Scaling up Kernel Ridge Regression via Locality Sensitive Hashing

Michael Kapralov
EPFL

Navid Nouri
EPFL

Ilya Razenshteyn
Microsoft Research

Ameya Velingker
Google Research

Amir Zandieh
EPFL

Abstract

Random binning features, introduced in the seminal paper of Rahimi and Recht '07, are an efficient method for approximating a kernel matrix using locality sensitive hashing. Random binning features provide a very simple and efficient way to approximate the Laplace kernel but unfortunately do not apply to many important classes of kernels, notably ones that generate *smooth* Gaussian processes, such as the Gaussian kernel and Matérn kernel. In this paper we introduce a simple *weighted* version of random binning features, and show that the corresponding kernel function generates Gaussian processes of any desired smoothness. We show that our weighted random binning features provide a spectral approximation to the corresponding kernel matrix, leading to efficient algorithms for kernel ridge regression. Experiments on large scale regression datasets show that our method outperforms the accuracy of random Fourier features method.

1 Introduction

Kernel methods are a powerful framework for applying non-parametric modeling techniques to a number of problems in statistics and machine learning, such as ridge regression, SVM, PCA, etc. While kernel methods have been well studied and are capable of achieving excellent empirical results, they often pose scalability challenges as they operate on the *kernel matrix* (Gram matrix) K of the data, whose size scales up quadratically in the number of training instances. Thus, much work has focused on scaling up kernel methods by producing suitable approximations to the kernel or its underlying kernel matrix.

One such approach for scaling up kernel methods was shown by Rahimi and Recht (2007), who showed how to approximate positive definite shift-invariant kernels using *random binning features*. The idea is to partition an input space into randomly shifted grids and map input points into bins such that the probability that two input points $\mathbf{x}$ and $\mathbf{y}$ are mapped to the same bin is proportional to $k(\mathbf{x}, \mathbf{y})$. This enables one to get an estimate for $k(\mathbf{x}, \mathbf{y})$ by counting the number of times $\mathbf{x}$ and $\mathbf{y}$ are binned together.

The above approach can also be viewed in the context of *locality sensitive hashing (LSH)* (Indyk and Motwani, 1998; Har-Peled et al., 2012), an algorithmic technique that hashes elements of an input space into "buckets" such that similar input items are hashed into the same buckets with high probability. More specifically, the hash collision probability between two items is desired to be proportional to the similarity index of the items, i.e., collisions should be more likely for more similar items. LSH has found practical uses for a number of problems such as nearest neighbor search, clustering, etc. The random binning features of Rahimi and Recht (2007) can be viewed as an LSH scheme in which the similarity measure is the kernel.

Rahimi and Recht (2007) show that random binning features yield an *unbiased* estimator $\widetilde{k}(\mathbf{x}, \mathbf{y})$ for $k(\mathbf{x}, \mathbf{y})$, provided that k satisfies certain conditions. They also establish *point-wise* concentration of $\widetilde{k}$ to k, but in many numerical linear algebra applications, point-wise concentration is insufficient. On the other hand, *spectral guarantees* for the kernel matrix K, whose $(\mathbf{x}, \mathbf{y})$-entry is given by $k(\mathbf{x}, \mathbf{y})$, are a popular sufficient condition that guarantees various statistical and algorithmic implications. One such guarantee is

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

captured by the (regularized) *oblivious subspace embedding (OSE)* property, as stated below.

Definition 1 (Oblivious subspace embedding (OSE))**.** Given $\epsilon, \delta, \lambda > 0$, and the positive semi-definite matrix $K \in \mathbb{R}^{n \times n}$, an $(\epsilon, \delta, \lambda)$-*oblivious subspace embedding (OSE)* for this kernel matrix is a distribution $\mathcal{D}$ over $n \times n$ matrices $\widetilde{K}$ such that with probability at least $1 - \delta$,

$$(1-\epsilon)(K+\lambda I) \preceq \widetilde{K} + \lambda I \preceq (1+\epsilon)(K+\lambda I). \quad (1)$$

Kernel Ridge Regression (KRR). One kernel method for which OSE has algorithmic implications is the problem of *kernel ridge regression* (KRR), which we focus on in this work. In KRR, one is given labeled training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ and a *regularization parameter* $\lambda > 0$, and the response of an input vector $\mathbf{x}$ is estimated as follows:

$$\overline{\eta}(\mathbf{x}) = \sum_{j=1}^{n} k(\mathbf{x}_j, \mathbf{x})\alpha_j,$$

where $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_n)^T$ is the solution of the equation $(K + \lambda I_n)\boldsymbol{\alpha} = \mathbf{y}$, where $\mathbf{y} = (y_1 \cdots y_n)^T$ and I_n is the $n \times n$ identity matrix. Solving this matrix equation generally requires $\Theta(n^3)$ time and $\Theta(n^2)$ memory, which is impractical for large datasets. Thus, the design of scalable methods for KRR and other kernel methods has been the focus of much recent research (Bach, 2013; Caponnetto and Vito, 2007; Alaoui and Mahoney, 2015; Zhang et al., 2015; Musco and Musco, 2017; Avron et al., 2017a,b).

The OSE property for $\widetilde{K}$ is useful because it allows $\widetilde{K} + \lambda I_n$ to be used as an effective preconditioner for the solution of the aforementioned matrix equation, while enabling one to bound the excess risk (Avron et al., 2017b). Thus, the approach we take is to find a new class of estimators that satisfies the OSE property while enabling fast matrix-vector computation.

WLSH estimators. Our main contribution is to formulate a new class of estimators, which we term *Weighted LSH (WLSH) estimators*, that generalize the random binning features of Rahimi and Recht (2007) and applies to a wider range of kernels. More specifically, given a probability density function $p(\cdot)$ with non-negative support over $\mathbb{R}^d$ and a *bucket-shaping function* $f(\cdot)$ (see discussion below), we can define a kernel with kernel matrix $K \in \mathbb{R}^{n \times n}$ as well as a corresponding WLSH estimator.

Our first main theorem shows that appropriately many independent instances $\widetilde{K}^1, \widetilde{K}^2, \ldots, \widetilde{K}^m$ of the WLSH estimator yield an OSE $\widetilde{K}$ for K:

$$\widetilde{K} = \frac{1}{m}\sum_{s=1}^{m} \widetilde{K}^s \quad (2)$$

Theorem 2 (Main Theorem, *informal version of Theorem 11*)**.** *Let $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$ be a collection of points and $\epsilon, \lambda > 0$. For any $p(\cdot)$ and any f supported on $[-1/2, 1/2]$ with $\|f\|_2 = 1$, the distribution $\widetilde{K}$ given by (2) is an $(\epsilon, 1/\mathrm{poly}(n), \lambda)$-OSE for K, provided that the number of independent instances of the WLSH estimator is $m = \Omega\left(\frac{\|f^{\otimes d}\|_\infty^2}{\epsilon^2} \cdot \frac{n}{\lambda} \cdot \log n\right)$.*

Our WLSH estimator reduces to standard random binning features when the *bucket-shaping function* f is chosen to be a *rectangle function* rect supported on $[-1/2, 1/2]$. However, the generalization of allowing different bucket-shaping functions f enables the estimator to be applied to a wider range of kernels, which we discuss below.

Standard random binning features work only for certain classes of shift-invariant kernels $k(\cdot)$ that satisfy a convex decomposition property (Rahimi and Recht, 2007). The Laplace kernel, given by $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \exp(-|\mathbf{x} - \mathbf{y}|)$, is an important example of such a kernel. However, note that the Laplace kernel does not satisfy *smoothness*, which is often a desired property. Indeed, the limitation of random binning features to non-smooth kernels is inherent, as any suitable shift-invariant kernel $k(\cdot)$ must have the property that $1 - k(\cdot)$ satisfies the triangle inequality (Charikar, 2002). This precludes the possibility of using random binning features to approximate any monotonically decreasing smooth kernel that is twice differentiable.

The non-smoothness limitation arises from the fact that the bins in random binning are discontinuous at the edges, as the shape of the corresponding bins is a rectangle. Our approach circumvents this limitation by generalizing random binning features to an estimator that allows "soft" buckets with smoother edges (specified by the bucket-shaping function f in Theorem 11). This allows us to construct new families of smooth kernels that can be estimated using our WLSH estimators but are not amenable to standard random binning features.

We complement Theorem 11 with a lower bound showing that the number of instances of the WLSH Estimator in Theorem 11 is essentially tight:

Theorem 3 (Main Theorem, *informal version of Theorem 12*)**.** *Let $p(w) = we^{-w}$ be the PDF for the Gamma distribution, and let $f(\cdot) = \mathrm{rect}(\cdot)$ be the bucket-shaping function. For any $\lambda > 0$, $d \geq 1$, and $n \geq 8\lambda$, there exists a dataset $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$ such that in order for $\widetilde{K}$ given by (2) to be an $(\epsilon, 1/n, \lambda)$-OSE for K with $\epsilon \leq 1/6$, one requires $m = \Omega\left(\frac{1}{\epsilon^2} \cdot \frac{n}{\lambda} \cdot \log n\right)$ independent instances of the WLSH estimator.*

Furthermore, our WLSH estimator allows $\widetilde{K}$ to

be stored with little memory while supporting fast matrix-vector multiplication, which allows it to be suitable for KRR. In this direction, we conduct a number of experiments on various datasets that show the accuracy and speed of approximate KRR using our WLSH kernels and estimator compared to exact KRR and other popular approximation methods. The results show that our WLSH-based method produces better accuracy than the popular method of random Fourier features on large datasets while still offering favorable running times. We additionally present experiments showing the performance of our WLSH-based kernel family for learning Gaussian processes through KRR.

1.1 Related work

Another line of work for producing low-rank approximations to kernel matrices is the Nyström method. A number of works have sought to improve the method using leverage score sampling, risk inflation bounds, etc. (Bach, 2013; Alaoui and Mahoney, 2015; Rudi et al., 2015; Musco and Musco, 2017). Although there has been much work on kernel approximation sketches that achieve the optimal target dimension, e.g., Nyström sampling (Musco and Musco, 2017), all such methods that are known are *data-dependent*, barring any strong assumptions on the kernel matrix. *Data-oblivious* approaches, on the other hand, have the advantage of being implementable in distributed settings. WLSH estimators (and random binning features), being OSEs, fall into this paradigm.

There are a number of works on devising OSEs. Most of these are related to the technique of *Random Fourier features*, which was also introduced by Rahimi and Recht (2007) and provides a popular data-oblivious approach for kernel approximation. Avron et al. (2017b) showed that a modification of Random Fourier features yields provably better target dimension. Ahle et al. (2020) improved upon this result and were able to embed the Gaussian kernel in Euclidean space with a target dimension that is not exponential in the dimension of the dataset. However, some Gaussian processes that arise in practice are less smooth than those arising from Gaussian kernels, and the result of Ahle et al. (2020) does not extend to the Laplace kernel [1] or Matérn kernels.

[1] One can trivially use the result of Ahle et al. (2020) for Laplace kernels by using a trivial embedding of l1 norms into l2, but this results in a blowup in dimension that is impractical

2 Preliminaries

In this section we introduce notations and present basic definitions and claims.

The *Fourier transform* of a continuous function $g : \mathbb{R}^d \to \mathbb{C}$ in $L_1(\mathbb{R}^n)$ is defined to be the function $\mathcal{F}g : \mathbb{R}^d \to \mathbb{C}$ given by $(\mathcal{F}g)(\boldsymbol{\xi}) = \int_{\mathbb{R}^d} g(\mathbf{t}) e^{-2\pi i \mathbf{t}^\top \boldsymbol{\xi}}\, d\mathbf{t}$. We also sometimes use the notation $\widehat{g}$ for the Fourier transform of g. We often informally refer to g as representing the function in *time domain* and $\hat{g}$ as representing the function in *frequency domain*. The original function g can also be obtained from $\hat{g}$ by the *inverse Fourier transform*: $g(\mathbf{t}) = \int_{\mathbb{R}^d} \widehat{g}(\boldsymbol{\xi}) e^{2\pi i \boldsymbol{\xi}^\top \mathbf{t}}\, d\boldsymbol{\xi}$. The *convolution* of two functions $g : \mathbb{R}^d \to \mathbb{C}$ and $h : \mathbb{R}^d \to \mathbb{C}$ is defined to be the function $(h * g) : \mathbb{R}^d \to \mathbb{C}$ given by $(h * g)(\boldsymbol{\eta}) = \int_{\mathbb{R}^d} h(\mathbf{t}) g(\boldsymbol{\eta} - \mathbf{t})\, d\mathbf{t}$ for $\boldsymbol{\eta} \in \mathbb{R}^d$. We use δ_d to denote the d-dimensional *Dirac delta function*.

We now define the *rectangle function* (boxcar).

Definition 4 (Rectangle Function)**.** For any $a > 0$ we define the 1-dimensional *rectangle function* $\mathrm{rect}_a : \mathbb{R} \to \mathbb{C}$ as

$$\mathrm{rect}_a(x) = \begin{cases} 0 & \text{if } |x| > a/2 \\ 1 & \text{if } |x| \le a/2 \end{cases}.$$

If $a = 1$, we omit the subscript and just write rect.

For any vector $\mathbf{w} = (w_1, w_2, \ldots, w_d)^\top$ we use the notation $[0, \mathbf{w}]$ to denote the set $[0, w_1] \times [0, w_2] \times \cdots \times [0, w_d]$. Moreover, if $\mathbf{j} = (j_1, j_2, \ldots, j_d)^\top$, then we use the notation $\mathbf{jw} = (j_1 w_1, j_2 w_2, \ldots, j_d w_d)^\top$ and $\mathbf{j}/\mathbf{w} = (j_1/w_1, j_2/w_2, \ldots, j_d/w_d)^\top$. Also, for any function $f : \mathbb{R} \to \mathbb{R}$ the notation $f^{\otimes d}$ denotes the function $f^{\otimes d} : \mathbb{R}^d \to \mathbb{R}$, defined as $f^{\otimes d}(\mathbf{x}) = \prod_{l=1}^d f(x_l)$ for every $\mathbf{x} \in \mathbb{R}^d$.

3 Weighted Locality Sensitive Hashing (WLSH) estimator

In this section we first provide background on random binning features and Locality Sensitive Hashing and then define our WLSH estimator in Section 3.1 and prove its smoothness properties in Section 3.2. *Random binning features* were introduced by Rahimi and Recht (2007) as an estimator for a certain class of kernel functions such as the Laplace kernel. The main building block of this estimator is a *Locality Sensitive Hashing (LSH)* family, defined as follows:

Definition 5 (Locality Sensitive Hash Family)**.** For any positive integer d, we define the Locality Sensitive Hash (LSH) family $\mathcal{H}$ as the collection of hash functions, $\mathcal{H} := \left\{h_{\mathbf{w},\mathbf{z}}(\cdot) : \mathbf{w} \in \mathbb{R}^d_+, \mathbf{z} \in [0, \mathbf{w}]\right\}$, where the

LSH function $h_{\mathbf{w},\mathbf{z}} : \mathbb{R}^d \to \mathbb{Z}^d$ is given by,

$$[h_{\mathbf{w},\mathbf{z}}(\mathbf{x})]_l = \text{round}\left(\frac{x_l - z_l}{w_l}\right), \tag{3}$$

for every $l \in [d]$ and $\mathbf{x} \in \mathbb{R}^d$. The parameters of the LSH functions in this family are distributed as follows: $\mathbf{w} = (w_1, w_2, \ldots, w_d)^\top$ is a random vector with iid entries $w_1, w_2, \ldots, w_d \sim p(w)$ for some probability distribution $p(\cdot)$ with non-negative support and $\mathbf{z}$ is a uniform random vector in $[0, \mathbf{w}]$.

Random binning features are given by the following estimator:

$$\widetilde{k}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } h_{\mathbf{w},\mathbf{z}}(\mathbf{x}) = h_{\mathbf{w},\mathbf{z}}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $h_{\mathbf{w},\mathbf{z}}(\mathbf{x}) \sim \mathcal{H}$ is an LSH function. Note that the expectation of this estimator is equal to the collision probability of the LSH function $h_{\mathbf{w},\mathbf{z}}$, i.e., $\mathbb{E}\left[\widetilde{k}(\mathbf{x}, \mathbf{y})\right] = \Pr_{h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}}[h_{\mathbf{w},\mathbf{z}}(\mathbf{x}) = h_{\mathbf{w},\mathbf{z}}(\mathbf{y})]$. It is shown in Rahimi and Recht (2007) that if $\mathcal{H}$ is the LSH family given in Definition 5 with $p(w) = we^{-w}$ (Gamma distribution), then the collision probability of two points $\mathbf{x}, \mathbf{y}$ is $\mathbb{E}_{h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}}\left[\widetilde{k}(\mathbf{x}, \mathbf{y})\right] = e^{-\|\mathbf{x}-\mathbf{y}\|_1}$, which is the Laplace kernel. The Laplace kernel is non-smooth due to the discontinuity of its derivative at the origin. There is a great deal of interest in using smooth kernels in many machine learning applications (Srinivas et al., 2009). By changing the distribution over the LSH family $\mathcal{H}$ via varying the PDF $p(w)$, one can obtain the random binning feature estimator for some class of kernels. One might hope to find a distribution over $\mathcal{H}$ such that $\mathbb{E}_{h_{\mathbf{w},\mathbf{z}}(\cdot) \sim \mathcal{H}}[\widetilde{k}(\mathbf{x}, \mathbf{y})]$ gives a smooth kernel such as the Squared exponential kernel or Matérn kernel. But it follows from Charikar (2002) that the random binning feature is only able to approximate kernel functions $k(\cdot)$ such that $1 - k(\mathbf{x} - \mathbf{y})$ satisfies the triangle inequality. This requirement is very restrictive and leaves the random binning features inapplicable to the most popular classes of smooth kernels including the Squared exponential kernel and Matérn family. In fact, any smooth kernel which is monotonically decreasing and is at least twice differentiable cannot be approximated using random binning features.

The random binning features estimator is an estimator whose output is either zero or one. We generalize this in Section 3.1 by allowing the estimator to assume a range of values and show that this estimator, unlike the random binning features estimator, is able to approximate a rich family of smooth kernels.

3.1 WLSH kernel family

We now define the *Weighted LSH (WLSH) Estimator*.

Definition 6 (WLSH Estimator)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be some even function with support $[-1/2, 1/2]$ and $\|f\|_2 = 1$ and let $p(\cdot)$ be some PDF with non-negative support. Also let $\mathcal{H}$ be the LSH family as in Definition 5. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the *Weighted LSH (WLSH) estimator* $\widetilde{k}_{f,p}$ is defined as:

$$\widetilde{k}_{f,p}(\mathbf{x}, \mathbf{y}) = \begin{cases} A & \text{if } h_{\mathbf{w},\mathbf{z}}(\mathbf{x}) = h_{\mathbf{w},\mathbf{z}}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $A = f^{\otimes d}(h_{\mathbf{w},\mathbf{z}}(\mathbf{x}) + \frac{\mathbf{z}-\mathbf{x}}{\mathbf{w}}) \cdot f^{\otimes d}(h_{\mathbf{w},\mathbf{z}}(\mathbf{y}) + \frac{\mathbf{z}-\mathbf{y}}{\mathbf{w}})$, and $h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}$.

For ease of notation, we often drop the subscripts and just write $\widetilde{k}(\cdot)$ to denote the WLSH. We show that the expectation of the WLSH estimator is a valid shift-invariant kernel. The expectation of the estimator is given by the following claim,

Claim 7. *For any PDF $p(\cdot)$ with non-negative support, any even function $f : \mathbb{R} \to \mathbb{R}$ with support $[-1/2, 1/2]$ and $\|f\|_2 = 1$, and any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the expectation of the WLSH kernel $\widetilde{k}(\mathbf{x}, \mathbf{y})$ over the random choice of LSH function $h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}$ is given by*

$$\mathbb{E}_{h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}}\left[\widetilde{k}(\mathbf{x}, \mathbf{y})\right] = \int_{\mathbb{R}^d} e^{2\pi i (\mathbf{x}-\mathbf{y})^\top \boldsymbol{\xi}} \prod_{l=1}^{d} \mathbb{E}_{w_l \sim p(w)}\left[w_l \cdot \left|\widehat{f}(w_l \xi_l)\right|^2\right] d\boldsymbol{\xi}.$$

Equivalently, it can be expressed as

$$\mathbb{E}_{h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}}\left[\tilde{k}(\mathbf{x}, \mathbf{y})\right] = \prod_{l=1}^{d} \mathbb{E}_{w_l \sim p(w)}\left[(f * f)\left(\frac{x_l - y_l}{w_l}\right)\right].$$

By Claim 7, $\mathbb{E}\left[\tilde{k}(\mathbf{x}, \mathbf{y})\right]$ is clearly shift-invariant. Moreover, by the convolution theorem (see Claim 13), the Fourier transform of the expectation is

$$\mathcal{F}\left[\mathbb{E}\left[\tilde{k}(\cdot + \mathbf{y}, \mathbf{y})\right]\right](\boldsymbol{\xi}) = \prod_{l=1}^{d} \mathbb{E}_{w_l \sim p(w)}\left[w_l \cdot \left|\widehat{f}(w_l \xi_l)\right|^2\right],$$

which is a positive function for every $\boldsymbol{\xi}$. Hence, the expectation of the WLSH kernel is a valid kernel. We now formally define *WLSH kernels families*.

Definition 8 (WLSH Kernel Family)**.** Let $p(\cdot)$ be some probability density function with support $\mathbb{R}_+$

and let $f : \mathbb{R} \to \mathbb{R}$ be some even function with support $[-1/2, 1/2]$ and $\|f\|_2 = 1$. The WLSH kernel function $k_{f,p} : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$k_{f,p}(\mathbf{x}) = \prod_{l=1}^{d} \left(\int_0^\infty p(w_l) \cdot (f * f) \left(\frac{x_l}{w_l} \right) dw_l \right),$$

for any $\mathbf{x} \in \mathbb{R}^d$. We often drop the subscripts f, p and just write $k(\cdot)$ to denote the WLSH kernel.

It follows from Claim 7 that for any WLSH kernel $k(\cdot)$, there exists an unbiased WLSH estimator

$$\mathbb{E}_{h_{\mathbf{w},\mathbf{z}} \sim \mathcal{H}} \left[\widetilde{k}(\mathbf{x}, \mathbf{y}) \right] = k(\mathbf{x} - \mathbf{y}).$$

3.2 Smoothness of WLSH Gaussian process

In the context of Bayesian estimation, some regularity assumptions are often made about the function being learned. Smoothness is the most common assumption. Suppose that $\eta : \mathbb{R}^d \to \mathbb{R}$ is a sample path from a *Gaussian process (GP)* $\mathrm{GP}(0, k(\mathbf{x}-\mathbf{y}))$, i.e., its mean is $\mathbb{E}[\eta(\mathbf{x})] = 0$ for every $\mathbf{x} \in \mathbb{R}^d$ and its covariance is given by the kernel function $\mathbb{E}[\eta(\mathbf{x})\eta(\mathbf{y})] = k(\mathbf{x}-\mathbf{y})$ for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where $k(\cdot)$ is a shift-invariant positive definite kernel. The Bayesian estimation algorithms commonly assume that the sample paths of the GP, $\eta(\mathbf{x})$, satisfy certain smoothness properties with high probability. For instance, in the context of Gaussian process optimization in bandit setting, to get a provable guarantee, the known algorithms require the derivatives of the GP's sample path, $\frac{\partial \eta(\mathbf{x})}{\partial \mathbf{x}}$, to be bounded everywhere with sub-Gaussian tail probability Srinivas et al. (2009). We prove that our WLSH construction (Definition 8) provides a class of smooth kernels.

In the following lemma we prove that the sample paths of $\mathrm{GP}(0, k_{f,p}(\mathbf{x} - \mathbf{y}))$ when the covariance $k_{f,p}(\cdot)$ is WLSH kernel (Definition 8) inherit their smoothness from the bucket-shaping function f. The lemma shows that our construction of *WLSH* family of kernels from Definition 8 is able to generate a GP such that the partial derivatives of a sample path from this GP is bounded everywhere with a sub-Gaussian distribution as long as the function $f(\cdot)$ is smooth. As shown in Figure 1, we use a bucket shape $f(\cdot)$ which has a smooth transition around the edges as opposed to random binning features whose bucket shape is $\mathrm{rect}(\cdot)$ with a discontinuity at the edges. Here we denote the partial derivative with respect to j^{th} coordinate ∂/∂_j by D_j. The partial derivative of the GP with respect to the j^{th} coordinate is denoted by $D_j\eta(\mathbf{x})$. The sample paths of this process are $D_j\eta(\mathbf{x})$, where $\eta(\mathbf{x})$ is a sample path from the original GP.

Lemma 9. *For any positive integer q, any integers $q_1, q_2, \cdots q_d \geq 0$ such that $\sum_j q_j = q$ let the derivative operator $\mathbf{D}$ be defined as $\mathbf{D} = D_1^{q_1} D_2^{q_2} \cdots D_d^{q_d}$. For any even function f with support $[-1/2, 1/2]$ which has bounded derivatives of up to $q+1$ order and any PDF $p(\cdot)$ with non-negative support, if $\eta : [0,1]^d \to \mathbb{R}$ is a sample path from $\mathrm{GP}(0, k(x-y))$, where $k(\cdot)$ is the WLSH kernel (Definition 8), then the mixed partial derivative of the sample path, $\mathbf{D}\eta(x)$, satisfies the following high probability bound:*

$$\Pr\left[\sup_{\mathbf{x} \in [0,1]^d} |\mathbf{D}\eta(\mathbf{x})| > M \right] \leq \left(\frac{LM}{\sigma^2} \right)^d e^{-\frac{M^2}{\sigma^2}},$$

where $\sigma^2 = \prod_{l=1}^d \left\| f^{(q_l)} \right\|_2^2 \int_{\mathbb{R}_+} \frac{p(w_l)}{w_l^{2q_l}} dw_l$ and $L = O\left(\sup_{j \in [d]} \left| \prod_{l \in [d]} \left\| f^{(q_l + \delta_{l,j})} \right\|_2^2 \int_{\mathbb{R}_+} \frac{p(w_l)}{w_l^{2(q_l+\delta_{l,j})}} dw_l \right| \right)$ where $\delta_{l,j} = 0$ for every $l \neq j$ and $\delta_{j,j} = 1$.

4 Spectral approximation and Kernel Ridge Regression (KRR)

In this section we prove our main results which show that our weighted LSH estimator provides an OSE for kernel matrices. Suppose that you are given a collection of points in the d dimensional Euclidean space $\mathbf{x}^1, \mathbf{x}^2, \ldots \mathbf{x}^n \in \mathbb{R}^d$ together with (noisy) measurements of some unknown function $\eta^* : \mathbb{R}^d \to \mathbb{R}$,

$$\gamma_i = \eta^*(\mathbf{x}^i) + \epsilon_i,$$

where the ϵ_i are iid Gaussians with variance σ_ϵ^2 and the aim is to estimate the underlying function $\eta^*(\mathbf{x})$ from the data. One simple yet powerful method for solving this problem is the *Kernel Ridge Regression* (KRR). To find the KRR estimator, one needs to solve the least squares problem $\min_\beta \|K\beta - \boldsymbol{\gamma}\|_2^2 + \lambda \beta^\top K \beta$, where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix defined as $K_{ij} = k(\mathbf{x}^i, \mathbf{x}^j)$. The least squares solution is $\beta^* = (K + \lambda I)^{-1} \boldsymbol{\gamma}$. If the function η^* is a sample path from a $\mathrm{GP}(0, k(\mathbf{x}, \mathbf{y}))$ then the KRR estimator (i.e., $\eta(\cdot) = \sum_{i \in [n]} \beta_i^* k(\cdot, \mathbf{x}^i)$) is optimal in the Bayesian sense.

In order to accelerate the computational complexity KRR, we approximate the kernel function $k(\cdot)$ using the WLSH estimator (Definition 6). For any $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$, the approximated kernel matrix $\widetilde{K} \in \mathbb{R}^{n \times n}$ is defined as, $[\widetilde{K}]_{ij} = \widetilde{k}_{f,p}(\mathbf{x}^i, \mathbf{x}^j)$, where $\widetilde{k}_{f,p}(\cdot)$ is the WLSH estimator as in Definition 6. One can see that the matrix $\widetilde{K}_{f,p}$ is very structured and typically sparse (it's ij^{th} entry is nonzero only if $\mathbf{x}^i$ and $\mathbf{x}^j$ get hashed into the same bucket, i.e., $h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^i) = h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^j)$). Hence, $\widetilde{K}_{f,p}$ supports fast matrix vector multiplication and can be stored in small memory.

Approximate kernel matrix $\widetilde{K}$ can be stored in small memory and supports fast matrix vector multiplication: Suppose that we want to build a data structure which can be stored in space $O(n)$ such that using this data structure we can compute the product $\widetilde{K}\beta$ for arbitrary vectors $\beta \in \mathbb{R}^n$ in linear time $O(n)$. It follows from Definition 6 that for any $s \in [n]$,

$$(\widetilde{K}\beta)_s = B_{h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^s)}(\beta) \cdot f^{\otimes d}\left(h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^s) + \frac{\mathbf{z}-\mathbf{x}^s}{\mathbf{w}}\right),$$

where $B_{\mathbf{j}}(\beta) = \sum_{i:h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^i)=\mathbf{j}} \beta_i \cdot f^{\otimes d}\left(\mathbf{j} + \frac{\mathbf{z}-\mathbf{x}^i}{\mathbf{w}}\right)$ for every bucket $\mathbf{j}$ and we call it the *load* of bucket $\mathbf{j}$. This is illustrated in Figure 1 for the one dimensional case. In dimension one, to compute the load of j^{th} bucket, we first shift the function f to $z + jw$ and then for every x^i which is hashed into j^{th} bucket, we scale β_i by the function value at point x^i, $f(\frac{x^i - jw - z}{w})$, and sum them all up.

Therefore we construct the data structure as follows: We first hash all the data points $\mathbf{x}^i$ using the LSH function $h_{\mathbf{w},\mathbf{z}}(\cdot)$ and keeps the lists $L_{\mathbf{j_1}}, L_{\mathbf{j_1}}, \ldots$, where each list corresponds to one of the non-empty buckets of this hashing. Each list $L_{\mathbf{j_r}}$ contains the points $\mathbf{x}^i$ which are hashed to bucket $\mathbf{j_r}$, i.e., $L_{\mathbf{j_r}} = \{i : h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^i) = \mathbf{j_r}\}$ for every r. All the lists can be formed in time $O(dn)$ which is the time to hash all data points. And the total size of all lists is the number of data points n, because each data point gets hashed into exactly one bucket, hence the data structure can be stored using $O(n)$ memory words. Then to compute the product $\widetilde{K}\beta$ first we compute the bucket load $B_{\mathbf{j_r}}(\beta)$ for every non-empty bucket $\mathbf{j_r}$,

$$B_{\mathbf{j_r}}(\beta) = \sum_{i \in L_{\mathbf{j_r}}} \beta_i \cdot f_d\left(\mathbf{j_r} + \frac{\mathbf{z}-\mathbf{x}^i}{\mathbf{w}}\right).$$

We can do this for all buckets using time $O(n)$. Then every coordinate s of the product $(\widetilde{K}\beta)_s$ is computed as follows:

$$(\widetilde{K}\beta)_s = B_{h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^s)}(\beta) \cdot f^{\otimes d}\left(h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^s) + \frac{\mathbf{z}-\mathbf{x}^s}{\mathbf{w}}\right),$$

where $B_{h_{\mathbf{w},\mathbf{z}}(\mathbf{x}^s)}(\beta)$ denotes the load of the bucket $\mathbf{x}^s$ is hashed into. Hence, the product can be computed in total time $O(n)$.

4.1 Oblivious subspace embedding via WLSH estimator

Recall that our aim is to solve the least squares problem $\min_\beta \|K\beta - \boldsymbol{\gamma}\|_2^2 + \lambda\beta^\top K\beta$ quickly by using an approximate kernel matrix $\widetilde{K}$. In order to get a provable $(1 \pm \epsilon)$-approximate solution to the least squares problem, $\widetilde{K}$ must be spectrally close to original K in some way. In this paper we focus on *oblivious subspace embeddings* (see Definition 1) and show that this property is enough to get a provably good approximation to the least squares problem. We need the following claim before proving the main result,

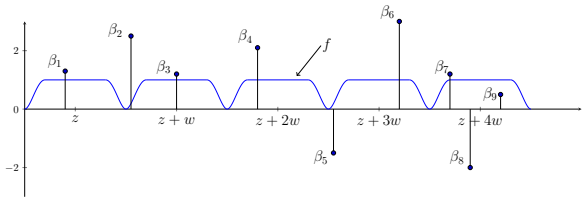

Figure 1: The *load* of $\mathbf{j}^{\text{th}}$ bucket corresponds to shifting the bucket-shaping function $f^{\otimes d}$ to $\mathbf{jw} + \mathbf{z}$ and then integrating it against $\alpha(\mathbf{x}) = \sum_{j=1}^n \beta_j \delta(\mathbf{x} - \mathbf{x}^j)$.

Claim 10. *For any dataset $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$, if $k(\cdot)$ is the WLSH estimator as in Definition 6 then its corresponding kernel matrix $\widetilde{K} \in \mathbb{R}^{n\times n}$, is symmetric and satisfies, $0 \preceq \widetilde{K} \preceq n\|f^{\otimes d}\|_\infty^2 \cdot I$.*

Now we are ready to prove the main theorem and show that WLSH estimator provides an oblivious subspace embedding for WLSH kernel matrix K.

Theorem 11. *For any positive integers d, n, any collection of points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$, any PDF $p(\cdot)$ with non-negative support, any even function $f(\cdot)$ with support $[-1/2, 1/2]$ and $\|f\|_2 = 1$, let $k(\cdot)$ be the WLSH kernel as in Definition 8 and let $K \in \mathbb{R}^{n\times n}$ be its kernel matrix. If $\widetilde{k}^1(\cdot), \widetilde{k}^2, \ldots, \widetilde{k}^m(\cdot)$ are independent instances of WLSH estimator as per Definition 6 and $\widetilde{K}^1, \widetilde{K}^2, \ldots, \widetilde{K}^m$ are their kernel matrices, then for any $\lambda, \epsilon > 0$, the matrix $\widetilde{K} := \frac{1}{m}\sum_{s=1}^m \widetilde{K}^s$ is an $\left(\epsilon, \frac{1}{\text{poly}(n)}, \lambda\right)$-oblivious subspace embedding (see Definition 1) for the kernel matrix K as long as $m = \Omega\left(\frac{\|f^{\otimes d}\|_\infty^2}{\epsilon^2} \cdot (n/\lambda) \cdot \log n\right)$.*

Proof. Let $U \in \mathbb{R}^{n\times n}$ be the unitary matrix of eigenvectors of K, i.e., i^{th} column of matrix U corresponds to i^{th} eigenvector of matrix K (The eigenvalues are ordered in the decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$). Since U is unitary ($U^\top U = I_{n\times n}$), it is enough to prove that with probability $1 - \frac{1}{\text{poly}(n)}$, $(1-\epsilon)U^\top(K + \lambda I)U \preceq U^\top(\widetilde{K} + \lambda I)U \preceq (1+\epsilon)U^\top(K + \lambda I)U$. Let $Z = (U^\top(K + \lambda I)U)^{-1/2}$. Since Z is a diagonal matrix with entries $Z_{i,i} = \frac{1}{\sqrt{\lambda_i + \lambda}}$ and is, therefore, positive definite, we can multiply the above identity from left and right by Z and equivalently prove that, $(1-\epsilon)I \preceq Z^\top U^\top(\widetilde{K} + \lambda I)UZ \preceq (1+\epsilon)I$. In order to satisfy the above it is sufficient to have $||Z^\top U^\top(\widetilde{K} + \lambda I)UZ - I||_{op} \leq \epsilon$ where $\|\cdot\|_{op}$ denotes the operator norm of matrices. Therefore, it suf-

fices to prove $\Pr\left[\left\|Z^\top U^\top(\widetilde{K}+\lambda I)UZ - I\right\|_{op} \le \epsilon\right] \ge 1 - \frac{1}{\mathrm{poly}(n)}$, which follows from Lemma 28 (see Appendix D).

By Claim 10 the estimators $\widetilde{K}^s$ are PSD, therefore, $0 \preceq Z^\top U^\top \left(\widetilde{K}^s + \lambda I\right) UZ$, for every $s \in [m]$. Also because of the unbiasedness of estimators, $\mathbb{E}\left[Z^\top U^\top \left(\widetilde{K}^s + \lambda I\right) UZ\right] = I$. Therefore we can invoke Lemma 28. In order to do so, we need to upper bound the operator norm of $Z^\top U^\top \left(\widetilde{K}^s_{f,p} + \lambda I\right) UZ$. By Claim 10, we have $\left\|\widetilde{K}^s + \lambda I\right\|_{op} = \left\|\widetilde{K}^s\right\|_{op} + \lambda \le n \cdot \|f^{\otimes d}\|_\infty^2 + \lambda$; thus,

$$\begin{aligned}
&\left\|Z^\top U^\top \left(\widetilde{K}^s + \lambda I\right) UZ\right\|_{op} \\
&\le \left\|\widetilde{K}^s + \lambda I\right\|_{op} \cdot \|Z^\top U^\top UZ\|_{op} \\
&\le \left(n\|f^{\otimes d}\|_\infty^2 + \lambda\right) \cdot ||Z^\top Z||_{op} \le \frac{n}{\lambda} \cdot \|f^{\otimes d}\|_\infty^2 + 1.
\end{aligned}$$

The result now follows by Lemma 28 (see Appendix D). □

Now we show that our analysis in Theorem 11 is not loose and in order to get an OSE for worst case datasets one needs $m = \Omega\left(\frac{1}{\epsilon^2}(n/\lambda)\log n\right)$.

Theorem 12 (Lower Bound in order to achieve OSE)**.** *Let $f(\cdot) = \mathrm{rect}(\cdot)$ and $p(w) = we^{-w}$ (Gamma distribution) and let $k(\cdot)$ be the WLSH kernel as in Definition 8. For any integer $d \ge 1$ any $\lambda > 0$ and any integer $n \ge 8\lambda$, there exists a dataset $\mathbf{x}^1, \cdots \mathbf{x}^n \in \mathbb{R}^d$ such that if $K \in \mathbb{R}^{n\times n}$ is the kernel matrix defined as $K_{ij} = k_{f,p}(\mathbf{x}^i - \mathbf{x}^j)$ and $\widetilde{k}^1(\cdot), \widetilde{k}^2, \cdots \widetilde{k}^m(\cdot)$ are independent instances of WLSH estimator as per Definition 6 and $\widetilde{K}^1, \widetilde{K}^2, \cdots \widetilde{K}^m$ are their kernel matrices then for any $0 < \epsilon \le 1/6$ in order for $\widetilde{K} := \frac{1}{m}\sum_{s=1}^m \widetilde{K}^s$ to be an $(\epsilon, \frac{1}{n}, \lambda)$-oblivious subspace embedding for K one needs to have $m = \Omega\left(\frac{1}{\epsilon^2} \cdot \frac{n}{\lambda} \cdot \log n\right)$.*

Proof sketch: let the points $\{\mathbf{x}^i\}_{i=1}^n \subseteq \mathbb{R}^d$ be positioned as $\mathbf{x}^1 = \cdots = \mathbf{x}^{n/2} = (-\lambda/n, 0, 0, \ldots 0)^\top$ and $\mathbf{x}^{n/2+1} = \cdots = \mathbf{x}^n = (\lambda/n, 0, 0, \cdots 0)^\top$. Let the vector $\beta \in \mathbb{C}^n$ be defined as, $\beta_1 = \beta_2 = \cdots = \beta_{n/2} = -1$ and $\beta_{n/2+1} = \cdots = \beta_n = 1$. The proof proceeds by showing that in order to preserve the quadratic form corresponding to this β, one needs to set $m = \Omega\left(\frac{1}{\epsilon^2} \cdot \frac{n}{\lambda} \cdot \log n\right)$. By some calculations, we see that $\beta^\top \widetilde{K}^s \beta$ has the following distribution:

$$\beta^\top \widetilde{K}^s \beta = \begin{cases} \frac{n^2}{2} & \text{with probability } p \le \frac{2\lambda}{n} \\ 0 & \text{with probability } 1-p \end{cases}.$$

Thus, to obtain a non-zero estimator with constant probability, one needs $m = \Omega(\frac{n}{\lambda})$. In order to obtain the $(1 \pm \epsilon)$-approximation guarantee with probability $1 - 1/n$, see Appendix D.

4.2 Approximate KRR via WLSH

In this section we give the algorithm for approximate KRR problem using the WLSH estimator. Let $\widetilde{k}^s_{f,p}(\cdot)$ be independent instances of the WLSH estimator for all $s \in [m]$. We define the approximate kernel function $\widetilde{k}(\cdot) := \frac{1}{m}\sum_{s=1}^m \widetilde{k}^s_{f,p}(\cdot)$ and let $\widetilde{K}$ be the corresponding kernel matrix. Suppose $\eta^* : \mathbb{R} \to \mathbb{R}$ is the underlying function to be learned via KRR and the measurements are $\gamma_i = \eta^*(\mathbf{x}^i) + \epsilon_i$, where ϵ_i's are iid normal noise with variance σ_ϵ^2. We solve the approximate regressor by solving the linear system, $(\widetilde{K} + \lambda I)\beta = \boldsymbol{\gamma}$, where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)^\top$. Then the approximate regressor estimates the function values at a point $\mathbf{x}$ as follows:

$$\begin{aligned}
\widetilde{\eta}(\mathbf{x}) &= \sum_{i\in[n]} \beta_i \widetilde{k}(\mathbf{x}, \mathbf{x}^i) \\
&= \frac{1}{m}\sum_{s=1}^m B_{h^s_{\mathbf{w},\mathbf{z}}(\mathbf{x})}(\beta) \cdot f^{\otimes d}\left(h^s_{\mathbf{w},\mathbf{z}}(\mathbf{x}) + \frac{\mathbf{z}-\mathbf{x}}{\mathbf{w}}\right)
\end{aligned}$$

where $B_{h^s_{\mathbf{w},\mathbf{z}}(\mathbf{x})}(\beta) = \sum_{i:h^s_{\mathbf{w},\mathbf{z}}(x^i) = h^s_{\mathbf{w},\mathbf{z}}(\mathbf{x})} \beta_i \cdot f^{\otimes d}\left(h^s_{\mathbf{w},\mathbf{z}}(\mathbf{x}^i) + \frac{\mathbf{z}-\mathbf{x}^i}{\mathbf{w}}\right)$ is the load of the bucket that $\mathbf{x}$ gets hashed into via s^{th} LSH function, $h^s_{\mathbf{w},\mathbf{z}}$.

We give an *empirical risk bound* for the WLSH estimator in Appendix E.

5 Experiments

Estimating a GP using the WLSH kernel: In the first set of experiments we show that our WLSH kernel family from Section 3 performs as accurately as the most popular kernel functions for learning Gaussian processes through KRR. Specifically, we generate a random function $\eta : [0,1]^d \to \mathbb{R}$ which is a sample path from a Gaussian process with zero mean whose covariance $\sigma(x,y) = \mathbb{E}[\eta(x)\eta(y)]$ is one of **(1)** Laplace $e^{-\|x-y\|_1}$ or **(2)** Squared Exponential $e^{-\|x-y\|_2^2}$ or **(3)** Matérn with $\nu = 5/2$: $C_{5/2}(x-y) = \left(1 + \|x-y\|_2 + \|x-y\|_2^2/3\right)e^{-\|x-y\|_2}$.

Table 1: Test set RMSE for estimating GPs.

Covariance of GP $\sigma(\cdot)$	Dim.	Laplace	Squared exponential	Matérn $\nu = 5/2$	WLSH $k_{f,p}(\cdot)$
$e^{-\|\cdot\|_2^2}$	30	0.128	0.086	0.093	0.088
	5	0.043	0.031	0.032	0.029
$e^{-\|\cdot\|_1}$	30	0.385	0.479	0.481	0.438
	5	0.103	0.230	0.226	0.166
$C_{5/2}(\cdot)$	30	0.335	0.291	0.299	0.294
	5	0.013	0.016	0.013	0.012

We run this experiment for two settings: Low-dimensional data ($d = 5$) and high-dimensional data

($d = 30$). In each case, we sample $\eta(\mathbf{x})$ uniformly over $[0,1]^d$ at 4000 points. We use 3000 samples for training the estimator and 1000 samples for testing. Then we estimate the function value on test data using KRR on the training data. We run KRR with various kernel function choices and show that our WLSH kernel (Definition 8) performs as well as the most popular kernel functions such as Matérn $\nu = 5/2$, Squared Exponential, and Laplace. The WLSH kernel we used for this experiment has the bucket-shaping function $f(x) = \left(\text{rect} * \text{rect}_{1/4} * \text{rect}_{1/4}\right)(2x)$. This function has a continuous derivative and a bounded second derivative. Moreover, we chose the PDF to be $p(w) = \frac{w^6}{5!}e^{-w}$. Thus, the resulting kernel has bounded mixed partial derivatives of up to the fourth order. This is the same type of smoothness as the Matérn kernel with $\nu = 5/2$, but in our experiments (see Table 1), we outperform Matérn kernel on all datasets. Moreover, in the low-dimensional setting $d = 5$, we outperform the Squared Exponential kernel.

Large scale KRR on real data: Our second set of experiments shows that the WLSH estimator speeds up KRR on standard real data sets by orders of magnitude compared to exact KRR and has better accuracy than the popular Random Fourier Features (RFF) Rahimi and Recht (2007). We evaluate the following methods:

Exact KRR using exact kernel computation for various shift-invariant kernel functions.

Random Fourier Features (RFF) for approximating the squared exponential kernel. The kernel value is approximated by $\widetilde{k}(x^i, x^j) = \phi(x^i)^\top \phi(x^j)$, where $\phi : \mathbb{R}^d \to \mathbb{R}^D$ is a random mapping and D denotes the number of random features.

WLSH using the procedure explained in Section 4.2 with bucket-shaping function $f(\cdot) = \text{rect}(\cdot)$ and PDF $p(w) = we^{-w}$.

Results: The Root Mean Square Error (RMSE) of different methods on the test data set as well as the time to train the regressors are presented in Table 2.[2,3] One can see the LSH method is as accurate as the exact KRR on the first two datasets while its running time is at least 3x faster. On the last two datasets, the exact method did not converge to a solution within 12 hours but the approximate methods could run pretty fast. The LSH method outperforms the accuracy of RFF on the large scale datasets. RFF requires a large number of features D in order to be accurate which leads to a huge memory usage therefore on the large scale datasets where we have a memory constraint and cannot use large D, RFF's performance deteriorate. The running time of RFF is better than LSH method because its implementation can be optimized but when data is large and there is a memory constraint, RFF performs worse than LSH.

Table 2: Test set RMSE of different regression methods together with the running times.

Dataset	**Exact Laplace**	**Exact Squared Exp.**	**Exact Matérn $\nu = \frac{5}{2}$**	**Random Fourier Features**	**WLSH**
Wine Quality $d = 11$ size: 6497	0.684 28 sec	0.728 30 sec	0.709 1 min	0.737 2 sec D=7000	0.701 5 sec m=450
Insurance Company $d = 85$ size: 9822	0.231 3 min	0.231 3 min	0.231 5.5 min	0.231 3 sec D=5000	0.232 2 sec m=250
CT Slices Location $d = 384$ size: 53500	N/A >12 hrs	N/A >12 hrs	N/A >12 hrs	4.10 0.5 min D=3500	3.45 1 min m=50
Forest Cover $d = 54$ size: 581012	N/A >12 hrs	N/A >12 hrs	N/A >12 hrs	0.968 6 min D=1500	0.720 7.5 min m=50

We use the following standard real datasets for Gaussian process regression: The first dataset we used for regression is the **Wine Quality** dataset. The dimensionality of this dataset is $d = 11$. We used 4000 samples for training the regressors and 2497 samples for testing the accuracy. The second dataset is **Insurance Company** dataset. The dimensionality of this dataset is $d = 85$. We used 5822 samples for training the regressors and 4000 samples for testing the performance of estimators. The third dataset is the **Location of CT Slices**. The dimensionality of this dataset is rather high $d = 384$. We used 35000 samples for training the regressors and 18500 samples for testing their performance. The last dataset is the **Forest Cover** dataset. The dimensionality of this dataset is $d = 54$. We used 500000 samples for training the regressors and 81012 samples for testing the regressors.

[2]All methods require solving a linear system which we do using the Conjugate Gradient method. The most expensive computation in each iteration is multiplying a vector by the (approximate) kernel matrix. This takes time $\approx n^2$ for exact methods and time $\approx nD$ for RFF, where D is the number features, and time $\approx nm$ for WLSH method, where m is the number of LSH functions.

[3]Since RFF and LSH method are randomized, we ran the experiments with 5 different random seeds and reported the avg. RMSE and running time in Table 2.

References

T. D. Ahle, M. Kapralov, J. B. Knudsen, R. Pagh, A. Velingker, D. Woodruff, and A. Zandieh. Oblivious sketching of high-degree polynomial kernels. *ACM-SIAM Symposium on Discrete Algorithms*, 2020.

A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees.

In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

H. Avron, K. L. Clarkson, and D. P. Woodruff. Faster kernel ridge regression using sketching and preconditioning. *SIAM J. Matrix Analysis Applications*, 38(4):1116–1138, 2017a.

H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 253–262, 2017b.

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

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

M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002.

S. Ghosal, A. Roy, et al. Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5):2413–2429, 2006.

S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(1):321–350, 2012.

P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613, 1998.

C. Musco and C. Musco. Recursive sampling for the nystrom method. In *Advances in Neural Information Processing Systems*, pages 3833–3845, 2017.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1177–1184. Curran Associates, Inc., 2007.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Neural Information Processing Systems (NIPS)*, 2015.

E. V. Slud. Distribution inequalities for the binomial law. *The Annals of Probability*, pages 404–412, 1977.

N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z. URL `https://doi.org/10.1007/s10208-011-9099-z`.

A. W. Van Der Vaart and J. A. Wellner. Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer, 1996.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.*, 16:3299–3340, 2015.