Quantization Algorithms for Random Fourier Features

Xiaoyun Li, Ping Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St Bellevue WA 98004 USA
{lixiaoyun996, pingli98}@gmail.com

Abstract

The method of random projection (RP) is the standard technique for dimensionality reduction, approximate near neighbor search, compressed sensing, etc., which provides a simple and effective scheme for approximating pairwise inner products and Euclidean distances in massive data. Closely related to RP, the method of random Fourier features (RFF) has also become popular for approximating the (nonlinear) Gaussian kernel. RFF applies a specific nonlinear transformation on the projected data from RP. In practice, using the Gaussian kernel often leads to better performance than the linear kernel (inner product).

After random projections, quantization is an important step for efficient data storage, computation and transmission. Quantization for RP has been extensively studied in the literature. In this paper, we focus on developing quantization algorithms for RFF. The task is in a sense challenging due to the tuning parameter γ in the Gaussian kernel. For example, the quantizer and the quantized data might be tied to each specific Gaussian kernel parameter γ. Our contribution begins with the analysis on the probability distributions of RFF, and an interesting discovery that the marginal distribution of RFF is free of the parameter γ. This significantly simplifies the design of the Lloyd-Max (LM) quantization scheme for RFF in that there would be only one LM quantizer (regardless of γ). Detailed theoretical analysis is provided on the kernel estimators and approximation error, and experiments confirm the effectiveness and efficiency of the proposed method.

1. Introduction

The method of random projections (RP) is a popular strategy to deal with big data, for instance, for efficient processing, computations, storage and transmissions of massive (and high-dimensional) datasets. The merit of RP is highlighted by the celebrated Johnson-Lindenstrauss Lemma (Johnson and Lindenstrauss, 1984), which states that with high probability the Euclidean distance between data points is approximately preserved in the projected space if the number of projections is sufficiently large. In the past two decades, RP has been used extensively in dimension reduction, approximate near neighbor search, compressed sensing, computational biology, etc. See examples of relatively early works on RP (Dasgupta, 2000; Bingham and Mannila, 2001; Buhler, 2001; Achlioptas, 2003; Fern and Brodley, 2003; Datar et al., 2004; Candès et al., 2006; Donoho, 2006; Li et al., 2006; Freund et al., 2007; Li, 2007). In this paper, we study quantization schemes for random Fourier features (RFF), which are nonlinear transformations of random projections, to accurately approximate the (nonlinear) Gaussian kernel.

1.1. Linear Kernel and Gaussian Kernel

Let $u, v \in \mathcal{X} \subseteq \mathbb{R}^d$ denote two d-dimensional data vectors. The linear kernel is the inner product $\langle u, v\rangle = u^T v$. For training large-scale linear learning algorithms such as linear support vector machine (SVM) and linear logistic regression, highly efficient training algorithms have been widely used (Joachims, 2006; Shalev-Shwartz et al., 2011; Fan et al., 2008). Despite their high efficiency, the drawback of linear learning methods is that they often do not provide a good accuracy as they neglect the nonlinearity of data. This motivates researchers to find efficient training algorithms for nonlinear kernels such as the Gaussian kernel (Hastie et al., 2001; Schölkopf and Smola, 2002; Bottou et al., 2007), which is defined through a real-valued kernel function

$$K_\gamma(u, v) = \langle \xi(u), \xi(v)\rangle = e^{-\frac{\gamma^2\|u-v\|^2}{2}},$$

where $\xi(\cdot) : \mathcal{X} \mapsto \mathbb{H}$ is the implicit feature map and γ is a hyper-parameter, with $\mathbb{H}$ the corresponding Reproducing

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Kernel Hilbert Space (RKHS). It is known that the Gaussian kernel is shift-invariant and positive definite. Throughout the paper, we assume that $\mathcal{X}$ lies on the unit sphere, i.e., all the data points are normalized to have unit l_2 norm. This will save us from book-keeping the sample norms. Note that, normalizing each data vector to unit l_2 norm before training is a fairly standard procedure in practice. In this case, denoting the correlation coefficient $\rho = \cos(u, v) = u^T v$, the Gaussian kernel can be formulated as

$$K_\gamma(u, v) = e^{-\frac{\gamma^2(2-2\rho)}{2}} = e^{-\gamma^2(1-\rho)}. \tag{1}$$

In the rest of the paper, we will omit the subscript "γ" in K_γ.

Storing/materializing a kernel matrix for a dataset of n samples would need n^2 entries, which may not be realistic even just for medium datasets (e.g., $n = 10^6$). To avoid this problem, the entries of the kernel matrix can be computed on the fly from the original dataset. This however will increase the computation time, plus storing the original high-dimensional dataset for on-demand distance computations can also be costly. Also, the training procedure for nonlinear kernel algorithms is known to be expensive (Platt, 1998; Bottou et al., 2007). Therefore, it has been an active area of research to speed up kernel machines, and using various types of random projections has become popular.

1.2. Random Projections (RP) and Random Fourier Features (RFF)

Again, consider two data vectors $u, v \in \mathbb{R}^d$. Further, we assume they are normalized to have unit l_2 norm and we denote $\rho = \langle u, v \rangle$. We generate a random Gaussian vector $w \in \mathbb{R}^d$ with i.i.d. entries in $N(0, 1)$.

$$\mathbb{E}[\langle w^T u, w^T v \rangle] = \langle u, v \rangle = \rho.$$

This is the basic idea of using random projections to approximate inner product. See Li et al. (2006) for the theoretical analysis (e.g., variance calculations) of this approximation.

With an additional step, one can use random projections to approximate the Gaussian kernel. The random Fourier Feature (Rudin, 1990; Rahimi and Recht, 2007) is defined as

$$\textbf{RFF:} \quad F(u) = \sqrt{2}\cos(\gamma w^T u + \tau), \tag{2}$$

where $\tau \sim uniform(0,\ 2\pi)$, the uniform random variable. Some basic probability calculations reveal that

$$\mathbb{E}\left[F(u)F(v)\right] = K(u, v) = e^{-\gamma^2(1-\rho)}.$$

In other words, the inner product between the RFFs of two data samples provides an unbiased estimate of the Gaussian kernel. The projections need to be repeated for a sufficient number of times in order to obtain reliable estimates. That is, we generate m independent RFFs using i.i.d. $w_1, ..., w_m$ and $\tau_1, ..., \tau_m$, and approximate the kernel $K(u, v)$ by

$$\hat{K}(u, v) = \frac{1}{m}\sum_{i=1}^{m} F_i(u)F_i(v), \tag{3}$$

where F_i denotes the RFF generated by w_i, τ_i. Furthermore, Li (2017b) showed that one can actually reduce the estimation variances by normalizing the RFFs.

In large-scale learning, using the above estimator simply requires taking the inner product between the RFF vectors of u and v. Thus, feeding RFFs into a linear model approximates training a non-linear kernel machine, known as *kernel linearization*, which may significantly accelerate training and alleviate memory burden for storing the kernel matrix, leading to numerous applications (Raginsky and Lazebnik, 2009; Yang et al., 2012; Affandi et al., 2013; Hernández-Lobato et al., 2014; Dai et al., 2014; Yen et al., 2014; Hsieh et al., 2014; Shah and Ghahramani, 2015; Chwialkowski et al., 2015; Richard et al., 2015; Sutherland and Schneider, 2015; Li, 2017b; Avron et al., 2017; Sun et al., 2018; Tompkins and Ramos, 2018; Li et al., 2020; Li and Li, 2021).

1.3. Quantized Random Projections (QRP)

One can further compress the projected data by quantization, into discrete integer (or even binary) values. The so-called quantized random projection (QRP) has found useful in many problems, e.g., theory, similarity search, quantized compressed sensing, classification and regression (Goemans and Williamson, 1995; Charikar, 2002; Datar et al., 2004; Zymnis et al., 2010; Jacques et al., 2013; Leng et al., 2014; Li et al., 2014; Li and Slawski, 2017; Slawski and Li, 2018; Li and Li, 2019b;a). The motivation is clear. If one can represent each RP (or RFF) using (e.g.,) 4 bits and still achieve similar accuracy as using 32 or 64 bits, it is then a substantial saving in storage. Typically, savings in storage can directly translate into savings in data transmissions and subsequent computations. In addition, quantization also provides the capability of indexing due to the integer nature of quantized data, which can be used to build hash tables for approximate near neighbor search (Indyk and Motwani, 1998).

The simplest quantization scheme is the 1-bit (sign) random projection, including sign Gaussian random projection (Goemans and Williamson, 1995; Charikar, 2002) and sign stable random projection (Li, 2017a) (for approximating the χ^2 kernel and others). Basically, one only keeps the signs of projected data. Even though the 1-bit schemes appear overly crude, in some cases 1-bit random projections can achieve better performance than full-precision RPs in similarity search and nearest neighbor classification tasks. Nevertheless, in general, one would need more than just 1-bit in order to achieve sufficient accuracy. For example, Li and Slawski (2017); Slawski and Li (2018); Li and Li (2019b) apply the (multi-bit) Lloyd-Max (LM) quantization (Max, 1960; Lloyd, 1982) on the projected data.

1.4. Summary of Contributions

Due to the tuning parameter γ in the Gaussian kernel, initially one might expect that a different LM quantizer would be needed for a different γ. In this paper, our contribution begins with an interesting finding that the marginal distribution of the RFF is actually free of the parameter γ. This result means that only one quantizer would be needed for all γ values. Based on the marginal distribution of the RFF, we incorporate the idea of *distortion optimal* LM quantization theory to nonlinear random feature compression by providing a thorough study on the theoretical properties and practical performance. Extensive simulations and learning experiments are provided to validate the theoretical results.

2. The Probability Distributions of RFF

This section provides an analysis on the probability distribution of RFF (2), which is key to the design of quantization schemes in Section 3. First, we introduce some notations.

Let $u, v \in \mathcal{X} \subseteq \mathbb{R}^d$ be two normalized data points, and $w \in \mathbb{R}^d$ be a random vector with i.i.d. $N(0,1)$ entries. The projected data $w^T u$ and $w^T v$ follow a bivariate normal $\begin{pmatrix} w^T u \\ w^T v \end{pmatrix} \sim N\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$, $\|u\| = \|v\| = 1, \rho = u^T v$. Two definitions, $\phi_\sigma(t)$ and $\Phi(t)$, are used in the paper:

$$\phi_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}},$$

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \int_{-\infty}^{t} \phi_1(z) dz,$$

where $\Phi(t)$ is the cumulative distribution function (cdf) of the standard normal $N(0,1)$ and $\phi_\sigma(t)$ is the probability density function (pdf) of $N(0,\sigma^2)$.

2.1. Marginal Distributions

The marginal distribution of the RFF serves as the foundation of our proposed quantization schemes.

Theorem 2.1. *Let $X \sim N(0,1)$, $\tau \sim uniform(0,2\pi)$ be independent. Denote $Z = \cos(\gamma X + \tau)$. We have the probability density function*

$$f_Z(z) = \frac{1}{\pi\sqrt{1-z^2}}, \quad z \in [-1,1], \tag{4}$$

for any $\gamma > 0$. In particular, $Z \overset{d}{\sim} \cos(\tau)$ in distribution.

Theorem 2.1 says that for any kernel parameter γ, the (unscaled) RFF follows the same distribution as the cosine of the uniform noise itself. Intuitively, this is because cosine is a 2π-periodic function and normal distribution is symmetric. As will be introduced in Section 3, Eq. (4) is the underlying signal distribution in our Lloyd-Max (LM) quantizer construction. This interesting result implies that we only need to design just one LM quantizer for all γ values.

Remark 2.1. *In Theorem 2.1 we consider $X \sim N(0,1)$ because we assume data samples are normalized for conciseness. It is easy to see that this result also holds without data normalization (i.e., X is Gaussian with arbitrary variance) since we can offset the variance of X by altering γ. Therefore, Theorem 2.1 is a universal result implying that the LM quantizer also works without data normalization.*

2.2. Joint Distribution

In the sequel, we analyze the joint distribution of RFFs of two data samples with correlation ρ, which will play an important role in later theoretical analysis.

Theorem 2.2. *Denote $z_x = \cos(\gamma X + \tau)$, $z_y = \cos(\gamma Y + \tau)$ where X, Y, τ are the same as Lemma D.2. Then we have the joint density function for $(z_x, z_y) \in [-1,1]^2$,*

$$f(z_x, z_y) = \frac{\sum_{k=-\infty}^{\infty} \left[\phi_\sigma(a_x^* - a_y^* + 2k\pi) + \phi_\sigma(a_x^* + a_y^* + 2k\pi)\right]}{\pi\sqrt{1-z_x^2}\sqrt{1-z_y^2}},$$

with $a_x^* = \cos^{-1}(z_x)$, $a_y^* = \cos^{-1}(z_y)$, $\sigma = \sqrt{2(1-\rho)}\gamma$.

In Figure 1, we plot the joint density at several γ values. We conclude several properties of the joint distribution. Firstly, it is obvious that z_x and z_y are exchangeable, i.e., $f(Z_x, Z_y) = f(Z_y, Z_x)$. Secondly, it is symmetric which means $f(Z_x, Z_y) = f(-Z_x, -Z_y)$. Moreover, the following result is important for our analysis on the monotonicity and variance of quantized kernel estimators in Section 4.

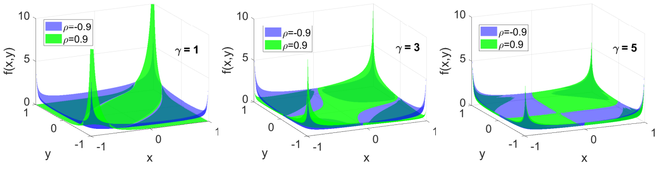

Figure 1. The joint density of RFF (Theorem 2.2) with example $\rho = -0.9, 0.9$, and $\gamma = 1, 3, 5$.

Proposition 2.3. *Let the density function f be defined as Theorem 2.2. If $\sqrt{2(1-\rho)}\gamma \le \pi$, then $f(z_x, z_y) > f(z_x, -z_y)$ for $\forall (z_x, z_y) \in (0,1]^2$ or $(z_x, z_y) \in [-1,0)^2$.*

In Proposition 2.3, the quantity $\sqrt{2(1-\rho)}\gamma$ will be reduced if we either increase ρ or decrease γ. Figure 1 illustrates that smaller $\sqrt{2(1-\rho)}\gamma$ reinforces the dependency between z_x and z_y. The density around $(1,1)$ and $(-1,-1)$ reaches the highest when $\rho = 0.9$ and $\gamma = 1$.

3. Quantization Schemes for RFF

Quantization is a basic topic in information theory and signal processing (Widrow and Kollár, 2008). On the other hand, many interesting research works appear in the literature even very recently, for achieving better efficiency in data storage, data transmission, computation, and energy consumption.

For example, there is a recent paper for using quantization to improve advertising click-through rate (CTR) models for a commercial search engine (Xu et al., 2021). Quantization for random projections has already been heavily studied.

In this paper, we focus on quantization with small (e.g., ≤ 4) bits. We consider general multi-bit quantizers, with 1-bit quantization as a special case, for the cosine feature in RFF bounded in $[-1, 1]$. Here, "b-bit" means the quantizer has 2^b levels. For simplicity, we denote $z = \cos(\gamma w^T u + \tau)$ as the item to be quantized. We study two algorithms: "round-random" (which we refer to as the "stochastic quantization (StocQ)" and "Lloyd-Max (LM) quantization".

3.1. Stochastic Quantization (StocQ)

The idea of stochastic rounding or stochastic quantization (StocQ) dated back at least to the 1950s (Forsythe, 1950; Barnes et al., 1951; Forsythe, 1959). A b-bit StocQ quantizer splits $[-1, 1]$ into $(2^b - 1)$ intervals, with consecutive borders $-1 = t_0 < ... < t_{2^b-1} = 1$. Let $[t_i, t_{i+1}]$ be the interval containing z. StocQ pushes z to either t_i or t_{i+1} depending on the distances. Concretely, with $\triangle_i = t_{i+1} - t_i$,

$$P(Q(z) = t_i) = \tfrac{t_{i+1} - z}{\triangle_i}, \quad P(Q(z) = t_{i+1}) = \tfrac{z - t_i}{\triangle_i}. \qquad (5)$$

It is not difficult to see that by this sampling procedure, conditional on the full-precision RFF z, the quantized value $Q(z)$ by StocQ is unbiased of z. On the other hand, also due to the Bernoulli sampling approach, StocQ has the extra variance especially when the number of bits is small.

Recently, Zhang et al. (2019) applied StocQ with uniform borders in machine learning tasks with RFF. In this paper, we consider a more general approach where the borders are not necessarily uniform, for a broader applicability.

3.2. Lloyd-Max (LM) Quantization

In quantization theory, the Lloyd-Max (LM) (Max, 1960; Lloyd, 1982) quantization scheme has a long history that also leads to some well-known methodsk, e.g., k-means. Interestingly, it has not been adopted to RFF in the prior literature. In contrast to StocQ, the proposed LM-RFF constructs a fixed quantizer Q_b. We call $[\mu_1, ..., \mu_{2^b}] \in \mathbb{C}$ the reconstruction levels, with $\mathbb{C}$ the "codebook" of Q_b. Also, $-1 = t_0 < ... < t_{2^b} = 1$ represent the borders of the quantizer. Then, LM-RFF quantizer Q_b defines a mapping: $[-1, 1] \mapsto \mathbb{C}$, where $Q_b(x) = \mu_i$ if $t_{i-1} < x \leq t_i$. By choosing the error function as the squared difference, given an underlying signal distribution $f(z)$ with support $\mathcal{S}$, the LM quantizer minimizes the *distortion* defined as

$$D_Q = \int_{\mathcal{S}} (z - Q(z))^2 f(z) dz,$$

aiming to keep most amount of information of the original signal. For the signal distribution, it is natural to set the target distribution as the distribution of RFF itself (4). Consequently, the LM quantizer is subject to the distortion:

$$\textbf{LM-RFF:} \quad D \triangleq \int_{[-1,1]} \left(z - Q(z)\right)^2 \tfrac{1}{\pi\sqrt{1-z^2}} dz. \qquad (6)$$

Conceptually, optimizing (6) minimizes the average difference between RFF z and $Q(z)$. To solve the optimization problem, we exploit classical Lloyd's algorithm, which alternatively updates two parameters until convergence. By (e.g.,) Wu (1992), the algorithm converges to the globally optimal solution since the squared loss is convex and symmetric. The algorithm terminates when the total absolute change in borders and reconstruction levels in two consecutive iterations is smaller than a given threshold (e.g., 10^{-5}). For practitioners to use our quantizers forthrightly, we provide the concrete quantizer construction and the derived LM-RFF quantizers in Appendix A.

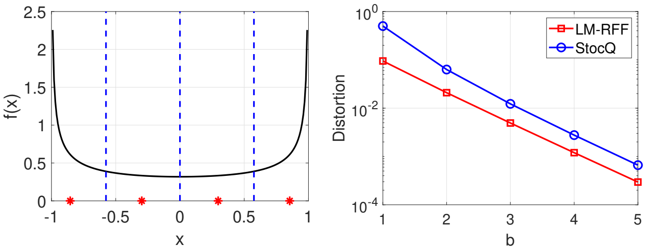

Figure 2. **Left**: LM-RFF quantizer, $b = 2$. Black curve is the RFF marginal density. **Right**: Distortion of LM-RFF and StocQ.

In Figure 2, we plot the 2-bit LM-RFF quantizer as an example (left), along with the distortions of LM-RFF and uniform StocQ with various number of bits (right). We see that the LM method generates non-uniform quantization borders and codes. From the distortion plot, we validate that LM-RFF indeed provides smaller distortion than StocQ.

As a remark, we note that LM quantization is more convenient and faster compared with StocQ. While LM is a fixed quantization approach, StocQ requires generating an extra random number for each RFF of each data point.

3.3. Quantized Kernel Estimators

In the following, we define the kernel estimators built upon quantized RFFs. For a fixed γ (Gaussian kernel parameter), with a general RFF quantizer $\tilde{Q}$, a simple quantized kernel estimator using m random features can be constructed as

$$\hat{K}_{\tilde{Q}}(u, v) = \frac{2}{m} \sum_{i=1}^{m} \tilde{Q}(z_{u,i}) \tilde{Q}(z_{v,i}), \qquad (7)$$

with $z_{u,i} = \cos(w_i^T u + \tau_i)$ and $z_{v,i} = \cos(w_i^T v + \tau_i)$ the i-th unscaled RFF of u and v, respectively. Moreover, for the proposed LM quantizers, we consider *normalized estimator*,

$$\hat{K}_{n,Q}(u, v) = \frac{\sum_{i=1}^{m} Q(z_{u,i}) Q(z_{v,i})}{\sqrt{\sum_1^m Q(z_{u,i})^2} \sqrt{\sum_1^m Q(z_{v,i})^2}}. \qquad (8)$$

This estimator can also be conveniently used, as we only need to normalize the quantized RFFs (per data point) before learning. We will analyze and compare the estimators using different quantization methods, theoretically and practically.

4. Theoretical Analysis

In this section, we analyze the mean, variance, and monotonicity properties of the quantized kernel estimators. The proofs are deferred to Appendix D.

4.1. StocQ Estimator

We start this section by considering the stochastic rounding method for RFF. In Zhang et al. (2019), the exact variance of the kernel estimator is not provided. In the following, we establish the precise variance calculation based on Theorem 2.2, which is in fact a more general result on any symmetric stochastic quantizer.

Theorem 4.1 (StocQ)**.** *Suppose a b-bit StocQ quantizer (5) applies stochastic rounding corresponding to arbitrary bin split $-1 = t_0 < ... < t_{2^b-1} = 1$ that is symmetric about 0. Denote $S_i = t_{i-1}+t_i$ and $P_i = t_{i-1}t_i$, $i = 1, ..., 2^b-1$. Let $f(z_u, z_v)$ be the RFF joint distribution in Theorem 2.2. Denote $\kappa_{i,j} = \int_{t_{i-1}}^{t_i}\int_{t_{j-1}}^{t_j} z_u z_v f(z_u, z_v)dz_u dz_v$, and $p_{i,j} = \int_{t_{i-1}}^{t_i}\int_{t_{j-1}}^{t_j} f(z_u, z_v)dz_u dz_v$. Then we have $\mathbb{E}[\hat{K}_{StocQ}] = K(u, v)$ and $Var[\hat{K}_{StocQ}] = \frac{V_{StocQ}}{m}$, with*

$$V_{StocQ} = 4\sum_{i=1}^{2^b-1}\sum_{j=1}^{2^b-1}\Big[S_iS_j\kappa_{i,j} + P_iP_jp_{i,j}\Big] - K(u, v)^2,$$

which is always greater than $Var[\hat{K}]$ defined in (3).

The important take-away messages are: 1) the StocQ kernel estimator is unbiased of the Gaussian kernel; 2) the variance is always larger than full-precision RFF estimate. Further, we have the following result for 1-bit StocQ, which is a straightforward consequence of Theorem 4.1

Corollary 4.1. *With 1-bit, $Var[\hat{K}_{StocQ}] = \frac{4-K(u,v)^2}{m}$.*

4.2. LM Estimators

We next study the moments of the LM estimators. Let Q denote the LM quantizer from Algorithm 1. Firstly, we have the formulation of the mean estimate of LM quantized estimator (7) based on Chebyshev functional approximation.

Theorem 4.2 (LM)**.** *Let u, v be two normalized data samples with $u^Tv = \rho$, and $\hat{K}_Q$ be as (7) using LM quantizer with distortion D. Let $z_x = \cos(\gamma X+\tau)$, $z_y = \cos(\gamma Y+\tau)$ where $(X, Y) \sim N(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix})$, $\tau \sim uniform(0, 2\pi)$. Let $\theta_{s,t} = \mathbb{E}[z_x^s Q(z_x)^t]$, $\zeta_{s,t} = \mathbb{E}[Q(z_x)^s Q(z_y)^t]$. Further define $\alpha_i = \frac{2}{\pi}\int_{-1}^{1} Q(x)T_i(x)\frac{dx}{\sqrt{1-x^2}}$, $\psi_{i,j} = \mathbb{E}[T_i(z_x)T_j(z_y)]$, where $T_i(x)$ is the i-th Chebyshev polynomial of the first kind. Then we have*

$$\mathbb{E}[\hat{K}_Q] = (1-2D)^2K(u, v) + \sum_{i=1,odd}^{\infty}\sum_{j=3,odd}^{\infty}\alpha_i\alpha_j\psi_{i,j},$$

$$Var[\hat{K}_Q] = \frac{4}{m}(\zeta_{2,2} - \zeta_{1,1}^2).$$

In particular, $\mathbb{E}[\hat{K}_Q] = (1 - 2D)^2K(u, v)$ when $\rho = 0$, and $\mathbb{E}[\hat{K}_Q] = 1 - 2D$ when $\rho = 1$.

Note that, since Chebyshev polynomials form an orthogonal basis of the function space on $[-1, 1]$ with finite number of discontinuities, we can show that $\alpha_i = \sqrt{1-2D} \cdot c_i$ where c_i is the cosine between $Q(x)$ and $T_i(x)$, and $\sum_{i=3,odd}^{\infty}\alpha_i^2 = 2D(1-2D)$ which is typically very small and decreases as the quantizer has more bits. Also, we have $|\psi_{i,j}| \leq \mathbb{E}[T_i(z_x)^2] = 1/2$. Consequently, in Theorem 4.2 the last term approximates zero in most cases. This translates into the following observation.

Observation 4.1. $\mathbb{E}[\hat{K}_Q(u, v)] \approx (1-2D)^2K(u, v)$.

Next, we provide an asymptotic analysis on the normalized quantized kernel estimate (8) under LM scheme.

Theorem 4.3 (Normalized estimator)**.** *Under same setting as Theorem 4.2, as $m \to \infty$,*

$$\mathbb{E}[\hat{K}_{n,Q}] = \frac{\zeta_{1,1}}{\zeta_{2,0}} + \mathcal{O}(\frac{1}{m}),\ \ Var[\hat{K}_{n,Q}] = \frac{V_n}{m} + \mathcal{O}(\frac{1}{m^2}),$$

$$\text{with}\ \ V_n = \frac{\zeta_{2,2}}{\zeta_{2,0}^2} - \frac{2\zeta_{1,1}\zeta_{3,1}}{\zeta_{2,0}^3} + \frac{\zeta_{1,1}^2(\zeta_{4,0}+\zeta_{2,2})}{2\zeta_{2,0}^4}.$$

In particular, $\mathbb{E}[\hat{K}_{n,Q}] = K(u, v) = 1$ when $\rho = 1$.

By the property of LM quantization, $\zeta_{2,0} = \frac{1}{2} - D$. Thus,

Observation 4.2. $\mathbb{E}[\hat{K}_{n,Q}] \approx (1-2D)K(u, v)$, $m \to \infty$.

Observation 4.1 and 4.2 says that, $\mathbb{E}[\hat{K}_Q]$ and $\mathbb{E}[\hat{K}_{n,Q}]$ approximately equal to some scaled version of true kernel, which will motivate our discussion in Section 5.4 on the robust kernel approximation error metrics.

Validation. We plot the empirical bias of LM-RFF against Observations 4.1 and 4.2 in Figure 3. As we see, the proposed surrogates for bias align with true biases very well when ρ is not very close to 1. The biases shrink to 0 as b increases (e.g., negligibly $\mathcal{O}(10^{-3})$ with $b = 4$). As $\rho \to 1$, at some "disjoint point" the absolute biases have sharp drops and quickly converge to the theoretical values (red dots) given in Theorem 4.2 and 4.3. As b or γ increases, the “disjoint point” gets closer to $\rho = 1$.

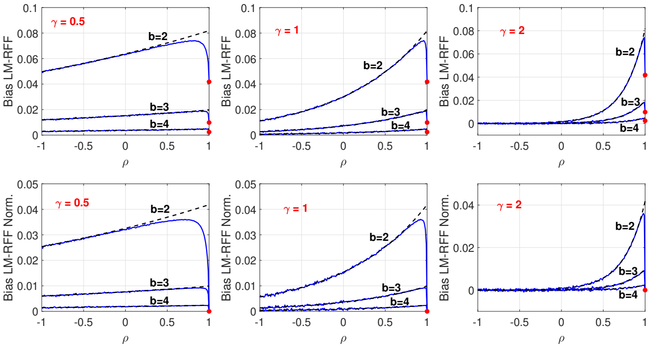

Figure 3. Observations 4.1 and 4.2 (black dash curves) vs. empirical bias (blue curves) of LM-RFF. Red dots are the biases given by the theorem at specific ρ values.

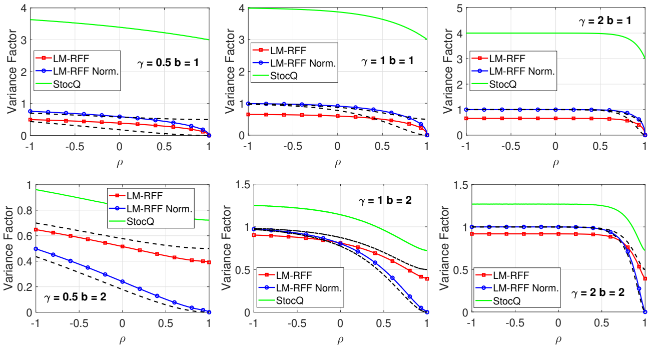

Figure 4. Variance (scaled by m) of StocQ, LM, and normalized LM kernel estimators, with $\gamma = 0.5, 1, 2$. The dashed curves are the variances of corresponding full-precision estimators, to which the variance of quantized estimators converges as b increases.

4.3. Variance Comparisons

Figure 4 provides variance comparisons. As b gets larger, the variances of quantized estimators converge to those of full-precision estimators. The variance of LM-RFF is significantly smaller than StocQ quantization, especially when $b = 1, 2$. This, together with the fact that the expectation of LM-RFF estimators can be approximately written as $cK + \delta$ where δ is some small term (Observation 4.1 & 4.2), to a good extent explains why StocQ performs poorly in approximate kernel learning with low bits (Section 5).

Variance of debiased kernel estimates. As LM estimators are slightly biased which brings theoretical challenges on finding a method to "properly" compare their variances, we investigate the concept of "debiased variance", which refers to the estimator variance after bias corrections.

Definition 4.1 (DB-variance). *For normalized data points u and v with $\rho = u^T v$, and a kernel estimator $\hat{K}(u, v; \rho)$ with $\mathbb{E}[\hat{K}] := E(\rho) > 0$ and $Var[\hat{K}] := V(\rho)$, the debiased variance of $\hat{K}(u, v; \rho)$ at ρ is $Var^{db}[\hat{K}] = V(\rho)\frac{K(\rho)^2}{E(\rho)^2}$.*

Intuitively, Definition 4.1 is reasonable in that it compares the variation of different estimation procedures given that they have same mean. It is worth mentioning that, DB-variance is invariant of linear scaling, i.e., $c\hat{K}$ and $\hat{K}$ have same DB-variance for $c > 0$. Classical metrics for estimation quality, such as the Mean Squared Error (MSE), might be largely affected by such simple scaling. Note that, the DB-variance of all 1-bit estimators (both simple and normalized) from fixed symmetric quantizers are identical. This can be easily verified by writing every 1-bit quantizer as $Q(z) = sign(z) \cdot C_Q$ for some $C_Q > 0$ in (7) and (8). Thus, we will focus on multi-bit quantizers (i.e., $b \geq 2$).

Benefit of normalization. Next we prove the theoretical merit of normalizing RFFs, in terms of DB-variance.

Theorem 4.4. *Suppose u, v are two samples with correlation ρ. Let the simple and normalized kernel estimator, $\hat{K}_Q$ and $\hat{K}_{n,Q}$, be defined as (7) and (8), respectively, where Q is the LM-RFF quantizer. Assume $\gamma \leq \pi/\sqrt{2}$. Then, $Var^{db}[\hat{K}_{n,Q}] \leq Var^{db}[\hat{K}_Q]$ on $\rho \in [0, 1]$ as $m \to \infty$.*

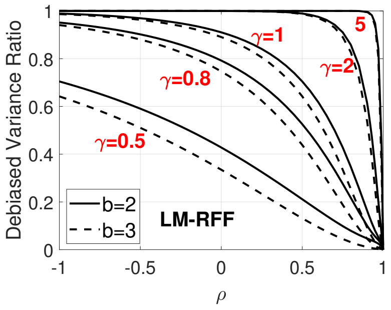

Figure 5. Debiased variance ratio, $\frac{Var^{db}[\hat{K}_{n,Q}]}{Var^{db}[\hat{K}_Q]}$, of normalized LM estimator against simple LM estimator, $b = 2, 3$.

Theorem 4.4 implies that when $\gamma \leq \pi/\sqrt{2} \approx 2.2$, normalization is guaranteed to reduce the DB-variance $\forall \rho \in [0, 1]$. In Figure 5, we plot the DB-variance ratio of $\frac{Var^{db}[\hat{K}_{n,Q}(x,y)]}{Var^{db}[\hat{K}_Q(x,y)]}$ at multiple γ and b for LM-RFF. We corroborate the advantage of normalized estimates over simple estimators in terms of DB-variance (ratio always < 1), especially with large ρ. The same conclusion appears to also hold for $\rho < 0$, but a technical proof might be difficult.

4.4. Monotonicity of Mean Kernel Estimation

For a kernel estimator $\hat{K}(\rho)$ (written as a function of ρ), the monotonicity of its mean estimation $\mathbb{E}[\hat{K}(\rho)]$ against ρ is important to ensure its "correctness". It guarantees that asymptotically ($m \to \infty$), the comparison of estimated kernel distances is always correct, i.e., $\hat{K}(u, v_1) > \hat{K}(u, v_2)$ if $K(u, v_1) > K(u, v_2)$ for data points u, v_1, v_2. Otherwise (say, $\mathbb{E}[\hat{K}]$ decreasing in ρ on $[s, t]$), the comparison of estimated kernel would be wrong for $\rho \in [s, t]$ even with infinite much data. By Theorem 4.1, StocQ estimator is unbiased with $\mathbb{E}[\hat{K}_{StocQ}] = e^{-\gamma^2(1-\rho)}$ strictly increasing in ρ. Hence, we will focus on the fixed LM quantization.

The following lemma gives the exact derivative of interest with continuous functions cast on RFF.

Lemma 4.5. *Suppose X, Y, τ are same as Theorem 4.2, and denote $s_x = \gamma X + \tau$, $s_y = \gamma Y + \tau$, such that $z_x = \cos(s_x)$ and $z_y = \cos(s_y)$ are RFFs. Assume $g_1, g_2 : [-1, 1] \mapsto \mathbb{R}$ are twice differentiable and bounded functions. Then,*

$$\frac{\partial \mathbb{E}[g_1(z_x)g_2(z_y)]}{\partial \rho} = \gamma^2 \mathbb{E}[g_1'(z_x)\sin(s_x)g_2'(z_y)\sin(s_y)].$$

Furthermore, when $\sqrt{2(1-\rho)}\gamma \leq \pi$, if g_1 and g_2 are both increasing odd functions or non-constant even functions, then the mean is increasing in ρ, i.e., $\frac{\partial \mathbb{E}[g_1(z_x)g_2(z_y)]}{\partial \rho} > 0$.

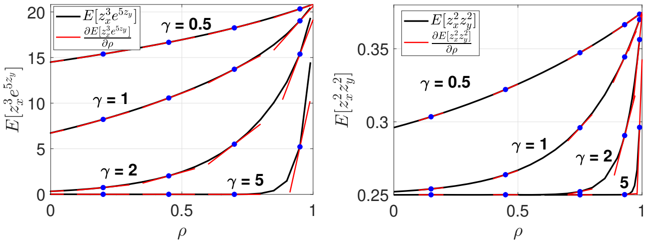

Figure 6. Validation of Lemma 4.5 with different g_1 and g_2, at multiple γ. Black curves are the function value, and red lines are the theoretical derivatives. **Left:** $g_1(x) = x^3$, $g_2(x) = e^{5x}$, increasing functions. **Right:** $g_1(x) = g_2(x) = x^2$, even functions.

Lemma 4.5 is a general result for the monotonicity when RFFs are processed by continuous functions. In Figure 6, we plot two examples of $\mathbb{E}[g_1(s_x)g_2(s_y)]$ against ρ, with continuously increasing functions $g_1(x) = x^3$ and $g_2(x) = e^{5x}$ and even functions $g_1(x) = g_2(x) = x^2$, respectively. As we can see, the expectation is increasing in ρ with true derivatives matching Lemma 4.5.

The next Theorem extends the above result to discrete functions, which include our LM quantizers as special cases.

Theorem 4.6. *Suppose Q_1 and Q_2 are bounded, discrete, and non-decreasing odd functions or non-constant even functions, with finite many discontinuities. Let z_x and z_y be defined as Lemma 4.5. If $\sqrt{2(1-\rho)}\gamma \leq \pi$, then $\mathbb{E}[Q_1(z_x)Q_2(z_y)]$ is increasing in ρ.*

Remark 4.1. *The condition $\sqrt{2(1-\rho)}\gamma \leq \pi$ in Theorem 4.6 implies that $\mathbb{E}[Q(z_x)Q(z_y)]$ for LM quantizer increases in $\rho \in [\max(-1, 1 - \frac{\pi^2}{2\gamma^2}), 1]$. Thus, larger γ typically requires higher ρ for this condition to hold. For example, when $\gamma = 1$ and $\gamma = 5$, monotonicity is ensured for $\rho \in [-1, 1]$ and $\rho \geq 0.8$, respectively. See Appendix B.2.*

5. Experiments

We conduct experiments with compressed RFFs on three popular learning tasks: kernel SVM (KSVM), kernel logistic regression (KLR) and kernel ridge regression (KRR). The summary statistics of datasets is given in Table 1. We will address the main results and place more detailed description and implementation in Appendix C.

Setup. For a dataset $U \in \mathbb{R}^{n \times d}$, we generate $m = 2^6 \sim 2^{16}$ RFFs for each sample. Three compression approaches are tested: 1) LM-RFF; 2) LM-RFF normalized; 3) StocQ with uniform borders (stochastic rounding). The RFFs are first quantized, and then fed into the target linear learner. For each task and m, we search for a best γ on fine grid for full-precision RFF, and use the same γ for all quantized RFF.

5.1. Kernel SVM (KSVM)

Firstly, we test the quantization methods on kernel SVM (KSVM) classification problem. We randomly split each dataset into 60% for training and 40% for testing, and preprocess the datasets by instance normalization. LIBLINEAR (Chang and Lin, 2011) is used as the solver. The parameter C in SVM is fine tuned for every compression method, b and m respectively. The best test accuracy is reported. All results are averaged over 10 independent runs.

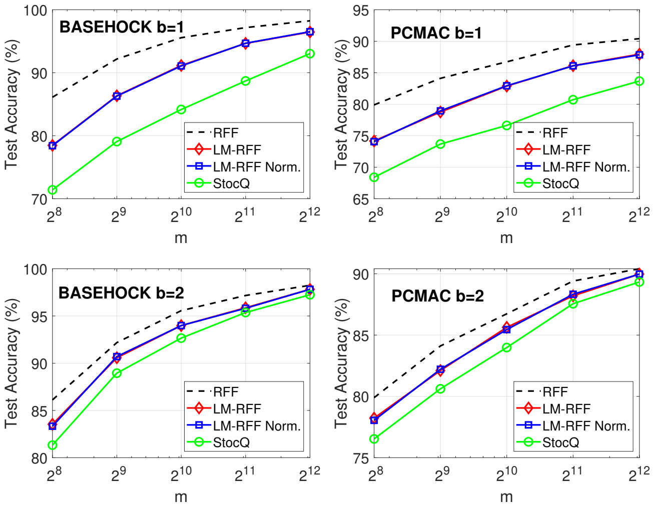

Figure 7. The test accuracy of kernel SVM using different compression schemes of RFFs vs. number of random features m.

To directly compare the learning power of different compression schemes, we provide the test accuracy vs. number of RFFs in Figure 7. We observe: 1) low-bit StocQ performs poorly and is outperformed by LM-RFF; 2) On all datasets, LM-RFF with $b = 2$ already approaches the accuracy of full-precision RFF with moderate $m \approx 4000$, indicating the superior learning capacity of LM-RFF under deep feature compression. When b gets larger, the performance of StocQ approaches that of LM-RFF, both approximating the full-precision RFF, as one would expect.

To characterize the memory efficiency, on each dataset, we first find the highest test accuracy (among m) of full-precision RFF, which requires M_{FP} bits per sample. Then, for each method we find the model (among b and m) with

Table 1. **Column 2-6:** Summary statistics of all datasets used in our experiments. **Column 7-8:** Compression ratio of different quantization schemes against full-precision RFF, when trained to match the test accuracy of the best full-precision RFF.

Model	Source	Dataset	n	d	# Class	LM-RFF	StocQ
KSVM	ASU-DB (Li et al., 2016)	BASEHOCK	$1,993$	$4,862$	2	8x ($b=2$)	5x ($b=4$)
		PCMAC	$1,943$	$3,289$	2	21x ($b=3$)	8x ($b=4$)
KLR	LIBSVM (Chang and Lin, 2011)	Webspam	$350,000$	254	2	21x ($b=1$)	11x ($b=2$)
		CoverType	$581,192$	54	2	53x ($b=1$)	7x ($b=4$)
KRR	Synthetic	Sim	$50,000$	10	-	4x ($b=4$)	2x ($b=4$)

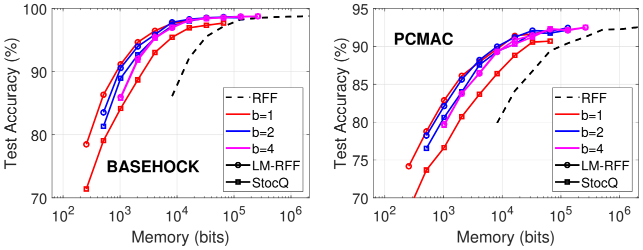

Figure 8. Test accuracy of KSVM using different compression schemes of RFFs vs. number of bits (memory usage) per sample.

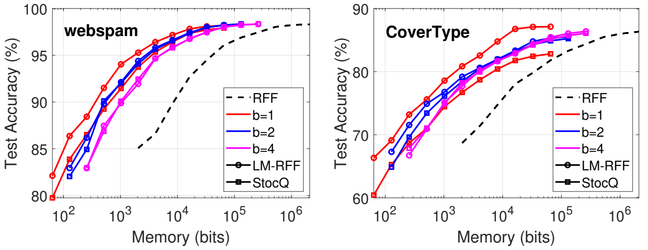

Figure 9. KLR: Test accuracy vs. number of bits (memory usage) per sample. Linear kernel: 91.5% and 75.5%, respectively.

smallest memory footprint to reach $\pm 0.2\%$ (to allow for some noise) of the best full-precision accuracy. Denote the number of total bits required as M_{LM} and M_{Stoc}, respectively. The compression ratio is then computed as $\frac{M_{FP}}{M_{LM}}$ and $\frac{M_{FP}}{M_{Stoc}}$. Here we assume that full-precision RFFs are represented by 32 bits, though it might require even more in practice. For robustness, we report the average compression ratio w.r.t. top 3 full-precision accuracies. The result is summarized in Figure 8 where we present LM-RFF as the representative of LM-type methods. A curve near upper-left corner is more desirable, which means that the method requires less memory to achieve some certain test accuracy. As we see, 1-bit or 2-bit LM-RFF generally perform the best on all datasets. From Table 1, we observe that LM-RFF consistently offers considerably higher compression ratio than StocQ on all datasets. Additionally, LM-RFF typically requires fewer-bit quantizers (smaller b) than StocQ to match the accuracy of best full-precision RFF.

5.2. Kernel logistic regression (KLR)

For kernel logistic regression (KLR), we adopt a different training mechanism on two popular datasets from LIBSVM (Chang and Lin, 2011) website. For `Webspam`, we use the standard train/test split provided by the online repository, where each sample is normalized to unit norm. For `CoverType`, the dataset is randomly divided into training and test set with equal size, and for generality we do not apply any pre-processing and directly use the raw data.

For this task, we train logistic regression using Stochastic Gradient Descent (SGD) with cross-entropy loss and mini-batch size 500. For each method and m, we tune the learning rate over a fine grid, and the optimal l_2 regularization term λ is also searched on the log-scale. We train the models for at least 50 epochs until the test accuracy stabilizes.

In Figure 9, we plot test accuracy vs. memory utilization for KLR. Firstly, learning with (quantized) RFFs gives significantly better performance compared with linear kernel (91.5% and 75.5%, respectively). Similar to KSVM, LM-RFF dominates on both datasets, significantly better than StocQ with $b = 1$ and 2. In Table 1, LM-RFF attains much higher compression ratio than StocQ (with fewer bits). On `CoverType` for instance, 1-bit LM-RFF matches full-precision RFF with $\sim$50x reduced memory.

5.3. Kernel ridge regression (KRR)

We use a synthetic dataset admitting high non-linearity for kernel ridge regression (KRR). Precisely, each data sample $u_i \in \mathbb{R}^{10}$ is drawn from i.i.d. $N(0, 1)$. We generate the response by $y_i = \sum_{p=1}^{3} \beta_p u_i^p + \epsilon$, where $\beta_1 = [1, 2, ..., 10]$, $\beta_2 = [1, 1, ..., 1]$, β_3 and ϵ also follow i.i.d. $N(0, 1)$. We simulate $40,000$ independent samples for training and $10,000$ for testing. The training and tuning procedure is similar to that for KLR, where SGD with mini-batch size 100 is implemented with MSE loss. We train each model for at least 100 epochs until the test MSE converges.

We summarize KRR results in Figure 10. Again, with same b and number of RFFs, LM-RFF consistently beats StocQ (1-bit LM-RFF even outperforms 2-bit StocQ). In the right panel, we present the memory efficiency comparison. Note that, due to high-order terms in the true model, the test MSE of linear kernel is 20.8, while learning with full-precision RFF significantly reduces it to 3.5. To match the test MSE of full-precision RFF (with ± 0.2 tolerance), LM-RFF merely needs half the memory consumption of StocQ (Table 1).

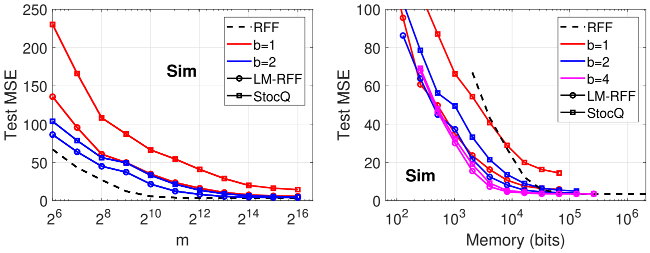

Figure 10. Test MSE of KRR. Linear kernel: 20.8.

Note that 1-bit and 2-bit LM-RFF yield 5.9 and 4.1 test MSE respectively, which are already quite close to 3.5, while for 1-bit and 2-bit StocQ, the test losses are 14.5 and 5.0 respectively, much worse than those of LM-RFF.

5.4. Scale-invariant Kernel Approximation Error

Recall the notation $U = [u_1, ..., u_n]^T$ as the data matrix. Let $\mathcal{K}$ be the $n \times n$ Gaussian kernel matrix, with $\mathcal{K}_{ij} = K(u_i, u_j)$. Denote $\hat{\mathcal{K}}$ as the estimated kernel matrix by an approximation algorithm. Kernel Approximation Error (KAE) has been shown to play an important role in the generalization of learning with random features, including the norms (Cortes et al., 2010; Gittens and Mahoney, 2013; Sutherland and Schneider, 2015) of $\hat{\mathcal{K}} - \mathcal{K}$ and spectral approximations (Bach, 2013; Alaoui and Mahoney, 2015; Avron et al., 2017; Zhang et al., 2019). We investigate the KAEs to better justify the impressive generalization ability of LM-RFF from a theoretical aspect.

Existing KAE metrics are not robust to bias. Consider $\mathbb{E}[\hat{\mathcal{K}}] = \beta\mathcal{K}$ with some $\beta > 0$. Obviously, learning with $\beta\mathcal{K}$ is equivalent to learning with $\mathcal{K}$ for kernel-distance based models, since with proper scaling of model parameters, the objective functions/predictions are invariant of multiplying the input kernel matrix with a scalar. However, traditional KAEs do not generalize to this case. For example, when $\beta = 0.1$, the 2-norm error $\|0.1\hat{\mathcal{K}} - \mathcal{K}\|_2$ would be very large. To make the KAE metrics more robust, we define the scale-invariant KAE metrics as follows.

Definition 5.1 (Scale-Invariant KAE). *Let $\mathcal{K}$ be a kernel matrix and $\hat{\mathcal{K}}$ be its approximation. Define 2-norm metric*

$$\|\hat{\mathcal{K}} - \mathcal{K}\|_2^* = \min_{\beta>0} \|\beta\hat{\mathcal{K}} - \mathcal{K}\|_2.$$

Denote the minimizer as β^*. The spectral approximation is

$$(\delta_1^*, \delta_2^*) = \inf_{(\delta_1,\delta_2)\geq 0} \{\delta_1, \delta_2 : (1-\delta_1)\mathcal{K} \preccurlyeq \beta^*\hat{\mathcal{K}} \preccurlyeq (1+\delta_2)\mathcal{K}\}.$$

Note that, we can also define additional sets of metrics by, for example, Frobenius norm. Our new KAE metrics are more general, adapted to the best scaling factor β^* of the estimated kernel. Since LM-RFF estimators are slightly biased (recall Observations 4.1 and 4.2), Definition 5.1 is important for appropriately evaluating the LM-RFF kernel estimation approach. In Figure 11, we provide scale-invariant KAE metrics on BASEHOCK and PCMAC dataset. As we can see, LM-RFF always has smaller KAEs than StocQ with equal bits. In particular, with extreme 1-bit compression, StocQ has exceedingly large loss due to its large variance, while in many cases the KAEs of 1-bit LM-RFF are already quite small. The KAE comparison well aligns with, and to a good extent explains, our empirical results that 1) LM-RFF consistently outperforms StocQ, and 2) low-bit StocQ generalizes poorly. Thus, it provides a general justification of the superior effectiveness of LM-RFF in machine learning.

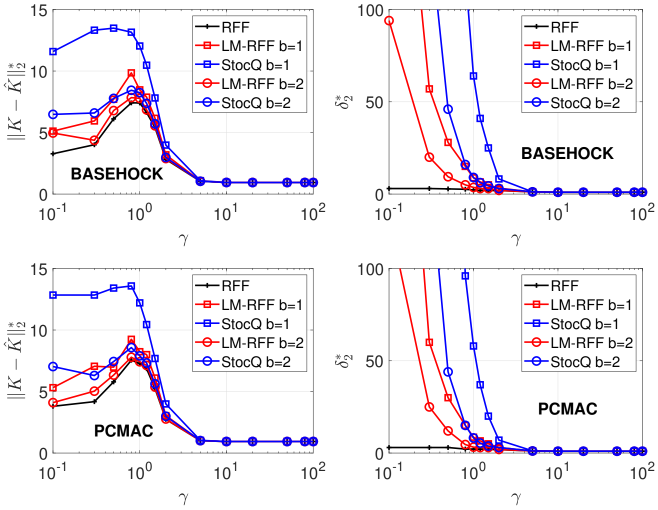

Figure 11. Scale-invariant KAE (Definition 5.1) of LM-RFF vs. StocQ, $m = 2^{10}$. **Left:** 2-norm. **Right:** spectral approximation. For both metrics, the smaller the better.

6. Conclusion

The technique of random Fourier features (RFF) is a popular method to solve the computational bottleneck in large-scale (Gaussian) kernel learning tasks. In this paper, we study quantization methods to compress RFFs for substantial memory savings and efficient computations. In particular, we develop LM-RFF quantization scheme based on the Lloyd-Max (LM) *distortion minimization* framework. According to our analysis on the probability distribution of RFF, the LM-RFF quantizer design is very simple as only one quantizer is needed for all γ values in the Gaussian kernel. In addition, we also analyze a method based on stochastic rounding (StocQ). Both theoretically and empirically, LM-RFF significantly outperforms StocQ on many tasks, especially when the number of bits is not large. Compared to full-precision (e.g., 32- or 64-bit) RFFs, the experiments demonstrate that often a 2-bit LM-RFF quantizer achieve comparable performance with full-precision RFF, at a substantial saving in memory cost, which would be highly beneficial in practical applications.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

Raja Hafiz Affandi, Emily B. Fox, and Ben Taskar. Approximate inference in continuous determinantal processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1430–1438, Lake Tahoe, NV, 2013.

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

Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 253–262, Sydney, Australia, 2017.

Francis R. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 185–209, Princeton University, NJ, 2013.

René Baire and Arnaud Denjoy. *Leçons sur les fonctions discontinues: professées au Collège de France*. Gauthier-Villars, 1905.

RCM Barnes, EH Cooke-Yarborough, and DGA Thomas. An electronic digital computor using cold cathode counting tubes for storage. *Electronic Engineering*, 1951.

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 245–250, San Francisco, CA, 2001.

Peter Borwein and Tamás Erdélyi. *Polynomials and polynomial inequalities*, volume 161. Springer Science & Business Media, 1995.

Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors. *Large-Scale Kernel Machines*. The MIT Press, Cambridge, MA, 2007.

Jeremy Buhler. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 17(5):419–428, 2001.

Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 380–388, Montreal, Canada, 2002.

Kacper Chwialkowski, Aaditya Ramdas, Dino Sejdinovic, and Arthur Gretton. Fast two-sample testing with analytic representations of probability measures. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1981–1989, Montreal, Canada, 2015.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 113–120, Chia Laguna Resort, Sardinia, Italy, 2010.

Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3041–3049, Montreal, Canada, 2014.

Sanjoy Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 143–151, Stanford, CA, 2000.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokn. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th ACM Symposium on Computational Geometr (SCG)*, pages 253 – 262, Brooklyn, NY, 2004.

David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference (ICML)*, pages 186–193, Washington, DC, 2003.

George E Forsythe. Round-off errors in numerical integration on automatic machinery-preliminary report. In *Bulletin of the American Mathematical Society*, volume 56, pages 61–61, 1950.

George E Forsythe. Reprint of a note on rounding-off errors. *SIAM Review*, 1(1):66, 1959.

Yoav Freund, Sanjoy Dasgupta, Mayank Kabra, and Nakul Verma. Learning the structure of manifolds using random projections. In *Advances in Neural Information Processing Systems (NIPS)*, pages 473–480, Vancouver, Canada, 2007.

Alex Gittens and Michael W. Mahoney. Revisiting the nyström method for improved large-scale machine learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 567–575, Atlanta, GA, 2013.

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

Trevor J. Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning:Data Mining, Inference, and Prediction*. Springer, New York, NY, 2001.

Felix Hausdorff. *Set theory*. American Mathematical Society (RI), 1991.

José Miguel Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 918–926, Montreal, Canada, 2014.

Cho-Jui Hsieh, Si Si, and Inderjit S. Dhillon. Fast prediction for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3689–3697, Montreal, Canada, 2014.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 604–613, Dallas, TX, 1998.

Laurent Jacques, Jason N. Laska, Petros T. Boufounos, and Richard G. Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Trans. Inf. Theory*, 59(4):2082–2102, 2013.

Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226, Philadelphia, PA, 2006.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Cong Leng, Jian Cheng, and Hanqing Lu. Random subspace for binary codes learning in large scale image retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1031–1034, Gold Coast, Australia, 2014.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Trevino Robert, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *arXiv:1601.07996*, 2016.

Ping Li. Very sparse stable random projections for dimension reduction in l_α $(0 < \alpha \leq 2)$ norm. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 440–449, San Jose, CA, 2007.

Ping Li. Binary and multi-bit coding for stable random projections. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1430–1438, Fort Lauderdale, FL, 2017a.

Ping Li. Linearized GMM kernels and normalized random Fourier features. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 315–324, 2017b.

Ping Li and Martin Slawski. Simple strategies for recovering inner products from coarsely quantized random projections. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4567–4576, Long Beach, CA, 2017.

Ping Li, Trevor J. Hastie, and Kenneth W. Church. Improving random projections using marginal information. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, pages 635–649, Pittsburgh, PA, 2006.

Ping Li, Michael Mitzenmacher, and Anshumali Shrivastava. Coding for random projections. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 676–684, Beijing, China, 2014.

Xiaoyun Li and Ping Li. Generalization error analysis of quantized compressive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2019a.

Xiaoyun Li and Ping Li. Random projections with asymmetric quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2019b.

Xiaoyun Li and Ping Li. One-sketch-for-all: Non-linear random features from compressed linear measurements. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2647–2655, Virtual Event, 2021.

Xiaoyun Li, Jie Gui, and Ping Li. Randomized kernel multi-view discriminant analysis. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, pages 1276–1284, Santiago de Compostela, Spain, 2020.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28(2):129–136, 1982.

Joel Max. Quantizing for minimum distortion. *IRE Trans. Information Theory*, 6(1):7–12, 1960.

John C. Platt. Using analytic QP and sparseness to speed training of support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 557–563, Denver, CO, 1998.

Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1509–1517, Vancouver, Canada, 2009.

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

Emile Richard, Georges Goetz, and E. J. Chichilnisky. Recognizing retinal ganglion cells in the dark. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2476–2484, Montreal, Canada, 2015.

Walter Rudin. *Fourier Analysis on Groups*. John Wiley & Sons, New York, NY, 1990.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3330–3338, Montreal, Canada, 2015.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.*, 127(1):3–30, 2011.

Martin Slawski and Ping Li. On the trade-off between bit depth and number of samples for a basic approach to structured signal recovery from b-bit quantized linear measurements. *IEEE Trans. Inf. Theory*, 64(6):4159–4178, 2018.

Yitong Sun, Anna C. Gilbert, and Ambuj Tewari. But how does it work in theory? linear SVM with random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3383–3392, Montréal, Canada, 2018.

Danica J. Sutherland and Jeff G. Schneider. On the error of random fourier features. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 862–871, Amsterdam, The Netherlands, 2015.

Anthony Tompkins and Fabio Ramos. Fourier feature approximations for periodic kernels in time-series modelling. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 4155–4162, New Orleans, LA, 2018.

Bernard Widrow and István Kollár. Quantization noise. *Cambridge University Press*, 2008.

Xiaolin Wu. On convergence of lloyd's method I. *IEEE Trans. Inf. Theory*, 38(1):171–174, 1992.

Zhiqiang Xu, Dong Li, Weijie Zhao, Xing Shen, Tianbo Huang, Xiaoyun Li, and Ping Li. Agile and accurate CTR prediction model training formassive-scale online advertising systems. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*, Online conference [Xi'an, China], 2021.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems (NIPS)*, pages 485–493, Lake Tahoe, NV, 2012.

Ian En-Hsu Yen, Ting-Wei Lin, Shou-De Lin, Pradeep Ravikumar, and Inderjit S. Dhillon. Sparse random feature algorithm as coordinate descent in hilbert space. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2456–2464, Montreal, Canada, 2014.

Jian Zhang, Avner May, Tri Dao, and Christopher Ré. Low-precision random fourier features for memory-constrained kernel approximation. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1264–1274, Naha, Okinawa, Japan, 2019.

Argyrios Zymnis, Stephen P. Boyd, and Emmanuel J. Candès. Compressed sensing with quantized measurements. *IEEE Signal Process. Lett.*, 17(2):149–152, 2010.