Optimizing the Kernel in the Empirical Feature Space

Huilin Xiong, M. N. S. Swamy, *Fellow, IEEE*, and M. Omair Ahmad, *Fellow, IEEE*

***Abstract*—In this paper, we present a method of kernel optimization by maximizing a measure of class separability in the empirical feature space, an Euclidean space in which the training data are embedded in such a way that the geometrical structure of the data in the feature space is preserved. Employing a data-dependent kernel, we derive an effective kernel optimization algorithm that maximizes the class separability of the data in the empirical feature space. It is shown that there exists a close relationship between the class separability measure introduced here and the alignment measure defined recently by Cristianini. Extensive simulations are carried out which show that the optimized kernel is more adaptive to the input data, and leads to a substantial, sometimes significant, improvement in the performance of various data classification algorithms.**

***Index Terms*—Class separability, data classification, empirical feature space, feature space, kernel machines, kernel optimization.**

I. Introduction

RECENTLY, there has been a lot of interest in kernel-based learning or kernel machines in areas such as pattern recognition and machine learning [1]. Basically, kernel machines work by mapping the input data $\mathcal{X}$ into a feature space $\mathcal{F}$, $\Phi : \mathcal{X} \longrightarrow \mathcal{F}$, and then building linear algorithms in the feature space to implement nonlinear counterparts in the input data space. The map Φ, rather than being given in an explicit form, is presented implicitly by specifying a kernel function as the inner product between each pair of points in the feature space. It is assumed that the mapped data in the feature space is linearly separable or at least possesses a better linear separability than that in the input space. However, the separability of the data in the feature space could be even worse if an inappropriate kernel is used. Since the geometrical structure of the mapped data in the feature space is totally determined by the kernel matrix, the selection of the kernel has a crucial effect on the performance of various kernel machines. It is desirable for the kernel machines to use an optimized kernel function that adapts well to the input data and the learning tasks.

Given a training data set $\{x_i\}, (i = 1, 2, \ldots, m)$, the different algorithms, such as the support vector machine (SVM) [6], [10], [16], [22], kernel Fisher discriminant (KFD) [4], [15], [20], and kernel principal component analysis (KPCA) [23], perform only in a subspace of the feature space spanned by the images of the training data, $\{\Phi(x_i)\}$, $(i = 1, 2, \ldots, m)$. This subspace can be embedded into an Euclidean space in such a way that all the geometrical measurements, such as the distance and angle, between each pair of $\Phi(x_i)$ are preserved. The embedding map is referred to as the "empirical kernel map" [22]. We shall call the embedding space the "empirical feature space." Since the training data have the same geometrical structure in both the empirical feature space and the feature space, and the former is easier to access than the latter, it is easier to study, in the former space than the latter, the adaptability of a kernel to the input data and how to improve it.

In the literature, kernel optimization is often considered as a problem of "model selection," which is usually tackled by cross validation. However, cross validation can only select the parameters of a kernel function just from a set of prespecified discrete values of the parameters. Recently, Cristianini *et al.* [9] and Lanckrict *et al.* [14] have for the first time proposed methods of selecting the kernel or kernel matrix by optimizing the measure of data separation in the feature space. While the authors in [9] use the measure called "alignment" to evaluate the adaptability of a kernel to the data, those in [14] employ the margin or soft margin as the measure of data separation in the feature space. In this paper, we propose an alternate method to optimize the kernel function by maximizing a class separability criterion in the empirical feature space. Employing the data-dependent kernel model, we develop an effective algorithm to maximize the class separability measure in the empirical feature space. The final optimized kernel shows that it is more adaptive to the data and leads to a substantial improvement in the performance of the kernel-based data classification.

The paper is organized as follows. Section II shows how to embed the mapped data in the feature space into an Euclidean space such that all the geometrical relations are preserved. In Section III, a measure for the class separability of the data in the empirical feature space is formulated in term of the kernel matrices, and then, based on this measure, an optimization algorithm for the data-dependent kernel is derived. Simulation studies are carried out using the proposed algorithm, and they show a substantial improvement in the class separability for both the training and the test data. In addition, the relationship between the proposed class separability measure and the alignment measure introduced in [9] is discussed. In Section IV, a number of experiments using real data sets are carried out to demonstrate the improvement in the performance of the data classification algorithms after using the optimized kernel. Section V contains the conclusion.

II. Feature Space and Empirical Feature Space

Let $\{x_i\}_{i=1}^m$ be a d-dimensional training data set, X denote the $m \times d$ sample matrix whose rows consist of x_i^T $(i = 1, 2, \ldots, m)$, and K denote the $m \times m$ kernel

Manuscript received June 17, 2003; revised March 5, 2004. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by Fonds la Formation des Chercheurs et l'Aideà la Recherche (FCAR) of Québec.

The authors are with the Center for Signal Processing and Communications, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8 Canada (e-mail: hlxiong@ece.concordia.ca; swamy@ece.concordia.ca; omair@ece.concordia.ca).

Digital Object Identifier 10.1109/TNN.2004.841784

1045-9227/$20.00 © 2005 IEEE

matrix of rank, say r, that is, $K = [k_{ij}]_{m\times m}$, where $k_{ij} = \Phi(x_i)\cdot\Phi(x_j) = k(x_i, x_j)$. Since K is a symmetrical positive–semidefinite matrix, K can be decomposed as

$$K_{m\times m} = P_{m\times r}\Lambda_{r\times r}P^T_{r\times m} \tag{1}$$

where Λ is a diagonal matrix containing only the r positive eigenvalues of K in decreasing order, and P consists of the eigenvectors corresponding to the positive eigenvalues. The map from the input data space to an r-dimensional Euclidean space Φ^e_r: $\mathcal{X} \longrightarrow \mathbf{R^r}$

$$x \longrightarrow \Lambda^{-1/2}P^T(k(x,x_1), k(x,x_2), \ldots, k(x,x_m))^T$$

is essentially the empirical kernel map in [22]. We shall call the embedding space $\Phi^e_r(\mathcal{X}) \subset \mathbf{R}^r$ the empirical feature space.

It is easy to verify that the empirical feature space preserves the geometrical structure of $\{\Phi(x_i)\}$ in the feature space. Let Y be an $m \times r$ matrix which has each $\Phi^e_r(x_i)$ for its rows. That is, $Y = KP\Lambda^{-1/2}$. Then, the dot product matrix of $\{\Phi^e_r(x_i)\}$ in the empirical feature space can be calculated as

$$YY^T = KP\Lambda^{-1/2}\Lambda^{-1/2}P^TK = K.$$

This is exactly the dot product matrix of $\{\Phi(x_i)\}$ in the feature space. Since the distances and angles of the m vectors $\{\Phi(x_i)\}^m_1$ in the feature space are uniquely determined by the dot product matrix ($\| \Phi(x_i) - \Phi(x_j) \|^2 = k(x_i,x_i) + k(x_j,x_j) - 2k(x_i,x_j)$), the training data has the same geometrical structure and, hence, the same class separability, in both the empirical feature space and the feature space. An interesting fact is that the empirical kernel map described previously is exactly the KPCA transform [3] when the kernel matrix K is substituted by the centered kernel matrix

$$K_c \triangleq K - \frac{1}{m}1_{m\times m}K - \frac{1}{m}K1_{m\times m} + \frac{1}{m^2}1_{m\times m}K1_{m\times m}$$

where $1_{m\times m}$ denotes the $m \times m$ matrix with all entries being equal to unity.

Given the training data set, all the kernel-based algorithms, such as the SVM, KFD, KPCA, and kernel minimum squared error (KMSE) [3], perform just in the subspace $\overline{\mathrm{Span}(\Phi(\mathrm{x_i}))} \subset \mathcal{F}$, which is isomorphic with the empirical feature space according to the previous discussion. From both the theoretical and practical points of view, it is easier to access the empirical feature space than the feature space. Since the geometrical structure of the training data in the empirical feature space is the same as that in the feature space, the former provides a tractable framework to study the spatial distribution of $\{\Phi(x_i)\}$, to measure the class separability of $\{\Phi(x_i)\}$, and more importantly, to optimize the kernel in order to increase the separability and, hence, improve the performance of the kernel machines.

Before we concentrate on the task of optimizing the kernel in the empirical feature space, let us get some intuitive feeling about the embedding of $\{\Phi(x_i)\}$ into the empirical feature space through two examples. For more information about data embedding, one can refer to [17]. Fig. 1(a) shows a two-dimensional (2-D) data set with 320 samples, whose coordinates are uncorrelated. The samples are separated into two classes, one containing 150 samples, and the other 170 samples, both being Gaussian with $\mu_x = -2$, $\mu_y = 0$, $\sigma_x = 2$, $\sigma_y = 1$ and with $\mu_x = 2$, $\mu_y = 0$, $\sigma_x = 1$, $\sigma_y = 2$, respectively. We see from this figure that there is some overlap between the two classes. Fig. 1(b) shows the projection of the data in the empirical feature space onto the first two significant dimensions corresponding to the first two largest eigenvalues of K, when the polynomial kernel function $k(x,y) = (x\cdot y)^2$ is used. Fig. 1(c) gives the corresponding projection when the Gaussian kernel function $k(x,y) = e^{-\gamma\|x-y\|^2}$ with $\gamma = 0.001$ is employed. From Fig. 1(b), it is seen that in the case of the polynomial kernel $k(x,y)$, the class separability is worse in the feature space than that in the input space, even though it is based on only the 2-D projection of the embedding. We shall show later, after having defined the measure for class separability, that it is indeed so for the example under consideration. In other words, it is possible that the class separability could be worse in the feature space than in the input space.

III. Kernel Optimization in Empirical Feature Space

A. Data-Dependent Kernel

Different kernels create different geometrical structures of the data in the feature space, and lead to different class discrimination. Since there is no general kernel function suitable to all data sets [9], it is reasonable to choose the objective kernel function to be data-dependent. In this paper, we employ a data-dependent kernel similar to that used in [2] as the objective kernel to be optimized.

Let us consider a training data (x_1,ξ_1), $(x_2,\xi_2),\ldots,$ $(x_m,\xi_m) \in \mathbf{R}^d \times \{\pm 1\}$. We use the so-called "conformal transformation of a kernel" [2] as our data-dependent kernel function

$$k(x,y) = q(x)q(y)k_0(x,y) \tag{2}$$

where $x, y \in \mathbf{R}^d$, $k_0(x,y)$, called the basic kernel, is an ordinary kernel such as a Gaussian or a polynormial kernel, and $q(.)$, the factor function, is of the form

$$q(x) = \alpha_0 + \sum_{i=1}^{n}\alpha_i k_1(x,a_i) \tag{3}$$

in which $k_1(x,a_i) = e^{-\gamma\|x-a_i\|^2}$, $a_i \in \mathbf{R}^d$, and α_i's are the combination coefficients. The set $\{a_i, i = 1,2,\ldots,n\}$, called the "empirical cores," can be chosen from the training data or determined according to the distribution of the training data. It is easy to see that the data-dependent kernel satisfies the Mercer condition for a kernel function [22]. In [2], Amari and Wu chose the support vectors as the empirical cores, aiming to enlarge the spatial resolution around the class boundary and, thus, increase the class margin or class separation. The authors in [2] did not consider as to how to optimize the data-dependent kernel, perhaps in view of the complexity of the Riemannian metric in the feature space. In the following sections, we introduce a measure for the class separability in the empirical feature space, and develop an effective algorithm for optimizing the combination coefficients α_i to maximize the separability measure. Let us first present some notations.

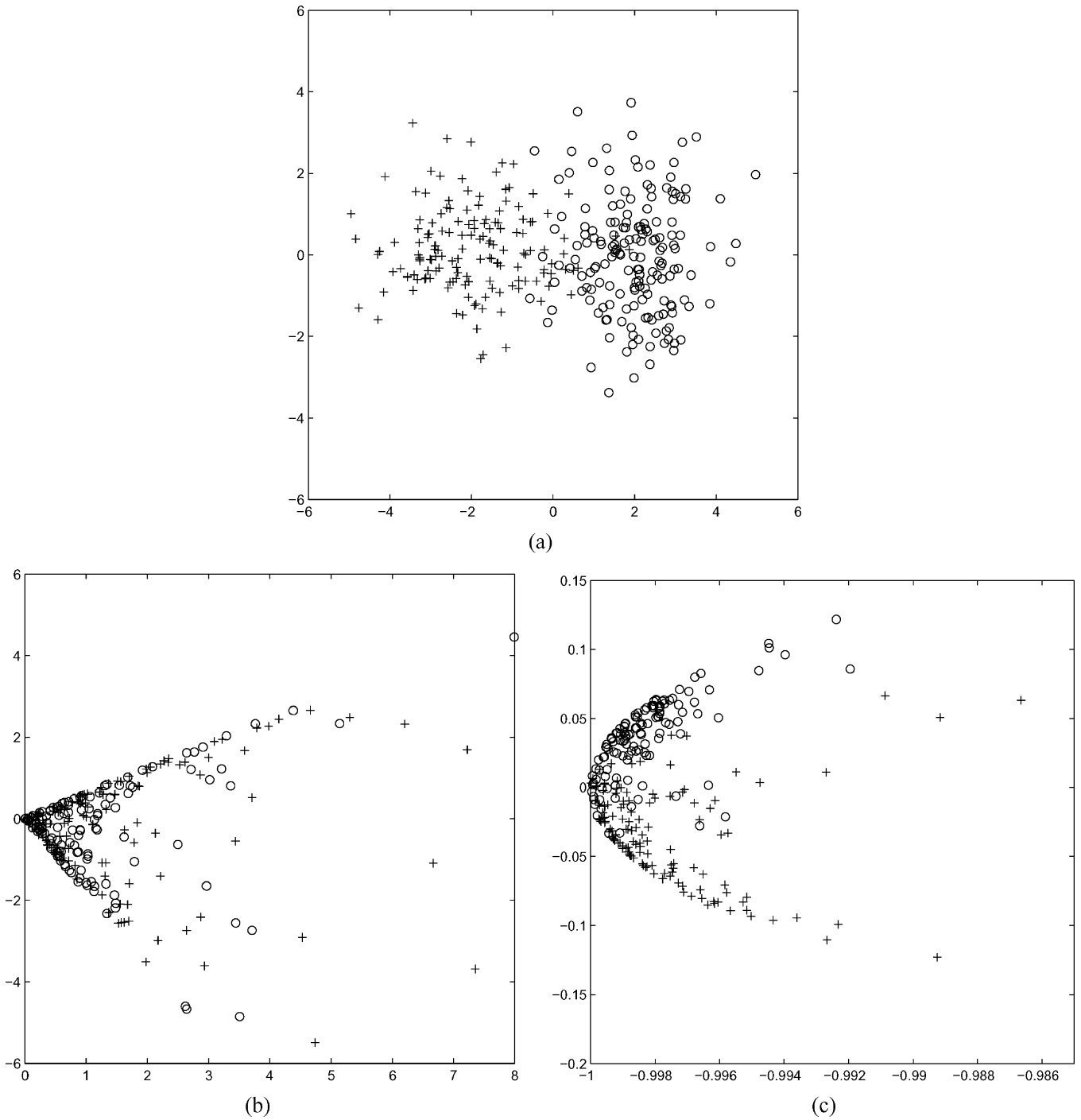


Fig. 1. Two-dimensional data set and its projections in the empirical feature space onto the first two significant dimensions. (a) Two-dimensional data set consisting of two Gaussian distributions. (b) Two-dimensional projection in the empirical feature space for the second-order polynomial kernel function. (c) Two-dimensional projection in the empirical feature space for the Gaussian kernel function.

The kernel matrices corresponding to $k(x,y)$ and $k_0(x,y)$ are denoted by K and K_0, respectively. That is, $K=[k(x_i,x_j)]_{m\times m}$ and $K_0=[k_0(x_i,x_j)]_{m\times m}$. It is easy to see that

$$K=[q(x_i)q(x_j)k_0(x_i,x_j)]_{m\times m}=QK_0Q \tag{4}$$

where Q is a diagonal matrix, whose diagonal elements are $\{q(x_1),q(x_2),\ldots,q(x_m)\}$. We denote the vectors $(q(x_1),q(x_2),\ldots,q(x_m))^T$ and $(\alpha_0,\alpha_1,\ldots,\alpha_n)^T$ by q and $\boldsymbol{\alpha}$, respectively. Then, we have

$$q=\begin{pmatrix}1 & k_1(x_1,a_1) & \cdots & k_1(x_1,a_n)\\ 1 & k_1(x_2,a_1) & \cdots & k_1(x_2,a_n)\\ \vdots & \vdots & \ddots & \vdots\\ 1 & k_1(x_m,a_1) & \cdots & k_1(x_m,a_n)\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\\ \vdots\\ \alpha_n\end{pmatrix}\triangleq K_1\alpha \tag{5}$$

where K_1 is an $m\times(n+1)$ matrix.

B. Class Separability Measure in the Empirical Feature Space

The training data set has the same geometrical structure in both the feature space and the empirical feature space. In view of this, it is better to measure the class separability of the data in the empirical feature space, since it is easier to access the empirical feature space than the feature space, as mentioned in Section II. We use the following quantity for measuring the class separability of the training data in the empirical feature space:

$$J = \frac{\mathrm{tr}S_b}{\mathrm{tr}S_w} \tag{6}$$

where S_b is the "between-class scatter matrix," S_w the "within-class scatter matrix," and "tr" denotes the trace of a matrix. J is also the well-known Fisher scalar for measuring the class linear separability, and is called "criteria J_4" in [12]. In Fisher discriminant analysis (FDA), the Rayleigh quotient $R(\Omega)$ is often used to measure the class separability, where

$$R(\Omega) = \frac{|\Omega^T S_b \Omega|}{|\Omega^T S_w \Omega|}$$

Ω being the projection matrix, which is to be determined later in the optimization algorithm. Compared with the Rayleigh measure $R(\Omega)$, the quantity J in (6) measures the class separability in the feature space rather than in the projection subspace. Moreover, since J is independent of the projections, it is more convenient to use it for the task of kernel optimization. Optimizing the data-dependent kernel through J means increasing the linear separability of the training data in the feature space, and this should lead to an improvement in the performance of the kernel machines, since they are all essentially linear machines in the feature space.

Let the number of samples in one of the classes, say C_1 (class label equals -1), be m_1, and the number of samples in the other class, say C_2 (class label equals $+1$), be m_2. Let $\{y_i\}_{i=1}^m$ be the images of the training data in the empirical feature space, where $m_1 + m_2 = m$. Let $\overline{y}$, $\overline{y}_1$ and $\overline{y}_2$, respectively, denote the center of the entire training data and those of C_1 and C_2 in the empirical feature space. Then, we have

$$S_b = \frac{1}{m}\sum_{i=1}^{2} m_i(\overline{y}_i - \overline{y})(\overline{y}_i - \overline{y})^T$$
$$S_w = \frac{1}{m}\sum_{i=1}^{2}\sum_{j=1}^{m_i}(y_j^i - \overline{y}_i)(y_j^i - \overline{y}_i)^T$$

where the vector y_j^i denotes the jth data in the ith class ($i = 1, 2$).

As an example, for the data set shown in Fig. 1(a), we calculate the values of the measure J when the polynomial kernel $k(x, y) = (x \cdot y)^p$ with $p = 1, 2$, and 3. These values are found to be 0.6027, 0.0234, and 0.1350, respectively. Considering the fact that using the polynomial kernel with $p = 1$ means the feature space and input space are identical, we see that for this example the class separability of the data in the feature space corresponding $p = 2$, or $p = 3$ is worse than that in the input space. This confirms the observation made earlier in Section II, which was based purely on a 2-D projection of the embedding.

Without loss of generality, let us now assume that the first m_1 data belong to class C_1, that is, $\xi_i = -1$, $i \le m_1$, and the remaining m_2 data belong to C_2 ($m_1 + m_2 = m$). This is done for the sake of convenience. Then, the kernel matrices can be written as

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

where K_{11}, K_{12}, K_{21}, and K_{22} represent the submatrices of K of order $m_1 \times m_1$, $m_1 \times m_2$, $m_2 \times m_1$, and $m_2 \times m_2$, respectively. Obviously, K_{11} is the kernel matrix corresponding to the data in class C_1, and K_{22} that for the data in class C_2.

Let us call the following matrices "between-class" and "within-class" kernel scatter matrices, and denote them by B and W, respectively

$$B = \begin{pmatrix} \frac{1}{m_1}K_{11} & 0 \\ 0 & \frac{1}{m_2}K_{22} \end{pmatrix} - \begin{pmatrix} \frac{1}{m}K_{11} & \frac{1}{m}K_{12} \\ \frac{1}{m}K_{21} & \frac{1}{m}K_{22} \end{pmatrix}$$
$$W = \begin{pmatrix} k_{11} & 0 & \cdots & 0 \\ 0 & k_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{mm} \end{pmatrix} - \begin{pmatrix} \frac{1}{m_1}K_{11} & 0 \\ 0 & \frac{1}{m_2}K_{22} \end{pmatrix}.$$

We also denote by B_0 and W_0, the between-class and the within-class kernel scatter matrices corresponding to the basic kernel K_0. Now, we establish a relation between J and the kernel scatter matrices by the following theorem.

Theorem 1: Let 1_k be the k-dimensional vector whose entries are all equal to unity. Then

$$J = \frac{1_m^T B 1_m}{1_m^T W 1_m} = \frac{q^T B_0 q}{q^T W_0 q}. \tag{7}$$

Proof: Suppose the dimension of the empirical feature space be r ($r \le m$), that is, the dot product matrix K has exactly r positive eigenvalues. Let Y denote the $m \times r$ matrix whose rows are the vectors $\{y_i^T\}$, Y_1 the $m_1 \times r$ matrix whose rows are $\{y_i^T\}$ ($i \le m_1$), and Y_2 denote the $m_2 \times r$ matrix whose rows are the vectors $\{y_i^T\}$ ($i > m_1$).

First, we have

$$\overline{y} = \frac{1}{m}\sum_{i=1}^{m} y_i = \frac{1}{m}Y^T 1_m$$
$$\overline{y}_1 = \frac{1}{m_1}\sum_{i=1}^{m_1} y_i = \frac{1}{m_1}Y_1^T 1_{m_1}$$
$$\overline{y}_2 = \frac{1}{m_2}\sum_{i=m_1+1}^{m} y_i = \frac{1}{m_2}Y_2^T 1_{m_2}.$$

Since the empirical feature space preserves the dot product

$$(Y_1\ Y_2)\begin{pmatrix} Y_1^T \\ Y_2^T \end{pmatrix} = YY^T = K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}.$$

Hence, we have

$$Y_1 Y_1^T = K_{11},\ Y_2 Y_2^T = K_{22},\ Y_1 Y_2^T = K_{12},\ Y_2 Y_1^T = K_{21}.$$

Therefore

$$\begin{aligned}
\mathrm{tr}S_b &= \frac{1}{m}\sum_{i=1}^{2} m_i(\overline{y}_i-\overline{y})^T(\overline{y}_i-\overline{y}) \\
&= \frac{1}{m}\sum_{i=1}^{2} m_i\overline{y_i}^T\overline{y_i} - \overline{y}^T\overline{y} \\
&= \frac{1}{m}\sum_{i=1}^{2}\frac{1}{m_1}1_{m_i}^T Y_i Y_i^T 1_{m_i} - \frac{1}{m^2}1_m^T YY^T 1_m \\
&= \frac{1}{m}\sum_{i=1}^{2}\frac{1}{m_i}1_{m_i}^T K_{ii}1_{m_i} - \frac{1}{m^2}1_m^T K 1_m \\
&= \frac{1}{m}(1_{m_1}^T\ 1_{m_2}^T)\begin{pmatrix}\frac{1}{m_1}K_{11} & 0\\ 0 & \frac{1}{m_2}K_{22}\end{pmatrix}\begin{pmatrix}1_{m_1}\\ 1_{m_2}\end{pmatrix} \\
&\quad - \frac{1}{m}1_m^T\begin{pmatrix}\frac{1}{m}K_{11} & \frac{1}{m}K_{12}\\ \frac{1}{m}K_{21} & \frac{1}{m}K_{22}\end{pmatrix}1_m = \frac{1}{m}1_m^T B 1_m \\
\text{and } \mathrm{tr}S_w &= \frac{1}{m}\sum_{i=1}^{2}\sum_{j=1}^{m_i}(y_j^i-\overline{y}_i)^T(y_j^i-\overline{y}_i) \\
&= \frac{1}{m}\sum_{i=1}^{2}\left(\sum_{j=1}^{m_i}(y_j^i)^T y_j^i - m_i\overline{y_i}^T\overline{y}_i\right) \\
&= \frac{1}{m}\left(\sum_{i=1}^{m} y_i^T y_i - \sum_{i=1}^{2}\frac{1}{m_i}1_{m_i}^T K_{ii}1_{m_i}\right) \\
&= \frac{1}{m}\left[\sum_{i=1}^{m} k_{ii} - 1_m^T\begin{pmatrix}\frac{1}{m_1}K_{11} & 0\\ 0 & \frac{1}{m_2}K_{22}\end{pmatrix}1_m\right] \\
&= \frac{1}{m}1_m^T W 1_m.
\end{aligned}$$

These two equalities prove the first part of the theorem. Furthermore, using (4), we can easily see that $B = QB_0Q$ and $W = QW_0Q$. Considering that $Q1_m = q$, the theorem is established.

C. Kernel Optimization

To maximize $J(\alpha)$, we follow the standard gradient approach. Let

$$J_1 = J_1(q(\alpha)) = 1_m^T B 1_m = q^T B_0 q \tag{8}$$
$$J_2 = J_2(q(\alpha)) = 1_m^T W 1_m = q^T W_0 q. \tag{9}$$

We now establish the following theorem.

Theorem 2:

$$\frac{\partial J_1}{\partial \alpha} = 2K_1^T B_0 K_1 \alpha$$
$$\frac{\partial J_2}{\partial \alpha} = 2K_1^T W_0 K_1 \alpha.$$

Proof: Since $J_i = J_i(q(\alpha))$ $(i = 1, 2)$, we have

$$\frac{\partial J_i}{\partial \alpha} = \left(\frac{\partial q}{\partial \alpha}\right)^T \frac{\partial J_i}{\partial q}. \tag{10}$$

Kernel Optimization Algorithm

1. Group the data according to their class labels. Calculate K_0 and K_1 first, then B_0 and W_0, and then M_0, N_0.
2. Initialize $\alpha^{(0)}$ by a vector $(1, 0, \ldots, 0)^T$, and set $n = 0$.
3. Calculate $q = K_1\alpha^{(n)}$.
4. Calculate $J_1 = q^T B_0 q$, $J_2 = q^T W_0 q$, and J.
5. Update $\alpha^{(n)}$
$$\alpha^{(n+1)} = \alpha^{(n)} + \eta(n)\,(\frac{1}{J_2}M_0 - \frac{J}{J_2}N_0)\alpha^{(n)}$$
and normalize $\alpha^{(n+1)}$ so that $\| \alpha^{(n+1)} \| = 1$.
6. If n reaches a pre-specified number N, stop. Otherwise, set $n = n + 1$, go to step 3.

Fig. 2. Kernel optimization algorithm.

From (8) and (9), we can see

$$\frac{\partial J_1}{\partial q} = 2B_0 q$$
$$\frac{\partial J_2}{\partial q} = 2W_0 q.$$

Considering $q = K_1\alpha$, we have

$$\frac{\partial q}{\partial \alpha} = \left[\frac{\partial q_i}{\partial \alpha_j}\right]_{m\times(n+1)} = K_1 \tag{11}$$

and

$$\frac{\partial J_1}{\partial q} = 2B_0 K_1 \alpha \tag{12}$$
$$\frac{\partial J_2}{\partial q} = 2W_0 K_1 \alpha. \tag{13}$$

From (10)–(13), the theorem follows.

Let $M_0 = K_1^T B_0 K_1$ and $N_0 = K_1^T W_0 K_1$. Then, according to Theorem 2, we have

$$\frac{\partial J}{\partial \alpha} = \frac{2}{J_2^2}(J_2 M_0 - J_1 N_0)\alpha.$$

To maximize J, let $\partial J/\partial\alpha = 0$, we obtain

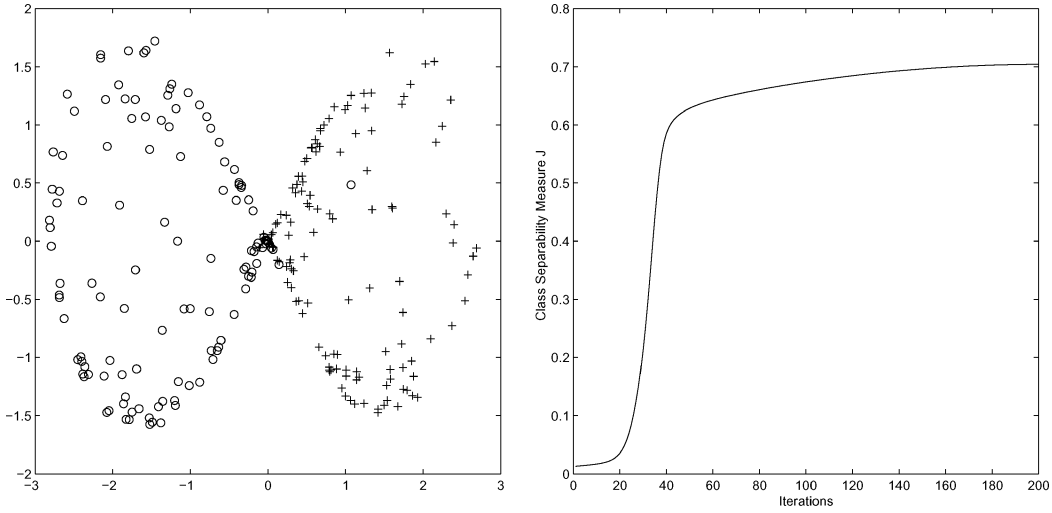
$$J_1 N_0 \alpha = J_2 M_0 \alpha.$$

If N_0^{-1} exists, we have

$$J\alpha = N_0^{-1} M_0 \alpha$$

which means that the maximum value of J equals to the largest eigenvalue of the matrix $N_0^{-1}M_0$, and the eigenvector corresponding to the largest eigenvalue is the optimal α. Unfortunately, matrix $N_0^{-1}M_0$ is generally not symmetrical, and even worse, N_0 may be a singular matrix.

To avoid using the eigenvalue resolution, we employ an updating algorithm to get an approximate value of the optimal α.



(a) (b)

Fig. 3. Improvement in the class separability for the polynomial basic kernel. (a) Two-dimensional projections after 200 iterations of the kernel optimization. (b) Class separability measure J as a function of the iteration number.

According to the general gradient method, the updating equation for maximizing the class separability J is given by

$$\alpha^{(n+1)} = \alpha^{(n)} + \eta \left(\frac{1}{J_2} M_0 - \frac{J}{J_2} N_0 \right) \alpha^{(n)} \tag{14}$$

where J and J_2 are functions of $\alpha^{(n)}$, M_0 and N_0 are two constant matrices, and η is the learning rate. To ensure the convergence of the algorithm, a gradually decreasing learning rate is adopted

$$\eta(t) = \eta_0 \left(1 - \frac{t}{N} \right) \tag{15}$$

where η_0 is the initial learning rate, N denotes a prespecified number of iterations, and t represents the current iteration number. Now, let us summarize our kernel optimization algorithm on the training data in Fig. 2.

It is easy to see that the algorithm is of $O(Nn^2)$ computational complexity, where n stands for the data size and N denotes the prespecified iteration number.

To show the effectiveness of the optimization algorithm, we test it on the synthetic data, shown in Fig. 1(a). The polynomial kernel, $k(x, y) = (x \cdot y)^p$ with $p = 2$, and the Gaussian kernel, $k(x, y) = e^{-\gamma_0 \|x-y\|^2}$ with $\gamma_0 = 0.001$ are used as the basic kernels. The parameter γ of the function $k_1(\cdot, \cdot)$ in (3) is set as $\gamma = 1.0$ for the polynomial basic kernel, and $\gamma = 0.5$ for the Gaussian basic kernel. One third of the data are randomly selected to form the empirical core set $\{a_i\}$. The initial learning rate η_0 of the algorithm is set to 0.5 for the polynomial kernel, and 0.01 for the Gaussian kernel. The total iteration number N is set to 200. Fig. 3(a) shows the projections of the data in the empirical feature space onto its first two dimensions corresponding to the first two significant eigenvalues of the matrix K, when the second-order polynomial kernel is used as the basic kernel. Fig. 3(b) shows the manner in which the value of J increases as the number of iterations is increased. The corresponding results, when the Gaussian kernel is used, are shown in Fig. 4. It is seen from Figs. 3 and 4 that the proposed kernel optimization algorithm substantially improves the class separability of the data in the empirical feature space and, hence, in the feature space.

D. Relation Between the Measure J and the Alignment Measure

The "alignment" measure was introduced by Cristianini *et al.* [9] in order to measure the adaptability of a kernel to the target data, and provide a practical objective for kernel optimization. In this section, we show that the alignment measure is related to our class separability measure J.

The alignment measure is defined in [9] as a normalized Frobenius inner product between the kernel matrix K and the target label matrix

$$A = \frac{\langle K, \xi\xi^T \rangle_F}{\sqrt{\langle K, K \rangle_F \langle \xi\xi^T, \xi\xi^T \rangle_F}}$$

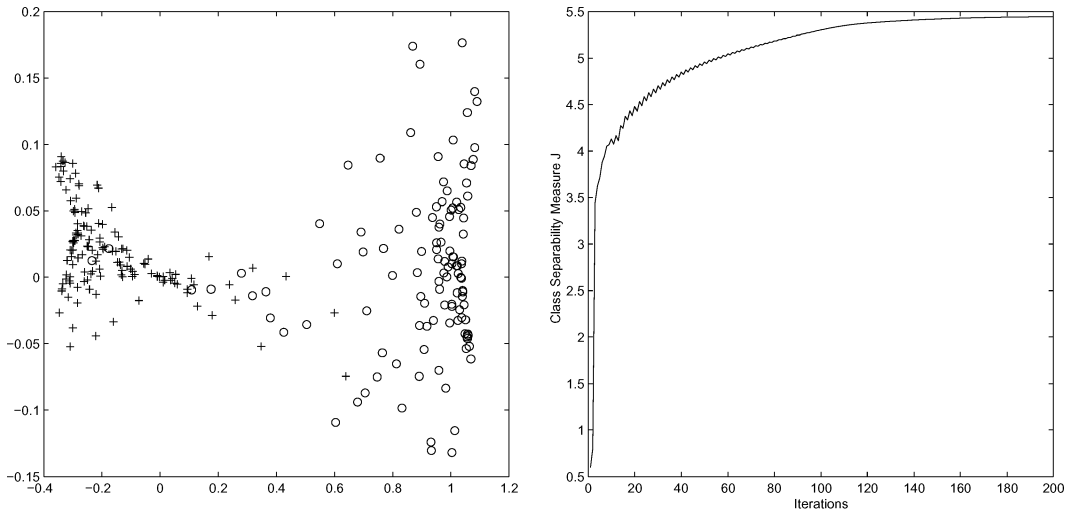
where ξ denotes the label vector of the training data, and the Frobenius product between two Gram matrices M and N is defined as $\langle M, N \rangle_F = \sum_{i,j} m_{ij} n_{ij} = \mathrm{tr}(MN)$. Obviously, the Frobenius product can also been written as $\langle M, N \rangle_F = 1_m^T (M .* N) 1_m$, where $M .* N$ denotes a matrix whose entries are obtained by multiplying the corresponding entries of the two $m \times m$ matrices M and N.

If we substitute the kernel and label matrices by their centered matrices K_c and $\xi_c \xi_c^T$, in which $\xi_c = \xi - (1/m) 1_{m \times m} \xi$, the alignment of the centered kernel matrices, called centered alignment, can be written as

$$A_c = \frac{\langle K_c, \xi_c \xi_c^T \rangle_F}{\sqrt{\langle K_c, K_c \rangle_F \langle \xi_c \xi_c^T, \xi_c \xi_c^T \rangle_F}}. \tag{16}$$

We now show that the following equality holds:

$$\frac{\langle K_c, \xi_c \xi_c^T \rangle_F}{\sqrt{\langle \xi_c \xi_c^T, \xi_c \xi_c^T \rangle_F}} = 1_m^T B_c 1_m \tag{17}$$



(a) (b)

Fig. 4. Improvement in the class separability for the Gaussian basic kernel. (a) Two-dimensional projections after 200 iterations of the kernel optimization. (b) Class separability measure J as a function of the iteration number.

where B_c represents the between-class kernel scatter matrix for the centered kernel matrix K_c.

Since

$$\xi_c^i = \begin{cases} -1 - \frac{m_2 - m_1}{m} & i \le m_1 \\ 1 - \frac{m_2 - m_1}{m} & m_1 < i \le m \end{cases} \quad (i = 1, 2, \ldots, m)$$

we have

$$\langle \xi_c \xi_c^T, \xi_c \xi_c^T \rangle_F = (\xi_c^T \xi_c)^2 = \left[\sum_{i=1}^{m} \left(\xi_c^i\right)^2 \right]^2 = 16 \left(\frac{m_1 m_2}{m} \right)^2 \tag{18}$$

and

$$\xi_c \xi_c^T = 4 \begin{pmatrix} \left(\frac{m_2}{m}\right)^2 1_{m_1 \times m_1} & -\left(\frac{m_1 m_2}{m^2}\right) 1_{m_1 \times m_2} \\ -\left(\frac{m_1 m_2}{m^2}\right) 1_{m_2 \times m_1} & \left(\frac{m_1}{m}\right)^2 1_{m_2 \times m_2} \end{pmatrix}.$$

Therefore

$$\begin{aligned} &\langle K_c, \xi_c \xi_c^T \rangle_F \\ &= 4\, 1_m^T \begin{pmatrix} \left(\frac{m_2}{m}\right)^2 K_{11}^c & -\frac{m_1 m_2}{m^2} K_{12}^c \\ -\frac{m_1 m_2}{m^2} K_{21}^c & \left(\frac{m_1}{m}\right)^2 K_{22}^c \end{pmatrix} 1_m \\ &= 4 \frac{m_1 m_2}{m} 1_m^T \begin{pmatrix} \left(\frac{1}{m_1} - \frac{1}{m}\right) K_{11}^c & -\frac{1}{m} K_{12}^c \\ -\frac{1}{m} K_{21}^c & \left(\frac{1}{m_2} - \frac{1}{m}\right) K_{22}^c \end{pmatrix} 1_m \\ &= 4 \frac{m_1 m_2}{m} 1_m^T B_c 1_m \end{aligned} \tag{19}$$

where K_{ij}^c denotes the submatrices of K_c corresponding to K_{ij}. Hence, from (18) and (19), the equality given by (17) is proved.

From (17) and (8), we see that the alignment measure is equal to J_1, provided the kernel matrix K has been centralized to K_c and normalized by its Frobenius length $\sqrt{\langle K, K \rangle_F}$ in the preprocessing. Therefore, optimizing the alignment measure essentially means increasing the between-class distance of the data in the feature space. Moreover, optimizing the class separability measure J in (6) not only increases the between-class distance J_1, but also reduces the within-class distance J_2 of the data in the feature space. In the next section, simulations on real data sets are carried out to demonstrate a corresponding increase of the centered alignment measure A_c and the class separability measure J in the process of the kernel optimization.

E. Kernel Optimization and Overfitting

Since the kernel optimization is carried out only on the training data, a question about the effect on the test data naturally arises. We assume the training data and the test data have the same spatial distribution; therefore, increasing the adaptation of a kernel to the training data, or increasing the class separability of the training data in the feature space, should lead to a similar effect on the test data. In this section, we illustrate that the class separability of the test data improves in the same manner as that of the training data to which the optimization algorithm is applied. In order to do so, three real data sets, namely, *Ionosphere*, *Wisconsin Breast Cancer*, and *Monks3*, adopted from the UCI benchmark repository [5], are considered. *Ionosphere* contains 351 34-dimensional samples, *Breast Cancer* contains 569 30-dimensional data, and *Monks3* includes 432 6-dimensional samples. Each data set is first normalized to a distribution with zero mean and unit variance, and then randomly partitioned into three equal and disjoint parts. One of these parts is used as the empirical core set $\{a_i\}$, and the other two as the training and test sets. The parameters in the optimization algorithm are set as $\gamma_0 = 0.0001, \gamma = 0.05$ for the *Ionosphere* and *Breast Cancer* data, and $\gamma_0 = 0.0001, \gamma = 0.1$ for the *Monks3* data. The initial learning rate η_0 in (15) and the iteration number N are set to 0.01 and 200, respectively, for all the three data sets.

Fig. 5(a) shows the projections of the training and test data in their respective empirical feature space onto the first two significant dimensions before the kernel optimization. Fig. 5(b) shows the corresponding projections of the training and test data after the kernel optimization. It can been seen that the class separability for the training as well as for the test data has substantially improved in a similar manner, although the optimization algorithm is performed only on the training data. Furthermore,

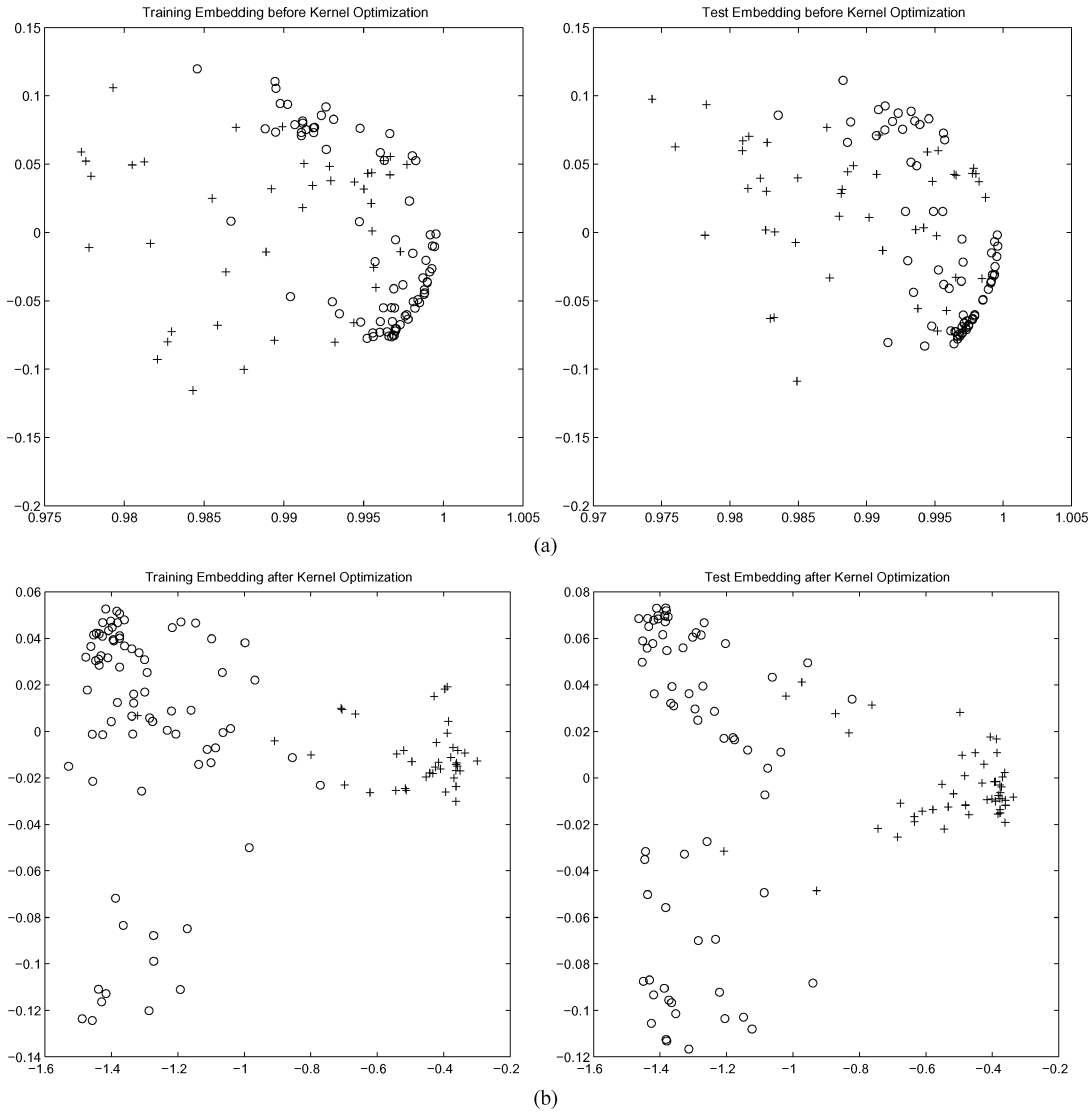


Fig. 5. Class separability for the *Ionosphere* training and test data. (a) Two-dimensional projection of the training and test data before the kernel optimization. (b) Two-dimensional projection of the training and test data after the kernel optimization.

Fig. 6(a) and (b) shows the increasing nature of the class separability measure J and the centered alignment measure A_c with respect to the number iterations of the algorithm not only for the training data but also for the test data. The corresponding results for the *Breast Cancer* and *Monks3* data are presented in Figs. 7–10.

IV. Classification With the Optimized Kernel

In this section, we conduct experiments on real data sets for data classification in order to show that using the optimized data-dependent kernel can further improve the performance of the data classification algorithms such as the k-nearest-neighbor (KNN), SVM, KFD, and KMSE. We also compare the proposed kernel optimization method with the alignment method [9] with regard to the data classification. Before we present the experiment results, we first summarize the algorithms mentioned previously.

A. Data Classification Algorithms

1) KNN Classification: The KNN method is the simplest, yet a useful one for data classification. Its performance, however, deteriorates dramatically when the input data set has a relatively low local relevance [11]. It is obvious that for the Gaussian kernel, there is no benefit in performing the KNN classification in the feature space, since the distance-based ranking in both the input and the feature space are the same. However, with the use of the data-dependent kernel in the KNN method, especially after the kernel is optimized according to the proposed algorithm, the distance metric between each pair of data is appropriately modified in the optimization process, and the local relevance of the data in the feature space could be significantly improved, as shown in Figs. 5 and 7, and especially Fig. 9. Therefore, the performance of the KNN classifier can be significantly improved by the use of the optimized kernel. Through the experimental results given in Section IV-B, we will see a remarkable reduction in the classification error of the KNN method, when the optimized kernel is used.

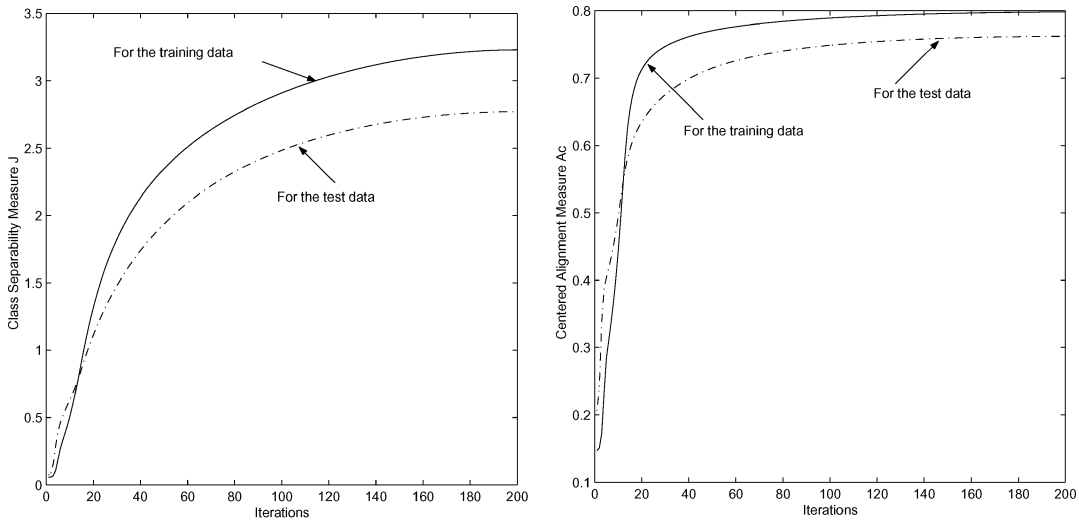


Fig. 6. (a) Class separability measure J of the *Ionosphere* training and test data as a function of the number of iterations. (b) Centered alignment measure A_c of the *Ionosphere* training and test data as a function of the number of iterations.

2) SVM: Given the training data set $\{x_i, \xi_i\}_{i=1}^m$, the support vector classifier (SVM) [6], [10], [16], [22] is designed to find the solution of the following quadratic programming problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & -\sum_{i=1}^{m} \alpha_i + \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j \xi_i \xi_j k(x_i, x_j) \\ \text{subject to} : & \sum_{i=1}^{m} \alpha_i \xi_i = 0 \\ & 0 \le \alpha_i \le C \quad i = 1, \ldots, m \end{aligned}$$

where $C > 0$ is a regularization constant. This quadratic programming can be solved by the interior point algorithm [28], and a more efficient implementation for the case of a large-scale matrix can be found in [13]. After solving the quadratic programming problem, the nonlinear discrimination function is constructed as

$$f(t) = \operatorname{sgn}\left(\sum_{i=1}^{m} \xi_i \alpha_i k(t, x_i) + b \right) \tag{20}$$

where b can be estimated by the so-called support vectors, as expained in [16]. In the experiments, we follow the method used in [18] to implement the SVM.

3) KFD: A kernel version of the Fisher discriminat has been proposed in [15]. However, for the sake of simplicity, we essentially follow the method given in [3] to summarize and implement the KFD algorithm.

Let S_b and S_w, respectively, denote the between-class and within-class scatter matrices of the training data, and S_m be the variance matrix of the training data, then, $S_m = S_b + S_w$. Under the class separability criterion of $\operatorname{tr}(S_m^{-1} S_b)$ [12], which is proven to be equivalent to the frequently used criterion of $\operatorname{tr}(S_w^{-1} S_b)$ [12], the optimal projection direction $\mathbf{w}$ that maximizes $\operatorname{tr}(S_m^{-1} S_b)$ is proportional to $S_m^{-1}(\mu_2 - \mu_1)$, where μ_1 and μ_2 are the means of the two classes. Let μ denote the mean of the entire training data. According to the derivation in [3], and the so-called kernel trick, the discrimination function, after modifying the center of the data to the origin, can be written as

$$f(t) = \operatorname{sgn}\left(m \lambda^T K \beta^T \Omega^{-2} \beta (t_X - \mu_X) \right)$$

where the m-dimensional vector λ is the modified label vector in which values -1 and $+1$ are divided by the sample size of each class, $\beta \Omega \beta^T$ is the eigen decomposition of the centered matrix K_c, t_X denotes the vector $(k(t, x_1), k(t, x_2), \ldots, k(t, x_m))^T$, and μ_X equals to $(1/m) K 1_m$.

4) KMSE: A kernel version of the minimum squared error machine, which is based on matrix pseudoinversion, has been proposed in [3]. The formula derived in [3] is suitable only when the center of the training data in the feature space is located at the origin, that is, $(1/m) \sum_{i=1}^{m} \Phi(x_i) = 0$. However, we can easily extend the formula in [3] to the general situation.

First, we need to solve a general linear system, $Xw + w_0 = b$, to minimize the MSE cost

$$J(w, w_0) = ||Xw + w_0 - b||^2$$

where X is the sample matrix defined in Section II, w denotes the d-dimensional weight vector, and b the vector of the associated class labels. Let $\widetilde{X} \triangleq (1_m \; X)$, and $\widetilde{w} \triangleq (w_0, w^T)^T$. Then the general linear system becomes

$$\widetilde{X} \widetilde{w} = b$$

with the following function to be minimized:

$$J(\widetilde{w}) = ||\widetilde{X} \widetilde{w} - b||^2.$$

The solution is

$$\widetilde{w}^* = \widetilde{X}^+ b$$

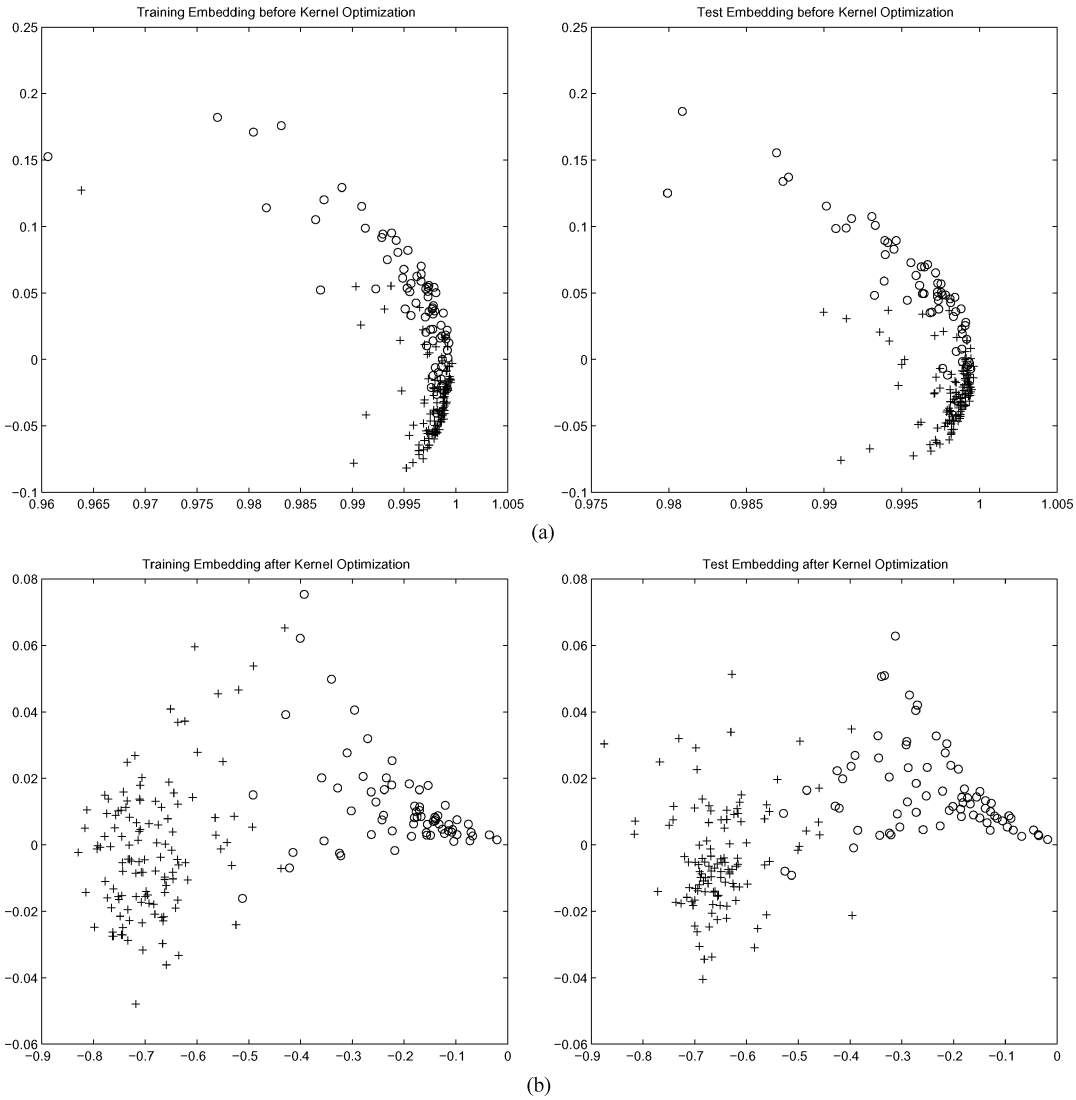


Fig. 7. Class separability for the *Breast Cancer* training and test data. (a) Two-dimensional projection of the training and test data before the kernel optimization. (b) Two-dimensional projection of the training and test data after the kernel optimization.

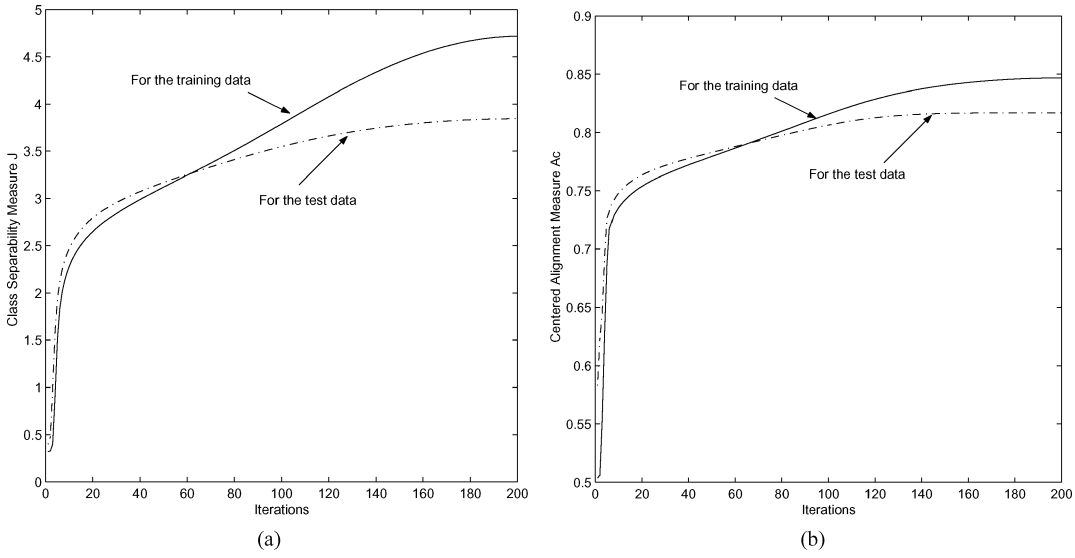


Fig. 8. (a) Class separability measure J of the *Breast Cancer* training and test data as a function of the number of iterations. (b) Centered alignment measure A_c of the *Breast Cancer* training and test data as a function of the number of iterations.

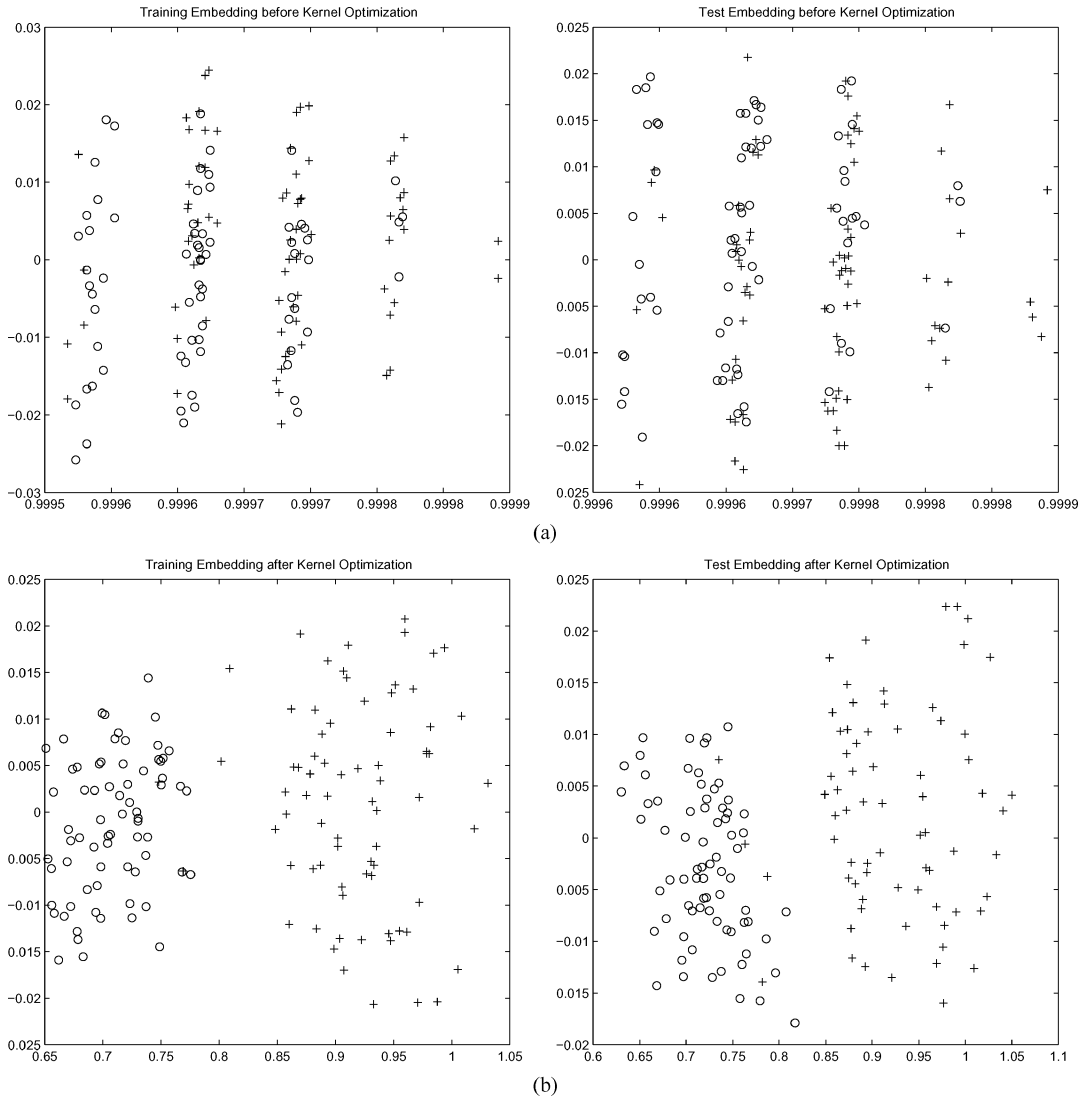


Fig. 9. Class separability for the *Monks3* training and test data. (a) Two-dimensional projection of the training and test data before the kernel optimization. (b) Two-dimensional projection of the training and test data after the kernel optimization.

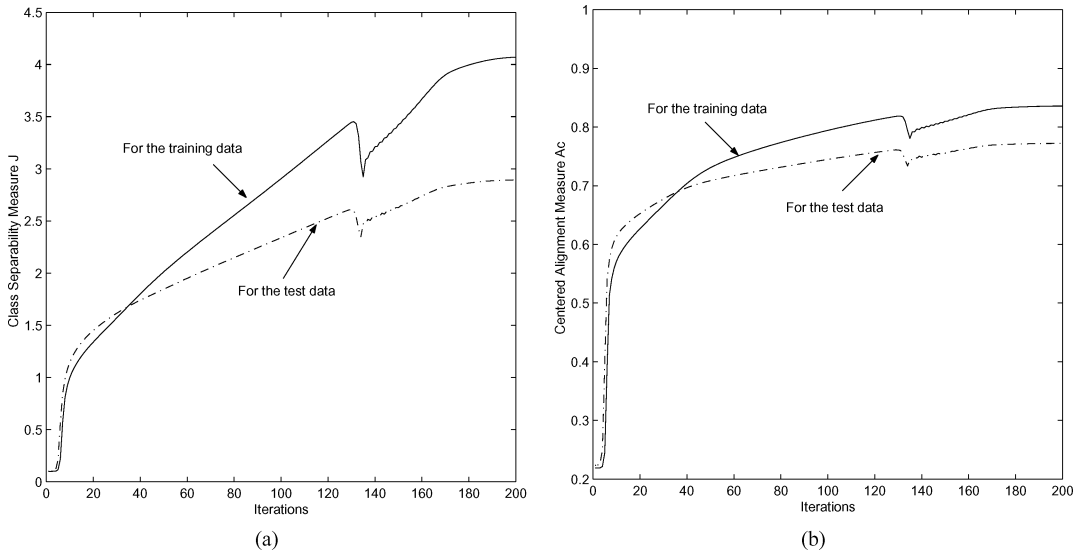


Fig. 10. (a) Class separability measure J of the *Monks3* training and test data as a function of the number of iterations. (b) Centered alignment measure A_c of the *Monks3* training and test data as a function of the number of iterations.

TABLE I
BASIC INFORMATION ABOUT THE UCI DATA SETS, WHERE N_s STANDS FOR THE NUMBER OF THE DATA POINTS IN EACH DATA SET, AND d DENOTES THE DIMENSION OF THE DATA

	Ionosphere	Breast	Liver	Pima	Monks1	Monks2	Monks3	Heart
N_s	351	569	345	768	432	432	432	297
d	34	30	6	8	6	6	6	13

where $\widetilde{X}^+$ is the pseudoinverse of the rectangular matrix $\widetilde{X}$. Using the so-called kernel trick as in [3], we obtain the general discrimination function

$$f(t) = \operatorname{sgn}\left((1_m + t_X)^T (1_m 1_m^T + K)^+ b\right). \qquad (21)$$

Comparing (21) with the formula in [3], we only need to modify the vector t_X and the matrix K by adding unity to each of the entries in t_X and K, respectively. The KMSE and KFD algorithms, which are based on the pseudoinverse calculation, have a disadvantage of numerical instability. To alleviate it, a threshold E is employed to discard the eigenvalues of K that are less than the threshold E.

B. Experimental Results

In this section, we conduct three sets of experiments in order to investigate the effect of using the optimized kernel in the classification algorithms, and to compare our kernel optimization with the alignment-based kernel adaptation [9] in terms of the improvements in the performance of the classification algorithms. In the first set of experiments, we study the performance of the classifiers under different parameter settings. The second set of experiments examines the effect on the performance when the parameters are chosen by cross validation. The final set of experiments compares our kernel optimization method with the alignment-based kernel adaptation method.

Eight data sets, namely, the *Ionosphere*, *Wisconsin Breast cancer*, *Liver disorder*, *Cleveland Heart disease* (where we have discarded the six instances containing missing values), *Pima Indians diabetes*, and the three monks data sets (*Monks1, Monks2, Monks3*), are adopted from the UCI benchmark repository [5] to test our algorithm. These eight data sets have been chosen, since they present different degrees of difficulty from the point of view of data classification. Table I presents some basic information about these data sets.

We only consider the Gaussian kernel function. As done in Section III-E, each of these eight data sets is first normalized to a distribution with zero mean and unit variance, and then randomly partitioned into three equal and disjoint parts. One part is used as the empirical core set $\{a_i\}$, and the other two as the training and test data sets. Besides the basic kernel parameter γ_0, the number of the nearest neighbors k for KNN, the regularization constant C for SVM, and the threshold values of E for the KMSE and KFD algorithms, need to be set in advance. As for the initial learning rate η_0 in (15) and the iteration number N, we again set them to 0.01 and 200, respectively, for all the data sets.

In the first set of experiments, the parameter k for KNN, C for SVM, E for KMSE, and E for KFD are set to 3, 10^3, 10^{-3}, and 10^{-3}, respectively. Tables II and III compare the average error rates of the classification algorithms on the *Ionosphere* data set before and after the kernel optimization for various settings of γ_0 and γ. In these tables, K_{Gauss} stands for the ordinary Gaussian kernel, and K_{opt} represents the optimized kernel. Table II presents the experimental results for the training data, and Table III the corresponding results for the test data. The values in the tables are the average error rates calculated over twenty trials, and in each trial, all the classification algorithms operate on the same training/test data partitions. The experimental results for the *Breast* and *Monks1* data sets are given in Tables IV–VII. It can be seen that the use of the optimized kernel substantially improves the performance of the classification algorithms. In particular, for the KNN algorithm, the improvement in the performance is most remarkable and this is due to an improved local relevance of the data in the feature space.

In the second set of experiments, we employ cross validation to choose the parameters. We choose k from {1, 3, 5, 7, 9}, C from {10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7}, E from $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$, γ_0 from {0.000 01, 0.000 05, 0.0001, 0.0005,0.001, 0.005, 0.01 ,0.05, 0.1, 0.5, 1.0}, and γ from {0.001, 0.005, 0.01, 0.05, 0.1 ,0.5, 1.0}. When the Gaussian kernel is used, which means without kernel optimization, we need to select k for KNN, (C, γ_0) for SVM, and (E, γ_0) for both KMSE and KFD. For the proposed kernel-optimization-based data classification, more parameters need to be chosen: (k, γ_0, γ) for KNN, (C, γ_0, γ) for SVM, and (E, γ_0, γ) for both KMSE and KFD. Among the three subsets for each data set, the one that is used as the empirical core set is also employed as the validation data set for choosing the optimal parameters. Then the algorithm with these optimal parameter settings is operated upon the test data.

Table VIII gives the average error rates on the test data over ten trials. We see that for the KNN algorithm there is a remarkable improvement in the performance for all the data sets, except for the *Heart* data. For the KMSE, KFD, and SVM algorithms, the improvement is still substantial for most of the data sets. However, we also find that for a sophisticated algorithm such as SVM or KMSE, the improvement resulting from the use of the optimized kernel is not always significant. It is seen that, for a sophisticated algorithm such as SVM, increasing the class separability of the data in the feature space does not necessarily lead to a significant improvement of the classification performance.

Finally, we compare our kernel optimization algorithm with the alignment-based kernel adaptation algorithm [9] in terms of their performance. For the sake of simplicity, we only compare the performance of the KNN algorithm ($k = 3$) for three kernels, namely, the Gaussian kernel K_{Gaus}, the alignment-based adapted kernel K_{alg}, and the optimized kernel K_{opt}. The parameters γ_0 for K_{alg}, and (γ_0, γ) for K_{opt} are chosen by cross validation as before. Table IX presents the experimental results in which the values are averaged over ten trials. We see that although the alignment-based method provides substantial improvement in the case of the training data, it provides a limited improvement in the case of the test data; in the case of the test data, the performance could even be worse than that achieved using the Gaussian kernel.

TABLE II
ERROR RATES FOR THE TRAINING SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE IONOSPHERE DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.0001, 0.01)$	17.39	6.62	4.83	2.52	6.54	3.38	3.39	1.97
$(0.0001, 0.05)$	17.39	4.96	4.83	2.18	6.54	3.68	3.39	2.18
$(0.0005, 0.01)$	17.39	7.86	2.31	0.81	3.72	1.52	1.07	0.38

TABLE III
ERROR RATES FOR THE TEST SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE IONOSPHERE DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.0001, 0.01)$	16.67	7.44	15.26	8.29	14.79	8.68	12.18	7.39
$(0.0001, 0.05)$	16.67	6.67	15.26	5.73	14.79	8.25	12.18	5.68
$(0.0005, 0.01)$	16.67	8.03	11.97	11.07	11.88	10.73	10.04	8.96

TABLE IV
ERROR RATES FOR THE TRAINING SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE BREAST CANCER DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.0001, 0.01)$	4.89	3.03	2.95	2.24	3.92	3.37	1.45	1.34
$(0.0001, 0.05)$	4.89	3.11	2.95	1.71	3.92	3.55	1.45	1.42
$(0.0005, 0.01)$	4.89	3.84	2.26	1.95	2.82	2.71	0.95	0.89

TABLE V
ERROR RATES FOR THE TEST SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE BREAST CANCER DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.0001, 0.01)$	4.50	3.18	3.89	3.68	5.16	4.29	2.97	2.71
$(0.0001, 0.05)$	4.50	3.84	3.89	3.39	5.16	5.00	2.97	3.07
$(0.0005, 0.01)$	4.50	3.55	4.42	3.45	5.47	3.92	3.58	3.34

TABLE VI
ERROR RATES FOR THE TRAINING SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE MONKS1 DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.00005, 0.05)$	29.31	13.61	32.71	11.49	32.81	11.56	30.07	10.66
$(0.0001, 0.05)$	29.31	13.58	32.71	13.78	32.81	13.51	29.27	11.91
$(0.0001, 0.1)$	29.31	14.93	32.71	10.45	32.81	11.11	29.10	9.97

TABLE VII
ERROR RATES FOR THE TEST SET BEFORE AND AFTER THE PROPOSED KERNEL OPTIMIZATION IN THE CASE OF THE MONKS1 DATA

(γ_0, γ)	KNN ($k=3$)		KMSE ($E=10^{-3}$)		KFD ($E=10^{-3}$)		SVM ($C=10^3$)	
	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{opt}(\%)$
$(0.00005, 0.05)$	30.24	17.12	34.51	17.60	33.72	16.81	31.98	15.35
$(0.0001, 0.05)$	30.24	16.15	34.51	19.93	33.72	18.96	31.60	16.88
$(0.0001, 0.1)$	30.24	19.72	34.51	18.23	33.72	17.88	31.60	15.97

TABLE VIII
ERROR RATES AND THE IMPROVEMENT IN ERROR RATES FOR VARIOUS TEST SETS USING CROSS-VALIDATION TO CHOOSE THE PARAMETERS

	KNN			KMSE			KFD			SVM		
	K_{Gaus} (%)	K_{opt} (%)	Impr. (%)	K_{Gaus} (%)	K_{opt} (%)	Impr. (%)	K_{Gaus} (%)	K_{opt} (%)	Impr. (%)	K_{Gaus} (%)	K_{opt} (%)	Impr. (%)
Ionosphere	16.41	6.58	59.9	7.27	5.81	20.1	10.94	6.32	42.2	6.73	5.27	21.7
Breast	4.21	3.21	23.8	4.03	3.38	16.1	4.95	4.06	18.0	3.10	2.98	3.9
Liver	40.74	34.48	15.4	33.78	32.00	5.3	35.57	31.83	10.5	29.30	28.91	1.4
Pima	29.49	26.52	10.1	23.91	22.66	5.2	28.09	27.34	2.7	21.78	21.54	1.1
monks1	29.31	16.04	45.3	16.25	15.63	3.8	17.78	17.04	4.2	9.93	7.28	26.7
Monks2	29.72	20.49	31.1	26.25	24.42	7.0	25.90	25.62	1.1	25.69	15.07	41.3
monks3	19.51	3.40	82.6	3.15	2.29	27.3	2.43	1.67	31.3	3.82	2.78	27.2
Heart	20.71	20.60	0.5	20.60	19.59	4.9	19.50	19.20	1.5	17.89	15.82	11.6

TABLE IX
ERROR RATES FOR THE KNN ALGORITHM FOR VARIOUS KERNELS

	Training Error Rate			Test Error Rate		
	$K_{Gaus}(\%)$	$K_{alg}(\%)$	$K_{opt}(\%)$	$K_{Gaus}(\%)$	$K_{alg}(\%)$	$K_{opt}(\%)$
Ionosphere	14.93	8.80	6.93	15.96	15.04	6.68
Breast	4.05	3.97	3.26	4.21	7.34	3.42
Liver	38.52	27.48	27.31	42.61	47.28	35.73
Pima	26.95	24.84	25.87	28.36	33.05	26.08
Monks1	25.35	12.07	13.33	28.31	16.97	15.73
Monks2	26.18	23.13	19.06	29.10	35.49	21.58
Monks3	14.23	3.81	2.50	18.38	7.34	3.27

V. CONCLUSION

We have presented in this paper a new approach of kernel optimization by maximizing a measure of the class separability in the empirical feature space. The main contributions of this paper can be summarized as follows.

1) We have defined a new space called the empirical feature space, a Euclidean space in which the data is embedded in such a way that the geometrical structure of the data in the feature apce is preserved. Compared with the feature space, the empirical feature space provides a more convenient framework to investigate the spatial distribution of the data in the feature space, to measure the class separability of the data in the feature space, and more importantly, to study how to improve this separability.
2) Inspired by the work contained in [2], we have present a general form of data-dependent kernel. Moreover, we have derived an effictive algorithm for optimizing the data-dependent kernel by maximizing the class linear separability of the data in the empirical feature space. With the optimized kernel, the data set in the feature space possesses a higher level of class linear separability and, therefore, a further improvement in the performance of the kernel machines can be achieved, since the kernel machines are all essentially linear machines in the feature space.
3) Based on the relationship we have established between the kernel matrices and the projection-independent measure J of the class separability in the empirical feature space, we have developed an updating algorithm to maximize the measure J.

Besides, we have discussed the close relation between the class separability measure and the alignment measure defined by Critanini *et al.* [9]. Our experiments confirm that the optimized kernel is more adaptive to both the training and test data, and leads to a substantial, sometimes significant, improvement in the performance of various data classification algorithms.

REFERENCES

[1] Kernel Machines (2002). [Online]. Available: http://www.kernel-machines.org
[2] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Netw.*, vol. 12, no. 6, pp. 783–789, 1999.
[3] A. Ruiz and P. E. Lopez-de Teruel, "Nonlinear kernel-based statistical pattern analysis," *IEEE Trans. Neural Netw.*, vol. 12, no. 1, pp. 16–32, Jan. 2001.
[4] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computat.*, vol. 12, no. 10, pp. 2385–2404, 2000.
[5] C. Blake, E. Keogh, and C. J. Merz. (1998) UCI Repository of Machine Learning Databases. Dept. Inform. Comput. Sci., Univ. California, Irvine, CA. [Online]. Available: http://www.ics.uci.edu/mlearn
[6] C. J. C. Burges, "Geometry and invariance in kernel based methods," in *Advance in Kernel Methods, Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1998.
[7] C. J. C. Burges and B. Scholkopf, "Improving the accuracy and speed of support vector learning machines," in *Advance in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 375–381.
[8] G. C. Cawley. (2000) MATLAB Support Vector Machine Toolbox. School of Inform. Syst., Univ. East Anglia, Norwich, Norfolk, U.K.. [Online]. Available: http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox
[9] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel target alignment," in *Proc. Neural Information Processing Systems (NIPS'01)*, pp. 367–373.
[10] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[11] J. H. Friedman, "Flexible metric nearest neighbor classification," Dept. Statist., Stanford Univ., Stanford, CA, 1994.
[12] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego: Academic, 1990.
[13] T. Joachims, "Making large-scale SVM learning practical," in *Advance in Kernel Methods-Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 169–184.
[14] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinte programming," *J. Mach. Learn. Res.*, vol. 5, 2004.

[15] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. Piscataway, NJ: IEEE, 1999, pp. 41–48.
[16] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
[17] E. Pekalska, P. Paclik, and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *J. Mach. Learn. Res.*, vol. 2, pp. 175–211, 2001.
[18] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*, B. Scholkopf, C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.
[19] G. Ratsch, S. Mika, B. Scholkopf, and K.-R. Muller, "Constructing boosting algorithms from SVMS: An application to one-class classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sep. 2002.
[20] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," in *Advance in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K.-R. Muller, Eds. Cambridge, MA: MIT Press, 2000, pp. 568–574.
[21] B. Scholkopf, "The kernel trick for distance," in *Proc. Neural Information Processing Systems Conf.*, Vancouver, BC, Canada, 2000.
[22] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, Sep. 1999.
[23] B. Scholkopf, A. Smola, and K.-R. Muller, "Kernel principal component analysis," in *Advance in Kernel Methods, Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999.
[24] J. A. K. Suykens, T. Van Gested, J. Vandewalle, and B. De Moor, "A support vector machines formulation to PCA analysis and its kernel version," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 447–450, Mar. 2003.
[25] B. Scholkopf, A. J. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computat.*, vol. 10, pp. 1299–1319, 1998.
[26] J. Peng, D. R. Heisterkamp, and H. K. Dai, "LDA/SVM driven nearest neighor classification," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 940–942, Jul. 2003.
[27] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Learning with the optimized data-dependent kernel," in *Proc. IEEE Workshop Learningn Computer Vision Pattern Recognition*, Washington, DC, Jun. 2004, pp. 95–98.
[28] R. J. Vanderbei, "An interior point code for quadratic programming," Princeton Univ., Princeton Univ., Tech. Rep. SOR-94-15, 1994.
[29] A. B. A. Graf, A. J. Smola, and S. Borer, "Classification in a normalized feature space using support vector machines," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 597–605, May 2003.

Huilin Xiong received the B.Sc and M.Sc. degrees in mathematics from Wuhan University, Wuhan, China, in 1985 and 1988, respectively, and the Ph.D. degree in pattern recognition and intelligent control from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 2000.

His research interests include neural networks, pattern recognition, machine learning, and wavelet-based image analysis and compression. Currently, he is a Postdoctoral Researcher in the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada.

M. N. S. Swamy (S'59–M'62–SM'74–F'80) received the B.Sc. (Hons.) degree in mathematics from Mysore University, India, in 1954, the Diploma in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1957, the M.Sc. and Ph.D. degrees in electrical engineering from the University of Saskatchewan, Saskatoon, Canada, in 1960 and 1963, respectively, and he was awarded the Doctor of Science in engineering (Honoris Causa) by Ansted University in recognition of his exemplary contributions to the research in electrical and computer engineering and to engineering education, as well as his dedication to the promotion of signal processing and communications applications, in 2001.

He is presently a Research Professor and the Director of the Center for Signal Processing and Communications in the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada, where he served as the Chair of the Department of Electrical Engineering from 1970 to 1977, and Dean of Engineering and Computer Science from 1977 to 1993. Since 2001, he holds the Concordia Chair (Tier I) in Signal Processing. He has also taught in the Electrical Engineering Department, Technical University of Nova Scotia, Halifax, and the University of Calgary, Calgary, Canada, as well as in the Department of Mathematics at the University of Saskatchewan. He has published extensively in the areas of number theory, circuits, systems and signal processing, and holds four patents. He is the coauthor of two book chapters and three books: *Graphs, Networks and Algorithms* (New York: Wiley, 1981), *Graphs: Theory and Algorithms* (New York: Wiley, 1992), and *Switched Capacitor Filters: Theory, Analysis and Design* (Englewood Cliffs, NJ: Prentice-Hall, 1995). A Russian translation of the first book was published by Mir Publishers, Moscow, in 1984, while a Chinese version was published by the Education Press, Beijing, in 1987.

Dr. Swamy is a Fellow of the Institute of Electrical Engineers (U.K.), the Engineering Institute of Canada, the Institution of Engineers (India), and the Institution of Electronic and Telecommunication Engineers (India). He is a Member of Micronet, a National Network of Centers of Excellence in Canada, and also its coordinator for Concordia University. Presently, he is the Past President for the CAS Society. He has served the IEEE CAS Society in various capacities such as President-Elect (2003), President (2004), Vice President (Publications) from 2001 to 2002 and as Vice-President in 1976, Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I from 1999 to 2001, Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1985 to 1987, Program Chair for the 1973 IEEE CAS Symposium, General Chair for the 1984 IEEE CAS Symposium, Vice-Chair for the 1999 IEEE CAS Symposium, and a Member of the Board of Governors of the CAS Society. He is the recipient of many IEEE-CAS Society awards including the Education Award in 2000, Golden Jubilee Medal in 2000, and the 1986 Guillemin-Cauer Best Paper Award.

M. Omair Ahmad (S'69–M'78–SM'83–F'01) received the B.Eng. degree from Sir George Williams University, Montreal, QC, Canada, and the Ph.D. degree from Concordia University, Montreal, QC, Canada, both in electrical engineering.

From 1978 to 1979, he was a Member of the Faculty of the New York University College, Buffalo. In 1979, he joined the Faculty of Concordia University, where he was an Assistant Professor of Computer Science. Subsequently, he joined the Department of Electrical and Computer Engineering of the same university, where presently he is a Professor and Chair of the department. He has published extensively in the area of signal processing and holds three patents. His current research interests include the areas of multidimensional filter design, image and video signal processing, nonlinear signal processing, communication DSP, artificial neural networks, and VLSI circuits for signal processing. He is a Researcher in the Micronet National Network of Centers of Excellence and was previously an Examiner of the Order of Engineers of Quebec.

Dr. Ahmad was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I: FUNDAMENTAL THEORY AND APPLICATIONS from 1999 to 2001. He was the Local Arrangements Chairman of the 1984 IEEE International Symposium on Circuits and Systems. During 1988, he was a Member of the Admission and Advancement Committee of the IEEE. Presently, he is the Chairman of the IEEE Circuits and Systems Chapter (Montreal Section). He is a recipient of the Wighton Fellowship from the Sandford Fleming Foundation.