Automated Spectral Kernel Learning

Jian Li,[1,2] **Yong Liu,**[*1,2] **Weiping Wang**[1]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{lijian9026, liuyong, wangweiping}@iie.ac.cn

Abstract

The generalization performance of kernel methods is largely determined by the kernel, but spectral representations of stationary kernels are both input-independent and output-independent, which limits their applications on complicated tasks. In this paper, we propose an efficient learning framework that incorporates the process of finding suitable kernels and model training. Using non-stationary spectral kernels and backpropagation w.r.t. the objective, we obtain favorable spectral representations that depends on both inputs and outputs. Further, based on Rademacher complexity, we derive data-dependent generalization error bounds, where we investigate the effect of those factors and introduce regularization terms to improve the performance. Extensive experimental results validate the effectiveness of the proposed algorithm and coincide with our theoretical findings.

Introduction

Kernel methods have achieved great success in many conventional domains over past decades, while they show relatively inferior performance on complicated tasks nowadays. The fundamental limitation of common kernels has been revealed that they are both *stationary* and *monotony* (Bengio, Delalleau, and Roux 2006). The *stationary* property shows that stationary kernels only depend on the distance $\|x - x'\|$ while are free from the input x itself. The *monotony* property indicates that values of stationary kernels decrease over the distance, ignoring the long-range interdependence.

Spectral approaches were developed to fully characterize stationary kernels with concise representation forms, such as sparse spectrum kernels (Quiñonero-Candela et al. 2010), sparse mixture kernels (Wilson and Adams 2013) and random Fourier features methods to handle with large scale settings (Rahimi and Recht 2007; Le, Sarlós, and Smola 2013; Li, Liu, and Wang 2019). With sound theoretical guarantees, namely Bochner's theorem (Rudin 1962; Stein 2012), spectral kernels are constructed from the inverse Fourier transform in the frequency domain. Although approximate spectral representation provides an efficient approach for stationary spectral kernels, the performance of
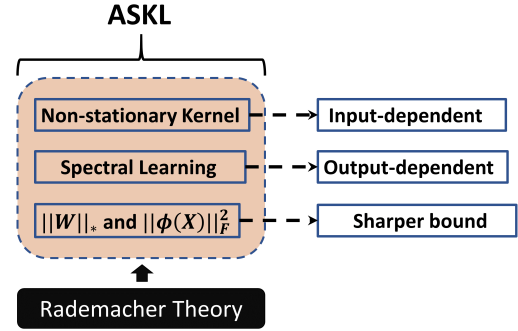
Figure 1: The overview of ASKL.

random features is limited because stationary kernels are *input-independent* and *output-independent*. Yaglom's theorem, rather than Bochner's theorem, provides more general forms which encompass both stationary and non-stationary kernels via inverse Fourier transform (Yaglom 1987; Samo and Roberts 2015). Recently, due to its general and concise spectral statements, non-stationary kernels have been applied to Gaussian process regression (Remes, Heinonen, and Kaski 2017; Sun et al. 2019).

Hyper parameters for kernels determine the performance of kernel methods. Meanwhile, kernel selection approaches have been well-studied (Liu and Liao 2014; Li et al. 2017; Ding et al. 2018; Liu et al. 2018; 2019). However, due to the separation of kernel selection and model training, those techniques are inefficient and lead the undesirable performance.

In this paper, to achieve better performance ability for kernel methods, we propose an efficient algorithm, namely Automated Spectral Kernel Learning (ASKL), learning suitable kernels and model weights together. A brief overview is illustrated in Figure 1. On the algorithmic front, ASKL consists of: (1) non-stationary kernels to obtain input-dependent features, (2) backpropagation w.r.t the objective to make features output-dependent, (3) regularization terms to achieve sharper generalization error bounds. On the theoretical front, the theoretical underpinning is Rademacher complexity theory, which indicates how the feature mappings affect the performance and suggests ways to refine the algorithm.

Background

In ordinary supervised learning settings, training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n\}$ are drawn i.i.d. from a fixed but unknown distribution ρ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ is the input space and $\mathcal{Y} \subseteq \mathbb{R}^K$ is the output space in single-valued ($K = 1$) or vector-valued ($K > 1$) forms. The goal is to learn an estimator $f : \mathcal{X} \to \mathcal{Y}$, which outputs K predictive labels. We define a standard hypothesis space for kernel methods

$$\mathcal{H} = \left\{ \boldsymbol{x} \to f(\boldsymbol{x}) = \boldsymbol{W}^T \phi(\boldsymbol{x}) \right\},$$

where $\boldsymbol{W} \in \mathbb{R}^{D \times K}$ is the model weight, $\phi(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^D$ is a nonlinear feature mapping. For kernel methods, $\phi(\boldsymbol{x})$ is an implicit feature mapping associated with a Mercer kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$. To improve the computational efficiency but also retain favorable statistical properties, random Fourier features were proposed to approximate kernel with explicit feature mappings $\phi(\boldsymbol{x})$ via $k(\boldsymbol{x}, \boldsymbol{x}') \approx \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ (Rahimi and Recht 2007).

In statistical learning theory, the supervised learning problem is to minimize the expected loss on $\mathcal{X} \times \mathcal{Y}$

$$\inf_{f \in \mathcal{H}} \mathcal{E}(f), \quad \mathcal{E}(f) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(\boldsymbol{x}), \boldsymbol{y}) \, d\rho(\boldsymbol{x}, \boldsymbol{y}), \quad (1)$$

where ℓ is a loss function related to specific tasks.

Stationary Kernels

The connection between the stationary kernel $k(\tau)$ and its spectral density $s(\boldsymbol{\omega})$ is revealed in Bochner's theorem via inverse Fourier transform.

Theorem 1 (Bochner's theorem). *A stationary continuous kernel $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x}')$ on $\mathbb{R}^d$ is positive definite if and only if it can be represented as*

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathbb{R}^d} e^{i \boldsymbol{\omega}^T (\boldsymbol{x} - \boldsymbol{x}')} s(\boldsymbol{\omega}) d\boldsymbol{\omega}, \quad (2)$$

where $s(\boldsymbol{\omega})$ is a non-negative measure.

The spectral density $s(\boldsymbol{\omega})$ is the probability density of the corresponding kernel. From the inverse Fourier transform, we find that spectral kernels are highly relevant to the probability measure $s(\boldsymbol{\omega})$. E.g. $s(\boldsymbol{\omega})$ for Gaussian kernels with the width σ correspond to Gaussian distribution $\mathcal{N}(0, 1/\sigma)$.

Non-Stationary Kernels

While the *stationary* kernels ignore input-dependent information and long-range correlations, non-stationary kernels alleviate these restrictions because they depend on the inputs themselves (Samo and Roberts 2015).

Theorem 2 (Yaglom's theorem). *A general kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ is positive definite on $\mathbb{R}^d$ is positive define if and only if it admits the form*

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathbb{R}^d \times \mathbb{R}^d} e^{i(\boldsymbol{\omega}^T \boldsymbol{x} - \boldsymbol{\omega}'^T \boldsymbol{x}')} \mu(d\boldsymbol{\omega}, d\boldsymbol{\omega}'), \quad (3)$$

where $\mu(d\boldsymbol{\omega}, d\boldsymbol{\omega}')$ is the Lebesgue-Stieltjes measure associated to some positive semi-definite (PSD) spectral density function $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ with bounded variations.

When μ is concentrated on the diagonal $\boldsymbol{\omega} = \boldsymbol{\omega}'$, the spectral characterization of stationary kernels in the Bochner's theorem is recovered. $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ is a joint probability density.
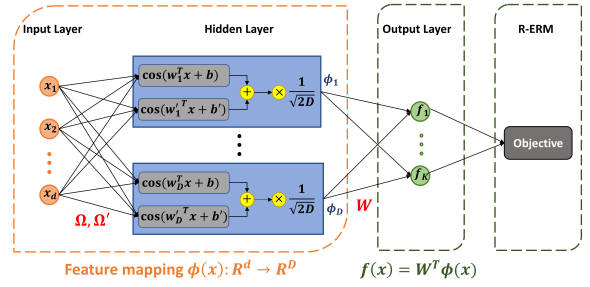

Figure 2: The architecture of learning framework

Automated Spectral Kernel Learning

In this section, we devise a learning framework for arbitrary kernel-based supervised applications:

- We present the learning framework with the minimization objective, integrating empirical risk minimization (ERM) with regularizers on feature mappings and model weights.
- Spectral representation for non-stationary spectral kernels based on Yaglom's theorem is conducted.
- We apply first-order gradient approaches to solve the minimization objective. We update frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ together with model weights $\boldsymbol{W}$ via backpropagation.

Learning Framework

The minimization of the expected loss (1) is hard to estimate in practical problems. In this paper, we put two additional regularization terms into the ERM, thus the empirical objective $\widehat{\mathcal{E}}(\boldsymbol{W}, \boldsymbol{\Omega}, \boldsymbol{\Omega}')$ becomes

$$\arg\min_{\boldsymbol{W}, \boldsymbol{\Omega}, \boldsymbol{\Omega}'} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(f(\boldsymbol{x}_i), \boldsymbol{y}_i)}_{g(\boldsymbol{W})} + \lambda_1 \|\boldsymbol{W}\|_* + \lambda_2 \|\phi(\boldsymbol{X})\|_F^2, \quad (4)$$

where both feature mappings $\phi(\boldsymbol{X}) \in \mathbb{R}^{D \times n}$ on all data and $f(\boldsymbol{x}_i) = \boldsymbol{W}^T \phi(\boldsymbol{x}_i) \in \mathbb{R}^D$ use the non-stationary spectral representation $\phi(\cdot)$, which is presented as

$$\phi(\boldsymbol{x}) = \frac{1}{\sqrt{2D}} \left[\cos(\boldsymbol{\Omega}^T \boldsymbol{x} + \boldsymbol{b}) + \cos(\boldsymbol{\Omega}'^T \boldsymbol{x} + \boldsymbol{b}') \right].$$

The trace norm $\|\boldsymbol{W}\|_*$ and the squared Frobenius norm $\|\phi(\boldsymbol{X})\|_F^2$ exerts constraints on updating model weights $\boldsymbol{W}$ and frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$, respectively. Those two regularization terms, especially the norm on feature mappings, are rarely used in the R-ERM, leading to the better generalization performance as proven in next section.

In the objective (4), we update model weights $\boldsymbol{W}$ and frequency pairs $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ to learn feature mappings both input-dependent (non-stationary kernels) and output-dependent (backpropagation towards the objective). As shown in Figure 2, the architecture can be regarded as a single hidden layer neural network with cosine as activation. The spectral density surface $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ (joint probability density), determining the performance of spectral kernels, is optimized by updating frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ via backpropagation (Rumelhart et al. 1988). In this structure, only $\boldsymbol{W}$ and $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ are trainable and optimized towards the objective.

Non-Stationary Spectral Kernels Representation

According to Yaglom's theorem, to produce a positive semi-definite (PSD) kernel, the spectral density needs to be a PSD function. In order to construct a PSD spectral density $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$, we include symmetries $s(\boldsymbol{\omega}, \boldsymbol{\omega}') = s(\boldsymbol{\omega}', \boldsymbol{\omega})$ and sufficient diagonal components $s(\boldsymbol{\omega}, \boldsymbol{\omega})$ and $s(\boldsymbol{\omega}', \boldsymbol{\omega}')$. We integrate exponential components and the corresponding integrated spectral density $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ and get

$$\mathcal{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{4}\big[e^{i(\boldsymbol{\omega}^T \boldsymbol{x} - \boldsymbol{\omega}'^T \boldsymbol{x}')} + e^{i(\boldsymbol{\omega}'^T \boldsymbol{x} - \boldsymbol{\omega}^T \boldsymbol{x}')}$$
$$+ e^{i(\boldsymbol{\omega}^T \boldsymbol{x} - \boldsymbol{\omega}^T \boldsymbol{x}')} + e^{i(\boldsymbol{\omega}'^T \boldsymbol{x} - \boldsymbol{\omega}'^T \boldsymbol{x}')}\big].$$

The PSD kernel can be rewritten as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') \mu(d\boldsymbol{\omega}, d\boldsymbol{\omega}'). \quad (5)$$

where $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ is the spectral density surface. Similar spectral representation forms for non-stationary kernels are also used in (Samo and Roberts 2015; Remes, Heinonen, and Kaski 2017). When $\boldsymbol{\omega} = \boldsymbol{\omega}'$, non-stationary kernel in (5) degrades into stationary kernel as in Bochner's theorem (2) and the probability density function becomes univariate $s(\boldsymbol{\omega})$.

In the stationary cases, random Fourier features are used to approximate stationary kernels (Rahimi and Recht 2007). Similarly, in the non-stationary cases, we can approximate (5) with Monte Carlo random sampling

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') \mu(d\boldsymbol{\omega}, d\boldsymbol{\omega}')$$
$$= \mathbb{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}' \sim s} \big[\mathcal{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') \big]$$
$$= \mathbb{E}_{\boldsymbol{\omega}, \boldsymbol{\omega}' \sim s} \frac{1}{4}\big[\cos(\boldsymbol{\omega}^T \boldsymbol{x} - \boldsymbol{\omega}'^T \boldsymbol{x}') + \cos(\boldsymbol{\omega}'^T \boldsymbol{x} - \boldsymbol{\omega}^T \boldsymbol{x}')$$
$$+ \cos(\boldsymbol{\omega}^T \boldsymbol{x} - \boldsymbol{\omega}^T \boldsymbol{x}') + \cos(\boldsymbol{\omega}'^T \boldsymbol{x} - \boldsymbol{\omega}'^T \boldsymbol{x}')\big]$$
$$\approx \frac{1}{4D} \sum_{i=1}^{D} \big[\cos(\boldsymbol{\omega}_i^T \boldsymbol{x} - \boldsymbol{\omega}_i'^T \boldsymbol{x}') + \cos(\boldsymbol{\omega}_i'^T \boldsymbol{x} - \boldsymbol{\omega}_i^T \boldsymbol{x}')$$
$$+ \cos(\boldsymbol{\omega}_i^T \boldsymbol{x} - \boldsymbol{\omega}_i^T \boldsymbol{x}') + \cos(\boldsymbol{\omega}_i'^T \boldsymbol{x} - \boldsymbol{\omega}_i'^T \boldsymbol{x}')\big]$$
$$= \langle \psi(\boldsymbol{x}), \psi(\boldsymbol{x}') \rangle$$

where $(\boldsymbol{\omega}_i, \boldsymbol{\omega}_i')_{i=1}^{D} \overset{\text{i.i.d.}}{\sim} s(\boldsymbol{\omega}, \boldsymbol{\omega}')$, D is the number of features and random Fourier feature mapping of spectral kernel is

$$\psi(\boldsymbol{x}) = \frac{1}{\sqrt{4D}} \begin{bmatrix} \cos(\boldsymbol{\Omega}^T \boldsymbol{x}) + \cos(\boldsymbol{\Omega}'^T \boldsymbol{x}) \\ \sin(\boldsymbol{\Omega}^T \boldsymbol{x}) + \sin(\boldsymbol{\Omega}'^T \boldsymbol{x}) \end{bmatrix},$$

where features mapping are $\mathbb{R}^d \to \mathbb{R}^{2D}$. To alleviate computational costs, we use the following mapping $\mathbb{R}^d \to \mathbb{R}^D$

$$\phi(\boldsymbol{x}) = \frac{1}{\sqrt{2D}} \big[\cos(\boldsymbol{\Omega}^T \boldsymbol{x} + \boldsymbol{b}) + \cos(\boldsymbol{\Omega}'^T \boldsymbol{x} + \boldsymbol{b}') \big], \quad (6)$$

where frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}' \in \mathbb{R}^{d \times D}$ are integrated with $(\boldsymbol{\omega}_i, \boldsymbol{\omega}_i')_{i=1}^{D}$ that $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_D]$ and $\boldsymbol{\Omega}' = [\boldsymbol{\omega}_1', \cdots, \boldsymbol{\omega}_D']$. The phase vectors $\boldsymbol{b}, \boldsymbol{b}'$ are drawn uniformly from $[0, 2\pi]^D$. In fact, the spectral kernels induced by $\psi(\boldsymbol{x})$ and $\phi(\boldsymbol{x})$ are equivalent in the expectation manner.

Remark 1. *Equation (6) provides explicit feature mappings to approximate non-stationary kernels, where suitable frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ are the key to obtain a favorable performance. Standard random Fourier features generate frequency matrices via samples from the **assigned** spectral density $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$, e.g frequency matrices correspond to random Gaussian matrix for Gaussian kernels. In the proposed algorithm* ASKL, *the frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ are jointly **optimized** together with model weights $\boldsymbol{W}$ during training.*

Update Trainable Matrices

As shown in the previous section, non-stationary spectral kernels are input-dependent. Further, we propose a general learning framework, namely Automated Spectral Kernel Learning (ASKL), that **optimizes** frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ for non-stationary kernels according to the objective via backpropagation. Therefore, the learned feature mappings are both *input-dependent* and *output-dependent*.

The learning frame ASKL can be solved by first-order gradient descent methods, such as stochastic gradient descent (SGD) and its variants Adadelta (Zeiler 2012) and Adam (Kingma and Ba 2014). Using the backpropagation, we derive how to update estimator weights $\boldsymbol{W}$ and frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ in the following analyses. More generally, we consider mini-batch gradient descent where we use m examples in each iteration. Specifically, full gradient descent is the special case $m = n$ where all examples are used and stochastic gradient descent (SGD) corresponds to the special case $m = 1$ where only one example is used.

Update $\boldsymbol{W}$ in Proximal Gradient Approach To minimize the objective (4), we use first-order gradient descent algorithms to update $\boldsymbol{W}$ in the direction of negative gradient. The updates of gradient of $\boldsymbol{W}$ depend on empirical loss and trace norm in (4), but trace norm is nondifferentiable on many points for each dimension (unlike hinge loss and Relu are nondifferentiable only on one point), thus the derivative/subgradient of the trace norm cannot be applied to standard gradient approaches. We employ singular value thresholding (SVT) to solve the minimization of trace norm with proximal gradient descent (Cai, Candès, and Shen 2010).

We simply the updates of $\boldsymbol{W}$ in two steps and put detailed deduction process in the Appendix section:

1) Update $\boldsymbol{W}$ with SGD on the empirical loss
$$\boldsymbol{Q} = \boldsymbol{W}^t - \eta \nabla g(\boldsymbol{W}^t), \quad (7)$$
   where η is the learning rate and the gradient of empirical loss on m-batch examples is
$$\nabla g(\boldsymbol{W}) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial \boldsymbol{W}}$$
$$= \frac{1}{m} \sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \cdot \left[\frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial f(\boldsymbol{x}_i)} \right]^T \in \mathbb{R}^{D \times K},$$
   and $\boldsymbol{Q}$ is an intermediate matrix and m examples are used.

2) Update $\boldsymbol{W}$ with SVT on the trace norm
$$\boldsymbol{W}^{t+1} = \boldsymbol{U} \text{diag}\big(\{\sigma_j - \lambda_1 \eta\}_+ \big) \boldsymbol{V}^T, \quad (8)$$
   where $\boldsymbol{Q} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$ is the singular values decomposition, $\boldsymbol{\Sigma}$ is the diagonal $\text{diag}(\{\sigma_j\}_{1 \le i \le r})$ and r is the rank of $\boldsymbol{Q}$.

Update $\mathbf{\Omega}, \mathbf{\Omega}'$ Using the chain rule for computing the derivative, the derivative of objective only depends on empirical loss $\ell(f(\boldsymbol{x}_i), \boldsymbol{y})$ in terms of $\mathbf{\Omega}, \mathbf{\Omega}'$. Both empirical risk $\ell(f(\boldsymbol{x}_i), \boldsymbol{y})$ and squared Frobenius norm $\|\phi(\boldsymbol{X})\|_F^2$ are differentiable with respect to $\mathbf{\Omega}, \mathbf{\Omega}'$. We derive the gradient of the minimization objective (4) w.r.t. $\mathbf{\Omega}$ as an example.

$$\mathbf{\Omega}^{t+1} = \mathbf{\Omega}^t - \eta \nabla \widehat{\mathcal{E}}(\mathbf{\Omega}^t) \qquad (9)$$

where the gradients w.r.t $\mathbf{\Omega}$ are

$$\nabla \widehat{\mathcal{E}}(\mathbf{\Omega}) = \frac{1}{m} \sum_{i=1}^m \frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial \mathbf{\Omega}} + \lambda_2 \frac{\partial \|\phi(\boldsymbol{X})\|_F^2}{\partial \mathbf{\Omega}},$$

$$\frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial \mathbf{\Omega}} = \boldsymbol{x}_i \cdot \left[\boldsymbol{D} \cdot \boldsymbol{W} \cdot \frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial f(\boldsymbol{x}_i)} \right]^T,$$

$$\frac{\partial \|\phi(\boldsymbol{X})\|_F^2}{\partial \mathbf{\Omega}} = \frac{1}{m} \sum_{i=1}^m 2 \boldsymbol{x}_i \cdot \phi(\boldsymbol{x}_i)^T \cdot \boldsymbol{D}$$

and $\boldsymbol{D}$ is a diagonal matrix in $D \times D$ size filled with a vector

$$\boldsymbol{D} = \mathrm{diag} \left\{ \frac{-1/\sqrt{2D}}{\sin(\mathbf{\Omega}^T \boldsymbol{x}_i + b)} \right\}_{D \times D}.$$

Specific Loss Functions For gradients in (7) and (9), only gradients w.r.t. the loss function $\frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial f(\boldsymbol{x}_i)}$ are uncertain.

- Hinge Loss for Classification Problems.
  Let the label $\boldsymbol{y} = [0, \cdots, 0, 1, 0, \cdots, 0]^T$ only one element (its category) is not zero. The hinge loss is defined as $\ell(f(\boldsymbol{x}_i), \boldsymbol{y}) = |1 - (\boldsymbol{y}^T f(\boldsymbol{x}_i) - \max_{\boldsymbol{y}' \neq \boldsymbol{y}} \boldsymbol{y}'^T f(\boldsymbol{x}_i))|_+$. The sub-gradient of hinge loss w.r.t the estimator is

$$\frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial f(\boldsymbol{x}_i)} = \begin{cases} \boldsymbol{0}, & \boldsymbol{y}^T f(\boldsymbol{x}_i) - \max_{\boldsymbol{y}' \neq \boldsymbol{y}} \boldsymbol{y}'^T f(\boldsymbol{x}_i) \geq 1, \\ \boldsymbol{y}' - \boldsymbol{y}, & \text{else.} \end{cases}$$

- Squared Loss for Regression Problems.
  Let $\boldsymbol{y}$ be the K-size vector-valued label where $K > 1$ for multi-label regression and $K = 1$ for univariate regression. The Squared loss function is $\ell(f(\boldsymbol{x}_i), \boldsymbol{y}) = \|f(\boldsymbol{x}_i) - \boldsymbol{y}\|_2^2$. Then, the gradient of squared loss is

$$\frac{\partial \ell(f(\boldsymbol{x}_i), \boldsymbol{y})}{\partial f(\boldsymbol{x}_i)} = 2(f(\boldsymbol{x}_i) - \boldsymbol{y}).$$

Theoretical Guarantee

In this section, we study the generalization performance for the proposed algorithm ASKL. A data-dependent excess risk bound is derived, and then we explore how the input-dependent and output-dependent feature mappings affect the statistical performance and how to improve the algorithm by utilizing additional regularization terms.

Definition 1. *The empirical Rademacher complexity of hypothesis space $\mathcal{H}$ is*

$$\widehat{\mathcal{R}}(\mathcal{H}) = \frac{1}{n} \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \sum_{i=1}^n \sum_{k=1}^K \epsilon_{ik} f_k(\boldsymbol{x}_i) \right],$$

where $f_k(\boldsymbol{x}_i)$ is the k-th value of the estimator $f(\boldsymbol{x}_i)$ with K outputs and $\epsilon_{ik}s$ are $n \times K$ independent Rademacher variables with equal probabilities $\mathbb{P}(\epsilon_{ik} = +1) = \mathbb{P}(\epsilon_{ik} = -1) = 1/2$. Its deterministic estimate is $\mathcal{R}(\mathcal{H}) = \mathbb{E}\,\widehat{\mathcal{R}}(\mathcal{H})$.

Theorem 3 (Excess Risk Bound). *Assume that $B = \sup_{f \in \mathcal{H}} \|\boldsymbol{W}\|_* < \infty$ and assume the loss function ℓ is L-Lipschitz for $\mathbb{R}^K$ equaipped with the 2-norm, with probability at least $1 - \delta$, the following excess risk bound holds*

$$\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f^*) \leq 4\sqrt{2} L \widehat{\mathcal{R}}(\mathcal{H}) + \mathcal{O}\left(\sqrt{\frac{\log 1/\delta}{n}} \right), \quad (10)$$

where $f^* \in \mathcal{H}$ is the most accurate estimator in the hypothesis space, $\widehat{f}_n$ is the empirical estimator and

$$\widehat{\mathcal{R}}(\mathcal{H}) \leq \frac{B}{n} \sqrt{K \sum_{i=1}^n \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i) \rangle}$$

$$= \frac{B}{n} \sqrt{\frac{K}{D} \sum_{i=1}^n \sum_{j=1}^D \frac{1}{2} \Big[\cos \big((\boldsymbol{\omega}_j - \boldsymbol{\omega}'_j)^T \boldsymbol{x}_i \big) + 1 \Big]}. \quad (11)$$

The proof is given in the Appendix. The error bounds depend on Rademacher complexity term. Due to $\cos \big((\boldsymbol{\omega}_j - \boldsymbol{\omega}'_j)^T \boldsymbol{x}_i \big) \leq 1$, Rademacher complexity in (11) is naturally bounded by $\widehat{\mathcal{R}}(\mathcal{H}) \leq B\sqrt{K/n}$, thus the convergence rate is

$$\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f^*) \leq \mathcal{O}\left(B \sqrt{\frac{K}{n}} \right). \quad (12)$$

Based on the theoretical findings, we make some technical comments to understand how the factors, including feature mappings and regularization terms, affect the generalization performance and suggest ways to refine the algorithm:

- **Influence of Non-Stationary Kernels.** As mentioned in the above section, non-stationary kernels depend on inputs themselves instead of the distance between inputs. We explore the impact of non-stationary kernels based on Rademacher complexity in (11), which depends on the trace $\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i)$. For stationary kernels (shift-invariant kernels), the spectral representation holds the diagonals as $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = \cos(\boldsymbol{\omega}^T(\boldsymbol{x}_i - \boldsymbol{x}_i)) = 1$, thus the trace of kernel matrix $\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i) = n$, which corresponds to the worst cases. While for non-stationary kernels, $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = \cos((\boldsymbol{\omega} - \boldsymbol{\omega}')^T \boldsymbol{x}_i) \in [-1, 1]$. For most instances $\boldsymbol{x}_i$, the diagonals are $k(\boldsymbol{x}_i, \boldsymbol{x}_i) < 1$, so the trace $\sum_{i=1}^n k(\boldsymbol{x}_i, \boldsymbol{x}_i) \ll n$. Therefore, error bounds of non-stationary spectral kernels are much tighter than bounds of stationary kernels, where non-stationary kernels achieve the better generalization performance.

- **Influence of Spectral Learning.** If frequency matrices $\mathbf{\Omega}, \mathbf{\Omega}'$ are just assigned according to specific spectral density $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$, the non-stationary kernels are input-dependent but output-independent. By dynamically optimizing frequency matrices $\mathbf{\Omega}, \mathbf{\Omega}'$ towards the objective, we acquire more powerful feature representations. Then, the spectral measure $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ of optimal kernels can be estimated from optimized frequency matrices $\mathbf{\Omega}, \mathbf{\Omega}'$. The learned feature mappings are dependent on both inputs and outputs, offering better feature representations.

- **Using the Trace Norm $\|\boldsymbol{W}\|_*$ as a Regularizer.** The convergence rate $B = \sup_{f \in \mathcal{H}} \|\boldsymbol{W}\|_* < \infty$ is also dependent on a constant B, that is the supremum of trace

norm $\|\boldsymbol{W}\|_*$ in terms of the specific hypothesis space. As a result, the minimization of trace norm $\|\boldsymbol{W}\|_*$ is useful to reduce B and obtain better error bounds. Based on Rademacher complexity theory, the use of trace norm $\|\boldsymbol{W}\|_*$ instead of squared Frobenius norm $\|\boldsymbol{W}\|_F^2$ as regularization terms was also explored for linear estimators in (Xu et al. 2016; Li et al. 2019).

- **Using Squared Frobenius Norm** $\|\phi(\boldsymbol{X})\|_F^2$ **as a Regularizer.** From (10), Rademacher complexity is bounded by the trace of the kernel and it can be written as

$$\sum_{i=1}^{n} \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i) \rangle = \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i)\|_2^2 = \|\phi(\boldsymbol{X})\|_F^2.$$

In standard theoretical learning for Rademacher complexity, the kernel trace cannot be used to improve learning algorithm because feature mappings are constants for specific inputs. For example, all diagonals are $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$ for shift-variant kernels thus the trace is the number of examples n. Note that, instead of assigned kernel parameters, our algorithm automatically learns optimal spectral density for kernels, thus the trace of the kernel is no longer a constant. Meanwhile, we use it as two regularization terms to improve the performance (avoid overfitting).

Experiments

In this section, compared with other algorithms, we evaluate the empirical behavior of our proposed algorithm ASKL on several benchmark datasets to demonstrate the effects of factors used in our algorithm, including the non-stationary spectral kernel, updating spectral density with backpropagation and additional regularization terms.

Experimental Setup

Based on random Fourier features, both our algorithm ASKL and compared methods apply nonlinear feature mapping into a fixed D-dimensional feature space where we set $D = 2000$. We apply Gaussian kernels as basic kernels because Gaussian kernels succeed in many types of data by mapping inputs into infinite-dimensional space, of which frequency matrices $\boldsymbol{\Omega}, \boldsymbol{\Omega}'$ are i.i.d. drawn from Gaussian distributions $\mathcal{N}(0, \sigma^2)$. The generalization ability of algorithms are highly dependent on different parameters on λ_1, λ_2 and Gaussian kernel parameter σ for spectral kernels. For fair comparisons, we tune those parameters to achieve optimal empirical performance for all algorithms on all dataset, by using 5-folds cross-validation and grid search over parameters candidate sets. Regularization parameters are selected in $\lambda_1, \lambda_2 \in \{10^{-10}, 10^{-9}, \cdots, 10^{-1}\}$ and Gaussian kernel parameter σ is selected from candidate set $\sigma \in \{2^{-10}, \cdots, 2^{10}\}$. Accuracy and mean squared error (MSE) are used to evaluate performance for classification and regression, respectively. We implement all algorithms based on Pytorch and use Adam as optimizer with 32 examples in a mini-batch to solve the minimization problem.

Compared Algorithms   To assess the effectiveness of factors used in our algorithm, we compare the proposed algorithm with several relevant algorithms. As shown in Table 1,

Methods	Kernel	Density	Regularizer
SK	Stationary	Assigned	$\|\boldsymbol{W}\|_F^2$
NSK	Non-stationary	Assigned	$\|\boldsymbol{W}\|_F^2$
SKL	Stationary	Learned	$\|\boldsymbol{W}\|_F^2$
NSKL	Non-stationary	Learned	$\|\boldsymbol{W}\|_F^2$
ASKL	Non-stationary	Learned	$\|\boldsymbol{W}\|_*, \|\phi(\boldsymbol{X})\|_F^2$

Table 1: Compared algorithms.

compared methods are special cases of ASKL:
(1) **SK** (Rahimi and Recht 2007): known as random Fourier features for the stationary spectral kernel. This approach directly assigned the spectral density for shift-invariant kernels and uses the regularizer $\|\boldsymbol{W}\|_F^2$ on model weights .
(2) **NSK** (Samo and Roberts 2015): Similar to **SK** but it uses spectral representation for non-stationary kernel which was introduced in (6) with assigned frequency matrices.
(3) **SKL** (Huang et al. 2014): Random Fourier features for stationary kernels and squared Frobenius norm as regularization term with updating spectral density during training.
(4) **NSKL**: A special case of ASKL with non-stationary spectral, learned spectral density. But it uses squared Frobenius norm on model weights $\|\boldsymbol{W}\|_F^2$ as regularization.

Datasets   We evaluate the performance of the proposed learning framework ASKL and compared algorithms based on several publicly available datasets, including both classification and regression tasks. Especially, we standardize outputs for regression tasks to $[0, 100]$ for better illustration. To obtain stable results, we run methods on each dataset 30 times with randomly partition such that 80% data for training and 20% data for testing. Further, those multiple test errors allow the estimation of the statistical significance of difference among methods. To explore the influence of factors upon convergence, we evaluate both test accuracy and objective on MNIST dataset (LeCun et al. 1998).

Empirical Results

Empirical results of all algorithms are shown in Table 2, where accuracy is used for classification tasks and root mean squared error (RMSE) is used for regression tasks. We bold results which have the best performance on each dataset, but also mark sub-optimal results with underlines which have a significant difference with the best ones, by using pairwise t-test on results of 30 times repeating data split and training.

The results in Table 2 show: (1) The proposed algorithm ASKL outperforms compared algorithms on almost all dataset that coincides with our theoretical results. (2) The use of non-stationary kernels brings notable performance improvement but approaches based on stationary kernels still perform well on easy tasks, e.g. *shuttle* and *cpusmall*. (3) Due to the difference between assigned spectral density and learned frequency matrices, there are significant performance gaps between those two groups {SK, NSK} and {SKL, NSKL}, especially on complicated datasets *satimage* and *letter*. It confirms that learning feature mappings in both input-dependent and output-dependent ways leads

		SK	NSK	SKL	NSKL	ASKL
Accuracy(↑)	segment	89.93±2.12	90.15±2.08	94.58±1.86	94.37±0.81	**95.02±1.54**
	satimage	74.54±1.35	75.15±1.38	83.61±1.08	83.74±1.34	**85.32±1.45**
	USPS	93.19±2.84	93.81±2.13	95.13±0.91	95.27±1.65	**97.76±1.14**
	pendigits	96.93±1.53	97.39±1.41	98.19±2.30	98.28±1.68	**99.06±1.26**
	letter	76.50±1.21	78.21±1.56	93.60±1.14	94.66±2.21	**95.70±1.74**
	porker	49.80±2.11	51.85±0.97	54.27±2.72	<u>54.69±1.68</u>	**54.85±1.28**
	shuttle	98.17±2.81	98.21±1.46	<u>98.87±1.42</u>	98.74±1.07	**98.98±0.94**
	MNIST	96.03±2.21	96.45±2.16	96.67±1.61	98.03±1.16	**98.26±1.78**
RMSE(↓)	abalone	10.09±0.42	9.71±0.28	8.35±0.28	**7.85±0.42**	<u>7.88±0.16</u>
	space_ga	11.86±0.26	11.58±0.42	11.40±0.18	11.39±0.46	**11.34±0.27**
	cpusmall	2.77±0.71	2.84±0.38	2.56±0.72	2.57±0.63	**2.42±0.48**
	cadata	50.31±0.92	51.47±0.32	47.67±0.33	47.71±0.30	**46.34±0.23**

Table 2: Classification accuracy (%) for classification datasets and RMSE for regression datasets. (↑) means the lager the better while (↓) indicates the smaller the better. 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.
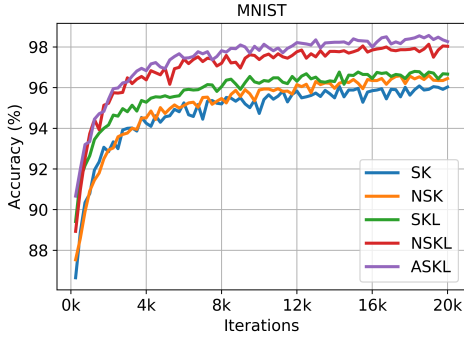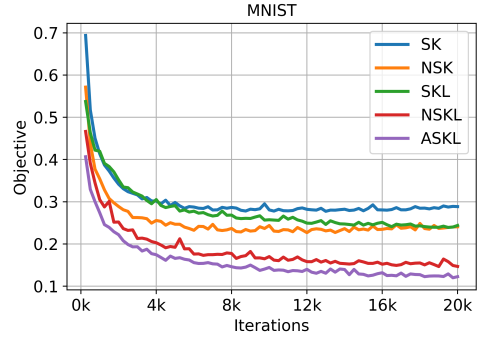

Figure 3: Accuracy curves on MNIST

Figure 4: Objective curves on MNIST

to better generalization performance. (4) The proposed algorithm ASKL usually provides better results than NSKL. That shows the effectiveness of regularization terms $\|W\|_*$ and $\|\phi(X)\|_F^2$. The experimental results demonstrate the excellent performance and stability of ASKL, corroborating the theoretical analysis and excellent performance of ASKL.

During iterations, test accuracy and the objective were recorded for every 200 iterations with batch size 32. Evaluation results in Figure 3 and Figure 4 show that ASKL outperforms other algorithms significantly. Accuracy curves and objective curves are correlated as the smaller objective corresponds to the higher accuracy. Figure 3 and Figure 4 empirically illustrate that ASKL achieves lower error bounds than with fast learning rates.

Conclusion and Discussion

In this paper, we propose automatically kernel learning framework, which jointly learns spectral density and the estimator. Both theoretical analysis and experiments illustrate that the framework obtains significant improvements in the generalization performance, owing to the use of three factors: non-stationary spectral kernel, backpropagation and

two regularization terms. The use of non-stationary spectral kernels makes feature mappings *input-dependent*, while updating frequency matrices w.r.t the objective via backpropagation that guarantees feature mappings are *output-dependent*, obtaining more powerful representation abilities. Further, we derive Rademacher complexity bounds for the algorithm. To achieve sharper bounds, we minimize two regularization terms together with ERM.

Connection with Deep Neural Networks. The proposed learning framework is also a neural network with a single hidden layer and the cosine activation, thus the theoretical findings are also applied to this kind of neural networks. Those results can be extended to deep neural networks (DNN) by stacking ϕ in the hierarchical structure. For example l-hidden layers are used that the spectral kernel is $k(x, x') = \langle \phi^l(\phi^{l-1}(\cdots \phi^1(x))), \phi^l(\phi^{l-1}(\cdots \phi^1(x'))) \rangle$. The outputs of other activations such as sigmoid and Relu can also be seen as random feature mapping $\varphi(x)$, corresponding kernel function is $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$. It is a possible way to understand deep neural networks in kernels view based on Rademacher complexity theory.

Appendix

Firstly, we introduce common notations used in Rademacher complexity theory. The space of loss function associated with $\mathcal{H}$ is denoted by

$$\mathcal{L} = \left\{ \ell(f(\boldsymbol{x}), \boldsymbol{y}) \mid f \in \mathcal{H} \right\}. \tag{13}$$

Definition 2 (Rademacher complexity on loss space). *Assume $\mathcal{L}$ is a space of loss functions as defined in Equation (13). Then the empirical Rademacher complexity of $\mathcal{L}$ is:*

$$\widehat{\mathcal{R}}(\mathcal{L}) = \frac{1}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\ell \in \mathcal{L}} \sum_{i=1}^n \epsilon_i \ell(f(\boldsymbol{x}_i), \boldsymbol{y}_i)\right],$$

where $\epsilon_i s$ are independent Rademacher random variables uniformly distributed over $\{\pm 1\}$. Its deterministic counterpart is $\mathcal{R}(\mathcal{L}) = \mathbb{E}\,\widehat{\mathcal{R}}(\mathcal{L})$.

Lemma 1 (Lemma 5 of (Cortes et al. 2016)). *Let the loss function ℓ be L-Lipschitz for $\mathbb{R}^K$ equaipped with the 2-norm,*

$$|\ell(f(\boldsymbol{x}), \boldsymbol{y}) - \ell(f(\boldsymbol{x}'), \boldsymbol{y}')| \le L\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\|_2.$$

Then, the following contraction inequation exists

$$\mathcal{R}(\mathcal{L}) \le \sqrt{2} L \mathcal{R}(\mathcal{H}).$$

Proof of Theorem 3

Proof. A standard fact is the derivation of expected loss and empirical means can be controlled by the Rademacher averages over loss space $\mathcal{L}$ (Lemma A.5 of (Bartlett et al. 2005))

$$\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f^*) \le 2\left[\sup_{f \in \mathcal{H}} \widehat{\mathcal{E}}(f) - \mathcal{E}(f)\right] \le 4\mathcal{R}(\mathcal{L}). \tag{14}$$

Combining (14) with the contraction in Lemma 1 and connection between $\widehat{\mathcal{R}}(\mathcal{H})$ and $\mathcal{R}(\mathcal{H})$ (Lemma A.4 of (Bartlett et al. 2005)), there holds with high probability at least $1 - \delta$

$$\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f^*) \le 4\sqrt{2} L \widehat{\mathcal{R}}(\mathcal{H}) + \mathcal{O}\left(\sqrt{\frac{\log 1/\delta}{n}}\right). \tag{15}$$

Then, we estimate empirical Rademacher complexity $\widehat{\mathcal{R}}(\mathcal{H})$

$$\widehat{\mathcal{R}}(\mathcal{H}) = \frac{1}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f \in \mathcal{H}} \sum_{i=1}^n \sum_{k=1}^K \epsilon_{ik} f_k(\boldsymbol{x}_i)\right]$$
$$= \frac{1}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f \in \mathcal{H}} \langle \boldsymbol{W}, \boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\rangle\right] \tag{16}$$

where $\boldsymbol{W}, \boldsymbol{\Phi}_{\boldsymbol{\epsilon}} \in \mathbb{R}^{D \times K}$ and $\langle \boldsymbol{W}, \boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\rangle = \mathrm{Tr}(\boldsymbol{W}^T \boldsymbol{\Phi}_{\boldsymbol{\epsilon}})$, we define the matrix $\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}$ as follows:

$$\boldsymbol{\Phi}_{\boldsymbol{\epsilon}} := \left[\sum_{i=1}^n \epsilon_{i1}\phi(\boldsymbol{x}_i), \sum_{i=1}^n \epsilon_{i2}\phi(\boldsymbol{x}_i), \cdots, \sum_{i=1}^n \epsilon_{iK}\phi(\boldsymbol{x}_i)\right].$$

Applying Hölder's inequality and $\|\boldsymbol{W}\|_*$ bounded by a constant B to (16), we can obtain

$$\widehat{\mathcal{R}}(\mathcal{H}) = \frac{1}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f \in \mathcal{H}} \langle \boldsymbol{W}, \boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\rangle\right]$$
$$\le \frac{1}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f \in \mathcal{H}} \|\boldsymbol{W}\|_* \|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F\right] \le \frac{B}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F\right] \tag{17}$$
$$\le \frac{B}{n}\, \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sqrt{\|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F^2}\right] \le \frac{B}{n}\, \sqrt{\mathbb{E}_{\boldsymbol{\epsilon}}\, \|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F^2}.$$

Then, we bound $\mathbb{E}_{\boldsymbol{\epsilon}}\, \|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F^2$ as follows

$$\mathbb{E}_{\boldsymbol{\epsilon}}\, \|\boldsymbol{\Phi}_{\boldsymbol{\epsilon}}\|_F^2 \le \mathbb{E}_{\boldsymbol{\epsilon}} \sum_{k=1}^K \left\|\sum_{i=1}^n \epsilon_{ik}\phi(\boldsymbol{x}_i)\right\|_2^2$$
$$\le \sum_{k=1}^K \mathbb{E}_{\boldsymbol{\epsilon}} \left\|\sum_{i=1}^n \epsilon_{ik}\phi(\boldsymbol{x}_i)\right\|_2^2$$
$$\le \sum_{k=1}^K \mathbb{E}_{\boldsymbol{\epsilon}} \sum_{i,k=1}^n \epsilon_{ik}\epsilon_{jk}\big[\langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle\big] \tag{18}$$
$$= K \sum_{i=1}^n \langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i)\rangle.$$

The last step is due to the symmetry of $\langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i)\rangle$ shown in (6). The result is similar to (Bartlett and Mendelson 2002; Li et al. 2018) Applying spectral representation in (5) of non-stationary kernels, we further bound the Rademacher complexity

$$\widehat{\mathcal{R}}(\mathcal{H}) \le \frac{B}{n}\sqrt{K \sum_{i=1}^n \langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i)\rangle}$$
$$= \frac{B}{n}\sqrt{\frac{K}{D} \sum_{i=1}^n \frac{1}{2}\Big[\cos\big((\boldsymbol{\Omega} - \boldsymbol{\Omega}')^T\boldsymbol{x}_i\big) + 1\Big]} \tag{19}$$

where $B = \sup_{f \in \mathcal{H}} \|\boldsymbol{W}\|_*$. Substituting the above inequation (19) to (15), we complete the proof. $\square$

Singular Values Thresholding (SVT)

In each iteration, to obtain a tight surrogate of Equation (4), we keep $\|\boldsymbol{W}\|_*$ while relaxing empirical loss $g(\boldsymbol{W})$ only, that leads proximal gradient (Parikh, Boyd, and others 2014)

$$\boldsymbol{W}^{t+1} = \arg\min_{\boldsymbol{W}} \lambda_1\|\boldsymbol{W}\|_* + g(\boldsymbol{W})$$
$$= \arg\min_{\boldsymbol{W}} \lambda_1\|\boldsymbol{W}\|_* + g(\boldsymbol{W}^t)$$
$$+ \langle\nabla g(\boldsymbol{W}^t), \boldsymbol{W} - \boldsymbol{W}^t\rangle + \frac{1}{2\eta}\|\boldsymbol{W} - \boldsymbol{W}^t\|_F^2$$
$$= \arg\min_{\boldsymbol{W}} \lambda_1\|\boldsymbol{W}\|_* + \frac{1}{2\eta}\|\boldsymbol{W} - (\boldsymbol{W}^t - \eta\nabla g(\boldsymbol{W}^t))\|_F^2$$
$$= \arg\min_{\boldsymbol{W}} \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{Q}\|_F^2 + \lambda_1\eta\|\boldsymbol{W}\|_*$$

where $\boldsymbol{Q} = \boldsymbol{W}^t - \eta\nabla g(\boldsymbol{W}^t)$ and η is the learning rate.

Proposition 1 (Theorem 2.1 of (Cai, Candès, and Shen 2010)). *Let $\boldsymbol{Q} \in \mathbb{R}^{D \times K}$ with rank r and its singular values decomposition (SVD) is $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{K \times r}$ have orthogonal columns, $\boldsymbol{\Sigma}$ is the diagonal $\mathrm{diag}(\{\sigma_i\}_{1 \le i \le r})$. Then,*

$$\arg\min_{\boldsymbol{W}} \left\{\frac{1}{2}\|\boldsymbol{W} - \boldsymbol{Q}\|_F^2 + \eta\|\boldsymbol{W}\|_*\right\} = \boldsymbol{U}\boldsymbol{\Sigma}_\eta\boldsymbol{V}^T,$$

where the diagonal is $\boldsymbol{\Sigma}_\eta = \mathrm{diag}(\{\sigma_i - \eta\}_+)$.

Applying Proposition 1 (Cai, Candès, and Shen 2010; Lu et al. 2015; Chatterjee and others 2015) to (8), we update $\boldsymbol{W}$ twice in each iteration, as shown in (7) and (8).

Acknowledgments

References

Bartlett, P. L., and Mendelson, S. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3(Nov):463–482.

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

Bengio, Y.; Delalleau, O.; and Roux, N. L. 2006. The curse of highly variable functions for local kernel machines. In *Advances in neural information processing systems*, 107–114.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Chatterjee, S., et al. 2015. Matrix estimation by universal singular value thresholding. *The Annals of Statistics* 43(1):177–214.

Cortes, C.; Kuznetsov, V.; Mohri, M.; and Yang, S. 2016. Structured prediction theory based on factor graph complexity. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2514–2522.

Ding, L.; Liao, S.; Liu, Y.; Yang, P.; and Gao, X. 2018. Randomized kernel selection with spectra of multilevel circulant matrices. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Huang, P.-S.; Avron, H.; Sainath, T. N.; Sindhwani, V.; and Ramabhadran, B. 2014. Kernel methods match deep neural networks on timit. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 205–209.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Le, Q.; Sarlós, T.; and Smola, A. 2013. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 85.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Li, J.; Liu, Y.; Lin, H.; Yue, Y.; and Wang, W. 2017. Efficient kernel selection via spectral analysis. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2124–2130.

Li, J.; Liu, Y.; Yin, R.; Zhang, H.; Ding, L.; and Wang, W. 2018. Multi-class learning: From theory to algorithm. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 1591–1600.

Li, J.; Liu, Y.; Yin, R.; and Wang, W. 2019. Multi-class learning using unlabeled samples : Theory and algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

Li, J.; Liu, Y.; and Wang, W. 2019. Distributed learning with random features. *arXiv preprint arXiv:1906.03155*.

Liu, Y., and Liao, S. 2014. Kernel selection with spectral perturbation stability of kernel matrix. *Science China Information Sciences* 57(11):1–10.

Liu, Y.; Lin, H.; Ding, L.-Z.; Wang, W.; and Liao, S. 2018. Fast cross-validation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2497–2503.

Liu, Y.; Liao, S.; Jiang, S.; Ding, L.; Lin, H.; and Wang, W. 2019. Fast cross-validation for kernel-based algorithms. *IEEE transactions on pattern analysis and machine intelligence*.

Lu, C.; Zhu, C.; Xu, C.; Yan, S.; and Lin, Z. 2015. Generalized singular value thresholding. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 1805–1811.

Parikh, N.; Boyd, S.; et al. 2014. Proximal algorithms. *Foundations and Trends® in Optimization* 1(3):127–239.

Quiñonero-Candela, J.; Rasmussen, C. E.; Figueiras-Vidal, A. R.; et al. 2010. Sparse spectrum gaussian process regression. *Journal of Machine Learning Research* 11(Jun):1865–1881.

Rahimi, A., and Recht, B. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 21 (NIPS)*, 1177–1184.

Remes, S.; Heinonen, M.; and Kaski, S. 2017. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 4642–4651.

Rudin, W. 1962. *Fourier analysis on groups*, volume 121967. Wiley Online Library.

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.; et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5(3):1.

Samo, Y.-L. K., and Roberts, S. 2015. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236*.

Stein, M. L. 2012. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Sun, S.; Zhang, G.; Shi, J.; and Grosse, R. 2019. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*.

Wilson, A., and Adams, R. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, 1067–1075.

Xu, C.; Liu, T.; Tao, D.; and Xu, C. 2016. Local rademacher complexity for multi-label learning. *IEEE Transactions on Image Processing* 25(3):1495–1507.

Yaglom, A. M. 1987. Correlation theory of stationary and related random functions. *Volume I: Basic Results.* 526.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.