Journal of Machine Learning Research 9 (2008) 719-758 Submitted 1/07; Revised 1/08; Published 4/08

Multi-class Discriminant Kernel Learning via Convex Programming

Jieping Ye JIEPING.YE@ASU.EDU
Shuiwang Ji SHUIWANG.JI@ASU.EDU
Jianhui Chen JIANHUI.CHEN@ASU.EDU
Department of Computer Science and Engineering
Center for Evolutionary Functional Genomics
The Biodesign Institute
Arizona State University
Tempe, AZ 85287, USA

Editor: Isabelle Guyon and Amir Saffari

Abstract

Regularized kernel discriminant analysis (RKDA) performs linear discriminant analysis in the feature space via the kernel trick. Its performance depends on the selection of kernels. In this paper, we consider the problem of multiple kernel learning (MKL) for RKDA, in which the optimal kernel matrix is obtained as a linear combination of pre-specified kernel matrices. We show that the kernel learning problem in RKDA can be formulated as convex programs. First, we show that this problem can be formulated as a semidefinite program (SDP). Based on the equivalence relationship between RKDA and least square problems in the binary-class case, we propose a convex quadratically constrained quadratic programming (QCQP) formulation for kernel learning in RKDA. A semi-infinite linear programming (SILP) formulation is derived to further improve the efficiency. We extend these formulations to the multi-class case based on a key result established in this paper. That is, the multi-class RKDA kernel learning problem can be decomposed into a set of binary-class kernel learning problems which are constrained to share a common kernel. Based on this decomposition property, SDP formulations are proposed for the multi-class case. Furthermore, it leads naturally to QCQP and SILP formulations. As the performance of RKDA depends on the regularization parameter, we show that this parameter can also be optimized in a joint framework with the kernel. Extensive experiments have been conducted and analyzed, and connections to other algorithms are discussed.

Keywords: model selection, kernel discriminant analysis, semidefinite programming, quadratically constrained quadratic programming, semi-infinite linear programming

1. Introduction

Formulation of machine learning problems as convex programs has been one of the recent trends in machine learning research. Such formulations offer global solutions and avoid some difficulties encountered by traditional learning algorithms (Lanckriet et al., 2003, 2004b; d'Aspremont et al., 2007). Kernel methods (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) work by embedding the input data into some high-dimensional feature space, and they are generally formulated as convex optimization problems. The key fact underlying the success of kernel methods is that the embedding into feature space can be determined uniquely by specifying a kernel function that computes the dot product between data points in the feature space. In other words, the kernel

function implicitly defines the nonlinear mapping to the feature space, and expensive computations in the high-dimensional feature space can be avoided by evaluating the kernel function. Thus, one of the central issues in kernel methods is the selection of kernels.

To automate kernel-based learning algorithms, it is desirable to integrate the tuning of kernels into the learning process. This problem has been addressed from different perspectives recently. Lanckriet et al. (2004b) pioneered the work of multiple kernel learning (MKL) in which the optimal kernel matrix is obtained as a linear combination of pre-specified kernel matrices. It was shown (Lanckriet et al., 2004b) that the coefficients in MKL can be determined by solving convex programs in the case of support vector machines (SVM) (Vapnik, 1998; Cristianini and Taylor, 2000). This MKL problem was formulated as support kernel machines (SKM) in Bach et al. (2004), and the sequential minimal optimization (SMO) algorithm (Platt, 1999) was proposed to solve it. Recently, this SKM was reformulated as semi-infinite linear program (SILP) which was shown to be scalable to large data sets and a large number of kernels (Sonnenburg et al., 2006; Rakotomamonjy et al., 2007). Micchelli and Pontil (2005, 2007) studied the problem of finding an optimal kernel from a prescribed convex set of kernels by regularization. To deal with problems with structured output, MKL for joint feature map was proposed in Zien and Ong (2007). While most existing work focuses on learning kernels for SVM, Fung et al. (2004) proposed to learn kernels for discriminant analysis. Based on ideas from MKL, this problem was reformulated as SDP in Kim et al. (2006). In general, approaches based on learning linear combination of kernel matrices offer the additional advantage of facilitating heterogeneous data integration from multiple sources. Such formulations have been applied to combine various biological data, for example, amino acid sequences, hydropathy profiles, and gene expression data, for enhanced biological inference (Lanckriet et al., 2004a).

Ong et al. (2005) showed that the learning of kernels can be accomplished by defining a reproducing kernel Hilbert space on the space of kernels itself, and the resulting optimization problem is an SDP. This formulation was recast into second order cone program (SOCP) (Lobo et al., 1998) in Tsang and Kwok (2006). Hoi et al. (2007) showed that the kernel matrix can be learned in a nonparametric manner by solving SDP. The kernel learning problem in the context of multiple tasks was considered in Jebara (2004). Some recent extensions of kernel learning produced nonstationary combinations (Lewis et al., 2006) and potentially infinite number of kernels (Argyriou et al., 2006).

This paper addresses the issue of kernel learning for regularized kernel discriminant analysis (RKDA) (Mika et al., 1999; Baudat and Anouar, 2000; Mika et al., 2001, 2003). Our proposed methods belong to the MKL framework, and they can thus be used for heterogeneous data integration. We systematically extend the kernel learning problem for RKDA in several directions. First, we extend the formulation in Kim et al. (2006) by proposing a simplified SDP formulation. Based on this simplified form and the equivalence relationship between KRDA and least square problems in the binary-class case, we propose a convex quadratically constrained quadratic programming (QCQP) formulation for this problem. To improve the efficiency of our formulations, we further develop a semi-infinite linear programming (SILP) formulation. While most existing work on kernel learning only deals with binary-class problems, we show that all of our formulations can be extended naturally to the multi-class setting. In particular, we show that the kernel learning problem for multi-class RKDA can be decomposed into a set of binary-class kernel learning problems that are constrained to share a common kernel. It is worth noting that the optimal kernel is the same for the original and the decomposed formulations, though the optimal transformation matrices may not coincide. In other words, the decomposed form is equivalent to the original one for the purpose of kernel learning. We further develop an approximate scheme to reduce the computational cost of

multi-class SDP formulation. Finally, we propose to optimize the regularization parameter along with the kernels in a joint framework. This joint optimization framework is derived from and similar to the work in De Bie et al. (2003); Lanckriet et al. (2004b).

The key contributions of this paper can be highlighted as follows:

- We propose a simplified SDP formulation for the RKDA kernel learning problem in the binary-class case. Based on this simplified form and the equivalence relationship between RKDA and least square problems in the binary-class case, we derive QCQP and SILP formulations for this problem.

- We show that the multi-class RKDA kernel learning problem can be decomposed into k binary-class kernel learning problems where k is the number of classes. This leads to two (exact and approximate) SDP formulations in the multi-class case. Based on this decomposition property, we show that the QCQP and SILP formulations for binary-class problems can be extended naturally to the multi-class case.

- We show that all the proposed formulations can be recast to optimize the regularization parameter simultaneously. This joint learning framework further automates the learning algorithms.

- We conduct extensive experiments using a collection of benchmark data sets to compare several relevant algorithms under a unified experimental setup. To demonstrate the effectiveness of the proposed formulations for heterogeneous data integration, we apply these formulations to combine multiple data sources derived from gene expression pattern images (Tomancak et al., 2002).

The rest of this paper is organized as follows: We derive the SDP, QCQP, and SILP formulations for the binary-class case in Section 2. Section 3 extends these formulations to the multi-class case. The joint optimization of regularization parameter is presented in Section 4. Section 5 presents the experimental evaluation, and this paper concludes with discussion and conclusion in Section 6.

Notation

$x \in \mathbb{R}^n$ denotes an n-dimensional vector. Similarly, $A \in \mathbb{R}^{m \times n}$ denotes a matrix with m rows and n columns. I is used to denote the identity matrix of an appropriate dimension and e_m denotes the vector of all ones of length m. For a square symmetric matrix S, $S \succeq 0$ means it is positive semidefinite. We also use the short-hand $x \geq 0$ to denote that each component of the vector x is non-negative.

2. Convex Formulations for Binary-class Problems

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. 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, underlying probability distribution over $\mathcal{X} \times \mathcal{Y}$.

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to R$ is called a kernel function (Schölkopf and Smola, 2002) if it satisfies the finitely positive semidefinite property. That is, 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 between the two classes 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:

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

and the two sample class covariance matrices as follows:

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

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 $w^T\left(m_+/mS_K^+ + m_-/mS_K^-\right)w$ within the classes. Thus, RKDA maximizes the following objective function:

$$F_1(w,K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T\left(m_+/mS_K^+ + m_-/mS_K^- + \lambda I\right)w}, \tag{1}$$

where $\lambda > 0$ is the regularization parameter. The optimal weight vector

$$w^* \equiv \underset{w}{\operatorname{argmax}}\{F_1(w,K)\}$$

that maximizes the objective function in Equation (1) for a fixed kernel function K and a fixed regularization parameter λ is given by

$$w^* = (m_+/mS_K^+ + m_-/mS_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-).$$

The maximum value

$$F_1^*(K) \equiv \max_w \{F_1(w,K)\}$$

of the objective function in Equation (1) achieved by the optimal weight vector w^* is given by

$$F_1^*(K) = (\mu_K^+ - \mu_K^-)^T \left(m_+/mS_K^+ + m_-/mS_K^- + \lambda I\right)^{-1} (\mu_K^+ - \mu_K^-). \tag{2}$$

It follows from the Representer Theorem (Schölkopf and 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

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

such that

$$w^* = \sum_{i=1}^{m_+} \alpha_i^+ \phi_K(x_i^+) + \sum_{i=1}^{m_-} \alpha_i^- \phi_K(x_i^-) = \phi_K(X)\alpha^*,$$

where $\phi_K(X)$ is the data matrix in the feature space given by

$$\phi_K(X) = \left[\phi_K(x_1^+), \cdots, \phi_K(x_{m_+}^+), \phi_K(x_1^-), \cdots, \phi_K(x_{m_-}^-)\right].$$

The optimal vector α^* is given by Kim et al. (2006) as

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

where I is the identity matrix, a is an m-dimensional vector given by

$$a = [1/m_+, \cdots, 1/m_+, -1/m_-, \cdots, -1/m_-]^T \in \mathbb{R}^m, \tag{3}$$

the matrix J is defined as:

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

G is restricted to be a linear combination of the p given kernel matrices $G_1, \cdots, G_p$ as

$$G \in \mathcal{G} = \left\{ G = \sum_{i=1}^p \theta_i G_i \;\middle|\; \sum_{i=1}^p \theta_i = 1\,, \quad \theta_i \geq 0 \quad \forall i \right\},$$

and e_{m_+} and e_{m_-} are vectors of all ones of length m^+ and m^-, respectively.

The optimal value $F_1^*(K)$ in Equation (2) is thus given by

$$\begin{aligned} F_1^*(K) &= (\mu_K^+ - \mu_K^-)^T \left(m_+/mS_K^+ + m_-/mS_K^- + \lambda I\right)^{-1} (\mu_K^+ - \mu_K^-) \\ &= (\mu_K^+ - \mu_K^-)^T w^* = (\mu_K^+ - \mu_K^-)^T \phi_K(X)\alpha^* = a^T \phi_K(X)^T \phi_K(X)\alpha^* \\ &= \frac{1}{\lambda} a^T G(I - J(\lambda I + JGJ)^{-1}JG)a. \end{aligned} \tag{4}$$

It was shown in Kim et al. (2006) that the optimal Gram matrix G based on the kernel function K that maximizes $F_1^*(K)$ given in Equation (4) can be obtained by solving a semidefinite program (SDP)

(Vandenberghe and Boyd, 1996; Boyd and Vandenberghe, 2004). General-purpose optimization packages such as SeDuMi (Sturm, 1999) use the interior-point methods (Nesterov and Nemirovskii, 1994) to solve SDP. However, for problems of moderate size in machine learning, this overhead of optimal kernel learning is large, and its computation time can easily exceed that of the learning algorithm itself.

We propose a new SDP formulation for this problem in the next subsection. The proposed SDP formulation has a simplified form. Experimental results presented in Section 5 show that the proposed formulation is comparable to the one in Kim et al. (2006). More importantly, this simplified formulation lays the foundation for the extensions to multi-class problems in Section 3 and the joint optimization of regularization parameter in Section 4.

2.1 Simplified SDP Formulation

In the rest of this paper, we work on the centered version of kernel matrices. This is equivalent to centering the data as preprocessed in linear discriminant analysis (LDA) and principal component analysis (PCA). More precisely, given a set of p kernel matrices $G_1, \cdots, G_p$, the proposed algorithms learn an optimal kernel matrix $\tilde{G} \in \tilde{\mathcal{G}}$, where

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

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

$$P = I - \frac{1}{m} e_m e_m^T, \tag{5}$$

and e_m is the vector of all ones of size m.

Consider the maximization of the following objective function:

$$F_2(w,K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K + \lambda I)w}, \tag{6}$$

where Σ_K is defined as follows:

$$\begin{aligned} \Sigma_K &= m_+ S_K^+ + m_- S_K^- + \frac{m_+ m_-}{m} (\mu_K^+ - \mu_K^-)(\mu_K^+ - \mu_K^-)^T \\ &= \sum_{i=1}^{m_+} (\phi_K(x_i^+) - \mu_K)(\phi_K(x_i^+) - \mu_K)^T + \sum_{i=1}^{m_-} (\phi_K(x_i^-) - \mu_K)(\phi_K(x_i^-) - \mu_K)^T \\ &= \phi_K(X) P \phi_K(X)^T, \end{aligned} \tag{7}$$

P is defined in Equation (5), and

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

is the global centroid of the data in the feature space. Note that the scaling factor $1/m$ has been omitted in the definition of Σ_K in Equation (7). It turns out that for fixed K and λ, Equations (1) and (6) are equivalent in terms of the computation of the optimal weight vector w. We show in the following theorem that optimizing $F_2(w,K)$ in Equation (6) with respect to the kernel function leads to a simplified SDP formulation.

Theorem 2.1 *Given a set of p 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 Equation (6) can be found by solving the following semidefinite programming problem:*

$$\begin{array}{lll} \min\limits_{\theta,t} & t & \\ \textit{subject to} & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i & a \\ a^T & t \end{pmatrix} \succeq 0, & (8) \\ & \theta \geq 0, & \\ & \theta^T r = 1, & \end{array}$$

where a is defined in Equation (3), $\theta = [\theta_1, \cdots, \theta_p]^T$, *and* $r = \left[trace(\tilde{G}_1), \cdots, trace(\tilde{G}_p)\right]^T$.

Proof The optimal weight vector

$$w^* \equiv \underset{w}{\text{argmax}} \{F_2(w,K)\}$$

is given by

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

The maximum value of the objective function in Equation (6) achieved by w^* is given by

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

It follows from Appendix A that

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

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 Equation (3) is of zero mean, that is, $Pa = a$, we have

$$\begin{aligned} F_2^*(K) &= \frac{1}{\lambda} a^T P \left(G - GP(\lambda I + PGP)^{-1} PG\right) Pa \\ &= \frac{1}{\lambda} a^T \left(\tilde{G} - \tilde{G}(\lambda I + \tilde{G})^{-1}\tilde{G}\right) a, \end{aligned} \tag{9}$$

where $\tilde{G}$ is derived from G with both rows and columns centered as

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

Since

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

the optimal value $F_2^*(K)$ in Equation (9) can be simplified as

$$F_2^*(K) = a^T a - \lambda a^T (\lambda I + \tilde{G})^{-1} a. \tag{10}$$

It follows that the optimal kernel learning problem in RKDA, which maximizes $F_2^*(K)$ in Equation (10) for a fixed regularization parameter λ, is equivalent to minimizing

$$\lambda a^T (\lambda I + \tilde{G})^{-1} a = a^T \left(I + \frac{1}{\lambda} \tilde{G} \right)^{-1} a, \tag{11}$$

subject to the constraint that $\tilde{G} \in \tilde{\mathcal{G}}$.

Mathematically, the optimal kernel learning problem can be formulated as follows:

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

We can write the inequality

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

equivalently as the linear matrix inequality (LMI) (Boyd and Vandenberghe, 2004)

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

via the Schur complement lemma (Golub and Van Loan, 1996; Lanckriet et al., 2004b). We complete the proof by a simple change of variable. ■

2.2 QCQP Formulation

The optimization problem proposed by Kim et al. (2006) and the one in Theorem 2.1 are both SDP problems, which are computationally very expensive to solve, even with the recent advances in interior point methods. In this subsection, we show that this kernel learning problem can be reformulated equivalently as a quadratically constrained quadratic program (QCQP) (Boyd and Vandenberghe, 2004), which can then be solved more efficiently than SDP.

It is known that discriminant analysis and least square problems are equivalent in the binary-class case (Mika, 2002). Consider the regularized least squares problem, which minimizes the following objective function:

$$F_3(w,K) = ||(\phi_K(X)P)^T w - a||^2 + \lambda ||w||^2. \tag{12}$$

The following lemma relates this problem to the problem of optimal kernel learning.

Lemma 2.1 *The optimal kernel function K solving the optimization problem in Equation (11) is also the minimizer of the objective function in Equation (12).*

Proof The optimal w^* that minimizes the objective function in Equation (12) for a fixed K and λ is given by

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

The optimal value of the objective function in Equation (12) is therefore given by

$$F_3^*(K) = a^T \left(I + \frac{1}{\lambda}\tilde{G}\right)^{-1} a,$$

where $\tilde{G} = PGP$. This completes the proof of this lemma. ■

Based on this equivalence result, we can formulate the kernel learning problem as a QCQP problem, as summarized in the following theorem.

Theorem 2.2 *Given a set of p centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix, in the form of a convex linear combination of the given p kernel matrices, that minimizes the objective function in Equation (12) can be found by solving the following convex QCQP problem:*

$$\begin{aligned} \max_{\beta, t} \quad & -\frac{1}{4}\beta^T\beta + \beta^T a - \frac{1}{4\lambda}t \\ \textit{subject to} \quad & t \geq \frac{1}{r_i}\beta^T \tilde{G}_i \beta, \ \textit{for } i = 1, \cdots, p, \end{aligned} \tag{13}$$

where $r_i = trace(\tilde{G}_i)$.

Proof We consider the dual formulation of the minimization of $F_3(w,K)$ in terms of w. Denote

$$\eta = (\phi_K(X)P)^T w - a.$$

It follows that

$$F_3(w,K) = ||\eta||^2 + \lambda||w||^2.$$

Define the Lagrangian function of the following optimization problem:

$$\begin{aligned} \min_{w,\eta} \quad & F_3(w,K) = ||\eta||^2 + \lambda||w||^2 \\ \text{subject to} \quad & \eta = (\phi_K(X)P)^T w - a \end{aligned}$$

as follows:

$$L(\eta, w, \beta) = ||\eta||^2 + \lambda||w||^2 - \beta^T((\phi_K(X)P)^T w - a - \eta),$$

where β is the vector of Lagrangian dual variables. Taking the derivatives of $L(\eta, w, \beta)$ with respect to η and w and setting them equal to zero, we get

$$\begin{aligned} \frac{\partial L(\eta, w, \beta)}{\partial \eta} &= 2\eta + \beta = 0, \\ \frac{\partial L(\eta, w, \beta)}{\partial w} &= 2\lambda w - \phi_K(X)P\beta = 0. \end{aligned}$$

It follows that

$$\eta = -\frac{\beta}{2}, \text{ and } w = \frac{\phi_K(X)P\beta}{2\lambda}.$$

Thus, we obtain the following Lagrangian dual function:

$$g(\beta) = \min_{w,\eta} L(\eta, w, \beta) = -\frac{1}{4}\beta^T \left(I + \frac{1}{\lambda} PGP \right) \beta + \beta^T a.$$

The optimal β^* is computed by maximizing $g(\beta)$ as

$$\beta^* = \operatorname*{argmax}_{\beta} g(\beta) = \operatorname*{argmax}_{\beta} \left\{ -\frac{1}{4}\beta^T \left(I + \frac{1}{\lambda} PGP \right) \beta + \beta^T a \right\}.$$

Since strong duality holds, the optimal kernel is given by solving the following optimization problem:

$$\min_{\tilde{G} \in \tilde{\mathcal{G}}} \max_{\beta} \left\{ -\frac{1}{4}\beta^T \left(I + \frac{1}{\lambda} \tilde{G} \right) \beta + \beta^T a \right\}.$$

We can rewrite the above optimization problem as

$$\begin{aligned}
& \min_{\theta:\theta\geq 0, \theta^T r = 1} \max_{\beta} \left\{ -\frac{1}{4}\beta^T \left(I + \frac{1}{\lambda} \sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta + \beta^T a \right\} \qquad (14) \\
= & \max_{\beta} \min_{\theta:\theta\geq 0, \theta^T r = 1} \left\{ -\frac{1}{4}\beta^T \left(I + \frac{1}{\lambda} \sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta + \beta^T a \right\} \\
= & \max_{\beta} \min_{\theta:\theta\geq 0, \theta^T r = 1} \left\{ -\frac{1}{4\lambda} \sum_{i=1}^{p} \theta_i \beta^T \tilde{G}_i \beta - \frac{1}{4}\beta^T \beta + \beta^T a \right\} \\
= & \max_{\beta} \left\{ -\frac{1}{4}\beta^T \beta + \beta^T a - \frac{1}{4\lambda} \max_{\theta:\theta\geq 0, \theta^T r = 1} \left(\sum_{i=1}^{p} \theta_i \beta^T \tilde{G}_i \beta \right) \right\} \\
= & \max_{\beta} \left\{ -\frac{1}{4}\beta^T \beta + \beta^T a - \frac{1}{4\lambda} \max_{i} \left(\frac{1}{r_i} \beta^T \tilde{G}_i \beta \right) \right\}. \qquad (15)
\end{aligned}$$

The exchange of minimization and maximization in deriving the second equation from the first holds since the objective function is convex in θ and concave in β, the minimization problem is strictly feasible in θ and the maximization problem is strictly feasible in β. Therefore, Slater's condition (Boyd and Vandenberghe, 2004) follows and strong duality holds (Lanckriet et al., 2004b; Boyd and Vandenberghe, 2004). By simply changing the last term in Equation (15) to t and moving it to the constraint, we prove this theorem. ■

Note that general-purpose optimization software packages like SeDuMi (Sturm, 1999) and MOSEK (Andersen and Andersen, 2000) employ the interior point methods, and they solve the primal and dual problems simultaneously. Thus, the coefficients, $\theta_1, \cdots, \theta_p$, can be obtained directly from the dual variables.

The formulation in Equation (13) is a quadratically constrained quadratic program (QCQP), which is a special form of second order cone program (SOCP) (Lobo et al., 1998; Alizadeh and

Goldfarb, 2003) and SDP. Theoretical results on interior point method (Nesterov and Nemirovskii, 1994) show that QCQP can be solved more efficiently than SDP, and it is therefore more scalable to large-scale problems. Similar ideas have been used in Lanckriet et al. (2004b) to learn a non-negative linear combination of kernel matrices.

2.3 SILP Formulation

Semi-infinite programming (SIP) (Hettich and Kortanek, 1993) refers to optimization problems that seek the maximum of the function $F(z)$ subject to a system of constraints on z, expressed as $g(z,t) \leq 0$, for all t in some set B. When both the objective and constraints are linear (and hence convex), it is known as semi-infinite linear programming (SILP). We show in this section that the kernel learning problem for RKDA can be formulated as an SILP problem, as summarized in the following theorem.

Theorem 2.3 *Given a set of p centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix, in the form of a convex linear combination of the given p kernel matrices, that maximizes the objective function in Equation (12) can be found by solving the following SILP problem:*

$$\begin{aligned} \max_{\theta,\gamma} \quad & \gamma && (16) \\ \textit{subject to} \quad & \theta \geq 0, \\ & \theta^T r = 1, \\ & \sum_{i=1}^{p} \theta_i S_i(\beta) \geq \gamma, \quad \textit{for all } \beta, && (17) \end{aligned}$$

where $S_i(\beta)$ is defined as

$$S_i(\beta) = \frac{r_i}{4}\beta^T\beta + \frac{1}{4\lambda}\beta^T\tilde{G}_i\beta - r_i\beta^T a, \quad \textit{for } i = 1, \cdots, p, \tag{18}$$

$r = (r_1, \cdots, r_p)^T$, and $r_i = trace(\tilde{G}_i)$.

Proof It follows from the definition of $S_i(\beta)$ in Equation (18) that the optimization problem in Equation (14) can be expressed equivalently as

$$\begin{aligned} \max_{\theta} \min_{\beta} \quad & \sum_{i=1}^{p} \theta_i S_i(\beta) && (19) \\ \text{subject to} \quad & \theta \geq 0, \\ & \theta^T r = 1. \end{aligned}$$

Assume β^* is the optimal solution to the problem in Equation (19) and define $\gamma^* = \sum_{i=1}^{p} \theta_i S_i(\beta^*)$ as the minimum objective value achieved by β^*. We have

$$\sum_{i=1}^{p} \theta_i S_i(\beta) \geq \gamma^*, \quad \text{for all } \beta.$$

By defining

$$\gamma = \min_{\beta} \sum_{i=1}^{p} \theta_i S_i(\beta)$$

and substituting γ into the objective, we prove this theorem. ■

Note that the optimization problem in Equation (16) is an SILP since both θ and γ are linearly constrained, and there are an infinite number of constraints, one for each possible value of β. As in Sonnenburg et al. (2006), we propose to use the *column generation* technique to solve this SILP problem. In this technique, the optimal θ and γ are computed for a restricted subset of constraints in Equation (17) and this problem is called the *restricted master problem*. Constraints that are not satisfied by current θ and γ are added successively to the restricted master problem until all constraints are satisfied. For fast convergence of the algorithm, it is desirable to add constraint that maximizes the violation for current θ and γ. That is, the β value that solves

$$\beta_\theta = \operatorname*{argmin}_\beta \sum_{i=1}^{p} \theta_i S_i(\beta), \tag{20}$$

is desired. If $\sum_{i=1}^{p} \theta_i S_i(\beta^\theta) \geq \gamma$, then all the constraints are satisfied, and θ and γ reach their optimal values. Otherwise, this constraint is added to the restricted master problem and the iteration continues.

It follows from the definition of $S_i(\beta)$ in Equation (18) that the problem in Equation (20) can be written as

$$\min_\beta \left\{ \frac{1}{4}\beta^T\beta + \frac{1}{4\lambda}\beta^T \left(\sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta - \beta^T a \right\}. \tag{21}$$

For a fixed θ, the problem in Equation (21) is an unconstrained convex quadratic program whose solution can be obtained analytically. To avoid computing matrix inverse, we obtain β by solving the following system of linear equations:

$$\left(\frac{1}{2} I + \frac{1}{2\lambda} \sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta = a.$$

After β is computed, the corresponding constraint is added to the restricted master problem to obtain the intermediate θ and γ. Note that the restricted master problem is a linear program. Thus, the proposed algorithm for solving the SILP problem proposed in Theorem 2.3 alternates between solving a linear system and a linear program. In contrast, the SILP formulation proposed in Sonnenburg et al. (2006) for SVM kernel learning involves solving a constrained quadratic program (QP) and a linear program. They shown that the constrained QP coincides with a single kernel SVM formulation, and thus existing software for solving SVM can be used directly.

The alternating algorithm for solving the proposed SILP problem belongs to a family of algorithms for solving general SIP problems called the *exchange methods*, in which the constraints are exchanged at each iteration. It follows from Theorem 7.2 in Hettich and Kortanek (1993) that these methods are guaranteed to converge. Similar to the convergence criterion used in Sonnenburg et al. (2006), the algorithm returns when

$$\left| 1 - \frac{\sum_{i=1}^{p} \theta_i^{(t-1)} S_i(\beta^{(t)})}{\gamma^{(t-1)}} \right| \leq \varepsilon, \tag{22}$$

where $\theta_i^{(t-1)}$, for $i = 1, \cdots, p$, and $\gamma^{(t-1)}$ are the optimal solutions to the restricted master problem at the $(t-1)$-th iteration, $\beta^{(t)}$ is the β value that maximizes the constraint violation at the t-th iteration, and ε is a user-specified tolerance parameter. We set $\varepsilon = 5 \times 10^{-4}$ in our experiments.

2.4 Time Complexity Analysis

We analyze the time complexity of the proposed formulations for the binary-class case. It follows from the analysis in Lanckriet et al. (2004b) that the proposed SDP and QCQP formulations have the worst-case time complexity of $O\left((p+n)^2 n^{2.5}\right)$ and $O\left(pn^2+n^3\right)$, respectively, where p is the number of candidate kernels and n is the number of training samples. The algorithm to solve the proposed SILP formulation alternates between solving a linear program (LP) and a linear system of equations. The LP formulation involved has a simple structure and its computation time is small, especially when p is much smaller than n. Note that the number of constraints in the LP depends on the number of iterations. Our experiments show that the algorithm converges within a small number of iterations. Thus, the time complexity of the SILP formulation is dominated by the time in solving the linear system which has a complexity of $O\left(n^3\right)$. Overall, the SILP formulation has a worst-case time complexity of $O\left(n^3 Ite\right)$ where Ite is the number of iterations.

All formulations discussed in Lanckriet et al. (2004b), Kim et al. (2006) and Sonnenburg et al. (2006) are constrained to binary-class problems. We show in the next section that our formulations in this section can be extended naturally to the multi-class case.

3. Convex Formulations for Multi-class Problems

In 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, and $k>2$. Similar to the binary-class case, let $X=[x_1,\cdots,x_m]$ be the data matrix.

In the multi-class RKDA formulation, the maximization of the following objective function is commonly used (Ye, 2005):

$$F_4(W,K) = \text{trace}\left(\left(W^T\left(\Sigma_K+\lambda I\right)W\right)^{-1} W^T B_K W\right), \tag{23}$$

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

$$B_K = \phi_K(X) H H^T \phi_K(X)^T,$$

$H=[h_1,h_2,\cdots,h_k]$, and h_i is a vector whose j-th entry is given by

$$h_i(j) = \begin{cases} \sqrt{\frac{n}{n_j}} - \sqrt{\frac{n_j}{n}} & \text{if the } j\text{-th data point belongs to the } i\text{-th class} \\ -\sqrt{\frac{n_j}{n}} & \text{otherwise.} \end{cases} \tag{24}$$

The optimal W is given by computing the eigenvectors of the following matrix:

$$\left(\Sigma_K+\lambda I\right)^{-1} B_K.$$

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

$$F_4(W,K) = \text{trace}\left(\left(A^T\left(GPG+\lambda G\right)A\right)^{-1} A^T GHH^T GA\right).$$

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

$$\begin{aligned} S_t^K &= GPG+\lambda G, \\ S_b^K &= GHH^T G. \end{aligned}$$

Since the null space of S_t^K lies in the null space of S_b^K (Ye and Xiong, 2006), there exists a nonsingular matrix Z such that

$$\begin{aligned} Z^T S_t^K Z &= \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, \\ Z^T S_b^K Z &= \begin{pmatrix} \Sigma_b & 0 \\ 0 & 0 \end{pmatrix}, \end{aligned}$$

where Σ_b is diagonal with the diagonal entries sorted in non-decreasing order. The optimal A^* is given by

$$A^* = Z_q = [z_1, \cdots, z_q],$$

where Z_q consists of the first q columns of Z, and $q = \text{rank}(S_b^K)$. It follows that the optimal value of $F_4(W,K)$ achieved by the optimal A^* is given by

$$F_4^*(K) = \text{trace}(\Sigma_b) = \text{trace}\left(\left(S_t^K\right)^{-1} S_b^K\right). \tag{25}$$

Here we have assumed that $S_t^K = GPG + \lambda G$ is nonsingular. We could use the pseudo-inverse to deal with the singular case, and all the following arguments still follow.

Thus, in the multi-class case, the optimal kernel function K can be computed by maximizing $F_4^*(K)$ in Equation (25), which is however highly nonlinear and difficult to solve. In the following, we present an equivalent formulation as the one in Equation (25), which is more tractable computationally.

3.1 SDP Formulation

Consider the maximization of the following objective function:

$$F_5(W,K) = \sum_{i=1}^{k} \frac{(w_i^T \phi_K(X) h_i)^2}{w_i^T (\Sigma_K + \lambda I) w_i}, \tag{26}$$

where

$$W = [w_1, \cdots, w_k]$$

is the transformation matrix, and h_i is defined in Equation (24). The following lemma shows that the optimal kernel function K coincides for F_4 and F_5.

Lemma 3.1 *Let F_4 and F_5 be defined as in Equation (23) and Equation (26), respectively. Let W^* and K^* be the optimal solution to the following optimization problem:*

$$\max_K \max_W F_4(W,K), \tag{27}$$

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

$$\max_K \max_W F_5(W,K). \tag{28}$$

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

Proof Since $W = \phi_K(X)A$, we have $w_i = \phi_K(X)\alpha_i$ and

$$F_5(W,K) = \sum_{i=1}^{k} \frac{(\alpha_i^T G h_i)^2}{\alpha_i^T (GPG + \lambda G)\alpha_i} = \sum_{i=1}^{k} \frac{(\alpha_i^T G h_i)^2}{\alpha_i^T S_t^K \alpha_i}.$$

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

$$\alpha_i^* = \left(S_t^K\right)^{-1} G h_i.$$

It follows that the maximum value of $F_5(W,K)$ achieved by the optimal $A^* = [\alpha_1^*, \cdots, \alpha_k^*]$ is given by

$$F_5^*(K) = \sum_{i=1}^{k} (G h_i)^T \left(S_t^K\right)^{-1} G h_i.$$

Based on the properties of matrix trace, we have

$$\begin{aligned}
F_5^*(K) &= \sum_{i=1}^{k} (G h_i)^T \left(S_t^K\right)^{-1} G h_i \\
&= \sum_{i=1}^{k} \text{trace}\left((G h_i)^T \left(S_t^K\right)^{-1} G h_i\right) \\
&= \sum_{i=1}^{k} \text{trace}\left(\left(S_t^K\right)^{-1} G h_i (G h_i)^T\right) \\
&= \text{trace}\left(\left(S_t^K\right)^{-1} \sum_{i=1}^{k} \left(G h_i h_i^T G^T\right)\right) \\
&= \text{trace}\left(\left(S_t^K\right)^{-1} \left(GHH^T G^T\right)\right) \\
&= \text{trace}\left(\left(S_t^K\right)^{-1} S_b^K\right) \\
&= F_4^*(K).
\end{aligned}$$

This completes the proof. ■

It is interesting to note that, in general, the optimal W^* and $\tilde{W}^*$ for the optimization problems in Equations (27) and (28) are different. However, it has been shown recently that, when the value of the regularization parameter is approaching zero, multi-class regularized least squares is equivalent to multi-class discriminant analysis under a mild condition (Ye, 2007). Empirical evidences show that when the value of the regularization parameter is small, which is usually the case in practice, their performance is similar.

The objective function in Equation (26) is closely related to its binary counterpart in Equation (6). Note that a variant of the Fisher discriminant ratio (FDR) (Kim et al., 2006) can be written as:

$$F_2(w,K) = \frac{(w^T \phi_K(X)\, a)^2}{w^T (\Sigma_K + \lambda I) w}.$$

Thus, $F_5(W,K)$ in Equation (26) can be interpreted as the weighted summation of the FDRs between the samples in the i-th class and the rest where $i = 1,\cdots,k$. The weights can be computed from the definition of H in Equation (24) as follows:

$$h_i = \begin{bmatrix} \vdots \\ \sqrt{\frac{n}{n_i}} - \sqrt{\frac{n_i}{n}} \\ \vdots \\ -\sqrt{\frac{n_i}{n}} \\ \vdots \end{bmatrix} = (n-n_i)\sqrt{\frac{n_i}{n}} \begin{bmatrix} \vdots \\ \frac{1}{n_i} \\ \vdots \\ -\frac{1}{n-n_i} \\ \vdots \end{bmatrix} = (n-n_i)\sqrt{\frac{n_i}{n}}\, a^{(i)},$$

where $a^{(i)}$ is obtained from Equation (3) by taking the samples from the i-th class as positive and the rest as negative. It follows that the weight for the i-th binary classification problem is: $(n-n_i)^2 n_i/n$.

Following the results from the last section for the binary-class case, the optimal kernel learning problem for multi-class RKDA can be formulated as follows:

$$\begin{aligned} \min_{t_1,\cdots,t_k,\theta} \quad & \sum_{j=1}^{k} t_j \\ \text{subject to} \quad & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i & h_j \\ h_j^T & t_j \end{pmatrix} \succeq 0, \text{ for } j = 1,\cdots,k, \\ & \theta \geq 0, \\ & \theta^T r = 1. \end{aligned} \tag{29}$$

Unfortunately, the SDP problem given in Equation (29) is computationally prohibitive due to the presence of positive semidefinite constraints. To alleviate this computational problem, we put all the constraints in a single larger constraint. This imposes stronger constraints than those on the original problem, but the computational cost can be reduced dramatically. It is based on the following lemma.

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

$$\begin{pmatrix} M & a_1 & a_2 & \cdots & a_k \\ a_1^T & t_1 & 0 & \cdots & 0 \\ a_2^T & 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_k^T & 0 & 0 & \cdots & t_k \end{pmatrix} \succeq 0 \tag{30}$$

implies

$$\begin{pmatrix} M & a_j \\ a_j^T & t_j \end{pmatrix} \succeq 0, \text{ for all } j. \tag{31}$$

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 Equation (31) are all principal submatrices of the matrix in Equation (30). This can be seen by removing 2 to j and $j+2$ to $k+1$ rows and columns of the block matrix in Equation (30). This completes the proof of the lemma. ■

We summarize the main result of this section in the following theorem:

Theorem 3.1 *Given a set of p centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix, in the form of linear combination of the given p kernel matrices, that maximizes the objective function in Equation (26) can be found by solving the SDP problem in Equation (29). This problem can be approximated by the following more restricted formulation:*

$$\begin{aligned} \min_{t_1,\cdots,t_k,\theta} \quad & \sum_{j=1}^{k} t_j \\ \textit{subject to} \quad & \begin{pmatrix} I + \frac{1}{\lambda}\sum_{i=1}^{p}\theta_i\tilde{G}_i & h_1 & h_2 & \cdots & h_k \\ h_1^T & t_1 & 0 & \cdots & 0 \\ h_2^T & 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_k^T & 0 & 0 & \cdots & t_k \end{pmatrix} \succeq 0, \\ & \theta \geq 0, \\ & \theta^T r = 1, \end{aligned} \tag{32}$$

where $r_i = trace(\tilde{G}_i)$. The optimal solution to the formulation in Equation (32) satisfies the constraints in Equation (29).

The formulation in Equation (32) is an approximation to the exact formulation in Equation (29). We use the approximate formulation in our experiments in Section 5, and empirical results show that it achieves comparable performance with other exact formulations.

3.2 QCQP Formulation

Although the approximate SDP formulation in the last section is scalable in terms of the number of classes, interior point algorithms for solving SDP have an inherently large time complexity, and thus it cannot be applied to large-scale problems. In this subsection, we propose a QCQP formulation which is more efficient than its SDP counterpart. The derivations here are similar to those in Section 2.2.

In order to formulate the multi-class RKDA kernel learning problem into a QCQP problem, we first consider the minimization of the following objective function:

$$F_6(W,K) = \sum_{i=1}^{k} \left(||(\phi_K(X)P)^T w_i - h_i||^2 + \lambda ||w_i||^2 \right), \tag{33}$$

where $W = [w_1, \cdots, w_k]$. It is clear that for a fixed K and λ, the computation of w_i and w_j for $i \neq j$ is independent of each other. By extending the results from Lemma 2.1 and Lemma 3.1, it is easy to show that the optimal kernel function K minimizing the objective function in Equation (26) coincides the minimizer of $F_6(W,K)$ in Equation (33). Motivated by this equivalence result, we derive an efficient QCQP formulation for the multi-class RKDA kernel learning problem, as summarized in the following theorem.

Theorem 3.2 *Given a set of p centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix, in the form of a convex linear combination of the given p kernel matrices, that minimizes the objective*

function in Equation (33) can be found by solving the following convex QCQP problem:

$$\begin{aligned} \max_{\beta_1,\cdots,\beta_k,t} \quad & \sum_{j=1}^{k} \beta_j^T h_j - \frac{1}{4}\sum_{j=1}^{k} \beta_j^T \beta_j - \frac{1}{4\lambda} t \\ \textit{subject to} \quad & t \geq \frac{1}{r_i}\sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j,\ i = 1,\cdots,p. \end{aligned} \tag{34}$$

where $r_i = trace(\tilde{G}_i)$.

Proof We first consider the dual formulation of the minimization of $F_6(W,K)$ in terms of W for fixed K and λ. Denote

$$\eta_i = (\phi_K(X)P)^T w_i - h_i.$$

It follows that

$$F_6(w,K) = \sum_{i=1}^{k} ||\eta_i||^2 + \lambda \sum_{i=1}^{k} ||w_i||^2.$$

Define the Lagrangian function of this problem as follows:

$$L(\{\eta_i\}_{i=1}^k, w, \{\beta_i\}_{i=1}^k) = \sum_{i=1}^{k} ||\eta_i||^2 + \lambda \sum_{i=1}^{k} ||w_i||^2 - \sum_{i=1}^{k} \beta_i^T \left((\phi_K(X)P)^T w_i - h_i - \eta_i\right),$$

where the β_i's are the vectors of Lagrangian dual variables. Taking the derivatives of L with respect to η_i and w_i for all i, and setting them equal to zero, we get

$$\begin{aligned} \frac{\partial L}{\partial \eta_i} &= 2\eta_i + \beta_i = 0, \\ \frac{\partial L}{\partial w_i} &= 2\lambda w_i - \phi_K(X)P\beta_i = 0. \end{aligned}$$

Thus, we have

$$\eta_i = -\frac{\beta_i}{2}, \text{ and } w_i = \frac{\phi_K(X)P\beta_i}{2\lambda},$$

and we obtain the following Lagrangian dual function:

$$\begin{aligned} g(\beta_1,\cdots,\beta_k) &= \min_{w_i,\eta_i,i=1,\cdots,k} L(\{\eta_i\}_{i=1}^k, w, \{\beta_i\}_{i=1}^k) \\ &= \sum_{i=1}^{k} \left(-\frac{1}{4}\beta_i^T \left(I + \frac{1}{\lambda} PGP\right)\beta_i + \beta_i^T h_i\right). \end{aligned} \tag{35}$$

The optimal $\beta_1^*,\cdots,\beta_k^*$ can be computed by maximizing $g(\beta_1,\cdots,\beta_k)$ in Equation (35) as

$$(\beta_1^*,\cdots,\beta_k^*) = \operatorname*{argmax}_{\beta_1,\cdots,\beta_k} \left\{ \sum_{i=1}^{k} \left(-\frac{1}{4}\beta_i^T \left(I + \frac{1}{\lambda} PGP\right)\beta_i + \beta_i^T h_i\right)\right\}.$$

Since strong duality holds, the optimal kernel matrix $\tilde{G}$ is given by solving the following optimization problem:

$$\min_{\tilde{G}\in\tilde{\mathcal{G}}} \max_{\beta_1,\cdots,\beta_k} \left\{ \sum_{i=1}^{k} \left(-\frac{1}{4}\beta_i^T \left(I + \frac{1}{\lambda}\tilde{G}\right)\beta_i + \beta_i^T h_i\right)\right\}.$$

Similar to the binary-class case, the above optimization problem can be written as

$$\begin{aligned}
& \min_{\theta:\theta\geq 0,\theta^T r=1} \max_{\beta_1,\cdots,\beta_k} \left\{ \sum_{j=1}^{k} \left(-\frac{1}{4}\beta_j^T \left(I + \frac{1}{\lambda}\sum_{i=1}^{p} \left(\theta_i \tilde{G}_i\right) \right) \beta_j + \beta_j^T h_j \right) \right\} \qquad (36)\\
= & \max_{\beta_1,\cdots,\beta_k} \min_{\theta:\theta\geq 0,\theta^T r=1} \left\{ \sum_{j=1}^{k} \left(-\frac{1}{4}\beta_j^T \left(I + \frac{1}{\lambda}\sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta_j + \beta_j^T h_j \right) \right\} \\
= & \max_{\beta_1,\cdots,\beta_k} \min_{\theta:\theta\geq 0,\theta^T r=1} \left\{ -\frac{1}{4}\sum_{j=1}^{k} \beta_j^T \beta_j - \frac{1}{4\lambda}\sum_{i=1}^{p} \theta_i \left(\sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j \right) + \sum_{j=1}^{k} \beta_j^T h_j \right\} \\
= & \max_{\beta_1,\cdots,\beta_k} \left\{ \sum_{j=1}^{k} \beta_j^T h_j - \frac{1}{4}\sum_{j=1}^{k} \beta_j^T \beta_j - \frac{1}{4\lambda} \max_{\theta:\theta\geq 0,\theta^T r=1} \left\{ \sum_{i=1}^{p} \theta_i \left(\sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j \right) \right\} \right\} \\
= & \max_{\beta_1,\cdots,\beta_k} \left\{ \sum_{j=1}^{k} \beta_j^T h_j - \frac{1}{4}\sum_{j=1}^{k} \beta_j^T \beta_j - \frac{1}{4\lambda} \max_i \left(\frac{1}{r_i} \sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j \right) \right\}.
\end{aligned}$$

By constraining

$$\max_i \left(\frac{1}{r_i} \sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j \right) \leq t$$

and putting t in the objective function, we prove the formulation in Equation (34). ■

3.3 SILP Formulation

The QCQP formulation in Theorem 3.2 has a worse-case time complexity of $O(pk^2n^2 + k^3n^3)$, which is cubic in terms of the number of classes and the number of data points. We show in this subsection that the RKDA kernel learning problem in the multi-class case can be formulated as an SILP problem, as summarized in the following theorem.

Theorem 3.3 *Given a set of p centered kernel matrices $\tilde{G}_1, \cdots, \tilde{G}_p$, the optimal kernel matrix, in the form of a convex linear combination of the given p kernel matrices, that minimizes the objective function in Equation (33) can be found by solving the following SILP problem:*

$$\begin{aligned}
\max_{\theta,\gamma} \quad & \gamma \qquad (37)\\
\textit{subject to} \quad & \theta \geq 0, \\
& \theta^T r = 1, \\
& \sum_{i=1}^{p} \theta_i S_i(\beta) \geq \gamma, \quad \textit{for all } \beta,
\end{aligned}$$

where $S_i(\beta)$ is defined as

$$S_i(\beta) = \sum_{j=1}^{k} \left(\frac{r_i}{4}\beta_j^T \beta_j + \frac{1}{4\lambda}\beta_j^T \tilde{G}_i \beta_j - r_i \beta_j^T h_j \right), \quad \textit{for } i = 1, \cdots, p, \qquad (38)$$

$r = (r_1, \cdots, r_p)^T$, and $r_i = trace(\tilde{G}_i)$.

Formulation	SDP	QCQP	SILP
Complexity	$O\left((p+n)^2(k+n)^{2.5}\right)$	$O(pk^2n^2+k^3n^3)$	$O(n^3Ite)$

Table 1: Time complexity of the proposed multi-class RKDA kernel learning formulations: p is the number of candidate kernels, n is the number of training samples, k is the number of classes, and Ite is the number of iterations in SILP.

Proof The proof follows the same procedure as in Theorem 2.3 by starting from Equation (36) and changing the definition of $S_i(\beta)$ from Equation (18) to Equation (38). ■

Note that the only difference between formulations in Theorem 2.3 and Theorem 3.3 lies in the definitions of $S_i(\beta)$. To find the β_j, for $j = 1,\cdots,k$, that maximize the constraint violation in the multi-class case, we need to solve the following k systems of linear equations:

$$\left(\frac{1}{2}I+\frac{1}{2\lambda}\sum_{i=1}^{p}\theta_i\tilde{G}_i\right)\beta_j = h_j, \quad \text{for } j = 1,\cdots,k.$$

Note that the coefficient matrix is the same for all of the k linear systems. Thus the LU decomposition (Golub and Van Loan, 1996) needs to be computed only once, and only the forward/backward substitution needs to be performed k times to obtain the solutions.

3.4 Time Complexity Analysis

In this subsection, we analyze the time complexity of the proposed formulations in the multi-class case. By following similar analysis in the binary-class case, we can show that the proposed (approximate) SDP and QCQP formulations have worse-case time complexity of $O\left((p+n)^2(k+n)^{2.5}\right)$ and $O(pk^2n^2+k^3n^3)$, respectively. For the SILP formulation in the multi-class case, the k linear systems involved in each iterative step share the same coefficient matrix, and they can be solved in $O(n^3)$ time. Thus, the overall complexity is still $O(n^3Ite)$ where Ite is the number of iterations. The complexity of multi-class RKDA kernel learning formulations is summarized in Table 1.

4. Joint Kernel and Regularization Parameter Learning

The formulations presented in the last two sections focus on the estimation of the kernel matrix only, while the regularization parameter λ is pre-specified. In some cases, the performance of RKDA algorithm depends critically on the value of λ. In this section, we show that all the formulations proposed in this paper can be reformulated equivalently, and this new formulation leads naturally to the estimation of the regularization parameter λ in a joint framework. The detailed derivations in this section are similar to those presented in Sections 2 and 3.

4.1 Joint Learning for Binary-class Problems

One key advantage of the kernel learning formulation in Equation (8) in comparison with the one in Kim et al. (2006) is that the regularization parameter λ can also be estimated in a joint optimization

framework. In particular, all the formulations (SDP, QCQP, and SILP) for the binary-class RKDA kernel learning problems, presented in Theorems 2.1–2.3, can be recast to optimize the regularization parameter simultaneously. The next three subsections provide details of these reformulations.

4.1.1 SDP FORMULATION

For the estimation of regularization parameter, we consider a slightly modified version of the regularized least squares formulation, which is equivalent to the standard formulation in Equation (12). The modified version minimizes the following objective function:

$$F_7(w,K,\tau) = \tau||(\phi_K(X)P)^T w - a||^2 + ||w||^2, \tag{39}$$

where $\tau = 1/\lambda$. We will first consider the case when τ is fixed. We will then extend to the general case when τ is optimized jointly.

The optimal w^* that minimizes the objective function in Equation (39) for a fixed K and a fixed τ is given by

$$\begin{aligned} w^* &= \left(\frac{1}{\tau}I + \phi_K(X)P\phi_K(X)^T\right)^{-1}\phi_K(X)Pa \\ &= \tau\phi_K(X)\left(I - P\left(\frac{1}{\tau}I + PGP\right)^{-1}PG\right)a. \end{aligned}$$

The optimal value of the objective function in Equation (39) is given by

$$F_7^*(K,\tau) = a^T\left(\frac{1}{\tau}I + \tilde{G}\right)^{-1}a, \tag{40}$$

where $\tilde{G} = PGP$.

We can observe from Equation (40) that the identity matrix appears in exactly the same form as other kernel matrices. We can thus treat the regularization parameter as one of the coefficients for the kernel matrix and optimize them simultaneously. This leads to the following formulation:

$$\begin{aligned} \min_{t,\tilde{\theta}} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} \sum_{i=0}^p \tilde{\theta}_i\tilde{G}_i & a \\ a^T & t \end{pmatrix} \succeq 0, \\ & \tilde{\theta} \geq 0, \\ & \sum_{i=0}^p \tilde{\theta}_i \text{trace}(\tilde{G}_i) = 1, \end{aligned} \tag{41}$$

where $\tilde{\theta} = [\theta_0, \theta_1, \cdots, \theta_p]^T$, $\theta_0 = \frac{1}{\tau} = \lambda$, and $\tilde{G}_0 = I$.

4.1.2 QCQP FORMULATION

In order to cast the formulation in Theorem 2.2 to optimize the regularization parameter, we again start from the modified least square problem in Equation (39). By following the same procedure as

in Theorem 2.2, the optimization problem in Equation (15) can be expressed as

$$\begin{aligned} & \min_{\theta:\theta\geq 0,\theta^T r=1} \max_{\beta} \left\{ -\frac{1}{4}\beta^T \left(\frac{1}{\tau} I + \sum_{i=1}^{p} \theta_i \tilde{G}_i \right) \beta + \beta^T a \right\} \\ = & \max_{\beta} \min_{\tilde{\theta}:\tilde{\theta}\geq 0,\tilde{\theta}^T r=1} \left\{ -\frac{1}{4}\beta^T \left(\sum_{i=0}^{p} \tilde{\theta}_i \tilde{G}_i \right) \beta + \beta^T a \right\}, \end{aligned} \tag{42}$$

where $\theta_0 = \frac{1}{\tau}$, and $\tilde{G}_0 = I$. This can be formulated to optimize the regularization parameter as one of the coefficients for the kernel matrix as follows:

$$\begin{aligned} \max_{\beta,t} \quad & \beta^T a - \frac{1}{4} t \\ \text{subject to} \quad & t \geq \frac{1}{r_i} \beta^T \tilde{G}_i \beta, \; i = 0, \cdots, p. \end{aligned} \tag{43}$$

This problem is a quadratically constrained linear program.

4.1.3 SILP FORMULATION

The SILP formulation proposed in Theorem 2.3 for the binary-class problem can also be reformulated to optimize λ jointly. It follows from Equation (42) that this joint learning problem can be formulated as follows:

$$\begin{aligned} \max_{\tilde{\theta},\gamma} \quad & \gamma \\ \text{subject to} \quad & \tilde{\theta} \geq 0, \\ & \tilde{\theta}^T r = 1, \\ & \sum_{i=0}^{p} \theta_i S_i(\beta) \geq \gamma, \quad \text{for all } \beta, \end{aligned} \tag{44}$$

where $S_i(\beta)$ is defined as

$$S_i(\beta) = \frac{1}{4}\beta^T \tilde{G}_i \beta - r_i \beta^T a, \quad \text{for } i = 0, \cdots, p,$$

$r = (r_0, \cdots, r_p)^T$, $r_i = \text{trace}(\tilde{G}_i)$, $\tilde{\theta} = [\theta_0, \theta_1, \cdots, \theta_p]^T$, $\theta_0 = \frac{1}{\tau} = \lambda$, and $\tilde{G}_0 = I$.

4.2 Joint Learning for Multi-class Problems

All formulations for the multi-class RKDA kernel learning problems presented in Section 3 can be recast to optimize the regularization parameter jointly. The next three subsections provide details of these reformulations.

4.2.1 SDP FORMULATION

In order to incorporate λ in the optimization problem, we modify the objective function in Equation (26) as follows:

$$F_8(W,K,\tau) = \sum_{i=1}^{k} \frac{(w_i^T \phi_K(X) h_i)^2}{w_i^T (\tau \Sigma_K + I) w_i}.$$

By following the same derivation in Lemma 3.1 and noticing the relationship with the binary-class case, we derive the following SDP formulation for the multi-class RKDA kernel learning problem:

$$\begin{aligned}
\min_{t_1,\cdots,t_k,\tilde{\theta}} \quad & \sum_{j=1}^{k} t_j \\
\text{subject to} \quad & \begin{pmatrix} \sum_{i=0}^{p} \tilde{\theta}_i \tilde{G}_i & h_1 & h_2 & \cdots & h_k \\ h_1^T & t_1 & 0 & \cdots & 0 \\ h_2^T & 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_k^T & 0 & 0 & \cdots & t_k \end{pmatrix} \succeq 0, \\
& \tilde{\theta} \geq 0, \\
& \tilde{\theta}^T r = 1,
\end{aligned} \tag{45}$$

where $\tilde{\theta} = [\theta_0, \theta_1, \cdots, \theta_p]^T$, $\theta_0 = \frac{1}{\tau} = \lambda$, and $\tilde{G}_0 = I$.

4.2.2 QCQP FORMULATION

Similar to the binary-class case, we modify the least square problem in Equation (33) as follows:

$$F_9(W,K,\tau) = \sum_{i=1}^{k} \left(\tau ||(\phi_K(X)P)^T w_i - h_i||^2 + ||w_i||^2 \right),$$

where $\tau = 1/\lambda$. By following the same derivation as in Theorem 3.2, we obtain the following joint optimization problem:

$$\begin{aligned}
\max_{\beta_1,\cdots,\beta_k,t} \quad & \sum_{j=1}^{k} \beta_j^T h_j - \frac{1}{4} t \\
\text{subject to} \quad & t \geq \frac{1}{r_i} \sum_{j=1}^{k} \beta_j^T \tilde{G}_i \beta_j, \; i = 0, \cdots, p.
\end{aligned} \tag{46}$$

This is a quadratically constrained linear program.

4.2.3 SILP FORMULATION

Similar to the reformulation in the binary-class case, the SILP formulation for multi-class problems can also be formulated to optimize λ simultaneously as follows:

$$\begin{aligned} \max_{\tilde{\theta},\gamma} \quad & \gamma \\ \text{subject to} \quad & \tilde{\theta} \geq 0, \\ & \tilde{\theta}^T r = 1, \\ & \sum_{i=0}^{p} \theta_i S_i(\beta) \geq \gamma, \quad \text{for all } \beta, \end{aligned} \tag{47}$$

where

$$S_i(\beta) = \sum_{j=1}^{k} \left(\frac{1}{4} \beta_j^T \tilde{G}_i \beta_j - r_i \beta_j^T h_j \right), \quad \text{for } i = 0, \cdots, p,$$

$r = (r_0, \cdots, r_p)^T$, $r_i = \text{trace}(\tilde{G}_i)$, $\tilde{\theta} = [\theta_0, \theta_1, \cdots, \theta_p]^T$, $\theta_0 = \frac{1}{\tau} = \lambda$, and $\tilde{G}_0 = I$.

The reformulations to optimize λ simultaneously proposed in this section are motivated from Lanckriet et al. (2004b) and De Bie et al. (2003). As has been show in Lanckriet et al. (2004b), this joint optimization of λ works well in most cases in comparison with the simple approach of pre-specifying λ, but improved performance is not guaranteed.

5. Experimental Study

We conduct extensive experiments in this section to compare various aspects of relevant algorithms. The first part of the experiments focuses on combining kernel matrices derived from a single source of data. We demonstrate the effectiveness of the proposed MKL formulations for heterogeneous data integration in the second part of the experiments. The SDP formulations in Equations (8), (32), (41), and (45) are solved using the optimization package SeDuMi (Sturm, 1999). The QCQP formulations in Equations (13), (34), (43), and (46) are solved using the MOSEK package (Andersen and Andersen, 2000). The linear programs involved in the SILP formulations in Equations (16), (37), (44), and (47) are solved using the MATLAB[1] build-in function *linprog*. The tolerance parameter ε, defined in Equation (22), is set to 5×10^{-4}. The source codes of the proposed formulations for the experiments are available online.[2]

We first evaluate the proposed formulations for binary-class problems in Section 5.1. The experimental results and analysis for the multi-class formulations are presented in Section 5.2. We demonstrate the effectiveness of the proposed formulations for heterogeneous data integration in Section 5.3. In Section 5.4, we analyze the relationship between RKDA and SVM, and Section 5.5 studies the effect of regularization parameter on classification performance.

5.1 Experiments on Binary-class Problems

In the binary-class case, we compare our formulations with the 1-norm soft margin SVM, 2-norm soft margin SVM with and without the regularization parameter C optimized jointly as proposed in

1. The URL is `http://www.mathworks.com`.
2. The URL is `http://www.public.asu.edu/~jye02/Software/DKL/`.

sonar	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ/C	TSA
SDP_θ	0	0	0	0	0.550	0.307	0.022	0.041	0.029	0.050	5.0e-04	89.27±5.34
$SDP_{\theta,\lambda}$	0	0	0	0	0.550	0.307	0.022	0.041	0.029	0.050	3.1e-08	89.35±5.34
$QCQP_\theta$	0.003	0.003	0.004	0.011	0.444	0.375	0.046	0.034	0.035	0.048	5.0e-04	89.76±5.34
$QCQP_{\theta,\lambda}$	0	0	0	0	0.550	0.307	0.022	0.041	0.029	0.050	5.0e-02	89.35±5.34
$SILP_\theta$	0	0	0	0	0.459	0.406	0.011	0.034	0.032	0.059	5.0e-04	89.76±5.37
$SILP_{\theta,\lambda}$	0	0	0	0	0.547	0.313	0.023	0.031	0.034	0.052	4.2e-10	89.43±5.18
SDP_{Kim}	0.167	0.048	0.175	0.072	0.251	0.173	0.031	0.025	0.015	0.044	1.0e-08	88.46±5.28
SM1	0	0	0	0.040	3.953	5.514	0.491	0	0	0	1	89.75±4.90
SM2	0	0	0	0	2.875	6.765	0.359	0	0	0	1	89.59±5.24
$SM2_C$	0	0.011	0.014	0.084	4.253	6.038	0.570	0.004	0.001	0	5.5e+7	89.84±4.80
$RKDA_{K,\lambda}$[3]	0	0	0	0	0	3	14	11	2	0	–	89.67±6.62
$SVM_{K,C}$[4]	0	0	0	0	0	2	16	7	5	0	–	89.35±5.18
$RKDA_\lambda$[5]	53.65	54.95	60.24	73.57	84.95	90.56	89.91	86.99	85.52	84.95	–	–
SVM_C[6]	53.65	54.63	59.91	73.41	86.09	89.67	90.65	89.59	86.58	84.22	–	–

Table 2: Comparison of twelve methods on the *sonar* data set. The twelve methods, listed from top to bottom are: SDP formulation with λ fixed as proposed in Theorem 2.1, SDP formulation with λ optimized jointly as proposed in Equation (41), QCQP formulation with λ fixed as proposed in Theorem 2.2, QCQP formulation with λ optimized jointly as proposed in Equation (43), SILP formulation with λ fixed as proposed in Theorem 2.3, SILP formulation with λ optimized jointly as proposed in Equation (44), SDP formulation proposed in Kim et al. (2006), 1-norm soft margin SVM, 2-norm soft margin SVM without and with C optimized as proposed in Lanckriet et al. (2004b), RKDA and SVM with the kernels and regularization parameters selected by double cross-validation. Generally, subscripts of names in the first column are used to denote quantities that are optimized. The ten pre-specified kernels are all RBF kernels and the σ values used are 0.10, 0.22, 0.46, 1.00, 2.15, 4.46, 10.00, 21.54, 46.42, 100.00, as in Kim et al. (2006). The table is partitioned into three sections row-wise. In the first section, the columns headed with θ_i are the coefficients learned from the corresponding methods. The coefficients for the proposed six formulations are normalized to sum to one while those for other compared approaches are reported as obtained from their formulations. The column headed with λ/C provides the values of the regularization parameters, whether fixed or learned, and the test set accuracies and standard deviations are given in the last column. The second section includes RKDA and SVM with kernel and regularization parameter chosen by double cross-validation. We also report the number of times that a particular kernel is selected by cross-validation. The third section shows the accuracies of RKDA and SVM when the kernel is fixed and the regularization parameters chosen by cross-validation. Dashes are used to denote non-applicable items.

Lanckriet et al. (2004b), and the SDP formulation proposed in Kim et al. (2006). Also, we use double cross-validation to choose kernels and regularization parameters for SVM and RKDA. The 1-norm SVM classifier used is the LIBSVM package (Chang and Lin, 2001) and the 2-norm SVM code was obtained by adapting Anton Schwaighofer's implementation.[7]

Four data sets are used in the binary-class case. The *sonar, ionosphere*, and *cancer* data were retrieved from the UCI Machine Learning Repository (Newman et al., 1998). The *heart* data were

3. The number of times that a kernel is chosen by doubly cross-validated RKDA over 30 randomizations.
4. The number of times that a kernel is chosen by doubly cross-validated SVM over 30 randomizations.
5. Accuracy of RKDA when the kernel is fixed to each of the ten candidate kernels and λ is chosen by cross-validation.
6. Accuracy of SVM when the kernel is fixed to each of the ten candidate kernels and C is chosen by cross-validation.
7. The URL is `http://ida.first.fraunhofer.de/~anton/software.html`.

obtained from the STATLOG project.[8] All data are normalized. For each data set, we randomly partition the entire data set into training and test sets using the ratio 8:2. Ten RBF kernels are constructed from the training set data with different choices of the parameter σ as in Kim et al. (2006). Then the ten kernels are fed into the optimization software packages to obtain the corresponding coefficients for each kernel. Finally, the kernels are combined and used to compute the accuracy. For formulations SDP_θ, QCQP_θ, and SILP_θ, the λ value is fixed to 5.0×10^{-4}. For SDP_{Kim}, this value is fixed to 10^{-8}, as used in Kim et al. (2006). Following Lanckriet et al. (2004b), we fix C to 1 for SM1 and SM2.

Tables 2–5 present the experimental results on *sonar, heart, ionosphere*, and *cancer* data sets, respectively. In terms of performance, formulations that optimize λ jointly achieve similar accuracies to the ones with λ fixed. Note that for our experiments, all the data are normalized and the λ value is tuned manually for formulations with λ fixed. In practice, the optimal λ value is data-dependent. Thus, formulations that optimize λ jointly are expected to work better in such situations. In cases where no numerical problems have been reported, all the twelve compared methods achieve similar performance. However, for the first ten methods, there is no need for cross-validation, and they can be used for heterogeneous data integration from various sources.

For MKL formulations in Tables 2–5, we present the coefficients learned for each kernel. For doubly cross-validated methods, that is, $\text{RKDA}_{K,\lambda}$ and $\text{SVM}_{K,C}$, we record the number of times that a particular kernel has been selected in cross-validation. To understand the relative importance of each kernel when they are used individually, we fix the kernel to each of the ten pre-specified kernels and tune the regularization parameter using cross-validation and the accuracy of each kernel is recorded. We expect these quantities to have some relationship with the coefficients learned by solving convex programs. For the *sonar* data, RKDA_λ achieves the best performance on kernels corresponding to θ_6 and θ_7 while SVM_C achieves the highest accuracy on θ_6, θ_7 and θ_8. On the other hand, methods using linear combination of kernels favor kernels corresponding to θ_5 and θ_6. For the *heart* data, cross-validated SVM favors kernels corresponding to θ_9 and θ_{10} (they were chosen 9 and 17 times out of 30, respectively) while cross-validated RKDA uses kernels corresponding to θ_7, θ_8, and θ_9 most frequently. Our six formulations all give kernels corresponding to θ_1 and θ_{10} large weights, especially to θ_1, while SVM-based MKL formulations all set θ_{10} to zero. This may be due to the fact that RKDA and SVM optimize different criteria and thus favor different kernels. Another interesting observation is that all the ten MKL formulations give the first kernel a large weight while it is the worst kernel when used individually. This implies that the best individual kernel may not lead to a large weight when used in combination with others and poorly-performed individual kernel may contain complementary information that is useful when combined with other kernels. Such complementary information can not be incorporated when cross-validation is used to choose a single best kernel. For the *ionosphere* data, the best three individual kernels chosen by cross-validation are kernels corresponding to θ_5, θ_6 and θ_7. Interestingly, the kernel corresponding to θ_5 is assigned a zero weight by nine out of the ten MKL-based methods. For the *cancer* data, all kernels achieve similar performance when used separately while MKL-based formulations tend to assign a large weight to the kernel corresponds to θ_2.

To compare the efficiency of the proposed formulations with methods based on cross-validation, we record the computation time of the proposed QCQP and SILP formulations along with that of methods based on double cross-validation. Figure 1 plots the computation time of these six methods.

8. The URL is `http://www.liacc.up.pt/ML/old/statlog/datasets.html`.

Note that methods based on SDP have a much larger computation time than these six methods and their results are thus omitted. It can be seen that the proposed SILP formulations are more efficient than cross-validation based methods. Note that the convergence rate of the algorithm for solving the QCQP formulation depends on the data and parameter setting. Thus, it may take a relatively long time to converge in some cases, as shown by QCQP_θ on the *cancer* data in Figure 1.

heart	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ/C	TSA
SDP_θ	0.783	0.089	0	0	0	0.001	0	0	0.005	0.123	5.0e-4	81.98±4.27
$\text{SDP}_{\theta,\lambda}$	0.753	0.089	0	0	0	0.001	0	0	0.005	0.123	3.0e-2	81.67±4.49
QCQP_θ	0.734	0.117	0.003	0.001	0.001	0.001	0.002	0.004	0.008	0.129	5.0e-4	81.85±4.17
$\text{QCQP}_{\theta,\lambda}$	0.753	0.089	0	0	0	0.001	0	0	0.005	0.123	1.2e-1	81.67±4.47
SILP_θ	0.742	0.115	0	0	0	0.001	0	0	0.006	0.137	5.0e-4	81.98±4.27
$\text{SILP}_{\theta,\lambda}$	0.744	0.095	0	0	0	0	0	0	0.007	0.121	3.4e-2	81.73±4.23
SDP_{Kim}	0.881	0.036	0.002	0	0	0.001	0.003	0.004	0.009	0.065	1.0e-8	82.22±3.79
SM1	7.688	0.479	0.001	0.002	0.002	0.024	1.813	0	0	0	1	82.59±4.55
SM2	7.317	0.669	0	0	0	0.029	1.994	0	0	0	1	82.71±4.41
SM2_C	6.746	0.626	0	0	0	0.036	1.991	0	0	0	4.4e+5	82.53±4.58
$\text{RKDA}_{K,\lambda}$	0	0	0	0	1	2	7	9	7	4	–	77.35±5.83
$\text{SVM}_{K,C}$	0	0	0	0	0	0	2	2	9	17	–	81.73±4.48
RKDA_λ	58.64	65.06	69.62	73.33	77.28	79.13	78.70	77.65	76.79	75.92	–	–
SVM_C	57.96	64.75	71.79	76.60	79.93	80.30	81.66	81.54	82.22	82.59	–	–

Table 3: See the caption and footnotes of Table 2 for explanation.

ionosphere	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ/C	TSA
SDP_θ	0.362	0.073	0.033	0.108	0	0.147	0.277	0	0	0	5.0e-4	94.67±2.25
$\text{SDP}_{\theta,\lambda}$	0.362	0.073	0.033	0.108	0	0.147	0.277	0	0	0	1.4e-7	94.67±2.25
QCQP_θ	0.222	0.116	0.081	0.074	0.042	0.182	0.236	0.022	0.014	0.012	5.0e-4	94.86±2.39
$\text{QCQP}_{\theta,\lambda}$	0.362	0.073	0.033	0.108	0	0.147	0.277	0	0	0	2.2e-4	94.67±2.25
SILP_θ	0.261	0.080	0.061	0.116	0	0.167	0.316	0	0	0	5.0e-4	94.90±2.33
$\text{SILP}_{\theta,\lambda}$	0.364	0.073	0.028	0.112	0	0.145	0.279	0	0	0	3.6e-9	94.81±2.23
SDP_{Kim}	0.942	0	0	0	0	0.006	0.038	0.013	0.001	0	1.0e-8	89.43±3.98
SM1	3.553	0.672	0.482	0.240	0	4.828	0.221	0	0	0	1	95.28±2.09
SM2	2.883	0.682	0.683	0.196	0	5.305	0.248	0	0	0	1	94.81±2.07
SM2_C	3.910	0.714	0.561	0.255	0	5.300	0.256	0	0	0	1.4e+7	95.19±2.17
$\text{RKDA}_{K,\lambda}$	0	0	0	4	5	10	8	3	0	0	–	92.33±5.51
$\text{SVM}_{K,C}$	0	0	0	0	8	9	7	4	2	0	–	94.48±2.39
RKDA_λ	65.71	76.47	90.33	92.14	93.33	94.28	93.14	91.71	90.61	89.00	–	–
SVM_C	65.38	66.57	89.38	93.00	94.57	95.04	93.80	93.42	92.61	91.95	–	–

Table 4: See the caption and footnotes of Table 2 for explanation.

5.2 Experiments on Multi-class Problems

In the multi-class experiments, we compare our formulations with KRDA and SVM with kernels and regularization parameters tuned using double cross-validation. The methods proposed in Lanckriet et al. (2004b) and Kim et al. (2006) are only applicable to binary-class problems. Five data sets with different numbers of classes are used for this experiment. The *USPS* handwritten digits database was described in Hull (1994). We choose the first $3, 6$, and 8 classes with 100 data points in each class for the experiment. The *wine* data set was obtained from UCI Machine Learning Repository and the *satimage* and *segment* were obtained from the STATLOG project. We use the first $3, 5$, and 6 classes for the *satimage* data and the first 3 and 4 classes for the *segment* data. The *waveform*

cancer	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ/C	TSA
SDP_θ	0.013	0.006	0.014	0	0.018	0.044	0.061	0.101	0.280	0.463	5.0e-4	96.05±2.65
$\text{SDP}_{\theta,\lambda}$	0	0.532	0.096	0.040	0.008	0.020	0.244	0.020	0	0.048	1.0e-8	96.00±1.44
QCQP_θ	0.147	0.312	0.207	0.080	0.052	0.055	0.051	0.038	0.031	0.028	5.0e-4	97.01±1.31
$\text{QCQP}_{\theta,\lambda}$	0.003	0.662	0.111	0.042	0.010	0.015	0.134	0.007	0	0.004	4.3e-3	96.20±2.21
SILP_θ	0	0.468	0.298	0.022	0.010	0.020	0.170	0.007	0	0.005	5.0e-4	97.01±1.20
$\text{SILP}_{\theta,\lambda}$	0.003	0.663	0.105	0.047	0.009	0.014	0.132	0.009	0	0.005	1.3e-2	96.98±1.28
SDP_{Kim}	0.970	0.006	0.005	0.004	0.004	0.003	0.003	0.002	0.002	0.002	5.0e-4	73.43±4.28
SM1	1.797	5.706	0.179	0.008	0	2.308	0	0	0	0	1	97.08±1.27
SM2	1.483	5.541	0.402	0.023	0.006	2.527	0.013	0	0	0	1	97.15±1.22
SM2_C	1.690	4.855	0.546	0.047	0.003	2.521	0.015	0	0	0	1.0e+4	97.01±1.22
$\text{RKDA}_{K,\lambda}$	0	0	0	2	8	2	3	4	4	7	–	95.79±1.55
$\text{SVM}_{K,C}$	0	0	0	0	0	7	10	4	6	3	–	96.81±1.28
RKDA_λ	94.54	95.32	96.05	96.15	96.30	95.74	95.59	95.59	95.49	95.64	–	–
SVM_C	92.21	94.93	96.03	96.30	96.81	96.88	96.86	96.66	96.69	96.64	–	–

Table 5: See the caption and footnotes of Table 2 for explanation.

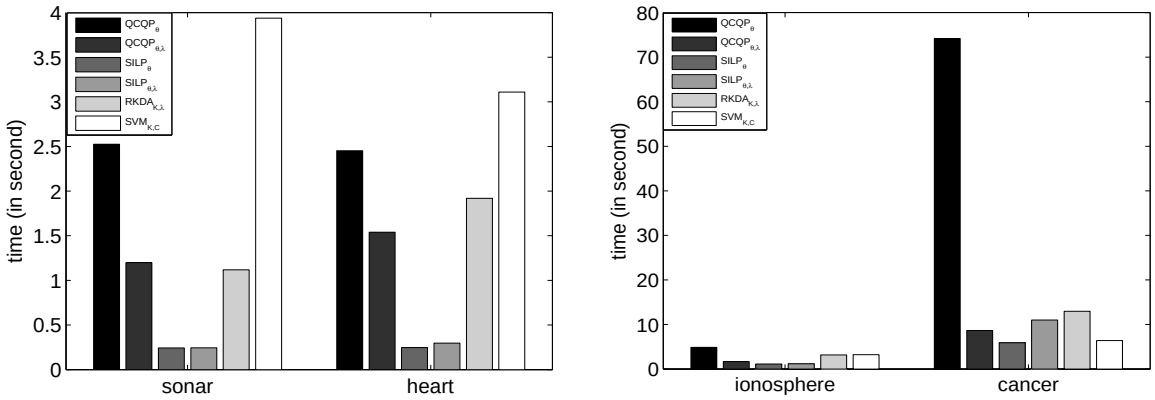

Figure 1: Computation time (in seconds) of the six methods on four binary-class data sets.

data set was described in Breiman et al. (1984) and are also available from UCI Machine Learning Repository. For each data set, we randomly partition the entire set into two subsets with 60% of the samples in the training set and 40% in the test set. Ten RBF kernels, with σ assigned the same values as in the binary-class case, are constructed from the training set.

Tables 6–15 present the experimental results on the ten data sets. In general, all the six proposed formulations achieve similar performance on the ten data sets. Compared to the QCQP and SILP formulations which are exact, our approximate SDP formulation for the multi-class problems work well in most cases. This implies that the approximate formulation is close to the exact one while the computational cost is lower. Furthermore, methods based on MKL and cross-validation achieve similar performance on all of the data sets.

In order to gain insights into the relative importance of each kernel when used in combination or separately, we use a similar experimental setup to the binary-class case. We found that for the *USPS*(3),[9] *USPS*(6), and *USPS*(8) data, all eight compared approaches favor the kernels corresponding to θ_9 and θ_{10}. Similar behavior has been observed for the *waveform*(3) data where only the last two kernels are selected by cross-validation and they are given large weights by all six MKL-based

9. The number in the parentheses denotes the number of classes used in the experiment.

USPS(3)[10]	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0	0	0	0	0	0.027	0.023	0.012	0.518	0.420	5.0e-4	99.64±0.57
$SDP_{\theta,\lambda}$	0.007	0.004	0.015	0.016	0.013	0.036	0.022	0.014	0.493	0.379	6.3e-7	99.69±0.46
$QCQP_\theta$	0	0	0	0	0	0.037	0.052	0.040	0.372	0.498	5.0e-4	99.72±0.51
$QCQP_{\theta,\lambda}$	0.007	0.004	0.021	0.009	0.008	0.067	0.029	0.054	0.345	0.457	1.2e-5	99.64±0.47
$SILP_\theta$	0	0	0	0	0	0.037	0.052	0.043	0.370	0.499	5.0e-4	99.72±0.51
$SILP_{\theta,\lambda}$	0.007	0.005	0.019	0.011	0.006	0.069	0.027	0.057	0.343	0.457	3.6e-7	99.61±0.48
$RKDA_{K,\lambda}$[11]	0	0	0	0	0	0	0	0	8	22	–	98.97±1.11
$SVM_{K,C}$[12]	0	0	0	0	0	0	0	0	24	6	–	99.50±0.60

Table 6: Comparison of eight methods on the *USPS* data set when the first three classes are used. The eight methods, listed from top to bottom, are the SDP formulation with λ fixed as proposed in Theorem 3.1, the SDP formulation with λ optimized jointly as proposed in Equation (45), the QCQP formulation with λ fixed as proposed in Theorem 3.2, the QCQP formulation with λ optimized jointly as proposed in Equation (46), the SILP formulation with λ fixed as proposed in Theorem 3.3, the SILP formulation with λ optimized jointly as proposed in Equation (47), RKDA and SVM with kernels and regularization parameters chosen by double cross-validation. Generally, subscripts of names in the first column are used to denote quantities that are optimized. Ten RBF kernels are pre-specified and the values for σ are the same as those used in the binary-class case (see caption of Table 2). This table is partitioned into two sections row-wise. In the first section, the columns headed with θ_i present the coefficients learned from each method. Note that all coefficients are normalized to sum to one. This is followed by the values for the λ, whether fixed or learned. The test set accuracies are given in the last column. In the second section, we report the number of times that each kernel has been selected by double cross-validation and the accuracies. Dashes are used to denote non-applicable items.

USPS(6)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0	0	0	0	0	0.001	0.009	0.195	0.655	0.141	5.0e-4	98.40±0.80
$SDP_{\theta,\lambda}$	0.018	0.001	0.011	0.038	0.013	0.011	0.036	0.222	0.516	0.134	1.9e-6	98.33±0.88
$QCQP_\theta$	0	0	0	0	0	0.001	0.023	0.165	0.564	0.247	5.0e-4	98.36±0.82
$QCQP_{\theta,\lambda}$	0.020	0.002	0.003	0.035	0.025	0.008	0.063	0.165	0.463	0.216	2.8e-5	98.28±0.89
$SILP_\theta$	0	0	0	0	0	0.002	0.028	0.156	0.569	0.245	5.0e-4	98.35±0.84
$SILP_{\theta,\lambda}$	0.021	0	0.003	0.037	0.017	0.011	0.064	0.169	0.459	0.218	1.4e-8	98.29±0.88
$RKDA_{K,\lambda}$	0	0	0	0	0	0	0	0	20	10	–	98.08±0.85
$SVM_{K,C}$	0	0	0	0	0	0	0	0	26	4	–	98.11±1.02

Table 7: See the caption and footnotes of Table 6 for explanation.

approaches. Thus for these data sets, the kernels selected by cross-validation and multiple kernel learning (MKL) agree. In contrast, for the *satimage*(3), *satimage*(5), and *satimage*(6) data sets, the proposed MKL-based approaches assign large weights to the first five kernels. In particular, θ_2, θ_3, and θ_5 are given large values for the *satimage*(3) data; θ_2, θ_4, and θ_5 are given large values for the *satimage*(5) data; θ_1, θ_2, and θ_4 are given large values for the *satimage*(6) data. On the other hand,

10. The number in parenthesis denotes the number of classes used in the experiments.
11. RKDA with kernel and λ chosen by double cross-validation. The first ten columns show the number of times that a kernel is chosen by doubly cross-validated RKDA over 30 randomizations.
12. SVM with kernel and λ chosen by double cross-validation. The first ten columns show the number of times that a kernel is chosen by doubly cross-validated SVM over 30 randomizations.

USPS(8)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0	0	0	0	0	0.034	0.057	0.118	0.792	0	5.0e-4	97.60±0.83
$\text{SDP}_{\theta,\lambda}$	0.032	0.002	0.016	0.011	0.009	0.130	0.056	0.104	0.641	0	4.4e-6	97.64±0.70
QCQP_θ	0	0	0	0	0	0.001	0.116	0.053	0.697	0.133	5.0e-4	97.57±0.77
$\text{QCQP}_{\theta,\lambda}$	0.025	0.003	0.021	0.023	0.007	0.066	0.149	0.029	0.573	0.106	3.2e-5	97.65±0.72
SILP_θ	0	0	0	0	0	0	0.112	0.060	0.695	0.134	5.0e-4	97.51±0.77
$\text{SILP}_{\theta,\lambda}$	0.024	0	0.020	0.025	0.006	0.071	0.144	0.034	0.571	0.106	7.8e-9	97.64±0.74
$\text{RKDA}_{K,\lambda}$	0	0	0	0	0	0	0	0	12	18	–	97.53±0.78
$\text{SVM}_{K,C}$	0	0	0	0	0	0	0	0	18	12	–	97.10±0.82

Table 8: See the caption and footnotes of Table 6 for explanation.

wine(3)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0.044	0.147	0.104	0.415	0.104	0.012	0.008	0.016	0.018	0.133	5.0e-4	97.98±1.60
$\text{SDP}_{\theta,\lambda}$	0.065	0.177	0.086	0.394	0.100	0.011	0.008	0.018	0.015	0.126	2.4e-7	97.79±1.60
QCQP_θ	0.028	0.128	0.202	0.181	0.302	0.010	0.005	0.01	0.009	0.125	5.0e-4	98.12±1.49
$\text{QCQP}_{\theta,\lambda}$	0.046	0.169	0.171	0.177	0.289	0.009	0.004	0.011	0.004	0.123	9.3e-7	98.12±1.45
SILP_θ	0.023	0.133	0.205	0.178	0.305	0.010	0.003	0.009	0.010	0.125	5.0e-4	98.12±1.49
$\text{SILP}_{\theta,\lambda}$	0.045	0.162	0.182	0.172	0.287	0.010	0.008	0	0.009	0.124	2.2e-7	98.12±1.45
$\text{RKDA}_{K,\lambda}$	0	0	0	2	5	2	6	3	5	7	–	98.31±1.63
$\text{SVM}_{K,C}$	0	0	0	6	4	11	4	4	1	0	–	97.65±1.90

Table 9: See the caption and footnotes of Table 6 for explanation.

satimage(3)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0	0.157	0.247	0.046	0.353	0.08	0.024	0.046	0.017	0.032	5.0e-4	98.03±0.94
$\text{SDP}_{\theta,\lambda}$	0.003	0.251	0.193	0.033	0.349	0.06	0.02	0.05	0.012	0.029	9.0e-4	98.00±0.97
QCQP_θ	0	0.131	0.249	0.068	0.265	0.165	0.034	0.03	0.009	0.049	5.0e-4	98.06±0.96
$\text{QCQP}_{\theta,\lambda}$	0.002	0.229	0.193	0.049	0.281	0.115	0.055	0.023	0.003	0.048	1.8e-3	98.08±0.93
SILP_θ	0	0.133	0.246	0.073	0.252	0.181	0.026	0.033	0.006	0.051	5.0e-4	98.06±0.96
$\text{SILP}_{\theta,\lambda}$	0.003	0.226	0.197	0.047	0.274	0.133	0.042	0.019	0.004	0.053	1.6e-3	98.08±0.93
$\text{RKDA}_{K,\lambda}$	0	0	0	3	0	7	5	6	5	4	–	97.56±1.26
$\text{SVM}_{K,C}$	0	0	0	0	5	5	8	3	5	4	–	97.92±1.09

Table 10: See the caption and footnotes of Table 6 for explanation.

the two methods based on cross-validation tend to use the last five kernels more frequently than the first five kernels. This demonstrates that the best kernels used in combination and separately differ significantly for the *satimage* data set. We expect that complementary information exists among kernels for this data set such that a subset of kernels can be combined to obtain the optimal performance though none of them is the best kernel when used individually. Similar phenomenon can be observed from the *segment*(3) and *segment*(4) data sets in which the first kernel is assigned the largest weight by MKL-based formulations while it is never selected by cross-validation. This analysis shows that the information used by methods based on MKL and cross-validation may coincide or differ depending on the data.

To compare the efficiency of various methods, we report the computation time of the eight methods on the ten data sets in Table 16. It can be seen that the SDP formulations are much slower than methods based on cross-validation due to its inherent large complexity. The QCQP formulations are relatively efficient for data sets with a small number of classes. When the number of classes increases, their computation time increases rapidly. This is consistent with the theoretical

satimage(5)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0	0.302	0.244	0.196	0.225	0	0	0	0	0.034	5.0e-4	93.50±1.57
$SDP_{\theta,\lambda}$	0.002	0.477	0.083	0.225	0.184	0	0	0	0	0.030	2.1e-7	93.42±1.69
$QCQP_\theta$	0	0.165	0.469	0.001	0.251	0.068	0	0	0.004	0.043	5.0e-4	93.52±1.59
$QCQP_{\theta,\lambda}$	0.011	0.337	0.310	0.017	0.235	0.048	0	0.001	0.003	0.039	2.4e-6	93.28±1.51
$SILP_\theta$	0	0.162	0.470	0.005	0.247	0.071	0	0	0.004	0.042	5.0e-4	93.48±1.60
$SILP_{\theta,\lambda}$	0.014	0.331	0.316	0.010	0.242	0.044	0	0.001	0.003	0.039	5.9e-9	93.33±1.52
$RKDA_{K,\lambda}$	0	0	0	3	2	7	7	6	4	1	–	93.15±1.73
$SVM_{K,C}$	0	0	0	1	11	13	5	0	0	0	–	93.48±2.08

Table 11: See the caption and footnotes of Table 6 for explanation.

satimage(6)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0.131	0.478	0.031	0.249	0.072	0	0	0	0	0.039	5.0e-4	87.65±1.85
$SDP_{\theta,\lambda}$	0.293	0.338	0.04	0.212	0.06	0	0	0	0	0.033	2.3e-2	86.69±1.97
$QCQP_\theta$	0.102	0.454	0.096	0.138	0.128	0.043	0.001	0.002	0.009	0.029	5.0e-4	87.96±1.78
$QCQP_{\theta,\lambda}$	0.282	0.295	0.114	0.107	0.111	0.035	0.001	0.002	0.008	0.024	2.0e-2	87.22±1.84
$SILP_\theta$	0.108	0.448	0.094	0.143	0.123	0.046	0	0.002	0.006	0.031	5.0e-4	87.97±1.74
$SILP_{\theta,\lambda}$	0.277	0.299	0.112	0.106	0.114	0.032	0.003	0.005	0.006	0.024	2.2e-2	87.26±1.81
$RKDA_{K,\lambda}$	0	0	0	5	7	7	2	5	4	0	–	87.71±1.55
$SVM_{K,C}$	0	0	0	1	16	10	3	0	0	0	–	88.50±2.11

Table 12: See the caption and footnotes of Table 6 for explanation.

segment(3)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0.329	0.040	0.061	0.349	0.094	0.003	0	0	0	0.125	5.0e-4	99.17±0.62
$SDP_{\theta,\lambda}$	0.215	0.083	0.053	0.314	0.089	0.003	0	0	0	0.114	1.3e-1	99.00±0.83
$QCQP_\theta$	0.314	0.046	0.075	0.257	0.127	0.087	0.002	0	0	0.091	5.0e-4	99.19±0.67
$QCQP_{\theta,\lambda}$	0.215	0.075	0.079	0.218	0.127	0.079	0	0	0	0.084	1.2e-1	99.03±0.76
$SILP_\theta$	0.312	0.049	0.073	0.263	0.118	0.093	0.003	0	0	0.090	5.0e-4	99.17±0.69
$SILP_{\theta,\lambda}$	0.210	0.083	0.071	0.222	0.128	0.078	0	0	0	0.085	1.2e-1	99.03±0.76
$RKDA_{K,\lambda}$	0	0	2	8	1	3	4	6	4	2	–	98.86±1.08
$SVM_{K,C}$	0	0	5	9	8	3	4	1	0	0	–	99.06±0.81

Table 13: See the caption and footnotes of Table 6 for explanation.

analysis in Section 3.4. In contrast, the proposed SILP formulations are more efficient than methods based on cross-validation on all of the ten data sets.

5.3 Gene Expression Pattern Image Classification

In this experiment, we demonstrate the effectiveness of the proposed multiple kernel learning (MKL) formulations for data (feature) integration. Gene expression pattern images of *Drosophila melanogaster* embryo at a given developmental stage (time) capture the spatial and temporal distribution of gene expression patterns (Tomancak et al., 2002). The identification of genes showing spatial overlaps in their expression patterns is fundamentally important to formulating and testing gene interaction hypotheses (Kumar et al., 2002; Peng and Myers, 2004). Estimation of pattern overlap is most biologically meaningful when images from a similar time point (developmental stage) are compared. Thus, one of the central issues in gene expression pattern image analysis is the classification of images into different developmental stage ranges (Ye et al., 2006).

segment(4)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0.376	0.149	0.074	0.019	0.363	0.005	0	0	0	0.014	5.0e-4	97.00±1.09
$SDP_{\theta,\lambda}$	0.379	0.104	0.073	0.018	0.324	0.005	0	0	0	0.012	8.5e-2	96.77±1.25
$QCQP_\theta$	0.368	0.117	0.114	0.031	0.306	0.035	0	0	0	0.030	5.0e-4	97.00±1.17
$QCQP_{\theta,\lambda}$	0.373	0.073	0.111	0.028	0.271	0.033	0	0	0	0.027	8.5e-2	96.81±1.28
$SILP_\theta$	0.372	0.110	0.117	0.028	0.310	0.033	0	0	0	0.030	5.0e-4	97.02±1.12
$SILP_{\theta,\lambda}$	0.369	0.075	0.112	0.031	0.267	0.033	0	0	0	0.027	8.7e-2	96.81±1.26
$RKDA_{K,\lambda}$	0	0	1	1	2	3	2	7	6	8	–	97.31±0.93
$SVM_{K,C}$	0	0	0	4	5	7	4	6	1	3	–	96.83±1.38

Table 14: See the caption and footnotes of Table 6 for explanation.

waveform(3)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	λ	TSA
SDP_θ	0.072	0.072	0.072	0.072	0.029	0	0.007	0.028	0.415	0.232	5.0e-4	83.03±2.68
$SDP_{\theta,\lambda}$	0.074	0.074	0.074	0.074	0.069	0	0.01	0.034	0.377	0.214	6.8e-7	83.22±2.61
$QCQP_\theta$	0.061	0.061	0.061	0.061	0.061	0.006	0.003	0.036	0.423	0.226	5.0e-4	83.03±2.68
$QCQP_{\theta,\lambda}$	0.071	0.071	0.071	0.071	0.071	0.006	0.006	0.041	0.385	0.209	6.2e-6	83.19±2.49
$SILP_\theta$	0.053	0.021	0.033	0.115	0.094	0	0	0.036	0.422	0.227	5.0e-4	83.08±2.74
$SILP_{\theta,\lambda}$	0.012	0.066	0.033	0.136	0.113	0	0.006	0.040	0.382	0.213	2.0e-7	83.22±2.50
$RKDA_{K,\lambda}$	0	0	0	0	0	0	0	0	6	24	–	84.17±3.14
$SVM_{K,C}$	0	0	0	0	0	0	0	0	12	18	–	81.86±2.99

Table 15: See the caption and footnotes of Table 6 for explanation.

Data	*USPS*			*wine*	*satimage*			*segment*		*waveform*
# of classes	3	6	8	3	3	5	6	3	4	3
SDP_θ	50.98	411.15	1021.16	18.91	95.95	415.58	753.07	74.38	163.26	64.09
$SDP_{\theta,\lambda}$	69.95	646.05	1642.17	27.02	130.51	710.70	1172.55	99.29	235.50	96.08
$QCQP_\theta$	5.19	81.24	276.27	1.15	4.23	36.56	79.61	4.16	14.89	4.09
$QCQP_{\theta,\lambda}$	5.96	88.05	286.49	1.29	4.67	39.09	82.93	4.50	16.20	4.65
$SILP_\theta$	0.32	1.52	3.03	0.30	0.59	1.97	3.65	0.62	1.03	0.24
$SILP_{\theta,\lambda}$	0.60	3.45	6.70	0.30	0.66	2.29	4.38	1.03	1.52	0.25
$RKDA_{K,\lambda}$	1.54	9.26	20.59	0.76	1.56	5.39	8.61	1.56	2.89	1.71
$SVM_{K,C}$	5.60	17.82	23.18	3.65	1.87	4.24	9.50	3.53	4.13	5.44

Table 16: Comparison of computation time (in seconds) of various methods. The reported time is averaged over 30 random partitions.

We collect 2705 gene expression pattern images in the first three stage ranges (1-3, 4-6, and 7-8) from the FlyExpress[13] database. The raw gene expression pattern images are of size 128×320. It has been observed (Gargesha et al., 2005) that across various developmental stages, a distinguishing feature is the image textural properties at sub-block level, because image texture at the sub-block level changes as embryonic development progresses. Gabor filters (Daugman, 1988) have been shown to be effective in detecting local texture features and are well suited for extracting textural features for gene expression pattern images.

We apply Log Gabor Filters to extract the texture features (Daugman, 1988). Gabor filters are the product of a complex sinusoidal function and a Gaussian-shaped function. We use Log Gabor filters with 4 different wavelet scales and 6 different filter orientations to extract the texture information. Hence, 24 Gabor images were obtained from the filtering operation. Note that all 24 Gabor images have the same size (i.e., 128×320) as the original one. Figure 2 plots the 24 Gabor im-

13. The URL is `http://www.flyexpress.net`.

ages extracted from a sample image. These images contain different but potentially complementary information for stage classification. Two RBF kernels are built from each of the 24 Gabor images with σ values assigned as 50 and 100, respectively. We thus obtain a total of 48 kernel matrices. To exploit the complementary information in kernels constructed from different Gabor images, we apply the proposed SILP formulation to learn a linear combination of the 48 kernel matrices.

The 2705 images are randomly partitioned into training and test sets using the ratio 1:9. Our experimental results show that SILP_θ achieves a classification accuracy of about 88.28%. To see how each of the 48 kernel matrices works when used individually, we fix the kernel matrix and tune the λ value using cross-validation. The maximum, minimum, and average accuracies achieved across the 48 kernel matrices are 72.03%, 54.37%, and 61.88%, respectively. We also assign a uniform weight of 1 to each of the 48 kernel matrices and the combined kernel matrix achieves an accuracy of about 72.65%. These results demonstrate that different Gabor images contain complementary information, which is critical for stage classification, and the proposed MKL formulations are effective in exploiting this information by combining different kernel matrices.

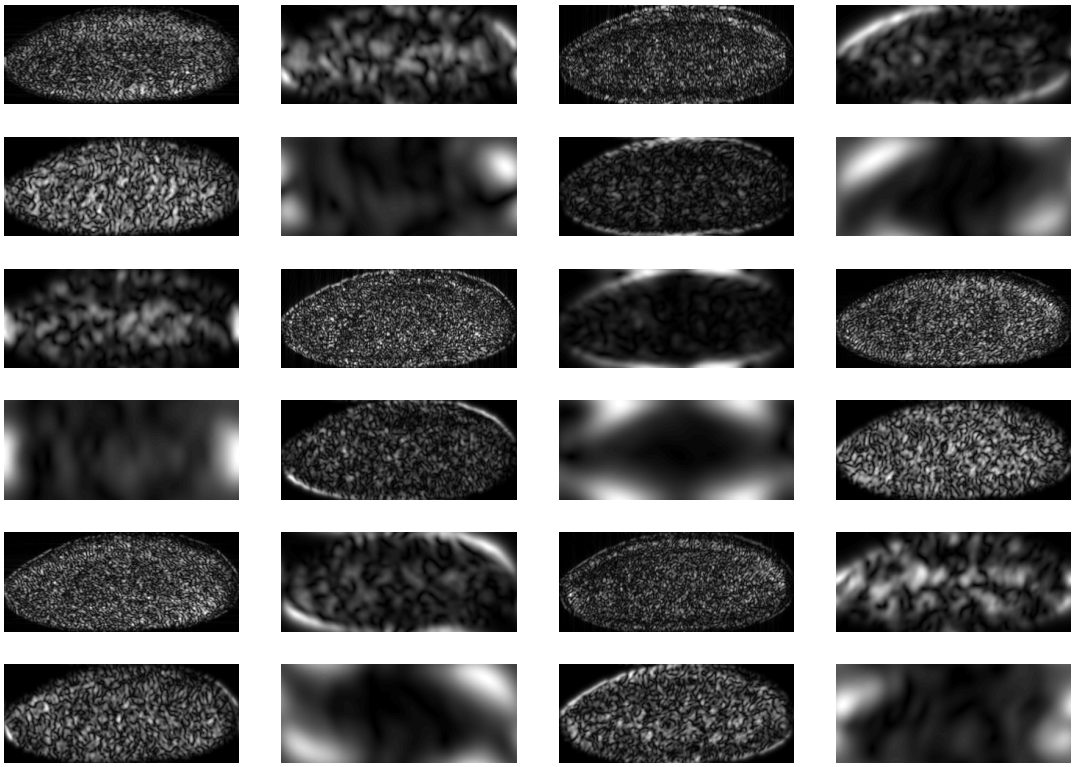

Figure 2: The 24 Gabor images extracted from a single sample image with 4 different wavelet scales and 6 different filter orientations.

		SM1			SM2			$SM2_C$		
data set	training size	C	SVs	PCT	C	SVs	PCT	C	SVs	PCT
sonar	167	1	155.4	93.05	1	152.2	91.12	2.43e7	156.8	93.89
heart	216	1	208.7	96.62	1	195.9	90.70	6.01e6	208.3	96.44
ionosphere	281	1	203.6	72.46	1	184.9	65.80	3.70e6	206.3	73.42
cancer	546	1	210.3	38.52	1	138.8	25.42	2.61e6	212.5	38.92

Table 17: The numbers of support vectors ("SVs") obtained from the 1-norm soft margin SVM, 2-norm soft margin SVM without and with C learned jointly that were proposed in Lanckriet et al. (2004b). These numbers are averaged over 30 random partitions. The total number of data points in the training set and the C values are also shown. The columns with title "PCT" show the percentage of support vectors over the training set.

5.4 SVM versus RKDA

It was shown (Shashua, 1999) that hard margin linear SVM is equivalent to linear discriminant analysis (LDA) when all the training points are support vectors. Through experiments, we found that the C values chosen by the 2-norm soft margin SVM proposed in Lanckriet et al. (2004b) are very large. Under such circumstances, soft margin SVM is approaching hard margin SVM. It has already been observed that SVM and kernel discriminant analysis usually have similar performance (Mika, 2002) and this has been confirmed by our experiments in the last two subsections. Thus it is interesting to report the number of support vectors for SVM. We record the average number of support vectors for 1-norm soft margin SVM, 2-norm soft margin SVM without and with C optimized jointly over the 30 random partitions reported in Section 5.1. As proposed in Lanckriet et al. (2004b), C is fixed to 1 for 1-norm and 2-norm soft margin SVM without C optimized. Table 17 reports the average C values obtained by the joint optimization 2-norm soft margin SVM and the average number of support vectors. For ease of comparison, we also report the size of training set and the averaged percentage of support vectors over 30 randomizations. It can be seen that for three out of four data sets, the percentages of support vectors are very high. This implies that SVM is similar to RKDA and explains why they have similar performance, as reported in the last two subsections.

5.5 The Effect of Regularization Parameter

In order to investigate the effect of regularization parameter in RKDA, we sampled 30 λ values between 10^{-10} and 10^2 uniformly over logarithmic scale and the accuracies of SDP_θ and QCQP_θ are plotted for two binary-class data sets (Figure 3) and two multi-class data sets (Figure 4). The results for SILP formulations are omitted since their performance is similar to their QCQP counterparts. It can be observed that as λ value changes, the accuracies oscillate in all cases. It can also be observed from the four figures that QCQP_θ tends to be less sensitive to the change of λ value than SDP_θ. This may be attributable to the fact that SDP is more computationally intensive and numerical problems may cause the poor performance. Indeed, we observed several reports of numerical problems from SeDuMi while conducting SDP experiments. The low accuracies of SDP_θ for some choices of λ in Figures 3 and 4 were caused by numerical problems and should be interpreted with caution.

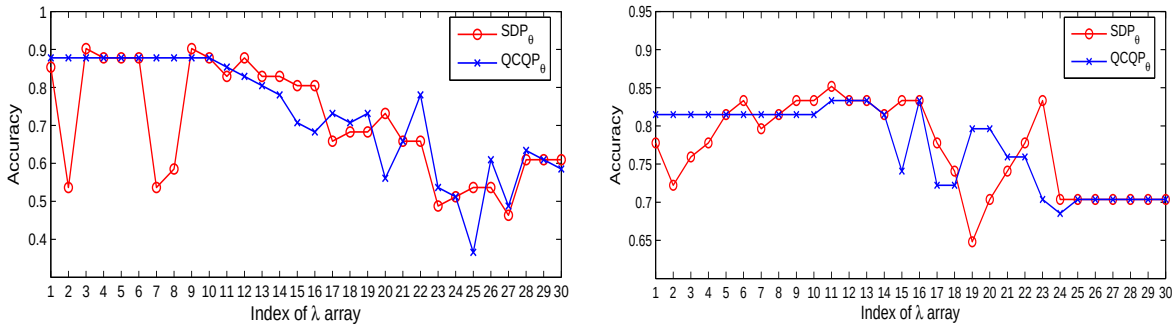

Figure 3: The change of accuracies for SDP_θ and QCQP_θ when λ varies from 10^{-10} to 10^2 for the *sonar* (left) and *heart* (right) data. The horizontal axis represents the indexes of the 30 λ values.

6. Discussion and Conclusion

We address the issue of learning appropriate kernels for RKDA in this paper. This problem is formulated as convex programs and thus globally optimal solutions are guaranteed. Practically, some convex optimization problems are computationally expensive and we propose approaches that are scalable and efficient to solve. While most existing work on kernel learning only deal with binary-class problems, we show that our binary-class formulations can be extended naturally to multi-class setting. Furthermore, we consider the problem of optimizing the kernel and regularization parameter in a joint framework, thus approaching the desirable goal of automated learning.

We have conducted extensive experiments to evaluate the proposed algorithms. When combining kernels from a single source of data, the proposed formulations have similar performance with approaches based on double cross-validation. When the candidate kernels contain complementary information, we show that the proposed formulations are effective to exploit such information. In terms of computation time, the SILP formulations are more efficient than approaches based on cross-validation. When evaluating the relative importance of each kernel (either used separately or in linear combination), we found that the best individual kernel sometimes coincides with the highly-weighted kernels in linear combination and sometimes disagrees considerably.

There are some directions for future work. Our experimental results have shown that the proposed approximate SDP formulation works well in most cases while it has a much lower computational cost in comparison with the exact formulation. We plan to compare the approximate formulation to the exact one in terms of complexity and performance. The derivation of multi-class formulations is based on an alternative criterion defined in Equation (23). This results in the same optimal transformation matrix as the original criterion in Equation (26) when a common (fixed) kernel matrix is used. However, they may differ when the kernel matrix is also optimized. We plan to investigate their differences further in the future. Most existing formulations for learning SVM kernels are restricted to the binary-class case. The idea from this paper may be useful for kernel learning in multi-class SVM. A more general problem is learning kernels for multi-label data in which each data point can be assigned to multiple classes. Such data are common in automatic image annotation problems (Lavrenko et al., 2004). We plan to explore these in the future.

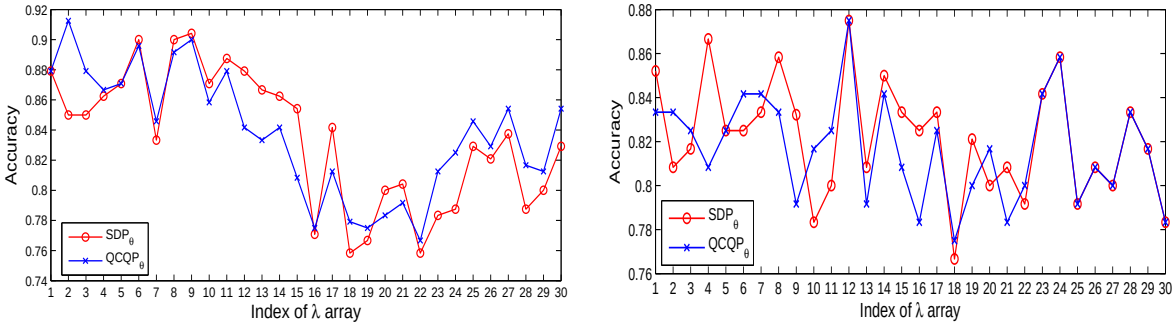

Figure 4: The change of accuracies for SDP_θ and QCQP_θ when λ varies from 10^{-10} to 10^2 for the *satimage*(6) (left) and *waveform*(3) (right) data. The horizontal axis represents the indexes of the 30 λ values.

Acknowledgments

The extensive experimental study in this paper was made possible with help from several researchers. Special thanks to Dr. Seung Jean Kim for helping in replicating their experiments. Dr. Johan Löfberg answered many questions about SeDuMi and his YALMIP through the online forum and E-mails. Dr. Anton Schwaighofer helped the authors to modify his SVM toolbox. The implementation of the proposed QCQP formulations is based on code for SVM kernel learning provided by Dr. Gert Lanckriet. This research is sponsored in part by funds from the Arizona State University and the National Science Foundation under Grant No. IIS-0612069.

Appendix A.

One of the basic tools used in our proof is the Sherman-Woodbury-Morrison formula (Golub and Van Loan, 1996): Let $S \in \mathbb{R}^{d\times d}$, and $Q, R \in \mathbb{R}^{d\times n}$. Assuming that both the matrices S and $(I + R^T S^{-1} Q)$ are nonsingular, we have

$$(S+QR^T)^{-1} = S^{-1} - S^{-1}Q(I+R^T S^{-1}Q)^{-1}R^T S^{-1}.$$

Since $P = PP$ and $P = P^T$, where P is the centering matrix defined in Equation (5), it follows that

$$\begin{aligned}
w^* &= (\Sigma_K + \lambda I)^{-1}(\mu_K^+ - \mu_K^-) \\
&= \left(\phi_K(X) P \phi_K(X)^T + \lambda I\right)^{-1} \phi_K(X) a \\
&= \left(\phi_K(X) PP \phi_K(X)^T + \lambda I\right)^{-1} \phi_K(X) a \\
&= \left((\phi_K(X)P)(\phi_K(X)P)^T + \lambda I\right)^{-1} \phi_K(X) a \\
&= \left(\frac{1}{\lambda} I - \frac{1}{\lambda^2}\phi_K(X) P \left(I + \frac{1}{\lambda} P\phi_K(X)^T \phi_K(X) P\right)^{-1} P \phi_K(X)^T\right) \phi_K(X) a \\
&= \frac{1}{\lambda}\phi_K(X)\left(I - P\left(\lambda I + PGP\right)^{-1} PG\right) a.
\end{aligned}$$

References

F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95: 3–51, 2003.

E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In T. Terlaky H. Frenk, K. Roos and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

A. Argyriou, R. Hauser, C. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 41–48, 2006.

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

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

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

L. Breiman, J. Friedman, C. J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

N. Cristianini and J.S. Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2000.

A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

J.G. Daugman. Complete discrete 2-D Gabor transform by neural networks for image analysis and compression. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

T. De Bie, G.R.G. Lanckriet, and N. Cristianini. Convex tuning of the soft margin parameter. Technical Report UCB/CSD-03-1289, EECS Department, University of California, Berkeley, 2003.

G. Fung, M. Dundar, J. Bi, and B. Rao. A fast iterative algorithm for Fisher discriminant using heterogeneous kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.

M. Gargesha, J. Yang, B. Van Emden, S. Panchanathan, and S. Kumar. Automatic annotation techniques for gene expression images of the fruit fly embryo. In S. Li, F. Pereira, H.-Y. Shum, and A. G. Tescher, editors, *Visual Communications and Image Processing 2005.*, pages 576–583, July 2005.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

S. C. H. Hoi, R. Jin, and M. R. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 361–368, 2007.

J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Analysis Machine Intelligence*, 16(5):550–554, 1994.

T. Jebara. Multi-task feature and kernel selection for SVMs. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.

S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel Fisher discriminant analysis. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 465–472, 2006.

S. Kumar, K. Jayaraman, S. Panchanathan, R. Gurunathan, A. Marti-Subirana, and S. J. Newfeld. BEST: a novel computational approach for comparing gene expression patterns from early stages of *Drosophila melanogaster* develeopment. *Genetics*, 169:2037–2047, 2002.

G.R.G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2003.

G.R.G. Lanckriet, T. De Bie, N. Cristianini, M.I. Jordan, and W.S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004a.

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

V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems 16*. 2004.

D. Lewis, T. Jebara, and W. S. Noble. Nonstationary kernel combination. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 553–560, 2006.

M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

C. A. Micchelli and M. Pontil. Feature space perspectives for learning the kernel. *Machine Learning*, 66(2-3):297–319, 2007.

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

S. Mika. *Kernel Fisher Discriminants*. PhD thesis, University of Technology, Berlin, Oct. 2002.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel Fisher algorithm. In *Advances in Neural Information Processing Systems 13*, pages 591–597, 2001.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(5):623–633, 2003.

Y. Nesterov and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics. SIAM, 1994.

D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

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

H. Peng and E. W. Myers. Comparing *in situ* mRNA expression patterns of *Drosophila* embryos. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology*, pages 157–166, 2004.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 775–782, 2007.

S. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines,Regularization, Optimization and Beyond*. MIT Press, 2002.

A. Shashua. On the relationship between the support vector machine for classification and sparsified Fisher's linear discriminant. *Neural Processing Letter*, 9(2):129–139, 1999.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

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

P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E Celniker, and G. M Rubin. Systematic determination of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 3(12), 2002.

I. W. Tsang and J. T. Kwok. Efficient hyperkernel learning using second-order cone programming. *IEEE Trans. on Neural Networks*, 17(1):48–58, 2006.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

J. Ye. Least squares linear discriminant analysis. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 1087–1093, 2007.

J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

J. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research*, 7:1183–1204, 2006.

J. Ye, J. Chen, Q. Li, and S. Kumar. Classification of *Drosophila* embryonic developmental stage range based on gene expression pattern images. In *Proceedings of the Computational Systems Bioinformatics Conference*, pages 293–298, 2006.

A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 1191–1198, 2007.