparative study, for two reasons. First, their method only works for two classes. Like SVM, extension to multiple classes is nontrivial. Second, the kernels for forming the linear combination and their parameters have to be chosen manually in advance. As a result, the degree of automation is not as high as desired.

10000 in the experiments. In the following subsections, we report extensive experiments on a toy problem (for illustration purpose), UCI data sets, and face recognition. More details about each task will be provided later.

3.2 A Toy Problem

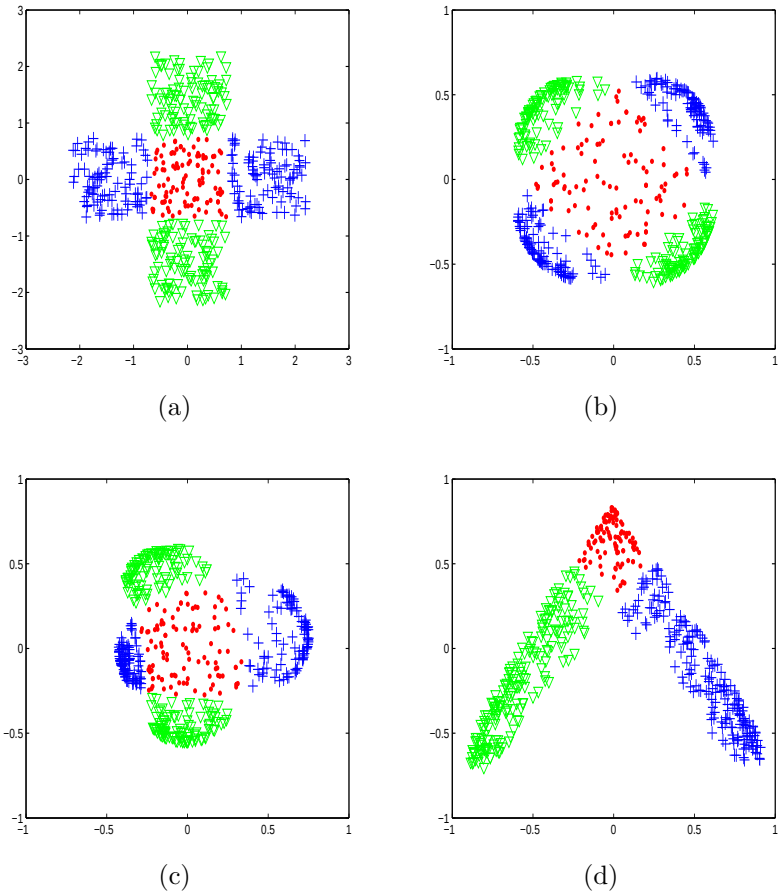

Figure 1: Comparison of different kernel learning methods on a toy data set. (a) original data set with three classes; and the embedded data set based on (b) RBF; (c) Xiong's; (d) Ours.

We first perform some experiments on a 2-dimensional toy data set, as shown in Figure 1(a). There are 500 points in this data set with three classes. Two classes have 200

points each corresponding to the up-down and left-right natural groups, while the third class has 100 points corresponding to the center group. Data points shown with the same point style and color belong to the same class. We randomly select 10% of the data points from each class to form the training set and then perform kernel learning using Xiong et al.'s and our methods. For the sake of visualization, we apply kernel PCA based on the initial RBF kernel and the learned kernel matrices to embed the points onto a 2-dimensional space, as shown in Figure 1(b)–(d). It can be seen that neither the RBF kernel nor the kernel learned by Xiong et al.'s method can give satisfactory result. On the other hand, our method can group the data points properly according to their class labels.

3.3 UCI Data Sets

We next perform some classification experiments on five data sets from the UCI Machine Learning Repository:[3] *Monks-3* (432/6/2), *Ionosphere* (351/34/2), *Breast cancer* (569/31/2), *Wine* (178/13/3), and *Boston housing* (506/13/3), where the numbers inside brackets $(n/d/c)$ denote the number of data points n, number of features d, and number of classes c. We perform kernel k-nearest neighbor (k-NN) classification and standard soft-margin SVM classification on the three data sets based on different kernel learning methods mentioned in Section 3.1. The optimal parameter values, k for kernel k-NN and C for SVM, may be different for different classification methods, so we set them to different values in our experiments. For all data sets, k is set to 1 and 3 for kernel k-NN and the regularization parameter C is set to 100 and 1000 for SVM.

Table 1-3 summarize the two-class classification results for different kernels and classifiers. Two different training set sizes (20% and 40%) are considered. Each classification rate reported is the average over 10 random trials. The lower value represents the standard deviation for each average test accuracy. While the method based on kernel alignment is

[3] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

not satisfactory, both the methods by Xiong et al. and us can improve over RBF kernel with our method giving better results. As pointed out by [27], the alignment-based method essentially learns a kernel matrix by maximizing the between-class distance in the feature space without considering the within-class distance. This limitation probably explains why its results are not satisfactory. Table 4-5 summarize the multi-class classification results. Xiong et al.'s method can naturally be extended for multi-class problems, so we compare our method with theirs. Experimental results verify the effectiveness of our method.

In the UCI repository web site, some reported classification results are better than ours, e.g., results for the Wine data set. However, it should be noted that their results are based on LOO evaluation, meaning that they use almost the entire data set for training. In our experiments, we use only 20% and 40% of the data to form the training sets. If we perform experiments based on LOO, the classification results will be improved significantly. For the Wine data set, the classification results based on LOO using kernel k-NN ($k = 3$) and SVM ($C = 1000$) are 99.46±1.13 and 99.68±1.02, respectively.

3.4 Face Recognition

We further assess the feasibility and performance of our method on the face recognition task, using a data set from the FERET database [19].[4] The FERET database contains 13,539 face images from 1,565 human subjects. The face images in the FERET database were acquired during different photo sessions, with variations in size, pose, illumination, facial expression, and aging. The FERET face database is from the FERET program sponsored by the US Department of Defense's Counterdrug Technology Development Program through the Defense Advanced Research Projects Agency (DARPA), and it has become the *de facto* standard for evaluating state-of-the-art face recognition algorithms. We use a subset of 470 images (10 images from each of 47 subjects) from the FERET

[4] `http://www.itl.nist.gov/iad/humanid/feret/`

Table 1: Classification results on the *Monks-3* data set for different kernel methods and training set sizes.

% labeled data	20%				40%			
	k-NN		SVM		k-NN		SVM	
	$k=1$	$k=3$	$C=10^2$	$C=10^3$	$k=1$	$k=3$	$C=10^2$	$C=10^3$
RBF	76.29	78.14	92.55	93.04	78.26	82.17	91.78	92.28
	±0.07	±0.12	±0.08	±0.06	±0.49	±0.20	±0.01	±0.01
Alignment	84.35	83.86	89.88	90.75	88.84	88.77	89.22	89.77
	±0.38	±0.24	±0.16	±0.28	±0.36	±0.10	±0.66	±0.31
Xiong's	87.65	89.25	**93.94**	92.71	88.17	89.68	**93.87**	94.25
	±0.05	±0.08	±0.08	±0.05	±0.37	±0.10	±0.02	±0.01
Ours	**88.36**	**89.54**	93.00	**93.30**	**90.70**	**90.95**	93.76	**94.36**
	±0.08	±0.04	±0.13	±0.12	±0.53	±0.11	±0.05	±0.06

database for our experiments. Each image is of size 46×56 with 256 gray levels. Figure 2 shows some sample images used in our experiments.

As the kernel direct discriminant analysis (KDDA) algorithm proposed by Lu et al. [13] has been shown to deliver appealing face recognition results when compared with KPCA and generalized discriminant analysis (GDA) [3], we use this method for classification with different kernel matrices.[5] We randomly select five images from each of the 47 classes to form the training set. The recognition results, averaged over 10 random trials, based on the standard RBF kernel and the kernel matrices learned by Xiong et al.'s and our methods are shown in Figure 3. As can be seen, our method outperforms the other two methods.

[5]The MATLAB code for KDDA is from the authors of [13].

Table 2: Classification results on the *Ionosphere* data set for different kernel methods and training set sizes.

% labeled data	20%				40%			
	k-NN		SVM		k-NN		SVM	
	$k=1$	$k=3$	$C=10^2$	$C=10^3$	$k=1$	$k=3$	$C=10^2$	$C=10^3$
RBF	78.82	79.00	83.14	82.82	82.26	81.55	89.19	87.63
	±0.07	±0.08	±0.06	±0.19	±0.25	±0.15	±0.28	±0.15
Alignment	78.89	78.37	78.68	80.90	82.00	81.77	84.86	88.33
	±0.26	±0.57	±0.37	±0.11	±0.10	±0.28	±0.51	±0.31
Xiong's	82.89	81.68	84.79	**85.21**	84.05	82.59	88.24	88.61
	±0.20	±0.12	±0.08	±0.21	±0.37	±0.20	±0.25	±0.10
Ours	**85.36**	**84.68**	**85.07**	85.15	**89.62**	**88.76**	**90.95**	**89.11**
	±0.06	±0.06	±0.13	±0.05	±0.53	±0.27	±0.10	±0.05

4 Conclusion

Recent years have seen intense research in kernel matrix learning to further enhance the power and potential of existing kernel methods. While optimization criteria such as kernel alignment and margin are commonly used, we propose in this paper a different criterion that is similar to the Fisher criteria used by LDA and KFD. Inspired by the parametric methods for solving nonlinear fractional programming problems, our class separability criterion can be optimized without encountering the singularity problem faced by many LDA and KFD applications. This new criterion function is also related to the maximum margin criterion proposed recently for LDA and KFD. Currently, we determine the parameter α via cross-validation. Another possibility is to formulate yet another optimization problem like those in fractional programming to find the optimal value of α.

Table 3: Classification results on the *Breast Cancer* data set for different kernel methods and training set sizes.

% labeled data	20%				40%			
	k-NN		SVM		k-NN		SVM	
	$k=1$	$k=3$	$C=10^2$	$C=10^3$	$k=1$	$k=3$	$C=10^2$	$C=10^3$
RBF	92.42	93.04	92.16	94.12	93.61	94.23	95.11	95.26
	±0.07	±0.10	±0.08	±0.11	±0.13	±0.14	±0.26	±0.24
Alignment	87.97	88.77	90.37	89.63	85.63	85.43	91.80	91.95
	±1.27	±1.87	±0.15	±0.15	±0.47	±0.59	±0.27	±0.40
Xiong's	90.79	91.78	94.50	95.86	93.85	93.85	95.53	96.12
	±0.12	±0.17	±0.16	±0.09	±0.17	±0.24	±0.13	±0.32
Ours	**94.19**	**94.48**	**96.06**	**96.77**	**94.23**	**94.94**	**97.11**	**97.61**
	±0.07	±0.03	±0.03	±0.04	±0.07	±0.10	±0.03	±0.07

Moreover, we will explore more general forms of transforming the initial kernel matrix in our future research.

Acknowledgment

The research described in this paper has been supported by Competitive Earmarked Research Grant (CERG) HKUST6174/04E from the Research Grants Council of the Hong Kong Special Administrative Region, China.

Table 4: Classification results on the *Wine* data set for different kernel methods and training set sizes.

% labeled data	20%				40%			
	k-NN		SVM		k-NN		SVM	
	$k=1$	$k=3$	$C=10^2$	$C=10^3$	$k=1$	$k=3$	$C=10^2$	$C=10^3$
RBF	91.63	94.07	95.11	95.04	94.24	94.81	95.81	96.04
	±0.16	±0.27	±0.38	±0.39	±0.56	±0.25	±0.37	±0.81
Xiong's	92.62	91.68	94.96	92.33	**95.52**	95.00	95.05	96.05
	±0.19	±1.28	±0.47	±1.10	±0.59	±0.40	±0.26	±0.55
Ours	**95.70**	**95.63**	**96.10**	**95.26**	95.38	**95.48**	**97.38**	**98.57**
	±0.42	±0.26	±0.01	±0.11	±0.18	±0.23	±0.07	±0.04

References

[1] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 41–48, Banff, Alberta, Canada, 4–8 July 2004.

[2] F.R. Bach, R. Thibaux, and M.I. Jordan. Computing regularization paths for learning multiple kernels. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 73–80. MIT Press, Cambridge, MA, USA, 2005.

[3] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[4] O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information*

Table 5: Classification results on the *Boston housing* data set for different kernel methods and training set sizes.

% labeled data	20%				40%			
	k-NN		SVM		k-NN		SVM	
	$k=1$	$k=3$	$C=10^2$	$C=10^3$	$k=1$	$k=3$	$C=10^2$	$C=10^3$
RBF	**76.53**	**74.63**	78.04	78.12	80.10	76.52	82.19	80.46
	±0.02	±0.06	±0.11	±0.22	±0.51	±0.47	±0.27	±0.64
Xiong's	72.92	73.22	75.15	74.70	79.47	74.83	79.54	79.11
	±0.06	±0.07	±0.03	±0.09	±0.39	±0.49	±0.34	±0.40
Ours	74.70	73.87	**79.34**	**79.38**	**80.40**	**77.14**	**82.51**	**83.84**
	±0.05	±0.12	±0.04	±0.10	±0.27	±0.45	±0.08	±0.22

Processing Systems 15, pages 399–406. MIT Press, Cambridge, MA, USA, 2003.

[5] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

[6] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 537–544. MIT Press, Cambridge, MA, USA, 2003.

[7] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, Cambridge, MA, USA, 2002.

[8] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

Figure 2: Twenty cropped sample images of two subjects from the FERET database.

[9] G. Fung, M. Dundar, J. Bi, and B. Rao. A fast iterative algorithm for Fisher discriminant using heterogeneous kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 313–320, Banff, Alberta, Canada, 4–8 July 2004.

[10] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 323–330, Sydney, Australia, 8–12 July 2002.

[11] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[12] H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA, 2004.

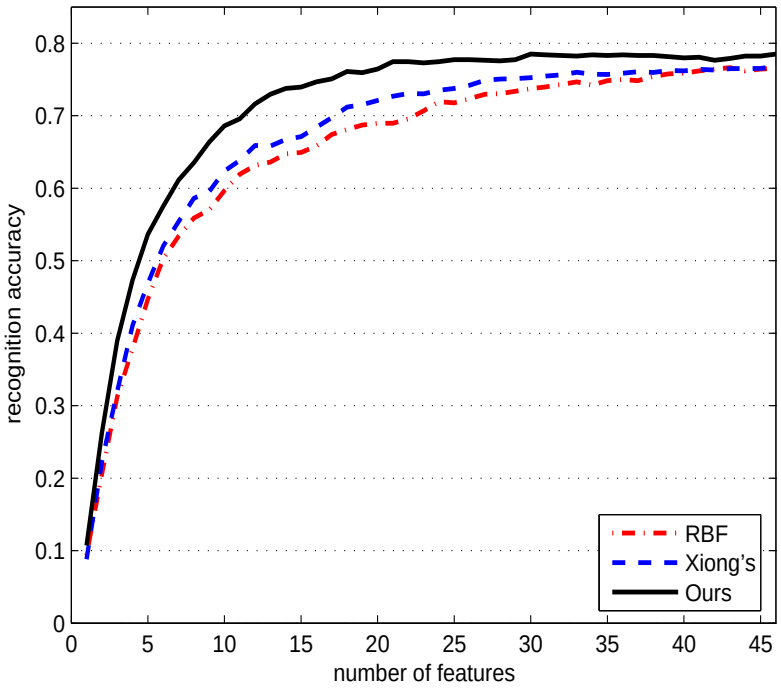

Figure 3: Face recognition results for different kernel methods.

[13] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, 14(1):117–126, 2003.

[14] C.A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[15] S. Mika, G. Rätsch, and K.R. Müller. A mathematical programming approach to the kernel Fisher algorithm. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 591–597. MIT Press, Cambridge, MA, USA, 2001.

[16] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of the*

1999 IEEE Signal Processing Society Workshop, pages 41–48, Madison, WI, USA, 23–25 August 1999.

[17] C.S. Ong and A.J. Smola. Machine learning using hyperkernels. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 568–575, Washington, DC, USA, 21–24 August 2003.

[18] C.S. Ong, A.J. Smola, and R.C. Williamson. Hyperkernels. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 478–485. MIT Press, Cambridge, MA, USA, 2003.

[19] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[20] C.R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society, Series B*, 10(2):159–203, 1948.

[21] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

[22] A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical Bayes. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1209–1216. MIT Press, Cambridge, MA, USA, 2005.

[23] I.M. Stancu-Minasian. *Fractional Programming: Theory, Methods and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[24] K. Tsuda, S. Akaho, and K. Asai. The *em* algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.

[25] K. Tsuda, G. Rätsch, and M.K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. In L.K. Saul, Y. Weiss, and L. Bottou,

editors, *Advances in Neural Information Processing Systems 17*, pages 1425–1432. MIT Press, Cambridge, MA, USA, 2005.

[26] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, USA, 1998.

[27] H. Xiong, M.N.S. Swamy, and M.O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474, 2005.

[28] Z. Zhang, D.Y. Yeung, and J.T. Kwok. Bayesian inference for transductive learning of kernel matrix using the Tanner-Wong data augmentation algorithm. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 935–942, Banff, Alberta, Canada, 4–8 July 2004.

[29] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1641–1648. MIT Press, Cambridge, MA, USA, 2005.

About the Author – DIT-YAN YEUNG received his B.Eng. degree in Electrical Engineering and M.Phil. degree in Computer Science from the University of Hong Kong, and his Ph.D. degree in Computer Science from the University of Southern California in Los Angeles. He was an Assistant Professor at the Illinois Institute of Technology in Chicago before he joined the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology, where he is currently an Associate Professor. His research interests are in machine learning and pattern recognition. He is currently serving on the editorial boards of *Journal of Artificial Intelligence Research* and *Pattern Recognition*.

About the Author – HONG CHANG received her Bachelor degree, M.Phil. degree and Ph.D. degree in Computer Science from Hebei University of Technology, Tianjin University, and Hong Kong University of Science and Technology, respectively. She is currently a Research Scientist in Xerox Research Centre Europe. Her main research interests include semi-supervised learning, nonlinear dimensionality reduction, and related applications.

About the Author – GUANG DAI received his B.Eng. degree in Mechanical Engineering from Dalian University of Technology and M.Phil. degree in Computer Science from Zhejiang University. He is currently a PhD student in the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. His main research interests include semi-supervised learning, nonlinear dimensionality reduction, and related applications.