Back to the Future: Radial Basis Function Networks Revisited

Qichao Que, Mikhail Belkin
Department of Computer Science and Engineering
Ohio State University
{que, mbelkin}@cse.ohio-state.edu

Abstract

Radial Basis Function (RBF) networks are a classical family of algorithms for supervised learning. The most popular approach for training RBF networks has relied on kernel methods using regularization based on a norm in a Reproducing Kernel Hilbert Space (RKHS), which is a principled and empirically successful framework. In this paper we aim to revisit some of the older approaches to training the RBF networks from a more modern perspective. Specifically, we analyze two common regularization procedures, one based on the square norm of the coefficients in the network and another one using centers obtained by k-means clustering. We show that both of these RBF methods can be recast as certain data-dependent kernels. We provide a theoretical analysis of these methods as well as a number of experimental results, pointing out very competitive experimental performance as well as certain advantages over the standard kernel methods in terms of both flexibility (incorporating of unlabeled data) and computational complexity. Finally, our results shed light on some impressive recent successes of using soft k-means features for image recognition and other tasks.

1 Introduction

Radial Basis Function (RBF) networks are a classical family of algorithms for supervised learning. The goal of RBF is to approximate the target function through a linear combination of radial kernels, such as Gaussian (often interpreted as a two-layer neural network). Thus the output of an RBF network learning algorithm typically consists of a set of centers and weights for these functions. Proposed in [3] as a way to connect function approximation to learning, RBF networks have drawn significant attention in the machine learning community due to their strong performance and nice theoretical properties. The key aspect of any RBF network algorithm is capacity control. It is easy to see that any input data (x_i, y_i) can be fitted *exactly* by allowing every data point to be a center and choosing appropriate coefficients. That, of course, is *overfitting* and thus RBF networks need to be regularized by penalizing the coefficients and/or choosing a set of the centers of smaller cardinality then the input data. A number of regularization approaches have been proposed in the literature with various theoretical properties, computational complexity and empirical performance. By far the most popular and successful approach to regularizing RBF's has been based on *kernel machines*, such as kernel SVM's (K-SVM) or kernel regularized least squares (K-RLS) algorithm. In these approaches the function space is constrained by the norm in a Reproducing Kernel Hilbert Space (RKHS). While kernel methods are often considered to be a different class of algorithms, they are, in fact, types of RBF networks when used with a radial kernel. The kernel methods have become very popular, easily eclipsing earlier RBF algorithms, due to their elegant mathematical formulation grounded in classical functional analysis, the convex nature of optimizations involved and to their strong empirical performance.

In this paper we take a step back by revisiting two common methods for training RBF networks suggested before the runaway success of kernel machines in machine learning. Specifically, we look at regularization by the squared norm of the coefficients in an RBF network and on selecting centers through k-means clustering. Perhaps surprisingly we are able to reinterpret these algorithms as kernel methods with explicit distribution-dependent kernels. We highlight certain advantages of these approaches compared to the standard kernel methods both in terms of flexibility (by easily incorporating unlabeled data) and scaling to large datasets. In particular, our results provide a kernel interpretation for the remarkable performance of methods based on soft k-means embeddings on certain computer vision tasks [6, 12].

Our contributions could be summarized as follows.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

- We provide a theoretical analysis of RBF networks whose centers are chosen at random from the same probability distribution as the input data and which is regularized based on the l^2 norm of the coefficient vector. In particular this setting applies to the case when the set of the centers is the training set. We provide generalization bounds under the usual statistical assumptions and show that in this case the RBF algorithm is equivalent to a kernel machine with a data dependent kernel whose limit form can be explicitly established. It follows from our analysis that the asymptotic convergence rate of this methods equals to the standard rate obtained for kernel machines.

- We analyze another common form of RBF networks, where the centers are obtained from a k-means clustering algorithm. We provide a bound on the generalization error in terms of the quantization error of the output of k-means algorithm. Moreover, when k is large (as is the case in many common applications), the distribution of k-means centers can be thought of as a density itself, related to the underlying density of the data. That allows us to reinterpret the k-means RBF network in terms of another density-dependent kernel. This observation sheds light on the strong performance shown by soft k-means feature embedding used in [6, 12], which are closely related to RBF networks with a certain radial kernel. Additionally, we discuss some non-asymptotic properties of k-means related to denoising and manifold learning.

- We discuss certain advantages of RBF networks over the standard kernel methods. In particular semi-supervised learning for these RBF's is achieved naturally and without any extra hyper-parameters as unlabeled data can simply be used as centers. We discuss why adding unlabeled data can be helpful and provide experimental support for this observation.

- Finally, we provide a number of experimental results to show that RBF's provide comparable performance to the kernel machines using both the square loss and the hinge loss. We also demonstrate that the unlabeled data is indeed helpful in most settings. Additionally we show that k-means RBF can achieve regularization by simply choosing the number of centers. This is encouraging as the amount of computation required depends on the number of centers and only linearly on the number of input points.

Related Work. There is a large body of work investigating RBF networks from many different perspectives. Proposed in [3], RBF networks were introduced as a function approximation method and interpreted as artificial neural networks. Analysis of RBF networks and the connections to approximation theory were explored in [20]. Results in [17, 18] showed that any function in the functional space $L^p(\mathbb{R}^d)$ could be approximated by a RBF network arbitrarily well, under a very mild condition on the RBF function. To control the approximation power of the RBF network and avoid overfitting, [16] suggested that RBF network could be regularized by the squared norm of the coefficients (ridge regression) or subset selection. Ridge regression-based regularization has been quite popular in the literature due to its mathematical and computational simplicity. Several other related forms of regularization such as using the information curvature information in [2], have also been proposed. A number of approaches exist for selecting a subset of centers for building a parsimonious RBF network, including [5, 14, 4, 15]. Furthermore, there have been work on the statistical properties of RBF networks. In particular, the insightful work [13] investigated the generalization error of RBF networks and provided generalization guarantees in terms of the number of training data and the number of function basis in the setting of the statistical learning theory. The version of RBF considered in [13] involved a non-convex optimization over the set of centers.

While the literature on RBF's is quite large, to the best of our knowledge there have been few in-depth empirical comparisons between older methods for training RBF networks and kernel machines. That was perhaps due to the fact that without a standardized center selection procedure it was hard to produce systematic comparisons. The well-known work [23] discussed the connection between RBF's and kernel SVM and provided some experimental results on hand-written digits giving a slight advantage to SVM.

The rest of the paper is organized as follows. In Section 2, we give a brief description of ridge-RBF networks and provide a theoretical analysis. We provide a discussion of semi-supervised learning in Section 3. In Section 4, we discuss using centers obtained by k-means clustering. We provide generalization bounds, a kernel interpretation of these methods as well as some observations on the regularization effect of k-means. In Section 5 we provide a number of experiments demonstrating (a) very competitive performance of ridge-RBF to kernel methods with both square and hinge losses; (b) consistent performance improvements from unlabeled data; (c) regularizing effect of k-means.

2 RBF networks: generalization analysis

We start by formulating the problem of RBF network learning. Given a set of k centers $z_1, \dots, z_k$, an RBF network is simply a function of the form

$$f(x) = \sum_{i=1}^{k} w_i h(\|x - z_i\|). \tag{1}$$

One of the most popular choices for h is the Gaussian kernel, defined by $h(\|x-z\|) = K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2t}\right)$.

Given n training data points $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the goal of a RBF network learning algorithm is to produce a set $z_1, \ldots, z_k$ and weights w_i, such that $f(x_i) \approx y_i$ (or $\text{sign}(f(x_i)) = y_i$ for most i for classification). It is clear that regularization is necessary as it is very easy to fit any data set by a function of this form.

In this section we will concentrate on a particularly simple form of RBF, where the centers are simply the data points (labeled or unlabeled) and the regularization term equals to the sum of squared coefficients.

Choosing the loss function L (and normalizing the coefficients by $\frac{1}{n}$), we can train the model through minimizing the empirical risk on the training data:

$$\begin{aligned} \boldsymbol{w}^* &= \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^n L(f(x_i), y_i) + \frac{\lambda}{n}\sum_{i=1}^n w_i^2 \\ &\text{where } f(x) = \frac{1}{n}\sum_{j=1}^n w_j h(\|x - x_j\|) \end{aligned} \tag{2}$$

This form of RBF is quite similar to kernel machine methods such as kernel support vector machine (K-SVM) [7] and kernel regularized least square classifier (K-RLSC). The important difference is that we use $\boldsymbol{w}^T\boldsymbol{w}$ rather than $\boldsymbol{w}^T K \boldsymbol{w}$ used in kernel methods (where k is the kernel matrix computed from the data). Additionally (and rather elegantly) in kernel methods the functional form of the classifier is the result of the representer theorem. On the other hand, the optimization problem in Eqn. 2 is very direct. Moreover unlabeled data can be incorporated into Eqn. 2 by simply using the unlabeled points as additional centers. We will provide some intuition and experimental results on adding unlabeled data later on in the paper.

RBF network as an embedding. A useful interpretation of RBF's is to consider them as linear classifiers after an embedding

$$\phi : \mathbb{R}^d \to \mathbb{R}^k, \phi(x) = (h(\|x - z_1\|), \ldots, h(\|x - z_k\|))$$

For example, for the square loss, the formulation in 2 becomes ordinary ridge regression in the embedding space. This point of view is closely related to the "feature map" representation of kernel methods (note the different norm) as well as the "random kitchen sink" idea proposed in [22], which are regularized by the norm $\|\cdot\|_\infty$ on the coefficients. We also note that "soft k-means embeddings" are in fact RBF networks.

RBF network as a data-dependent kernel. For our analysis, we consider the square loss as it leads to an explicit solution to Eqn. (2), which will simplify the discussion. However the form of the kernel does not depend on the loss function.

Proposition 1. *Using the square loss $L(f(x), y) = (f(x) - y)^2$, the solution to Eqn (2) is*

$$\boldsymbol{w}^* = \left(\frac{1}{n}\boldsymbol{K}^T\boldsymbol{K} + n\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{K}^T\boldsymbol{y}.$$

where $\boldsymbol{K}$ is a $n \times n$ matrix with $\boldsymbol{K}_{ij} = K(x_i, x_j)$. The classifier function

$$f_n^*(x) = \frac{1}{n}\sum_{i=1}^n w_i K(x, x_i),$$

is equivalent to the solution to a regularized least-square kernel machine with a data-dependent kernel $\hat{K}_W$, where

$$\hat{K}_W(x, z) = \frac{1}{n}\sum_{i=1}^n K(x, x_i)K(z, x_i). \tag{3}$$

The proof is standard and is given in Section A.1 of the supplementary material. Assuming that the training data x_i are i.i.d. samples from a probability distribution p, it is easy to see that as $n \to \infty$, $\hat{K}_W(x, z)$ converges to a continuous density-dependent kernel $K_W(x, z) = \int k(x, u)k(z, u)p(u)du$. We will explore how this data dependent kernel affects performance through an example of semi-supervised learning in Section 3.

2.1 Connection to the Fredholm equation and generalization bounds

Even though l^2 regularized RBF networks were proposed long ago and perform well in practice, our understanding of these algorithms seems to be quite limited compared to the rich literature on kernel machines. Below we will provide a generalization analysis of the algorithm in a regression setting.

Given n training data, $(x_1, y_1), \ldots, (x_n, y_n)$, let us assume that x_i are i.i.d. samples from a probability distribution p and the outputs y_i are determined[1], that is $y_i = g(x_i)$. We assume the target function g is bounded by $|g(x)| \leq M$. We will also assume that the kernel $K(x, z) = h(\|x - z|)$ is positive definite.

Now consider the following continuous optimization algorithm for approximating the target function $g(x)$,

$$\begin{aligned} w^* = \min_{w\in L_p^2} \|\mathcal{K}_p w - g\|_p^2 + \lambda\|w\|_p^2, \\ \text{with the approximator function } f^* = \mathcal{K}_p w^*, \end{aligned} \tag{4}$$

The norm $\|\cdot\|_p$ is defined by $\|w\|_p = \left(\int w(x)^2 p(x)dx\right)^{\frac{1}{2}}$ and $L_p^2 = \{w, \|w\|_p^2 < \infty\}$. $\mathcal{H}$ is the RKHS with the RBF kernel K, which is assumed to be positive semi-definite. $\mathcal{K}_p : L_p^2 \to \mathcal{H}$ is an integral operator associated with the kernel K, defined by

$$\mathcal{K}_p w(x) = \int K(x, u)w(u)p(u)du.$$

It is easy to see Eqn (4) aims to approximate the target function g through an integral equation $\mathcal{K}_p w \approx g$, also

[1] The case when y is also a random variable can also be analyzed by taking $g(x) = E(y|x)$, cf [24].

known as a Fredholm equation. This approach for supervised learning by solving an integral equation with regularization is closely related to the Fredholm learning framework proposed in [21] (where an RKHS regularizer was used). By introducing this problem, we want to provide the continuous counterpart of Eqn. (2). Using f^*, we can decompose the generalization error $\|g - f_n^*\|_p$ into approximation error and estimation error.

$$\|g - f_n^*\|_p \leq \underset{\text{(Approx. Error)}}{\|g - f^*\|_p} + \underset{\text{(Est. Error)}}{\|f^* - f_n^*\|_p}. \tag{5}$$

Using the techniques in [24], we have the following proposition.

Proposition 2. *For approximation error in Eqn (5), assuming the target function g satisfies $\|{\mathcal{K}_p}^{-r} g\|_p < \infty$ for $0 < r \leq 2$, we have*

$$\|g - \mathcal{K}_p w^*\|_p \leq \lambda^{\frac{r}{2}} \|{\mathcal{K}_p}^{-r} g\|_p. \tag{6}$$

Note that the approximation error depends on the smoothness of the target function g, characterized by $\|{\mathcal{K}_p}^{-r} g\|_p < \infty$ for $0 < r \leq 2$. While this is a strong smoothness assumption, it is a standard setting in a number of learning theory papers including [24]. As usual the approximation error tends to zero as the regularization coefficient λ decreases to 0.

Now let us present the result for the estimation error (see the supplementary material for the proof)

Theorem 1. *Assuming the target function is uniformly bounded, that is $g(x) < M$ for any x, with probability at least $1 - 2e^{-\tau}$, we have*

$$\begin{aligned} &\|f_n^* - f^*\|_p \\ \leq &\frac{3\kappa^2 M(\sqrt{2\tau} + 1 + \sqrt{8\tau})}{\lambda\sqrt{n}} + \frac{4\kappa^2 M\tau}{3\lambda n} + \frac{4\kappa^3 M\tau}{3\lambda^{\frac{3}{2}} n} \end{aligned} \tag{7}$$

where $\kappa = \max_x K(x, x)$.

Combine the results from Eqn. (6) and (7), we will have the following result for the generalization error for the l^2 regularized RBF network.

Corollary 1. *Assume the target function g satisfies $\|{\mathcal{K}_p}^{-r} g\|_p < \infty$ for $0 < r \leq 2$ and $g(x) < M$ for any x. With probability at least $1 - 2e^{-\tau}$, we have*

$$\|f_n^* - g\|_p \leq C_{\tau,\kappa,M} n^{-\frac{r}{2r+4}},$$

where $C_{\tau,\kappa,M}$ is a constant depending on τ, κ and M.

Thus, as $n \to \infty$, the generalization error will converge to 0 with rate $O(n^{-\frac{r}{2r+4}})$ in probability. In particular, when g is in the range of $\mathcal{K}_p^2$, the convergence rate is $O(n^{-\frac{1}{4}})$. This is the same rate as the one for the least square kernel machine given in [24].

3 RBF networks for semi-supervised learning

In this section, will highlight the difference between RBF's and the standard kernel methods in the semi-supervised setting, which makes dependence of the classifier on the probability distribution more explicit. We first observe that using unlabeled data in the RBF setting is a simple matter of adding additional center for unlabeled points, writing $f(x) = \frac{1}{n}\sum_{i=1}^{n+m} w_i h(\|x - x_i\|)$ where m is the number of unlabeled points in Eqn. (2). While it may seem to lead to potential overfitting due to the extra parameters, this is actually not the case as the regularization penalty constrains the complexity of the function class. A version of Theorem 1 for the generalization error including unlabeled data is given in the Theorem 4 in the supplementary material.

It is easy to see that unlabeled data changes the resulting RBF classifier. A natural question of comparison to kernel machines arises. We can put $f(x) = \frac{1}{n}\sum_{j=1}^{n+m} w_j h(\|x - x_i\|)$ in the standard kernel framework, where the only difference will be using the norm $\boldsymbol{w}^T K \boldsymbol{w}$ (instead of $\boldsymbol{w}^T \boldsymbol{w}$ for RBF). However it follows from the representer theorem[2] that the output of a kernel machine will ignore the unlabeled data by putting zero weights on unlabeled points.

We will illustrate this difference by a simple example. Consider a classification problem with the (marginal) data distribution $p(x) = N(\mathbf{0}, \text{diag}([9, 1]))$. Given two labeled points, positive example $x_p = (-4, 3)$ and negative example $x_n = (4, -3)$, consider two candidate classifier functions using the kernel $K(x, z) = \exp\left(-\frac{\|x-z\|^2}{4}\right)$,

1) $f_1(x) = K(x, x_p) - K(x, x_n), x_p = (-4, 3), x_n = (4, -3)$;

2) $f_2(x) = K(x, z_p) - K(x, z_n), z_p = (-5, 0), z_n = (5, 0)$.

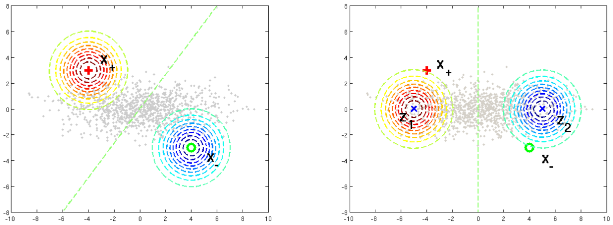

Figure 1: Contours and classification boundaries for f_1 (left) and f_2 (right). Two labeled points x_+ and x_-, grey unlabeled points are sampled from p. Note that $\|f_1\|_{\mathcal{H}} = \|f_2\|_{\mathcal{H}}$, however $\|f_1\|_{\mathcal{H}_W} \gg \|f_2\|_{\mathcal{H}_W}$

From Figure 1, it is clear that both f_1 and f_2 have 0 empirical risk on the two labeled data x_p and x_n. However, their

[2] Observe that the solution of the kernel machine is optimal over the whole RKHS space. As f belongs to the RKHS, the extra centers will make no difference in the final form of the solution.

norms are different in the standard RKHS $\mathcal{H}$ corresponding to K and the data-dependent RKHS $\mathcal{H}_W$ corresponding to the kernel K_W in Eqn. (3).

First, we observe that f_1, f_2 that

$$\begin{aligned}\|f_1\|_{\mathcal{H}}^2 &= K(x_p, x_p) + K(x_n, x_n) - 2K(x_n, x_p)\\ =&K(z_p, z_p) + K(z_n, z_n) - 2K(z_n, z_p) = \|f_2\|_{\mathcal{H}}^2.\end{aligned}$$

Thus, f_1 and f_2 are equivalently good solutions from the point of a kernel machine with kernel K, as both the empirical risk and regularization term are the same.

Estimating the RBF-related norm $\mathcal{H}_W$ as a bit trickier and we omit the details here, and just give the result

$$\frac{\|f_1\|_{\mathcal{H}_W}}{\|f_2\|_{\mathcal{H}_W}} \approx \frac{\frac{1}{p(x_p)} + \frac{1}{p(x_n)}}{\frac{1}{p(z_p)} + \frac{1}{p(z_p)}} \approx 54.6$$

The solution f_1 has a much higher regularization penalty and in the RBF framework would select f_2 over f_1.

This density dependence may or may not be desirable depending on the assumptions but is generally consistent with density and manifold-based semi-supervised learning. RBF networks prefer boundaries orthogonal to the local principal components of the density. In practice there seems to be a small but consistent improvement from unlabeled data without any additional hyper-parameters, see Section 5 for the experiments.

4 k-means RBF networks

From a practical point of view, the efficiency of RBF network directly depends on the number of centers, which determines how much computational power we need for each data point. Even though including both labeled and unlabeled data as basis could potentially improve performance, it also makes it impractical for large scale data sets. Thus, we need to find a way to choose a smaller set of centers, while retaining performance as much as possible. Historically, people have used the k-means centers for RBF network, which usually performs quite well in practice. Recent research showed that non-linear features learned using k-means were quite effective for a number of problems, including visual object recognition and optical character recognition [6, 12]. In this section, we will discuss why k-means are a good choice for the centers of RBF networks, and how the asymptotic properties of the RBF algorithm will be affected by the k-means quantization.

4.1 k-means RBF algorithm

As a method for vector quantization, k-means splits the data set into k subsets such that each data point is close to the center of its cluster. More formally, given a data set of size n, $X = \{x_1, \dots, x_n\}$, it seeks to find k centers $\mathcal{C}_k = \{c_1, \dots, c_k\}$, by minimizing the quantization error,

$$\begin{aligned}\mathcal{C}_k &= \arg\min_{\mathcal{C}, |\mathcal{C}|=k.} Q_k(\mathcal{C}),\\ \text{where } Q_k(\mathcal{C}) &:= \sum_{i=1}^{n} \min_{c\in\mathcal{C}} \|x_i - c\|^2.\end{aligned} \tag{8}$$

The clusters, defined by

$$C_i = \{x_j, \|x_j - c_i\| = \min_{c\in\mathcal{C}_k} \|x_j - c\|, 1 \le j \le n\}$$

form a k-partition of the data set. Solving the problem exactly is difficult, since the existing work [11] shows that even the planar case is NP-hard. The most common method used in practice is the greedy iterative Lloyd's algorithm proposed in [10], which is guaranteed to converge to a local minimum. Moreover, the quantization loss of k-means after the intelligent initialization provided by k-means++ [1] is shown to be within a factor of $O(\log k)$ of the optimal loss $Q_k(\mathcal{C}_k)$.

As k-means provides a concise representation of the data, it is natural to replace the training set with its k-means centers for radial basis functions. It gives us a classifier that could be evaluated more efficiently than a full network. In this section, we consider two types of k-means RBF networks:

(1) Weighted k-means network. Given the cluster weights, $P_n(C_i) = \#\{x_j \in C_i\}/n$, the classifier is learned by

$$\begin{aligned}\boldsymbol{w}_{k,p}^* &= \arg\min_{\boldsymbol{w}\in\mathbb{R}^k} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \frac{\lambda}{k}\sum_{i=1}^{k} P_n(C_i) w_i^2\\ \text{where } f(x) &= \sum_{i=1}^{k} w_i h(\|x - c_i\|) P_n(C_i).\end{aligned}$$

The output classifier is denoted by $f_{k,p}$.

(2) Unweighted k-means network, trained using

$$\begin{aligned}\boldsymbol{w}_k^* &= \arg\min_{\boldsymbol{w}\in\mathbb{R}^k} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \frac{\lambda}{k}\sum_{i=1}^{k} w_i^2\\ \text{where } f(x) &= \sum_{i=1}^{k} w_i h(\|x - c_i\|),\end{aligned}$$

whose output is denoted by f_k.

We note that the difference is in the density weighting of the regularization term. Most applications use standard (unweighted k-means networks), however weighted k-means networks turn out to be easier to analyze and seem to give similar performance in practice.

Remark: We note that k-means RBF is equivalent to linear classification/regression using "soft k-means features", that is applying the embedding $x \to (h(\|x - c_1\|), \dots, h(\|x - c_k\|))$.

We also note, the solution in Proposition 1 also applies to the case of f_k, as the only difference is the choice of the centers. For $f_{k,p}$, the solution is slightly different as extra weights $P_n(C_i)$ are involved. For square loss, the classifier weights for $f_{k,p}$ will be

$$\boldsymbol{w}^*_{k,p} = \boldsymbol{K}^T(\boldsymbol{KPK}^T + \lambda \boldsymbol{I})^{-1}\boldsymbol{y},$$

where $\boldsymbol{K}$ is a $n \times k$ matrix with $\boldsymbol{K}_{ij} = K(x_i, c_j)$ and $\boldsymbol{P}$ is a diagonal matrix of size $k \times k$ with $\boldsymbol{P}_{ii} = P_n(C_i)$. Similar to our analysis before, this classifier is equivalent to a kernel machine that uses a data dependent kernel $\hat{K}_W(x,z) = \sum_{i=1}^k K(x,c_i)K(z,c_i)P(C_i)$. As more clusters are used, $\hat{K}_W$ converges to a density dependent kernel, $K_W(x,z) = \int K(x,u)K(z,u)p(u)du$, which is the same as for the case of RBF networks considered earlier.

For the standard (unweighted) k-means, the empirical distribution of the centers converges to a distribution that is closely related to p as $k \to \infty$, which allows us to also write a form for the limiting kernel. The details are given below.

4.2 Generalization bounds via quantization error and kernel interpretation of k-means RBF

Under the setting of k-means RBF networks, it is interesting to see how the quantization process affects the generalization error and how it relates to the RBF network that uses the whole training data as the set of centers. First, let us provide an analysis for the generalization error of the k-means RBF network. For this analysis, we will consider the weighted k-means network, since the k-means with the cluster weights provide an estimator of the distribution density. In particular, $f^*_{k,p}$ will converge to the f^* from Eqn. (4) and the estimation error $\|f^* - f^*_{k,p}\|$ could be bounded in terms of the quantization loss. We give this result in the following theorem.

Theorem 2. *Suppose the target function is uniformly bounded $g(x) \leq M$ for any x, and the RBF kernel K is translation invariant such that $K(x,z) = h(\|x-z\|^2)$ with a monotonic decreasing function h satisfying the Lipschitz condition: $|h(v) - h(u)| \leq L|u-v|$. For the estimation error $\|f^* - f^*_{k,p}\|_p$, we have*

$$\|f^*_{k,p} - f^*\|_p \leq \left(\frac{\kappa^2 + \kappa^{\frac{5}{2}}}{\lambda} + \frac{2\kappa^3}{\lambda^{\frac{3}{2}}}\right)\frac{\sqrt{2\tau}M}{\sqrt{n}} + 8L\left(\frac{\kappa^2}{\lambda} + \frac{\kappa^3}{\lambda^{\frac{3}{2}}}\right)MQ_k(\mathcal{C})$$

with probability at least $1 - 2e^{-\tau}$.

In addition to the error term depending on n, the estimation error bound for k-means RBF network also contains a term that depends on the quantization error $Q_k(\mathcal{C}_k)$. As k approaches to n, the quantization error decreases to 0, the k-means RBF network could be viewed as an approximation of the one use the full data set.

For unweighted k-means networks, giving an explicit analysis for the generalization error is more subtle. However, we could still understand their behavior by looking at the limit of the data dependent kernel induced by the network. First, the following theorem summarizes the limit of the empirical distribution of the k-means centers.

Theorem 3. *[8] Suppose p is absolutely continuous w.r.t. Lebesgue measure in $\mathbb{R}^d$ and $E\|X\|^{2+\delta} < \infty$ for some $\delta > 0$. Let $(\mathcal{C}_{p,k})_{k\geq 1}$ be the solution to,*

$$\begin{aligned} &\mathcal{C}_{p,k} = \arg\min_{\mathcal{C}, |\mathcal{C}|=k.} Q_{p,k}(\mathcal{C}), \\ &\text{where } Q_{p,k}(\mathcal{C}) := \int \min_{c\in\mathcal{C}} \|x-c\|^2 p(x)dx. \end{aligned} \qquad (9)$$

Let μ_k be the empirical measure of the cluster centers $\mu_k = \frac{1}{|\mathcal{C}_{p,k}|}\sum_{c\in\mathcal{C}_{p,k}} \mathbf{1}_c$. As $k \to \infty$ we have

$$\mu_k \xrightarrow{D} p_2,$$

where p_2 is a distribution with density $p_2(x) = \frac{p(x)^{\frac{d}{d+2}}}{\int p(x)^{\frac{d}{d+2}}dx}$. Here $\xrightarrow{D}$ denotes convergence in distribution.

There are several notable aspects to this result. First, the empirical measure of k-means centers converges to a probability distribution despite the deterministic process to learn the centers. Second, if dimension d of the space is sufficiently high such that $\frac{d}{d+2} \approx 1$, the centers can be viewed as a density estimator of the original density. However, as $\frac{d}{d+2} < 1$ this estimator over-emphasizes the areas with low density. Interestingly, this tendency can be counteracted by a finite sample phenomenon as k-means tends to shrink ”short” directions. We will discuss this in more details in Section 4.3.

Thus, RBF networks using k-means centers without weights should be converging to the same Fredholm equation in Eqn. (4) while using a slightly different integral operator $\mathcal{K}_{p_2}$. Hence the induced data dependent kernel for the unweighted k-means network converges to K'_W given by

$$K'_W(x,z) = \int K(x,u)K(z,u)p_2(u)du.$$

Due to the close relationship between p and p_2, it performs similarly to the weighted k-means RBF network.

4.3 Denoising effect of k-means

A k-means RBF network gives us a compact model for the data, that makes large scale learning possible for RBF networks. On the other hand, it also introduces extra error due to the quantization. Regarding this trade-off between computational cost and the learning error, in this section we would like to give some intuition for the empirical choice of k based on our observations. It turns out that k-means clustering has local denoising properties related to manifold learning.

As we know, the Lloyd's algorithm for k-means is essentially an expectation maximization (EM) algorithm for the equally weighted spherical Gaussian Mixture Model (GMM) with infinitesimal variance [9]. In other words, we can think of k-means as a GMM with small variances. In this sense, the distribution of the k-means centers could be considered a *deconvolution* of the data distribution with the Gaussian kernel, whose variance is on the order of the average distance between the neighboring cluster centers. That distance is on the order of $O(k^{-\frac{1}{d}})$, where d is the dimension. Thus, the distribution of k-means centers will remove all directions whose variance is less than $O(k^{\frac{1}{d}})$ and shrink all other directions locally by that amount. This can be viewed as a form of denoising/manifold learning.

We can use the example of a circular distribution with Gaussian noise in Figure 2 to illustrate this point. When k is too small (the left panel), the original distribution is not well approximated well by the means. As k becomes larger (the center panel) the set of the means ignores the "noisy" thin local direction thus learning the manifold, the circle. When k is even larger (the right panel) the noise suppressing property becomes insignificant and the set of means can be viewed as a density approximation. Thus for certain data distribution, with a properly chosen k, the k-means RBF network will perform as well as the full RBF netowrk, but with less computation overhead. We explore this regularization effects of k-means by an experiment in Section 5.3.

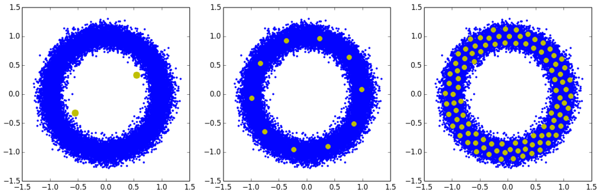

Figure 2: Left: $k = 2$; Middle: $k = 10$; Right: $k = 100$.

5 Experiments

5.1 Kernel machines and RBF networks

There have been few recent comparisons between kernel machines and RBF networks. In this section, we compare these methods on a number of datasets demonstrating RBF networks perform comparably to kernel machines. We also explore the performance of RBF's in semi-supervised problem. We choose several benchmark datasets for our experiments, including (1) handwritten digits recognition, including MNIST, and MNIST variants (2) street view house number (SVHN) recognition; (3) Adult, Cover Type and Cod-RNA data sets form the UCI repository.

Supervised learning. The original data set is split into three parts: a training set, a validation set (a randomly chosen subset of 10%) and a testing set. For ridge RBF, the training set is also used as the centers. The parameters, regularization coefficient λ and kernel width t were chosen based on the performance on the validation set. The final performance was be evaluated on the testing set, which is shown in Table 1.

	K-SVM	K-RLSC	RBFN-hinge	RBFN-LS
MNIST	1.5	**1.32**	1.72	1.35
MNIST-rand	15.8	**13.7**	16.6	14.3
MNIST-img	23.4	20.9	23.4	**20.8**
SVHN 60k	20.5	**18.7**	24.2	18.8
Adult	**14.5**	15.6	15.8	15.6
Cover Type	28.4	27.8	28.0	**27.6**
Cod-RNA	4.60	**3.55**	3.94	3.73

Table 1: Classification **Errors** (%) for *supervised learning* with whole training data using K-SVM, K-RLSC, RBF network with hinge loss and least-square loss.

Semi-supervised learning. When both labeled and unlabeled are used as centers, RBF becomes a semi-supervised learning algorithm. To explore the performance of RBF network for this situation, we randomly choose 100 labeled points from the original training set and use the whole set as unlabeled. The final performance is evaluated on the held-out testing set, which is shown in Table 2.

	K-SVM	K-RLSC	RBFN-hinge	RBFN-LS
MNIST	26.8	26.0	27.4	**23.3**
MNIST rand	51.4	48.5	**35.9**	38.3
MNIST img	59.2	52.7	63.1	**51.0**
SVHN 60k	75.5	73.1	79.5	**72.4**
Adult	18.8	19.1	19.5	**18.4**
Cover Type	58.3	58.7	**57.3**	57.9
Cod-RNA	6.62	**6.30**	7.83	7.12

Table 2: Classification **Errors** (%) for *semi-supervised learning*, with 100 labeled points, using K-SVM, K-RLSC, ridge RBF network with hinge loss and least-square loss.

Performance improvements from using unlabeled data are consistent, appearing in all but one data sets. Notably, unlike other semi-supervised methods (admittedly with potentially superior performance) no extra hyperparameters are needed.

5.2 RBF network with k-means centers

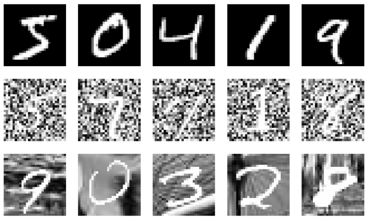

Figure 3: MNIST, MNIST-rand, MNIST-img

Using RBF network for supervised or semi-supervised learning is appealing considering its performance. However its use on large data sets (similarly to that of kernel machines) is hindered by the computational complexity. k-means RBF network provides a more compact model than a full RBF network and can lead to far more efficient algorithms with competitive performance. Moreover k-means can serve as regularization allowing to optimize computation and minimize the error simultaneously.

We explore performance of the k-means RBF networks. It is interesting to note that for the original MNIST data, the centers tend to smooth out the quirky styles in some of the digits, and represent average digits in the data set. For MNIST-rand data, the background are samples from a uniform distribution in $[0, 1]^{784}$ around the clean digits, while the digits usually come from low-dimensional manifolds. k-means alleviates the noise for this classic manifold+noise distribution. Finally, the background for MNIST-img comes from the distribution of nature images, which also form a low-dimensional manifold themselves. Thus, k-means recovers not only the digits manifold, but the manifold for the natural images leading to decreased performance in our classification task.

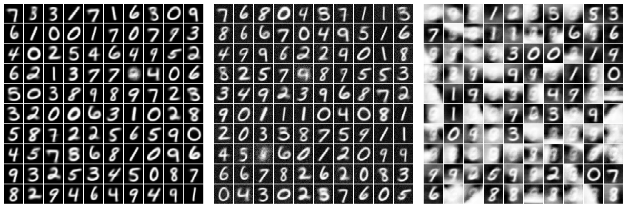

Figure 4: k-means centers represented as images for MNIST (left); MNIST-rand (center); MNIST-img (right).

Now let us apply the k-means RBF network to these three variations of MNIST and fix $k = 1000$ for k-means. For our experiments, the images are preprocessed so that all values are in the range of $[0, 1]$. The k-means are trained on the whole training+testing dataset. The kernel width t are chosen from $\{300, 100, \ldots, 1\}$ and the regularization parameter λ are chosen from $1^0, \ldots, 10^{-8}$. Kernel Regularized Least-square classifier (K-RLSC) is used as the benchmark. To better evaluate the effect of k-means, we also consider the RBF network using k random sampled points as centers, denoted by RBF-k-rand. The classification errors are shown in Table 3. We observe that RBF network using k-means RBF performs consistently better than the RBF with k points chosen at random from the data. That is consistent with Theorem 2 showing that the learning error could be bounded in terms of the quantization error, that is minimized by k-means. While the performance of k-means RBF is generally worse than that of the full network, we note that the number of centers $k = 1000$ is far smaller than the data size.

	K-RLSC	RBF-rand	k-means RBFN	k-means RBFN(w)
MNIST	**1.32**	4.0	3.3	3.3
MNIST-rand	13.6	22.3	10.9	**10.6**
MNIST-img	**21.2**	26.4	25.3	25.5
SVHN-60k	**18.7**	26.2	26.2	26.1
Adult	15.4	**15.0**	15.7	15.6
Cover Type	**27.2**	37.9	35.7	35.7
Cod-RNA	**3.55**	3.87	3.99	4.05

Table 3: Classification **Errors** (%) for K-RLSC, RBF network with 1000 randomly selected points as centers, k-means centers, unweighted and weighted, $k = 1000$.

5.3 Regularization effect of k-means

As we discussed in Section 4.3, the number of centers used in k-means also serves as a kind of regularization. To explore this effect, we fix small $\lambda = 10^{-10}$ and choose different k from $\{62, 125, 250, 500, 1000, 2000, 4000\}$. The classification error on MNIST-rand of K-RLSC (the constant line) and k-means kernel are plotted in Figure 5. The optimal performance is achieved at a certain number of centers and deteriorates if more centers are used.

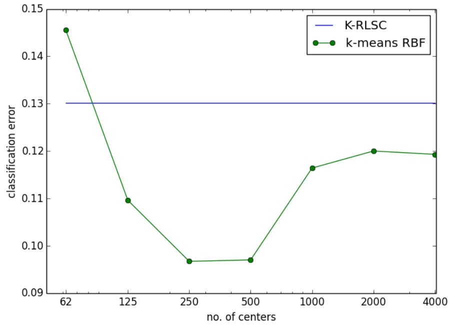

Figure 5: The regularization effect of k-means, based on the classification error of RBFs network on MNIST-rand.

Acknowledgments

We thank Vikas Sindhwani for encouraging us to understand the difference between l^2 and RKHS regularization. The work was supported by NSF Grants 1117707, 1422830, 1550757.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[2] C. Bishop. Improving the generalization properties of radial basis function neural networks. *Neural computation*, 3(4):579–588, 1991.

[3] D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document, 1988.

[4] S. Chen, E. Chng, and K. Alkadhimi. Regularized orthogonal least squares algorithm for constructing radial basis function networks. *International Journal of Control*, 64(5):829–837, 1996.

[5] S. Chen, C. F. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, 2(2):302–309, 1991.

[6] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions*. Springer, 2000.

[9] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proceedings of the 29st International Conference on Machine Learning (ICML 2012)*, 2012.

[10] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[11] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. *Theoretical Computer Science*, 442:13–21, 2012.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS 2011, Workshop on deep learning and unsupervised feature learning*, 2011.

[13] P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842, 1996.

[14] M. J. Orr. Regularised centre recruitment in radial basis function networks. In *Centre for Cognitive Science, Edinburgh University*. Citeseer, 1993.

[15] M. J. Orr. Regularization in the selection of radial basis function centers. *Neural computation*, 7(3):606–623, 1995.

[16] M. J. Orr et al. Introduction to radial basis function networks, 1996.

[17] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[18] J. Park and I. W. Sandberg. Approximation and radial-basis-function networks. *Neural computation*, 5(2):305–316, 1993.

[19] I. Pinelis. An approach to inequalities for the distributions of infinite-dimensional martingales. In *Probability in Banach Spaces, 8: Proceedings of the Eighth International Conference*, pages 128–134. Springer, 1992.

[20] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[21] Q. Que, M. Belkin, and Y. Wang. Learning with fredholm kernels. In *Advances in Neural Information Processing Systems 28 (NIPS 2014)*, pages 2951–2959, 2014.

[22] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems, (NIPS 2009)*, pages 1313–1320, 2009.

[23] B. Schölkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *Signal Processing, IEEE Transactions on*, 45(11):2758–2765, 1997.

[24] S. Smale and D.-X. Zhou. Shannon sampling ii: Connections to learning theory. *Applied and Computational Harmonic Analysis*, 19(3):285–302, 2005.

A Proofs to the Theorems

A.1 Proof to Proposition 1

Proof. Given the kernel matrix $\boldsymbol{K}$ of size $n \times n$ with $\boldsymbol{K}_{ij} = K(x_i, x_j)$, the optimization problem in Eqn (2) could be reformulated as

$$\boldsymbol{w}^* = \min_{\boldsymbol{w} \in \mathbb{R}^n} \frac{1}{n} \left(\frac{1}{n} \boldsymbol{K}\boldsymbol{w} - \boldsymbol{y} \right)^T \left(\frac{1}{n} \boldsymbol{K}\boldsymbol{w} - \boldsymbol{y} \right) + \lambda \frac{\boldsymbol{w}^T \boldsymbol{w}}{n}.$$

It is an unconstrained quadratic optimization problem, whose solution is given by the following equation,

$$\left(\frac{1}{n^2} \boldsymbol{K}^T \boldsymbol{K} + \lambda \mathcal{I} \right) \boldsymbol{w}^* = \frac{1}{n} \boldsymbol{K}^T \boldsymbol{y}.$$

Thus, $\boldsymbol{w}^* = \left(\frac{1}{n} \boldsymbol{K}^T \boldsymbol{K} + n\lambda \mathcal{I} \right)^{-1} \boldsymbol{K}^T \boldsymbol{y}$. Define the kernel $\hat{K}_W$ as in the proposition, the final classifier function will be

$$\begin{aligned} f_n^*(x) =& \frac{1}{n} \sum_{i=1}^n K(x, x_i) w_i^* = \sum_{i=1}^n \hat{K}_W(x, x_i) \alpha_i, \\ & \text{where } \boldsymbol{\alpha} = \left(\frac{1}{n} \boldsymbol{K}\boldsymbol{K}^T + n\lambda \mathcal{I} \right)^{-1} \boldsymbol{y}. \end{aligned} \qquad (10)$$

And $\frac{1}{n} \boldsymbol{K}\boldsymbol{K}^T$ is the kernel matrix for $\hat{K}_W$ because $(\frac{1}{n} \boldsymbol{K}\boldsymbol{K}^T)_{ij} = \hat{K}_W(x_i, x_j)$. The solution classifier f_n^* in Eqn (10) is then equivalent to the regularized least square kernel machine using kernel $\hat{K}_W$. □

A.2 Proof to Theorem 1 and Theorem 2

Before giving the proof, let us first introduce several important objects that will be used in the proof.

Suppose K is a positive semi-definite kernel function, which is associated with a reproducing kernel Hilbert space (RKHS), denoted by $\mathcal{H}$. Given n data points, $X = \{x_1, \ldots, x_n\}$, define a sample operator $S_X : \mathcal{H} \to l_n^2$

$$S_X f = [f(x_1), \ldots, f(x_n)]. \qquad (11)$$

This operator was introduced in [24], which provided a very simple framework to prove the consistency of kernel methods for function reconstruction.

Now suppose the inner product in l_n^2 is defined by $\langle \boldsymbol{y}, \boldsymbol{z} \rangle_n = \frac{1}{n} \sum_{i=1}^n y_i z_i$. Then the conjugate operator of S_X, $S_X^* : l_n^2 \to \mathcal{H}$ is

$$S_X^* \boldsymbol{y}(x) = \frac{1}{n} \sum_{i=1}^n K(x, x_i) y_i.$$

And we denote S_Z for the sampling operator corresponding to the training data for k-means, $Z = \{z_1, \ldots, z_m\}$. In our theorem in the main text, we use the training data for training the k-means, thus $S_Z = S_X$.

Given k points $\mathcal{C} = \{c_1, \ldots, c_k\}$, and the corresponding Voronoi diagram C_i, we can have a discrete square summable space l_k^2. For $\boldsymbol{u}, \boldsymbol{v} \in l_k^2$, we have that $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_k = \sum_{i=1}^k u_i v_i P(C_i)$, where $P(C_i) = \frac{m_i}{m}$ and $m_i = \#\{x_j \in C_i, 1 \leq j \leq m\}$. We can also define a sample operator on these k points, denoted by $S_k : \mathcal{H} \to l_k^2$,

$$S_k f = [f(c_1), \ldots, f(c_k)]. \qquad (12)$$

Due to the different inner product we used for l_k^2, we will have its conjugate operator defined by

$$S_k^* \boldsymbol{v}(x) = \sum_{i=1}^k P(C_i) K(x, c_i) v_i.$$

To simplify our notation, we denote that $\mathcal{A} = \mathcal{K}_{p_p}$ and $\mathcal{A}_X = S_X^* S_X$. Before giving the proof for Theorem 1, we need the following two lemmas.

Lemma 1. *Suppose $\{x_1, \ldots, x_n\}$ are i.i.d. samples from a probability distribution p and $\mathcal{A}$ and $\mathcal{A}_X$ are the operators defined above. For function h with $\|h\|_\infty < \infty$, with probability at least $1 - 2e^{-\tau}$, we have*

$$\|(\mathcal{A}^2 - \mathcal{A}_X^2)h\|_{\mathcal{H}} \leq \frac{\kappa^{\frac{3}{2}} \|h\|_p}{\sqrt{n}} \left(\sqrt{2\tau} + 1 + \sqrt{8\tau} \right) + \frac{4\kappa^{\frac{3}{2}} \|h\|_\infty}{3n} \tau.$$

Lemma 2. *Suppose the kernel function K satisfies the condition we given the Theorem 2. We have*

$$\|S_Z^* S_Z - S_k^* S_k\|_{HS} \leq 8L\kappa Q_k(\mathcal{C}).$$

Now we can give the proof for Proposition 2.

Proof. Firstly, let us given the bound for the approximation error. Note that we can have the closed form solution for Eqn (4), $f^* = \mathcal{K}_p h^* = \left(\mathcal{K}_p{}^2 + \lambda \mathcal{I} \right)^{-1} \mathcal{K}_p{}^2 g$. As K is a positive semi-definite kernel, the operator $\mathcal{K}_p$ has positive eigven values, $\lambda_1, \lambda_2, \ldots$, and its eigenvectors ψ_i form a complete orthogonal basis for L_p^2. Since $\|\mathcal{K}_p{}^{-r} g\|_p < \infty$, there exists a sequence $d_1, d_2, \ldots$ such that $\mathcal{K}_p{}^{-r} g = \sum_{i=1}^\infty d_i \psi_i$ and $\sum_{i=1}^\infty d_i^2 < \infty$. Thus g could be represented as $g = \sum_{i=1}^\infty \lambda_i^r d_i \psi$ and $f^* = \sum_{i=1}^\infty \frac{\lambda_i^2}{\lambda_i^2 + \lambda} \lambda_i^r d_i \psi$. And we have

$$g - f^* = \sum_{i=1}^\infty \frac{\lambda}{\lambda_i^2 + \lambda} \lambda_i^r d_i \psi.$$

Thus,

$$\begin{aligned} \|g - f^*\|_p^2 =& \sum_{i=1}^\infty \left(\frac{\lambda}{\lambda_i^2 + \lambda} \lambda_i^r d_i \right)^2 \\ =& \lambda^r \sum_{i=1}^\infty \left(\frac{\lambda}{\lambda_i^2 + \lambda} \right)^{2-r} \left(\frac{\lambda_i^2}{\lambda_i^2 + \lambda} \right)^r d_i^2 \\ \leq& \lambda^r \|\mathcal{K}_p{}^{-r} g\|_p^2. \end{aligned}$$

□

Now let us give the proof for the estimation error $\|f_n^* - f^*\|_{\mathcal{H}}$ in Theorem 1.

Proof. Use the operator we define before, we can rewrite the f_n^* as follows,

$$\begin{aligned} f_n^* =& S_X^*((S_X S_X^*)^2 + \lambda\mathcal{I})^{-1} S_X S_X^* S_X g \\ =& ((S_X^* S_X)^2 + \lambda\mathcal{I})^{-1} (S_X^* S_X)^2 g \end{aligned}$$

To simplify our notation, we let $\mathcal{A} = \mathcal{K}_p$ and $\mathcal{A}_X = S_X^* S_X$.

$$\begin{aligned} &\|f_n^* - f^*\|_{\mathcal{H}} \\ =& \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}_X^2 g - \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}^2 g \right\|_{\mathcal{H}} \\ \leq& \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}_X^2 g - \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}^2 g \right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}^2 g - \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \mathcal{A}^2 g \right\|_{\mathcal{H}} \\ =& \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \left(\mathcal{A}^2 - \mathcal{A}_X^2\right) g \right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \left(\mathcal{A}_X^2 - \mathcal{A}^2\right) f^* \right\|_{\mathcal{H}}. \\ \leq& \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \right\| \left\| \left(\mathcal{A}^2 - \mathcal{A}_X^2\right) g \right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}_X^2 + \lambda\mathcal{I}\right)^{-1} \right\| \left\| \left(\mathcal{A}_X^2 - \mathcal{A}^2\right) f^* \right\|_{\mathcal{H}}. \end{aligned} \tag{13}$$

It is not hard to see that $\|\left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\| \leq \frac{1}{\lambda}$. We can use Lemma 3 to bound $\left\|\left(\mathcal{A}^2 - \mathcal{A}_X^2\right) g\right\|_{\mathcal{H}}$ and $\left\|\left(\mathcal{A}^2 - \mathcal{A}_X^2\right) f^*\right\|_{\mathcal{H}}$.

For g, we have $\|g\|_p \leq \|g\|_\infty < M$. For f^*, as $f^* = \mathcal{K}_p w^*$ optimizes Eqn. (4), letting $w = 0$, we have

$$\|f^* - g\|_p^2 + \lambda\|w^*\|_p^2 \leq \|g\|_p^2 \leq M^2.$$

So $\|f^*\|_p \leq 2M$ and $\|f^*\|_{\mathcal{H}} \leq \frac{\kappa^{\frac{1}{2}} M}{\sqrt{\lambda}}$. It implies that $\|f^*\|_\infty \leq \frac{\kappa M}{\sqrt{\lambda}}$.

Thus,

$$\begin{aligned} &\|f_n^* - f^*\|_{\mathcal{H}} \\ \leq& \frac{3\kappa^{\frac{3}{2}} M(\sqrt{2\tau} + 1 + \sqrt{8\tau})}{\lambda\sqrt{n}} + \frac{4\kappa^{\frac{3}{2}} M\tau}{3\lambda n} + \frac{4\kappa^{\frac{5}{2}} M\tau}{3\lambda^{\frac{3}{2}} n}. \end{aligned}$$

Using the fact that $\|f\|_p \leq \kappa^{\frac{1}{2}} \|f\|_{\mathcal{H}}$, we will get the theorem. □

Similarly, we can also prove the Theorem 2 about the estimation error for the RBF network with k-means centers.

Proof. Using the operator we defined before, we can rewrite f_k^* as follows,

$$\begin{aligned} f_k^* =& S_k^* \boldsymbol{w}^* = S_k^* \left(S_k S_X^* S_X S_k^* + \lambda\mathcal{I}\right)^{-1} S_k S_X^* S_X g \\ =& \left(S_k^* S_k S_X^* S_X + \lambda\mathcal{I}\right)^{-1} S_k^* S_k S_X^* S_X g \end{aligned}$$

In addition to the notation used in previous proof, we also denote that $\mathcal{A}_k = S_k^* S_k$. Similar with Eqn (19), we have

$$\begin{aligned} &\|f_k^* - f^*\| \\ \leq& \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \right\| \|\mathcal{A}^2 - \mathcal{A}_k \mathcal{A}_X\| \|f_k^*\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \right\| \left\| \left(\mathcal{A}_k \mathcal{A}_X - \mathcal{A}^2\right) g \right\|_{\mathcal{H}}. \end{aligned}$$

It is not hard to see that $\|\left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\| \leq \frac{1}{\lambda}$.

For $\mathcal{A}^2 - \mathcal{A}_k \mathcal{A}_X$, we have

$$\begin{aligned} &\mathcal{A}^2 - \mathcal{A}_k \mathcal{A}_X \\ =& (\mathcal{A} - \mathcal{A}_X + \mathcal{A}_X - \mathcal{A}_k)\mathcal{A} + \mathcal{A}_k(\mathcal{A} - \mathcal{A}_X) \end{aligned}$$

Note that the only difference from the previous proof is the extra term $\|\mathcal{A}_X - \mathcal{A}_k\|$, which is bounded by

$$\|\mathcal{A}_X - \mathcal{A}_k\| \leq \|\mathcal{A}_X - \mathcal{A}_k\|_{HS} \leq 8L\kappa Q_k(\mathcal{C})$$

Thus, with probability at least $1 - 2e^{-\tau}$,

$$\|\mathcal{A}^2 - \mathcal{A}_k \mathcal{A}_X\| \leq \kappa^2\sqrt{2\tau}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + 8L\kappa^2 Q_k(\mathcal{C})$$

and

$$\begin{aligned} &\|(\mathcal{A}^2 - \mathcal{A}_k \mathcal{A}_X)g\|_{\mathcal{H}} \\ \leq& \frac{\kappa^2\sqrt{2\tau}\|g\|_\infty}{\sqrt{n}} + \frac{\kappa^{\frac{3}{2}}\sqrt{2\tau}}{\sqrt{n}}\|g\|_\infty + 8L\kappa^{\frac{3}{2}} Q_k(\mathcal{C})\|g\|_\infty \end{aligned}$$

And using the same process for bounding the $\|f_{m,n}^*\|$, we have

$$\|f_k^*\|_{\mathcal{H}} \leq \frac{\sqrt{\kappa}}{\sqrt{\lambda}} M$$

Thus,

$$\begin{aligned} &\|f_k^* - f^*\|_{\mathcal{H}} \\ \leq& \frac{1}{\lambda}\left(\frac{\kappa^2\sqrt{2\tau}}{\sqrt{n}} + \frac{\kappa^{\frac{3}{2}}\sqrt{2\tau}}{\sqrt{n}} + 8L\kappa^{\frac{3}{2}} Q_k(\mathcal{C})\right) M \\ &+ \frac{1}{\lambda}\left(\frac{\kappa^2\sqrt{2\tau}}{\sqrt{n}} + \frac{\kappa^2\sqrt{2\tau}}{\sqrt{n}} + 8L\kappa^2 Q_k(\mathcal{C})\right) \frac{\kappa^{\frac{1}{2}}}{\lambda^{\frac{1}{2}}} M \\ =& \left(\frac{\kappa^{\frac{3}{2}} + \kappa^2}{\lambda} + \frac{2\kappa^{\frac{5}{2}}}{\lambda^{\frac{3}{2}}}\right) \frac{\sqrt{2\tau} M}{\sqrt{n}} \\ &+ 8L\left(\frac{\kappa^{\frac{3}{2}}}{\lambda} + \frac{\kappa^{\frac{5}{2}}}{\lambda^{\frac{3}{2}}}\right) M Q_k(\mathcal{C}) \end{aligned}$$

□

A.3 Proof to Lemma 2

Proof. We know that for a positive semi-definite kernel K, there exists a feature map Φ, that maps each point x to an element in RKHS. By the property of reproducing kernel, $K(x, z) = \langle \Phi(x), \Phi(z) \rangle$. That is, the kernel function $K(x, z)$ is the inner product of $\Phi(x)$ and $\Phi(z)$. Now we can define the outer product operator $\Phi(x) \otimes \Phi(x) : \mathcal{H} \to \mathcal{H}$, where $(\Phi(x) \otimes \Phi(x)) f = f(x)\Phi(x)$.

Using this notation, we can redefine the $S_Z^* S_Z$ and $S_k^* S_k$,

$$\begin{aligned} S_Z^* S_Z =& \frac{1}{n} \sum_{i=1}^{m} \Phi(z_i) \otimes \Phi(z_i) \\ S_k^* S_k =& \sum_{i=1}^{k} P(C_i) \Phi(c_i) \otimes \Phi(c_i). \end{aligned}$$

To bound $\|S_Z^* S_Z - S_k^* S_k\|_{HS}$, we have

$$\begin{aligned} &\|S_Z^* S_Z - S_k^* S_k\|_{HS}^2 \\ =& \left\| \frac{1}{m} \sum_{i=1}^{m} \Phi(z_i) \otimes \Phi(z_i) - \sum_{i=1}^{k} P(C_i) \Phi(c_i) \otimes \Phi(c_i) \right\|_{HS}^2 \\ =& \left\| \frac{1}{m} \sum_{i=1}^{k} \sum_{z_j \in C_i} \left(\Phi(z_j) \otimes \Phi(z_j) - \Phi(c_i) \otimes \Phi(c_i) \right) \right\|_{HS}^2 \end{aligned}$$

We could use the counterpart of Jensen's inequality for the Hilbert-Schmidt norm of an operator, which gives us,

$$\begin{aligned} &\|S_Z^* S_Z - S_k^* S_k\|_{HS}^2 \\ \leq& \frac{1}{m} \sum_{i=1}^{k} \sum_{z_j \in C_i} \|\Phi(z_j) \otimes \Phi(z_j) - \Phi(c_i) \otimes \Phi(c_i)\|_{HS}^2 . \end{aligned} \tag{14}$$

Suppose $e_1, e_2, \ldots$ is a orthogonal basis of $\mathcal{H}$. By the definition of Hilbert-Schmidt norm, for any x and z, we have

$$\begin{aligned} &\|\Phi(x) \otimes \Phi(x) - \Phi(z) \otimes \Phi(z)\|_{HS}^2 \\ =& \sum_i \|(\Phi(x) \otimes \Phi(x) - \Phi(z) \otimes \Phi(z)) e_i\|_{\mathcal{H}}^2 \\ =& \sum_i \|e_i(x)\Phi(x) - e_i(z)\Phi(z)\|_{\mathcal{H}}^2 \\ \leq& 2 \sum_i \|e_i(x)\Phi(x) - e_i(z)\Phi(x)\|_{\mathcal{H}}^2 \\ &+ 2 \sum_i \|e_i(z)\Phi(x) - e_i(z)\Phi(z)\|_{\mathcal{H}}^2 \qquad (15) \\ =& 2 \sum_i \|\langle e_i, \Phi(x) - \Phi(z) \rangle \Phi(x)\|_{\mathcal{H}}^2 \\ &+ 2 \sum_i \|\langle e_i, \Phi(z) \rangle (\Phi(x) - \Phi(z))\|_{\mathcal{H}}^2 \\ \leq& 4 \|\Phi(x)\|_{\mathcal{H}}^2 \|\Phi(x) - \Phi(z)\|_{\mathcal{H}}^2 \\ =& 4\kappa \|\Phi(x) - \Phi(z)\|_{\mathcal{H}}^2, \end{aligned}$$

where $\kappa = \max_x \|\Phi(x)\|_{\mathcal{H}}^2 = \max_x K(x, x)$. And because of the property of RKHS, we have that

$$\|\Phi(x) - \Phi(z)\|_{\mathcal{H}}^2 = K(x, x) + K(z, z) - 2K(x, z).$$

As we assumed, K is a translation invariant kernel such that $K(x, z) = f(\|x - z\|^2)$ for a monotonic decreasing function f that satisfies the Lipschitz condition, we have that

$$|K(x, x) - K(x, z)| = \left| f(0) - f(\|x - z\|^2) \right| \leq L\|x - z\|^2.$$

And we have the same inequality for $K(z, z) - K(x, z)$. Thus, we have

$$\|\Phi(x) - \Phi(z)\|_{\mathcal{H}}^2 \leq 2L\|x - z\|_2^2 \tag{16}$$

Combine the results in Eqn (14), (15) and (16), we have the Lemma. □

A.4 Proof to Lemma 3

Lemma 3.

$$\begin{aligned} \|(\mathcal{A}^2 - \mathcal{A}_X^2)h\|_{\mathcal{H}} \leq& \frac{\kappa^{\frac{3}{2}} \|h\|_p}{\sqrt{n}} \left(\sqrt{2\tau} + 1 + \sqrt{8\tau} \right) \\ &+ \frac{4\kappa^{\frac{3}{2}} \|h\|_\infty}{3n} \tau. \end{aligned}$$

Proof. For $\|(\mathcal{A}^2 - \mathcal{A}_X^2)h\|_{\mathcal{H}}$, we have

$$\begin{aligned} &\|(\mathcal{A}^2 - \mathcal{A}_X^2)h\|_{\mathcal{H}} \\ \leq& \|\mathcal{A} - \mathcal{A}_X\| \|\mathcal{A}h\|_{\mathcal{H}} + \|\mathcal{A}_X\| \|(\mathcal{A} - \mathcal{A}_X)h\|_{\mathcal{H}} \end{aligned}$$

Firstly, we have $\|\mathcal{A}h\|_{\mathcal{H}} \leq \kappa^{\frac{1}{2}} \|h\|_p$.

By the Lemma 3 in [24] and concentration inequality in RKHS in [19], we have

$$\|(\mathcal{A} - \mathcal{A}_X)h\|_{\mathcal{H}} \leq \frac{4\kappa^{\frac{1}{2}} \|h\|_\infty}{3n} \tau + \frac{\kappa^{\frac{1}{2}} \|h\|_p}{\sqrt{n}} \left(1 + \sqrt{8\tau} \right)$$

and $\|\mathcal{A} - \mathcal{A}_X\| \leq \frac{\kappa\sqrt{2\tau}}{\sqrt{n}}$ with probability at least $1 - 2e^{-\tau}$.

Thus, for any $g \in L_p^2$ with $\|g\|_\infty < \infty$, with probability at least $1 - 2e^{-\tau}$, we have

$$\begin{aligned} \|(\mathcal{A}^2 - \mathcal{A}_X^2)h\|_{\mathcal{H}} \leq& \frac{\kappa^{\frac{3}{2}} \|h\|_p}{\sqrt{n}} \left(\sqrt{2\tau} + 1 + \sqrt{8\tau} \right) \\ &+ \frac{4\kappa^{\frac{3}{2}} \|h\|_\infty}{3n} \tau. \end{aligned}$$

□

B Estimation error for more general RBF network

In many applications of machine learning, we have both labeled and unlabeled points available for training. Suppose we have m points $\{x_i, 1 \leq i \leq m\}$, and the first n are labeled $\{(x_i, y_i), 1 \leq i \leq n\}$. We can also include the unlabeled for the centers in RBF network. In this case, the classifier function has the form,

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} w_i K(x, x_i),$$

where $\{x_i, 1 \le i \le m\}$ including both labeled and unlabeled points. The weights $(w_1, \dots, w_m)$ could be learned by minimizing the regularized empirical loss on the training data of size n. More specifically,

$$\begin{aligned} \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \frac{\lambda}{m}\sum_{i=1}^{m} w_i^2 \\ \text{where } \; f(x) = \frac{1}{m}\sum_{j=1}^{m} w_j h(\|x - x_j\|). \end{aligned} \tag{17}$$

Denote the output classifier as $f_{m,n}$.

We will have the following theorem regarding the estimation error of this RBF network.

Theorem 4. *Suppose the target function is uniformly bounded, $g(x) < M$ for any x and assume that all the data points $\{x_i, 1 \le i \le m\}$ are i.i.d. samples from a probability distribution p. With probability at least $1 - 2e^{-\tau}$, we have*

$$\begin{aligned} \|f^* - f^*_{m,n}\|_p \le \left(\frac{1}{\lambda^{\frac{3}{2}}}\kappa^3 + \frac{1}{\lambda}\kappa^2\right)\frac{\sqrt{2\tau}M}{\sqrt{m}} \\ + \left(\frac{1}{\lambda^{\frac{3}{2}}}\kappa^3 + \frac{1}{\lambda}\kappa^{\frac{5}{2}}\right)\frac{\sqrt{2\tau}M}{\sqrt{n}} \end{aligned} \tag{18}$$

where $\kappa = \max_x K(x, x)$.

Proof. Use the operator we define before, we can rewrite the $f^*_{m,n}$ as follows,

$$\begin{aligned} f^*_{m,n} =& S_Z^*(S_Z S_X^* S_X S_Z^* + \lambda\mathcal{I})^{-1} S_Z S_X^* S_X g \\ =& (S_Z^* S_Z S_X^* S_X + \lambda\mathcal{I})^{-1} S_Z^* S_Z S_X^* S_X g \end{aligned}$$

To simplify our notation, we let $\mathcal{A} = \mathcal{K}_p$ and $\mathcal{A}_X = S_X^* S_X$ and $\mathcal{A}_Z = S_Z^* S_Z$.

$$\begin{aligned} &\|f^*_{m,n} - f^*\| \\ =& \left\| (\mathcal{A}_Z\mathcal{A}_X + \lambda\mathcal{I})^{-1}\mathcal{A}_Z\mathcal{A}_X g - \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\mathcal{A}^2 g \right\|_{\mathcal{H}} \\ \le& \left\| (\mathcal{A}_Z\mathcal{A}_X + \lambda\mathcal{I})^{-1}\mathcal{A}_Z\mathcal{A}_X g - \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\mathcal{A}_Z\mathcal{A}_X g \right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\mathcal{A}_Z\mathcal{A}_X g - \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\mathcal{A}^2 g \right\|_{\mathcal{H}} \\ =& \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\left(\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X\right) f^*_{m,n} \right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1}\left(\mathcal{A}_Z\mathcal{A}_X - \mathcal{A}^2\right) g \right\|_{\mathcal{H}}. \\ \le& \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \right\| \left\|\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X\right\| \left\|f^*_{m,n}\right\|_{\mathcal{H}} \\ &+ \left\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \right\| \left\| \left(\mathcal{A}_Z\mathcal{A}_X - \mathcal{A}^2\right) g \right\|_{\mathcal{H}}. \end{aligned} \tag{19}$$

It is not hard to see that $\| \left(\mathcal{A}^2 + \lambda\mathcal{I}\right)^{-1} \| \le \frac{1}{\lambda}$.

For $\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X$, we have

$$\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X = (\mathcal{A} - \mathcal{A}_Z)\mathcal{A} + \mathcal{A}_Z(\mathcal{A} - \mathcal{A}_X)$$

Thus,

$$\begin{aligned} &\|(\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X)g\|_{\mathcal{H}} \\ \le& \|\mathcal{A} - \mathcal{A}_Z\|\|\mathcal{A}g\|_{\mathcal{H}} + \|\mathcal{A}_Z\|\|(\mathcal{A} - \mathcal{A}_X)g\|_{\mathcal{H}} \end{aligned}$$

Firstly, we have $\|\mathcal{A}g\|_{\mathcal{H}} \le \kappa^{\frac{1}{2}}M$. By the concentration inequality in RKHS [19], we have $\|(\mathcal{A} - \mathcal{A}_X)g\| \le \frac{\kappa\sqrt{2\tau}M}{\sqrt{n}}$ and $\|\mathcal{A} - \mathcal{A}_Z\| \le \frac{\kappa\sqrt{2\tau}}{\sqrt{m}}$ with probability at least $1 - 2e^{-\tau}$. With probability at least $1 - 2e^{-\tau}$, we have

$$\|(\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X)g\|_{\mathcal{H}} \le \kappa^{\frac{3}{2}}\sqrt{2\tau}M\left(\frac{1}{\sqrt{m}} + \frac{\kappa^{\frac{1}{2}}}{\sqrt{n}}\right).$$

Similarly, we have

$$\|\mathcal{A}^2 - \mathcal{A}_Z\mathcal{A}_X\| \le \kappa^2\sqrt{2\tau}\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}}\right)$$

Now let us look at $\|f^*_{m,n}\|_{\mathcal{H}}$.

$$\begin{aligned} \|f^*_{m,n}\|^2_{\mathcal{H}} =& \sum_{i,j=1}^{m} w_i w_j P_m(C_i)P_m(C_j)K(z_i, z_j) \\ \le& \kappa\sum_{i,j=1}^{m}\frac{w_i^2 + w_j^2}{2}P_m(C_i)P_m(C_j) = \kappa\sum_{i=1}^{m} w_i^2 P_m(C_i) \end{aligned}$$

Note that $\sum_{i=1}^m w_i^2 P_m(C_i)$ is the regularizer in Eqn. (). As $f^*_{m,n}$ is the optimizer to Eqn. (), we have $\|f^*_{m,n}\|_{\mathcal{H}} \le \frac{\sqrt{\kappa}M}{\sqrt{\lambda}}$.

Thus,

$$\begin{aligned} &\|f^*_{m,n} - f^*\|_{\mathcal{H}} \\ \le& \frac{1}{\lambda^{\frac{3}{2}}}\kappa^{\frac{5}{2}}\sqrt{2\tau}M\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}}\right) \\ &+ \frac{1}{\lambda}\kappa^{\frac{3}{2}}\sqrt{2\tau}M\left(\frac{1}{\sqrt{m}} + \frac{\kappa^{\frac{1}{2}}}{\sqrt{n}}\right) \\ =& \left(\frac{1}{\lambda^{\frac{3}{2}}}\kappa^{\frac{5}{2}} + \frac{1}{\lambda}\kappa^{\frac{3}{2}}\right)\frac{\sqrt{2\tau}M}{\sqrt{m}} \\ &+ \left(\frac{1}{\lambda^{\frac{3}{2}}}\kappa^{\frac{5}{2}} + \frac{1}{\lambda}\kappa^2\right)\frac{\sqrt{2\tau}M}{\sqrt{n}} \end{aligned}$$

Using the fact that $\|\cdot\|_p \le \kappa^{\frac{1}{2}}\|\cdot\|_{\mathcal{H}}$, we have the theorem. □