Optimal Kernel Selection in Kernel Fisher Discriminant Analysis

Seung-Jean Kim SJKIM@STANFORD.ORG
Alessandro Magnani ALEM@STANFORD.EDU
Stephen Boyd BOYD@STANFORD.EDU
Department of Electrical Engineering, Stanford University, Stanford, CA 94304 USA

Abstract

In Kernel Fisher discriminant analysis (KFDA), we carry out Fisher linear discriminant analysis in a high dimensional feature space defined implicitly by a kernel. The performance of KFDA depends on the choice of the kernel; in this paper, we consider the problem of finding the optimal kernel, over a given convex set of kernels. We show that this optimal kernel selection problem can be reformulated as a tractable convex optimization problem which interior-point methods can solve globally and efficiently. The kernel selection method is demonstrated with some UCI machine learning benchmark examples.

1. Introduction

Recently, KFDA has received a lot of interest in the literature (Mika et al., 2001; Yang et al., 1989). A main advantage of KFDA over other kernel-based methods is that it is computationally simple: it requires the factorization of the Gram matrix computed with given training examples, unlike other methods which solve dense (convex) optimization problems. The classification performance of KFDA is comparable to that of support vector machines (SVMs) (Mika et al., 2003), which are regarded as the state-of-the-art kernel methods.

KFDA finds the direction in a 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 (FDR). Like other kernel-based classification methods, its classification performance depends very much on the choice of the kernel. Typically, a parameterized family of kernels, *e.g.*, the Gaussian or polynomial kernel family, is chosen and the kernel parameters are tuned via cross-validation or generalized cross-validation (Hastie et al., 2001).

In this paper, we consider the problem of finding the kernel, over a given convex set of kernels, that is optimal in terms of maximum achievable FDR. The main contribution of this paper is to show that this kernel selection problem can be reformulated as a tractable convex optimization problem, and hence the globally optimal kernel can be found with efficiency. In particular when the convex kernel set consists of affine combinations of a finite number of given kernels, the optimal kernel selection problem can be cast as a semidefinite program (SDP) which interior-point methods can solve with great efficiency.

The kernel selection problem has been studied by Fung et al. (2004). The authors formulate an optimal kernel selection problem, based on the quadratic programming formulation of Fisher linear discriminant analysis given in Mika et al. (2001). This optimal kernel selection problem is not jointly convex in the variables (the feature weights and Gram matrix). They develop an iterative method that alternates between optimizing the weight vector and the Gram matrix, without exploiting the fact that the problem can be reformulated as a convex problem.

Recently, Micchelli and Pontil (2005) have shown that, for a general class of kernel-based classification methods, the associated optimal kernel selection problems are in fact convex problems. The optimal kernel selection problem in KFDA does not fall into the class, so our convex formulation of the problem does not follow from the general result.

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

1.1. Outline

In the remainder of this section, we introduce some notation and definitions. We review KFDA in §2. We describe the optimal kernel selection problem in KFDA and give its convex formulation in §3. The kernel selection method is demonstrated with some UCI machine learning benchmark examples in §4. We review related work on optimal kernel selection in kernel-based methods in §5 including the general result in Micchelli and Pontil (2005). We give our conclusions in §6.

1.2. Notation and Definitions

We use $\mathcal{X}$ to denote the input or instance set, which is an arbitrary subset of $\mathbb{R}^n$, and $\mathcal{Y} = \{-1, +1\}$ to denote the output or class label set. An input-output pair (x, y) where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ is called an example. An example is called positive (negative) if its class label is $+1$ (-1). We assume that the examples are drawn randomly and independently from a fixed, but unknown, probability distribution over $\mathcal{X} \times \mathcal{Y}$.

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel (function)* if it satisfies the finitely positive semidefinite property: for any $x_1, \ldots, x_m \in \mathcal{X}$, the *Gram matrix* $G \in \mathbb{R}^{m \times m}$, defined by

$$G_{ij} = K(x_i, x_j), \tag{1}$$

is positive semidefinite. Mercer's theorem (Shawe-Taylor & Cristianini, 2004) tells us that any kernel function K implicitly maps the input set $\mathcal{X}$ to a high-dimensional (possibly infinite) Hilbert space $\mathcal{H}$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ through a mapping $\phi : \mathcal{X} \to \mathcal{H}$:

$$K(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{H}}, \quad \forall\, x, z \in \mathcal{X}.$$

We often write the inner product $\langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$ as $\phi(x)^T \phi(z)$, when the Hilbert space is clear from the context. This space is called the *feature space*, and the mapping is called the *feature mapping*. They depend on the kernel function K and will be denoted as ϕ_K and $\mathcal{H}_K$. The Gram matrix $G \in \mathbb{R}^{m \times m}$, defined in (1), will be denoted G_K when it is necessary to indicate the dependence on K.

2. Kernel Fisher Discriminant Analysis

2.1. Setup

Let $\{x_1, \ldots, x_{m_+}\} \subset \mathbb{R}^n$ denote training inputs from the positive class and $\{x_{m_+ + 1}, \ldots, x_m\} \subset \mathbb{R}^n$ denote those from the negative class. (The total number of negative inputs is $m_- = m - m_+$.) Let K be a kernel function. The two sets $\{\phi_K(x_i)\}_{i=1}^{m_+}$ and $\{\phi_K(x_i)\}_{i=m_+ + 1}^{m}$ represent the positive class and the negative class, respectively, in the feature space.

We learn a classifier $h : \mathcal{X} \to \{-1, +1\}$ from the training inputs whose decision boundary between the two classes is affine in the feature space $\mathcal{H}_K$:

$$h(x) = \operatorname{sgn}\left(w^T \phi_K(x) + b\right),$$

where $w \in \mathcal{H}_K$ is the vector of feature weights, $b \in \mathbb{R}$ is the intercept, and

$$\operatorname{sgn}(u) = \begin{cases} +1 & \text{if } u > 0 \\ -1 & \text{if } u < 0. \end{cases}$$

The data required to carry out KFDA are the means and covariances of the positive and negative classes in the feature space. In practice, we carry out KFDA with the sample means

$$\begin{aligned} \mu_K^+ &= \frac{1}{m_+} \sum_{i=1}^{m_+} \phi_K(x_i), \\ \mu_K^- &= \frac{1}{m_-} \sum_{i=m_+ + 1}^{m} \phi_K(x_i), \end{aligned}$$

and the sample covariances

$$\begin{aligned} \Sigma_K^+ &= \frac{1}{m_+} \sum_{i=1}^{m_+} (\phi_K(x_i) - \mu_K^+)(\phi_K(x_i) - \mu_K^+)^T, \\ \Sigma_K^- &= \frac{1}{m_-} \sum_{i=m_+ + 1}^{m} (\phi_K(x_i) - \mu_K^-)(\phi_K(x_i) - \mu_K^-)^T. \end{aligned}$$

2.2. Maximum Achievable FDR

The basic idea of KFDA is to find a direction in the feature space $\mathcal{H}_K$ onto which the projections of the two sets $\{\phi_K(x_i)\}_{i=1}^{m_+}$ and $\{\phi_K(x_i)\}_{i=m_+ + 1}^{m}$ are well separated. (Once the direction is fixed, the intercept can be chosen appropriately, taking into account misclassification costs.) Specifically, the separation between the two sets is measured by the ratio of the variance $(w^T \mu_K^+ - w^T \mu_K^-)^2$ between the classes to the variance $w^T(\Sigma_K^+ + \Sigma_K^-)w$ within the classes. Since the covariances may be singular, we add a (small) regularization term to the variance within the classes. Thus, KFDA maximizes the FDR

$$F_\lambda(w, K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)w}, \tag{2}$$

where λ is a positive regularization parameter and I is the identity operator in $\mathcal{H}_K$.

Using the Cauchy-Schwartz inequality, we can show that the weight vector

$$w^\star = (\Sigma_K^+ + \Sigma_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-) \tag{3}$$

maximizes the FDR. The maximum FDR achieved by $w^\star$ is given by

$$\begin{aligned} & F_\lambda^\star(K) \\ &= \max_{w \in \mathcal{H}_K \setminus \{0\}} F_\lambda(w, K) \\ &= (\mu_K^+ - \mu_K^-)^T (\Sigma_K^+ + \Sigma_K^- + \lambda I)^{-1} (\mu_K^+ - \mu_K^-). \end{aligned}$$

The maximum FDR depends on the kernel function K through the feature mapping ϕ_K. The square root of the maximum FDR is an empirical Mahalanobis distance between the positive and negative classes in the feature space. It measures the distance between the means μ_K^+ and μ_K^- of $\{\phi_K(x_i) \mid i = 1, \ldots, m_+\}$ and $\{\phi_K(x_i) \mid i = m_+ + 1, \ldots, m\}$, taking into account their distribution.

2.3. KFDA via Kernel Trick

An important result in KFDA (Mika et al., 2003) is that the optimal weight vector, given in (3), that maximizes the FDR is in the span of the image of the training inputs through the feature mapping. In other words, there is $\alpha^\star \in \mathbb{R}^m$ such that

$$w^\star = \sum_{i=1}^m \alpha_i^\star \phi_K(x_i) = U_K \alpha^\star, \tag{4}$$

where

$$U_K = [\phi_K(x_1) \ \cdots \ \phi_K(x_m)].$$

(This result can be viewed as an extension of the representer theorem (Shawe-Taylor & Cristianini, 2004) for SVMs.) Moreover, this weight vector can be found via solving a quadratic program in which the objective and constraints depend on the Gram matrix not on the kernel function (Mika et al., 2003).

In fact, we can find a closed-form expression for $\alpha^\star$ in (4):

$$\alpha^\star = \frac{1}{\lambda} \left[I - J(\lambda I + J G_K J)^{-1} J G_K \right] a, \tag{5}$$

where

$$\begin{aligned} a &= a_+ - a_-, \\ a_+ &= \begin{bmatrix} (1/m_+) \mathbf{1}_{m_+} \\ 0 \end{bmatrix}, \\ a_- &= \begin{bmatrix} 0 \\ (1/m_-) \mathbf{1}_{m_-} \end{bmatrix}, \\ J &= \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix}, \\ J_+ &= \frac{1}{\sqrt{m_+}} \left(I - \frac{1}{m_+} \mathbf{1}_{m_+} \mathbf{1}_{m_+}^T \right), \\ J_- &= \frac{1}{\sqrt{m_-}} \left(I - \frac{1}{m_-} \mathbf{1}_{m_-} \mathbf{1}_{m_-}^T \right). \end{aligned}$$

Here, $\mathbf{1}_n$ denotes the vector of all ones in $\mathbb{R}^n$. (When the dimension is obvious, we will drop the subscript.) The derivation of (5) is given in Appendix A.

We can represent the optimal decision boundary using the kernel function. Specifically, for a given point $x \in \mathcal{X}$, we can compute the inner product $\langle w^\star, \phi_K(x) \rangle_{\mathcal{H}_K}$ as

$$\begin{aligned} \langle w^\star, \phi_K(x) \rangle &= \sum_{i=1}^m \alpha_i^\star \phi_K^T(x_i) \phi_K(x) \\ &= \sum_{i=1}^m \alpha_i^\star K(x_i, x). \end{aligned}$$

To compute the inner product, we evaluate the kernel function at the pairs (x_i, x), $i = 1, \ldots, m$, not the feature mapping, which is known as the *kernel trick*.

3. Optimal Kernel Selection via Convex Optimization

3.1. Optimal Kernel Selection Problem

Let $\mathcal{K}$ be a convex set $\mathcal{K}$ of kernel functions, meaning that for any $K_1, K_2 \in \mathcal{K}$,

$$\theta K_1 + (1 - \theta) K_2 \in \mathcal{K}, \quad \forall \theta \in [0, 1].$$

The problem of finding the optimal kernel, over $\mathcal{K}$, in terms of maximum achievable FDR can now be written as

$$\begin{array}{ll} \text{maximize} & F_\lambda^\star(K) \\ \text{subject to} & K \in \mathcal{K}, \end{array} \tag{6}$$

where the variable is the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (and the problem data are the training examples).

3.2. General Convex Formulation

We show that the objective of the kernel selection problem (6) can be expressed as a function of the Gram matrix. Note from (3) and (4) that the objective can be written as

$$\begin{aligned} F_\lambda^\star(K) &= (\mu_K^+ - \mu_K^-)^T U_K \alpha^\star \\ &= a^T U_K^T U_K \alpha^\star. \end{aligned}$$

(Here, we use $\mu_K^+ - \mu_K^- = U_K a$.) Note that $U_K^T U_K$ is in fact the Gram matrix G_K:

$$\begin{aligned} & U_K^T U_K \\ = & \begin{bmatrix} \phi_K^T(x_1)\phi_K(x_1) & \cdots & \phi_K^T(x_1)\phi_K(x_m) \\ \vdots & \ddots & \vdots \\ \phi_K^T(x_m)\phi_K(x_1) & \cdots & \phi_K^T(x_m)\phi_K(x_m) \end{bmatrix} \\ = & \begin{bmatrix} K(x_1,x_1) & \cdots & K(x_1,x_m) \\ \vdots & \ddots & \vdots \\ K(x_m,x_1) & \cdots & K(x_m,x_m) \end{bmatrix} \\ = & \; G_K. \end{aligned}$$

We can now see from (5) that

$$F_\lambda^\star(K) = \frac{1}{\lambda}\left[a^T G_K a - a^T G_K J(\lambda I + JG_K J)^{-1} JG_K a\right].$$

The righ-hand side is a function of the Gram matrix.

Let $\mathcal{G}$ denote the set of Gram matrices consistent with the assumption made on the kernel function:

$$\mathcal{G} = \{G_K \mid K \in \mathcal{K}\}.$$

This set is a convex subset of $\mathbb{S}_+^m$. Here we use $\mathbb{S}_+^m$ ($\mathbb{S}_{++}^m$) to denote the set of all $m \times m$ symmetric positive semidefinite (definite) matrices. The convexity follows directly from convexity of $\mathcal{K}$. Moreover, any $G \in \mathcal{G}$ is positive semidefinite, since any $K \in \mathcal{K}$ satisfies the finitely positive semidefinite property (by the definition of kernel functions).

It is now clear that the optimal kernel selection problem (6) is equivalent to

$$\begin{array}{ll} \text{minimize} & f_\lambda^\star(G) \\ \text{subject to} & G \in \mathcal{G}, \end{array} \tag{7}$$

where the variable is $G = G^T \in \mathbb{R}^{m\times m}$ and

$$f_\lambda^\star(G) = \frac{1}{\lambda}\left[a^T GJ(\lambda I + JGJ)^{-1} JGa - a^T Ga\right].$$

(The two problems are equivalent in the sense that a solution of each problem is readily obtained from a solution of the other.) Note that the objective and constraints of this problem depend on a semidefinite matrix, and not a kernel function.

We establish the convexity of the objective function $f_\lambda^\star(G)$, and therefore also the problem (7). Note that $a^T Ga$ is linear in G. Thus, it suffices to show the convexity of $a^T GJ(\lambda I + JGJ)^{-1} JGa$. This function can be expressed as the composite function $f(h(G), s(G))$, where

$$\begin{aligned} f(x,X) &= x^T X^{-1} x, \\ h(G) &= JGa, \\ s(G) &= \lambda I + JGJ. \end{aligned}$$

The matrix fractional function $f(x,X)$ is convex on $\mathbb{R}^m \times \mathbb{S}_{++}^m$. Note that, for any $G \in \mathcal{G}$, $s(G)$ is positive definite ($\lambda > 0$). Since h and s are linear in G, the convexity of $f(h(G), s(G))$ now follows from a basic composition rule for convex functions: the composition of a convex function with an affine mapping is always convex.

3.3. Convex Combinations of Kernels

Here we focus on the special case in which $\mathcal{K}$ consists of convex combinations of given kernel functions $K_1, \ldots, K_p$:

$$\mathcal{K} = \left\{ K : \mathcal{X}\times\mathcal{X} \to \mathbb{R} \;\middle|\; K = \sum_{i=1}^p \theta_i K_i,\; \mathbf{1}^T\theta = 1,\; \theta \succeq 0 \right\},$$

where $\theta \succeq 0$ means that its elements θ_i are nonnegative. Here, we impose the normalization condition on the kernels K_i that they have the same trace. See Lanckriet et al. (2004b) for a discussion on the condition.

The set $\mathcal{G}$ of Gram matrices consistent with this set is given by

$$\mathcal{G} = \left\{ G \;\middle|\; G = \sum_{i=1}^p \theta_i G_i,\; \mathbf{1}^T\theta = 1,\; \theta \succeq 0 \right\},$$

where $G_i = G_i^T \in \mathbb{R}^{m\times m}$ is the Gram matrix computed with the kernel function K_i (and the training inputs). Any matrix in $\mathcal{G}$ is positive semidefinite, since it is a convex combination of the positive semidefinite matrices $G_1, \ldots, G_p$.

The convex problem (7) corresponding to the convex combinations above can be written as

$$\begin{array}{ll} \text{minimize} & f_\lambda^\star\left(\sum_{i=1}^p \theta_i G_i\right) \\ \text{subject to} & \theta \succeq 0, \\ & \mathbf{1}^T\theta = 1. \end{array} \tag{8}$$

This problem is simple: it involves minimizing a convex function over p nonnegative variables, with one equality constraint.

The cost of solving the problem is not significantly larger than that of SVMs applied to the same training inputs. The cost of forming and summing the Gram matrices G_i is $O(m^2 n + pm^2)$, and the additional cost of computing the gradient and Hessian of the objective (which requires the inversion of an $m \times m$ matrix) is $O(m^3)$. The Cholesky factorization of the $p\times p$ Hessian requires $O(p^3)$ flops. The total cost per Newton step of interior-point methods (Nesterov & Nemirovsky, 1994) applied to (8) is therefore $O(p^3 + m^2 n + pm^2 + m^3)$. In the case of $p, n \ll m$, the total cost grows like $O(m^3)$, which is the same as that of SVMs (but with a larger constant hidden in the $O(\cdot)$ notation).

We give another convex formulatino of (8). Using the Schur complement technique (Boyd & Vandenberghe, 2004), we can write the inequality

$$a^T GJ(\lambda I + JGJ)^{-1} JGa \leq t$$

equivalently as the linear matrix inequality

$$\begin{bmatrix} \lambda I + JGJ & JGa \\ a^T GJ & t \end{bmatrix} \succeq 0.$$

Here, for a symmetric matrix A, $A \succeq 0$ means that A is positive semidefinite. The convex problem (8) is now equivalent to

$$\begin{array}{ll} \text{minimize} & (1/\lambda)\left(t - \sum_{i=1}^{p} \theta_i a^T G_i a\right) \\ \text{subject to} & H(t,\theta) \succeq 0, \\ & \theta \succeq 0, \\ & \mathbf{1}^T \theta = 1, \end{array} \tag{9}$$

where the variables are $t \in \mathbb{R}$ and $\theta \in \mathbb{R}^m$, and $H(t,\theta) \in \mathbb{R}^{n+1 \times n+1}$ is defined as

$$H(t,\theta) = \begin{bmatrix} \lambda I + \sum_{i=1}^{p} \theta_i JG_iJ & \sum_{i=1}^{p} \theta_i JG_i a \\ \sum_{i=1}^{p} \theta_i a^T G_i J & t \end{bmatrix}.$$

This problem is a semidefinite program (SDP); see, *e.g.*, Vandenberghe and Boyd (1996).

This SDP can be solved by interior-point methods, with the same complexity as that of (8). One advantage of the SDP formulation (9) is that we can find the optimal kernel using standard SDP solvers such as SeDuMi (Sturm, 2001) or SDPT3 (Toh et al., 2002).

4. Numerical Results

We demonstrate the optimal kernel selection method described above with several machine learning benchmark examples from the UCI repository (Newman et al., 1998). The examples considered are shown in Table 1.

Each data set was randomly partitioned into a training set and a test set. We used 70% of the data points as the training set to perform KFDA and optimal kernel selection, and tested the generalization performance using the remaining data points. We generated 100 random partitions of the data (for each of the benchmark problems) and collected the results.

Our kernel is a convex combination of 10 Gaussian kernels:

$$K(x,z) = \sum_{i=1}^{10} \theta_i e^{-\|x-z\|^2/\sigma_i^2},$$

where θ_i are the weights of the kernels to be determined. The values of σ_i were chosen uniformly over the interval $[10^{-1}, 10^{2}]$ on the logarithmic scale. The regularization parameter in KFDA was fixed to 10^{-8}. The performance of KFDA for the benchmark examples does not appear to depend much on the regularization parameter, as long as it is neither too small nor too large.

Table 1. Classification results: KFDA with the optimal kernel $K^\star$ versus KFDA with K^{cv} found via cross-validation.

DATA SET (m,n)	MEAN TSA $(K^\star)$	MEAN TSA (K^{cv})
SONAR $(208,60)$	84.4%	83.3%
IONOSPHERE $(351,34)$	94.1%	92.6%
HEART $(297,13)$	81.7%	82.0%
PIMA $(768,8)$	74.9%	75.1%

For each of the benchmark problems, we computed the optimal weights $\theta_i^\star$ of the 10 kernels, using SeDuMi (Sturm, 2001). This solver is effective for the benchmark examples in Table 1, since their sizes are modest. For instance, we can solve the optimal kernel selection problem for the ionosphere data set in a few seconds.

For each of the benchmark examples, we compare the optimal kernel

$$K^\star(x,z) = \sum_{i=1}^{10} \theta_i^\star e^{-\|x-z\|^2/\sigma_i^2}$$

with the kernel K^{cv} found via cross-validation to tune the kernel parameter over σ_i given above. To do so, we computed the mean test set accuracy (TSA) (over the 100 instances of each problem). Table 1 summarizes the comparison results.

To better compare the generalization performances of the optimal kernel and the kernel found via cross-validation, we compare the receiver operating characteristic (ROC) curves (Pepe, 2000) of the classifiers combined with the kernels. The curves were found by carrying out ROC analysis over the 100 instances of each problem and then taking the average of the resulting ROC curves along the x-axis. As an illustrative example, we plot the ROC curve comparison results for the sonar data set in Figure 1. This figure shows that KFDA with the optimal kernel performs slightly better than KFDA with K^{cv}. While guaranteeing a false positive rate (the probability that a

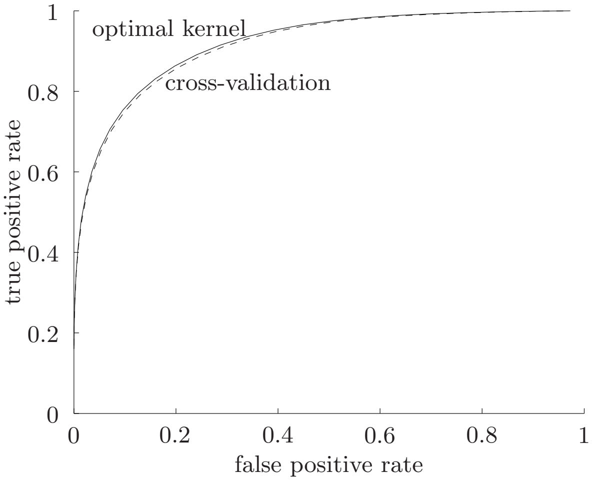

Figure 1. ROC curve comparison results for sonar data. Solid line: KFDA with the optimal kernel. Dashed line: KFDA with the kernel found via cross-validation.

negative example is misclassified) of 20%, KFDA with the optimal kernel achieves a slightly higher true positive rate (the probability that a positive example is classified correctly) than KFDA with K^{cv}. The areas under the two ROC curves are 0.91 (optimal kernel) and 0.90 (cross-validation). For the other benchmark examples considered, we see similar results.

We observe from the empirical results above that the optimal kernel selection method for KFDA has the potential of replacing cross-validation to tune kernel parameters, *i.e.*, we can carry out KFDA without cross-validation to tune kernel parameters. This observation is in line with the conclusions of Lanckriet et al. (2004b) and Fung et al. (2004). We expect that the improvement in mean TSA is more significant in heterogeneous data fusion in which we want to combine several kernels for learning heterogeneous data, as Lanckriet et al. (2004a) have demonstrated with other kernel-based methods.

5. Related Work

Many researchers have studied optimal kernel selection in kernel-based classification methods, which is called *kernel learning* (Bach et al., 2004; Bennett et al., 2002; Bi et al., 2004; Bousquet & Herrmann, 2003; Cristianini et al., 2001; Crammer et al., 2003; Fung et al., 2004; Lanckriet et al., 2004b; Lanckriet et al., 2004a; Ong et al., 2005; Xiong et al., 2005). The main emphasis is on formulating kernel learning as a tractable convex optimization problem.

A general result on the convexity of kernel learning has been established in Micchelli and Pontil (2005). The authors consider a general optimal kernel selection problem of the form

$$\begin{array}{ll} \text{minimize} & \displaystyle\inf_{w\in\mathcal{H}_K} \sum_{i=1}^{m} \psi\left(y_i, w^T\phi_K(x_i)\right) + \lambda\|w\|^2_{\mathcal{H}_K} \\ \text{subject to} & K \in \mathcal{K}, \end{array}$$

where the variable is the kernel function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$. Here, λ is a positive regularization parameter, $\{(x_i, y_i)\}_{i=1}^m$ is the set of given training examples, $\mathcal{K}$ is a set of of kernel functions, $\psi : \mathbb{R}^2 \to \mathbb{R}_+$ is a loss function (*e.g.*, the hinge loss or logistic loss). Kernel learning problems that arise in many kernel-based problems including 1-norm soft margin and 2-norm soft margin SVMs have this form. They show that if the loss function ψ is convex, then the problem above is in fact a convex optimization problem.

Evidently, the FDR in (2) does not satisfy the convexity condition. The convex formulation of kernel learning in KFDA given in §3 is therefore not a direct consequence of the general result above.

6. Conclusions

We have shown how to formulate the optimal kernel selection problem in KFDA as a tractable convex optimization. This convex formulation leads to a more efficient method than the iterative one proposed in Fung et al. (2004) that optimizes the weight vector and the Gram matrix alternatively. In fact, optimizing over the Gram matrix with a fixed weight vector has the same complexity as the convex formulation. Moreover, the convex formulation always finds the globally optimal kernel, while there is no such guarantee in the alternating method.

The general-purpose solvers for SDPs can solve problems of the form (9) up to a few thousand examples and a few hundred kernels in a reasonable amount of time on a PC. For larger problems, we need special-purpose solvers. We are currently developing a custom interior-point method for the original convex formulation (8). In doing so, we would like to exploit the fact that the regularization parameter appears only in the objective. This fact allows us to incorporate the so-called 'warm-start' strategy easily in the interior-point method, which can greatly facilitate tuning the regularization parameter.

Acknowledgments

The authors thank Kwangmoo Koh for helpful comments and suggestions. This material is supported in part by the National Science Foundation under grants

#0423905 and (through October 2005) #0140700, by the Air Force Office of Scientific Research under grant #F49620-01-1-0365, by MARCO Focus center for Circuit & System Solutions contract #2003-CT-888, and by MIT DARPA contract #N00014-05-1-0700.

References

Bach, F., Lanckriet, G., & Jordan, M. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proceedings of the* 21*th International Conference on Machine Learning (ICML)*. ACM.

Bennett, K., Momma, M., & Embrechts, J. (2002). MARK: A boosting algorithm for heterogeneous kernel models. *Proceedings of the Eighth SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Bi, J., Zhang, T., & Bennett, K. (2004). Column-generation boosting methods for mixture of kernels. *Proceeding of the Tenth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 521–526). ACM.

Bousquet, O., & Herrmann, D. (2003). On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems*, 15, MIT Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Crammer, K., Keshet, J., & Singer, Y. (2003). Kernel design using boosting. In *Advances in Neural Information Processing Systems*, 15, MIT Press.

Cristianini, N., Elisseeff, A., Shawe-Taylor, J., & Kandla, J. (2001). On kernel target alignment. In *Advances in Neural Information Processing Systems*, 13, pp. 367-373, MIT Press.

Fung, G., Dundar, M., Bi, J., & Rao, B. (2004). A fast iterative algorithm for Fisher discriminant using heterogeneous kernels. *Proceedings of the* 21*th International Conference on Machine Learning (ICML)*. ACM.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. Springer-Verlag.

Lanckriet, G., Bie, T. D., Cristianini, N., Jordan, M., & Noble, W. (2004a). A statistical framework for genomic data fusion. *Bioinformatics*, *20*, 2626–2635.

Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L., & Jordan, M. (2004b). Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Micchelli, C., & Pontil, M. (2005). Learning the kernel function via regularization. *Journal of Machine Learning Research*, *6*, 1099–1125.

Mika, S., Rätsch, G., & Müller, K. (2001). A mathematical programming approach to the kernel Fisher algorithm. In *Advances in Neural Information Processing Systems*, 13, pp. 591-597, MIT Press.

Mika, S., Rätsch, J., Weston, G., Schölkopf, B., Smola, A., & Müller, K. (2003). Constructing descriptive and discriminative non-linear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 623–628.

Nesterov, Y., & Nemirovsky, A. (1994). *Interior-point polynomial methods in convex programming*, vol. 13 of *Studies in Applied Mathematics*. Philadelphia, PA: SIAM.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. Available from `www.ics.uci.edu/~mlearn/MLRepository.html`.

Ong, C., Smola, A., & Williamson, R. (2005). Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, *6*, 1043–1071.

Pepe, M. (2000). Receiver operating characteristic methodology. *Journal of the American Statistical Association*, *95*, 308–311.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.

Sturm, J. (2001). *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*. Available from `sedumi.mcmaster.ca/`.

Toh, K., Tütüncü, R., & Todd, M. (2002). *SDPT3 version 3.02. a Matlab software for semidefinite-quadratic-linear programming*. Available from `www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM Review*, *38*, 49–95.

Xiong, H., Swamy, M., & Ahmad, M. (2005). Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, *16*, 460–474.

Yang, J., Frangi, A., Yang, J.-Y., Zhang, D., & Jin, Z. (1989). KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, *27*, 230–244.

A. Derivation of (5)

In what follows, we drop the subscript in $\mathcal{H}$, ϕ_K, U_K, μ_K^+, Σ_K^+, and so on.

Define

$$\begin{aligned} U_+ &= \begin{bmatrix} \phi(x_1) & \cdots & \phi(x_{m_+}) \end{bmatrix}, \\ U_- &= \begin{bmatrix} \phi(x_{m_++1}) & \cdots & \phi(x_m) \end{bmatrix}. \end{aligned}$$

We can write the samples means as

$$\mu^+ = U_+ a_+, \qquad \mu^- = U_- a_-.$$

Therefore,

$$\mu^+ - \mu^- = Ua.$$

Here a, a_+, and a_- are defined in §2.3.

The sample covariance Σ^+ can be written as

$$\begin{aligned} \Sigma^+ &= \frac{1}{m_+} \sum_{i=1}^{m_+} \phi(x_i)\phi(x_i)^T - \mu^+ {\mu^+}^T \\ &= \frac{1}{m_+} U_+ U_+^T - \frac{1}{m_+^2} U_+ \mathbf{1}_{m_+} \mathbf{1}_{m_+}^T U_+^T \\ &= U_+ J_+ J_+ U_+^T. \end{aligned}$$

Similarly, Σ^- can be written as

$$\Sigma^+ = U_- J_- J_- U_-^T.$$

The sum of Σ^+ and Σ^- is therefore given by

$$\Sigma^+ + \Sigma^- = UJJU^T.$$

We can now write the weight vector $w^\star$ given in (3) as

$$w^\star = \left(UJJU^T + \lambda I\right)^{-1} Ua.$$

Here, we can write the inverse of $UJJU^T + \lambda I$ as

$$\begin{aligned} &\left(UJJU^T + \lambda I\right)^{-1} \\ &\quad = \frac{1}{\lambda}\left[I - UJ\left(\lambda I + JU^T UJ\right)^{-1} JU^T\right], \end{aligned}$$

which is a straightforward extension of the matrix inversion formula to the Hilbert space $\mathcal{H}$. (Note that both the operator $UJJU^T + \lambda I$ from $\mathcal{H}$ into $\mathcal{H}$ and the matrix $\lambda I + JU^T UJ \in \mathbb{R}^{m \times m}$ are positive definite.)

Putting all pieces established above together and using $U^T U = G$, we can write $w^\star$ as

$$\begin{aligned} w^\star &= \frac{1}{\lambda}\left[I - UJ\left(\lambda I + JU^T UJ\right)^{-1} JU^T\right] Ua \\ &= \frac{1}{\lambda} U \left[I - J\left(\lambda I + JGJ\right)^{-1} JG\right] a. \end{aligned}$$