Journal of Machine Learning Research 22 (2021) 1-38 Submitted 6/19; Revised 3/21; Published 6/21

Generalization Properties of hyper-RKHS and its Applications

Fanghui Liu* FANGHUI.LIU@KULEUVEN.BE
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Lei Shi* LEISHI@FUDAN.EDU.CN
Shanghai Key Laboratory for Contemporary Applied Mathematics
School of Mathematical Sciences, Fudan University
Shanghai, 200433, China

Xiaolin Huang XIAOLINHUANG@SJTU.EDU.CN

Jie Yang JIEYANG@SJTU.EDU.CN
Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University
Institute of Medical Robotics, Shanghai Jiao Tong University
Shanghai, 200240, China

Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Editor: Jean-Philippe Vert

Abstract

This paper generalizes regularized regression problems in a hyper-reproducing kernel Hilbert space (hyper-RKHS), illustrates its utility for kernel learning and out-of-sample extensions, and proves asymptotic convergence results for the introduced regression models in an approximation theory view. Algorithmically, we consider two regularized regression models with bivariate forms in this space, including kernel ridge regression (KRR) and support vector regression (SVR) endowed with hyper-RKHS, and further combine divide-and-conquer with Nyström approximation for scalability in large sample cases. This framework is general: the underlying kernel is learned from a broad class, and can be positive definite or not, which adapts to various requirements in kernel learning. Theoretically, we study the convergence behavior of regularized regression algorithms in hyper-RKHS and derive the learning rates, which goes beyond the classical analysis on RKHS due to the non-trivial independence of pairwise samples and the characterisation of hyper-RKHS. Experimentally, results on several benchmarks suggest that the employed framework is able to learn a general kernel function form an arbitrary similarity matrix, and thus achieves a satisfactory performance on classification tasks.

Keywords: hyper-RKHS, approximation theory, kernel learning, out-of-sample extensions

1. Introduction

Reproducing kernel Hilbert spaces (RKHS) (Aronszajn, 1950; Saitoh and Sawano, 2016) provide the ability to approximate functions by nonparametric functional representations, and thus have developed into an important tool in many areas, especially kernel methods in machine learning (Suykens et al., 2002; Schölkopf and Smola, 2018). For any two data points $\boldsymbol{x}, \boldsymbol{x}' \in X$, kernel

∗. Fanghui Liu and Lei Shi contributed equally to this work. Corresponding authors: Fanghui Liu and Jie Yang.

methods work under the setting: the original data are mapped to high or infinite dimensional RKHS such that $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle_{\mathcal{H}}$ with an implicit feature mapping $\varphi : X \to \mathcal{H}$. Here the kernel is required to be symmetric and positive definite (PD) [1], and corresponds a unique RKHS. The "reproducing" terminology indicates the *reproducing property* of RKHS

$$f(\boldsymbol{x}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H} \text{ and } \boldsymbol{x} \in X \ .$$

Accordingly, for each $\boldsymbol{x} \in X$, the point evaluation function $f \to f(\boldsymbol{x})$ is continuous, showing that strong convergence in RKHS implies point-wise convergence. This property makes RKHS an appealing choice in machine learning problems with nice theoretical guarantees in an approximation theory view (Caponnetto and De Vito, 2007; Cucker and Zhou, 2007). The structure of RKHS is determined by the choice of the kernel k, but selecting appropriate kernels is not a trivial task. In fact, RKHS is not large enough (Bach, 2017; Steinwart, 2020) and thus would lead to a lack of adaptivity in many learning problems. In this paper, we consider to learn the kernel from a hyper-reproducing kernel Hilbert space (hyper-RKHS) (Ong et al., 2005) associated with the reproducing hyper-kernel (Kondor and Jebara, 2007). Different from RKHS, every element in hyper-RKHS is a kernel function, which allows for significant model flexibility from a broad class. Specifically, the learned kernel endowed by hyper-RKHS has the property of translation and rotation invariant simultaneously (Motai, 2015), and thus is extensively applied to feature representations (Raj et al., 2017) and other applications such as classification (Tsang and Kwok, 2006), density estimation (Ganti et al., 2008), and out-of-sample extensions (Pan et al., 2017).

Learning in hyper-RKHS $\underline{\mathcal{H}}$ is general to cover various settings or applications, e.g., kernel learning, out-of-sample extensions, and indefinite kernels (real, symmetric but not positive definite). First, in kernel learning aspect, let X be a compact metric space, $\widetilde{Y} \subseteq \mathbb{R}$ be the output space[2], $\mathcal{D} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^m$ be the training set with $\boldsymbol{x}_i \in X$ and the response $\tilde{y}_i \in \widetilde{Y}$. Let $k(\cdot, \cdot) : X \times X \to \mathbb{R}$ be a positive definite kernel function that we need to learn. Figure 1 shows the kernel learning framework in hyper-RKHS via two stages. Stage 1 (in blue) formulates kernel learning as a regression problem in hyper-RKHS by minimizing the quality function $\mathcal{T}(k, \boldsymbol{K})$ between the learned kernel k and the *target kernel* $\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\top$. Here the used *target kernel*, regarded as an "ideal kernel", is able to guide the kernel learning task in hyper-RKHS as it can directly recognise the training data with certainly 100% accuracy. The used quality function $\mathcal{T}(k, \boldsymbol{K})$ evaluates the similarity between the learned kernel k and the pre-given $\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\top$, which will be formally defined in Section 2. This scheme is similar to *target alignment* (Cortes et al., 2012; Wang et al., 2015) that evaluates how well the learned Gram matrix aligns to the *target kernel* based on the multiple kernel learning framework. However, different from them, the studied kernel learning framework here is formulated as a regularized regression problem in hyper-RKHS from a broader class instead of only acquiring the linear combination of basic kernel(s). In stage 2 (in red), we aim to find a hypothesis function $f \in \mathcal{H}$ evaluated by a convex continuous loss functional $\mathbb{E}_{\boldsymbol{x},\tilde{y}}[\ell(f(\boldsymbol{x}), \tilde{y})]$ and a Tikhonov regularizer $\|f\|_{\mathcal{H}}^2$, where the convex loss $\ell : \mathcal{H} \times \widetilde{Y} \to \mathbb{R}$ quantifies the merit of the evaluation $f(\boldsymbol{x})$ at $\boldsymbol{x} \in X$. Note that the learned kernel can be indefinite that is not associated with RKHS, we then discuss it later in this section.

Second, if we consider other types beyond the *target kernel*, e.g., a pre-given kernel matrix $\boldsymbol{K}$, the above kernel learning process is transformed to tackle the out-of-sample extensions problem

1. Because of the confusing terminology on functions and their counterparts on matrices, here, we follow the convention that a positive definite function corresponds to a positive semi-definite (PSD) matrix.
2. The symbol Y is used to denote another output space that will be introduced later.

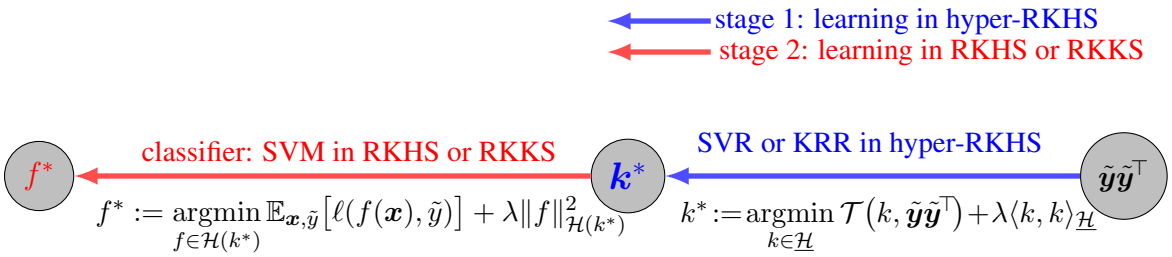


Figure 1: The two-stage kernel learning framework endowed by hyper-kernels with stage 1 (in blue): learning the kernel k by minimizing the quality functional $\mathcal{T}$ between k and the target kernel $\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\top$ in hyper-RKHS and stage 2 (in red): learning the hypothesis f in RKHS or RKKS for classification.

(Bengio et al., 2004; Pan et al., 2017), i.e., learning an underlying/unknown kernel k from a pre-defined or manually specified kernel/similarity matrix $\boldsymbol{K}$. In fact, the above kernel learning process by the *target kernel* can be also regarded as a special case of this framework due to the *target kernel* only defined on the training data. The out-of-sample extension topic widely exists in many research areas, such as 1) nonparametric kernel learning (Lu et al., 2009; Liu et al., 2020b): the kernel learning scheme is in a data specific manner, i.e., obtains the "similarity values" instead of learning a similarity function. 2) metric learning (Kulis, 2013; Jain et al., 2017): it often learns a Mahalanobis-like matrix from the given training data, but is infeasible to new data. 3) nonlinear manifold learning (Hong et al., 2013; Ho et al., 2013): the low-dimensional data coordinates are computed only for the initially available training data and can not be extended to the test data in a straightforward way. In these cases, since the learned Mahalanobis-like matrix can be regarded as a kernel matrix, and the nonlinear mapping in manifold learning can be represented by a kernel, here we investigate them in a unified framework. Hence, we aim to tackle the following question:

How to learn an underlying kernel/similarity function from the pre-given data-specific matrix?

Third, the learned kernel in stage 1 is not limited to be positive definite since hyper-RKHS has the capability of generating an indefinite kernel that is associated with a reproducing kernel Kreĭn space (RKKS) (Ong et al., 2004; Bognár, 1974) instead of RKHS. This operation is reasonable since we can hardly predict whether the underlying kernel is positive definite or indefinite even if the pre-defined $\boldsymbol{K}$ or the *target kernel* is PSD in the above two situations. Additionally imposing the positive definiteness on the learned kernel would exclude indefinite kernel learning (Loosli et al., 2016; Schleif and Tino, 2015). In practice, indefinite kernel learning is ubiquitous in many real-world applications, e.g., the hyperbolic tangent kernel (Smola et al., 2001) and the "dot-product attention" in Transformers (Wright and Gonzalez, 2021). Besides, some PD kernels would degenerate to indefinite ones, e.g., a linear combination of PD kernels (with negative coefficient) (Ong et al., 2005), dot-product kernels by ℓ_2 normalization (Pennington et al., 2015; Liu et al., 2021a), and Gaussian kernels with some geodesic distances (Feragen et al., 2015). Regarding to descriptions about RKKS, and justification, model formulation, optimization for indefinite kernel based algorithms, see (Schleif and Tino, 2015; Oglic and Gäertner, 2018; Liu et al., 2021b) among others. Accordingly, Firgure 1 include indefinite kernel learning to cover various requirements for a general kernel learning framework endowed by hyper-RKHS.

Now that learning in hyper-RKHS is adopted for numerous research fields, there is a key question left unanswered in the theoretical aspect. The convergence behavior of learning algorithms in $\underline{\mathcal{H}}$ has not been fully investigated in learning theory. In this paper, we generalize two regularized regression problems in hyper-RKHS, illustrates its utility for kernel learning and out-of-sample extensions, and proves asymptotic convergence results for the introduced regression models in an approximation theory. In particular, we make the following contributions:

Algorithmically, in Section 2, motivated by Ong et al. (2005), we consider regularized regression problems with squared loss and ε-insensitive loss (i.e., KRR and SVR) in hpyer-RKHS for kernel learning and out-of-sample extensions. Specifically, the developed models are general to output PD or indefinite kernels, which allows for significant model flexibility and universality. To make our kernel learning framework applicable to large scale situations, we combine the divide-and-conquer scheme with Nyström approximation for further improvement on computational efficiency.

Theoretically, our main results on generalization properties of KRR and SVR in hyper-RKHS are presented in Section 3, and the proofs are given in Section 4. Since learning in hyper-RKHS involves pairs of samples, which is no longer mutual pairwise independent (Luby and Wigderson, 2006), the standard approximation analysis for RKHS (Cucker and Zhou, 2007; Suzuki and Sugiyama, 2012; Mendelson and Neeman, 2010) in learning theory cannot be directly applied to hyper-RKHS. This work addresses this issue, provides the asymptotic analysis of regularized regression problems in hyper-RKHS, and fills a theoretical gap.

Experimentally, in Section 5 , we present numerical results on several benchmark datasets to verify the effectiveness of our two-stage kernel learning framework. For stage 1, we observe that our regression methods in hyper-RKHS accurately fits the given kernel matrix including PSD and non-PSD ones with small approximation errors. For stage 2, the learned kernel incorporated into SVM performs well in terms of classification accuracy whatever the pre-given kernel matrix is. Further, the developed kernel scalability method reduces the complexity of our kernel learning algorithms by orders of magnitude. Finally we discuss the related work close to our framework in Section 6 and draw the conclusion in Section 7 .

2. Learning in hyper-RKHS

In this section, we formulate the regression problem in hyper-RKHS as a regularized risk minimization problem, and then devise two regression algorithms associated with hyper-RKHS.

2.1 Regularized Risk Minimization in hyper-RKHS

The elements in hyper-RKHS are kernel functions, and thus the associated reproducing kernel is called the hyper-kernel (kernel of kernel), termed as $\underline{k}$. The definition of this space and its associated (reproducing) hyper-kernel is presented as follows.

Definition 1 *[hyper-RKHS and its (reproducing) hyper-kernel (Ong et al., 2005)] Let X be a compact metric space, $\underline{X} = X \times X$ and $\underline{\mathcal{H}}$ denotes a Hilbert space of functions $k : \underline{X} \to \mathbb{R}$. Then for any $\underline{\boldsymbol{x}}, \underline{\boldsymbol{x}}' \in \underline{X}$, the inner product space $\underline{\mathcal{H}}$ is called a hyper-RKHS endowed with the dot product $\langle \cdot, \cdot \rangle_{\underline{\mathcal{H}}}$ (and the norm $\|k\|_{\underline{\mathcal{H}}} = \sqrt{\langle k, k \rangle_{\underline{\mathcal{H}}}}$) if there exists a hyper-kernel $\underline{k} : \underline{X} \times \underline{X} \to \mathbb{R}$ with the following properties:*

- *(reproducing) $\underline{k}$ has the reproducing property $\langle k, \underline{k}(\underline{\boldsymbol{x}}, \cdot) \rangle_{\underline{\mathcal{H}}} = k(\underline{\boldsymbol{x}})$ for all $k \in \underline{\mathcal{H}}$; in particular, we have $\langle \underline{k}(\underline{\boldsymbol{x}}, \cdot), \underline{k}(\underline{\boldsymbol{x}}', \cdot) \rangle_{\underline{\mathcal{H}}} = \underline{k}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{x}}')$.*

- *(symmetric)* $\underline{k}((\boldsymbol{x},\boldsymbol{y}),(\boldsymbol{r},\boldsymbol{s})) = \underline{k}((\boldsymbol{y},\boldsymbol{x}),(\boldsymbol{r},\boldsymbol{s}))$ *for all* $\boldsymbol{x},\boldsymbol{y},\boldsymbol{r},\boldsymbol{s} \in X$. *Further, for any fixed* $\underline{\boldsymbol{x}} \in \underline{X}$, *the hyper kernel* $\underline{k}$ *is a kernel in its second argument, i.e.,* $k(\boldsymbol{x},\boldsymbol{x}') := \underline{k}(\underline{\boldsymbol{x}},(\boldsymbol{x},\boldsymbol{x}'))$ *with* $\boldsymbol{x},\boldsymbol{x}' \in X$.
- *(positive definite)* $\underline{k}$ *is positive definite on* $\underline{X}$ *and* $\underline{k}(\underline{\boldsymbol{x}},\cdot)$ *is positive definite on* X *for any* $\underline{\boldsymbol{x}} \in \underline{X}$.
- $\underline{k}$ *spans* $\underline{\mathcal{H}}$, *i.e.,* $\underline{\mathcal{H}} = \overline{span\{\underline{k}(\underline{\boldsymbol{x}},\cdot)|\underline{\boldsymbol{x}} \in \underline{\mathcal{X}}\}}$.

Here we can view the hyper-kernel as a function of four arguments, $\underline{k}((\boldsymbol{x}_1,\boldsymbol{x}_2),(\boldsymbol{x}_1',\boldsymbol{x}_2'))$ or a function of two pairs, $\underline{k}(\underline{\boldsymbol{x}},\underline{\boldsymbol{x}'})$ with $\underline{\boldsymbol{x}} := (\boldsymbol{x}_1,\boldsymbol{x}_2)$ and $\underline{\boldsymbol{x}'} := (\boldsymbol{x}_1',\boldsymbol{x}_2'))$. The reproducing and symmetric property ensures $\underline{k}$ to be a kernel, and $\underline{k}(\underline{\boldsymbol{x}},(\boldsymbol{x},\boldsymbol{x}'))$ is also a kernel for any fixed pair $\underline{\boldsymbol{x}}$. Besides, $\underline{k}$ should be positive definite so as to induce a hyper-RKHS (a special case of RKHS) based on Definition 1. Denote $C(\underline{X})$ as the space of continuous functions on $\underline{X}$ with the norm $\|f\|_\infty := \sup_{\underline{\boldsymbol{x}} \in \underline{X}} |f(\underline{\boldsymbol{x}})|$ for $f \in C(\underline{X})$. Due to the continuity of the kernel function $\underline{k}$ and compactness of $\underline{X}$, we have

$$\mathcal{G} := \sup_{\underline{\boldsymbol{x}} \in \underline{X}} \sqrt{\underline{k}(\underline{\boldsymbol{x}},\underline{\boldsymbol{x}})} < \infty\,.$$

Hence the reproducing property in hyper-RKHS indicates that $\underline{\mathcal{H}} \subset C(\underline{X})$ and

$$\|k\|_\infty = \sup_{\underline{\boldsymbol{x}} \in \underline{X}} \left|\langle k, \underline{k}(\underline{\boldsymbol{x}},\cdot)\rangle_{\underline{\mathcal{H}}}\right| \le \mathcal{G}\langle k,k\rangle_{\underline{\mathcal{H}}} = \mathcal{G}\|k\|_{\underline{\mathcal{H}}}^2, \quad \forall k \in \underline{\mathcal{H}}\,. \tag{1}$$

We can see that $\underline{\mathcal{H}}$ is different from a normal RKHS $\mathcal{H}$ on the particular form of its index set $\underline{X}$ and the additional condition on the hyper-kernel $\underline{k}$ to be symmetric in its first two arguments, and thus in its second two arguments as well. Here we investigate the regularized regression problem in hyper-RKHS, which is formulated as

$$\min_{k \in \underline{\mathcal{H}}} \frac{1}{m^2} \sum_{i,j=1}^{m} \mathcal{T}(Y_{ij}, k(\boldsymbol{x}_i,\boldsymbol{x}_j)) + \lambda\langle k,k\rangle_{\underline{\mathcal{H}}}\,, \tag{2}$$

where the first term is the quality functional $\mathcal{T}(\boldsymbol{Y},k)$ based on its point-wise definition and the regularization parameter $\lambda := \lambda(m) > 0$ satisfies $\lim_{m\to\infty}\lambda(m) = 0$. The response variable is $\boldsymbol{Y}$. In kernel learning via target alignment, $\boldsymbol{Y}$ is chosen as the target kernel (matrix), i.e., $\boldsymbol{Y} = \tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\top$. For the out-of-sample extensions issue, $\boldsymbol{Y}$ is chosen as a pre-given kernel/similarity matrix $\boldsymbol{K}$. The quality functional $\mathcal{T}(\boldsymbol{Y},k)$ focuses on the approximation ability of a kernel function k to the given $\boldsymbol{Y}$. In regression problem, it should satisfy $\mathcal{T}(Y_{ij},k(\boldsymbol{x}_i,\boldsymbol{x}_j)) = \ell(Y_{ij} - k(\boldsymbol{x}_i,\boldsymbol{x}_j))$, where the loss function $\ell(\cdot)$ can be chosen as the squared loss in least-squares, the ε-insensitive loss function in SVR, and so on. Using the representer theorem in hyper-RKHS (Ong et al., 2005), the minimizer $k^* \in \underline{\mathcal{H}}$ of problem (2) admits

$$k^*(\boldsymbol{x},\boldsymbol{x}') = \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{ij}\underline{k}\big((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x},\boldsymbol{x}')\big) \quad \text{with } \boldsymbol{x},\boldsymbol{x}' \in X,\ \beta_{ij} \in \mathbb{R}\,, \tag{3}$$

where $\boldsymbol{\beta}$ is the expansion coefficient matrix. In our formulation, k^* can be a general kernel, *i.e.*, PD or indefinite. To be exact, in hyper-RKHS, the hyper-kernel $\underline{k}$ is positive definite, but the coefficient β_{ij} in the above formulation might be negative, which results in an indefinite kernel endowed by RKKS. Therefore, as we expect, the learned solution k^* can be a positive definite kernel or an indefinite one. Such a general framework in hyper-RKHS provides strong adaptivity in kernel learning.

2.2 Regression Models in hyper-RKHS

Here we consider two regression algorithms including KRR and SVR in hyper-RKHS. By choosing the squared loss, the least-squares regression algorithm in hyper-RKHS is

$$\min_{k\in\underline{\mathcal{H}}} \frac{1}{m^2}\sum_{i,j=1}^{m}\Big(k(\boldsymbol{x}_i,\boldsymbol{x}_j)-Y_{ij}\Big)^2+\lambda\langle k,k\rangle_{\underline{\mathcal{H}}}\,, \tag{4}$$

where λ seeks for a tradeoff between the complexity of k and the fitting ability in regression. Compared to the conventional kernel ridge regression problem, our formulation in Eq. (4) is in a bivariate form because we optimize over the kernel function.

Using the representer theorem in hyper-RKHS, problem (4) can be reformulated as

$$\min_{\boldsymbol{\beta}} \frac{1}{m^2}\big\|\underline{\boldsymbol{K}}\,\mathrm{vec}(\boldsymbol{\beta})-\mathrm{vec}(\boldsymbol{Y})\big\|_2^2+\lambda\,\mathrm{vec}(\boldsymbol{\beta})^\top\underline{\boldsymbol{K}}\,\mathrm{vec}(\boldsymbol{\beta})\,, \tag{5}$$

with the coefficient vector $\mathrm{vec}(\boldsymbol{\beta})\in\mathbb{R}^{m^2}$. The hyper-kernel matrix is $\underline{\boldsymbol{K}}\in\mathbb{R}^{m^2\times m^2}$ with entries $\underline{\boldsymbol{K}}_{d(i,j,m)d(r,s,m)}=\underline{k}\big((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_r,\boldsymbol{x}_s)\big)$, in which the function $d(i,j,m)=m(i-1)+j$ maps the pair (i,j) to the row or column index of $\underline{\boldsymbol{K}}$. The hyper-kernel matrix $\underline{\boldsymbol{K}}$ is PSD when we choose a positive definite hyper-kernel k. This model is studied in (Pan et al., 2017) by additionally adding the non-negative constraint on $\boldsymbol{\beta}$ so as to output a PD kernel. Comparably, the expansion coefficient β_{ij} is not constrained to be nonnegative, which breaks through the restriction of the nonnegative constraint on the expansion coefficients in the representer theorem (3) in hyper-RKHS and thus is able to yield an indefinite kernel k. Accordingly, the solution to problem (5) can be directly given by

$$\mathrm{vec}(\boldsymbol{\beta})=\Big(\underline{\boldsymbol{K}}+\lambda m^2\boldsymbol{I}\Big)^{-1}\mathrm{vec}(\boldsymbol{Y})\,. \tag{6}$$

It can be noticed that, solving this model would be time-consuming due to the hyper-kernel matrix $\underline{\boldsymbol{K}}\in\mathbb{R}^{m^2\times m^2}$. In Section 2.3, we will consider its scalability in large scale datasets by the combination of distributed learning and Nyström approximation.

Apart from exploiting the squared loss in hyper-RKHS, we study the ε-insensitive loss for bivariate-support vector regression as a quality functional for regression, namely

$$\begin{aligned}\min_{k\in\underline{\mathcal{H}},b,\hat{\boldsymbol{\xi}},\check{\boldsymbol{\xi}}}\ & \frac{1}{2}\langle k,k\rangle_{\underline{\mathcal{H}}}+C\sum_{i,j=1}^{m}(\hat{\xi}_{ij}+\check{\xi}_{ij})\\ s.t.\ & k(\boldsymbol{x}_i,\boldsymbol{x}_j)+b-Y_{ij}\le\varepsilon+\hat{\xi}_{ij}\\ & Y_{ij}-k(\boldsymbol{x}_i,\boldsymbol{x}_j)-b\le\varepsilon+\check{\xi}_{ij}\\ & \hat{\xi}_{ij},\ \check{\xi}_{ij}\ge 0\quad\forall i,j=1,2,\cdots,m\,,\end{aligned} \tag{7}$$

where b is a bias term, C is a tradeoff between the fitting ability and the smoothness of the learned k. The notations $\hat{\boldsymbol{\xi}},\check{\boldsymbol{\xi}}$ are two slack variables associated with the quality functional $\mathcal{T}(\boldsymbol{Y},k)$. Analogous to the derived KRR in hyper-RKHS, our SVR formulation is also in a bivariate form. By the representer theorem in hyper-RKHS, the dual form of problem (7) is formulated as

$$\begin{aligned}&\max_{\hat{\boldsymbol{\beta}},\check{\boldsymbol{\beta}}} -\frac{1}{2}\sum_{i,j,r,s=1}^{m}(\hat{\beta}_{ij}-\check{\beta}_{ij})(\hat{\beta}_{rs}-\check{\beta}_{rs})\underline{k}\big((\boldsymbol{x}_i,\boldsymbol{x}_j),(\boldsymbol{x}_r,\boldsymbol{x}_s)\big)+\sum_{i,j=1}^{m}\Big\{Y_{ij}(\hat{\beta}_{ij}-\check{\beta}_{ij})-\varepsilon(\hat{\beta}_{ij}+\check{\beta}_{ij})\Big\}\\ &s.t.\ 0\le\hat{\beta}_{ij},\check{\beta}_{ij}\le C,\ \sum_{i,j=1}^{m}(\hat{\beta}_{ij}-\check{\beta}_{ij})=0\,,\end{aligned}$$

with the expansion coefficient $\beta_{ij} = \hat{\beta}_{ij} - \check{\beta}_{ij}$. We can see that the expansion coefficients $\beta_{ij} \in [-C, C]$ may be negative, which has the capability of resulting in an indefinite kernel k even if we choose a positive definite hyper-kernel $\underline{k}$. Further, the above equation can be rewritten in a compact form

$$\begin{aligned} \max_{\underline{\hat{\boldsymbol{\beta}}}, \underline{\check{\boldsymbol{\beta}}}} \ & -\frac{1}{2}(\underline{\hat{\boldsymbol{\beta}}} - \underline{\check{\boldsymbol{\beta}}})^\top \underline{\boldsymbol{K}} (\underline{\hat{\boldsymbol{\beta}}} - \underline{\check{\boldsymbol{\beta}}}) + (\underline{\hat{\boldsymbol{\beta}}} - \underline{\check{\boldsymbol{\beta}}}) \mathrm{vec}(\boldsymbol{Y}) - \varepsilon (\underline{\hat{\boldsymbol{\beta}}} + \underline{\check{\boldsymbol{\beta}}})^\top \mathbf{1} \\ s.t. \ & 0 \le \underline{\hat{\boldsymbol{\beta}}}, \underline{\check{\boldsymbol{\beta}}} \le C, \ (\underline{\hat{\boldsymbol{\beta}}} - \underline{\check{\boldsymbol{\beta}}})^\top \mathbf{1} = 0 \,, \end{aligned} \tag{8}$$

where $\underline{\hat{\boldsymbol{\beta}}} = \mathrm{vec}(\hat{\boldsymbol{\beta}}) \in \mathbb{R}^{m^2}$, $\underline{\check{\boldsymbol{\beta}}} = \mathrm{vec}(\check{\boldsymbol{\beta}}) \in \mathbb{R}^{m^2}$, and $\mathbf{1}$ is an all-one vector. One can see that the derived SVR model in hyper-RKHS shares the similar formulation with that in RKHS, and can be also solved by the SMO algorithm (Platt, 1998).

2.3 Kernel Approximation in Large Scale Situations

Regarding to optimization algorithms for problems (5) and (8), our regression models can be solved by standard optimization algorithms, i.e., the matrix inversion operator for KRR in Eq. (6) and the SMO algorithm (Platt, 1998) for SVR. While solving these algorithms are time-consuming due to the m^2 variables. Precisely, KRR in hyper-RKHS takes $\mathcal{O}(m^6)$ time complexity and requires $\mathcal{O}(m^4)$ space to store the hyper-kernel matrix. Thankfully, we do not need to simultaneously consider all pairs, though the hyper-kernel matrix is an $m^2 \times m^2$ matrix. Here we develop a divide-and-conquer approach with Nyström approximation (Williams and Seeger, 2001; Hsieh et al., 2014; Yin et al., 2020; Lin and Cevher, 2020) to speed up our method and reduce the required storage in large scale situations.

We take KRR in hyper-RKHS as an example to illustrate our two kernel approximation schemes, i.e., dividing the training data into several partitions and conducting Nyström approximation on each subset. Such approximation strategy for SVR in hyper-RKHS works in the similar fashion with KRR, and each sub-problem can be efficiently solved by liblinear (Ho and Lin, 2012). To detail our scalable scheme, we begin with KRR in hyper-RKHS with Nyström approximation, and then present the divide-and-conquer strategy. To scale KRR in hyper-RKHS to large sample situations, the Nyström scheme randomly selects a subset of M (often $M \ll m$) training data $\{\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \cdots, \widetilde{\boldsymbol{x}}_M\} \subset \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$, termed as landmarks or centers, to approximate the original hyper-kernel matrix. Here the used sampling strategy can be uniform or advanced ones, e.g., leverage scores based sampling (Alaoui and Mahoney, 2015). The solution of KRR-Nyström in hyper-RKHS via the used pairs $\{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)\}_{i,j=1}^M$ is given by

$$\begin{aligned} \widetilde{k}_{M,\lambda}(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{i,j=1}^{M} \widetilde{\beta}_{ij} \underline{k}\big((\boldsymbol{x}, \boldsymbol{x}'), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)\big) \\ \text{with } \mathrm{vec}(\widetilde{\boldsymbol{\beta}}) &= \Big(\underline{\boldsymbol{K}}_{mM}^\top \underline{\boldsymbol{K}}_{mM} + \lambda m^2 \underline{\boldsymbol{K}}_{MM}\Big)^{-1} \underline{\boldsymbol{K}}_{mM}^\top \mathrm{vec}(\boldsymbol{Y}) \,, \end{aligned}$$

where $\underline{\boldsymbol{K}}_{mM} \in \mathbb{R}^{m^2 \times M^2}$ is obtained from the whole hyper-kernel matrix $\underline{\boldsymbol{K}}$ across samples $\{(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1}^m$ and $\{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)\}_{i,j=1}^M$, and $\underline{\boldsymbol{K}}_{MM} \in \mathbb{R}^{M^2 \times M^2}$ is constructed by $\underline{k}\big((\widetilde{\boldsymbol{x}}_r, \widetilde{\boldsymbol{x}}_s), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)\big)$ with $i, j, r, s \in \{1, 2, \cdots, M\}$. Accordingly, the original hyper-kernel matrix $\underline{\boldsymbol{K}}$ can be approximated by Nyström approximation

$$\underline{\boldsymbol{K}} \approx \underline{\boldsymbol{K}}_{mM} \underline{\boldsymbol{K}}_{MM}^\dagger \underline{\boldsymbol{K}}_{mM}^\top \,,$$

Algorithm 1: Divide-and-conquer with Nyström approximation for KRR in hyper-RKHS

Input: Data points $\{\boldsymbol{x}_i\}_{i=1}^m$, the response matrix $\boldsymbol{Y}$, the (Gaussian/Wishart) hyper-kernel matrix $\underline{\boldsymbol{K}}$, and regularized paramter λ, the number of partitions v, and the number of Nyström centers $M \leq m/v$

Output: the final estimator $\bar{k}_{M,\lambda}(\boldsymbol{x}, \boldsymbol{x}')$

randomly partition the data points into v disjoint subsets $\{\mathcal{V}_t\}_{t=1}^v$.

//`in parallel: handle` $\mathcal{V}_t$ `with a local processor`

randomly select M data points from $\mathcal{V}_t$ as Nyström landmarks $\{\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \cdots, \widetilde{\boldsymbol{x}}_M\}$.

Obtain the KRR-Nyström estimator $\widetilde{k}_{M,\mathcal{V}_c,\lambda}$ on the subset $\mathcal{V}_c$ by Eq. (9).

//`end parallelism`

computer the final estimator by averaging: $\bar{k}_{M,\lambda}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{v}\sum_{c=1}^v \widetilde{k}_{M,\mathcal{V}_c,\lambda}(\boldsymbol{x}, \boldsymbol{x}')$.

where $(\cdot)^\dagger$ denotes the pseudo-inverse. By virtue of Nyström approximation, the time complexity is reduced from $\mathcal{O}(m^6)$ to $\mathcal{O}(m^2M^4)$, and the space complexity is from $\mathcal{O}(m^4)$ to $\mathcal{O}(m^2M^2)$.

Further, the computational efficiency can be improved if we incorporate the divide-and-conquer scheme into our Nyström approximation framework. We split the training data $\{\boldsymbol{x}_i\}_{i=1}^m$ into v disjoint subsets $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_v\}$, and assume that the sample size of each partition is the same for simplicity, i.e., $|\mathcal{V}_1| = |\mathcal{V}_2| = \cdots = |\mathcal{V}_v| = n$ such that $m = nv$. Then the used divide-and-conquer framework generates the global solution as the average of local estimators

$$\bar{k}_{M,\lambda}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{v}\sum_{c=1}^v \widetilde{k}_{M,\mathcal{V}_c,\lambda}(\boldsymbol{x}, \boldsymbol{x}')\,,$$

where $\widetilde{k}_{M,\mathcal{V}_c,\lambda}(\boldsymbol{x}, \boldsymbol{x}')$ is the Nyström estimator on $\mathcal{V}_c$ ($c = 1, 2, \ldots, v$) satisfying

$$\begin{aligned} &\widetilde{k}_{M,\mathcal{V}_c,\lambda}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i,j=1}^M \bar{\beta}_{ij}\underline{k}\big((\boldsymbol{x}, \boldsymbol{x}'), (\boldsymbol{x}_i, \boldsymbol{x}_j)\big) \\ &\text{with } \operatorname{vec}(\bar{\boldsymbol{\beta}}) = \Big(\underline{\boldsymbol{K}}_{nM}^\top \underline{\boldsymbol{K}}_{nM} + \lambda n^2 \underline{\boldsymbol{K}}_{MM}\Big)^{-1} \underline{\boldsymbol{K}}_{nM}^\top \operatorname{vec}(\boldsymbol{Y}_{nn})\,, \end{aligned} \tag{9}$$

where the Nyström landmarks $\{\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \cdots, \widetilde{\boldsymbol{x}}_M\}$ are from $\mathcal{V}_c$ satisfying $M \leq |\mathcal{V}_c| = n$. The matrix $\underline{\boldsymbol{K}}_{nM} \in \mathbb{R}^{n^2 \times M^2}$ is obtained from the sub-hyper-kernel matrix $\underline{\boldsymbol{K}}^{(c,c)} \in \mathbb{R}^{n^2 \times n^2}$ corresponding to the c-th partition $\mathcal{V}_c$. The matrix $\underline{\boldsymbol{K}}_{MM} \in \mathbb{R}^{M^2 \times M^2}$ corresponds to the subsampling data $\underline{\boldsymbol{K}}^{(c,c)}$ across $\{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)\}_{i,j=1}^M$ from $\mathcal{V}_c$. The matrix $\boldsymbol{Y}_{nn}$ derives from the response matrix $\boldsymbol{Y}$ on $\mathcal{V}_c$ with n training data. Under this setting, the time and space complexity are further reduced to $\mathcal{O}(m^2M^4/v)$ and $\mathcal{O}(m^2M^2/v)$, respectively. The detailed process of the approximation algorithm for KRR in hyper-RKHS is summarized in Algorithm 1.

2.4 The Used Hyper-kernels

The remaining question with respect to our regression models is how to choose the hyper-kernel $\underline{k}$. It can be noticed that numerous kernels, either PD or indefinite, can be flexibly learned in hyper-RKHS associated with a given hyper-kernel. That means the learned kernel can be data-specific rather than manually designed. Specifically, the learning behavior is independent of the choices of hyper-kernel

and the associated kernel parameters, but approximation performance on specific data indeed relies on them. Following (Kondor and Jebara, 2007), we adopt two hyper-kernels including the Gaussian hyper-kernel and Wishart hyper-kernel in this paper. The Gaussian hyper-kernel is defined as

$$\underline{k}\Big((\boldsymbol{x}_1,\boldsymbol{x}_1'),(\boldsymbol{x}_2,\boldsymbol{x}_2')\Big)=\langle \boldsymbol{x}_1,\boldsymbol{x}_1'\rangle_{\sigma^2}\langle \boldsymbol{x}_2,\boldsymbol{x}_2'\rangle_{\sigma^2}\times\Big\langle\frac{\boldsymbol{x}_1+\boldsymbol{x}_1'}{2},\frac{\boldsymbol{x}_2+\boldsymbol{x}_2'}{2}\Big\rangle_{\sigma^2+\sigma_h^2},\ \ \forall \boldsymbol{x}_1,\boldsymbol{x}_1',\boldsymbol{x}_2,\boldsymbol{x}_2'\in X\,,$$

with the notation $\langle \boldsymbol{x},\boldsymbol{x}'\rangle_{\sigma^2}=\frac{1}{(2\pi\sigma^2)^{d/2}}e^{-\|\boldsymbol{x}-\boldsymbol{x}'\|/(2\sigma^2)}$ as the Gaussian kernel, where d is the feature dimension and σ_h controls the relevance between the pairs. We can see that this hyper-kernel not only considers the similarity between two points but also takes the similarity computed by the mean of two pairs into consideration, which is useful to enhance the representation ability of the learned kernels. If we take the limit $\sigma_h\to\infty$, the Gaussian hyper-kernel decouples into the product of two Gaussian kernels.

Different from the Gaussian hyper-kernel that has a locally isotropic character, the Wishart hyper-kernel (Kondor and Jebara, 2007) is an anisotropic one to hold for rescaling data structure, defined as

$$\underline{k}\Big((\boldsymbol{x}_1,\boldsymbol{x}_1'),(\boldsymbol{x}_2,\boldsymbol{x}_2')\Big)=\int_{\boldsymbol{\Sigma}\succeq 0}\int_X\langle \boldsymbol{x}_1,\boldsymbol{x}\rangle_{\boldsymbol{\Sigma}}\,\langle \boldsymbol{x},\boldsymbol{x}_1'\rangle_{\boldsymbol{\Sigma}}\,\langle \boldsymbol{x}_2,\boldsymbol{x}\rangle_{\boldsymbol{\Sigma}}\,\langle \boldsymbol{x},\boldsymbol{x}_2'\rangle_{\boldsymbol{\Sigma}}\,\mathcal{IW}(\boldsymbol{\Sigma};\boldsymbol{D},b)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\Sigma}\,,$$

where

$$\langle \boldsymbol{x}_1,\boldsymbol{x}\rangle_{\boldsymbol{\Sigma}}=\frac{1}{(2\pi)^{m/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{x}')\Big)\,,$$

and $\mathcal{IW}(\boldsymbol{\Sigma};\boldsymbol{D},b)$ is the inverse Wishart distribution with the parameter matrix $\boldsymbol{D}\in\mathbb{R}^{m\times m}$ and an integer parameter b. The notion $\boldsymbol{\Sigma}\succeq 0$ means PSD matrices. The Wishart hyper-kernel can be regarded as the anisotropic version of the Gaussian hyper-kernel by taking $\sigma_h\to\infty$, and its second argument can be also linked to Bhattacharyya kernel (Kondor and Jebara, 2003) for any fixed pair.

Based on above descriptions, we formulate two regression algorithms in hyper-RKHS to output PD or indefinite kernels, which is demonstrated by stage 1 in Figure 1. Then the following classification task to learn the hypothesis f in stage 2 can be achieved by kernel machines, e.g., SVM used in this paper. Note that, if the learned kernel is indefinite, the SVM solver is still valid, but outputs a stationary point instead of the optimal minimum. In fact, we can also choose some advanced algorithms in RKKS, e.g., (Loosli et al., 2016; Oglic and Gäertner, 2018), as alternative ways for learning in RKKS.

3. Generalization Properties of Learning in Hyper-RKHS

In this section, we study the convergence analysis of learning problems in hyper-RKHS with squared loss and ε-insensitive loss. Although approximation analysis of classical regression algorithms including least-squares regularized regression (Wu et al., 2006; Caponnetto and De Vito, 2007; Dieuleveut et al., 2017), support vector regression (Xiang et al., 2012), quantile regression (Shi et al., 2014) in RKHS are provided, the generalization properties (in an approximation theory view) of regression problems in hyper-RKHS have not yet been fully investigated.

3.1 Problem Settings and Notations

In the context of statistical learning theory, to investigate a general regularized regression problem in hyper-RKHS, the learned regression function, also the kernel function k is defined on a compact

metric space $X \times X$ denoted by $\underline{X}$. The hyper-kernel $\underline{k} : \underline{X} \times \underline{X} \to \mathbb{R}$ is a continuous, symmetric, positive definite function. Then the associated hyper-RKHS $\underline{\mathcal{H}}$ in Definition 1 is the completion of the linear span of the set of function $\{\underline{k}(\underline{\boldsymbol{x}}, \cdot) : \underline{\boldsymbol{x}} \in \underline{X}\}$ with the inner product $\langle \cdot, \cdot \rangle_{\underline{\mathcal{H}}}$.

Let ρ be a non-degenerate Borel probability measure on $Z = \underline{X} \times Y = X \times X \times Y$ which can be factorized as

$$\rho(\underline{\boldsymbol{x}}, y) = \rho(\boldsymbol{x}, \boldsymbol{x}', y) = \rho_X(\boldsymbol{x})\rho_X(\boldsymbol{x}')\rho(y|(\boldsymbol{x}, \boldsymbol{x}'))\,,$$

where ρ_X is a probability measure on X and $\rho(y|(\boldsymbol{x}, \boldsymbol{x}'))$ is the conditional distribution on Y given $(\boldsymbol{x}, \boldsymbol{x}') \in \underline{X}$. The *target function* (also a kernel function) of ρ is defined by

$$k_\rho(\boldsymbol{x}, \boldsymbol{x}') = \int_Y y \mathrm{d}\rho\big(y|(\boldsymbol{x}, \boldsymbol{x}')\big)\,, \quad \boldsymbol{x}, \boldsymbol{x}' \in X\,.$$

The target of regression problem in hyper-RKHS is to find a good approximation of k_ρ from the pairwise sample set $Z = \{\boldsymbol{z}_{ij}\}_{i,j=1}^m = \{(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij})\}_{i,j=1}^m$, where $\{\boldsymbol{x}_i\}_{i=1}^m$ are sampled independently according to ρ_X and y_{ij} is drawn from the conditional distribution $\rho(y|(\boldsymbol{x}, \boldsymbol{x}'))$. Note that these m^2 pairwise samples $\{(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij})\}_{i,j=1}^m$ are not mutual pairwise independent (Luby and Wigderson, 2006). Actually, for $i \neq j$, $\boldsymbol{z}_{ij}$ is drawn according to ρ, while $\boldsymbol{z}_{ii}$ is distributed according to $\rho'(\boldsymbol{x}, y) = \rho_X(\boldsymbol{x})\rho(y|(\boldsymbol{x}, \boldsymbol{x}))$.

The target function k_ρ is estimated by minimizing the expected risk

$$\mathcal{E}(k) := \mathcal{E}_\rho(k) = \int_{\underline{X}\times Y} \mathcal{T}\big(y, k(\underline{\boldsymbol{x}})\big)\mathrm{d}\rho = \int_X\int_X\int_Y \mathcal{T}\big(k(\boldsymbol{x}, \boldsymbol{x}'), y\big)\mathrm{d}\rho(y|(\boldsymbol{x}, \boldsymbol{x}'))\mathrm{d}\rho_X(\boldsymbol{x})\mathrm{d}\rho_X(\boldsymbol{x}')\,.$$

We additionally suppose that there exits a constant $M^* \geq 1$, such that

$$|k_\rho(\boldsymbol{x}, \boldsymbol{x}')| \leq M^* \text{ for all } \boldsymbol{x}, \boldsymbol{x}' \in X\,.$$

For the squared loss, we have $\mathcal{T}(y, k(\boldsymbol{x}, \boldsymbol{x}')) = \big(y - k(\boldsymbol{x}, \boldsymbol{x}')\big)^2$, and thus the empirical risk functional is defined as

$$\mathcal{E}_{\boldsymbol{z}}(k) = \frac{1}{m^2}\sum_{i,j=1}^m \mathcal{T}(y_{ij}, k(\boldsymbol{x}_i, \boldsymbol{x}_j)) = \frac{1}{m^2}\sum_{i,j=1}^m \Big(k(\boldsymbol{x}_i, \boldsymbol{x}_j) - y_{ij}\Big)^2\,.$$

Hence, given the sample set Z, KRR in hyper-RKHS aims at finding a kernel function $k : X \times X \to \mathbb{R}$ such that $k_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{x}')$ is a good estimate of y for a new pair input $(\boldsymbol{x}, \boldsymbol{x}')$. To be specific, the learning algorithm generated by regularized least squares in hyper-RKHS takes the form

$$k_{\boldsymbol{z},\lambda} := \operatorname*{argmin}_{k\in\underline{\mathcal{H}}} \left\{ \frac{1}{m^2}\sum_{i,j=1}^m \Big(k(\boldsymbol{x}_i, \boldsymbol{x}_j) - y_{ij}\Big)^2 + \lambda\langle k, k\rangle_{\underline{\mathcal{H}}} \right\}. \tag{10}$$

For SVR in hyper-RKHS, it is a little sophisticated due to the insensitive parameter ε. Here we consider SVR with $\mathcal{T}(y, k(\boldsymbol{x}, \boldsymbol{x}')) = |y - k(\boldsymbol{x}, \boldsymbol{x}')|$, and then introduce ε-insensitive loss in SVR. For any $\boldsymbol{x}, \boldsymbol{x}' \in X$, the target kernel function k_ρ is defined by its value $k_\rho(\boldsymbol{x}, \boldsymbol{x}')$ to be a median function of $\rho(\cdot|(\boldsymbol{x}, \boldsymbol{x}'))$, that is

$$\begin{cases} \rho\Big(\big\{y \in Y : y \leq k_\rho(\boldsymbol{x}, \boldsymbol{x}')\big\}\big|(\boldsymbol{x}, \boldsymbol{x}')\Big) \geq \dfrac{1}{2}\,, \\ \rho\Big(\big\{y \in Y : y \geq k_\rho(\boldsymbol{x}, \boldsymbol{x}')\big\}\big|(\boldsymbol{x}, \boldsymbol{x}')\Big) \geq \dfrac{1}{2}\,. \end{cases}$$

In order to obtain a sparse solution, we introduce the ε-insensitive loss function in SVR

$$\mathcal{T}^{\varepsilon}\big(y, k(\boldsymbol{x}, \boldsymbol{x}')\big) = |y - k(\boldsymbol{x}, \boldsymbol{x}')|_{\varepsilon} = \begin{cases} 0, \text{ if } |y - k(\boldsymbol{x}, \boldsymbol{x}')| < \varepsilon; \\ |y - k(\boldsymbol{x}, \boldsymbol{x}')| - \varepsilon, \text{ otherwise}, \end{cases}$$

where the insensitivity parameter ε aims at balancing the approximation and sparsity of the algorithm and thus should change with the sample size m satisfying $\lim_{m\to\infty} \varepsilon(m) = 0$. Given the sample set Z, SVR in hyper-RKHS with the ε-insensitive loss takes the form

$$k_{\boldsymbol{z},\lambda}^{(\varepsilon)} := \underset{k\in\underline{\mathcal{H}}}{\operatorname{argmin}} \left\{ \frac{1}{m^2} \sum_{i,j=1}^{m} \mathcal{T}^{\varepsilon}\big(y_{ij}, k(\boldsymbol{x}_i, \boldsymbol{x}_j)\big) + \lambda \langle k, k \rangle_{\underline{\mathcal{H}}} \right\}. \tag{11}$$

3.2 Definitions and Assumptions

To illustrate the convergence analysis, we need the following definitions and assumptions. Note that all of the presented assumptions in hyper-RKHS here are defined on pairs but can be analogous to that in RKHS, and are hence standard and fair in approximation analysis.

We first state the definition of *projection operator* introduced in (Chen et al., 2004).

Definition 2 *(projection operator) For $B > 0$, the projection operator $\pi = \pi_B$ is defined on the space of measurable functions $k : X \times X \to \mathbb{R}$ as*

$$\pi_B(k)(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} B, \ \textit{if} \ k(\boldsymbol{x}, \boldsymbol{x}') > B; \\ -B, \ \textit{if} \ k(\boldsymbol{x}, \boldsymbol{x}') < -B; \\ k(\boldsymbol{x}, \boldsymbol{x}'), \ \textit{if} \ -B \le k(\boldsymbol{x}, \boldsymbol{x}') \le B, \end{cases}$$

and then the projection of k is denoted as $\pi_B(k)(\boldsymbol{x}, \boldsymbol{x}') = \pi_B(k(\boldsymbol{x}, \boldsymbol{x}'))$.

Since k_ρ takes the value in $[-M^*, M^*]$ almost surely, the projection operator is beneficial to estimate k_ρ by $\pi_{M^*}(k_{\boldsymbol{z},\lambda}^{(\varepsilon)})$ instead of $k_{\boldsymbol{z},\lambda}^{(\varepsilon)}$ for sharp estimation. Therefore, for SVR in hyper-RKHS, our approximation analysis attempts to bound the error $\|\pi_{M^*}(k_{\boldsymbol{z},\lambda}^{(\varepsilon)}) - k_\rho\|_{L^{p^*}_{\rho_X}}$ in the space $L^2_{\rho_X}(\underline{X})$ with some $p^* > 0$, where $L^p_{\rho_X}$ is a weighted L^p-space with the norm

$$\|k\|_{L^p_{\rho_X}} = \Big(\int_X \int_X |k(\boldsymbol{x}, \boldsymbol{x}')|^p \mathrm{d}\rho_X(\boldsymbol{x}) \mathrm{d}\rho_X(\boldsymbol{x}') \Big)^{1/p}.$$

To estimate the approximation error, we need the following assumptions with respect to the unbounded outputs, noise condition on ρ, and covering numbers for the hypothesis space. Here we consider a general setting with respect to the unbounded outputs (Wang and Zhou, 2011).

Definition 3 *(moment hypothesis) There exist constants $M \ge 1$ and $c > 0$ such that*

$$\int_Y |y|^\iota \mathrm{d}\rho(y|(\boldsymbol{x}, \boldsymbol{x}')) \le c\iota! M^\iota, \ \forall \iota \in \mathbb{N}, \ (\boldsymbol{x}, \boldsymbol{x}') \in \underline{X}. \tag{12}$$

Remark: Compared to the standard uniform boundedness assumption with $|y| \le M$ almost surely, this assumption is general since it covers Gaussian noise, sub-Gaussian noise, etc. If the condition distribution $\rho(\cdot|(\boldsymbol{x}, \boldsymbol{x}'))$ is a Gaussian distribution with variance σ^2_X bounded by B_0, then Eq. (12) is satisfied with $M := \max\{\sqrt{2}B_0, M^*\}$ and $c = 4$.

The noise condition on ρ (Christmann and Steinwart, 2007) via pairs can be defined in a similar fashion with that in RKHS.

Definition 4 *(noise condition) Let $p \in (0, \infty]$ and $q \in [1, \infty)$. A distribution ρ on $X \times X \times R$ is said to have a median of $p-$average type q if for any $(\boldsymbol{x}, \boldsymbol{x}') \in \underline{X}$, there exist a median t^* and constants $0 < a_{(\boldsymbol{x},\boldsymbol{x}')} \leq 1$, $b_{(\boldsymbol{x},\boldsymbol{x}')} > 0$ such that for each $u \in [0, a_{(\boldsymbol{x},\boldsymbol{x}')}]$,*

$$\begin{cases} \rho((t^* - u, t^*)|(\boldsymbol{x}, \boldsymbol{x}')) \geq b_{(\boldsymbol{x},\boldsymbol{x}')} u^{q-1} \\ \rho((t^*, t^* + u)|(\boldsymbol{x}, \boldsymbol{x}')) \geq b_{(\boldsymbol{x},\boldsymbol{x}')} u^{q-1}, \end{cases} \tag{13}$$

and that the function on $X \times X$ taking values $\left(b_{(\boldsymbol{x},\boldsymbol{x}')} a_{(\boldsymbol{x},\boldsymbol{x}')}^{q-1}\right)^{-1}$ at $(\boldsymbol{x}, \boldsymbol{x}') \in X \times X$ lies in $L^p_{\rho_X}$.

The noise condition in Eq. (13) ensures that $k_\rho(\boldsymbol{x}, \boldsymbol{x}') = t^*$ is uniquely defined at every $(\boldsymbol{x}, \boldsymbol{x}') \in \underline{X}$.

Apart from the above conditions, our main results about learning rates also involve the approximation ability of $\underline{\mathcal{H}}$ with respect to its capacity and k_ρ. The approximation ability can be characterised by the regularization error.

Definition 5 *The regularization error is defined as*

$$D(\lambda) = \inf_{k \in \underline{\mathcal{H}}} \left\{ \mathcal{E}(k) - \mathcal{E}(k_\rho) + \lambda \|k\|^2_{\underline{\mathcal{H}}} \right\}. \tag{14}$$

The target kernel function k_ρ can be approximated by $\underline{\mathcal{H}}$ with exponent $0 < r \leq 1$ if there exists a constant C_0 such that

$$D(\lambda) \leq C_0 \lambda^r, \ \ \forall \lambda > 0. \tag{15}$$

Remark: This is a natural assumption in approximation theory, e.g., (Wu et al., 2006; Wang and Zhou, 2011; Steinwart and Andreas, 2008). Note that $r = 1$ is the best choice as we expect, which is equivalent to $k_\rho \in \underline{\mathcal{H}}$ when $\underline{\mathcal{H}}$ is dense. In fact, the assumption in Eq. (15) can be also characterized by the *source condition* via integral operator, refer to (Caponnetto and De Vito, 2007) for details.

Further, to quantitatively understand that how the complexity of $\underline{\mathcal{H}}$ affects the learning ability of algorithm in Eq. (11), we need the capacity (roughly speaking the "size") of $\underline{\mathcal{H}}$ measured by covering numbers (Cucker and Zhou, 2007).

Definition 6 *For a subset S of $C(\underline{X})$ and $\epsilon > 0$, the* covering number $\mathscr{N}(S, \epsilon)$ *is the minimal integer $l \in \mathbb{N}$ such that there exist l disks with radius ϵ covering S.*

In this paper, the covering numbers of balls are defined by

$$\mathcal{B}_R = \{k \in \underline{\mathcal{H}} : \|k\|_{\underline{\mathcal{H}}} \leq R\},$$

where we assume that for some $s > 0$ and $C_s > 0$ such that

$$\log \mathscr{N}(\mathcal{B}_1, \epsilon) \leq C_s \left(\frac{1}{\epsilon}\right)^s, \quad \forall \epsilon > 0. \tag{16}$$

Remark: This is a standard assumption to measure the capacity of $\underline{\mathcal{H}}$ that follows with RKHS (Cucker and Zhou, 2007; Wang and Zhou, 2011; Shi et al., 2019). When $\underline{X}$ is a bounded domain and $\underline{k} \in C^\tau(\underline{X} \times \underline{X})$, Eq. (16) holds true with $s = 2m^2/\tau$. In particular, if $\underline{k} \in C^\infty(\underline{X} \times \underline{X})$, condition (16) is valid for an arbitrarily small $s > 0$. In fact, the capacity of a (hyper)-RKHS can be also measured by eigenvalue decay of the reproducing (hyper)-kernel matrix $\underline{\boldsymbol{K}}$ or effective dimension in integral operator theory (Caponnetto and De Vito, 2007). As demonstrated by (Bach, 2013; Belkin, 2018), a small (hyper)-RKHS often indicates a fast eigenvalue decay so as to obtain a promising prediction performance. In other words, functions in the (hyper)-RKHS are potentially smoother than what is necessary, which means an arbitrary small s in Eq. (16).

3.3 Main Results

Formally, our main results about SVR in hyper-RKHS are stated as follows. For $p \in (0, \infty]$ and $q \in (1, \infty)$, we denote

$$\theta = \min\left\{\frac{2}{q}, \frac{p}{p+1}\right\} \in (0, 1]. \tag{17}$$

Theorem 7 *Suppose that $|k_\rho(\boldsymbol{x}, \boldsymbol{x}')| \leq M^*$ with $M^* \geq 1$, ρ has a median of p-average type q with some $p \in (0, \infty]$ and $q \in (1, \infty)$ and satisfies assumptions Eq.* (15) *with $0 < r \leq 1$ and Eq.* (12)*. Assume that for some $s > 0$, take $\lambda = m^{-\alpha}$ with $0 < \alpha \leq 1$ and $\alpha < \frac{1+s}{s(2+s-\theta)}$, and set $\varepsilon = m^{-\gamma}$ with $\alpha r \leq \gamma \leq \infty$. Then with $p^* = \frac{pq}{p+1}$, for any $0 < \epsilon < \Theta/q$ and $0 < \delta < 1$, with confidence $1 - \delta$, we have*

$$\left\|\pi_{M^*}(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\right\|_{L^{p^*}_{\rho_X}} \leq \widetilde{C}^{\epsilon}_{\underline{X},\rho,\alpha,\gamma}\left(\log\frac{4}{\delta}\right)^{1/q} m^{\epsilon - \frac{\Theta}{q}},$$

where $\widetilde{C}^{\epsilon}_{\underline{X},\rho,\alpha,\gamma}$ is a constant independent of m or δ and the power index Θ is

$$\Theta = \min\left\{\alpha r, \frac{1 + \alpha r - \alpha}{2 - \theta}, \frac{1}{2 + s - \theta} - \frac{\alpha s}{1 + s}\right\}. \tag{18}$$

The power index Θ can be viewed as a function of variables r, s, p, q, α. The restriction $\alpha < \frac{1+s}{s(2+s-\theta)}$ ensures that Θ is positive, which verifies the valid learning rate in Theorem 7.

Remark: Note that s can be arbitrarily small when the hyper-kernel $\underline{k}$ is smooth enough. In this case, the power index Θ in Eq. (18) can be arbitrarily close to $\min(\alpha r, \frac{1}{2-\theta} + \frac{r-1}{2-\theta}\alpha)$. Regarding to SVR in RKHS, Xiang et al. (2012) demonstrates that the power index Θ in Eq. (18) can be arbitrarily close to $\min(\alpha r, \frac{1}{2-\theta})$ when the reproducing kernel is smooth enough. In this case, the derived learning rate in hyper-RKHS is not faster than that in RKHS, which is mainly effected by the approximation ability since the spanning space by hyper-RKHS is larger than RKHS. Nevertheless, if we further consider $k_\rho \in \underline{\mathcal{H}}$, that means the approximation error in Eq. (15) can be upper bounded with $r = 1$, the derived learning rate in hyper-RKHS is the same as that in RKHS, approaching to $\min(\alpha, \frac{1}{2-\theta})$.

Regarding to KRR in hyper-RKHS, the excess error for squared loss is exactly the distance in the space $L^2_{\rho_X}(\underline{X})$, i.e., $\mathcal{E}(k) - \mathcal{E}(k_\rho) = \left\|k - k_\rho\right\|^2_{L^2_{\rho_X}}$, which yields a direct variance-expectation bound. Our results about least-squares in hyper-RKHS are presented as follows.

Theorem 8 *Suppose that $|k_\rho(\boldsymbol{x}, \boldsymbol{x}')| \leq M^*$ with $M^* \geq 1$, ρ satisfies the condition in Eq.* (15) *with $0 < r \leq 1$ and the moment hypothesis in Eq.* (12) *with $c > 0$. Assume that for some $s > 0$, take $\lambda = m^{-\alpha}$ with $0 < \alpha \leq 1$ and $\alpha < \frac{1+s}{s(2+s)}$. Then for any $0 < \epsilon < \Theta$ and $0 < \delta < 1$, with confidence $1 - \delta$, we have*

$$\left\|\pi_{M^*}(k_{\boldsymbol{z},\lambda}) - k_\rho\right\|_{L^2_{\rho_X}} \leq \widetilde{C}_{\underline{X},\rho,\alpha,\gamma} \log\frac{4}{\delta} m^{\epsilon - \Theta},$$

where $\widetilde{C}_{\underline{X},\rho,\alpha,\gamma}$ is a constant independent of m or δ and the power index Θ is

$$\Theta = \min\left\{\alpha r, \frac{1}{2 + s} - \frac{\alpha s}{1 + s}\right\}. \tag{19}$$

Remark: In the special case that $k_\rho \in \underline{\mathcal{H}}$ (i.e., $r = 1$) and $\underline{k} \in C^\infty(\underline{X} \times \underline{X})$, condition (16) is satisfied for an arbitrarily small $s > 0$. Accordingly, the excess error $\left\|\pi_{M^*}(k_{\boldsymbol{z},\lambda}) - k_\rho\right\|^2_{L^2_{\rho_X}}$ can converge to zero at the (arbitrary close to) optimal rate $\mathcal{O}(1/m)$ if we take $\alpha \geq 1/2$, which matches to results on least squares in RKHS under the same assumptions, e.g., Theorem 1 in (Wang and Zhou, 2011), and Corollary 1 in (Guo and Zhou, 2013).

4. Framework of Proofs

In this section, we establish the framework of proofs for Theorem 7. We use the error decomposition technique to analyze the convergence behavior of SVR in hyper-RKHS. The key challenges in our theoretical analyses include analyzing the bias of the estimator, the effect of noise on the unbounded outputs, the non-trivial independence of pairwise samples, and the characterisation of hyper-RKHS. The last two points are the main elements on novelty in the proof. Since the proofs about the learning rate for least-squares regression in hyper-RKHS can be regarded as a simplified version of SVR in hyper-RKHS, we concentrate our proof on ε-insensitive loss in hyper-RKHS and omit the detailed proofs for the squared loss.

Before proving Theorem 7 , we need the proposition introduced in (Steinwart and Christmann, 2011; Shi et al., 2014).

Proposition 9 *Suppose that $|k_\rho(\boldsymbol{x}, \boldsymbol{x}')| \leq M^*$ with $M^* \geq 1$, ρ has a median of p-average type q with some $p \in (0, \infty]$ and $q \in (1, \infty)$, for any $k : X \times X \to [-B, B]$ with $B > 0$, there holds*

$$\left\|k - k_\rho\right\|_{L^{p^*}_{\rho_X}} \leq C_q \max\{M^*, B\}^{1-1/q} \Big\{\mathcal{E}(k) - \mathcal{E}(k_\rho)\Big\}^{\frac{1}{q}}, \tag{20}$$

where $p^* = \frac{pq}{p+1}$ and $C_q = 2^{1-1/q} q^{1/q} \|\{(b_{(\boldsymbol{x},\boldsymbol{x}')} a^{q-1}_{(\boldsymbol{x},\boldsymbol{x}')})^{-1}\}_{(\boldsymbol{x},\boldsymbol{x}')\in X\times X}\|^{1/q}_{L^p_{\rho_X}}$.

This proposition demonstrates that the excess error $\mathcal{E}(k) - \mathcal{E}(k_\rho)$ can be analysed by k_ρ and its approximation k in $L^{p^*}_{\rho_{\underline{X}}}$. Instead, the excess error for squared loss is exactly the distance in the space $L^2_{\rho_X}(\underline{X})$, i.e., $\mathcal{E}(k) - \mathcal{E}(k_\rho) = \left\|k - k_\rho\right\|^2_{L^2_{\rho_X}}$.

4.1 Error Decomposition

In order to estimate error $\|\pi_{M^*}(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\|$ in the $L^{p^*}_{\rho_X}$ space, i.e., to bound $\|\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\|$ for any $B \geq M^*$. Accordingly, by Proposition 9 , we need to estimate the excess error $\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)$ which can be conducted by an error decomposition technique (Cucker and Zhou, 2007). Note that the insensitivity parameter ε changes with m, we consider the insensitivity relation with additional ε on the error decomposition (Xiang et al., 2011), that is

$$\mathcal{T}(y, k(\boldsymbol{x}, \boldsymbol{x}')) - \varepsilon \leq \mathcal{T}^\varepsilon(y, k(\boldsymbol{x}, \boldsymbol{x}')) \leq \mathcal{T}(y, k(\boldsymbol{x}, \boldsymbol{x}')), \quad \forall \boldsymbol{x}, \boldsymbol{x}' \in X\,. \tag{21}$$

Formally, the error decomposition is given by the following proposition, with proof deferred to Appendix A.1.

Proposition 10 *Let*

$$k_\lambda = \operatorname*{argmin}_{k\in\underline{\mathcal{H}}} \Big\{\mathcal{E}(k) - \mathcal{E}(k_\rho) + \lambda\|k\|^2_{\underline{\mathcal{H}}}\Big\}\,.$$

Then the excess error $\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)$ *can be bounded by*

$$\begin{aligned}\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho) &\le \mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho) + \lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}} \\ &\le D(\lambda) + S(\boldsymbol{z},\lambda) + \frac{1}{m^2}\sum_{i,j=1}^{m}\Big|\pi_B(y_{ij}) - y_{ij}\Big| + \varepsilon\,,\end{aligned}$$

where $D(\lambda)$ *is the regularization error defined by Eq.* (14)*. The sample error* $S(\boldsymbol{z},\lambda)$ *is denoted as*

$$S(\boldsymbol{z},\lambda) = \mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) + \mathcal{E}_{\boldsymbol{z}}\big(k_\lambda\big) - \mathcal{E}\big(k_\lambda\big) = S_1(\boldsymbol{z},\lambda) + S_2(\boldsymbol{z},\lambda)\,,$$

with

$$\begin{aligned}S_1(\boldsymbol{z},\lambda) &= \Big\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big\} - \Big\{\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}_{\boldsymbol{z}}(k_\rho)\Big\}\,,\\ S_2(\boldsymbol{z},\lambda) &= \Big\{\mathcal{E}_{\boldsymbol{z}}\big(k_\lambda\big) - \mathcal{E}_{\boldsymbol{z}}(k_\rho)\Big\} - \Big\{\mathcal{E}(k_\lambda) - \mathcal{E}(k_\rho)\Big\}\,.\end{aligned}$$

By Proposition 10 , the excess error can be bounded by the sample error $S(\boldsymbol{z},\lambda)$, the regularization error $D(\lambda)$, and the output error. The regularization error is bounded by Eq. (15). Besides, by virtue of Eq. (1) and Eq. (14), k_λ under supremum norm can be also upper bounded by

$$\|k_\lambda\|_\infty \le \mathcal{G}\|k_\lambda\|_{\underline{\mathcal{H}}} \le \mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}} \le \mathcal{G}\sqrt{C_0}\lambda^{\frac{r-1}{2}}\,. \tag{22}$$

In the next, our error analysis mainly focuses on how to estimate the sample error and the output error. We expect that these approximation errors will approximate to zero at a certain rate as the sample size tends to infinity.

4.2 Estimate Sample Error and Output Error

This section is devoted to estimating the sample error $S(\boldsymbol{z},\lambda)$ and the output error. Our error analysis mainly focuses on how to estimate $S_1(\boldsymbol{z},\lambda)$ and $S_2(\boldsymbol{z},\lambda)$. The asymptotical behaviors of S_1 and S_2 are usually illustrated by the convergence of the empirical mean $\frac{1}{m^2}\sum_{i,j=1}^{m}\xi_{ij}$ to its expectation $\mathbb{E}\xi$, where $\{\xi_{ij}\}_{i,j=1}^{m}$ are "independent" random variables on (Z,ρ) defined as

$$\xi(\boldsymbol{x},\boldsymbol{x}',y) := \mathcal{T}\big(y,k_\lambda(\boldsymbol{x},\boldsymbol{x}')\big) - \mathcal{T}\big(y,k_\rho(\boldsymbol{x},\boldsymbol{x}')\big)\,,\ \text{and}\ \xi_{ij} := \xi(\boldsymbol{x}_i,\boldsymbol{x}_j,y_{ij})\,. \tag{23}$$

Note that the Lipschitz property of the ε-insensitive loss in SVR guarantees the boundedness of ξ when k is bounded. So ξ defined by Eq. (23) is a bounded random variable even if y is unbounded. When k is fixed, which is exactly the case as we estimate S_2, the convergence is guaranteed by the following lemma.

For $R \ge 1$, denote

$$\mathscr{W}(R) = \Big\{\boldsymbol{z} \in Z^{m\times m} : \|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|_{\underline{\mathcal{H}}} \le R\Big\}\,. \tag{24}$$

Lemma 11 *If* ξ *is a symmetric real-valued function on* $X\times X\times Y$ *with mean* $\mathbb{E}(\xi)$*. Assume that* $\mathbb{E}(\xi)\ge 0$*,* $|\xi(\boldsymbol{x},\boldsymbol{x}',y) - \mathbb{E}\xi| \le T$ *almost surely and* $\mathbb{E}\xi^2 \le c_1(\mathbb{E}\xi)^\theta$ *for some* $0\le\theta\le 1$ *and* $c_1 \ge 0$*,* $T\ge 0$*. Then for every* $\epsilon > 0$ *there holds*

$$\text{Prob}\left\{\frac{\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j=1,j\ne i}^{m}\xi(\boldsymbol{x}_i,\boldsymbol{x}_j,y_{ij}) - \mathbb{E}\xi}{\sqrt{(\mathbb{E}\xi)^\theta + \epsilon^\theta}} \ge \epsilon^{1-\frac{\theta}{2}}\right\} \le \exp\left\{-\frac{(m-1)\epsilon^{2-\theta}}{4c_1 + \frac{4}{3}T\epsilon^{1-\theta}}\right\}. \tag{25}$$

Proof Define

$$U_i = \frac{1}{2\lfloor m/2 \rfloor}\left\{\xi_{i_1 i_2} + \xi_{i_2 i_1} + \xi_{i_3 i_4} + \xi_{i_4 i_3} + \cdots + \xi_{i_{2\lfloor m/2\rfloor-1} i_{2\lfloor m/2\rfloor}} + \xi_{i_{2\lfloor m/2\rfloor} i_{2\lfloor m/2\rfloor-1}}\right\},$$

where $\lfloor m/2 \rfloor$ denotes the greatest integer not exceeding $m/2$, and $(i_1, i_2, \ldots, i_m)$ is a permutation of $(1, 2, \ldots, m)$. Then

$$\frac{1}{m(m-1)}\sum_{i,j=1, i\neq j}^{m} \xi(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij}) = \frac{1}{m!}\sum_{m\cdot m} U_i, \text{ and } \mathbb{E}(U_i) = \mathbb{E}\xi\,,$$

where the notation $\sum_{m\cdot m}$ is the summation taken over all permutations of the integers $1, 2, \ldots, m$. Note that for distinct integers k, k', l, l' not exceeding m, random variables $\xi_{i_k i_l} + \xi_{i_l i_k}$ and $\xi_{i_{k'} i_{l'}} \xi_{i_{l'} i_{k'}}$ are independent. Then each U_i is a summation of $\lfloor m/2 \rfloor$ independent random variables. Therefore, we have

$$\begin{aligned}
&\text{Prob}\left\{\frac{\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j=1,j\neq i}^{m}\xi(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij}) - \mathbb{E}\xi}{\sqrt{(\mathbb{E}\xi)^\theta + \epsilon^\theta}} \geq \epsilon^{1-\frac{\theta}{2}}\right\} = \text{Prob}\left\{\frac{\frac{1}{m!}\sum_{m\cdot m}\Big[U_i - \mathbb{E}\xi\Big]}{\sqrt{(\mathbb{E}\xi)^\theta + \epsilon^\theta}} \geq \epsilon^{1-\frac{\theta}{2}}\right\} \\
&\leq \frac{1}{m!}\,\text{Prob}\left\{\frac{U_i - \mathbb{E}U_i}{\sqrt{(\mathbb{E}\xi)^\theta + \epsilon^\theta}} \geq \epsilon^{1-\frac{\theta}{2}}\right\} \leq \exp\left\{-\frac{\lfloor m/2 \rfloor \epsilon^{2-\theta}}{2\big(c_1 + \frac{1}{3}T\epsilon^{1-\theta}\big)}\right\}.
\end{aligned} \tag{26}$$

Here we derive the last inequality by applying the Bernstein inequality. We thus complete the proof by noting $\lfloor m/2 \rfloor \geq \frac{m-1}{2}$. ■

When the random variables are given by Eq. (23), for a general distribution ρ, the variance-expectation condition $\mathbb{E}\xi^2 \leq c_1(\mathbb{E}\xi)^\theta$ is satisfied with $\theta = 0$ and $c_1 = 1$. Specifically, if ρ satisfies the noise condition (i.e., Definition 4), the variance-expectation bound can be improved by the following lemma.

Lemma 12 *Under the same assumption of Proposition 9, for any $k : X \times X \to [-B, B]$ with $B > 0$, there holds*

$$\mathbb{E}\Big\{\mathcal{T}\big(y, k(\boldsymbol{x}, \boldsymbol{x}')\big) - \mathcal{T}\big(y, k_\rho(\boldsymbol{x}, \boldsymbol{x}')\big)\Big\}^2 \leq C_\theta \max\{B, M^*\}^{2-\theta}\Big(\mathcal{E}(k) - \mathcal{E}(k_\rho)\Big)^\theta, \tag{27}$$

where θ is given by Eq. (17) *and $C_\theta = 2^{2-\theta}q^\theta \|\{(b_{(\boldsymbol{x},\boldsymbol{x}')}a_{(\boldsymbol{x},\boldsymbol{x}')}^{q-1})^{-1}\}_{(\boldsymbol{x},\boldsymbol{x}')\in \underline{X}}\|_{L^p_{\rho_X}}^\theta$ with the constant $a_{(\boldsymbol{x},\boldsymbol{x}')} \in (0, 1]$ and $b_{(\boldsymbol{x},\boldsymbol{x}')} > 0$.*

This lemma is a direct corollary of Proposition 9 . The positive θ here will lead to sharper estimates and play an essential role in the convergence analysis.

Now we can bound $S_2(\boldsymbol{z}, \lambda)$ by the following proposition, refer to the proof in Appendix A.2.

Proposition 13 *Under the same assumption of Proposition 9, for any $0 < \delta < 1$, there exists a subset of Z_1 of $Z^{m\times m}$ with measure at least $1 - \delta/4$, such that for any $\forall \boldsymbol{z} \in Z_1$*

$$S_2(\boldsymbol{z}, \lambda) \leq \frac{1}{2}D(\lambda) + 16(C_\theta + 1)\left(B + M^* + \mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}\right)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}} + \frac{1}{m}\Big(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}} + M^*\Big). \tag{28}$$

In the next, we aim to bound $S_1(\boldsymbol{z}, \lambda)$ with respect to the samples $\boldsymbol{z}$. Thus a uniform concentration inequality for a family of functions containing $k^{(\varepsilon)}_{\boldsymbol{z},\lambda}$ is needed to estimate S_1. Since we have $k^{(\varepsilon)}_{\boldsymbol{z},\lambda} \in \mathcal{B}_R$ by Eq. (3.2), we shall bound S_1 by the following concentration inequality with a properly chosen R, with proof deferred to Appendix A.3.

Proposition 14 *Under the same assumption of Proposition 9 and Eq.* (16)*, for any* $0 < \delta < 1$*,* $R \geq 1$*,* $B > 0$*, there exists a subset* Z_2 *of* $Z^{m\times m}$ *with measure at least* $1 - \delta/4$*, such that for any* $\boldsymbol{z} \in \mathscr{W}(R) \cap Z_2$*,*

$$\begin{aligned} S_1(\boldsymbol{z}, \lambda) &= \Big\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big\} - \Big\{\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}_{\boldsymbol{z}}(k_\rho)\Big\} \\ &\leq 20\epsilon^*(m, R, \frac{\delta}{4}) + \frac{1}{2}\Big\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big\} + \frac{1}{m}(B + M^*)\,, \end{aligned} \tag{29}$$

where $\epsilon^*(m, R, \frac{\delta}{4})$ *is given by*

$$\epsilon^*(m, R, \frac{\delta}{4}) = 16(M^* + B)(C_\theta + 1)\left(\log\frac{4}{\delta} m^{-\frac{1}{2-\theta}} + C_s R^s m^{-\frac{1}{2+s-\theta}} R^{\frac{s}{1+s}}\right).$$

The left in the error decomposition demonstrated by Proposition 10 is to bound $\frac{1}{m^2}\sum_{i,j=1}^m |\pi_B(Y_{ij}) - Y_{ij}|$, which involves the unboundedness of the output y. Following Proposition 5 in (Shi et al., 2014), under the assumption of Eq. (12), for any $d \in \mathbb{N}$ and $0 < \delta < 1$, there exists a subset Z_3 of $Z^{m\times m}$ with measure at least $1 - \delta/4$, such that

$$\frac{1}{m^2}\sum_{i,j=1}^m |\pi_B(y_{ij}) - y_{ij}| \leq c\Big\{(d+1)! + 2^{d+2}d^d\Big\}M^{d+1}B^{-d} + \frac{12M2^{d+1}}{m}\log\frac{4}{\delta} \quad \forall \boldsymbol{z} \in Z_3 \cap \mathscr{W}(R)\,, \tag{30}$$

where we use $\frac{6M}{m-1} \leq \frac{12M}{m}$ for any $m > 1$.

4.3 Derive Convergence Rates

Based on above analyses, combining the bounds in Proposition 10 , 13 , 14 , Eq. (22) and Eq. (30) , the excess error $\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)$ can be bounded by the following proposition, with the proof in Appendix A.4.

Proposition 15 *Assume that* $|k_\rho(\boldsymbol{x}, \boldsymbol{x}')| \leq M^*$ *with* $M^* \geq 1$*.* ρ *has a median of* p*-average type* q *with some* $p \in (0, \infty]$ *and* $q \in (1, \infty)$ *and satisfies assumptions Eq.* (17) *with* $0 < \theta \leq 1$ *, Eq.* (15) *with* $0 < r \leq 1$*, and Eq.* (12) *with* $c > 0$*. Assume that for some* $s > 0$*, take* $\lambda = m^{-\alpha}$ *with* $0 < \alpha \leq 1$ *and* $\alpha < \frac{1+s}{s(2+s-\theta)}$*. Set* $\varepsilon = m^{-\gamma}$ *with* $\alpha r \leq \gamma \leq \infty$*. Then for* $B \geq M^*$*,* $d \in \mathbb{N}$*, and* $0 < \delta < 1$ *with confidence* $1 - \delta$*, there holds*

$$\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho) \leq 4\widetilde{C}\log\frac{4}{\delta}m^{-\Theta} + 2c\Big\{(d+1)! + 2^{d+2}d^d\Big\}M^{d+1}B^{-d} + \frac{12M2^{d+1}}{m}\log\frac{4}{\delta}\,,$$

where $\widetilde{C}$ *is a constant independent of* m *or* δ *and the power index* Θ *is*

$$\Theta = \min\left\{\alpha r, \frac{1 + \alpha r - \alpha}{2 - \theta}, \frac{1}{2 + s - \theta} - \frac{\alpha s}{1 + s}\right\}. \tag{31}$$

Now we are ready to give the proof of Theorem 7 .

Proof Using Proposition 9 and 15 , for any $B \geq M^*$ with confidence $1-\delta$, there holds

$$\begin{aligned}\left\|\pi_{M^*}(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\right\|_{L^{p^*}_{\rho_X}} \leq \left\|\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\right\|_{L^{p^*}_{\rho_X}} \leq C_q \left(4\widetilde{C}\log\frac{4}{\delta}m^{-\Theta}\right)^{\frac{1}{q}} B \\ + B\left(2c\Big[(d+1)! + 2^{d+2}d^d\Big]M^{d+1}B^{-d}\right)^{1/q} + \left(12M2^{d+1}\log\frac{4}{\delta}\right)^{1/q} B^{1-1/q}m^{-1/q}\,,\end{aligned} \tag{32}$$

where Θ is given by Eq. (31), and the first inequality admits by $|k_\rho| \leq M^*$ almost surely. Following (Shi et al., 2014), we have

$$\left(2c\Big[(d+1)! + 2^{d+2}d^d\Big]M^{d+1}B^{-d}\right)^{1/q} \leq 2^{\frac{\Theta+2\epsilon}{q\epsilon}}(20cM)m^{-\Theta/q}\,, \tag{33}$$

and

$$\left(12M2^{d+1}\log\frac{4}{\delta}\right)^{1/q} B^{1-1/q}m^{-1/q} \leq 2^{\frac{d+1}{q}}\left(12M\log\frac{4}{\delta}\right)^{1/q} Bm^{-1/q}\,. \tag{34}$$

Finally, we complete the proof by combining Eqs. (33) and (34) into Eq. (32)

$$\left\|\pi_{M^*}(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}) - k_\rho\right\|_{L^{p^*}_{\rho_X}} \leq \widetilde{C}^{\epsilon}_{\underline{X},\rho,\alpha,\gamma}\left(\log\frac{4}{\delta}\right)^{1/q} m^{\epsilon-\frac{\Theta}{q}}\,,$$

with

$$\widetilde{C}^{\epsilon}_{\underline{X},\alpha} = 3(M+M^*)\max\left\{C_q(4\widetilde{C})^{\frac{1}{q}}, (20cM)^{1/q}, (12M)^{1/q}\right\}2^{\frac{\Theta+2\epsilon}{q\epsilon}}\Theta\epsilon^{-1}\,,$$

which concludes the proof. ■

4.4 Theoretical Results on Kernel Approximation

A series of kernel approximation schemes, e.g., divide-and-conquer (Zhang et al., 2013), distributed learning (Lin et al., 2017), Nyström approximation (Rudi et al., 2015), random features (Rudi and Rosasco, 2017), have been extensively studied in learning theory, mainly on kernel ridge regression in RKHS. Recently, much efforts focus on the combination of several strategies, e.g., divide-and-conquer with Nyström approximation (Yin et al., 2020), distributed learning with stochastic gradient descent (SGD) (Lin and Cevher, 2020), random features with SGD (Carratino et al., 2018). Accordingly, following (Yin et al., 2020; Rudi et al., 2017), our derived theoretical result on the full problem can be extended to the approximation version with divide-and-conquer and Nyström approximation. Here we briefly present the error decomposition result of KRR in hyper-RKHS under such two kernel approximation settings and sketch our key ideas.

Define the noise-free version of $\widetilde{k}_{M,\mathcal{V}_c,\lambda}$ by Nyström approximation on the subset $\mathcal{V}_c$ as

$$\widetilde{k}_{M,\mathcal{V}_c,\rho,\lambda}(\boldsymbol{x},\boldsymbol{x}') = \sum_{i,j=1}^{M}\widetilde{\beta}_{ij}\underline{k}\big((\boldsymbol{x},\boldsymbol{x}'),(\widetilde{\boldsymbol{x}}_i,\widetilde{\boldsymbol{x}}_j)\big)\,,$$

$$\text{with}\quad \mathrm{vec}(\widetilde{\boldsymbol{\beta}}) = \Big(\underline{\boldsymbol{K}}_{nM}^\top\underline{\boldsymbol{K}}_{nM} + \lambda n^2\underline{\boldsymbol{K}}_{MM}\Big)^{-1}\underline{\boldsymbol{K}}_{nM}^\top\mathrm{vec}(k_\rho(\boldsymbol{x}_r,\boldsymbol{x}_s))\;\; r,s\in\{1,2,\cdots,n\}\,,$$

and further $\bar{k}_{M,\mathcal{V},\rho,\lambda} = 1/v \sum_{c=1}^{v} \tilde{k}_{M,\mathcal{V}_c,\rho,\lambda}$ is the average of all the v partitions. Define the noise version of the local estimator (without Nyström approximation) on the subset $\mathcal{V}_c$ as

$$\widetilde{k}_{\mathcal{V}_c,\rho,\lambda}(\boldsymbol{x},\boldsymbol{x}') = \sum_{i,j=1}^{n} \widetilde{\beta}_{ij} \underline{k}\big((\boldsymbol{x},\boldsymbol{x}'),(\boldsymbol{x}_i,\boldsymbol{x}_j)\big)\,,$$
$$\text{with}\;\; \mathrm{vec}(\widetilde{\boldsymbol{\beta}}) = \Big(\underline{\boldsymbol{K}}_{nn} + \lambda n^2 \boldsymbol{I}\Big)^{-1} \mathrm{vec}(k_\rho(\boldsymbol{x}_r,\boldsymbol{x}_s))\;\; r,s \in \{1,2,\cdots,n\}\,,$$

and further $\bar{k}_{\mathcal{V},\rho,\lambda} = 1/v \sum_{c=1}^{v} \tilde{k}_{\mathcal{V}_c,\rho,\lambda}$ is the average of all the v partitions. Then the error decomposition for KRR in hyper-RKHS under such two approximation strategies can be similarly obtained by[3]

$$\mathbb{E}\mathcal{E}(\bar{k}_{M,\lambda}) - \mathcal{E}(k_\rho) \lesssim \frac{1}{v}\mathbb{E}\|\widetilde{k}_{M,\mathcal{V}_c,\lambda} - \widetilde{k}_{M,\mathcal{V}_c,\rho,\lambda}\|^2_{L^2_{\rho_X}} + \underbrace{\mathbb{E}\|\bar{k}_{M,\mathcal{V},\rho,\lambda} - \bar{k}_{\mathcal{V},\rho,\lambda}\|^2_{L^2_{\rho_X}}}_{\text{Nyström error}}$$
$$+ \frac{1}{v}\mathbb{E}\|\bar{k}_{\mathcal{V},\rho,\lambda} - k_\lambda\|^2_{L^2_{\rho_X}} + \underbrace{\|k_\lambda - k_\rho\|^2_{L^2_{\rho_X}}}_{\text{approximation error}}, \quad c = 1,2,\ldots,v\,,$$

where the first term and the third term are sample error which controls the variance of the outputs y and sample variance, the second term involves with Nyström approximation and the last term is the bias, i.e., the approximation error.

In particular, the approximation error can be directly upper bounded by Eq. (15). The key part in the analysis is to use the Bernstein's inequality to study the relationship between the empirical pair sample and its expectation, which has been established in Lemma 11. Accordingly, proofs for sample error and Nyström error can be exactly obtained by combining Lemma 11 with previous results (Yin et al., 2020; Rudi et al., 2017). We therefore omit the proof in this paper.

5. Experiments

We evaluate the proposed two regression models with squared loss and the ε-insensitive loss in hyper-RKHS, termed as "hyper-KRR" and "hyper-SVR" for learning kernels, and then apply them to classification tasks. First, we experimentally investigate the approximation performance of our methods for the known kernels on the UCI repository[4]. Second, we conduct experiments to learn an underlying kernel from the "ideal" kernel on a wide range of classification problems on the UCI classification datasets. Third, for scalability, we test our methods on two large datasets including *ijcnn1* and *covtype*[5]. Last, for out-of-sample extensions, we apply our method to non-parametric kernel learning on the *MNIST* handwritten digits dataset (Lecun et al.). The experiments implemented in MATLAB are conducted on a PC with Intel® i7-8700K CPU (3.70 GHz) and 64 GB RAM. The source code of our implementation can be found in `http://www.lfhsgre.org`.

During training, σ^2 in the Gaussian hyper-kernel is set to the variance of data, and σ_h^2 is tuned via 5-fold cross validation over the values $\{0.25\sigma^2, 0.5\sigma^2, \sigma^2, 2\sigma^2, 4\sigma^2\}$. The regularization parameters

3. The notation $a(v,M,m) \lesssim b(v,M,m)$ means that $a(v,M,m) \leq Cb(v,M,m)$ where C is some absolute constant independent of v, M, m.
4. `https://archive.ics.uci.edu/ml/datasets.html`
5. Both data sets are available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

λ in KRR and C in SVR are searched on grids of $\log_{10}$ scale in the range of $[10^{-5}, 10^{5}]$. The two slack variables $\hat{\xi}_{ij}, \check{\xi}_{ij}$ in SVR are set to 0.1 and 0.01, respectively.

5.1 Approximating known PD/non-PD kernels

Here we carry out experiments to investigate the approximation performance for known kernels. The out-of-sample extension based algorithm (Pan et al., 2017) is taken into comparisons. This method solves a nonnegative least squares in hyper-RKHS, which can be regarded a special case of hyper-KRR. Nevertheless, we do not want to claim that the learned (indefinite) kernel in our framework is better than the PD one from (Pan et al., 2017). Instead, our target is to show the utility or flexibility of our framework. For fair comparison, these three algorithms in hyper-RKHS are associated with the same hyper-kernel, i.e., the hyper-Gaussian kernel used in this subsection.

For the experiments on UCI data sets, the data points are partitioned into 40% labeled data, 40% unlabeled data, and 20% test data. The labeled and unlabeled data points form the training dataset. Such setting follows with (Pan et al., 2017), which simultaneously considers tranductive learning and inductive learning. Here the pre-given kernel matrix is generated by a known kernel including a positive definite one and an indefinite one. Learning on known kernels focuses on the approximation performance of the compared algorithms on these kernels. The used evaluation metric here is relative mean square error (RMSE) between the learned regression function $k^*(\boldsymbol{x}, \boldsymbol{x}')$ and the pre-given kernel matrix $\boldsymbol{K}$ over m^2 pairwise data points. Besides, we also evaluate our kernel learning methods incorporated into SVM for classification. As a consequence, such experimental setting on known kernels help us to comprehensively investigate the approximation ability of the compared algorithms on PD or non-PD kernels.

Results on the known Gaussian kernel: Here the pre-given kernel matrix is generated by a known Gaussian kernel. Table 1 reports the experimental results in terms of classification accuracy and test root mean squared error (RMSE) for out-of-sample extensions on the known Gaussian kernel. From the results, we can see that the proposed hyper-SVR and hyper-KRR have the capability of approximating the Gaussian kernel. Further, in terms of the test accuracy, hyper-SVR performs better than the other two methods on test data. But, regarding to hyper-KRR and hyper-SVR, in general, we see that classification performance and approximation accuracy are not well correlated. A better approximation quality cannot guarantee better classification performance. This is not a unique phenomenon in our algorithm but a common issue in the kernel approximation topic (Avron et al., 2017; Munkhoeva et al., 2018; Liu et al., 2020a). Approximation and generalization appears two correlated tasks but experimentally not. How to bridge the gap between good (distinct) approximation and indistinctive generalization performance still remains an open question in theory.

Results on the known $\log$ **kernel:** Here we conduct experiments on a known indefinite kernel to generate the pre-given output. The used $\log$ kernel (Boughorbel et al., 2005) is given by $k(\boldsymbol{x}, \boldsymbol{x}') = -\log(1 + \frac{\|\boldsymbol{x}-\boldsymbol{x}'\|}{\sigma})$ with the chosen $\sigma = 1$. Table 1 reports the classification accuracy and test root mean squared error for out-of-sample extensions across the $\log$ kernel. In terms of the approximation ability of these compared algorithms, hyper-SVR performs best to approximate these two indefinite kernels. It can be noticed that, the algorithm in (Pan et al., 2017) is not able to approximate the $\log$ kernel due to its negative values, and thus is infeasible for such kernel. In general, we see improvement yielded by our two regression models on the final classification accuracy, which also shows the flexibility of indefinite kernel learning models.

Table 1: RMSE performance on test data and classification accuracy of (mean±std. deviation) of each compared algorithm on unlabeled data and test data, in which the given kernel matrix is generated by a Gaussian kernel and a log kernel, respectively. The best performance is highlighted in **bold**. The results directly achieved by these three known kernels do not participate in ranking.

	Dataset	type	fertility	australian	wine	sonar	heart	guide1-t
	(#data, #feature)		(100, 9)	(690, 14)	(178, 13)	(208, 60)	(270, 13)	(4000, 4)
Gaussian kernel	the *known* Gaussian kernel	unlabel	95.02±7.11	83.92±1.90	96.68±1.93	75.34±3.07	78.63±4.52	95.67±3.71
		test	87.45±6.77	82.64±2.23	98.02±1.81	72.14±1.38	80.91±4.89	95.11±3.33
	(Pan et al., 2017)	unlabel	88.03±8.11	81.56±3.78	96.89±1.4 1	73.78±5.67	**81.45**±3.56	**95.89**±2.83
		test	85.51±7.72	82.64±3.60	96.33±2.67	73.83±8.02	79.04±5.11	93.42±2.81
		RMSE	0.152	**0.088**	0.123	0.165	0.128	0.288
	hyper-KRR	unlabel	92.53±7.80	81.52±4.01	**97.33**±1.45	**77.12**±4.21	80.14±5.02	95.81±2.67
		test	86.01±9.33	**83.63**±4.20	96.62±3.41	67.82±8.20	**82.73**±5.22	95.65±2.54
		RMSE	**0.081**	0.138	**0.062**	**0.085**	**0.095**	**0.104**
	hyper-SVR	unlabel	**95.64**±8.52	**82.24**±3.70	97.13±1.20	73.24±6.13	81.30±3.62	95.14±3.33
		test	**88.53**±7.10	82.61±3.93	**98.04**±2.22	**75.54**±6.42	80.12±3.20	**97.92**±2.32
		RMSE	0.102	0.108	0.089	0.143	0.120	0.120
log kernel	the *known* log kernel	unlabel	95.23±5.72	84.04±1.93	98.04±1.32	74.44±7.32	80.51±1.42	95.24±4.11
		test	81.52±8.81	83.90±1.92	96.12±2.62	78.33±6.82	81.44±6.50	95.02±3.71
	(Pan et al., 2017)[1]	-	-	-	-	-	-	-
	hyper-KRR	unlabel	**98.01**±2.62	73.84±1.92	**97.31**±1.33	72.34±5.32	81.02±2.21	95.12±1.43
		test	88.02±4.81	76.64±5.14	60.13±8.08	65.84±9.56	77.63±7.02	91.84±3.93
		RMSE	0.005	0.717	0.748	0.697	0.435	0.827
	hyper-SVR	unlabel	97.83±4.22	**84.12**±1.73	97.02±1.91	**72.72**±4.67	**81.74**±3.12	**96.84**±1.34
		test	**93.53**±2.44	**83.42**±3.80	**95.53**±5.12	**66.74**±6.32	**80.93**±4.70	**94.22**±1.63
		RMSE	**0.002**	**0.474**	**0.138**	**0.174**	**0.196**	**0.523**

[1] We omit the results provided by Pan et al. (2017) on the log kernel because this method cannot output a nonnegative coefficient vector for approximation due to the negative values in the log kernel.

5.2 Learning by approximating the "ideal" kernel

As aforementioned, the "ideal" kernel can be used to guide the kernel learning task. Here we evaluate our methods with other representative kernel learning based algorithms embedded in SVM for classification.

Experimental settings: Table 2 lists a brief description of six UCI datasets including the number of data n and the feature dimension d. The data are normalized to $[0, 1]^d$ in advance. The compared algorithms include

- KTA (Cortes et al., 2012): A two-stage kernel learning framework jointly learns the weights of base kernels by maximizing the alignment with the "ideal" kernel in stage 1, and then is embedded into SVM for classification. Here the base kernels are chosen as eleven Gaussian kernels with the kernel width $\sigma \in \{2^{-5}, 2^{-4}, \cdots, 2^5\}$.

- BMKL (Gonen, 2012): A Bayesian multiple kernel learning algorithm ensemble eleven Gaussian kernels with different kernel widths $\sigma \in \{2^{-5}, 2^{-4}, \cdots, 2^5\}$ and three polynomial kernels with degrees $1, 2, 3$.

Table 2: Classification accuracy of (mean±std. deviation) of our algorithms on test data for the "ideal" kernel versus representative kernel learning based approaches equipped with various base kernels. The best performance is highlighted in **bold**.

Dataset	fertility	australian	wine	sonar	heart	guide1-t
(#data, #feature)	(100, 9)	(690, 14)	(178, 13)	(208,60)	(270, 13)	(4000, 4)
KTA	86.67±2.05	82.52±1.55	96.20±2.22	80.41±2.88	83.24±1.62	88.22±0.68
BMKL	85.50±3.34	84.61±1.56	95.48±2.21	81.68±2.93	**84.46**±1.88	96.05±0.46
RF	81.33±5.31	82.17±2.02	94.88±2.94	80.52±3.58	82.46±2.12	95.83±0.48
MIKL	86.83±1.46	**86.45**±0.98	94.78±2.02	77.72±5.57	82.90±2.21	89.74±0.57
SVM-CV	86.50±2.53	85.14±0.84	95.14±2.32	80.40±4.48	80.43±3.35	96.49±0.44
hyper-KRR(Gaussian)	88.50±3.37	82.82±4.80	92.53±4.83	77.67±5.07	81.11±3.12	93.92±3.44
hyper-SVR(Gaussian)	89.50±5.50	85.68±3.56	**97.32**±1.82	**82.32**±3.34	82.65±2.35	93.22±2.88
hyper-KRR(Wishart)	90.25±2.34	81.21±2.84	94.83±2.77	78.45±2.24	80.15±2.71	92.65±2.58
hyper-SVR(Wishart)	**90.30**±2.12	84.23±2.27	96.65±2.35	81.13±2.94	81.43±2.52	**96.52±1.48**

- RF (Sinha and Duchi, 2016): A nonparametric kernel learning framework generates a large number of random features (we set to 10,000 in our experiment) by the Gaussian kernel, and then learn their weights based on target alignment.

- MIKL (Kowalski et al., 2009): A multiple indefinite kernel learning framework ensembles a linear kernel and two Gaussian kernels with $\sigma = 0.1$ and $\sigma = 100$ via a mixed norm regularization scheme. The combination coefficient can be negative, which allows for indefinite kernel learning. In our experiments, we use the ℓ_1-norm regularization as an example for comparison.

- SVM-CV: The SVM classifier with the Gaussian kernel is served as a baseline, where the balance parameter C and the kernel width σ are tuned by 5-fold cross validation on a grid of points, i.e., $\sigma = [2^{-5}, 2^{-4}, \ldots, 2^5]$ and $C = [2^{-5}, 2^{-4}, \ldots, 2^5]$.

Our methods includes four version determined by the used two regressors: KRR and SVR in hyper-RKHS and the used two hyper-kernels: hyper-Gaussian kernel and hyper-Wishart kernel. We follow with the setting in Section 5.1, these kernel learning based algorithms are conducted by randomly picking 40% of the data for training and the rest for test. The experiments are repeated 10 trials on these six datasets.

Experimental results: Table 2 reports the test classification accuracy of all compared methods. Compared with KTA and RF based on kernel target alignment, our methods perform better to learn the underlying kernel from the "ideal" kernel in hyper-RKHS, and thus achieve promising performance with noticeable margins. When compared to BMKL and MIKL based on multiple kernel learning, the proposed SVR with Gaussian/Wishart hyper-kernel performs well on several datasets, which verifies the effectiveness of our kernel learning scheme. It indicates that the learned underlying kernel is flexible beyond a linear combination of several base kernels. For self comparisons, in terms of the test accuracy, the proposed hyper-SVR with Gaussian/Wishart kernel is superior to the remaining three versions as a whole. Regarding to the loss function, our regression model with the squared

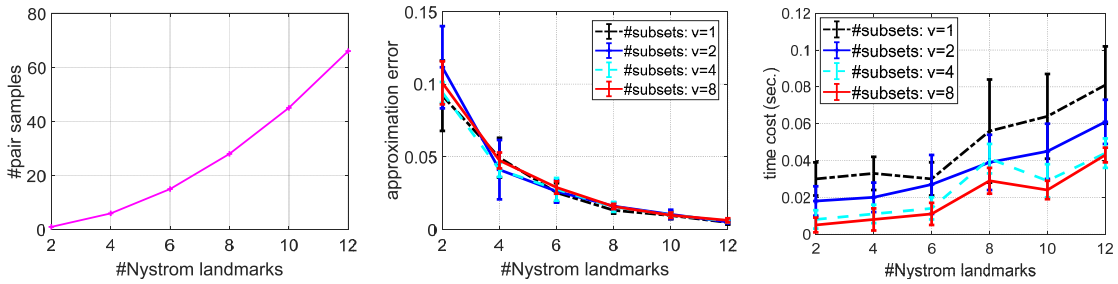


Figure 2: Approximating the Gaussian hyper-kernel matrix with varying #Nyström landmarks M and #subsets v on the *heart* dataset.

loss in hyper-RKHS is often inferior to that with the ε-insensitive loss whatever the hyper-kernel is chosen.

5.3 Validation of Kernel Approximation and Results on Large Scale Datasets

In this subsection, we first quantitatively evaluate the Gaussian hyper-kernel approximation effect by the number of Nyström landmarks M and subsets v, and then apply such two schemes to large scale datasets.

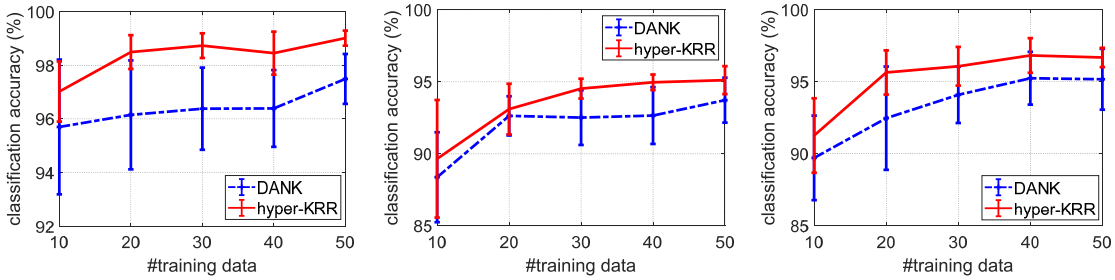
In our experiment, we equally divide the m training data into v partitions $\{\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_v\}$. Then on the subset $\mathcal{V}_c$ ($c = 1, 2, \ldots, v$), we use M landmarks for Nyström approximation to obtain an approximated hyper-kernel matrix $\widetilde{\underline{\boldsymbol{K}}}^{(c)} = \underline{\boldsymbol{K}}_{nM}\underline{\boldsymbol{K}}^{\dagger}_{MM}\underline{\boldsymbol{K}}^{\top}_{nM}$. Finally, the approximated hyper-kernel matrix averaged on v subsets is given by $\widetilde{\underline{\boldsymbol{K}}} = 1/v\sum_{c=1}^{v}\widetilde{\underline{\boldsymbol{K}}}^{(c)}$ such that $\underline{\boldsymbol{K}} \approx \widetilde{\underline{\boldsymbol{K}}}$. In this case, the approximation error is evaluated by $\|\widetilde{\underline{\boldsymbol{K}}} - \underline{\boldsymbol{K}}\|_{\mathrm{F}}/\|\underline{\boldsymbol{K}}\|_{\mathrm{F}}$ on the m^2 pair samples. Strictly speaking, the number of unduplicated pair samples is $\begin{pmatrix} 2 \\ m \end{pmatrix}$.

Since the number of pair samples dramatically increases with m, we consider a small-scale *heart* dataset with $m = 108$ training data for evaluation. If we take M Nyström landmarks, the number of unduplicated pair samples can be reduced to $\widetilde{M} = \begin{pmatrix} 2 \\ M \end{pmatrix}$. Figure 2 shows the number of unduplicated pair samples by M landmarks, approximation error, and time cost for approximation (mean $\pm$ std. across 10 trials) on the *heart* dataset. We take $M = 2, 4, 6, 8, 10, 12$ and $v = 1, 2, 4, 8$ into comparison. The approximation error under different number of landmarks M and subsets v is shown in Figure 2(b). We find that, as the number of landmarks M increases, the approximation error dramatically decreases even if M is much smaller than training data size m. Nevertheless, the divide-and-conquer scheme, e.g., $v = 2, 4, 8$, does not incur in extra approximation error when compared to the original case with $v = 1$, which demonstrates its utility. Specifically, this scheme is able to decrease time cost for kernel approximation as shown in Figure 2(c), which validates its effectiveness in terms of computational efficiency.

After quantitatively evaluating the performance of the developed kernel approximation scheme (divide-and-conquer and Nyström approximation), we incorporate them into the studied model in hyper-RKHS on large scale datasets for prediction. Here we choose two large scale datasets

Table 3: Classification accuracy and total time cost of each compared algorithm on two large scale datasets for the ideal kernel.

Dataset	v	(Pan et al., 2017)	hyper-KRR	hyper-SVR	distributed BMKL
ijcnn1	5	90.49%(1354.2s)	90.72%(1375.2s)	90.22%(1322.4s)	97.35%(230576s)
#train=49,990	10	89.72%(835.8s)	89.71%(846.8s)	89.37%(1156.1s)	97.36%(12020s)
#test=91,701	20	90.49%(743.5s)	90.94%(752.1s)	90.97%(1035.9s)	97.34%(5844.8s)
covtype	50	68.32%(4231.5s)	70.41%(4276.3s)	70.64%(4353.5s)	77.03%(185182s)
#train=232,405	100	69.67%(3213.4s)	69.82%(3241.5s)	76.58%(3352.2s)	76.30%(81551s)
#test=232,405	200	70.61%(2300.6s)	70.45%(2305.8s)	70.64%(2317.5s)	76.83%(18001s)



(a) digits 1 *vs.* 9 (b) digits 2 *vs.* 7 (c) digits 4 *vs.* 6

Figure 3: Classification accuracy of DANK and hyper-KRR with varying number of training data on the *MNIST* dataset.

including *ijcnn1* and *covtype* to test the compared algorithms in hyper-RKHS on the ideal kernel. Instead, for these three learning algorithms in hpyer-RKHS, we divide the data into v disjoint subsets $\{\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_v\}$, and then conduct Nyström approximation on each subset. The number of subsets is set to $v = 5, 10, 20$ on the *ijcnn1* dataset, and $v = 50, 100, 200$ on the *covtype* dataset. Following (Pan et al., 2017), the number of Nyström landmarks is set to $M = 0.05m$. Besides, we also include BMKL equipped with Gaussian kernels and polynomial kernels for comparison. Note that, Nyström approximation on BMKL appears non-trivial, and thus we just incorporate BMKL into the divide-and-conquer framework and cooperate without the rankings.

Table 3 reports the number of training and test data, the number of subsets, the mean classification accuracy and the total time cost. Experimental results show that all the three methods in hyper-RKHS can be feasible to large scale case, owing much to the developed kernel approximation scheme. We find that, these three algorithms achieve similar performance in terms of classification accuracy and time cost. As the number of subsets increases, these three algorithms achieve slight fluctuation on the test accuracy but significantly improve the computational efficiency. Besides, BMKL achieves the best performance on classification accuracy but takes much more time cost for kernel learning. Here we just report its results but do not include it for fair comparison as distributed BMKL is just equipped with the divide-and-conquer scheme without Nyström approximation.

5.4 Out-of-sample extensions for nonparametric kernel learning

As mentioned in the introduction, nonparametric kernel learning in a data-driven manner is often faced with the out-of-sample extensions issue, i.e., a non-parametric kernel/similarity matrix is learned but the underlying kernel function is unknown. For example, Liu et al. (2020b) propose a data-adaptive non-parametric kernel (DANK) learning framework to improves the model flexibility. In DANK, a data-adaptive matrix is learned based on the training data but is unknown on test data. To address the out-of-sample extension issues, they directly choose a simple reciprocal nearest neighbor scheme that extends the data-adaptive matrix from training to test data. We have to be faced with the inconsistency when using such interpolation scheme. Since the out-of-sample extension issue can be addressed by learning a kernel function in hyper-RKHS, here we compare hyper-KRR with the simple interpolation strategy on the *MNIST* handwritten digits dataset (Lecun et al.) for evaluation.

In our experimental setting, several (easily confused) digit pairs, including 1 *vs.* 9, 2 *vs.* 7, and 4 *vs.* 6, are taken into comparison. Specifically, we choose a few number of training data (i.e., 10, 20, 30, 40, 50) to validate the effectiveness of the employed out-of-sample extension strategy. We first use DANK to learn a non-parametric kernel matrix on the training data, then adopt the out-of-sample extension strategies to extend the kernel matrix from training data to test data, including the original reciprocal nearest neighbor scheme and the studied hyper-KRR, and finally incorporate it into SVM for classification on the test data. Figure 3 shows the test accuracy across 10 trials (mean$\pm$std. deviation) of the compared two out-of-sample extensions schemes. Results on classification accuracy indicate that the developed framework in hpyer-RKHS is able to achieve improvement with about 2% margin on DANK when compared to the original reciprocal nearest neighbor scheme (Liu et al., 2020b). Such improvement on these three digit pairs demonstrates the effectiveness of the studied hyper-RKHS based algorithms for out-of-sample extension, especially when the training data size is small or limited.

6. Discussion

Here we briefly discuss the related topics on kernel learning and neural networks close to the studied framework in this paper.

Our kernel learning framework belongs to a *two-stage* process that first learns a suitable kernel from the training data, and then uses the learned kernel in a conventional kernel machine, such as SVM or SVR for prediction. One representative approach is developed by *target alignment* (Cristianini et al., 2001; Cortes et al., 2012; Kumar et al., 2012). In stage 1, they consider finding a "good" combination of base kernels using the training data based on *target alignment* (Cortes et al., 2012; Wang et al., 2015). Accordingly, the learned weight vector yields the learned kernel for the subsequent prediction process. Stage 2 is a standard learning problem in RKHS $\mathcal{H}$ associated with the learned kernel. Our framework for kernel learning is different from them in the hypothesis space. Classical two-stage kernel learning framework in essence belongs to multiple kernel learning (Gönen and Alpaydın, 2011) in RKHS due to the pre-given kernels. Nevertheless, our framework in hyper-RKHS does not restrict specific formulation on kernel. Since a pre-given positive definite kernel can correspond to a fixed combination of pre-given elements in hyper-RKHS, the space spanned by a linear combination of PD kernels is only a small subspace in hyper-RKHS. Hence, our kernel learning framework can be learned in this space from a broader class, which allows for significant model flexibility. More importantly, the application of the studied framework is not limited to kernel learning. It can be also applied to out-of-sample extensions in non-parametric

kernel learning to learn a underlying kernel/similarity function from a pre-given similarity matrix as demonstrated by Section 5.4. This is actually beyond the topic of kernel learning, which in turn expands the application of learning in hyper-RKHS.

Actually, several representative approaches are able to achieve the similar effect as the used learning framework in hyper-RKHS for stage 1. For example, learning by random features (Sinha and Duchi, 2016) is able to work in a two-stage setting by first learning the weights of random features based on target alignment, and then obtaining a predictor. Such learning strategy in the spectral density sense is also used in (Bullins et al., 2018) and can be further improved by generative models (Li et al., 2019). Besides, pairwise learning (Stock et al., 2018; Lei et al., 2020) is an alternative way to achieve this target by constructing pairwise kernels, which measures the similarity between two pairs $(\boldsymbol{x}_1, \boldsymbol{x}_1')$ and $(\boldsymbol{x}_2, \boldsymbol{x}_2')$. Such similarity learning on pair samples is also popular in deep metric learning, e.g., contrastive embedding via Siamese networks (Bromley et al., 1994; Guo et al., 2017), triplet embedding (Salakhutdinov and Hinton, 2007; Hoffer and Ailon, 2015) that jointly constitutes a positive pair and a negative pair.

It is worth nothing that kernel learning is not conflict with existing works in deep learning. In fact, the connections between kernel methods and (deep) neural networks in over-parameterized setting have been extensively explored in recent years, e.g., the relations between Gaussian processes and infinitely wide multi-layer networks (Lee et al., 2018); the equivalence between *weakly/fully-trained* neural networks (Chizat et al., 2019; Ghorbani et al., 2019) and kernel regression by random features (Rahimi and Recht, 2007; Mei and Montanari, 2019) or neural tangent kernel (Jacot et al., 2018) under some proper initialization; the equivalence between training a two-layer neural network via gradient descent and learning a data-adaptive kernel in a dynamic RKHS (Dou and Liang, 2020). We remark upfront that connections to kernel methods is not the only way for analyzing (deep) neural networks. The spanning space of neural networks is also not limited to RKHS. For example, the "dot-product attention" in Transformers can be characterized as kernel learning in Banach spaces (Wright and Gonzalez, 2021) instead of RKHS, which also leads to an indefinite kernel but not in RKKS. The functional space of two layer wide-width neural networks can be induced by the variation norm (Bach, 2017; Chizat and Bach, 2020), which is much larger than that of the RKHS norm for better understanding. Further, many other approaches, with different points of views, have been proposed for deep learning theory, but they are out of scope of our discussion here.

7. Conclusion

In this paper, we have studied the generalization properties of regularized regression models in hyper-RKHS. The excess error converges at a certain learning rate as the sample size increases. The derived learning rate provides a justification for us to learn the kernel in hyper-RKHS with theoretical guarantees. Hence, we characterize a kernel learning framework in this space for kernel learning and out-of-sample extensions. The studied framework in hyper-RKHS is quite general to cover a series of applications, e.g., kernel/metric learning, out-of-sample extensions.

Acknowledgments

We thank the anonymous reviewers for their constructive and insightful comments. The research leading to these results has received funding from the European Research Council under the Euro-

pean Union's Horizon 2020 research and innovation program / ERC Advanced Grant E-DUALITY (787960). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. This work was supported in part by Research Council KU Leuven: Optimization frameworks for deep kernel machines C14/18/068; Flemish Government: FWO projects: GOA4917N (Deep Restricted Kernel Machines: Methods and Foundations), PhD/Postdoc grant. This research received funding from the Flemish Government (AI Research Program). This work was supported in part by Ford KU Leuven Research Alliance Project KUL0076 (Stability analysis and performance improvement of deep reinforcement learning algorithms), EU H2020 ICT-48 Network TAILOR (Foundations of Trustworthy AI - Integrating Reasoning, Learning and Optimization), Leuven.AI Institute; and in part by the National Natural Science Foundation of China (Grants Nos. 61876107, 61977046, U1803261), National Key R&D Program of China (No. 2019YFB1311503), and NSFC/RGC Joint Research Scheme (Nos. 1201101029 and N_CityU102/20), in part by Shanghai Science and Technology Research Program (20JC1412700 and 19JC1420101), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and SJTU Global Strategic Partnership Fund (2020 SJTU-CORNELL).

A. Proofs

A.1 Proofs of Proposition 10

Proof According to the project operator π_B in Definition 2, for any given $a, b \in \mathbb{R}$, if $a \geq b$, we have

$$\pi_B(a) - \pi_B(b) = \begin{cases} 0 & \text{if } a \geq b \geq B \text{ or } -B \geq a \geq b\,, \\ \min\{a, B\} + \min\{-b, B\} & \text{otherwise }. \end{cases}$$

Then we have $0 \leq \pi_B(a) - \pi_B(b) \leq a - b$ if $a \geq b$. Similarly, when $a \leq b$, we have $a - b \leq \pi_B(a) - \pi_B(b) \leq 0$. Hence for any $(\boldsymbol{x}, \boldsymbol{x}', y) \in Z$ and $k : X \times X \to \mathbb{R}$, there holds

$$\mathcal{T}^{\varepsilon}\big(\pi_B(y), \pi_B(k(\boldsymbol{x}, \boldsymbol{x}'))\big) \leq \mathcal{T}^{\varepsilon}\big(y, k(\boldsymbol{x}, \boldsymbol{x}')\big)\,.$$

Recall Eq. (11), $\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)$ can be bounded by

$$\begin{aligned} \mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) &= \frac{1}{m^2}\sum_{i,j=1}^{m} \mathcal{T}^{\varepsilon}\big(y_{ij}, \pi_B(k(\boldsymbol{x}_i, \boldsymbol{x}_j))\big) \\ &\leq \frac{1}{m^2}\sum_{i,j=1}^{m} \mathcal{T}^{\varepsilon}\big(\pi_B(y_{ij}), \pi_B(k(\boldsymbol{x}_i, \boldsymbol{x}_j))\big) + \frac{1}{m^2}\sum_{i,j=1}^{m} \Big|\pi_B(y_{ij}) - y_{ij}\Big| \qquad (35) \\ &\leq \mathcal{E}_{\boldsymbol{z}}\big(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\big) + \frac{1}{m^2}\sum_{i,j=1}^{m} \Big|\pi_B(y_{ij}) - y_{ij}\Big|\,, \end{aligned}$$

where the second term $\frac{1}{m^2}\sum_{i,j=1}^{m}|\pi_B(y_{ij})-y_{ij}|$ is termed as the output error. Accordingly, we have

$$\begin{aligned}
&\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}(k_\rho)+\lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}}\\
&=\left\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}(k_\rho)\right\}+\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)+\lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}}\\
&\leq\left\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}(k_\rho)\right\}+\mathcal{E}_{\boldsymbol{z}}\big(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\big)+\lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}}+\frac{1}{m^2}\sum_{i,j=1}^{m}\Big|\pi_B(y_{ij})-y_{ij}\Big|\\
&\leq\left\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}(k_\rho)\right\}+\mathcal{E}_{\boldsymbol{z}}\big(k_\lambda\big)+\lambda\|k_\lambda\|^2_{\underline{\mathcal{H}}}+\varepsilon+\frac{1}{m^2}\sum_{i,j=1}^{m}\Big|\pi_B(y_{ij})-y_{ij}\Big|\\
&:=D(\lambda)+\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)-\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big)+\mathcal{E}_{\boldsymbol{z}}\big(k_\lambda\big)-\mathcal{E}\big(k_\lambda\big)+\varepsilon+\frac{1}{m^2}\sum_{i,j=1}^{m}\Big|\pi_B(y_{ij})-y_{ij}\Big|,
\end{aligned}$$

where the first inequality holds by Eq. (35), the second inequality satisfies because $k^{(\varepsilon)}_{\boldsymbol{z},\lambda}$ is the minimizer of Eq. (11) and the insensitivity condition in Eq. (21), and the last equality admits by Eq. (14). Finally, we draw our conclusion. ■

A.2 Proofs of Proposition 13

Proof Considering the random variable ξ in Eq. (23) on (Z,ρ), we have

$$S_2(\boldsymbol{z},\lambda)=\frac{1}{m^2}\sum_{i,j=1}^{m}\xi(\boldsymbol{z}_{ij})-\mathbb{E}(\xi)\leq\frac{1}{m(m-1)}\sum_{i,j=1}^{m}\sum_{j=1,j\neq i}^{m}\xi(\boldsymbol{z}_{ij})+\frac{1}{m^2}\sum_{i=1}^{m}\xi(\boldsymbol{z}_{ii})-\mathbb{E}(\xi)\,.$$

First, we consider the non-diagonal elements $\xi(\boldsymbol{z}_{ij})$ with $i\neq j$. Since $\|k_\lambda\|_\infty\leq\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}$ by Eq. (22) and $k_\rho(\boldsymbol{x},\boldsymbol{x}')$ contained in $[-M^*,M^*]$. Accordingly, we can get

$$\big|\xi-\mathbb{E}(\xi)\big|\leq\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*\,.$$

By Lemma 12, the variance-expectation condition of $\xi(\boldsymbol{x},\boldsymbol{x}',y)$ is satisfied with θ given by Eq. (17) and $c_1=C_\theta\max\{B,M^*\}^{2-\theta}$. Applying Lemma 11, there exists a subset Z_1 of $Z^{m\times m}$ with confidence $1-\delta/4$, we have

$$\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\xi(\boldsymbol{z}_{ij})-\mathbb{E}\xi\leq\sqrt{(\mathbb{E}\xi)^\theta+\epsilon^\theta}\epsilon^{1-\frac{\theta}{2}}\leq(\mathbb{E}\xi)^{\frac{\theta}{2}}\epsilon^{1-\frac{\theta}{2}}+\epsilon\leq\frac{\theta}{2}\mathbb{E}\xi+\left(2-\frac{\theta}{2}\right)\epsilon,\ (36)$$

where the last inequality is from Young's inequality. Let ϵ be the solution of the equation

$$\exp\left\{-\frac{(m-1)\epsilon^{2-\theta}}{4C_\theta\max\{B,M^*\}^{2-\theta}+\frac{4}{3}(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*)\epsilon^{1-\theta}}\right\}=\delta/4\,.$$

Using Lemma 7.2 in Cucker and Zhou (2007), we find

$$\epsilon \leq \max\left\{ \frac{8\Big(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*\Big)\log\frac{4}{\delta}}{3(m-1)}, \left(\frac{8C_\theta \max\{B,M^*\}^{2-\theta}\log\frac{4}{\delta}}{m-1}\right)^{\frac{1}{2-\theta}} \right\}$$
$$\leq 8(C_\theta+1)\left(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*+B\right)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}},$$

where we use $\frac{1}{m} \leq m^{-\frac{1}{2-\theta}}$ and $(\frac{1}{m-1})^{-\frac{1}{2-\theta}} \leq m^{-\frac{1}{2-\theta}}$ with $\theta \in (0,1]$ in Eq. (17). Substituting the above bound to Eq. (36), we obtain

$$\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\xi(\boldsymbol{z}_{ij}) - \mathbb{E}\xi \leq \frac{\theta}{2}\Big\{\mathcal{E}(k_\lambda)-\mathcal{E}(k_\rho)\Big\}+8\left(2-\frac{\theta}{2}\right)(C_\theta+1)\left(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*+B\right)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}}$$
$$\leq \frac{1}{2}D(\lambda)+16(C_\theta+1)\left(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*+B\right)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}}.$$

Next we consider the diagonal elements $\xi(\boldsymbol{z}_{ij})$ with $i=j$, that is

$$\frac{1}{m^2}\sum_{i=1}^{m}\xi(\boldsymbol{x}_i,\boldsymbol{x}_i,y_{ii}) \leq \frac{1}{m}\Big(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*\Big).$$

Finally, combining above two equations, we have

$$S_2(\boldsymbol{z},\lambda) \leq \frac{1}{2}D(\lambda)+16(C_\theta+1)\left(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*+B\right)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}}+\frac{1}{m}\Big(\mathcal{G}\sqrt{\frac{D(\lambda)}{\lambda}}+M^*\Big),$$

which concludes the proof. ■

A.3 Proofs of Proposition 14

Proof Consider the function set $\mathcal{F}_R$ with $R>0$ by

$$\mathcal{F}_R := \big\{\mathcal{T}\big(y,\pi_B(k)(\boldsymbol{x},\boldsymbol{x}')\big)-\mathcal{T}\big(y,k_\rho(\boldsymbol{x},\boldsymbol{x}')\big) : k\in\mathcal{B}_R\big\}.$$

Each function $g\in\mathcal{F}_R$ has the form $g(\boldsymbol{x},\boldsymbol{x}',y)=\mathcal{T}\big(y,\pi_B(k)(\boldsymbol{x},\boldsymbol{x}')\big)-\mathcal{T}\big(y,k_\rho(\boldsymbol{x},\boldsymbol{x}')\big)$ with some $k\in\mathcal{B}_R$. Hence, S_1 can be bounded by

$$S_1(\boldsymbol{z},\lambda) \leq \left(\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}g(\boldsymbol{x}_i,\boldsymbol{x}_j,y_{ij})-\mathbb{E}g+\frac{1}{m^2}\sum_{i=1}^{m}g(\boldsymbol{x}_i,\boldsymbol{x}_i,y_{ii})\right). \tag{37}$$

We can easily see that $\|g\|_\infty \leq B+M^*$, and thus we have $|g-\mathbb{E}g| \leq B+M^*$. By Lemma 12, the variance-expectation condition of $\xi(z)$ is satisfied with θ given by Eq. (17) and $c_1 = C_\theta \max\{B,M^*\}^{2-\theta}$.

First, we consider the $i \neq j$ case by Lemma 11. The Lipschitz property of the ε-insensitive loss yields $\mathscr{N}(\mathcal{F}_R, \epsilon) \leq \mathscr{N}(\mathcal{B}_1, \epsilon)$. So applying Lemma 11 to the function set $\mathcal{F}_R$ with the covering number condition in Eq. (16), we have

$$\begin{aligned}
&\operatorname*{Prob}_{\boldsymbol{z}\in Z^{m\times m}} \left\{ \sup_{k\in\mathcal{F}_R} \frac{\mathbb{E}g - \frac{1}{m(m-1)}\sum_{i=1}^m \sum_{j\neq i}^m g(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij})}{\sqrt{(\mathbb{E}g)^\theta + \epsilon^\theta}} \geq 4\epsilon^{1-\frac{\theta}{2}} \right\} \\
&\qquad \leq \mathscr{N}(\mathcal{B}_1, \epsilon) \exp\left\{ - \frac{(m-1)\epsilon^{2-\theta}}{4C_\theta \max\{B, M^*\}^{2-\theta} + \frac{16}{3} M^* \epsilon^{1-\theta}} \right\} \\
&\qquad \leq \exp\left\{ C_s \Big(\frac{R}{\epsilon}\Big)^s - \frac{(m-1)\epsilon^{2-\theta}}{4C_\theta \max\{B, M^*\}^{2-\theta} + \frac{4}{3}(B+M^*)\epsilon^{1-\theta}} \right\},
\end{aligned}$$

where $\mathbb{E}g = \mathcal{E}\big(\pi_B(k)\big) - \mathcal{E}(k_\rho)$. Hence there holds a subset Z_2 of $Z^{m\times m}$ with confidence at least $1 - \delta/4$ such that

$$\sup_{k\in\mathcal{F}_R} \frac{\mathbb{E}g - \frac{1}{m(m-1)}\sum_{i=1}^m \sum_{j\neq i}^m g(\boldsymbol{x}_i, \boldsymbol{x}_j, y_{ij})}{\sqrt{(\mathbb{E}g)^\theta + \Big(\epsilon^*(m, R, \frac{\delta}{4})\Big)^\theta}} \leq 4\big(\epsilon^*(m, R, \frac{\delta}{4})\big)^{1-\frac{\theta}{2}} \quad \forall \boldsymbol{z} \in Z_2 \cap \mathscr{W}(R),$$

where $\epsilon^*(m, R, \frac{\delta}{4})$ is the smallest positive number ϵ satisfying

$$C_s \Big(\frac{R}{\epsilon}\Big)^s - \frac{m\epsilon^{2-\theta}}{4C_\theta \max\{B, M^*\}^{2-\theta} + \frac{4}{3}(M^*+B)\epsilon^{1-\theta}} = \log\frac{\delta}{4},$$

using Lemma 7.2 in Cucker and Zhou (2007), we have

$$\begin{aligned}
\epsilon^* &\leq \max\left\{ \frac{16(M^*+B)}{3m}\log\frac{4}{\delta}, \left(\frac{16C_\theta \max\{B, M^*\}^{2-\theta}}{m}\log\frac{4}{\delta}\right)^{\frac{1}{2-\theta}}, \left(\frac{16(M^*+B)}{3m} C_s R^s\right)^{\frac{1}{1+s}}, \right. \\
&\qquad \left. \left(\frac{16\max\{B, M^*\}^{2-\theta}}{3m} C_s R^s C_\theta\right)^{\frac{1}{2-\theta+s}} \right\} \\
&\leq 16(M^*+B)(C_\theta+1)\log\frac{4}{\delta} m^{-\frac{1}{2-\theta}} + 16C_s(M^*+B)(C_\theta+1) m^{-\frac{1}{2+s-\theta}} R^{\frac{s}{1+s}},
\end{aligned}$$

where we use $\frac{1}{m} \leq m^{-\frac{1}{2-\theta}}$, $M^* \geq 1$ and $(M^*)^{\frac{1}{1+s}} \leq (M^*)^{\frac{2-\theta}{2-\theta+s}}$.

Next we consider the $i = j$ case in $S_1(\boldsymbol{z}, \lambda)$. Since $\|g\|_\infty \leq B + M^*$, we have

$$\frac{1}{m^2}\sum_{i=1}^m g(\boldsymbol{x}_i, \boldsymbol{x}_i, y_{ii}) \leq \frac{1}{m}\Big(B + M^*\Big).$$

Combining above two equations, for $\boldsymbol{z} \in \mathcal{B}(R) \cap Z_2$, we have

$$\begin{aligned}
S_1(\boldsymbol{z}, \lambda) &= \Big\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big\} - \Big\{\mathcal{E}_{\boldsymbol{z}}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}_{\boldsymbol{z}}(k_\rho)\Big\} \\
&\leq 4\big[\epsilon^*(m, R, \frac{\delta}{4})\big]^{1-\frac{\theta}{2}} \sqrt{\Big(\mathcal{E}\big(\pi(k_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big)^\theta + \Big(\epsilon^*(m, R, \frac{\delta}{4})\Big)^\theta} + \frac{1}{m}(B+M^*) \\
&\leq (1 - \frac{\theta}{2}) 4^{2/(2-\theta)} \epsilon^*(m, R, \frac{\delta}{4}) + \frac{\theta}{2}\Big(\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big) + 4\epsilon^*(m, R, \frac{\delta}{4}) + \frac{1}{m}(B+M^*) \\
&\leq 20\epsilon^*(m, R, \frac{\delta}{4}) + \frac{1}{2}\Big\{\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)\Big\} + \frac{1}{m}(B+M^*),
\end{aligned}$$

where the second inequality is from Young's inequality. Finally, we complete the proof. ■

A.4 Proofs of Proposition 15

Proof Combining the bounds in Proposition 10, 13, 14, Eq. (22) and Eq. (30), the excess error $\mathcal{E}\big(\pi_B(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho)$ can be bounded by

$$\begin{aligned}\mathcal{E}\big(\pi(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho) \le{}& 2\varepsilon + 3C_0\lambda^r + 32(C_\theta+1)\Big(M^* + \mathcal{G}\sqrt{C_0}\lambda^{\frac{r-1}{2}}\Big)\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}} + \frac{2\mathcal{G}\sqrt{C_0}\lambda^{\frac{r-1}{2}}}{m}\\ &+ 640(C_\theta+1)(M^*+B)\bigg(\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}} + C_s m^{-\frac{1}{2+s-\theta}}R^{\frac{s}{1+s}}\bigg)\\ &+ \frac{2B+4M^*}{m} + 2c\Big\{(d+1)! + 2^{d+2}d^d\Big\}M^{d+1}B^{-d} + \frac{12M2^{d+1}}{m}\log\frac{4}{\delta}\,.\end{aligned} \tag{38}$$

In the next, we attempt to find a $R>0$ by giving a bound for $\lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}}$. Form the definition of $k^{(\varepsilon)}_{\boldsymbol{z},\lambda}$ in Eq. (11), we have

$$\lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}} \le \mathcal{E}_{\boldsymbol{z}}\big(k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\big) + \lambda\|k^{(\varepsilon)}_{\boldsymbol{z},\lambda}\|^2_{\underline{\mathcal{H}}} \le \mathcal{E}_{\boldsymbol{z}}(0) \le \frac{1}{m^2}\sum_{i,j=1}^{m}|Y_{ij}|\,.$$

Using Eq. (30) with confidence $1-\delta/4$, we have

$$\frac{1}{m^2}\sum_{i,j=1}^{m}|Y_{ij}| \le cM + 4M(1+\sqrt{2c})\frac{\log\frac{4}{\delta}}{\sqrt{m}} \le (3cM+4M)\log\frac{4}{\delta} := M_\delta. \tag{39}$$

This yields the measure of the set $\mathscr{W}(\frac{M_\delta}{\lambda})$ is at least $1-\delta/4$, thus the measure of the set $\mathscr{W}(\frac{M_\delta}{\lambda})\cap Z_3\cap Z_2\cap Z_1$ is at least $1-\delta$. We substitute $R=\frac{M_\delta}{\lambda}$ to Eq. (38) and let Eq. (16) with $s>0$, Eq. (15) with $0<r\le 1$, take $\lambda = m^{-\alpha}$ with $0<\alpha\le 1$ and $\alpha < \frac{1+s}{s(2+s-\theta)}$. Set $\varepsilon = m^{-\gamma}$ with $\alpha r \le \gamma \le \infty$, we have

$$\begin{aligned}\mathcal{E}\big(\pi(k^{(\varepsilon)}_{\boldsymbol{z},\lambda})\big) - \mathcal{E}(k_\rho) \le{}& 2m^{-\gamma} + 3C_0m^{-\alpha r} + \widetilde{C}_1\log\frac{4}{\delta}m^{-\frac{1}{2-\theta}} + \widetilde{C}_2\log\frac{4}{\delta}m^{-\left[\frac{1}{2-\theta}+\frac{\alpha(r-1)}{2-\theta}\right]}\\ &+ 2\mathcal{G}\sqrt{C_0}m^{-\left[\frac{\alpha(r-1)}{2}+1\right]} + \widetilde{C}_3(3cM+4M)m^{\frac{\alpha s}{1+s}-\frac{1}{2+s-\theta}}\\ &+ \frac{2B+4M^*}{m} + 2c\Big\{(d+1)! + 2^{d+2}d^d\Big\}M^{d+1}B^{-d} + \frac{12M2^{d+1}}{m}\log\frac{4}{\delta}\,,\end{aligned}$$

where $\widetilde{C}_1$, $\widetilde{C}_2$ and $\widetilde{C}_3$ are constants given by

$$\widetilde{C}_1 = (C_\theta+1)(672M^*+640B),\quad \widetilde{C}_2 = 32(C_\theta+1)\mathcal{G}\sqrt{C_0},\quad \widetilde{C}_3 = 640(C_\theta+1)(M^*+B)C_s\,.$$

Formally, we choose

$$\widetilde{C} = \max\Big\{3C_0, \widetilde{C}_1, \widetilde{C}_2, 2\mathcal{G}\sqrt{C_0}, \widetilde{C}_3(3cM+4M), 2B+4M^*\Big\} = \max\Big\{3C_0, \widetilde{C}_1, \widetilde{C}_2, 2\mathcal{G}\sqrt{C_0}, \widetilde{C}_3(3cM+4M)\Big\}\,,$$

and the power index Θ

$$\begin{aligned}\Theta &= \min\left\{\gamma, \alpha r, \frac{1}{2-\theta}, \frac{1}{2-\theta}+\frac{\alpha(r-1)}{2-\theta}, \frac{\alpha(r-1)}{2}+1, \frac{1}{2+s-\theta}-\frac{\alpha s}{1+s}\right\} \\ &= \min\left\{\alpha r, \frac{1+\alpha r-\alpha}{2-\theta}, \frac{1}{2+s-\theta}-\frac{\alpha s}{1+s}\right\},\end{aligned}$$

which concludes the proof. ■

References

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, pages 253–262, 2017.

Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209, 2013.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

Mikhail Belkin. Approximation beats concentration? an approximation view on inference with smooth radial kernels. In *Conference On Learning Theory*, pages 1348–1361, 2018.

Yoshua Bengio, Jean Francois Paiement, and Pascal Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems*, pages 177–184, 2004.

János Bognár. *Indefinite inner product spaces*. Springer, 1974.

Sabri Boughorbel, J-P Tarel, and Nozha Boujemaa. Conditionally positive definite kernels for SVM based image recognition. In *IEEE International Conference on Multimedia and Expo*, pages 113–116, 2005.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a Siamese time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–737, 1994.

Brian Bullins, Cyril Zhang, and Yi Zhang. Not-so-random features. In *International Conference on Learning Representations*, 2018.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. Learning with SGD and random features. In *Advances in Neural Information Processing Systems*, pages 10212–10223, 2018.

Dirong Chen, Qiang Wu, Yiming Ying, and Dingxuan Zhou. Support vector machine soft margin classifiers: error analysis. *Journal of Machine Learning Research*, 5(3):1143–1175, 2004.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.

A. Christmann and I. Steinwart. How SVMs can estimate quantiles and the median. In *Advances in Neural Information Processing Systems*, pages 305–312, 2007.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(2):795–828, 2012.

Nello Cristianini, Andre Elisseeff, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, pages 367–373, 2001.

Felipe Cucker and Dingxuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research*, 18(1):3520–3570, 2017.

Xialiang Dou and Tengyuan Liang. Training neural networks as learning data-adaptive kernels: Provable representation and approximation benefits. *Journal of the American Statistical Association*, pages 1–14, 2020.

Aasa Feragen, François Lauze, and Søren Hauberg. Geodesic exponential kernels: when curvature and linearity conflict. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3032–3042, 2015.

R Ganti, Nikolaos Vasiloglou, and Alexander Gray. Hyper-kernel based density estimation. In *NIPS Workshop on Automatic Selection of Optimal Kernel*, pages 1–4, 2008.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *Annals of Statistics*, 2019.

Mehmet Gonen. Bayesian efficient multiple kernel learning. In *International Conference on Machine Learning*, pages 1–8, 2012.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

Zheng-Chu Guo and Ding-Xuan Zhou. Concentration estimates for learning with unbounded sampling. *Advances in Computational Mathematics*, 38(1):207–223, 2013.

Zheng-Chu Guo, Lei Shi, and Qiang Wu. Learning theory of distributed regression with bias corrected regularization kernel network. *Journal of Machine Learning Research*, 18(1):4237–4261, 2017.

Chia-Hua Ho and Chih-Jen Lin. Large-scale linear support vector regression. *Journal of Machine Learning Research*, 13(1):3323–3348, 2012.

Jeffrey Ho, Yuchen Xie, and Baba Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *International conference on machine learning*, pages 1480–1488, 2013.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-based Pattern Recognition*, pages 84–92. Springer, 2015.

Qiao Hong, Peng Zhang, Di Wang, and Bo Zhang. An explicit nonlinear mapping for manifold learning. *IEEE Transactions on Systems Man and Cybernetics Part B*, 43(1):51–63, 2013.

Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. A divide-and-conquer solver for kernel support vector machines. In *International Conference on Machine Learning*, pages 566–574, 2014.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.

Lalit Jain, Blake Mason, and Robert Nowak. Learning low-dimensional metrics. In *Advances in Neural Information Processing Systems*, pages 4142–4150, 2017.

Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning*, pages 361–368, 2003.

Risi Kondor and Tony Jebara. Gaussian and Wishart hyper-kernels. In *Advances in Neural Information Processing Systems*, pages 729–736, 2007.

Matthieu Kowalski, Marie Szafranski, and Liva Ralaivola. Multiple indefinite kernel learning with mixed norm regularization. In *International Conference on Machine Learning*, pages 545–552, 2009.

Brian Kulis. Metric learning: a survey. *Foundations and Trends in Machine Learning*, 5(4), 2013.

Abhishek Kumar, Alexandru Niculescumizil, Koray Kavukcuoglu, and Hal Daume Iii. A binary classification framework for two-stage multiple kernel learning. In *International Conference on Machine Learning*, pages 1295–1302, 2012.

Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian Processes. In *International Conference on Learning Representations*, 2018.

Yunwen Lei, Antoine Ledent, and Marius Kloft. Sharper generalization bounds for pairwise learning. In *Advances in Neural Information Processing Systems*, 2020.

Chun-Liang Li, Wei-Cheng Chang, Youssef Mroueh, Yiming Yang, and Barnabas Poczos. Implicit kernel learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2007–2016, 2019.

Junhong Lin and Volkan Cevher. Optimal convergence for distributed learning with stochastic gradient methods and spectral algorithms. *Journal of Machine Learning Research*, 21(147):1–63, 2020.

Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan A.K. Suykens. Random features for kernel approximation: A survey in algorithms, theory, and beyond. *arXiv preprint arXiv:2004.11154*, 2020a.

Fanghui Liu, Xiaolin Huang, Chen Gong, Jie Yang, and Li Li. Learning data-adaptive non-parametric kernels. *Journal of Machine Learning Research*, 21(208):1–39, 2020b.

Fanghui Liu, Xiaolin Huang, Yingyi Chen, and Johan A.K. Suykens. Fast learning in reproducing kernel Kreĭn spaces via signed measures. In *International Conference on Artificial Intelligence and Statistics*, pages 1–11, 2021a.

Fanghui Liu, Lei Shi, Xiaolin Huang, Jie Yang, and Johan A.K. Suykens. Analysis of regularized least squares in reproducing kernel kreĭn spaces. *Machine Learning*, pages 1–29, 2021b.

Gaëlle Loosli, Stéphane Canu, and Soon Ong Cheng. Learning SVM in Kreĭn spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1204–1216, 2016.

Zhengdong Lu, Prateek Jain, and Inderjit S. Dhillon. Geometry-aware metric learning. In *International Conference on Machine Learning*, pages 673–680, 2009.

Michael Luby and Avi Wigderson. Pairwise independence and derandomization. *Foundations and Trends® in Theoretical Computer Science*, 1(4):237–301, 2006.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Shahar Mendelson and Joseph Neeman. Regularizaton in kernel learning. *Annals of Statistics*, 38(1): 526–565, 2010.

Yuichi Motai. Kernel association for classification and prediction: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 26(2):208–223, 2015.

Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, and Ivan Oseledets. Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems*, pages 9147–9156, 2018.

Dino Oglic and Thomas Gäertner. Learning in reproducing kernel Kreĭn spaces. In *International Conference on Machine Learning*, pages 3859–3867, 2018.

Cheng Soon Ong, Xavier Mary, and Alexander J. Smola. Learning with non-positive kernels. In *International Conference on Machine Learning*, pages 81–89, 2004.

Cheng Soon Ong, Alexander J. Smola, and Robert C Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6(Jul):1043–1071, 2005.

Binbin Pan, Wen Sheng Chen, Bo Chen, Chen Xu, and Jianhuang Lai. Out-of-sample extensions for non-parametric kernel methods. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):334–345, 2017.

Jeffrey Pennington, Felix Xinnan X. Yu, and Sanjiv Kumar. Spherical random features for polynomial kernels. In *Advances in Neural Information Processing Systems*, pages 1846–1854, 2015.

John C. Platt. Sequential minimal optimization: a fast algorithm for training support vector machines. In *Advances in Kernel Methods-Support Vector Learning*, pages 212–223, 1998.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007.

Anant Raj, Abhishek Kumar, Youssef Mroueh, Tom Fletcher, and Bernhard Schölkopf. Local group invariant representations via orbit embeddings. In *Artificial Intelligence and Statistics*, pages 1225–1235. PMLR, 2017.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.

Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. FALKON: an optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3891–3901, 2017.

Saburou Saitoh and Yoshihiro Sawano. *Theory of reproducing kernels and applications*. Springer, 2016.

Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419, 2007.

Frank Michael Schleif and Peter Tino. Indefinite proximity learning: a review. *Neural Computation*, 27(10):2039–2096, 2015.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.

Lei Shi, Xiaolin Huang, Zheng Tian, and Johan A.K. Suykens. Quantile regression with ℓ_1-regularization and Gaussian kernels. *Advances in Computational Mathematics*, 40(2):517–551, 2014.

Lei Shi, Xiaolin Huang, Yunlong Feng, and Johan AK Suykens. Sparse kernel regression with coefficient-based ℓ_q- regularization. *Journal of Machine Learning Research*, 20(161):1–44, 2019.

Aman Sinha and John C. Duchi. Learning kernels with random features. In *Proceedins of Advances in Neural Information Processing Systems*, pages 1298–1306, 2016.

Alex J. Smola, Zoltan L. Ovari, and Robert C. Williamson. Regularization with dot-product kernels. In *Advances in Neural Information Processing Systems*, pages 308–314, 2001.

Ingo Steinwart. Reproducing kernel hilbert spaces cannot contain all continuous functions on a compact metric space. *arXiv preprint arXiv:2002.03171*, 2020.

Ingo Steinwart and Christmann Andreas. *Support Vector Machines*. Springer Science and Business Media, 2008.

Ingo Steinwart and Andreas Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011.

Michiel Stock, Tapio Pahikkala, Antti Airola, Bernard De Baets, and Willem Waegeman. A comparative study of pairwise learning methods based on kernel ridge regression. *Neural computation*, 30(8):2245–2283, 2018.

Johan A.K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

Taiji Suzuki and Masashi Sugiyama. Fast learning rate of multiple kernel learning: Trade-off between sparsity and smoothness. In *International Conference on Artificial Intelligence and Statistics*, pages 1152–1161, 2012.

Wai Hung Tsang and Tin Yau Kwok. Efficient hyper-kernel learning using second-order cone programming. *IEEE Transactions on Neural Networks*, 17(1):48–58, 2006.

Cheng Wang and Ding-Xuan Zhou. Optimal learning rates for least squares regularized regression with unbounded sampling. *Journal of Complexity*, 27(1):55–67, 2011.

Tinghua Wang, Dongyan Zhao, and Shengfeng Tian. An overview of kernel alignment and its applications. *Artificial Intelligence Review*, 43(2):179–192, 2015.

Christopher K.I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688, 2001.

Matthew A Wright and Joseph E Gonzalez. Transformers are deep infinite-dimensional non-mercer binary kernel machines. *arXiv preprint arXiv:2106.01506*, 2021.

Qiang Wu, Yiming Ying, and Dingxuan Zhou. Learning rates of least-square regularized regression. *Foundations of Computational Mathematics*, 6(2):171–192, 2006.

Daohong Xiang, Ting Hu, and Dingxuan Zhou. Learning with varying insensitive loss. *Applied Mathematics Letters*, 24(12):2107–2109, 2011.

Daohong Xiang, Ting Hu, and Dingxuan Zhou. Approximation analysis of learning algorithms for support vector regression and quantile regression. *Journal of Applied Mathematics*, 2012(19): 1–17, 2012.

Rong Yin, Yong Liu, Lijing Lu, Weiping Wang, and Dan Meng. Divide-and-conquer learning with nyström: Optimal rate and algorithm. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 6696–6703, 2020.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592–617, 2013.