Noname manuscript No.
(will be inserted by the editor)

Learning the Coordinate Gradients

Yiming Ying · Qiang Wu · Colin Campbell

Received: January 2009 / Accepted: July 2010

Abstract In this paper we study the problem of learning the gradient function with application to variable selection and determining variable covariation. Firstly, we propose a novel unifying framework for coordinate gradient learning from the perspective of multi-task learning. Various variable selection methods can be regarded as special instances of this framework. Secondly, we formulate the dual problems of gradient learning with general loss functions. This enables the direct application of standard optimization toolboxes to the case of gradient learning. For instance, gradient learning with SVM loss can be solved by quadratic programming (QP) routines. Thirdly, we propose a novel gradient learning formulation which can be cast as a learning the kernel matrix problem. Its relation with sparse regularization is highlighted. A semi-infinite linear programming (SILP) approach and an iterative optimization approach are proposed to efficiently solve this problem. Finally, we validate our proposed approaches on both synthetic and real datasets.

Keywords Learning the Gradient · Multi-task Kernel · Feature Selection · Sparse Regularization · Learning the Kernel Matrix

Mathematics Subject Classification (2000) 68T05 · 68Q32 · 68T10 · 62P10

1 Introduction

Kernel methods [31,32], such as Support Vector Machines (SVMs), have been successfully demonstrated in many supervised learning tasks. In this case, we are interested in learning an appropriate function for regression or classification. However, in many

Y. Ying: College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, Exeter, EX4 4EE, UK
Tel.: +44(0)1392 723591 Fax: +44 (0)1392 217965
E-mail: Y.Ying@exeter.ac.uk
Q. Wu: Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA
E-mail: wuqiang@math.msu.edu
C. Campbell: Department of Engineering Mathematics, University of Bristol, Queens's Building, Bristol, BS8 1TR, UK
E-mail: C.Campbell@bris.ac.uk

applications we are not only interested in learning a target function, but also wish to learn salient coordinate variables and their interaction with each other. This problem has strong practical motivations: to facilitate data visualization and dimensionality reduction, for example. Such a motivation is important when there are many redundant variables and we wish to find the salient features among these. These problems can occur in many contexts. For example, with gene expression array data sets, the vast majority of features may be redundant to a classification task and we need to find a small subset of genuinely distinguishing features (genes). These motivations have driven the design of various statistical and machine learning models [8,15,35,36] for variable (feature) selection.

In this paper we are concerned with learning the gradient function and its applications to variable selection. Specifically, let $X \subseteq \mathbb{R}^d$ be compact, $Y \subseteq \mathbb{R}$, $Z = X \times Y$ and $\mathbb{N}_n = \{1, 2 \ldots, n\}$ for any $n \in \mathbb{N}$. For any $x \in X$, we denote x by $(x^1, x^2, \ldots, x^d)$. The relation between the unknown target function f_* and the input space is described by an observed set of input/outputs $\mathbf{z} = \{(x_i, y_i) : i \in \mathbb{N}_m\}$. Our aim is to simulate the gradient of a target function f_* (if it exists) denoted by $\nabla f_*(x) = \big(\frac{\partial f_*}{\partial x^1}, \ldots, \frac{\partial f_*}{\partial x^d}\big)$ and apply it to variable selection. The intuition behind gradient learning for variable selection [29,27,28] is as follows. The specific *norm* of $\frac{\partial f_*}{\partial x^p}$ can indicate the salience of the p-th variable: the smaller the norm is, the less important this variable will be.

In this paper we address the problem of learning the gradient and its applications to variable selection. Our main contributions are summarized as follows.

- We propose a unified framework for gradient learning which allows the more flexible recovery of structures underlying the problem of gradient learning [1]. The main idea is to formulate gradient learning using matrix-valued kernels for multi-task learning [23,24,6].
- We establish a dual formulation for this unified framework with general loss functions. This enables the efficient implementation of gradient learning with SVM hinge loss. In contrast, the implementation of gradient learning in [29,27] is based on Newton's method. The loss function is required to be twice differentiable and hence excludes the SVM hinge loss.
- A novel gradient learning algorithm is proposed from the perspective of learning the kernel matrix problem [19] and sparse nonparametric group lasso [3]. A semi-infinite linear programming (SILP) approach and an alternative approach of iterative optimization are introduced to efficiently solve this problem.

The above contributions rely on very recent developments from machine learning: multi-task learning [23,24,6] and multiple kernel learning with sparse regularization (aka nonparametric group lasso) [3,19,34]. These new insights introduced for gradient learning are important for the following reasons. Firstly, the multi-task formulation for gradient learning allows a deeper understanding of current gradient learning algorithms. Secondly, the novel gradient learning from the perspective of learning the kernel problem enables a natural sparse regularization and allows an efficient implementation. Note this is less straightforward for the gradient learning in [29,27]. Thirdly, we will show that, due to the sparsity, our new gradient learning algorithms are more effective to remove redundant features and can achieve state-of-the-art feature selection results on real cancer datasets.

[1] Part of this multi-task formulation first appeared as a conference paper [38]. This journal version includes more illustrative examples.

The organization of this paper is as follows. In Section 2, we briefly review the theory of multi-task kernels and introduce a unifying approach for gradient learning. In particular, we provide several concrete examples that can be derived from this framework and highlight their relationships with existing methods. Section 3 investigates the dual problem of gradient learning algorithms with general loss functions. In Section 4, we propose a novel gradient learning algorithm which can be cast as the problem of learning the kernel matrix. We validate our proposed approaches on both synthetic and real data sets in Section 5. In particular, for the Leukemia cancer data set [14] our proposed method achieves zero test error with only 3 genes which is more promising than the solution using 8 genes found by recursive feature elimination with a SVM [15].

2 General Framework for Gradient Learning

As described above, we are concerned with learning the target function f_* and its gradient ∇f_*. This naturally can be regarded as multi-task learning: learning a vector-valued function $\mathbf{F}_* := (f_*, \nabla f_*)$. With this motivation, this section briefly describes a general formulation of the gradient learning problem using ideas from multi-task learning. Throughout this paper, we use the notation $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ to denote the standard Euclidean inner product and norm respectively. A description of notations is summarized in Table 1.

2.1 General Learning Model

We depart from the definition of multi-task kernels, see [23, 24] and the reference therein.

Definition 1 We say that a function $\mathcal{K} : X \times X \to \mathbb{R}^{(d+1)\times(d+1)}$ is a multi-task (matrix-valued) kernel on X if, for any $x, t \in X$, $\mathcal{K}(x,t)^T = \mathcal{K}(t,x)$, and it is *positive semi-definite*, i.e., for any $m \in \mathbb{N}$, $\{x_j \in X : j \in \mathbb{N}_m\}$ and $\{\mathbf{c}_j \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$ there holds

$$\sum_{i,j \in \mathbb{N}_m} \langle \mathbf{c}_i, \mathcal{K}(x_i, x_j)\mathbf{c}_j \rangle \geq 0. \tag{1}$$

In the spirit of Moore-Aronszjain's theorem, there exists a one-to-one correspondence between the multi-task kernel $\mathcal{K}$ with property (1) and a vector-valued reproducing kernel Hilbert space (RKHS) of functions $\mathbf{F} : X \to \mathbb{R}^{d+1}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ denoted by $\mathscr{H}_{\mathcal{K}}$, see e.g. [23, 24, 6]. Moreover, for any $x \in X$, $\mathbf{c} \in \mathbb{R}^{d+1}$ and $\mathbf{F} \in \mathscr{H}_{\mathcal{K}}$, we have the reproducing property

$$\langle \mathbf{F}(x), \mathbf{c} \rangle = \langle \mathbf{F}, \mathcal{K}_x \mathbf{c} \rangle_{\mathcal{K}} \tag{2}$$

where $\mathcal{K}_x \mathbf{c} : X \to \mathbb{R}^{d+1}$ is defined, for any $t \in X$, by $\mathcal{K}_x \mathbf{c}(t) := \mathcal{K}(t,x)\mathbf{c}$.
In the following we describe our unified framework for gradient learning. To this end, we adopt the notations $\mathbf{f} = (f_1, \ldots, f_d)$, $\mathbf{F} = (f_0, \mathbf{f})$ and $x_{ij} = x_i - x_j$. The derivation of gradient learning algorithms is motivated by the Taylor expansion[2] of the target

[2] Our form of Taylor expansion is slightly different from that used in [27, 29]. However, the essential idea is the same.

Table 1 Notations

name	notation	meaning
input space	X	subset of Euclidean space $\mathbb{R}^d$
	x, t	elements of X
	x^ℓ	ℓ-th coordinate of x
	x_{ij}	$x_i - x_j$
	$N_k(i)$	k-nearest neighbor of x_i
	$\widetilde{x}$	extended vector of x: $(0, x^\top)^\top$
	$\widetilde{x}_{ij}$	$\widetilde{x}_i - \widetilde{x}_j$
	$\widetilde{e}_1$	first coordinate basis in $\mathbb{R}^{d+1}$
output space	Y	binary valued space $\{\pm 1\}$
	Z	product space: $X \times Y$
	m	number of training samples
	$\mathbf{z}$	input/outputs data $\{(x_i, y_i)\}_{i=1}^m$
	f_*	target function
	∇f_*	d-dim gradient of target function
	$\mathbf{F}_*$	vector-valued function $(f_*, \nabla f_*)$
scalar kernel	G	real valued reproducing kernel
	$\mathscr{H}_G$	scalar RKHS space with kernel G
	$\langle \cdot, \cdot \rangle_G$	inner product in $\mathscr{H}_G$
multi-task kernel	$\mathcal{K}$	matrix-valued kernel, see Def. 1
	$\mathscr{H}_{\mathcal{K}}$	RKHS with multi-task kernel $\mathcal{K}$
	$\langle \cdot, \cdot \rangle$	standard Euclidean inner product
	$\langle \cdot, \cdot \rangle_{\mathcal{K}}$	inner product in $\mathscr{H}_{\mathcal{K}}$
vector function	$\mathbf{F} = (f_0, f_1, \ldots, f_d)$	$(d+1)$-dim function in $\mathscr{H}_{\mathcal{K}}$
	$\mathbf{f} = (f_1, \ldots, f_d)$	d-dim vector-valued function
	$\mathbf{F_z} = (f_{0\mathbf{z}}, f_{1\mathbf{z}}, \ldots, f_{d\mathbf{z}})$	$(d+1)$-dim optimization solution
	$\mathbf{f_z} = (f_{1\mathbf{z}}, \ldots, f_{d\mathbf{z}})$	components of $\mathbf{F_z}$

function f_*: $f_*(x_i) \approx f_*(x_j) + \nabla f_*(x_j) x_{ij}$. Then, a function f_0 is used to learn f_*, and $\mathbf{f}$ to simulate ∇f_*. Replacing $f_*(x_i)$ by y_i, the error

$$y_i \approx f_0(x_j) + \mathbf{f}(x_j) x_{ij}$$

is expected to be small whenever x_i is close to x_j. To enforce the constraint that x_i is close to x_j, we could use a weight function produced by a Gaussian with deviation s defined by $w_{ij} = s^{-(d+2)} e^{-\|x_i - x_j\|^2/2s^2}$. Another alternative is to use the k-nearest neighborhood (kNN), e.g.

$$w_{ij} = \begin{cases} \frac{1}{mk} & \text{if } x_j \in N_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $N_k(i)$ denotes the kNN of x_i. This implies that $w_{ij} \approx 0$ if x_i is far away from x_j. We now propose the following formulation for gradient learning:

$$\mathbf{F_z} = \arg \min_{\mathbf{F} \in \mathscr{H}_{\mathcal{K}}} \Big\{ C \sum_{i,j \in \mathbb{N}_m} w_{ij} L\Big(y_i, f_0(x_j) + \mathbf{f}(x_j) x_{ij}\Big) + \frac{1}{2} \|\mathbf{F}\|_{\mathcal{K}}^2 \Big\}. \tag{4}$$

where $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is a prescribed loss function and C is a positive parameter. The minimum is taken over a vector-valued RKHS with multi-task kernel $\mathcal{K}$. The first component $f_{0\mathbf{z}}$ of the minimizer $\mathbf{F_z} = (f_{0\mathbf{z}}, \mathbf{f_z})$ of the above algorithm is used to simulate the target function and $\mathbf{f_z} := (f_{1\mathbf{z}}, \ldots, f_{d\mathbf{z}})$ to learn the true gradient.

We further introduce an equivalent formulation of (4) which will be convenient for the derivation of its dual problem and the design of optimization approaches in subsequent sections. To this end, let $\widetilde{e}_p$ be the p-th coordinate basis in $\mathbb{R}^{d+1}$. For any $x \in \mathbb{R}^d$, denote the vector $\widetilde{x}^\top$ by $(0, x^\top)$ and, for any $i, j \in \mathbb{N}_m$, $\widetilde{x}_{ij} = \widetilde{x}_i - \widetilde{x}_j$. By the reproducing property (2), we have that $f_0(x_j) = \langle \mathbf{F}(x_j), \widetilde{e}_1 \rangle = \langle \mathbf{F}, \mathcal{K}_{x_j} \widetilde{e}_1 \rangle_{\mathcal{K}}$ and likewise, $\langle \mathbf{f}(x_j), x_{ij} \rangle = \langle \mathbf{F}(x_j), \widetilde{x}_{ij} \rangle = \langle \mathbf{F}, \mathcal{K}_{x_j} \widetilde{x}_{ij} \rangle_{\mathcal{K}}$. Then, formulation (4) can be rewritten as

$$\min_{\mathbf{F} \in \mathscr{H}_{\mathcal{K}}} \Big\{ C \sum_{i,j \in \mathbb{N}_m} w_{ij} L\Big(y_i, \langle \mathbf{F}, \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij}) \rangle_{\mathcal{K}} \Big) + \frac{1}{2} \|\mathbf{F}\|_{\mathcal{K}}^2 \Big\}. \tag{5}$$

In the next subsections we will start with its representer theorem, and then unify different feature selection approaches [15, 27, 29] in the framework of multi-task learning.

2.2 Representer Theorem

In analogy with standard kernel methods [31,32], we have the following representer theorem for the minimization problem (5) by using the properties of multi-task kernels.

Theorem 1 *For any multi-task kernel $\mathcal{K}$, consider the gradient learning formulation (5). Then, there exists representer coefficients $\{c_{j,\mathbf{z}} \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$ such that*

$$\mathbf{F}_{\mathbf{z}} = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,\mathbf{z}}$$

and, for every $j \in \mathbb{N}_m$, the representer coefficient $c_{j,\mathbf{z}} \in span\{\widetilde{e}_1, \widetilde{x}_i : i \in \mathbb{N}_m\}$.

Proof We can write any minimizer $\mathbf{F}_{\mathbf{z}} \in \mathscr{H}_{\mathcal{K}}$ as $\mathbf{F}_{\mathbf{z}} = \mathbf{F}_{\|} + \mathbf{F}_{\perp}$ where $\mathbf{F}_{\|}$ is in the linear span of $\big\{\mathcal{K}_{x_j} \widetilde{e}_1, \mathcal{K}_{x_j} \widetilde{x}_i, i, j \in \mathbb{N}_m\big\}$ and $\mathbf{F}_{\perp}$ is perpendicular to this span space. By the reproducing property (2), we have that $\langle \mathbf{F}(x_j), \widetilde{e}_1 + \widetilde{x}_{ij} \rangle = \langle \mathbf{F}, \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij}) \rangle_{\mathcal{K}} = \langle \mathbf{F}_{\|}, \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij}) \rangle_{\mathcal{K}}$. Hence, $\mathbf{F}_{\perp}$ makes no contribution to the loss function in the formulation (5). However, the norm $\|\mathbf{F}\|_{\mathcal{K}}^2 = \|\mathbf{F}_{\|}\|_{\mathcal{K}}^2 + \|\mathbf{F}_{\perp}\|_{\mathcal{K}}^2 > \|\mathbf{F}_{\|}\|_{\mathcal{K}}^2$ unless $\mathbf{F}_{\perp} = 0$. This implies that any solution $\mathbf{F}_{\mathbf{z}}$ belongs to the span space $\big\{\mathcal{K}_{x_j} \widetilde{e}_1, \mathcal{K}_{x_j} \widetilde{x}_i, i, j \in \mathbb{N}_m\big\}$ and the corresponding representer coefficients belong to the span of $\{\widetilde{e}_1, \widetilde{x}_i : i \in \mathbb{N}_m\}$. □

The representer theorem indicates that the optimal solution $\mathbf{F}_{\mathbf{z}}$ of problem (5) lives in the finite span of training samples which paves the way for designing efficient optimization algorithms for gradient learning.

2.3 Examples and Related Work

As listed below, we can get different gradient learning algorithms by choosing different multi-task kernel $\mathcal{K}$. We note that if $\mathcal{K}$ is a diagonal matrix-valued kernel, then each component of a vector-valued function in the associated RKHS of $\mathcal{K}$ can be represented, independently of the other components, as a function in the RKHS of a scalar kernel. Consequently, in our first example we choose, for a prescribed scalar kernel G, the multi-task kernel $\mathcal{K}$ defined, for any $x, t \in X$, by $\mathcal{K}(x, t) = G(x, t) I_{d+1}$. Then, for any function $\mathbf{F} \in \mathscr{H}_{\mathcal{K}}$, it is equivalent to the case that $f_p \in \mathscr{H}_G$ for any $p \in \mathbb{N}_{d+1}$. Hence we can use the ranking criterion $\{\|f_{p\mathbf{z}}\|_G : p \in \mathbb{N}_d\}$ for variable selection.

Example 1 Let G be a scalar kernel and $\mathcal{K}(x,t) = G(x,t)I_{d+1}$. Then, problem (5) is reduced to gradient learning methods proposed by [27,29,28].

Our next example is motivated by the Hessian of Gaussian kernel proposed in [6] and introduced for gradient learning in [38].

Example 2 For any smooth RBF kernel G, i.e. $G(x,t) = \phi(\|x-t\|^2)$ for some smooth function $\phi : (0,\infty) \to \mathbb{R}$, define the multi-task kernel $\mathcal{K}$ by

$$\mathcal{K}(x,t) = \begin{pmatrix} G(x,t), & (\nabla_t G(x,t))^T \\ \nabla_x G(x,t) & \nabla^2_{xt} G(x,t) \end{pmatrix}. \tag{6}$$

Then, problem (5) is reduced to the following formulation

$$f_{\mathbf{z}} = \arg\min_{f\in\mathscr{H}_G} \Big\{ C \sum_{i,j\in\mathbb{N}_m} w_{ij} L(y_i, f(x_j) + \nabla f(x_j)^\top x_{ij}) + \frac{1}{2}\|f\|^2_G \Big\} \tag{7}$$

Moreover, $\mathbf{F_z} = (f_{\mathbf{z}},\ \nabla f_{\mathbf{z}})$ where, for any $x \in X$, $f_{\mathbf{z}}(x) = \sum_{j\in\mathbb{N}_m}(c_{j0}G(x_j,x) + \sum_{\ell\in\mathbb{N}_d} c_{j\ell}\frac{\partial G(x_j,x)}{\partial x^\ell})$.

Proof From the representer theorem above, we see that every solution of problem (5) can be written as

$$\mathbf{F_z} = (f_{0\mathbf{z}}, f_{1\mathbf{z}}, \ldots, f_{d\mathbf{z}}) = \sum_{j\in\mathbb{N}_m} \mathcal{K}_{x_j}\mathbf{c}_j.$$

With $\mathcal{K}$ defined by equation (6), we know from [38, Proof of Proposition 1], for any smooth Mercer kernel G, that

$$f_{p\mathbf{z}} = \frac{\partial f_{0\mathbf{z}}}{\partial x^p} \in \mathscr{H}_G$$

and

$$\langle \frac{\partial G(\cdot,x_i)}{\partial x_i^p}, G(\cdot,x_j)\rangle_G = \frac{\partial G(x_i,x_j)}{\partial x_i^p},\ \langle \frac{\partial G(\cdot,x_j)}{\partial x_j^p}, \frac{\partial G(\cdot,x_i)}{\partial x_i^q}\rangle_G = \frac{\partial^2 G(x_i,x_j)}{\partial x_i^p \partial x_j^q}.$$

Combining this observation with property (2) we have that

$$\|\mathbf{F_z}\|^2_{\mathcal{K}} = \sum_{i,j\in\mathbb{N}_m} \mathbf{c}_i^\top \mathcal{K}(x_i,x_j)\mathbf{c}_j = \|f_{0\mathbf{z}}\|^2_G.$$

Letting $f_{0\mathbf{z}} = f_{\mathbf{z}}$ yields the desired assertion. □

Moreover, one can exactly compute the criterion $\big\{\|\frac{\partial f_{\mathbf{z}}}{\partial x^p}\|_G : p \in \mathbb{N}_d\big\}$ for the solution of problem (7) and use them for variable selection, see the appendix in [38].
Further specifying G as a linear kernel in the multi-task kernel $\mathcal{K}$ defined by (6), it is easy to see that problem (7) is reduced to a standard regularization scheme.

Example 3 Let $w_{ij} = \frac{1}{m}$ for any $i,j \in \mathbb{N}_m$ and

$$\mathcal{K}(x,t) = \begin{pmatrix} x^\top t & t^\top \\ x & I_d \end{pmatrix}. \tag{8}$$

Then, problem (5) is reduced to the following standard regularization scheme

$$w_{\mathbf{z}} = \arg\min_{w\in\mathbb{R}^d} \Big\{ C \sum_{i\in\mathbb{N}_m} L(y_i, w^\top x_i) + \frac{1}{2}\|w\|^2 \Big\}$$

with $\mathbf{F_z} = (w_{\mathbf{z}}^\top x, w_{\mathbf{z}})$.

Note the gradient in this example is exactly $w_{\mathbf{z}}$. If we choose the loss function as SVM hinge loss we get the SVM-RFE for feature selection [15].
We now turn our attention to a discussion of existing methods for feature selection. A number of statistical models have been proposed for variable (feature) selection using sparse regularization. Least absolute shrinkage and selection operator (LASSO) [35] and basis pursuit denoising [8] suggested use of ℓ^1 regularization to remove redundant features. Several authors [26, 7] proposed and studied feature selection methods for SVMs that incorporate scaling factors into the kernel: $K_A(x,y) = K(Ax, Ay)$ where A is a diagonal matrix of scaling factors $a_1, a_2, \ldots, a_n$. Training is performed by iteratively optimizing α for a fixed A (regular dual problem of SVM training) and optimizing A for fixed α by gradient descent. Another direction for variable selection is motivated by the margin bound criterion of SVM: variables with least influence on the margin are considered least important. Guyon et al. [15] proposed a recursive feature elimination (RFE) strategy which used a linear SVM. In the linear case, as shown in Example 3 gradient learning and SVM margin-based criterion for variable selection coincides with each other since both of them use the criterion $\{|w_p| : p \in \mathbb{N}_d\}$ for variable ranking. Weston et al. [36] also introduced a method for selecting features by minimizing generalization bounds.

3 Dual Formulation for Gradient Learning

The above unifying formulation for gradient learning enables us to directly derive the dual problem of gradient learning. This will facilitate optimization design of gradient learning for general loss functions such as SVM hinge loss. Similar dual problems are discussed for standard regularization schemes [41] and multi-kernel regularization algorithms [37].

Theorem 2 *Suppose that the loss function $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is convex with respect to the second argument. Let L^* be the adjoint function of L defined, for any $y \in Y$ and $\alpha \in \mathbb{R}$, by $L^*(y, \alpha) = \max_{s \in \mathbb{R}} \{s\alpha - L(y, s)\}$. Then the dual problem of (5) is given by*

$$\begin{aligned}\alpha_{\mathbf{z}} = \arg\max_{\alpha} -C \sum_{i,j \in \mathbb{N}_m} L^*\left(y_i, -\frac{\alpha_{ij}}{C}\right) w_{ij} - \frac{1}{2} \sum_{i,j,i',j' \in \mathbb{N}_m} & w_{ij}\alpha_{ij} w_{i'j'}\alpha_{i'j'} \\ \times & \left[(\widetilde{e}_1 + \widetilde{x}_{ij})^T \mathcal{K}(x_j, x_{j'})(\widetilde{e}_1 + \widetilde{x}_{i'j'})\right].\end{aligned} \tag{9}$$

and, the solution is given by

$$\mathbf{F}_{\mathbf{z}} = \sum_{i,j} \alpha_{ij,\mathbf{z}} w_{ij} \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij}). \tag{10}$$

Proof Since L is convex with respect to the second argument, for any $y \in Y, t \in \mathbb{R}$ there holds that $L(y,t) = \max_{\alpha \in \mathbb{R}} \{-t\alpha - L^*(y, -\alpha)\}$. Consequently, the objective function in problem (5) can be written as

$$\min_{\mathbf{F} \in \mathscr{H}_{\mathcal{K}}} C \sum_{i,j \in \mathbb{N}_m} w_{ij} \max_{\alpha_{ij}} \{-\alpha_{ij} \langle \mathbf{F}, \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij})\rangle_{\mathcal{K}} - L^*(y_i, -\alpha_{ij})\} + \frac{1}{2}\|\mathbf{F}\|^2_{\mathcal{K}}.$$

Using the min-max theorem (e.g. [41]), this is identical to the following

$$\max_{\alpha} \min_{\mathbf{F}\in\mathscr{H}_{\mathcal{K}}} \Big\{-\big\langle \mathbf{F}, C\sum_{ij} w_{ij}\alpha_{ij}\mathcal{K}_{x_j}(\widetilde{e}_1+\widetilde{x}_{ij})\big\rangle_{\mathcal{K}} + \frac{1}{2}\|\mathbf{F}\|^2_{\mathcal{K}}\Big\} \\ -C\sum_{ij} w_{ij} L^*(y_i, -\alpha_{ij}). \tag{11}$$

The minimum of the term within the first parenthesis is attained at

$$\mathbf{F} = C\sum_{i,j}\alpha_{ij}w_{ij}\mathcal{K}_{x_j}(\widetilde{e}_1+\widetilde{x}_{ij}),$$

and its minimum equals

$$-\frac{C^2}{2}\sum_{i,j,i',j'\in\mathbb{N}_m} w_{ij}\alpha_{ij}w_{i'j'}\alpha_{i'j'}\left[(\widetilde{e}_1+\widetilde{x}_{ij})^T\mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\right].$$

Replacing α by α/C, from equation (11) we get the desired result. □

By the above theorem, we can derive the dual problem of gradient learning by calculating the adjoint function L^* for a different loss function L. Here are some examples.

Least Square Loss: For the least square loss $L(y,t) = (y-t)^2$, $L^*(y,\alpha) = y\alpha + \frac{\alpha^2}{4}$. The dual problem becomes

$$\max_{\alpha}\sum_{i,j\in\mathbb{N}_m}\Big(y_i\alpha_{ij}-\frac{\alpha_{ij}^2}{4C}\Big)w_{ij} - \frac{1}{2}\sum_{i,j,i',j'\in\mathbb{N}_m}\alpha_{ij}w_{ij}w_{i'j'}\alpha_{i'j'} \\ \times\left[(\widetilde{e}_1+\widetilde{x}_{ij})^T\mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\right].$$

In binary classification, replacing α_{ij} by $y_i\alpha_{ij}$, the above equation is further reduced to

$$\max_{\alpha}\sum_{i,j\in\mathbb{N}_m} w_{ij}\Big(\alpha_{ij}-\frac{\alpha_{ij}^2}{4C}\Big) - \frac{1}{2}\sum_{i,j,i',j'\in\mathbb{N}_m} y_iw_{ij}\alpha_{ij}y_j\alpha_{i'j'}w_{i'j'}y_{i'}y_{j'} \\ \times\left[(\widetilde{e}_1+\widetilde{x}_{ij})^T\mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\right].$$

Soft Margin Loss: For the SVM hinge loss $L(y,t) = (1-yt)_+$,

$$L^*(y,\alpha) = \begin{cases} y\alpha, \ -1\le y\alpha\le 0 \\ \infty, \ \text{otherwise}\end{cases}$$

The dual formulation in Theorem 2 is reduced to the following:

$$\max_{\alpha}\sum_{i,j\in\mathbb{N}_m} y_iw_{ij}\alpha_{ij} - \frac{1}{2}\sum_{i,j,i',j'\in\mathbb{N}_m}\alpha_{ij}w_{ij}\alpha_{i'j'}w_{i'j'} \\ \times\left[(\widetilde{e}_1+\widetilde{x}_{ij})^\top\mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\right] \\ \text{s.t.}\quad 0\le\alpha_{ij}y_i\le C,\quad \forall i,j\in\mathbb{N}_m.$$

Replace α_{ij} by $y_i\alpha_{ij}$ yields the following dual problem of gradient learning with the SVM hinge loss:

$$\begin{aligned}\max_{\alpha} \sum_{i,j\in\mathbb{N}_m} w_{ij}\alpha_{ij} - \frac{1}{2} &\sum_{i,j,i',j'\in\mathbb{N}_m} w_{ij}\alpha_{ij}y_i\alpha_{i'j'}w_{i'j'}y_{i'}y_jy_{j'} \\ &\times \Big[(\widetilde{e}_1+\widetilde{x}_{ij})^\top \mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\Big] \\ \text{s.t.} \quad 0 \le \alpha_{ij} \le C,& \quad \forall i,j\in\mathbb{N}_m.\end{aligned} \tag{12}$$

2-Norm Soft Margin Loss: For the 2-norm SVM loss $L(y,t)=(1-yt)_+^2$,

$$L^*(y,\alpha)=\begin{cases} y\alpha+\frac{\alpha^2}{4},\ y\alpha<0 \\ \infty, \qquad\quad \text{otherwise}\end{cases}$$

The corresponding dual problem becomes

$$\begin{aligned}\min_{\alpha} \sum_{i,j\in\mathbb{N}_m} w_{ij}(\alpha_{ij}-\frac{\alpha_{ij}^2}{4C}) - \frac{1}{2} &\sum_{i,j,i',j'\in\mathbb{N}_m} w_{ij}\alpha_{ij}y_iy_jw_{i'j'}\alpha_{i'j'}y_{i'}y_{j'} \\ &\times \Big[(\widetilde{e}_1+\widetilde{x}_{ij})^T \mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})\Big] \\ \text{s.t.} \quad 0 \le \alpha_{ij},& \quad \forall i,j\in\mathbb{N}_m.\end{aligned}$$

In the above three examples, note that

$$\Big(w_{ij}(\widetilde{e}_1+\widetilde{x}_{ij})^T\mathcal{K}(x_j,x_{j'})(\widetilde{e}_1+\widetilde{x}_{i'j'})w_{i'j'}\Big)_{(i,j),(i',j')}$$

is a scalar kernel matrix with double indices (i,j) and (i',j'). The dual problem of SVM-like gradient learning can be solved by quadratic programming. Once we get the dual solution $\alpha_{\mathbf{z}}$ then the solution of gradient learning can be represented by

$$\mathbf{F}_{\mathbf{z}} = \sum_{i,j\in\mathbb{N}_m} y_iw_{ij}\alpha_{ij,\mathbf{z}}\mathcal{K}_{x_j}(\widetilde{e}_1+\widetilde{x}_{ij}).$$

It is also worth mentioning that we know from equation (9) that optimization is only needed to be implemented in the space $\{\alpha_{ij} : w_{ij}\neq 0, i,j\in\mathbb{N}_m\}$ which is useful to reduce computational complexity. We will come back to this point in Section 5.

4 Learning the Kernel Matrix for Gradient Learning

In this section we depart from the unified dual problem in the above section and formulate gradient learning as a well-established kernel matrix learning problem. This important observation enables us to directly apply the optimization approaches in [19, 34] to the scenario of gradient learning.

To this end, let G be a prescribed scalar kernel. We first introduce the notation

$$S_0(\alpha) = C\sum_{i,j\in\mathbb{N}_m} L^*(y_i,-\alpha_{ij}/C)w_{ij} + \frac{1}{2}\sum_{i,j,i',j'\in\mathbb{N}_m} w_{ij}\alpha_{ij}w_{i'j'}\alpha_{i'j'}G(x_j,x_{j'}). \tag{13}$$

Also, for any $\ell \in \mathbb{N}_d$, let $x_{ij}^\ell = x_i^\ell - x_j^\ell$ and define the kernel matrix $\mathbf{K}_\ell$ related to the ℓ-th coordinate by

$$\mathbf{K}_\ell = \Big(w_{ij} x_{ij}^\ell x_{i'j'}^\ell w_{i'j'} G(x_j, x_{j'}) \Big)_{ij,i'j'}. \tag{14}$$

In addition, for any matrix $\alpha = (\alpha_{ij})_{i,j=1}^m$ we denote by $\mathbf{vec}(\alpha)$ a column vector which consecutively stacks the column vectors of the matrix α. Let

$$S_\ell(\alpha) = \frac{1}{2} \mathbf{vec}(\alpha)^\top \mathbf{K}_\ell \mathbf{vec}(\alpha). \tag{15}$$

Consequently, when the multi-task kernel $\mathcal{K} = G\mathbf{I}_{d+1}$, we know from Theorem 2 that the dual problem of algorithm (5) is exactly

$$\max_\alpha -S_0(\alpha) - \sum_{\ell \in \mathbb{N}_d} S_\ell(\alpha).$$

Furthermore, the optimum is given, for any $\ell \in \mathbb{N}_d$, by $\mathbf{f}_{\ell,\mathbf{z}} = \sum_{ij} w_{ij} \alpha_{ij,\mathbf{z}} x_{ij}^\ell G_{x_j}$ which means that

$$\|\mathbf{f}_{\ell,\mathbf{z}}\|_G^2 = \mathbf{vec}(\alpha_\mathbf{z})^\top \mathbf{K}_\ell \mathbf{vec}(\alpha_\mathbf{z}) = 2S_\ell(\alpha_\mathbf{z}). \tag{16}$$

Since the norm of $\{\|\mathbf{f}_{\ell,\mathbf{z}}\|_G : \ell \in \mathbb{N}_d\}$ measures the saliency of coordinate variables, we know from equation (16) that the kernel matrix $\mathbf{K}_\ell$ determines the importance of the ℓ-th variable. For this reason we later refer to $\mathbf{K}_\ell$ as the ℓ-th *coordinate kernel matrix*. Consequently, from the dual perspective it indicates that the gradient learning algorithm (5) is identical to the case of taking summation of all coordinate kernel matrices, i.e.

$$\max_\alpha -S_0(\alpha) - \frac{1}{2} \mathbf{vec}(\alpha)^\top \Big(\sum_{\ell \in \mathbb{N}_d} \mathbf{K}_\ell \Big) \mathbf{vec}(\alpha). \tag{17}$$

With this motivation we propose the following formulation to learn the linear combination of coordinate kernel matrices:

$$\min_\beta \max_\alpha -S_0(\alpha) - \frac{1}{2} \mathbf{vec}(\alpha)^\top \Big(\sum_{\ell \in \mathbb{N}_d} \beta_\ell \mathbf{K}_\ell \Big) \mathbf{vec}(\alpha), \quad \text{s.t.} \sum_\ell \beta_\ell = 1, \quad \beta_\ell \geq 0, \ \forall \ \ell. \tag{18}$$

Equivalently,

$$\min_\beta \max_\alpha -S_0(\alpha) - \sum_{\ell \in \mathbb{N}_d} \beta_\ell S_\ell(\alpha), \quad \text{s.t.} \sum_\ell \beta_\ell = 1, \quad \beta_\ell \geq 0, \ \forall \ \ell, \tag{19}$$

where S_ℓ is defined by equation (15). For this reason, we refer to our gradient learning algorithm (19) as *GradKL*. Since the weights β is restricted to the simplex $\Delta = \{\beta : \sum_\ell \beta_\ell = 1, \ \beta_\ell \geq 0, \ \forall \ \ell\}$, this ℓ^1-regularization restriction on β promotes the sparsity of the coordinate kernel matrices $\{\mathbf{K}_\ell : \ell \in \mathbb{N}_d\}$, and hence intuitively gives the saliency of coordinates (variables).

In principal, we can formulate the above problem as semi-definite programming (SDP) or quadratically constrained quadratic programming (QCQP) as in [19]. We use the efficient semi-infinite linear optimization approach [34] which scales well with respect to large data sets. The GradKL algorithm (19) is a typical concave convex problem, i.e. concave with respect to α and convex with respect to β. Efficient bundle methods [20] would also be possible for the gradient learning problem (19).

4.1 Semi-infinite Linear Approach

In this subsection we show that algorithm (19) can be solved by a semi-infinite linear programming (SILP), see e.g. [16,34]. The SILP problem refers to an optimization problem that solves a linear objective function subject to infinite number of linear constraints.

Theorem 3 *The gradient learning algorithm (19) can be reformulated as a SILP problem:*

$$\begin{aligned}\max_{\gamma,\beta}\ & \gamma \\ s.t.\ & \sum_{\ell\in\mathbb{N}_d}\beta_\ell = 1, \quad 0\le\beta\le 1 \\ & \gamma - \sum_{\ell\in\mathbb{N}_d}\beta_\ell S_\ell(\alpha) \le S_0(\alpha), \forall\alpha.\end{aligned} \tag{20}$$

Proof We know that problem (19) can be equivalently formulated as

$$\max_\beta \min_\alpha S_0(\alpha) + \sum_{\ell\in\mathbb{N}_d}\beta_\ell S_\ell(\alpha).$$

Then, the proof is direct from the equivalence between $\gamma = \min_\alpha S_0(\alpha)+\sum_{\ell=0}^d \beta_\ell S_\ell(\alpha)$, and the equation $\gamma - \sum_{\ell\in\mathbb{N}_d}\beta_\ell S_\ell(\alpha) \le S_0(\alpha), \forall\alpha$. □

In equation (20), there are infinite number of constraints (indexed by α) which is usually termed *semi-infinite linear programming* (SILP). The SILP can be solved by an iterative algorithm called *column generation* (or exchange methods) which is guaranteed to converge to a global optimum. The basic idea is to compute the optimum (β,γ) by linear programming for a restricted subset of constraints, and update the constraint subset based on the obtained suboptimal (β,γ).
Given restricted constraints $\{\alpha_p : p = 1, ..., P\}$, first we find the intermediate solution (β,γ) by the following linear programming optimization with P linear constraints

$$\begin{aligned}\max_{\gamma,\beta}\ & \gamma \\ \text{s.t.}\ & \sum_\ell \beta_\ell = 1, \quad 0\le\beta\le 1 \\ & \gamma - \sum_\ell \beta_\ell S_\ell(\alpha_p) \le S_0(\alpha_p), \forall p = 1,\dots,P.\end{aligned} \tag{21}$$

This problem is often called the *restricted master problem*. Then, we find the next constraint with the maximum violation for the given intermediate solution (β,γ), i.e.

$$\min_\alpha \sum_{\ell\in\mathbb{N}_d}^{d} \beta_\ell S_\ell(\alpha) + S_0(\alpha). \tag{22}$$

If the optimizer α^* of the above equation satisfies $\sum_\ell \beta_\ell S_\ell(\alpha^*) + S_0(\alpha) \ge \gamma$ then the current intermediate solution (β,γ) is optimal for the optimization (20). Otherwise α^* should be added to the restriction set. We repeat the above iteration until convergence which is guaranteed to be globally optimal, see e.g. [16,34]. Similar to the convergence criterion in [34], the algorithm stops when

$$\left| 1 - \frac{\sum_{\ell} \beta_{\ell}^{(t-1)} S_{\ell}(\alpha^{(t)}) + S_0(\alpha^{(t)})}{\gamma^{(t-1)}} \right| \leq \epsilon.$$

By Theorem 2 the sub-optimization (22) is gradient learning with multi-task kernel $\mathcal{K} = \mathbf{B}G = \text{diag}([1,\beta])G$. We list below some examples for the sub-optimization problem (22).

Least Square Loss For least square loss, the maximum violation problem (22) can be obtained by the linear system

$$\begin{aligned} \min_{\alpha} \sum_{i,j \in \mathbb{N}_m} w_{ij} \Big(-\alpha_{ij} + \frac{\alpha_{ij}^2}{4C} \Big) + \frac{1}{2} \sum_{i,j,i',j' \in \mathbb{N}_m} & y_i w_{ij} \alpha_{ij} y_i w_{i'j'} \alpha_{i'j'} y_{i'} \\ & \times \Big[(\widetilde{e}_1 + \widetilde{x}_{ij})^{\top} \text{diag}([1,\beta])(\widetilde{e}_1 + \widetilde{x}_{i'j'}) G(x_j, x_{j'}) \Big] . \end{aligned}$$

Soft Margin Loss For SVM hinge loss, the sub-algorithm (22) is reduced to the following QP problem

$$\begin{aligned} \min_{\alpha} - \sum_{i,j \in \mathbb{N}_m} w_{ij} \alpha_{ij} + \frac{1}{2} \sum_{i,j,i',j' \in \mathbb{N}_m} & w_{ij} \alpha_{ij} y_i w_{i'j'} \alpha_{i'j'} y_{i'} \\ \times \Big[(\widetilde{e}_1 + \widetilde{x}_{ij})^{\top} & \text{diag}([1,\beta])(\widetilde{e}_1 + \widetilde{x}_{i'j'}) G(x_j, x_{j'}) \Big] \\ \text{s.t.} \;\; 0 \leq \alpha_{ij} \leq C, \;\; & \forall i, j = 1, \ldots, m. \end{aligned} \tag{23}$$

2-Norm Soft Margin Loss For 2-norm SVM hinge loss, subproblem (22) can be solved by the following QP optimization

$$\begin{aligned} \min_{\alpha} \sum_{i,j \in \mathbb{N}_m} w_{ij} \Big(-\alpha_{ij} + \frac{\alpha_{ij}^2}{4C} \Big) + \frac{1}{2} \sum_{i,j,i',j' \in \mathbb{N}_m} & w_{ij} \alpha_{ij} y_i w_{i'j'} \alpha_{i'j'} y_{i'} \\ \times \Big[(\widetilde{e}_1 + \widetilde{x}_{ij})^{\top} & \text{diag}([1,\beta])(\widetilde{e}_1 + \widetilde{x}_{i'j'}) G(x_j, x_{j'}) \Big] \\ \text{s.t.} \;\; 0 \leq \alpha_{ij}, \;\; \forall i, j = 1, \ldots, m. & \end{aligned}$$

The above formulation for gradient learning focuses on the learning weights β only when the trade-off parameter C is fixed. In analogy to the spirit of [12], we can easily extend this formulation with least square loss or 2-norm soft margin SVM loss to the case of automatically learning C and β together. However, no improved performance is guaranteed for this joint optimization compared with the case of fixed C.

4.2 Relation with ℓ^1-Regularization and Iterative Optimization

We further explore the equivalent form of algorithm (19) from the sparse regularization perspective. In particular we investigate its close relationship with the well-established sparse ℓ^1-regularization. To this end, given a scalar kernel G we consider the following set of multi-task kernels

$$\mathscr{K} = \Big\{\mathbf{B}G : \mathbf{B} = \operatorname{diag}(1, \beta_1, \ldots, \beta_d), \sum_{\ell\in\mathbb{N}_d} \beta_\ell = 1, \beta_\ell \geq 0, \forall \ell \in \mathbb{N}_d\Big\}. \tag{24}$$

We consider the following algorithm to jointly learn the gradient function and the multi-task kernel $\mathcal{K} \in \mathscr{K}$

$$\min_{\mathcal{K}\in\mathscr{K}} \min_{\mathbf{F}\in\mathscr{H}_{\mathcal{K}}} \|\mathbf{F}\|^2_{\mathcal{K}} + C \sum_{i,j\in\mathbb{N}_m} w_{ij} L\left(y_i, \langle \mathbf{F}, \mathcal{K}_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij})\rangle_{\mathcal{K}}\right). \tag{25}$$

Theorem 4 *Let* $\mathbf{B}$ *and* $\mathscr{K}$ *be defined by equation (24). Then, the dual problem of algorithm (25) is exactly (19). Moreover, (25) can be equivalently rewritten as*

$$\begin{aligned} &\min_{\mathbf{F}} \|f_0\|^2_G + \sum_{\ell\in\mathbb{N}_d} \frac{\|f_\ell\|^2_G}{\beta_\ell} + C \sum_{i,j\in\mathbb{N}_m} w_{ij} L\left(y_i, \langle \mathbf{F}, \mathbf{B}G_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij})\rangle_{\mathcal{K}}\right) \\ &s.t. \sum_{\ell\in\mathbb{N}_d} \beta_\ell = 1,\ \beta_\ell \geq 0,\quad \forall \ell \in \mathbb{N}_d. \end{aligned} \tag{26}$$

Proof For fixed $\mathcal{K} \in \mathscr{K}$ (i.e. fixed β), by Theorem 2 it is easy to check that the dual problem is identical to the following:

$$\max_{\alpha} - \sum_{\ell\in\mathbb{N}_d}^{d} \beta_\ell S_\ell(\alpha) - S_0(\alpha).$$

Consequently, the dual problem of algorithm (25) is reduced to algorithm (19), i.e.

$$\min_{\beta} \max_{\alpha} - \sum_{\ell\in\mathbb{N}_d}^{d} \beta_\ell S_\ell(\alpha) - S_0(\alpha),$$

The second argument is directly from the observation that, for any $\mathcal{K} = \mathbf{B}G$,

$$\|\mathbf{F}\|^2_{\mathcal{K}} = \|f_0\|^2_G + \sum_{\ell\in\mathbb{N}_d} \frac{\|f_\ell\|^2_G}{\beta_\ell}.$$

This completes the proof of the theorem. □

As mentioned above, the restriction of β can be regarded as ℓ^1-regularization. From the above equation, if some β_ℓ is close to zero then $\|f_\ell\|_G$ should be close to zero since we are minimizing the objective function. This will give sparsity for RKHS norms $\{\|f_\ell\|_G : \ell \in \mathbb{N}_d\}$. Indeed, it is shown in [25] that

$$\min_{\beta}\Big\{\sum_{\ell\in\mathbb{N}_d} \|f_\ell\|^2_G/\beta_\ell : \sum_{\ell} \beta_\ell = 1, \beta_\ell \geq 0\Big\} = \Big(\sum_{\ell} \|f_\ell\|_G\Big)^2. \tag{27}$$

Hence, algorithm (25) can be further reformulated as a direct minimization with ℓ^1-regularizer over the RKHS norms $\{\|f_\ell\|_G : \ell \in \mathbb{N}_d\}$:

$$\min_{\mathbf{F}} \|f_0\|_G^2 + \Big(\sum_{\ell \in \mathbb{N}_d} \|f_\ell\|_G\Big)^2 + C \sum_{i,j \in \mathbb{N}_m} w_{ij} L\left(y_i, \langle \mathbf{F}, \mathbf{B} G_{x_j}(\widetilde{e}_1 + \widetilde{x}_{ij})\rangle_{\mathcal{K}}\right). \qquad (28)$$

The regularization term in equation (27) is usually referred to as *block ℓ^1 regularization*. The parallel work of [34,25] used similar regularization terms but from the motivation of multiple kernel learning for data integration which is also known as *nonparametric groupd lasso* [3].
The SILP approach iteratively solves the linear programming problem using the simplex method, for example. When the feature dimension d is very large, of the order of thousands, the SILP approach suffers from computational inefficiency. As an alternative approach, we employ an alternate updating method which avoids linear programming. In particular, this alternative optimization approach builds on the following observation.

Theorem 5 *Suppose that the loss function $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is convex with respect to the second argument. Then, the gradient learning problem (25) is jointly convex with respective to $\mathbf{F}$ and λ.*

Proof It suffices to prove the joint convexity of $\|f\|_G^2/\lambda$ with respect to $f \in \mathscr{H}_G$ and $\lambda \in (0, 1)$. Its proof is parallel to that in [1] except replacing Euclidean space by RKHS and the positive semi-definite matrix D by λ. For completeness, we briefly prove it for our specific scenario. We need to show, for any $f_1, f_2 \in \mathscr{H}_G$ and $\lambda_1, \lambda_2 \in (0, 1)$ and $\theta \in (0, 1)$, that

$$\frac{\|\theta f_1 + (1-\theta) f_2\|_G^2}{\theta\lambda_1 + (1-\theta)\lambda_2} \leq \frac{\|\theta f_1\|_G^2}{\theta\lambda_1} + \frac{\|(1-\theta) f_2\|_G^2}{(1-\theta)\lambda_2}$$

Let $a = \frac{1}{\lambda_1\theta}, b = \frac{1}{(1-\theta)\lambda_2}, c = \frac{1}{\theta\lambda_1 + (1-\theta)\lambda_2}$ and $f = \theta f_1 + (1-\theta) f_2, g = \theta f_1$. Since f_1, f_2 is arbitrary, the above equation is reduced to the following:

$$c\|f\|_G^2 \leq a\|g\|_G^2 + b\|f - g\|_G^2, \quad \forall f, g \in \mathscr{H}_G.$$

Equivalently,

$$\begin{aligned} c\|f\|_G^2 &\leq \min_{g \in \mathscr{H}_G} a\|g\|_G^2 + b\|f - g\|_G^2 \\ &= \|f\|_G^2 \tfrac{b^2 a}{(a+b)^2} + \|f\|_G^2 \tfrac{a^2 b}{(a+b)^2}, \quad \forall f \in \mathscr{H}_G. \end{aligned}$$

which is obviously true by the definition of a, b, c. This completes the proof of the convexity. □

We now propose an extremely simple iterative approach from the primal problem (26). In particular, we first initialize $\beta_\ell^{(0)} = \frac{1}{d}$ for any $\ell \in \mathbb{N}_d$. Then, for this fixed kernel coefficient $\beta^{(0)}$, solve the dual formulation (9) for gradient learning with fixed multi-task kernel $\mathcal{K} = \operatorname{diag}\big(1, \beta_1^{(0)}, \ldots, \beta_d^{(0)}\big)G$. Next, for any $t \in \mathbb{N}$ we update $\beta^{(t)}$ for fixed $\mathbf{F}^{(t-1)}$ and update $\mathbf{F}^{(t)}$ for fixed $\beta^{(t)}$. We repeat the above iteration until convergence. The convergence can reasonably be monitored by changes of the objective function. Based on the joint convexity of algorithm (26) proved in Theorem 5, global convergence is expected. The detailed updates at step $t \in \mathbb{N}$ for general loss function L are listed as follows:

- For fixed $\mathbf{F}^{(t)}$, $\beta_\ell^{(t)} = \dfrac{\|f_\ell^{(t-1)}\|_G}{\sum\limits_\ell \|f_\ell^{(t-1)}\|_G}$ for any $\ell \in \mathbb{N}_d$.
- For given $\beta^{(t)}$, $f_\ell^{(t)}(\cdot) = \beta_\ell^{(t)} \sum\limits_i \alpha_{ij}^{(t)} w_{ij}(x_i^\ell - x_j^\ell)G(x_i, \cdot)$. Here, $\alpha^{(t)} = (\alpha_{ij}^{(t)})$ is given by solving the dual formulation

$$\alpha_{\mathbf{z}} = \arg\max_\alpha -C \sum_{i,j\in\mathbb{N}_m} L^*\Big(y_i, -\frac{\alpha_{ij}}{C}\Big) w_{ij} - \frac{1}{2} \sum_{i,j,i',j'\in\mathbb{N}_m} w_{ij}\alpha_{ij}w_{i'j'}\alpha_{i'j'} \\ \times \Big[(\widetilde{e}_1 + \widetilde{x}_{ij})^T \mathcal{K}(x_j, x_{j'})(\widetilde{e}_1 + \widetilde{x}_{i'j'})\Big].$$

where $\mathcal{K} = \mathrm{diag}(1, \beta_1^{(t)}, \ldots, \beta_d^{(t)})G$.

Recall [25] that

$$\Big(\frac{|w_1|}{\sum_{\ell\in\mathbb{N}_m} |w_\ell|}, \ldots, \frac{|w_m|}{\sum_{\ell\in\mathbb{N}_m} |w_\ell|}\Big) = \arg\min\Big\{ \sum_{\ell\in\mathbb{N}_d} \frac{w_\ell^2}{\beta_\ell} : \sum_{\ell\in\mathbb{N}_d} \beta_\ell = 1, \beta_\ell \geq 0\Big\}.$$

Hence, the first update for β follows directly by replacing, for any $\ell \in \mathbb{N}_d$, w_ℓ with $\|f_\ell\|_G$. By the dual formulation (9) of gradient learning with general multi-task kernels, letting $\mathcal{K} = \mathrm{diag}(1, \beta_1, \ldots, \beta_d)G$ directly yields the second update.
In [5] the LASSO type regularization $\sum_\ell \|f_\ell\|_G$ was introduced to gradient learning for regression. Precisely they estimate ∇f_* by

$$\mathbf{f}_{\mathbf{z}} = \arg\min_{\mathbf{f}\in\mathscr{H}_G^d} \sum_{ij\in\mathbb{N}_m} w_{ij}\Big(y_i - y_j - \mathbf{f}(x_j)\tilde{x}_{ij}\Big)^2 + \lambda \sum_\ell \|f_\ell\|_G.$$

Note in this algorithm y_j replaces $f_0(x_j)$ and only the gradient function is estimated. It can be solved by the proximal forward-backward splitting iteration technique [10]. Similar to the block regularization above, sparsity is used to realize automatic feature subset selection. When the least square loss is used, the empirical error term in the gradient learning formulation is exactly the same as the local polynomial modeling [13]. Kernel models introduced in [29,27,28,5] and this paper enable various penalties and the use of sparsity. This is important for variable selection.

5 Experiments

In this section we validate the proposed kernel matrix learning approaches for gradient learning with applications to variable selection and covariation.
To this end, we first discuss the computational complexity of gradient learning. From Theorem 2, the dual problem of SVM-like gradient learning can be solved by quadratic programming with respect to $\alpha \in \mathbb{R}^{m^2}$ with time complexity $\mathcal{O}(m^6)$ which makes gradient learning computationally prohibitive. For least square loss function or a proper class of multi-task kernels such as radial basis kernels, the computation time can be reduced to low dimension formulation with $\alpha \in \mathbb{R}^{sm}$ where s is the rank of data matrix $\{x_1 - x_m, x_2 - x_m, \ldots, x_{m-1} - x_m\}$, see [27,29,38]. However, in most cases s is either the number of training sample m or dimension of feature space d.

To reduce computational complexity, we define w_{ij} by equation (3) using k-nearest neighbors to restrict the Taylor expansion to first order. Hence, optimization is restricted to the space $\alpha \in \mathbb{R}^{mk}$ with $\alpha = \{\alpha_{ij} : i \in \mathbb{N}_m, j \in N_k(i)\}$. This will yield reduced complexity $\mathcal{O}(k^3m^3)$. When k is too small, we may lose useful information for learning gradient. On other hand if k is too large, as mentioned above we have high computational complexity. Moreover we may include noise information since the approximation to gradient function using a Taylor expansion is unreliable if samples are far away from each other. Although the parameter k should be chosen from appropriate cross-validation on the training samples, in our experience $k = 8$ is generally a proper choice from computational efficiency and performance.

Throughout all experiments, the parameter C is fixed to be $C = 1$, for simplicity, and the weights w_{ij} are given by equation (3) taking $k = 8$ nearest neighbor. All experiments are implemented using MATLAB[3]. For the implementation of quadratic programming and linear programming, we use the MOSEK package[4]. As in [27,29,38], for the solution $\mathbf{F_z} = (f_{0\mathbf{z}}, \mathbf{f_z})$ of gradient learning algorithms (19) (i.e. (26)) we use the coordinate covariance

$$\mathrm{Cov}(\mathbf{f_z}) = \Big(\langle f_{p\mathbf{z}}, f_{q\mathbf{z}}\rangle_G\Big)_{p,q\in\mathbb{N}_d} \tag{29}$$

to measure how the variables covary. Also, the variable (feature) ranking can be done according to the following relative magnitude of norm of each component of $\mathbf{f_z}$

$$s_p = \frac{\|f_{p\mathbf{z}}\|_G}{\sum\limits_{q\in\mathbb{N}_d} \|f_{q\mathbf{z}}\|_G}, \qquad \forall p \in \mathbb{N}_d. \tag{30}$$

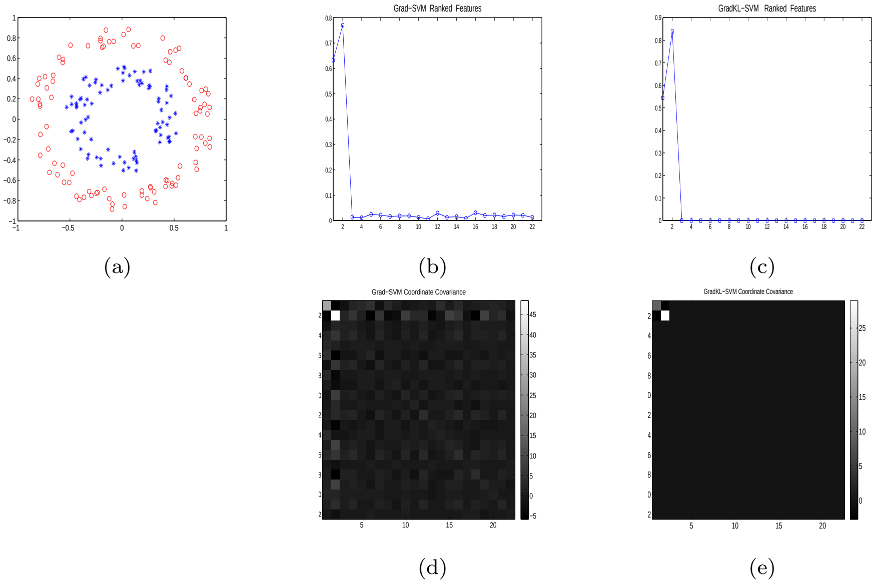
5.1 Synthetic Data

In the first experiment we compare two types of gradient learning with SVM soft margin loss. The first one uses an independent multi-task kernel $\mathcal{K} = GI_{(d+1)\times(d+1)}$ (i.e. algorithm (17)) (Grad-SVM) and the other one is kernel matrix learning for gradient learning (GradKL-SVM) defined by equation (19) (i.e. algorithm (26)). For the algorithm GradKL-SVM, we use the SILP approach.

In this experiment, only the first two features are relevant to the classification task. The remaining 20 redundant features are distributed according to a small Gaussian random deviate. The distribution for the first two features is shown in subfigure (a) of Figure 1. Subfigure (b) and (d) respectively shows ranked features and the covariance matrix $\mathrm{Cov}(\mathbf{f_z})$ using $\{s_p : p \in \mathbb{N}_d\}$ for gradient learning with SVM loss (Grad-SVM) which is equivalent to standard ℓ^2 regularization. Subfigure (c) and (e) are results of our proposed algorithm GradKL-SVM (19) which is shown to be equivalent to algorithm (26) with ℓ^1-regularization. As expected, GradKL-SVM is more effective at removing the redundant features.

[3] MATLAB code will soon be available in http://www.cs.ucl.ac.uk/staff/Y.Ying

[4] http://www.mosek.com/



(a) (b) (c)

(d) (e)

Fig. 1 Performance of Gradient learning with soft margin SVM; (a) illustration of the dataset with the first two features; (b) and (d) ranked features and coordinate covariance by Grad-SVM; (c) and (e) ranked features and coordinate covariance by GradKL-SVM.

5.2 Cancer Datasets

Next we test our gradient learning algorithm (19) with least square loss (GradKL-LS) and SVM hinge loss (GradKL-SVM) on real cancer datasets. We also compare it with the ordinary gradient learning with independent multi-task kernel $\mathcal{K} = GI_{(d+1)\times(d+1)}$ which respectively is denoted by Grad-LS and Grad-SVM depending on the loss function. Since the dimension of features is usually larger than thousands, we use the alternative iterative optimization to solve GradKL algorithms as presented in Section 4.2. To test the effect of variable selection by gradient learning, we first rank the genes by gradient learning on training samples and then make prediction on the test set which is performed by hard margin linear SVM.

The first dataset is a well-known Leukemia dataset [14] for analyzing gene expression data obtained from DNA micro-arrays in order to classify types of cancer. The problem is to distinguish between two variants of leukemia (ALL and AML). The data is split into a training set and a test set. The training set consists of 38 samples (27 ALL and 11 AML) from bone marrow specimens. The test set has 34 samples (20 ALL and 14 AML), prepared under different experimental conditions and including 24 bone marrow and 10 blood sample specimens. All samples have 7129 features, corresponding to some normalized gene expression value extracted from the micro-array image. The dataset is normalized to be mean zero and unit deviation. In [26], the authors got zero error on the test set using the top 40 genes. In [36], the authors report zero error with 20 genes and 1 error with 5 genes. The state-of-art result was achieved by [15] with zero test error for 8 genes.

Table 2 Test set accuracy comparison on Leukemia cancer dataset [14]; The result for SVM-RFE is cited from [15]. The missing results of SVM-RFE for top $3, 5, 6$ ranked genes are denoted by dash line. The best test set accuracy for each method is marked in bold.

No. of genes	GradKL-LS	GradKL-SVM	Grad-LS	Grad-SVM	SVM-RFE
1	0.91	0.91	0.76	0.76	0.79
2	0.94	0.94	0.76	0.79	0.88
3	**1.00**	**1.00**	0.73	0.79	–
4	**1.00**	**1.00**	0.70	0.76	0.91
5	**1.00**	**1.00**	0.82	0.82	–
6	**1.00**	**1.00**	0.82	0.82	–
8	0.82	**1.00**	0.88	0.79	**1.00**
16	0.91	0.94	0.91	0.91	**1.00**
32	0.91	0.91	0.94	0.94	0.97
64	0.91	0.94	0.91	0.91	0.94
128	0.85	0.94	**0.97**	**0.97**	0.97
256	0.94	0.97	**0.97**	**0.97**	0.94
512	0.97	0.94	0.91	0.91	0.88
1024	0.94	0.91	0.91	0.91	0.94
2048	0.94	0.94	0.91	0.91	0.85
4096	0.91	0.91	0.91	0.91	0.71
7129	0.91	0.91	0.91	0.91	0.85

We compare our method with SVM-RFE [15][5]. The result of SVM-RFE is directly taken from [15]. As listed in Table 2, it is quite notable that both GradKL-LS and GradKL-SVM achieved zero test error with only 3 genes while SVM-RFE needs 8 genes. This outperforms SVM-RFE with fewer genes to attain zero test error. The top six ranked genes by GradKL-LS is M23197, M19507, M20902, X70297, D49950, Y12670 among which gene M23197 and Y12670 are also observed to be among 50 genes with most prediction powers observed in [14] using correlation analysis. It is also worth mentioning that we did not use a backward recursive feature elimination strategy for GradKL in contrast to SVM-RFE method [15].

The second dataset is prostate cancer dataset [33]. In this dataset there are 12600 gene probes. The dataset is split into a training set and a test set. The training set has 102 samples with 52 tumor samples and 50 non-tumor samples. The independent test set has 34 samples from a different experiment.

All five methods are competitive with each other: only 1 error in the test set using the top two genes. Grad-LS and SVM-RFE performs slightly better since it attained zero test error respectively with 4 genes and 3 genes. This means that the sparse regularization formulation (25) for gradient learning (equivalently learning the coordinate matrix formulation (17)) does not guarantee good gradient learning with respect to variable selection. This is consistent with the fact that kernel matrix learning method does not necessarily guarantee better performance than the naive method of taking all average of base kernel matrices [18] unless we include irrelevant data sources. More interestingly, all methods indicate that gene X07732 (no. 6185) is the top ranked gene.

[5] MATLAB code can be obtained from http://www.kyb.mpg.de/bs/people/spider/

Table 3 Test set accuracy comparison on Prostate cancer dataset [33]. For each method, the best test set accuracy is marked in bold.

No. of genes	GradKL-LS	GradKL-SVM	Grad-LS	Grad-SVM	SVM-RFE
1	**0.97**	**0.97**	0.97	**0.97**	0.97
2	**0.97**	**0.97**	0.97	**0.97**	0.97
3	0.91	**0.97**	0.97	**0.97**	**1.00**
4	0.94	0.91	**1.00**	**0.97**	0.97
5	0.94	**0.97**	**1.00**	**0.97**	0.94
6	**0.97**	0.85	0.97	**0.97**	0.91
13	0.85	0.88	0.88	0.79	0.85
25	0.73	0.70	0.91	0.91	0.88
50	0.64	0.67	0.91	0.91	0.88
99	0.64	0.73	0.91	0.91	0.91
197	0.76	0.82	0.94	0.94	0.82
394	0.76	0.82	0.85	0.88	0.88
788	0.82	0.76	0.79	0.79	0.79
1575	0.82	0.82	0.79	0.79	0.79
3150	0.82	0.82	0.85	0.85	0.85
6300	0.85	0.85	0.85	0.85	0.91
12600	0.94	0.94	0.94	0.94	0.94

6 Conclusion

In this paper we first introduced a unifying approach for gradient learning in the framework of multi-task learning. Then, under this formulation dual problems for general gradient learning algorithms were straightforwardly established. Finally, motivated by this dual formulation we proposed a novel gradient learning formulation which can be cast as the problem of learning the kernel matrix [19]. A SILP optimization approach and an alternative iterative optimization were proposed to solve this problem efficiently. We validated our proposed approaches on both synthetic and real datasets. In particular our proposed method achieved competitive feature selection results on the Leukemia dataset [14] and Prostate cacner dataset [33] compared with existing method such as SVM-RFE [15].

There are several questions remaining to be further studied. Firstly, it will be interesting to apply the spectral decomposition of the gradient outer products to dimension reduction (see e.g. [28]), and the possible use for gene network inference from the covariance of the learned gradient function. The sparsity may benefit the dimension reduction [22,5]. Secondly, gradient learning aims at estimating the gradient ∇f_* of the target function f_*. When the SVM hinge loss is used, the target function is known to be the Bayes rule [21,42] which is usually not continuous. Hence, gradient learning with SVM loss could be mathematically inappropriate. However our simulations shows it works well in practice. It should be interesting to understand this theoretically. Last but not least, generalization analysis and optimal convergence rates of gradient learning algorithms (19) would be another direction for investigation using Rademacher complexity approaches [4,17].

Acknowledgements We thank the anonymous referees for their constructive comments. This work is supported by EPSRC grant EP/E027296/1

References

1. A. Argyriou, C. A. Micchelli, M. Pontil and Y. Ying. A spectral regularization framework for multi-task structure learning. *NIPS*, 2007.
2. N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.* 68: 337–404, 1950.
3. F. Bach. Consistency of the group Lasso and multiple kernel learning, J. of Machine Learning Research, 9, 1179-1225, 2008.
4. P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *J. of Machine Learning Research*, 3:463–482, 2002.
5. J. Cai and G. Ye. Variable selection and linear feature construction via sparse Gradients. Preprint, 2008.
6. A. Caponnetto, C.A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels, *J. of Machine Learning Research*, 9: 1615–1646, 2008.
7. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46: 131–159, 2002.
8. S. S. Chen, D. L. Donoho and M.A. Saunders. Atomic decomposition pursuits. *SIAM J. of Scientific Computing*, 20: 33-61, 1999.
9. D. R. Chen, Q. Wu, Y. Ying, and D. X. Zhou. Support vector machine soft margin classifiers: error analysis. *J. of Machine Learning Research*, 5:1143–1175, 2004.
10. P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200, 2005.
11. F. Cucker and S. Smale. On the mathematical foundations of learning, *Bull. Amer. Math. Soc.* 39: 1-49, 2001.
12. 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.
13. J. Fan and I. Gijbels, *Local Polynomial Modelling and its Applications*, Chapman and Hall, London, 1996.
14. T. R. Golub et. al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, 286: 531-537, 1999.
15. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning* 46: 389–422, 2002.
16. R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 3: 380–429, 1993.
17. V.I. Koltchinskii and D. Panchenko. Rademacher processes and bounding the risk of function learning. In J. Wellner E. Gin, D. Mason, editor, High Dimensional Probability II, pages 443–459, 2000.
18. 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: 2626-2653, 2004
19. G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *J. of Machine Learning Research* 5: 27–72 (2004).
20. C. Lemaréchal, A. Nemirovski, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, **69**, 1995.
21. Y. Lin. Support vector machines and the Bayes rule classification. *Data Mining and Knowledge Discovery* 6: 259C275, 2002.
22. L. Li and X. Yin. Sliced inverse regression with regularizations. Biometrics, 64: 124–131, 2008.
23. C. A. Micchelli and M. Pontil. Kernels for multi-task learning. *NIPS*, 2004.
24. C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17: 177–204, 2005.
25. C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. of Machine Learning Research* 6: 1099 -1125, 2005.
26. S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, A., T. Golub, J. P. Messirov, and T. Poggio. Support vector machine classification of microarray data. AI memo 182. CBCL paper 182, MIT.
27. S. Mukherjee and Q. Wu. Estimation of gradients and coordinate covariation in classification. *J. of Machine Learning Research* 7: 2481–2514, 2006.
28. S. Mukherjee, Q. Wu, and D. X. Zhou. Learning gradients on manifolds. *Bournoulli*, 2009, to appear.

29. S. Mukherjee and D. X. Zhou. Learning coordinate covariances via gradient. *J. of Machine Learning Research* 7: 519–549, 2006.
30. I. J. Schoenberg. Metric spaces and completely monotone functions. *Ann. of Math.* 39: 811-841, 1938.
31. B. Schölkopf and A. J. Smola. *Learning with Kernels.* The MIT Press, Cambridge, MA, USA, 2002.
32. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, 2004.
33. D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. DAmico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.
34. S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *J. of Machine Learning Research*, 7: 1531–1565, 2006.
35. R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.* 58: 267–288, 1996.
36. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs, *NIPS*, 2001.
37. Q. Wu, Y. Ying, and D. X. Zhou. Multi-kernel regularized classifiers. *J. of Complexity* **23** 108–134, 2007.
38. Y. Ying and C. Campbell. Learning coordinate gradients with multi-task kernels. *COLT*, 2008.
39. Y. Ying and D. X. Zhou. Learnability of Gaussians with fexible variances. *J. of Machine Learning Research*, 8: 249–276, 2007.
40. V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.
41. T. Zhang. On the dual formulation of regularized linear systems with convex risks. *Machine Learning*, 46, 2002.
42. T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals Statis.*, 32: 56-85, 2004.