Infinite Kernel Learning: Generalization Bounds and Algorithms

Yong Liu,[1] **Shizhong Liao,**[2] **Hailun Lin,**[1] **Yinliang Yue,**[1*] **Weiping Wang**[1]
[1]Institute of Information Engineering, CAS
[2]School of Computer Science and Technology, Tianjin University
{liuyong,linhailun,yueyinliang,wangweiping}@iie.ac.cn, {szliao,yongliu}@tju.edu.cn

Abstract

Kernel learning is a fundamental problem both in recent research and application of kernel methods. Existing kernel learning methods commonly use some measures of generalization errors to learn the optimal kernel in a convex (or conic) combination of prescribed basic kernels. However, the generalization bounds derived by these measures usually have slow convergence rates, and the basic kernels are finite and should be specified in advance. In this paper, we propose a new kernel learning method based on a novel measure of generalization error, called principal eigenvalue proportion (PEP), which can learn the optimal kernel with sharp generalization bounds over the convex hull of a possibly infinite set of basic kernels. We first derive sharp generalization bounds based on the PEP measure. Then we design two kernel learning algorithms for finite kernels and infinite kernels respectively, in which the derived sharp generalization bounds are exploited to guarantee faster convergence rates, moreover, basic kernels can be learned automatically for infinite kernel learning instead of being prescribed in advance. Theoretical analysis and empirical results demonstrate that the proposed kernel learning method outperforms the state-of-the-art kernel learning methods.

Introduction

Kernel methods have been successfully applied in solving various problems in machine learning community. The performance of these methods strongly depends on the choice of kernel functions (Micchelli and Pontil 2005). The earliest learning method of a kernel function is cross-validation, which is computationally expensive and only applicable to kernels with a small number of parameters. Minimizing theoretical estimate bounds of generalization error is an alternative to cross-validation (Liu, Jiang, and Liao 2014). To this end, some measures are introduced: such as VC dimension (Vapnik 2000), covering number (Zhang 2002), Rademacher complexity (Bartlett and Mendelson 2002), radius-margin (Vapnik 2000), maximal discrepancy (Anguita et al. 2012), etc. Unfortunately, the generalization bounds derived by these measures usually have slow convergence rates with order at most $O\big(\frac{1}{\sqrt{n}}\big)$, where n is the size of data set.

Instead of learning a single kernel, multiple kernel learning (MKL) follows a different route to learn a set of combination coefficients of basic kernels (Lanckriet et al. 2004; Bach, Lanckriet, and Jordan 2004; Ong, Smola, and Williamson 2005; Sonnenburg et al. 2006; Rakotomamonjy et al. 2008; Kloft et al. 2009; 2011; Cortes, Mohri, and Rostamizadeh 2010; Cortes, Kloft, and Mohri 2013; Liu, Liao, and Hou 2011). Within this framework, the final kernel is usually a convex (or conic) combination of finite basic kernels that should be specified in advance by users. To improve the accuracy of MKL, some researchers studied the problem of learning a kernel in the convex hull of a prescribed set of continuously parameterized basic kernels (Micchelli and Pontil 2005; Argyriou, Micchelli, and Pontil 2005; Argyriou et al. 2006; Gehler and Nowozin 2008; Ghiasi-Shirazi, Safabakhsh, and Shamsi 2010). This flexibility of the kernel class can translate into significant improvements in terms of accuracy. However, the measures used for the existing finite and infinite MKL algorithms are usually radius-margin (SVM objective) or other related regularization functionals, which usually have slow convergence rate.

In this paper, we introduce a novel measure of generalization errors based on spectral analysis, called principal eigenvalue proportion (PEP), and create a new kernel learning method over a possibly infinite set of basic kernels. We first derive generalization bounds with convergence rates of order $O(\frac{\log(n)}{n})$ based on the PEP measure. By minimizing the derived PEP-based sharp generalization bounds, we design two new kernel learning algorithms for finite kernel and infinite kernels respectively: one can be formulated as a convex optimization and the other one as a semi-infinite program. For infinite case, the basic kernels are learned automatically instead of being specified in advance. Experimental results on lots of benchmark data sets show that our proposed method can significantly outperform the existing kernel learning methods. Theoretical analysis and experimental results demonstrate that our PEP-based kernel learning method is sound and effective.

Related Work

In this subsection, we introduce the related measures of generalization error and infinite kernel learning methods.

*Corresponding author

Measures of Generalization Error In recent years, local Rademacher complexities were used to derive sharper generalization bounds. Koltchinskii and Panchenko (2000) first applied the notion of the local Rademacher complexity to obtain data dependent upper bounds using an iterative method. Lugosi and Wegkamp (2004) established the oracle inequalities using this notion and demonstrated its advantages over those based on the complexity of the whole model class. Bartlett, Bousquet, and Mendelson (2005) derived generalization bounds based on a local and empirical version of Rademacher complexity, and further presented some applications to classification and prediction with convex function classes. Koltchinskii (2006) proposed new bounds in terms of the local Rademacher complexity, and applied these bounds to develop model selection techniques in abstract risk minimization problems. Srebro, Sridharan, and Tewari (2010) established an excess risk bound for ERM with local Rademacher complexity. Mendelson (2003) presented sharp bounds on the local Rademacher complexity of the reproducing kernel Hilbert space in terms of the eigenvalues of integral operator associated with kernel function. Based on the connection between local Rademacher complexity and the tail eigenvalues of integral operator, Kloft and Blanchard (2012) derived generalization bounds for MKL. Unfortunately, the eigenvalues of integral operator of kernel function are difficult to compute, so Cortes, Kloft, and Mohri (2013) used the tail eigenvalues of kernel matrix, that is, the empirical version of the tail eigenvalues of integral operator, to design new kernel learning algorithms. However, the generalization bound based on the tail eigenvalues of kernel matrix was not established. Moreover, for different kinds of kernel functions (or the same kind but with different parameters), the discrepancies of eigenvalues of different kernels may be very large, hence the absolute value of the tail eigenvalues of kernel function can not precisely reflect the goodness of different kernels. Liu and Liao (2015) first considered the relative value of eigenvalues for kernel methods. In this paper, we consider another measure of the relative value of eigenvalues, that is, the proportion of the sum of the first t largest principal eigenvalues to that of the all eigenvalues, for kernel learning. Moreover, we derive sharper bounds with order $O(\frac{\log(n)}{n})$ using this relative value of eigenvalues of kernel matrix.

Infinite Kernel Learning Lots of work focuses on the finite kernel learning, but few studies the infinite case. The seminal work (Micchelli and Pontil 2005) generalized kernel learning to convex combination of an infinite number of kernels indexed by a compact set. Argyriou et al. (2006) also proposed an efficient DC programming algorithm for learning kernels. Gehler and Nowozin (2008) reformulated the optimization problem of (Argyriou et al. 2006), and proposed an infinite kernel learning framework for solving it numerically. Ghiasi-Shirazi, Safabakhsh, and Shamsi (2010) considered the problem of optimizing a kernel function over the class of translation invariant kernels. Although using the infinite kernels can improve the accuracy, the objective of the existing infinite kernel learning methods are all SVM object or other related regularization functionals, which have slow convergence rates and usually only applicable for classification. In this paper, we propose an infinite kernel learning method, which use the PEP-based measure with fast convergence rate and can be applicable both for classification and regression.

Notations and Preliminaries

We consider supervised learning with a sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of size n drawn i.i.d. from a fixed, but unknown probability distribution P on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes the output domain, $\mathcal{Y} = \{-1, +1\}$ for classification, $\mathcal{Y} \subseteq \mathbb{R}$ for regression.

Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function, and $\mathbf{K} = \left[\frac{1}{n}K(\mathbf{x}_i, \mathbf{x}_j)\right]_{i,j=1}^n$ be its corresponding kernel matrix. For most of MKL algorithms, the kernel is learned over a convex combination of finite basic kernels:

$$\mathcal{K}^{\text{finite}} = \left\{ \sum_{i=1}^m d_i K_{\theta_i}, d_i \geq 0, \sum_{i=1}^m d_i = 1 \right\}, \qquad (1)$$

where, $K_{\theta_i}, i = 1, \ldots, m$, are the basic kernels. In (Micchelli and Pontil 2005), they show that keeping the number m of basic kernels fixed is an unnecessary restriction and one can instead search over a possibly infinite set of basic kernels to improve the accuracy. Therefore, in this paper, we consider the general kernel classes:

$$\mathcal{K}^{\text{infinite}} = \left\{ \int_\Omega K_\theta \mathrm{d}p(\theta) : p \in \mathcal{M}(\Omega) \right\}, \qquad (2)$$

where K_θ is a kernel function associated with the parameter $\theta \in \Omega$, Ω is a compact set and $\mathcal{M}(\Omega)$ is the set of all probability measures on Ω. Note that Ω can be a continuously parameterized set of basic kernels. For example, $\Omega \subset \mathbb{R}_+$ and $K_\theta(\mathbf{x}, \mathbf{x}') = \exp(-\theta\|\mathbf{x} - \mathbf{x}'\|^2)$, or $\Omega = [1, c]$ and $K_\theta(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}')^\theta$. If $\Omega = \mathbb{N}_m$, $\mathcal{K}^{\text{infinite}}$ corresponds to the $\mathcal{K}^{\text{finite}}$.

The *generalization error* (or *risk*)

$$R(f) := \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(\mathbf{x}), y) \mathrm{d}P(\mathbf{x}, y)$$

associated with a hypothesis f is defined through a loss function $\ell(f(\mathbf{x}), y) : \mathcal{Y} \times \mathcal{Y} \to [0, M]$, M is a constant. In this paper, for classification, ℓ is the hinge loss: $\ell(t, y) = \max(0, 1 - yt)$; for regression, ℓ is the ϵ-loss: $\ell(t, y) = \max(0, |y - t| - \epsilon)$. Since the probability distribution P is unknown, $R(f)$ cannot be explicitly computed, thus we have to resort to its empirical estimator:

$$\hat{R}(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i), y_i).$$

In the following, assume that $\kappa = \sup_{\mathbf{x}\in\mathcal{X}} K_\theta(\mathbf{x}, \mathbf{x}) < \infty$ for any $\theta \in \Omega$, and K_θ is differentiable w.r.t θ.

Generalization Bounds

In this section, we will introduce a novel measure of generalization errors, called principal eigenvalue proportion (PEP), and use the measure to derive generalization bounds with fast convergence rates.

Principal Eigenvalue Proportion (PEP)

Definition 1 (Principal Eigenvalue Proportion)**.** *Let K be a Mercer kernel and $K(\mathbf{x},\mathbf{x}') = \sum_{j=1}^{\infty}\lambda_j(K)\phi_j(\mathbf{x})\phi_j(\mathbf{x}')$ be its eigenvalue decomposition, where $(\lambda_j(K))_{j=1}^{\infty}$ is the sequence of eigenvalues of kernel function arranged in descending order. Then, the t-principal eigenvalue proportion (t-PEP) of K, $t \in \mathbb{N}_+$, is defined as*

$$\beta(K,t) = \frac{\sum_{i=1}^{t}\lambda_i(K)}{\sum_{i=1}^{\infty}\lambda_i(K)}.$$

One can see that the t-PEP is the proportion of the sum of the first t principal eigenvalues to that of all eigenvalues of a kernel function. Note that $\beta(K,t) \in [0,1]$, and it is invariant with respect to scaling of K, that is $\beta(K,t) = \beta(cK,t)$ for any $c \neq 0$. In the next subsection, we will show that the larger the value $\beta(K,t)$ is, the sharper the generalization bound is. Thus, $\beta(K,t)$ can be considered as a complexity of K, and the larger value means the less complexity of K.

Definition 2 (Empirical PEP)**.** *Let K be a kernel function and $\mathbf{K} = \left[\frac{1}{n}K(\mathbf{x}_i,\mathbf{x}_j)\right]_{i,j=1}^{n}$ be its kernel matrix. Then, the t-empirical PEP of K, $t \in \{1,\ldots,n-1\}$, is defined as*

$$\hat{\beta}(K,t) = \Big(\sum_{i=1}^{t}\lambda_i(\mathbf{K})\Big)/\mathrm{tr}(\mathbf{K}),$$

where $\lambda_i(\mathbf{K})$ is the i-th eigenvalue of $\mathbf{K}$ in descending order, and $\mathrm{tr}(\mathbf{K})$ is the trace of $\mathbf{K}$.

The empirical PEP can considered as an empirical version of PEP, which can also be used to yield generalization bounds. The empirical PEP can be computed from empirical data, so we can use it for practical application. Note that the t-empirical PEP can be computed in $O(tn^2)$ time for each kernel because it is sufficient to compute the t largest eigenvalues and the trace of $\mathbf{K}$.

Generalization Bounds with PEP

Theorem 1. *Let $\ell(\cdot,\cdot)$ be an L-Lipschitz loss function associated with the first variable. For $\forall\, K \in \mathcal{K}^{\mathrm{infinte}}$, denote*

$$\mathcal{H}_K^{\mathrm{infinite}} = \{\langle \mathbf{w}, \Phi_K(\mathbf{x})\rangle : \|\mathbf{w}\| \leq \Delta\},$$

where $\mathcal{K}^{\mathrm{infinte}}$ is the infinite kernel class defined in Equation (2)*. Then, $\forall\delta > 0$, with probability at least $1-\delta$ over the choice of a sample $S = \{(\mathbf{x}_i, y)\}_{i=1}^{n}$ drawn i.i.d according to P, the following inequality holds: $\forall\ k > 1$ and $\forall f \in \mathcal{H}_K^{\mathrm{infinite}}$,*

$$R(f) \leq \frac{k}{k-1}\hat{R}(f) + c_1\frac{k\int_{\Omega}(1-\beta(K_\theta,t))\mathrm{d}p(\theta)}{n} + c_2\frac{k}{n}$$

where $c_1 = 20\Delta^2L^2\kappa$, $c_2 = (4+11M)\log\frac{1}{\delta} + 20L^2t$.

According to the above theorem, one can see that, for any t, larger value of $\int_{\Omega}\beta(K_\theta,t)\mathrm{d}p(\theta)$ leads to sharper generalization bound. Moreover, if we set $k = \log(n)$, the convergence rate of $R(f) - \frac{k}{k-1}\hat{R}(f)$ is

$$O\left(k\Big(\frac{1-\int_{\Omega}\beta(K_\theta,t)\mathrm{d}p(\theta)}{n} + \frac{1}{n}\Big)\right) = O\left(\frac{\log(n)}{n}\right).$$

When n is not very small, we know that $\frac{k}{k-1} = \frac{\log(n)}{\log(n)-1} \approx 1$, so $R(f) - \frac{k}{k-1}\hat{R}(f) \approx R(f) - \hat{R}(f)$.

Note that the hinge loss and ϵ-loss are L-Lipschitz loss functions with $L = 1$, which shows that the assumption of L-Lipschitz on loss function is reasonable.

Traditional Generalization Error Bounds Generalization bounds based on Rademacher complexity are standard (Bartlett and Mendelson 2002). For any $\delta > 0$, $f \in \mathcal{H}$, with probability $1-\delta$,

$$R(f) - \hat{R}(f) \leq \mathfrak{R}_n(\mathcal{H})/2 + \sqrt{\ln(1/\delta)/2n},$$

where $\mathfrak{R}_n(\mathcal{H})$ is Rademacher averages of $\mathcal{H}$. $\mathfrak{R}_n(\mathcal{H})$ is in the order of $O(\frac{1}{\sqrt{n}})$ for various kernel classes used in practice. Thus, this bound converges at rate $O(\frac{1}{\sqrt{n}})$ at most.

For radius-margin bound (Vapnik 2000) which is usually adopted for MKL, with probability at least $1-\delta$, the following inequality holds:

$$R(f) - \hat{R}(f) \leq \sqrt{c\left(\frac{R^2\log^2 n}{\rho^2} - \log\delta\right)/n}.$$

This bound also converges at rate $O\Big(\sqrt{\frac{\log^2 n}{n}}\Big)$.

Bousquet and Elisseeff (2002) derived the stability-based generalization bounds, and proved that if the algorithm has uniform stability η, then with $1-\delta$,

$$R(f) - \hat{R}(f) \leq 2\eta + (4n\cdot\eta + M)\sqrt{\log\Big(\frac{1}{\delta}\Big)/2n}.$$

The order of convergence rate is at most $O(\frac{1}{\sqrt{n}})$.

The above theoretical analysis indicates that using the PEP measure can obtain sharper bounds with fast convergence rates, which also demonstrates the effectiveness of the use of PEP to estimate generalization error.

If Ω is a finite set, from Theorem 1, it is easy to prove that:

Corollary 1. *Assume that ℓ is an L-Lipschitz loss function. $\forall K \in \mathcal{K}^{\mathrm{finite}}$, denote*

$$\mathcal{H}_K^{\mathrm{finite}} = \{\langle \mathbf{w}, \Phi_K(\mathbf{x})\rangle : \|\mathbf{w}\| \leq \Delta\},$$

where $\mathcal{K}^{\mathrm{finite}}$ is the finite kernel class defined in (1)*. Then, with probability $1-\delta$, the following inequality holds: $\forall\, k \geq 1$ and $f \in \mathcal{H}_K^{\mathrm{finite}}$,*

$$R(f) \leq \frac{k}{k-1}\hat{R}(f) + c_1\frac{k\sum_{i=1}^{m}d_i(1-\beta(K_{\theta_i},t))}{n} + c_2\frac{k}{n},$$

where $c_1 = 20\Delta^2L^2\kappa$, $c_2 = (4+11M)\log\frac{1}{\delta} + 20kL^2t$.

This result is remarkable since the generalization bound admits an explicit dependency on the combination coefficients d_i. It is a weighted average of $1-\beta(K_{\theta_i},t)$ with combination weights d_i. Note that $\beta(K_{\theta_i},t)$ can be considered as a complexity of K_{θ_i}, the larger the value $1-\beta(K_{\theta_i},t)$ is, the more complex the K_{θ_i} is. This bound suggests that, while some K_{θ_i} could have large complexities, they may not be detrimental to generalization if the corresponding total combination weight is relatively small. Thus, to guarantee good generalization performance, it is reasonable to learn a kernel function by minimizing $\hat{R}(f)$, $\|\mathbf{w}\|^2$ and $\sum_i d_i(1-\beta(K_{\theta_i},t))$. Moreover, we can see that the convergence rate can reach $O(\frac{\log(n)}{n})$ when setting $k = \log(n)$.

Generalization Bounds with Empirical PEP

Since it's not easy to compute the value of PEP, we have to use the empirical PEP for practical kernel learning. Fortunately, we can also use it to derive sharp bound.

Theorem 2. *Assume ℓ is an L-Lipschitz loss function. Then, with probability $1-\delta$, $\forall\, k \geq 1$ and $f \in \mathcal{H}_K^{\text{infinite}}$,*

$$R(f) \leq \frac{k}{k-1}\hat{R}(f) + c_3\frac{k\int_\Omega(1-\hat{\beta}(K_\theta,t))\mathrm{d}p(\theta)}{n} + c_4\frac{k}{n}$$

where $c_3 = 40\Delta^2L^2\kappa$, $c_4 = (14+2M)\log\frac{1}{\delta} + 40L^2t$.

One can see that the convergence rate can also reach $O(\frac{\log(n)}{n})$ when setting $k = \log(n)$, which indicates the effectiveness of empirical PEP to estimate the generalization error.

Corollary 2. *Assume ℓ is an L-Lipschitz loss function. Then, with $1-\delta$, $\forall\, k \geq 1$, $f \in \mathcal{H}_K^{\text{finite}}$, we have*

$$R(f) \leq \frac{k}{k-1}\hat{R}(f) + c_3\frac{k\sum_{i=1}^m d_i(1-\hat{\beta}(K_{\theta_i},t))}{n} + c_4\frac{k}{n}$$

where $c_3 = 40\Delta^2L^2\kappa$, $c_4 = (14+2M)\log\frac{1}{\delta} + 40kL^2t$.

Kernel Learning Algorithms

In this section, we will exploit the empirical PEP to devise kernel learning algorithms for finite and infinite kernels.

Finite Kernel Learning

This subsection studies the finite case. In this case, the finial kernel $K_{\mathbf{d}}$ can be written as

$$K_{\mathbf{d}} = \sum_{i=1}^m d_iK_{\theta_i}, d_i \geq 0, \sum_{i=1}^m d_i = 1,$$

where K_{θ_i}, $i = 1,\ldots,m$, are the basic kernels. According to the theoretical analysis of the above section, to guarantee generalization performance, we can formulate the following optimization:

$$\begin{aligned}\min_{\mathbf{d}}\mathfrak{P}(\mathbf{d}) = \min_{\mathbf{w},b}\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^n\ell(f(\mathbf{x}_i),y_i)\\ -\lambda\sum_{i=1}^m d_i\hat{\beta}_i(t),\\ \text{s.t. } \mathbf{d}\geq\mathbf{0}, \|\mathbf{d}\|_1 = 1,\end{aligned} \tag{3}$$

where $\hat{\beta}_i(t)$ is the t-empirical PEP of K_{θ_i}, C and λ are two trade off parameters to control the balance between the empirical loss and empirical PEP, $f(\mathbf{x}) = \langle\mathbf{w},\Phi_{\mathbf{d}}(\mathbf{x})\rangle + b$ with $K_{\mathbf{d}}(\mathbf{x},\mathbf{x}') = \langle\Phi_{\mathbf{d}}(\mathbf{x}),\Phi_{\mathbf{d}}(\mathbf{x}')\rangle$.

To solve the optimization (3), we need to calculate the gradient $\nabla\mathfrak{P}(\mathbf{d})$. This can be achieved by moving to the dual formulation of $\mathfrak{P}(\mathbf{d})$ given by (for classification and regression respectively)

$$\begin{aligned}\mathfrak{D}_C(\mathbf{d}) = \max_{\boldsymbol{\alpha}}\mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}\mathbf{K}_{\mathbf{d}}\mathbf{Y}\boldsymbol{\alpha} - \lambda\mathbf{d}^{\mathrm{T}}\hat{\boldsymbol{\beta}}(t),\\ \text{s.t. } \mathbf{1}^{\mathrm{T}}\mathbf{Y}\boldsymbol{\alpha} = 0, 0\leq\boldsymbol{\alpha}\leq C,\end{aligned}$$

and

$$\begin{aligned}\mathfrak{D}_R(\mathbf{d}) = \max_{\boldsymbol{\alpha}}\mathbf{1}^{\mathrm{T}}\mathbf{Y}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}_{\mathbf{d}}\boldsymbol{\alpha} - \lambda\mathbf{d}^{\mathrm{T}}\hat{\boldsymbol{\beta}}(t) - \epsilon\mathbf{1}^{\mathrm{T}}|\boldsymbol{\alpha}|\\ \text{s.t. } \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} = 0, 0\leq|\boldsymbol{\alpha}|\leq C,\end{aligned}$$

where $\mathbf{K}_{\mathbf{d}} = \sum_{i=1}^m d_i\mathbf{K}_{\theta_i}$ is the kernel matrix for a given $\mathbf{d}$, $\mathbf{Y} = \mathrm{diag}(y_1,\ldots,y_n)$, and $\hat{\boldsymbol{\beta}}(t) = (\hat{\beta}_1(t),\ldots,\hat{\beta}_m(t))^{\mathrm{T}}$. Note that we can write $\mathfrak{P} = E - \lambda\mathbf{d}^{\mathrm{T}}\hat{\boldsymbol{\beta}}(t)$ and $\mathfrak{D} = W - \lambda\mathbf{d}^{\mathrm{T}}\hat{\boldsymbol{\beta}}(t)$ with strong duality holding between E and W. Thus, given any value of $\mathbf{d}$, from Lemma 2 in (Chapelle et al. 2002), we have

$$\frac{\partial\mathfrak{P}}{\partial d_k} = \frac{\partial\mathfrak{D}}{\partial d_k} = -\lambda\hat{\beta}_k(t) - \frac{1}{2}\boldsymbol{\alpha}^*\mathbf{H}\boldsymbol{\alpha}^*,$$

where $\mathbf{H} = \mathbf{Y}\mathbf{K}_{\theta_k}\mathbf{Y}$ for classification and $\mathbf{H} = \mathbf{K}_{\theta_k}$ for regression, $\boldsymbol{\alpha}^*$ is the optimal solution of the dual optimization $\mathfrak{D}$. Thus, all we need to take a gradient step is to obtain $\boldsymbol{\alpha}^*$. If $\mathbf{d}$ is fixed, $\mathfrak{P}$ or $\mathfrak{D}$ are equivalent to their corresponding single kernel with kernel matrix $\mathbf{K}_{\mathbf{d}}$. The PEP-based finite kernel learning algorithm (FKL) is summarized in Algorithm 1. The step size η is chosen based on the Armijo rule to guarantee convergence and the projection step, for the constraints $\mathbf{d}\geq 0$ and $\|\mathbf{d}\|_1 = 1$, is as simple as $\mathbf{d}\leftarrow\max(0,\mathbf{d})$, $\mathbf{d}\leftarrow\frac{\mathbf{d}}{\|\mathbf{d}\|_1}$. From the similar convergence analysis of (Rakotomamonjy et al. 2008), it is easy to verity that our FKL algorithm is convergent.

Algorithm 1 Finite Kernel Learning (FKL)

1: **Initialize:** Basic kernels $\{K_{\theta_i}\}_{i=1}^m$, $d_i^0 = \frac{1}{m}$, $\hat{\beta}_i(t) = \sum_{j=1}^t\lambda_j(\mathbf{K}_{\theta i})/\mathrm{tr}(\mathbf{K}_{\theta_i})$, $i = 1,\ldots,m$, $s = 0$.
2: **repeat**
3: $\quad\mathbf{K}_{\mathbf{d}^s} = \sum_{i=1}^m d_i^s\mathbf{K}_{\theta_i}$.
4: $\quad$Use an SVM solver to solve the single kernel problem with $\mathbf{K}_{\mathbf{d}^s}$ and obtain the optimal solution $\boldsymbol{\alpha}^s$.
5: $\quad d_k^{s+1}\leftarrow d_k^s + \eta(\lambda\hat{\beta}_k(t) + \frac{1}{2}\boldsymbol{\alpha}^s\mathbf{H}\boldsymbol{\alpha}^s)$, $k = 1,\ldots,m$.
6: $\quad\mathbf{d}^{s+1}\leftarrow\max(0,\mathbf{d}^{s+1})$, $\mathbf{d}^{s+1} = \mathbf{d}^{s+1}/\|\mathbf{d}^{s+1}\|_1$.
7: $\quad s\leftarrow s+1$.
8: **until** converged

The time complexity of FKL algorithm is $O(tmn^2 + kn^2 + kA)$, where m is size of basic kernels, n is the size of data set, A is the time complexity of training a single kernel problem, and k is the number of iterations of FKL. In our experiments, we find that our FKL algorithm converges very quickly.

Infinite Kernel Learning

In this subsection, we consider the infinite case. The objective we are interested in is the best possible finite kernel learning object:

$$\begin{aligned}\inf_{\Omega_{\text{finite}}\subset\Omega}\min_{\mathbf{d},\mathbf{w},b}\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^n\ell(f(\mathbf{x}_i),y_i)\\ -\lambda\mathbf{d}^{\mathrm{T}}\hat{\boldsymbol{\beta}}(t),\\ \text{s.t.}\sum_{\theta\in\Omega_{\text{finite}}}d_\theta = 1, d_\theta\geq 0, \forall\theta\in\Omega_{\text{finite}},\end{aligned} \tag{4}$$

where Ω_{finite} is a finite set and Ω is a continuous parameterized set. The inner minimization problem is a finite kernel learning problem, from (Sonnenburg, Rätsch, and Schäfer 2006; Sonnenburg et al. 2006), the dual problem of optimization (4) can be written as

$$\begin{aligned} \sup_{\Omega_{\text{finite}} \subset \Omega} \max_{\boldsymbol{\alpha}, \xi} & h(\boldsymbol{\alpha}) - \xi, \\ \text{s.t. } & \xi \in \mathbb{R}, 0 \leq q(\alpha_i) \leq C, \\ & Q(\theta; \boldsymbol{\alpha}) \leq \xi, \forall \theta \in \Omega_{\text{finite}}, \end{aligned} \tag{5}$$

for classification, $h(\boldsymbol{\alpha}) = \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha}, q(\alpha_i) = \alpha_i, Q(\theta; \boldsymbol{\alpha}) = \lambda\hat{\beta}_\theta(t) + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}\mathbf{K}_\theta\mathbf{Y}\boldsymbol{\alpha}$; for regression, $h(\boldsymbol{\alpha}) = \mathbf{1}^{\mathrm{T}}\mathbf{Y}\boldsymbol{\alpha} - \epsilon\mathbf{1}^{\mathrm{T}}|\boldsymbol{\alpha}|, q(\alpha_i) = |\alpha_i|, Q(\theta; \boldsymbol{\alpha}) = \lambda\hat{\beta}_\theta(t) + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}_\theta\boldsymbol{\alpha}$, where $\hat{\beta}_\theta(t) = \sum_{j=1}^{t}\lambda_j(\mathbf{K}_\theta)/\mathrm{tr}(\mathbf{K}_\theta)$. Note that if some point $(\boldsymbol{\alpha}^*, \xi^*)$ satisfies the last condition of the optimization (5) for all $\theta \in \Omega$, then it also satisfies the condition for all finite subsets thereof. Thus we omit the supermum of optimization (5) and extend the program to the following semi-infinite program:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \xi} & h(\boldsymbol{\alpha}) - \xi, \\ \text{s.t. } & \xi \in \mathbb{R}, 0 \leq q(\alpha_i) \leq C \\ & Q(\theta; \boldsymbol{\alpha}) \leq \xi, \forall \theta \in \Omega. \end{aligned} \tag{6}$$

The dual form (6) of the problem suggests a delayed constraint generation approach to solve it. We start with a finite constraint set $\Omega^0 \subset \Omega$, and search for the optimal $\boldsymbol{\alpha}^*$. Then, we find the new θ, $\theta \leftarrow \arg\max_{\theta\in\Omega} Q(\theta, \boldsymbol{\alpha}^*)$, to add it into the set of Ω^0. The violated constraints are subsequently included in $\Omega^s \subset \Omega^{s+1} \subset \Omega$. The algorithm of PEP-based infinite kernel learning (IFKL) is summarized in Algorithm 2. One can see that the basic kernels are selected automatically during the learning phase.

Algorithm 2 Infinite Kernel Learning (IFKL)

1: **Initialize:** Continuous parameterized set Ω, randomly select $\theta^0 \in \Omega$, $\Omega^0 = \emptyset$, $s = 0$, $\boldsymbol{\alpha}^0 = 0$, $\xi = -\infty$.
2: **while** $Q(\theta^s; \boldsymbol{\alpha}^s) > \xi$ **do**
3: $\quad\Omega^{s+1} = \Omega^s \cup \{\theta^s\}$.
4: $\quad$Using FKL (Algorithm 1) with Ω^{s+1} as basic kernels to obtain the optimal solution $\boldsymbol{\alpha}^{s+1}$.
5: $\quad\xi = \max_{\theta\in\Omega^{s+1}} Q(\theta, \boldsymbol{\alpha}^{s+1})$.
6: $\quad\theta^{s+1} \leftarrow \arg\max_{\theta\in\Omega} Q(\theta, \boldsymbol{\alpha}^{s+1})$ {Sub-Problem, solved in Algorithm 3}.
7: $\quad s \leftarrow s + 1$
8: **end while**

From Theorem 7.2 in (Hettich and Kortanek 1993), we know that if the sub-problem (line 6 in Algorithm 2) can be solved, Algorithm 2 stops after a finite number of iterations or has at least one point of accumulation and each one of these points solve optimization (6).

To run Algorithm 2, we should solve the sub-problem (line 6 of Algorithm 2):

$$\arg\max_{\theta\in\Omega} Q(\theta; \boldsymbol{\alpha}) = \lambda\hat{\beta}_\theta(t) + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}\boldsymbol{\alpha}, \tag{7}$$

where $\mathbf{H} = \mathbf{Y}\mathbf{K}_\theta\mathbf{Y}$ for classification, $\mathbf{H} = \mathbf{K}_\theta$ for regression. We want to use the gradient-based algorithm to search the maxima of $Q(\theta, \boldsymbol{\alpha})$ over Ω. Thus, we should compute the $\frac{\partial\hat{\beta}_\theta(t)}{\partial\theta}$. However, $\hat{\beta}_\theta(t)$ is not differentiable.

In the following, we will show how to obtain an approximation solution of $\hat{\beta}_\theta(t)$. For any $\theta^s \in \Omega$, let $\boldsymbol{\mu}_j^s$ be the j-th eigenvector of $\mathbf{K}_{\theta^s}$, $j = 1, \ldots, t$, then

$$\hat{\beta}_\theta(t) = \frac{\sum_{j=1}^{t}\lambda_j(\mathbf{K}_\theta)}{\mathrm{tr}(\mathbf{K}_\theta)} \geq \frac{\sum_{j=1}^{t}\boldsymbol{\mu}_j^{s\mathrm{T}}\mathbf{K}_\theta\boldsymbol{\mu}_j^s}{\mathrm{tr}(\mathbf{K}_\theta)} =: \tilde{\beta}_\theta(t, \theta^s).$$

Thus, we can use $\arg\max_{\theta\in\Omega}\tilde{Q}(\theta; \boldsymbol{\alpha}) = \lambda\tilde{\beta}_\theta(t, \theta^s) + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}\boldsymbol{\alpha}$ to approximate the optimization (7). Note that

$$\frac{\partial\tilde{Q}(\theta, \boldsymbol{\alpha})}{\partial\theta} = \lambda\frac{\partial\tilde{\beta}_\theta(t, \theta^s)}{\partial\theta} + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\frac{\partial\mathbf{H}}{\partial\theta}\boldsymbol{\alpha}, \tag{8}$$

where $\frac{\partial\tilde{\beta}_\theta(t,\theta^s)}{\partial\theta} = \frac{\sum_{j=1}^{t}\boldsymbol{\mu}_j^{s\mathrm{T}}\frac{\partial\mathbf{K}_\theta}{\partial\theta}\boldsymbol{\mu}_j^s}{\mathrm{tr}(\mathbf{K}_\theta)} - \frac{\tilde{\beta}_\theta(t,\theta^s)}{\mathrm{tr}(\mathbf{K}_\theta)}\frac{\partial\mathrm{tr}(\mathbf{K}_\theta)}{\partial\theta}$. Thus, we can apply the gradient-based algorithm to solve the sub-problem, which is summarized in Algorithm 3.

The time complexity of IFKL is $O\big(k(tn^2 + sn^2 + A)\big)$, where k is the number of iterations of IFKL (Algorithm 2), s is the number of iterations of Algorithm 3, and A the time complexity of FKL (Algorithm 1). In our experiments, we find that IFKL converges very quickly, and k is smaller than 10 on all data sets.

Algorithm 3 Sub-Problem

Initialize: Randomly select any $\theta^0 \in \Omega$, $s = 0$.
repeat
$\quad$Compute the eigenvectors $\boldsymbol{\mu}_1^s, \ldots, \boldsymbol{\mu}_t^s$ of $\mathbf{K}_{\theta^s}$.
$\quad\theta^{s+1} \leftarrow \theta^s + \eta\big(\lambda\frac{\partial\tilde{\beta}_\theta(t,\theta^s)}{\partial\theta} + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\frac{\partial\mathbf{H}}{\partial\theta}\boldsymbol{\alpha}\big)$ (see Equation (8)).
$\quad s \leftarrow s + 1$.
until converged

Experiments

In this section, we will empirically compare our PEP-based finite kernel learning (FKL) and infinite kernel learning (IFKL) with 9 popular finite and infinite kernel learning methods: centered-alignment based MKL with linear combination (CABMKL (linear)) and conic combination (CABMKL (conic)) (Cortes, Mohri, and Rostamizadeh 2010), SimpleMKL (Rakotomamonjy et al. 2008), generalized MKL algorithm (GMKL) (Varma and Babu 2009), nonlinear MKL algorithm with L1-norm (NLMKL ($p = 1$)) and L2-norm (NLMKL ($p = 2$)) (Cortes, Mohri, and Rostamizadeh 2009), group Lasso-based MKL algorithms with L1-norm (GLMKL ($p = 1$)) and L2-norm (GLMKL ($p = 2$)) (Kloft et al. 2011), and the state-of-the-art infinite kernel learning (IKL) (Gehler and Nowozin 2008).

The data sets are 10 publicly available data sets from LIBSVM Data seen in Table 1. For finite kernels, we use the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\tau\|\mathbf{x} - \mathbf{x}'\|_2^2\right)$ and polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}')^d$ as our basic kernels, $\tau \in \{2^i, i = -10, -9 \ldots, 10\}$ and $d \in \{1, 2, \ldots, 20\}$. For

Table 1: The average test accuracies of our IFKL and FKL, and other ones including: CABMKL (linear), CABMKL (conic), SimpleMKL, GMKL, GLMKL ($p = 1$), GLMKL ($p = 2$), NLMKL ($p = 1$), NLMKL ($p = 2$) and IKL. We bold the numbers of the best method, and underline the numbers of the other methods which are not significantly worse than the best one.

Datasets	IFKL (ours)	FKL (ours)	CABMKL(linear)	CABMKL(conic)	SimpleMKL	GMKL
australian	**85.87±0.88**	85.85±0.84	84.41±0.86	85.69±0.92	85.39±0.92	85.40±0.93
a2a	**81.14±0.07**	80.87±0.08	80.12±0.09	74.42±0.07	80.53±0.07	80.56±0.07
diabetes	**76.25±0.99**	76.19±0.94	75.19±1.40	75.55±0.89	75.71±0.85	75.71±0.85
german.numer	**74.88±1.08**	74.32±1.31	71.94±1.69	73.77±1.38	73.99±1.47	73.99±1.48
heart	**82.27±2.92**	82.15±5.50	82.02±3.85	82.05±5.47	81.53±3.31	82.12±3.81
ionosphere	**93.67±1.07**	93.66±1.02	91.57±1.49	91.59±1.43	91.53±1.26	91.52±1.24
liver-disorders	**71.25±3.83**	70.52±5.95	68.50±6.08	69.04±6.32	66.92±6.64	70.52±5.95
sonar	**84.90±6.08**	82.53±5.35	81.99±5.94	81.99±5.94	82.15±5.09	82.15±5.09
splice	**85.48±0.76**	85.46±0.82	84.20±0.90	84.20±0.90	84.31±0.85	84.55±0.81
svmguide3	**82.96±0.71**	82.83±0.72	80.76±0.78	80.34±0.77	82.81±0.71	82.81±0.71

Datasets	NLMKL(p=1)	NLMKL(p=2)	GLMKL(p=1)	GLMKL(p=2)	IKL
australian	85.35±0.90	84.70±0.94	85.46±0.88	85.39±0.92	85.78±0.85
a2a	79.16±0.07	79.07±0.08	80.71±0.07	80.82±0.08	80.84±0.08
diabetes	74.57±0.80	73.94±1.00	75.81±0.84	75.77±0.78	76.02±1.00
german.numer	73.55±1.04	72.45±1.22	74.07±1.53	73.77±1.38	74.83±1.02
heart	81.51±3.00	80.17±5.45	82.10±3.63	82.07±3.69	82.12±3.81
ionosphere	91.53±1.66	91.65±1.54	91.50±1.37	91.08±0.98	93.63±1.05
liver-disorders	68.61±5.47	68.55±4.30	64.57±3.43	66.97±6.24	70.58±5.29
sonar	81.25±6.44	81.89±6.27	82.24±5.13	81.86±6.43	83.04±6.50
splice	83.45±1.14	83.69±0.99	85.40±0.79	53.29±1.05	85.46±0.82
svmguide3	82.19±0.71	82.53±0.68	82.75±0.72	80.23±1.29	82.83±0.72

infinite kernels. we use Gaussian kernel and polynomial kernel with continuously parameterized sets, $\tau \in [2^{-10}, 2^{10}]$ and $d \in [1, 20]$. The regularization parameter C of all algorithms is set to be 1. The other parameters for the compared algorithms follow the same experimental setting in their papers. The parameters $\lambda \in \{2^i, i = -5, \ldots, 5\}$ and $t \in \{2^i, i = 1, \ldots, 4\}$ of our algorithms are determined by 3-fold cross-validation on training set. For each data set, we run all methods 30 times with randomly selected 50% of all data for training and the other 50% for testing. We use t-test to describe the statistical discrepancy of different methods. All statements of statistical significance in the remainder refer to a 95% level of significance.

The average test accuracies are reported in Table 1, which can be summarized as follows: 1) Our IFKL gives the best results on all data sets and is significantly better than the compared finite kernel selection methods (CABMKL, SimpleMKL, GMKL and NLMKL) on nearly all data sets. Thus, it implicates that learning the kernel with PEP over a prescribed set of continuously parameterized basic kernels can guarantee good generalization performance. 2) IFKL is significantly better than the state-of-the-art infinite kernel learning (IKL) on 4 out of 10 data sets, and FKL is significantly better than the compared finite kernel learning methods (CABMKL, SimpleMKL, GMKL and NLMKL) on most of data sets; 3) FKL gives comparable results to the IFKL; 4) The infinite kernel learning methods (IFKS and IKL) are usually better than the compared finite kernel learning methods (CABMKL, SimpleMKL, GMKL and NLMKL), which manifests the effectiveness of the use of infinite kernels. The above results show that the use of PEP can significantly improve the performance of kernel learning algorithms in both finite and infinite kernels.

Conclusion

In this paper, we have created a new kernel learning method based on the principal eigenvalue proportion (PEP) measure of generalization errors, which can learn the optimal kernel with fast convergence rate over a convex hull of possibly infinite basic kernels. Using the PEP measure, we have derived sharper generalization bounds with fast convergence rates, which improve the results of generalization bounds for most of kernel learning algorithms. Exploiting the derived generalization bounds, we have designed two effective kernel learning algorithms with statistical guarantees and fast convergence rates, which outperform the state-of-the-art kernel learning methods.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grant No. 61602467, No. 61303056 and Youth Innovation Promotion Association, CAS, No.2016146.

References

Anguita, D.; Ghio, A.; Oneto, L.; and Ridella, S. 2012. In-sample and out of sample model selection and error esti-

mation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems* 23(9):1390–1406.

Argyriou, A.; Hauser, R.; Micchelli, C. A.; and Pontil, M. 2006. A DC-programming algorithm for kernel selection. In *Proceedings of the 23rd international conference on Machine learning (ICML 2006)*, 41–48. ACM.

Argyriou, A.; Micchelli, C. A.; and Pontil, M. 2005. Learning convex combinations of continuously parameterized basic kernels. In *Proceeding of the 18th Annual Conference on Learning Theory (COLT 2005)*. Springer. 338–352.

Bach, F.; Lanckriet, G.; and Jordan, M. 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, 41–48.

Bartlett, P. L., and Mendelson, S. 2002. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research* 3:463–482.

Bartlett, P. L.; Bousquet, O.; and Mendelson, S. 2005. Local Rademacher complexities. *The Annals of Statistics* 33(4):1497–1537.

Bousquet, O., and Elisseeff, A. 2002. Stability and generalization. *Journal of Machine Learning Research* 2:499–526.

Chapelle, O.; Vapnik, V.; Bousquet, O.; and Mukherjee, S. 2002. Choosing multiple parameters for support vector machines. *Machine Learning* 46(1-3):131–159.

Cortes, C.; Kloft, M.; and Mohri, M. 2013. Learning kernels using local Rademacher complexity. In *Advances in Neural Information Processing Systems 25 (NIPS 2013)*. MIT Press. 2760–2768.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2009. Learning non-linear combinations of kernels. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 396–404.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2010. Two-stage learning kernel algorithms. In *Proceedings of the 27th Conference on Machine Learning (ICML 2010)*, 239–246.

Gehler, P., and Nowozin, S. 2008. Infinite kernel learning. In *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*.

Ghiasi-Shirazi, K.; Safabakhsh, R.; and Shamsi, M. 2010. Learning translation invariant kernels for classification. *Journal of Machine Learning Research* 11:1353–1390.

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

Kloft, M., and Blanchard, G. 2012. On the convergence rate of ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research* 13(1):2465–2502.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; Laskov, P.; Müller, K.-R.; and Zien, A. 2009. Efficient and accurate ℓ_p-norm multiple kernel learning. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 997–1005.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; and Zien, A. 2011. ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research* 12:953–997.

Koltchinskii, V., and Panchenko, D. 2000. *Rademacher processes and bounding the risk of function learning*. Springer.

Koltchinskii, V. 2006. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics* 34(6):2593–2656.

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

Liu, Y., and Liao, S. 2015. Eigenvalues ratio for kernel selection of kernel methods. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*.

Liu, Y.; Jiang, S.; and Liao, S. 2014. Efficient approximation of cross-validation for kernel methods using Bouligand influence function. In *Proceedings of The 31st International Conference on Machine Learning (ICML 2014 (1))*, 324–332.

Liu, Y.; Liao, S.; and Hou, Y. 2011. Learning kernels with upper bounds of leave-one-out error. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, 2205–2208.

Lugosi, G., and Wegkamp, M. 2004. Complexity regularization via localized random penalties. *The Annals of Statistics* 32:1679–1697.

Mendelson, S. 2003. On the performance of kernel classes. *Journal of Machine Learning Research* 4:759–771.

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

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

Rakotomamonjy, A.; Bach, F.; Canu, S.; and Grandvalet, Y. 2008. SimpleMKL. *Journal of Machine Learning Research* 9:2491–2521.

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

Sonnenburg, S.; Rätsch, G.; and Schäfer, C. 2006. A general and efficient multiple kernel learning algorithm. In *Advances in Neural Information Processing Systems 18 (NIPS 2006)*.

Srebro, N.; Sridharan, K.; and Tewari, A. 2010. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 22*. MIT Press. 2199–2207.

Vapnik, V. 2000. *The nature of statistical learning theory*. Springer Verlag.

Varma, M., and Babu, B. R. 2009. More generality in efficient multiple kernel learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*, 134–141.

Zhang, T. 2002. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research* 2:527–550.