Discriminant Kernel and Regularization Parameter Learning via Semidefinite Programming

Jieping Ye JIEPING.YE@ASU.EDU
Jianhui Chen JIANHUI.CHEN@ASU.EDU
Shuiwang Ji SHUIWANG.JI@ASU.EDU
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287 USA

Abstract

Regularized Kernel Discriminant Analysis (RKDA) performs linear discriminant analysis in the feature space via the kernel trick. The performance of RKDA depends on the selection of kernels. In this paper, we consider the problem of learning an optimal kernel over a convex set of kernels. We show that the kernel learning problem can be formulated as a semidefinite program (SDP) in the binary-class case. We further extend the SDP formulation to the multi-class case. It is based on a key result established in this paper, that is, the multi-class kernel learning problem can be decomposed into a set of binary-class kernel learning problems. In addition, we propose an approximation scheme to reduce the computational complexity of the multi-class SDP formulation. The performance of RKDA also depends on the value of the regularization parameter. We show that this value can be learned automatically in the framework. Experimental results on benchmark data sets demonstrate the efficacy of the proposed SDP formulations.

1. Introduction

Regularized Kernel Discriminant Analysis (RKDA) works by embedding the data into a high-dimensional feature space through a nonlinear mapping, where a linear transformation is applied to achieve the maximum class discrimination (Baudat & Anouar, 2000; Mika et al., 2001; Mika et al., 2003). The nonlinear mapping is implicitly specified by a kernel function, which computes the inner products between the images of every possible data pair in the feature space. Thus, one of the key issues in RKDA is the selection (learning) of kernels.

The problem of kernel learning has been addressed by many researchers recently. Crammer et al. (2003) proposed to design kernels using boosting. Lanckriet et al. (2004b) pioneered the work of learning a linear combination of pre-specified kernels for Support Vector Machines (SVM) (Vapnik, 1998; Cristianini & Taylor, 2000) using convex programming and the work has been improved in (Bach et al., 2004) using the Sequential Minimal Optimization (SMO) algorithm (Platt, 1999). Recently, formulation of the kernel learning problem as semi-infinite linear program has been proposed along with extensions to regression and one-class classification (Sonnenburg et al., 2006). While most approaches produce stationary combinations, Lewis et al. (2006) proposed to use different combinations for different inputs. Argyriou et al. (2006) extended such method for combining potentially infinite number of kernels and the resulting problem is a difference of convex (DC) program. In general, approaches based on learning a convex combination of kernels offer the additional advantage of facilitating heterogeneous data integration from different sources. They have been applied for combining various biological data, *e.g.*, amino acid sequences, hydropathy profiles, and gene expression data for enhanced biological inference (Lanckriet et al., 2004a). Jebara (2004) considered the kernel selection problem in the context of multi-task learning. Other approaches for kernel learning includes those based on hyperkernels (Ong et al., 2005; Tsang & Kwok, 2006) and regularization (Micchelli & Pontil, 2005). The problem of kernel learning for discriminant analysis has been addressed originally in (Fung et al., 2004) and Kim et al. (2006) formulated this problem as a semidefinite program (SDP).

In this paper, we consider the problem of learning an

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

optimal kernel over a convex combination of kernels for RKDA. First, we show that the kernel learning problem can be formulated as a SDP in the binary-class case. Second, we extend the SDP formulation to the multi-class case based on a key result established in this paper, that is, the multi-class kernel learning problem can be decomposed into k binary-class kernel learning problems, which are constrained to share a common kernel, where k is the number of classes. The multi-class kernel learning problem and the corresponding decomposed formulation share the same optimal kernel, although they may have different optimal transformation matrices. In other words, the decomposed formulation is equivalent to the original one for the purpose of kernel learning. In addition, we show that the computational complexity of the multi-class SDP formulation can be reduced by applying an approximation scheme. In many cases, the performance of RKDA also depends on the value of the regularization parameter. We show that this value can be learned simultaneously and automatically in the framework. We have conducted experiments using benchmark data sets to show the effectiveness of the proposed SDP formulations.

2. Binary-class Kernel Learning

In this section, we first review the basics of kernel methods, as well as the SDP formulation for binary-class kernel learning in (Kim et al., 2006). Then we propose a simplified SDP formulation, which will be extended to the multi-class case in the next section.

We use $\mathcal{X}$ to denote the input or instance space, which is a subspace of $\mathbb{R}^d$, 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. Let $X = [x_1, \cdots, x_m]$ be the data matrix. A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel function (Schölkopf & Smola, 2002) if it satisfies the finitely positive semidefinite property: for any $x_1, \cdots, x_m \in \mathcal{X}$, the *Gram* matrix $G \in \mathbb{R}^{m \times m}$, defined by $G_{ij} = K(x_i, x_j)$, is positive semidefinite. Any kernel function K implicitly maps the input set $\mathcal{X}$ to a high-dimensional (possibly infinite) Hilbert space $\mathcal{H}_K$ equipped with the inner product $(\cdot, \cdot)_{\mathcal{H}_K}$ through a mapping ϕ_K from $\mathcal{X}$ to $\mathcal{H}_K$: $K(x, z) = (\phi_K(x), \phi_K(z))_{\mathcal{H}_K}$. In kernel-based classification, the algorithms learn a classifier $f : \mathcal{X} \to \{-1, +1\}$ whose decision boundary is affine in the feature space $\mathcal{H}_K$:

$$f(x) = sgn(w^T \phi_K(x) + b),$$

where $w \in \mathcal{H}_K$ is the vector of feature weights, $b \in \mathbb{R}$ is the intercept, and $sgn(u) = +1$, if $u > 0$, and -1 otherwise.

Let $\{x_1^+, \cdots, x_{m_+}^+\}$ and $\{x_1^-, \cdots, x_{m_-}^-\}$ denote the collections of data points from the positive and negative classes, respectively. The total number of data points in the training set is $m = m_+ + m_-$. For a given kernel function K, the basic idea of RKDA in the binary-class case 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 $\{\phi_K(x_i^-)\}_{i=1}^{m_-}$ are well separated. Define the centroids of the two classes as follows:

$$\mu_K^+ = \frac{1}{m_+} \sum_{i=1}^{m_+} \phi_K(x_i^+), \quad \mu_K^- = \frac{1}{m_-} \sum_{i=1}^{m_-} \phi_K(x_i^-),$$

and the two class covariance matrices as follows:

$$S_K^+ = \frac{1}{m_+} \sum_{i=1}^{m_+} (\phi_K(x_i^+) - \mu_K^+)(\phi_K(x_i^+) - \mu_K^+)^T, \quad (1)$$

$$S_K^- = \frac{1}{m_-} \sum_{i=1}^{m_-} (\phi_K(x_i^-) - \mu_K^-)(\phi_K(x_i^-) - \mu_K^-)^T. \quad (2)$$

Specifically, in RKDA the separation between the two classes is measured by the ratio of the variance $(w^T(\mu_K^+ - \mu_K^-))^2$ between the classes to the variance $w^T \left(m_+/m S_K^+ + m_-/m S_K^-\right) w$ within the classes. Thus, the following objective function is maximized:

$$F_1(w, K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T \left(\frac{m_+}{m} S_K^+ + \frac{m_-}{m} S_K^- + \lambda I\right) w}, \quad (3)$$

where $\lambda > 0$ is a regularization parameter and I is the identity matrix. The optimal weight vector that maximizes the objective function in Eq. (3) for fixed K and λ is given by

$$w^* = (m_+/m S_K^+ + m_-/m S_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-). \quad (4)$$

The maximum value of the objective function in Eq. (3) achieved by w^* is given by

$$F_1^*(K) = (\mu_K^+ - \mu_K^-)^T w^*. \quad (5)$$

It follows from the *Representer Theorem* (Schölkopf & Smola, 2002) that the optimal weight vector in RKDA is in the span of the images of the training points in the feature space. In other words, there exists a vector

$$\alpha^* = [\alpha_1^+, \cdots, \alpha_{m_+}^+, \alpha_1^-, \cdots, \alpha_{m_-}^-] \in \mathbb{R}^m, \quad (6)$$

such that $w^* = \phi_K(X)\alpha^*$, where $\phi_K(X)$ is the data matrix in the feature space. The optimal vector α^* is given by (Kim et al., 2006)

$$\alpha^* = \frac{1}{\lambda}(I - J(\lambda I + JGJ)^{-1} JG)a, \quad (7)$$

where a is an m-dimensional vector given by

$$a = [1/m_+, \cdots, 1/m_+, -1/m_-, \cdots, -1/m_-]^T, \quad (8)$$

and the matrix J is defined as:

$$J = \begin{pmatrix} \frac{1}{\sqrt{m_+}}(I - \frac{1}{m_+}e_{m_+}e_{m_+}^T) & 0 \\ 0 & \frac{1}{\sqrt{m_-}}(I - \frac{1}{m_-}e_{m_-}e_{m_-}^T) \end{pmatrix}.$$

where e_{m_+} and e_{m_-} are vectors of all ones of length m_+ and m_-, respectively. The optimal value $F_1^*(K)$ in Eq. (5) is thus given by

$$F_1^*(K) = \frac{1}{\lambda}a^T G(I - J(\lambda I + JGJ)^{-1}JG)a. \quad (9)$$

In (Kim et al., 2006), G is restricted to be a convex combination of p given kernel matrices $G_1, \cdots, G_p$ as

$$G \in \mathcal{G} = \left\{ G \middle| G = \sum_{i=1}^{p} \theta_i G_i, \sum_{i=1}^{p} \theta_i = 1, \theta_i \geq 0 \right\}. \quad (10)$$

It was shown in (Kim et al., 2006) that the optimal $G \in \mathcal{G}$ based on the kernel function K that maximizes $F_1^*(K)$ in Eq. (9) can be obtained by solving a semidefinite program (SDP) (Vandenberghe & Boyd, 1996; Boyd & Vandenberghe, 2004). General-purpose optimization packages such as SeDuMi (Sturm, 1999) can be used to solve the SDP problem.

2.1. Proposed SDP Formulation

We work on the centered version of the kernel matrices in the following discussion. This is equivalent to data centering as commonly used for data pre-processing. We learn an optimal kernel matrix $\tilde{G} \in \tilde{\mathcal{G}}$:

$$\tilde{\mathcal{G}} = \left\{ \tilde{G} \middle| \tilde{G} = \sum_{i=1}^{p} \theta_i \tilde{G}_i, \quad \sum_{i=1}^{p} \theta_i r_i = 1, \quad \theta_i \geq 0 \right\}, \quad (11)$$

where $\tilde{G}_i = PG_iP$, $r_i = \text{trace}(\tilde{G}_i)$, and $P \in \mathbb{R}^{m \times m}$ is the centering matrix defined as

$$P = I - \frac{1}{m}e_m e_m^T, \quad (12)$$

and e_m is the vector of all ones of size m. Consider the maximization of the following objective function:

$$F_2(w, K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K + \lambda I)w}, \quad (13)$$

where Σ_K, the so-called *total scatter matrix* in the feature space is defined as follows:

$$\Sigma_K = \frac{1}{m}\phi_K(X)\, P\, \phi_K(X)^T, \quad (14)$$

and μ_K is the global centroid of the data in the feature space given by

$$\mu_K = \frac{1}{m}\left(\sum_{i=1}^{m_+} \phi_K(x_i^+) + \sum_{i=1}^{m_-} \phi_K(x_i^-) \right). \quad (15)$$

It is easy to verify that Eqs. (3) and (13) are equivalent in terms of the computation of the optimal w for fixed K and λ. We show in the following theorem that optimizing $F_2(w, K)$ in Eq. (13) with respect to the kernel leads to a simplified SDP formulation:

Theorem 2.1. *Given a set of centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix $\tilde{G} \in \tilde{\mathcal{G}}$ that maximizes the objective function in Eq. (13) can be found by solving the following semidefinite programming problem:*

$$\begin{aligned} \min_{\theta, t} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i & a \\ a^T & t \end{pmatrix} \succeq 0, \\ & \theta \geq 0, \quad \theta^T r = 1, \end{aligned} \quad (16)$$

where a is defined in Eq. (8), $\theta = [\theta_1, \cdots, \theta_p]^T$, $r = \left[trace(\tilde{G}_1), \cdots, trace(\tilde{G}_p) \right]^T$, and $M \succeq 0$ implies that matrix M is positive semidefinite.

Proof. The optimal weight vector that maximizes $F_2(w, K)$ in Eq. (13) is given by

$$w^* = (\Sigma_K + \lambda I)^{-1}(\mu_K^+ - \mu_K^-), \quad (17)$$

and the maximum value of $F_2(w, k)$ is given by

$$F_2^*(K) = (\mu_K^+ - \mu_K^-)^T(\Sigma_K + \lambda I)^{-1}(\mu_K^+ - \mu_K^-). \quad (18)$$

Using the Sherman-Woodbury-Morrison formula (Golub & Van Loan, 1996), we have

$$w^* = \frac{1}{\lambda}\phi_K(X)\left(I - P\left(\lambda I + PGP\right)^{-1} PG \right) a, \quad (19)$$

and

$$\begin{aligned} F_2^*(K) &= (\mu_K^+ - \mu_K^-)^T w^* = a^T \phi_K(X)^T w^* \\ &= \frac{1}{\lambda}a^T \left(G - GP\left(\lambda I + PGP\right)^{-1} PG \right) a. \end{aligned}$$

Since the vector a defined in Eq. (8) is of zero mean, i.e., $Pa = a$, we have

$$\begin{aligned} F_2^*(K) &= \frac{1}{\lambda}a^T P \left(G - GP(\lambda I + PGP)^{-1}PG \right) Pa \\ &= \frac{1}{\lambda}a^T \left(\tilde{G} - \tilde{G}(\lambda I + \tilde{G})^{-1}\tilde{G} \right) a, \end{aligned} \quad (20)$$

where $\tilde{G} = PGP$. Since

$$\begin{aligned} \tilde{G} - \tilde{G}(\lambda I + \tilde{G})^{-1}\tilde{G} &= \lambda\tilde{G}(\lambda I + \tilde{G})^{-1} \\ &= \lambda I - \lambda^2(\lambda I + \tilde{G})^{-1}, \end{aligned}$$

the optimal value $F_2^*(K)$ in Eq. (20) is given by

$$F_2^*(K) = a^T a - \lambda a^T(\lambda I + \tilde{G})^{-1}a. \quad (21)$$

Thus, the maximization of $F_2^*(K)$ in Eq. (21) with a fixed λ, is equivalent to the minimization of

$$\lambda a^T(\lambda I + \tilde{G})^{-1}a = a^T\left(I + \frac{1}{\lambda}\tilde{G}\right)^{-1}a, \qquad (22)$$

subject to the constraint that $\tilde{G} \in \tilde{\mathcal{G}}$, which can be formulated as the following minimization problem:

$$\begin{array}{ll} \min\limits_{\theta} & a^T\left(I + \frac{1}{\lambda}\sum\limits_{i=1}^{p}\theta_i\tilde{G}_i\right)^{-1}a \\ \text{subject to} & \theta \geq 0, \quad \theta^T r = 1. \end{array} \qquad (23)$$

It follows from the Schur complement lemma (Golub & Van Loan, 1996; Lanckriet et al., 2004b) that the following inequality

$$a^T\left(I + \frac{1}{\lambda}\tilde{G}\right)^{-1}a \leq t$$

is equivalent to the Linear Matrix Inequality (LMI) (Boyd & Vandenberghe, 2004)

$$\begin{pmatrix} I + \frac{1}{\lambda}\tilde{G} & a \\ a^T & t \end{pmatrix} \succeq 0.$$

This completes the proof by adding a variable t. □

3. Multi-class Kernel Learning

In this section, we extend the SDP formulation to the multi-class case. We are given a data set that consists of m samples $\{(x_i, y_i)\}_{i=1}^m$, where $x_i \in \mathbb{R}^d$, and $y_i \in \{1, 2, \cdots, k\}$ denotes the class label of the i-th sample. The following objective function is maximized in multi-class RKDA:

$$F_3(W, K) = \text{trace}\left(\left(W^T(\Sigma_K + \lambda I)W\right)^{-1}W^T B_K W\right), \qquad (24)$$

where W is the transformation matrix, B_K is the so-called *between-class scatter matrix* defined as

$$B_K = \phi_K(X)HH^T\phi_K(X)^T, \qquad (25)$$

$H = [h_1, h_2, \cdots, h_k]$, where h_i is a vector whose j-th entry is given by $\sqrt{n/n_j} - \sqrt{n_j/n}$ if x_j belongs to the i-th class, and $-\sqrt{n_j/n}$ otherwise. Thus, the optimal W for RKDA can be obtained by computing the top eigenvectors of $(\Sigma_K + \lambda I)^{-1}B_K$.

Since the weight vectors are in the span of the images of the data points in the feature space, $W = \phi_K(X)A$ for some matrix $A = [\alpha_1, \cdots, \alpha_\ell] \in \mathbb{R}^{m\times\ell}$. Thus, $F_3(W, K)$ can be written as

$$\text{trace}\left(\left(A^T(GPG + \lambda G)A\right)^{-1}A^T GHH^T GA\right). \qquad (26)$$

Define two matrices S_t^K and S_b^K as follows:

$$S_t^K = GPG + \lambda G, \quad S_b^K = GHH^T G. \qquad (27)$$

Since the null space of S_t^K lies in the null space of S_b^K, there exists a nonsingular matrix Z such that

$$Z^T S_t^K Z = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, Z^T S_b^K Z = \begin{pmatrix} \Sigma_b & 0 \\ 0 & 0 \end{pmatrix}, \qquad (28)$$

where Σ_b is diagonal with the diagonal entries sorted in non-decreasing order. The optimal A^* consists of the first q columns of Z, that is, $A^* = [z_1, \cdots, z_q]$, and $q = \text{rank}(S_b^K)$. It follows that the optimal value of $F_3(W, K)$ achieved by A^* is given by

$$F_3^*(K) = \text{trace}(\Sigma_b) = \text{trace}\left(\left(S_t^K\right)^{-1}S_b^K\right). \qquad (29)$$

We can use the pseudo-inverse instead if S_t^K is singular. All the following arguments still follow.

The optimal kernel function K can be computed by maximizing $F_3^*(K)$ in Eq. (29), which is however highly nonlinear and difficult to solve. In the following, we present a formulation equivalent to the one in Eq. (29), which is more tractable computationally.

Consider the maximization of the following objective function:

$$F_4(W, K) = \sum_{i=1}^{k}\frac{(w_i^T\phi_K(X)h_i)^2}{w_i^T(\Sigma_K + \lambda I)w_i}, \qquad (30)$$

where $W = [w_1, \cdots, w_k]$, and h_i is the i-th column of H from Eq. (25). The following lemma shows that the optimal kernel matrix coincides for F_3 and F_4.

Lemma 3.1. *Let F_3 and F_4 be defined as in Eq. (24) and Eq. (30), respectively. Let W^* and K^* be the optimal solution to the following optimization problem:*

$$\max_K \max_W F_3(W, K), \qquad (31)$$

and let $\tilde{W}^*$ and $\tilde{K}^*$ be optimal solution to the following optimization problem:

$$\max_K \max_W F_4(W, K). \qquad (32)$$

Then $K^* = \tilde{K}^*$.

Proof. Since $W = \phi_K(X)A$, where $A = [\alpha_1, \cdots, \alpha_\ell]$, we have $w_i = \phi_K(X)\alpha_i$ and

$$F_4(W, K) = \sum_{i=1}^{k}\frac{(\alpha_i^T Gh_i)^2}{\alpha_i^T(GPG + \lambda G)\alpha_i} = \sum_{i=1}^{k}\frac{(\alpha_i^T Gh_i)^2}{\alpha_i^T S_t^K\alpha_i}.$$

The computation of α_i and α_j, for $i \neq j$, is independent of each other when the kernel function K

and λ are fixed. The optimal α_i^* is given by $\alpha_i^* = \left(S_t^K\right)^{-1} Gh_i$. It follows that the maximum value of $F_4(W,K)$ achieved by $A^* = [\alpha_1^*, \cdots, \alpha_k^*]$ is given by

$$F_4^*(K) = \sum_{i=1}^{k} (Gh_i)^T \left(S_t^K\right)^{-1} Gh_i. \tag{33}$$

Based on the properties of the trace (Golub & Van Loan, 1996), we have

$$\begin{aligned} F_4^*(K) &= \sum_{i=1}^{k} \text{trace}\left((Gh_i)^T \left(S_t^K\right)^{-1} Gh_i\right) \\ &= \text{trace}\left(\left(S_t^K\right)^{-1} \sum_{i=1}^{k} \left(Gh_i h_i^T G^T\right)\right) \\ &= \text{trace}\left(\left(S_t^K\right)^{-1} \left(GHH^T G^T\right)\right) \\ &= \text{trace}\left(\left(S_t^K\right)^{-1} S_b^K\right) = F_3^*(K). \end{aligned}$$

This completes the proof. □

Following the result for the bianry case, we can formulate the multi-class RKDA kernel learning problem as the following SDP problem:

$$\begin{aligned} \min_{t_1,\cdots,t_k,\theta} \quad & \sum_{j=1}^{k} t_j \\ \text{subject to} \quad & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i & h_j \\ h_j^T & t_j \end{pmatrix} \succeq 0, \text{for all } j \\ & \theta \geq 0, \quad \theta^T r = 1. \end{aligned} \tag{34}$$

Solving the SDP problem in Eq. (34) is computationally expensive for large k, as the complexity of the constraint depends on the number of classes. To alleviate this computational problem, we replace the k positive semidefinite (PSD) constraints in Eq. (34) with a more strict but relatively simple one. It is based on the result summarized in the following lemma:

Lemma 3.2. *Let $M \in I\!R^{m\times m}$ be any positive definite matrix, $a_1, \cdots, a_k \in I\!R^m$, and $t_1, \cdots, t_k \in I\!R$. Then*

$$\begin{pmatrix} M & a_1 & a_2 & \cdots & a_k \\ a_1^T & t_1 & 0 & \cdots & 0 \\ a_2^T & 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_k^T & 0 & 0 & \cdots & t_k \end{pmatrix} \succeq 0 \tag{35}$$

implies

$$\begin{pmatrix} M & a_j \\ a_j^T & t_j \end{pmatrix} \succeq 0, \text{ for all } j. \tag{36}$$

Proof. For a symmetric and positive semidefinite matrix, it is known that all of its principal submatrices are also symmetric and positive semidefinite. Matrices in Eq. (36) are all principal submatrices of the matrix in Eq. (35). This can be seen by removing 2 to j and $j+2$ to $k+1$ rows and columns of the block matrix in Eq. (35). This completes the proof of the lemma. □

Thus, we obtain the following approximate SDP formulation for multi-class kernel learning problem:

$$\begin{aligned} \min_{t_1,\cdots,t_k,\theta} \quad & \sum_{j=1}^{k} t_j \\ \text{s.t.} \quad & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i & h_1 & h_2 & \cdots & h_k \\ h_1^T & t_1 & 0 & \cdots & 0 \\ h_2^T & 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_k^T & 0 & 0 & \cdots & t_k \end{pmatrix} \succeq 0, \\ & \theta \geq 0, \quad \theta^T r = 1. \end{aligned} \tag{37}$$

Note that the optimal solution to the formulation in Eq. (37) satisfies the constraints in Eq. (34).

4. Joint Kernel and Regularization Parameter Learning

The SDP formulations discussed in the last two sections focus on the kernel learning only, while the regularization parameter λ is pre-specified. In some cases, the performance of RKDA depends on the value of λ. In this section, we show that the proposed SDP formulations can be reformulated for the automatic estimation of λ. This is motivated from the work in (Lanckriet et al., 2004b).

Recall from Eq. (22) that the following objective function is minimized in binary-class RKDA:

$$a^T \left(I + \frac{1}{\lambda}\tilde{G}\right)^{-1} a.$$

We aim to estimate λ in a joint framework. However, a very small value of λ, or a large trace value of $I + \frac{1}{\lambda}\tilde{G}$ will make the objective function small. To deal with this problem, we propose the following formulation for joint kernel and regularization parameter learning:

$$\begin{aligned} \min_{\theta,\lambda,\tau} \quad & a^T \left(I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i\right)^{-1} a \cdot \tau \\ \text{subject to} \quad & \theta \geq 0, \lambda > 0, \\ & \text{trace}\left(I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i\right) = \tau, \end{aligned} \tag{38}$$

Table 1. Comparison of eight algorithms using four binary-class data sets. The mean classification accuracy of 30 different partitions is reported. For SDP_θ and SDP_{Kim}, λ is set to 10^{-8} as used in (Kim et al., 2006). For SM1 and SM2, C is set to 1 as used in (Lanckriet et al., 2004b). The accuracy of SDP_{Kim} on the Cancer data set is not shown, as we observed several cases of numerical problem from the SDP solver when running SDP_{Kim} on Cancer. See the text for a detailed description of various algorithms.

Data set	SDP_θ	$\text{SDP}_{\theta,\lambda}$	SDP_{Kim}	SM1	SM2	SM2_C	RKDA_{CV}	SVM_{CV}
Sonar	85.60	90.16	88.46	89.75	89.59	89.84	88.86	89.27
Heart	76.85	81.54	81.98	82.59	82.71	82.53	76.54	82.22
Cancer	96.05	96.00	—	97.08	97.15	97.01	95.30	96.62
Ionosphere	89.90	95.10	89.14	95.28	94.81	95.19	93.62	93.52

which aims to minimize the objective function and control the trace value simultaneously. It is equivalent to the following formulation:

$$\begin{aligned} \min_{\theta,\lambda,\tau} \quad & a^T \left(\frac{1}{\tau} I + \frac{1}{\tau\lambda} \sum_{i=1}^{p} \theta_i \tilde{G}_i \right)^{-1} a \\ \text{subject to} \quad & \theta \geq 0, \lambda > 0, \\ & \text{trace}\left(\frac{1}{\tau} I + \frac{1}{\tau\lambda} \sum_{i=1}^{p} \theta_i \tilde{G}_i \right) = 1. \end{aligned} \tag{39}$$

We set $\tilde{G}_0 = I$ and treat $1/\tau$ as its coefficient. The binary-class SDP formulation for joint kernel and regularization parameter learning can be derived as

$$\begin{aligned} \min_{t,\tilde{\theta}} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} \sum_{i=0}^{p} \tilde{\theta}_i \tilde{G}_i & a \\ a^T & t \end{pmatrix} \succeq 0, \\ & \tilde{\theta} \geq 0, \quad \sum_{i=0}^{p} \tilde{\theta}_i \text{trace}(\tilde{G}_i) = 1, \end{aligned} \tag{40}$$

where $\tilde{\theta} = [\tilde{\theta}_0, \tilde{\theta}_1, \cdots, \tilde{\theta}_p]^T$, $\tilde{\theta}_0 = 1/\tau$, $\tilde{\theta}_i = \theta_i/(\tau\lambda)$, for $i = 1, \cdots, p$, and $\tilde{G}_0 = I$. With the computed $\tilde{\theta}$, we can obtain λ and θ_i for all i up to a scaling factor.

Similarly, we can derive the following SDP formulation for multi-class problems:

$$\begin{aligned} \min_{t_1,\cdots,t_k,\tilde{\theta}} \quad & \sum_{j=1}^{k} t_j \\ \text{subject to} \quad & \begin{pmatrix} \sum_{i=0}^{p} \tilde{\theta}_i \tilde{G}_i & h_j \\ h_j^T & t_j \end{pmatrix} \succeq 0, \text{for all } j \\ & \tilde{\theta} \geq 0, \quad \sum_{i=0}^{p} \tilde{\theta}_i \text{trace}(\tilde{G}_i) = 1, \end{aligned} \tag{41}$$

where $\tilde{\theta} = [\tilde{\theta}_0, \tilde{\theta}_1, \cdots, \tilde{\theta}_p]^T$, $\tilde{\theta}_0 = 1/\tau$, $\tilde{\theta}_i = \theta_i/(\tau\lambda)$, for $i = 1, \cdots, p$, and $\tilde{G}_0 = I$. This can be adapted for the approximate SDP formulation in Eq. (37).

5. Experimental Study

In this section, we evaluate the SDP formulations proposed in this paper using a collection of benchmark data sets. The reported experimental results are averaged over 30 random partitions of the data into a training and a test set of ratio 4:1 (binary-class problems) and 3:2 (multi-class problems). Following (Kim et al., 2006), we focus on learning a convex combination of ten Gaussian kernels: $K(x,z) = \sum_{i=1}^{10} \theta_i \, e^{-||x-z||^2/\sigma_i^2}$. The values of σ_i are chosen uniformly over the interval $[10^{-1}, 10^2]$ on the logarithmic scale, as in (Kim et al., 2006). We use the standard SDP solver SeDuMi (Sturm, 1999) for computing the optimal kernel.

5.1. Experiments on Binary-class Problems

For the binary-class case, we evaluate two formulations: SDP with regularization parameter λ fixed (SDP_θ) and SDP with λ learned automatically ($\text{SDP}_{\theta,\lambda}$). We compare them with relevant algorithms: 1-norm soft margin SVM (SM1), 2-norm soft margin SVM with or without the regularization parameter C optimized automatically (SM2_C and SM2) as proposed in (Lanckriet et al., 2004b), and SDP formulation (SDP_{Kim}) proposed in (Kim et al., 2006), as well as RKDA and SVM with the optimal kernel and regularization parameter selected via double cross-validation (RKDA_{CV} and SVM_{CV}). We conduct the experiments using four binary-class data sets as used in (Lanckriet et al., 2004b). Sonar, Ionosphere, and Breast Cancer are from the UCI Machine Learning Repository (Newman et al., 1998). Heart is from the STATLOG project[1].

Several observations can be made from the results presented in Table 1. First, $\text{SDP}_{\theta,\lambda}$ outperforms SDP_θ and SDP_{Kim} on Sonar, Heart, and Ionosphere data sets, and is comparable to SDP_θ on Cancer data set. This result shows the effectiveness of the automatic

[1] http://www.is.umk.pl/projects/datasets-stat.html#Heart

learning of λ in $\text{SDP}_{\theta,\lambda}$. Second, $\text{SDP}_{\theta,\lambda}$ outperforms RKDA_{CV} and is comparable to SVM_{CV} on all test data sets. Note that double cross-validation selects the single best kernel. The favorite performance of $\text{SDP}_{\theta,\lambda}$ over RKDA_{CV} may be due to the existence of complementary information in different kernels. Finally, $\text{SDP}_{\theta,\lambda}$, SM1, SM2, SM2_C, and SVM_{CV} are comparable. However, the first four methods determine the kernel automatically and avoid the cross-validation.

Table 2. Comparison of four algorithms using five multi-class data sets. The approximate formulations for multi-class SDP_θ and $\text{SDP}_{\theta,\lambda}$ have been used for the comparison. In SDP_θ, λ is set to 10^{-8}.

Data Set	SDP_θ	$\text{SDP}_{\theta,\lambda}$	RKDA_{CV}	SVM_{CV}
Wine (3)	96.97	98.66	97.69	98.30
USPS (3)	94.41	99.41	99.13	99.33
USPS (6)	87.14	97.93	98.08	97.62
USPS (8)	78.35	96.93	96.03	95.96
Waveform (3)	81.95	83.08	83.05	83.41

5.2. Experiments on Multi-class Problems

For the multi-class case, we compare SDP_θ and $\text{SDP}_{\theta,\lambda}$ with RKDA_{CV} and SVM_{CV}. We also compare the exact and approximate formulations in terms of classification accuracy and computational cost.

Experimental results on five data sets are summarized in Table 2, where we have used the approximate formulations for SDP_θ and $\text{SDP}_{\theta,\lambda}$. The USPS data set is described in (Hull, 1994), and the Wine and Waveform data sets are from UCI Machine Learning Repository. We select the first 3, 6, and 8 classes in USPS, while all 3 classes in Wine and Waveform are used. We thus obtain five distinct data sets: USPS ($k = 3$), USPS ($k = 6$), USPS ($k = 8$), Wine ($k = 3$), and Waveform ($k = 3$). For each class, we randomly choose 100 samples from the original data sets.

We can observe from Table 2 that $\text{SDP}_{\theta,\lambda}$ outperforms SDP_θ in all cases, while it is comparable to RKDA_{CV} and SVM_{CV}. This result shows the effectiveness of the proposed multi-class SDP formulation for joint kernel and regularization parameter learning. It is expected to be more effective for heterogeneous data integration.

Figure 1 shows the comparison between the exact and approximate multi-class SDP formulations using three data sets: Waveform ($k = 3$), Wine ($k = 3$), and USPS ($k = 3$). The comparison is based on classification accuracy and computation time (in seconds). The main observation from Figure 1 is that the SDP formulations under the approximate constraint ($\text{SDP}_\theta^{\text{approx}}$ and $\text{SDP}_{\theta,\lambda}^{\text{approx}}$) are comparable to the exact SDP formulations ($\text{SDP}_\theta^{\text{exact}}$ and $\text{SDP}_{\theta,\lambda}^{\text{exact}}$) in terms of classification accuracy, while they are much more efficient. Thus, the use of the approximate constraints doesn't degrade the classification performance. Interestingly, $\text{SDP}_\theta^{\text{approx}}$ and $\text{SDP}_{\theta,\lambda}^{\text{approx}}$ achieve similar performance in classification, however the latter has a much less computational cost. This implies that with the automatic learning of λ, $\text{SDP}_{\theta,\lambda}^{\text{approx}}$ requires a smaller number of iterations before convergence than $\text{SDP}_\theta^{\text{approx}}$.

6. Conclusion and Future Work

We propose a simplified SDP formulation for the binary-class kernel learning problem in RKDA, which can be extended naturally to the multi-class case. We also show that the regularization parameter in RKDA can be learned automatically in the framework.

Our experimental results have shown that the proposed approximate SDP formulations work well in most cases, especially when λ is tuned automatically, while they have a much less computational cost in comparison with the exact formulations. We plan to carry out theoretical analysis on the approximate scheme. The semidefinite program is expensive to solve even for problems of moderate size. We are currently investigating efficient formulations for kernel and regularization parameter learning based on Quadratically Constrained Quadratic Program (QCQP) (Boyd & Vandenberghe, 2004). The proposed formulations can be applied for heterogeneous data integration. We plan to apply these approaches for the analysis of biological images (Ye et al., 2006), where various types of feature sources will be integrated.

Acknowledgments

This research is sponsored by the Center for Evolutionary Functional Genomics of the Biodesign Institute at Arizona State University and by the National Science Foundation Grant IIS-0612069.

References

Argyriou, A., Hauser, R., Micchelli, C., & Pontil, M. (2006). A DC-programming algorithm for kernel selection. *ICML* (pp. 41–48).

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *ICML*.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Compututation*, *12*, 2385–2404.

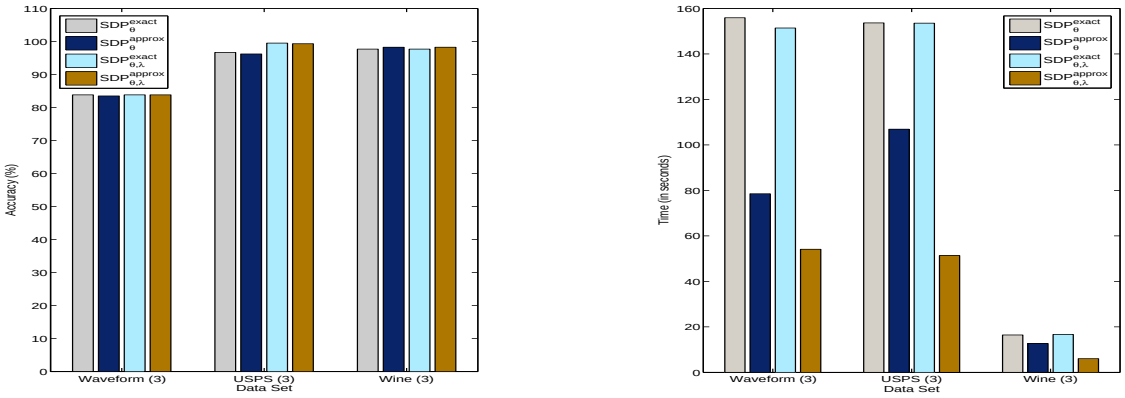

Figure 1. Comparison of the exact and approximate multi-class SDP formulations using three data sets in terms of classification accuracy (left) and computational time in seconds (right).

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

Crammer, K., Keshet, J., & Singer, Y. (2003). Kernel design using boosting. *NIPS* (pp. 537–544).

Cristianini, N., & Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Fung, G., Dundar, M., Bi, J., & Rao, B. (2004). A fast iterative algorithm for Fisher discriminant using heterogeneous kernels. *ICML*.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations*. The Johns Hopkins University Press. Third edition.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Trans. Pattern Analysis Machine Intelligence*, *16*, 550–554.

Jebara, T. (2004). Multi-task feature and kernel selection for SVMs. *ICML*.

Kim, S.-J., Magnani, A., & Boyd, S. (2006). Optimal kernel selection in kernel Fisher discriminant analysis. *ICML* (pp. 465–472).

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., Ghaoui, L. E., & Jordan, M. I. (2004b). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Lewis, D., Jebara, T., & Noble, W. S. (2006). Nonstationary kernel combination. *ICML* (pp. 553–560).

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

Mika, S., Rätsch, G., & Müller, K.-R. (2001). A mathematical programming approach to the kernel fisher algorithm. *NIPS* (pp. 591–597). MIT Press.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., & Müller, K. (2003). Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Trans. Pattern Analysis Machine Intelligence*, *25*, 623–633.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

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

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, 185–208. Cambridge, MA, USA: MIT Press.

Schölkopf, S., & Smola, A. (2002). *Learning with kernels: Support vector machines,regularization, optimization and beyond*. MIT Press.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11-12*, 625–653.

Tsang, I. W., & Kwok, J. T. (2006). Efficient hyperkernel learning using second-order cone programming. *IEEE Trans. on Neural Networks*, *17*, 48–58.

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

Vapnik, V. (1998). *Statistical learning theory*. Wiley, New York.

Ye, J., Chen, J., Li, Q., & Kumar, S. (2006). Classification of drosophila embryonic developmental stage range based on gene expression pattern images. *Computational Systems Bioinformatics Conference* (pp. 293–298).