Learning Optimal Kernel for Pattern Classification

MINMIN GE
Liaocheng University
School of Mathematics Sciences
Liaocheng, 252059
P. R. CHINA
geminmin2008@163.com

LIYA FAN
Liaocheng University
School of Mathematics Sciences
Liaocheng, 252059
P. R. CHINA
Corresponding author:fanliya63@126.com

Abstract: Kernel methods provide an efficient mechanism to derive nonlinear algorithms. Using a kernel function, original data can be implicitly mapped to a very high or even infinite dimensional feature space where the data is approximately linearly separable. For it, a main challenge is to select an appropriate kernel. In this paper, we optimize combinative weight coefficients and combination kernel is constructed by two methods. one method is learning optimal kernel for kernel fisher discriminant analysis (KFDA) for finding optimally combinative weight coefficients. In this method, we treat optimizing combinative weight coefficients as optimization problem over the convex set of finitely many basic kernels. Besides, in order to solve the optimization problem, we use a new iterative method. Another method is a feature space based class separability measure which is introduced in order to further show the efficacy of combination kernel .With this measure, the weight coefficients of combination kernel were optimized. Experiments on five real-words data sets are performed to test and evaluate the efficacy of combination kernel on classification accuracy. The results show that the efficacy of combination kernel is very significant.

Key–Words: Fisher discriminant analysis; kernel function; support vector machines; combination kernel; kernel optimization; iterative method;

1 Introduction

In the area of pattern recognition, kernel methods have attracted much attention [1-5]. The kernel machine technique has been widely used to tackle complicated classification problems by a nonlinear mapping from the original input space to a kernel feature space. Although in general the dimensionality of the kernel feature could be arbitrarily large or even infinite, which makes direct analysis in this space very difficult, the nonlinear mapping can be specified implicitly by replacing the dot products in the kernel feature space with a kernel function defined in the original input space. Therefore, the key task of a kernel-based solution is to generalize the linear representation in the form of dots products. In kernel-based learning algorithms, choosing an appropriate kernel, which is model selection problem, is crucial to ensure good performance since the geometrical structure of the mapped samples is determined by the selected kernel and its parameters. Thus, this paper focuses on optimal combinative kernels for pattern classification in supervised setting, that is, considering to optimize combinative weight coefficients of combinative kernels.

KFDA finds a direction in feature space, defined implicitly by a kernel, onto which the projections of positive and negative classes are well separated in terms of Fisher discriminant ratio. The performance of KFDA depends on the choice of a kernel function. The kernel selection problem has been studied by several authors [6-10]. They proposed the use of a non-negative linear combination of kernels chosen from families of different kernel functions and found an optimal combination kernel by means of an optimization problem. With the optimal combination kernel, the prediction accuracy of a specific classification problem can be improved, which is the ultimate goal of classification.

In kernel combination methods, the semi-definite programming is usually used to learn the composite kernel matrix [6]. A hyper-kernel space is defined on the space of kernels in order to learn the composite kernel within a specific parametric family [11]. Gaussian kernels under a support vector machine (SVM) frame-work are combined into an expanded composite kernel [12] and a non-stationary kernel combination approach is presented which allows for variation on the relative weights of the base kernels among the input examples [13]. When using a kernel method for classification, such as nonlinear SVM, the performance of the algorithm highly depends on the data

distribution structure in kernel induced feature space.

SVM is a state-of-the-art machine learn method which has gained popularity due to promising performance in a wide range of applications [14-16]. The training algorithms based on SVMs try to find an optimal separating hyperplane by maximizing the margin between different class data. Delivering promising results makes the SVMs extensively applicable in many information processing tasks, including data classification, pattern recognition and function estimation. SVMs are ordinarily used as binary classifiers that separate the data space into two areas. The separating hyperplane is not explicitly given, which is represented by a small number of data points called support vectors (SVs).

The rest of this paper is organized as follows. In Section 2, we provide a brief review for linear discriminant analysis (LDA) and SVM. In Sections 3 and 4, we learn an optimal kernel for KFDA by means of an iterative method and class separability measure (CSM), respectively. In section 5, a series of experiments are carried out in order to show the classification performance of optimal combination kernels obtained by our methods. Finally, we conclude the paper in Section 6.

2 Related works

2.1 Fisher discriminant analysis (FDA)

FDA is a well known linear classification method, which mainly finds a direction that maximizes the projected class mean, while minimizing the projected class variance in this direction [17]. Given a training data $\{(x_i, y_i)\}_{i=1}^N$ with k classes, where $x_i \in R^n$ and $y_i \in \{1, \cdots, k\}$ is the class label of $x_i, i = 1, \cdots, N$. Let N_i is the sample number belonged to class i and $N = \sum_{i=1}^k N_i$. Let $X = [X_1, \cdots, X_k]$ and $X_i = [x_1^i, x_2^i, \ldots, x_{N_i}^i]$ with samples belonging to class i. LDA is to find a vector $\omega \in R^n$ by maximizing the following Fisher discriminant ratio:

$$J(\omega) = \frac{\omega^T S_b \omega}{\omega^T S_w \omega}, \tag{1}$$

where S_w and S_b are within-class and between-class scatter matrixes, respectively. They are defined by

$$S_b = \sum_{i=1}^{k} N_i (m_i - m)(m_i - m)^T,$$
$$S_w = \sum_{i=1}^{k} \sum_{x \in X_i} (x - m_i)(x - m_i)^T,$$

where $m = (1/N) \sum_{l=1}^N x_l$ and $m_i = 1/N_i \sum_{l=1}^{N_i} x_l^i$. Due to $S_t = S_b + S_w$, we can see that the criterion (1) is equivalent to

$$J(\omega) = \frac{\omega^T S_b \omega}{\omega^T S_t \omega},$$

where $S_t = \sum_{j=1}^N (x_i - m)(x_i - m)^T$ is the total scatter matrix.

2.2 Kernel support vector machine (KSVM)

SVM is a well-known machine learning method and used widely in many domains, such as classification and regression. We here consider soft margin KSVM [18-19] for binary classification problems. Let $\{(x_i, y_i)\}_{i=1}^N$ be a given sample set and $y_i \in \{1, -1\}$ is the class label of $x_i, i = 1, \cdots, N$. The samples x_i can be mapped into a high dimensional feature space H by using a nonlinear feature function $\phi : X \to H$ of a kernel function $k : R^n \times R^n \to R$. Soft margin KSVM is to find an optimal separating hyperplane $\omega \cdot \phi(x) + b = 0$ by solving the following optimization problem:

$$\begin{aligned} &\max_{\omega,b,\xi} \tfrac{1}{2} \| \omega \|_2^2 + C \sum_{i=1}^{N} \xi_i \\ &s.t. \quad y_i(\langle \omega, \phi(x_i) \rangle + b) \geq 1 - \xi_i, i = 1, 2, \ldots, N, \\ &\qquad \xi_i \geq 0, i = 1, 2, \ldots, N, \end{aligned}$$

where $\omega = \sum_{i=1}^N y_i \rho_i \phi(x_i)$ is a weight vector, $b = y_j - \sum_{i=1}^N y_i \rho_i k(x_j, x_i)$ is a bias, C is a regularization parameter and $\{\xi_i\}_{i=1}^N$ is slack variable. This optimization problem can be solved by solving its dual optimization problem:

$$\begin{aligned} &\min_{\rho} \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \rho_i \rho_j y_i y_j k(x_i, x_j) - \sum_{i=1}^{N} \rho_j \\ &s.t. \quad \sum_{i=1}^{N} y_i \rho_i = 0, i = 1, 2, \ldots, N, \\ &\qquad 0 \leq \rho_j \leq C, j = 1, 2, \ldots, N, \end{aligned}$$

where the kernel function $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ defines implicitly the feature mapping $\phi : X \to H$ and the weight ω and bias b can be determined based on the solution ρ_i, that is . Then, we can obtain a decision function $f(x) = \sum_{i=1}^N y_i \rho_i k(x_i, x_j) + b$ and a separating hyperplane $\sum_{i=1}^N y_i \rho_i k(x_i, x_j) + b = 0$. If $f(x) > 0$ for a sample x, we can predict that it belongs to class 1. Otherwise, to class -1.

3 Learning an optimal kernel with an iterative method

In this section, we learn an optimal kernel for KFDA by means of an iterative method.

3.1 Combination of kernels

A good combination kernel can perform better transformation from the initial data space to a higher dimensions feature space. A combination kernel is a linear combination of several basic kernels. Let $k_i, i = 1, \cdots, \gamma$ be γ positive basic kernels, then the combination kernel k can be defined as

$$k = \sum_{i=1}^{\gamma} \beta_i k_i$$

where β_i is combinative coefficients. The performance of the combination kernel mainly depends on the choice of combinative coefficients. So, optimization problem of combinative coefficients becomes very important.

3.2 Optimization of combinative coefficients

In this subsection, we use KFDA to select optimal combinative coefficients. KFDA is a kernel version of FDA [17,20-21]. For a given training data $\{(x_i, y_i)\}_{i=1}^N$ with k classes, where $x_i \in R^n$ and $y_i \in \{1, \cdots, k\}$ is the class label of $x_i, i = 1, \cdots, N$. Let N_i is the sample numbers belonged to class i and $N = \sum_{i=1}^k N_i$. Let $X = [X_1, \cdots, X_k]$ and $X_i = [x_1^i, \ldots, x_{N_i}^i]$ with samples belonging to class i. Let $k : R^n \times R^n \to R$ be a kernel function, $\phi : R^n \to H$ and H be the feature mapping and Reproduce Kernel Hilbert Space (RKHS) of the kernel k, respectively. By means of the feature mapping ϕ, the training data $\{x_i\}_{i=1}^N$ can be mapped into the RKHS H. Let $m_i^\phi = \frac{1}{N_i} \sum_{l=1}^{N_i} \phi(x_l^i)$ and $m^\phi = \frac{1}{N} \sum_{l=1}^{N} \phi(x_l)$ be class and global centroids of the data $\{\phi(x_i)\}_{i=1}^N$, respectively. Let $\phi(X) = [\phi(x_1), \ldots, \phi(x_N)], D = [\frac{1}{N}]_{N\times N}, B_i = [\frac{1}{N_i}]_{N_i \times N_i}$ and $B = diag(B_1, \ldots, B_k)$. Let $K = [k(x_i, x_j)]_{N\times N}$ denote the kernel matrix of the kernel k. The between-class and global scatter matrixes S_b^ϕ and S_t^ϕ of the data $\{\phi(x_i)\}_{i=1}^N$ are given by (see [22])

$$\begin{aligned} S_b^\phi &= \sum_{i=1}^{k} N_i (m_i^\phi - m^\phi)(m_i^\phi - m^\phi)^T \\ &= QBQ^T \\ &= \phi(X)(I - D)B(I - D)\phi(X)^T, \\ S_t^\phi &= \sum_{j=1}^{N} (\phi(x_j) - m^\phi)(\phi(x_j) - m^\phi)^T \\ &= QQ^T \\ &= \phi(X)(I - D)(I - D)\phi(X)^T, \end{aligned}$$

where $Q = [\phi(x_1) - m^\phi, \ldots, \phi(x_N) - m^\phi] = \phi(X)(I - D)$. The optimal transformation vector ω^ϕ for KFDA can be obtained by maximizing the Fisher criterion in RKHS H:

$$\max_{\omega^\phi \in H} \frac{{\omega^\phi}^T S_b^\phi \omega^\phi}{{\omega^\phi}^T S_t^\phi \omega^\phi}. \tag{2}$$

Since H is RKHS, we can let that

$$\begin{aligned} \omega^\phi &= \alpha_1 \phi(x_1) + \alpha_2 \phi(x_2) + \ldots + \alpha_N \phi(x_N) \\ &= \phi(X)\alpha, \end{aligned}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)^T$. We can deduce that the Fisher criterion (2) is equivalent to

$$\max_{\alpha \in R^N} \frac{\alpha^T u u^T \alpha}{\alpha^T N \alpha}, \tag{3}$$

where $u = K(I - D)B^{1/2}$ and $N = K(I - D)^2 K$. According to works in [17, 23-24], the optimization problem (3) is equivalent to

$$\begin{aligned} &\min_{\alpha} \{\alpha^T N \alpha + CP(\alpha)\} \\ &s.t.\ \alpha^T u = 2. \end{aligned} \tag{4}$$

where $2 \in R^N$ is the vector of all two. Putting $\xi = (I - D)K\alpha$, the problem (4) can be written as

$$\begin{aligned} &\min_{\alpha,\xi,b} \{\tfrac{1}{2} \| \xi \|^2 + \tfrac{1}{2}(\alpha^T \alpha + b^2)\} \\ &s.t.\ \xi = (I - D)K\alpha. \end{aligned} \tag{5}$$

Let $1b = (2I - D)K\alpha - y$, then $\xi = -K\alpha + 1b + y$ and the problem (5) can be transformed into

$$\begin{aligned} &\min_{\alpha,\xi,b} \{\tfrac{1}{2} \| \xi \|^2 + \tfrac{1}{2}(\alpha^T \alpha + b^2)\} \\ &s.t.\ K\alpha - 1b + \xi = y. \end{aligned} \tag{6}$$

where $y = [y_1, \ldots, y_N]^T$, $b \in R$, $1 \in R^N$ is the vector of all ones. The Lagrangian function associated with the problem (6) is given by

$$\begin{aligned} L(\xi, \alpha, b, P) = &\tfrac{1}{2} \| \xi \|^2 + \tfrac{1}{2}(\alpha^T \alpha + b^2) \\ &- P^T(K\alpha - 1b + \xi - y), \end{aligned} \tag{7}$$

where $P \in R^N$ is the Lagrangian multiplier vector. The Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions for the problem (7) is

$$\begin{aligned} &\nabla_\xi L(\xi, \alpha, b, P) = \xi - P = 0, \\ &\nabla_\alpha(\xi, \alpha, b, P) = \alpha - K^T P = 0, \\ &\nabla_P(\xi, \alpha, b, P) = K\alpha - 1b + \xi - y = 0, \\ &\nabla_b(\xi, \alpha, b, P) = b + 1^T P = 0. \end{aligned}$$

Consequently, we obtain the Wolfe dual problem of the problem (7) (see [25]):

$$\max_P P^T y - P^T(\tfrac{1}{2} K K^T + \tfrac{1}{2} 1 1^T - \tfrac{I}{2})P. \tag{8}$$

From practical considerations, we can assume that $-M1 \leq P \leq M1$, where M is a large positive (finite) number. Thus, the problem (8) can be rewritten as

$$\max_{-M1\leq P\leq M1} P^T y - P^T(\tfrac{1}{2}KK^T + \tfrac{1}{2}11^T - \tfrac{I}{2})P. \tag{9}$$

According to [17], we can interpret that the objection function of the problem (9) is an upper bound on the misclassification probability for a given kernel matrix K. Thus, solving (9) for a given kernel matrix is a way for optimizing an upper bound on error probability. We can transform the problem (9) into the following problem

$$\begin{aligned}\min_{V} \max_{-M1\leq P\leq M1} &\{2P^T y - P^T(11^T - I + V)P\}\\ s.t.\ & Trace(V) = m,\\ & V = \sum_{j=1}^{q} \beta_j G_j\\ & \beta_j \geq 0, j = 1, 2 \ldots, q,\end{aligned} \tag{10}$$

where $G = KK^T, Trace(G_j) = r_j > 0, r = (r_1, \ldots, r_q)^T, \beta = (\beta_1, \ldots, \beta_q)^T$, $V = \sum_{j=1}^{q} \beta_j G_j$ and $\beta^T r = m$. The problem (10) is also equivalent to

$$\begin{aligned}\min_{\beta^T r=m,\beta\geq 0} \max_{-M1\leq P\leq M1} & 2P^T y - P^T(11^T - I)P -\\ & P^T(\sum_{j=1}^{q} \beta_i G_i)P.\end{aligned} \tag{11}$$

By the standard min-max theorem, the problem (11) is equivalent to the following optimization problem

$$\begin{aligned}\max_{-M1\leq P\leq M1} \min_{\beta^T r=m,\beta\geq 0} & 2P^T y - P^T(11^T - I)P\\ & -P^T(\sum_{j=1}^{q} \beta_i G_i)P.\end{aligned} \tag{12}$$

In order to solve the problem (12), we use an iterative method to get optimal weight coefficient $\beta_1, \ldots, \beta_q$. For a fixed P, we can obtain the following optimization problem from the problem (12)

$$\begin{aligned}&\min_{\beta} \beta^T t\\ &s.t.\ \beta^T r = m, \beta \geq 0,\end{aligned} \tag{13}$$

where $t = [-P^T G_1 P, \ldots, -P^T G_q P]^T$. If an optimal solution of the problem (13) is $\overline{\beta}$, then the problem (12) becomes

$$\begin{aligned}&\max_{P}\{2P^T y - P^T(11^T - I + \sum_{j=1}^{q} \overline{\beta} G_j)P\}\\ &s.t. - M1 \leq P \leq M1.\end{aligned} \tag{14}$$

By solving iteratively problems (13) and (14), we can get a group of optimal combinative coefficients $\beta_1^*, \cdots, \beta_q^*$. We note that the kernel matrix corresponding to large $\beta_j^*, (j = 1, \cdots, q)$ are important in describing the data and are significant in characterizing of the properties and structure of the data, while kernel matrices corresponding to small coefficients β_j^* are not very significant. The specific algorithm is as follows

Algorithm 1.

• Step0. Initialization. Any fixed P^0 and a sufficiently small positive number ε. Let $n = 0$.

• Step1. solve the problem (13) with P^n and get a solution β^n.

• Step2. solve the problem (14) with β^n and get a solution P^{n+1}.

• Step3. solve the problem (13) with P^{n+1} and get a solution β^{n+1}.

• Step4. If $\|\beta^{n+1} - \beta^n\| \leq \varepsilon$, stop the algorithm and let $\beta^* = \beta^{n+1}$. Otherwise, fix $n \leftarrow n + 1$ and return Step 2.

4 Learning an optimal kernel with a class separability measure

In this section, we introduce another method for learning an optimal kernel by means of a class separability measure (see [24-25]), which is different from Algorithm 1. Given a binary data $\{(x_i, y_i)\}_{i=1}^N$ with $x_i \in R^n$ and $y_i \in \{1, -1\}$. Without loss of generality, let the first N_+ samples belong to positive class and the rest samples belong to negative class. Let N_+ and N_- denote the sample numbers of positive and negative classes, respectively, and $N = N_+ + N_-$. Let $k : R^n \times R^n \to R$ be a kernel function, $\phi : R^n \to H$ and H be the feature mapping and Reproduce Kernel Hilbert Space (RKHS) of the kernel k, respectively. By means of the feature mapping ϕ, the data $\{x_i\}_{i=1}^N$ can be mapped into the RKHS H. Let $\varphi_+ = \frac{1}{N_+}\sum_{i=1}^{N_+} \phi(x_i), \varphi_- = \frac{1}{N_-}\sum_{i=N_++1}^{N} \phi(x_i)$ and $\varphi = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)$. We denote by I_i the i-th column vector of the identity matrix and by $1_{N\times 1} \in R^N$ the vector consisting of 1. Consider a combinative kernel $k = \sum_{i=1}^{q} \beta_i k_i$, corresponding kernel matrix being $K = \sum_{i=1}^{q} \beta_i K_i$, and let $\beta = (\beta_1, \cdots, \beta_q)^T$. In RKHS H, the variance of data $\{\phi(x_i)\}_{i=1}^N$ is defined by

$$\begin{aligned}var &= \tfrac{1}{N}\sum_{i=1}^{N} \| \phi(x_i) - \varphi \|^2\\ &= \tfrac{1}{N}\sum_{i=1}^{N} [\phi(x_i)]^T\phi(x_i) - 2[\phi(x_i)]^T\varphi + \varphi^T\varphi\end{aligned}$$

$$= \frac{1}{N}\sum_{i=1}^{N} k(x_i,x) - \varphi^T\varphi$$
$$= \frac{1}{N}\sum_{i=1}^{N} k(x_i,x) - \sum_{i,j=1}^{N}\frac{1}{N}\frac{1}{N}k(x_i,x_j)$$
$$= \frac{1}{N}\sum_{i=1}^{N} I_i^T K I_i - \frac{1}{N^2}1^T K 1,$$

and the class separability measure (CSM) is defined by (for details, see [26-27])

$$\mathrm{CSM}(K) = \frac{var}{\|\varphi_+ - \varphi_-\|^2}.$$

We can deduce that

$$\begin{aligned}
&\| \varphi_+ - \varphi_- \|^2 \\
&= {\varphi_+}^T\varphi_+ - 2{\varphi_+}^T\varphi_- + {\varphi_-}^T\varphi_- \\
&= \frac{1}{{N_+}^2}\sum_{i=1}^{N_+}[\phi(x_i)]^T\sum_{j=1}^{N_+}\phi(x_j) \\
&\quad -\frac{2}{N_+N_-}\sum_{i=1}^{N_+}[\phi(x_i)]^T\sum_{j=N_++1}^{N}\phi(x_j) \\
&\quad +\frac{1}{{N_-}^2}\sum_{i=N_++1}^{N}[\phi(x_i)]^T\sum_{j=N_++1}^{N}\phi(x_i) \\
&= \sum_{i=1}^{N_+}\sum_{i=1}^{N_+}\frac{1}{{N_+}^2}k(x_i,x_j) \\
&\quad -2\sum_{i=1}^{N_+}\sum_{j=N_++1}^{N}\frac{1}{N_+N_-}k(x_i,x_j) \\
&\quad +\sum_{i=N_++1}^{N}\sum_{j=N_++1}^{N}\frac{1}{{N_-}^2}k(x_i,x_j) \\
&= \upsilon^T K \upsilon,
\end{aligned}$$

where $\upsilon \in R^N$ with the first N_+ elements being $1/N_+$ and the others being $1/N_-$. Consequently, the weight coefficients $\beta_1.\cdots,\beta_q$ of the combination kernel k are optimized by solving the following optimization problem

$$\beta_{opt} = arg\min_{\beta_i}\mathrm{CSM}[\sum_{i=1}^{q}\beta_i K_i]. \qquad (15)$$

We can't solve the problem (15) directly. So, an iterative method based on gradient is presented for finding the optimal combination weight coefficients. The specific algorithm is as follows (for details, see [26]).

Algorithm 2.

• Step0. Initialization. Let $\beta^0 = (1,0,\ldots,0)^T$, M be a given maximal iterative time, $\eta(t)$ be a learning rate and iterative time $t = 0$.

• Step1. Select $\eta(t) = 0.01[1 - \frac{t}{M}]$ and compute $\beta_i^{t+1} = \beta_i^t + \eta(t)[\frac{\partial CSM(K)}{\partial \beta^t}]_i\beta_i^t, i = 1,\cdots,q$, where $\frac{\partial CSM(K)}{\partial \beta^t} = \frac{\partial CSM(K)}{\partial \beta}|_{\beta^t}$.

• Step2. Normalize β^{t+1} and get $\Sigma_{i=1}^{q}\beta_i^{t+1} = 1$.

• Step3. If $t < M$, fixing $t = t+1$ and return step1. Or else, end cycling procedure and get the optimal weight coefficient $\beta^* = \beta^{t+1}$.

5 Experiments and analysis

In this section, in order to evaluate the efficacy of the proposed methods and illustrate the effect of combination kernel for classification result, we conduct a series of experiments with five different data sets and three kernel functions. We perform the experiments on E.coli, Iris, Wine, Yeast and Breast Cancer Wisconsin data Sets, respectively, which are taken from UCI machine learning repository [28]. The three kernel functions are respectively two Gaussian RBF kernels with different kernel parameters

$$k(x,y) = \exp(-\| x - y \|^2/\sigma^2), \sigma \geq 0$$

and a polynomial kernel

$$k_{poly}(x,y) = (< x.y > +c)^m, c \geq 0, m > 0.$$

In the experiments, We select $\eta(t) = 0.01[1 - \frac{t}{M}]$ for a sufficiently large natural number M. We don't consider the optimization problem of kernel parameters and select $\sigma \in \{10^{-3}, 10^{-2}, \ldots, 10^5\}$. We use the fivefold cross validation when using SVM models for binary-class classification problems.

5.1 Experiments on Iris data set

The Iris date set includes 3 classes, 4 features and 150 instances. In this experiment, we select 4 features, 100 instances and 2 classes: Iris Versicolour concluding 50 instances and Iris Virginica concluding 50 instances. Firstly, we consider a combination kernel function

$$k(x,y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 2$ and $\sigma = 10$:

$$k_1 = \exp(-\|x-y\|^2/2^2),$$
$$k_2 = \exp(-\|x-y\|^2/10^2),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using Algorithm 1, we get optimal weight coefficients $\beta_1 = 0.6732, \beta_2 = 0.3266, \beta_3 = 0.0002$. Because of weight coefficient $\beta_3 = 0.0002$, the performance of kernel k_3 is very small for this data sets . So we select kernel k_1 and kernel k_2 for combination kernel and normalize β_1 and β_2.then we get an optimal kernel function:

$$k(x,y) = 0.6733k_1 + 0.3267k_2. \qquad (16)$$

Secondly, we classify iris data using SVM with $C = 100$ and optimal kernel (16). The result can be found in Table 1.

Table 1. Misclassification rate of Iris data

Kernel	Kernel parameters	Error rate(%)
RBF	2	8
RBF	10	5
Poly	$0.01, 0.1$	30
Combination kernel	-	2

From Table 1, we can see that, for classification result, kernel k_2 is better than kernels k_1 and k_3 and k_1 is better than k_3. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

In Algorithm 2, firstly, we fix $M = 10^4$ and consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 0.1$ and $\sigma = 2$:

$$k_1 = \exp(-\|x - y\|^2/0.1^2),$$
$$k_2 = \exp(-\|x - y\|^2/2^2),$$

and a polynomial kernel with parameter $c = 2$ and $m = 0.1$:

$$k_3 = (< x.y > +2)^{0.1}.$$

By using the Algorithm 2, we get optimal weight coefficients $\beta_1 = 0.4276, \beta_2 = 0.3369, \beta_3 = 0.2355$. And we normalize β_1 and β_2. Then we get an optimal kernel function:

$$k(x, y) = 0.5593k_1 + 0.4407k_2. \qquad (17)$$

Secondly, we classify iris data using SVM with $C = 100$ and optimal kernel (17). The result can be found in Table 2.

Table 2. Misclassification rate of Iris data

Kernel	Kernel parameters	Error rate(%)
RBF	0.1	7
RBF	2	8
Poly	$2, 0.1$	27
Combination kernel	-	0

From Table 2, we can see that, for classification result, kernel k_1 is better than kernels k_2 and k_3 and k_2 is better than k_3. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

5.2 Experiment on E.coli data set

E.coli data set includes 8 classes, 7 features, 336 instances. We select 6 features, and 220 instances and 2 classes: cytoplasm concluding 143 instances and inner membrane without signal sequence concluding 77 instances in the experiment. Firstly, we consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 1$ and $\sigma = 100$:

$$k_1 = \exp(-\|x - y\|^2/1^2),$$
$$k_2 = \exp(-\|x - y\|^2/10^4),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using Algorithm 1, we get optimal weight coefficients $\beta_1 = 0.8543, \beta_2 = 0.0011, \beta_3 = 0.1446$. Because of weight coefficient $\beta_2 = 0.0011$, the performance of kernel k_2 is very small for this data sets . So we select kernel k_1 and kernel k_3 for combination kernel and normalize β_1 and β_3. Then we get an optimal kernel function:

$$k(x, y) = 0.8552k_1 + 0.1448k_3. \qquad (18)$$

Secondly, we classify e.coli data using SVM with $C = 2$ and optimal kernel (18). The result can be found in Table 3.

Table 3. Misclassification rate of E.Coli data

Kernel	Kernel parameters	Error rate(%)
RBF	1	22.29
RBF	100	47.71
Poly	$0.01, 0.1$	46.84
Combination kernel	-	14.57

From Table 3, we can see that, for classification result, kernel k_1 is better than kernels k_2 and k_3 and k_3 is better than k_2. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

In Algorithm 2, firstly, we fix $M = 10^3$ and consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 1000$ and $\sigma = 100$:

$$k_1 = \exp(-\|x - y\|^2/10^6),$$
$$k_2 = \exp(-\|x - y\|^2/10^4),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using the Algorithm 2, we get optimal weight coefficients $\beta_1 = 0.3333, \beta_2 = 0.3308, \beta_3 = 0.3359$ and then we get an optimal kernel function:

$$k(x, y) = 0.3333k_1 + 0.3308k_2 + 0.3359k_3. \quad (19)$$

Secondly, we classify e.coli data using SVM with $C = 100$ and optimal kernel (19). The result can be found in Table 4.

Table 4. Misclassification rate of E.Coli data

Kernel	Kernel parameters	Error rate(%)
RBF	1000	47.71
RBF	100	47.71
Poly	0.01, 0.1	38.57
Combination kernel	-	14.57

From Table 4, we can see that, for classification result, kernel k_3 is better than kernels k_1 and k_2. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

5.3 Experiment on Wine data set

Wine data set includes 178 instances, 13 features, 3 classes. we only use class 1 concluding 59 instances and class 2 concluding 71 instances. We have 130 instances, 11 features and 2 classes in this experiment. Firstly, we consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 2$ and $\sigma = 100$:

$$k_1 = \exp(-\|x - y\|^2/2^2),$$
$$k_2 = \exp(-\|x - y\|^2/10^4),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using Algorithm 1, we get optimal weight coefficients $\beta_1 = 0.7721, \beta_2 = 0.2279, \beta_3 = 0$ and then we get an optimal kernel function:

$$k(x, y) = 0.7721k_1 + 0.2279k_2. \quad (20)$$

Secondly, we classify wine data using SVM with $C = 100$ and optimal kernel (20). The result can be found in Table 5.

Table 5. Misclassification rate of Wine data

Kernel	Kernel parameters	Error rate (%)
RBF	2	41.6
RBF	100	28
Poly	0.01, 0.1	51.6
Combination kernel	-	5.2

From Table 5, we can see that, for classification result, kernel k_2 is better than kernels k_1 and k_3 and k_1 is better than k_3. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

In Algorithm 2, firstly, we fix $M = 10^3$ and consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 10000$ and $\sigma = 0.01$:

$$k_1 = \exp(-\|x - y\|^2/10^8),$$
$$k_2 = \exp(-\|x - y\|^2/0.01^2),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using the Algorithm 2, we get optimal weight coefficients $\beta_1 = \frac{1}{3}, \beta_2 = \frac{1}{3}, \beta_3 = \frac{1}{3}$ and then we get an optimal kernel function:

$$k(x, y) = \tfrac{1}{3}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3. \quad (21)$$

Secondly, we classify wine data using SVM with $C = 10^5$ and optimal kernel (21). The result can be found in Table 6.

Table 6. Misclassification rate of Wine data

Kernel	Kernel parameters	Error rate(%)
RBF	10000	41.2
RBF	0.01	47.6
Poly	0.01, 0.1	28.4
Combination kernel	-	14

From Table 6, we can see that, for classification result, kernel k_3 is better than kernels k_1 and k_2 and k_1 is better than k_2. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

5.4 Experiment on Yeast data set

Yeast data set includes 1484 instances, 8 features and 10 classes. We only use 214 instances, 8 features and 2 classes: membrane protein, uncleaved signal including 51 instances and membrane protein, cleaved signal including 163 instances in this experiment. Firstly, we consider a combination kernel function

$$k(x,y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 1$ and $\sigma = 100$:

$$k_1 = \exp(-\|x-y\|^2/1^2),$$
$$k_2 = \exp(-\|x-y\|^2/10^4),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using Algorithm 1, we get optimal weight coefficients $\beta_1 = 0.7253, \beta_2 = 0.2706, \beta_3 = 0.0041$. Because of weight coefficient $\beta_3 = 0.0041$, the performance of kernel k_3 is very small for this data sets. So we select kernel k_1 and kernel k_2 for combination kernel and normalize weight coefficients β_1, β_2. And then we get an optimal kernel function:

$$k(x,y) = 0.7283k_1 + 0.2717k_2. \qquad (22)$$

Secondly, we classify yeast data using SVM with $C = 10$ and optimal kernel (22). The result can be found in Table 7.

Table 7. Misclassification rate of Yeast data

Kernel	Kernel parameters	Error rate(%)
RBF	1	18.75
RBF	100	49.69
Poly	0.1, 0.01	49.69
Combination kernel	-	15.94

From Table 7, we can see that, for classification result, kernel k_1 is better than kernels k_2 and k_3 . However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

In Algorithm 2, firstly, we fix $M = 2000$ consider a combination kernel function

$$k(x,y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 2$ and $\sigma = 10$:

$$k_1 = \exp(-\|x-y\|^2/2^2),$$
$$k_2 = \exp(-\|x-y\|^2/10^2),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using the Algorithm 2, we get optimal weight coefficients $\beta_1 = 0.0405, \beta_2 = 0.3031, \beta_3 = 0.6564$. Because of weight coefficient $\beta_1 = 0.0405$, the performance of kernel k_1 is very small for this data set. So we select kernel k_2 and kernel k_3 for combination kernel and normalize β_2 and β_3. And then we get an optimal kernel function:

$$k(x,y) = 0.3159k_2 + 0.6841k_3. \qquad (23)$$

Secondly, we classify yeast data using SVM with $C = 100$ and optimal kernel (23). The result can be found in Table 8.

Table 8. Misclassification rate of Yeast data

Kernel	Kernel parameters	Error rate(%)
RBF	2	20.31
RBF	10	31.25
Poly	0.01, 0.1	26.56
Combination kernel	-	26.56

From Table 8, we can see that, for classification result, kernel k_1 is better than kernels k_2 and k_3 and k_3 is better than k_2. However, the combined kernel k is obviously superior kernels k_2.

5.5 Experiment on Breast Cancer Wisconsin data set

Breast Cancer data sets concludes 699 instances, 9 features and 2 classes. We use 9 features, 2 classes and 300 instances from the former 150 instances in class Benign and the former 150 instances in class Malignant. Firstly, we consider a combination kernel function

$$k(x,y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 0.01$ and $\sigma = 100$:

$$k_1 = \exp(-\|x-y\|^2/0.01^2),$$
$$k_2 = \exp(-\|x-y\|^2/10^4),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using Algorithm 1, we get optimal weight coefficients $\beta_1 = 0.1804, \beta_2 = 0.8195, \beta_3 = 0.0001$. Because of weight coefficient $\beta_3 = 0.0001$, the performance of kernel k_3 is very small for this data sets. So we select kernel k_1 and kernel k_2 for combination kernel and normalize weight coefficients β_1, β_2. And then we get an optimal kernel function:

$$k(x, y) = 0.1804k_1 + 0.8196k_2. \quad (24)$$

Secondly, we classify breast cancer wisconsin data using SVM with $C = 100$ and optimal kernel (24). The result can be found in Table 9.

Table 9. Misclassification rate of Breast Cancer Wisconsin data

Kernel	Kernel parameters	Error rate (%)
RBF	0.01	45.5
RBF	100	45
Poly	0.01, 0.1	33
Combination kernel	-	25

From Table 9, we can see that, for classification result, kernel k_3 is better than kernels k_1 and k_2 and k_2 is better than k_1. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

In Algorithm 2, firstly, we fix $M = 5000$ and consider a combination kernel function

$$k(x, y) = \beta_1 k_1 + \beta_2 k_2 + \beta_3 k_3$$

with two Gaussian RBF kernels with different parameters $\sigma = 0.1$ and $\sigma = 10$:

$$k_1 = \exp(-\|x - y\|^2/0.1^2),$$
$$k_2 = \exp(-\|x - y\|^2/10^2),$$

and a polynomial kernel with parameter $c = 0.01$ and $m = 0.1$:

$$k_3 = (< x.y > +0.01)^{0.1}.$$

By using the Algorithm 2, we get optimal weight coefficients $\beta_1 = 0.5103, \beta_2 = 0.2812, \beta_3 = 0.2085$. And then we get an optimal kernel function:

$$k(x, y) = 0.5103k_1 + 0.2812k_2 + 0.2085k_3. \quad (25)$$

Secondly, we classify breast cancer data using SVM with $C = 100$ and optimal kernel (25). The result can be found in Table 10.

Table 10. Misclassification rate of Breast Cancer Wisconsin data

Kernel	Kernel parameters	Error rate(%)
RBF	0.1	32.5
RBF	10	45.5
Poly	0.01, 0.1	33
Combination kernel	-	27.5

From Table 10, we can see that, for classification result, kernel k_1 is better than kernels k_2 and k_3 and k_3 is better than k_2. However, the combined kernel k is obviously superior kernels k_1, k_2 and k_3.

6 Conclusion

In this paper, we use two methods to show that the performance of combination kernel is very significant. We use a new iterative method instead of second order cone programming in learning the optimal kernel for KFDA. We find optimal weight coefficient of each kernel function, while the combination kernel is constructed. In order to further illustrate performance of the combination kernel , we use FCSM method which has fewer limitations in application of kernel optimization and has better theoretical guarantees compared with the kernel matrix evaluation measure (FSM). We conduct a series of experiments on 5 different data sets from UCI Machine Learning Repository and use SVM and optimal kernel for classification. Results show that combination kernel is superior to single kernel for improving classification accuracy. Hence, classification performance depends very much on the choice of the kernel function. The combination kernel is very important.

Acknowledgements: This work is supported by Natural Science Foundation of Shandong Province (ZR2009AL006) in China.

References:

[1] H. Xiong, M. Swamy, M. Ahmad, Optimizing the kernel in the empirical feature space, *IEEE Transactions on Neural Networks* 16, 2005, pp. 460-474.

[2] S.-J. Kim, A. Magnani, and S. Boyd, Optimal kernel selection in kernel fisher discriminant analysis. *In Proceedings of the 23rd International conference on Machine Learning*, 2006, pp. 465-472.

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

[4] Cao, Shen, Sun, Yang, Chen, Feature selection in kernel space, *In International conference on machine learning (ICML) Oregon*, 2007, pp. 121-128.

[5] X. Z. Liu, G. C. Feng, Multiple kernel discriminant analysis with optimized weight, *Journal of computer applications* 29,2009.

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

[7] F. Bach, G. Lanckriet, M. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, *In: Proceedings of the 21th International conference on Machine Learning (ICML)*, ACM, New York (USA), 2004.

[8] B. Hamers, J. Suykens, V. Leemans, B. D. Moor, Ensemble learning of coupled parameterized kernel models, *In: International Conference on Neural Information Processing*, 2003, pp. 130-133.

[9] Y. Yajima, Linear Programming approaches for multicategory support vector machines, *European Journal of Operational Research* 162, 2005, pp.514-531.

[10] S. Yang, S. Yan, D. Xu, X. Tang, C. Zhang, Fisher + kernel criterion for discriminant analysis, *In: CVPR 2005. IEEE Computer Society Conference on computer Vision and Pattern Recognition*, vol.2, CA(USA) 20-25, 2005, pp. 197-202.

[11] P. Pavlidis, J. Weston, J. Cai, N. W. Grundy, Gene functional classification from heterogenous data. *In: Fifth Annual International Conference on Computational Molecular Biology*, 2001, pp. 249-255.

[12] W.-J. Lee, S. Verzakov, R. P. Duin, Kernel combination versus classifier combination, *In: Seventh International Workshop on Multiple Classifier systems*, 2007.

[13] D. P. Lewis, T. Jebara, W. S. Noble, Nonstationary kernel combination. *In: The 23rd International Conference on Machine Learning*, 2006, pp.553-560.

[14] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning method, *Cambridge University Press*, 2000.

[15] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, *Cambridge University Press*, 2004.

[16] V. N. Vapnik, *The nature of statical learning theory*, New York: Springer-Verlag, 1995.

[17] Reshma Khemchandani, Jayadeva, Suresh Chandra, Learning the optimal kernel for Fisher discriminant analysis via second order cone programming, *European Journal of Operational Research* 203, 2010, pp. 692-697.

[18] S.-K Kim, Y. J. Park, Kar-Ann Toh, Sangyoun Lee, SVM-based feature extraction for face recognition. *Pattern recognition* 43, 2010, pp. 2871-2881.

[19] S. C. Gu, Y. Tan, X. G. He, Discriminant analysis via support vector, *Neurocomputing* 73, 2010, pp. 1669-1675.

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

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

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

[23] J. Yang, Kernel feature extraction methods observed from the viewpoint of generating- kernels, *Front. Electr. Electron. Eng. China* 6(1), 2011, pp. 43-45.

[24] S. Mika, G. Rätsch, K. Müller, A mathematical programming approach to the kernel Fisher algorithm, *In: Jai., M.(Ed.), Neural Information Processing Systems* 13, 2001, pp. 591-597.

[25] O. L. Mangasarian, Nonlinear Programming, *SIAM, Philadelphia*, PA, 1994.

[26] X. Yang, X.Yang, H. L. Xiong, A method for optimizing the combinational kernel of support vector machine classifier, *Journal of Shang Hai Jiao Tong university* 44, 2010.

[27] C. H. Nguyen, T. B. Ho, An efficient kernel matrix evaluation measure [J]. *Pattern Recognition* 41(11), 2008, pp. 3366-3372.

[28] A. Asuncion, D. Newman. *UCI machine learning repository* <http://www.ics.uci.edu/ mlearn/MLRepository.html>, School of Information and Computer Science, University of California, Irvine, 2007.