One-Sketch-for-All: Non-linear Random Features from Compressed Linear Measurements

Xiaoyun Li
Department of Statistics
Rutgers University
110 Frelinghuysen Rd. Piscataway, NJ 08854
xiaoyun.li@rutgers.edu

Ping Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004
liping11@baidu.com

Abstract

RFF (random Fourier features) is a popular technique for approximating the commonly used Gaussian kernel. Due to the crucial tuning parameter γ in the Gaussian kernel, the design of effective quantization schemes for RFF appears to be challenging. Intuitively one would expect that a different quantizer is needed for a different γ value (and we need to store a different set of quantized data for each γ). Interestingly, the recent work (Li and Li, 2021) showed that only one Lloyd-Max (LM) quantizer is needed by showing that the marginal distribution of RFF is free of the tuning parameter γ. On the other hand, Li and Li (2021) still required to store a different set of quantized data for each γ value.

In this paper, we adopt the "one-sketch-for-all" paradigm for quantizing RFFs. Basically, we only store one set of quantized linear sketches after applying random projections on the original data. From the same set of quantized data, we construct RFFs to approximate Gaussian kernels for any tuning parameter γ. Compared with Li and Li (2021), our proposed "one-sketch-for-all" scheme would inevitably lose some accuracy as one should expect. Nevertheless, our proposed method still performs noticeably better than other quantization algorithms such as stochastic rounding. We provide statistical analysis on properties of the proposed quantization method, and conduct experiments to empirically illustrate its effectiveness.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 Introduction

Non-linear kernels are proven more powerful than linear kernel in various machine learning tasks. Given two (normalized) data vectors $x, y \in \mathbb{S}^{d-1}$ with $\rho = \cos(x, y)$, i.e., the "cosine similarity" between x and y, in this paper we consider the following well-known RBF (Gaussian) kernel defined as

$$K(x, y) = e^{-\frac{\gamma^2\|x-y\|^2}{2}} = e^{-\gamma^2(1-\rho)}, \tag{1}$$

where γ is a tuning parameter. Here, we assume that the data space belongs to the unit sphere for the ease of presentation. Given a dataset composing n samples, standard kernel methods require computing the $n \times n$ kernel matrix consisting of the kernel values between all pairs of samples. In large-scale applications (large n), however, the memory and computational cost would explode (Bottou et al., 2007). To resolve this bottleneck, the scheme of random Fourier features (RFF) (Rahimi and Recht, 2007) provides an effective way to *linearize* the non-linear kernel by approximation. It is an application of Bochner's Theorem (Rudin, 1990), which says that a shift-invariant kernel is positive definite (which is true for RBF kernel) if and only if it is the inverse Fourier transform of a non-negative measure Ψ. It then holds that

$$\begin{aligned} K(x, y) = (\mathcal{F}^{-1}\Psi)(x-y) &= \int e^{iv^T(x-y)} d\Psi(w) \\ &= \mathbb{E}_{w\sim\Psi}[\cos(w^T(x-y))], \end{aligned}$$

where $\mathcal{F}$ denotes the Fourier transform operator. There are actual two popular formulations of RFF, and in this paper we consider the following form

$$F(x) = [\sin(w^T x) \quad \cos(w^T x)]^T \tag{2}$$

where $w \sim N(0, \gamma^2 I_d)$. This formulation is known to have smaller kernel estimation variance (Sutherland and Schneider, 2015) than the other formulation,[1]

[1] As shown in Li (2017), the variance of this RFF formulation can be substantially reduced by a normalization step.

i.e., $F(x) = \sqrt{2}\cos(w^T x + \tau)$ with $\tau \sim unif(0, 2\pi)$. Of course, our proposed approach can be applied to both formulations of RFF. Note that, this formulation $F(x) = \sqrt{2}\cos(w^T x + \tau)$ was considered in Li and Li (2021) due to its convenience for LM quantizer design.

With the formulation in Eq. (2), the inner product admits $\mathbb{E}[F(x)^T F(y)] = e^{-\gamma^2(1-\rho)} = K(x, y)$. Hence, by using k independent w_i to generate i.i.d. random features F_i, $i = 1, ..., k$, we obtain an unbiased kernel estimator as

$$\hat{K}(x, y) = \frac{1}{k}\sum_{i=1}^{k} F_i(x)^T F_i(y) \approx K(x, y), \tag{3}$$

where we treat each RFF as a 2-dimensional vector. Imposing linear kernel on the RFFs would be equivalent to learning with the RBF kernel on the original data. This builds the foundation of approximate non-linear learning with RFF, which has 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, 2017; Avron et al., 2017; Sun et al., 2018; Tompkins and Ramos, 2018; Li et al., 2020).

In practice, storing full-precision RFF (non-linear sketches) sometimes is not feasible due to memory constraints. In this case, further condensing the RFFs becomes important, by quantizing the full-precision RFFs ($F(x)$ in Eq. (2)) into low-bit (integer) representations by $Q(F(x))$, where Q is a general quantizing function. For example, Li and Li (2021) studied distortion optimal quantizer design for RFFs, and showed its superior performance in approximate non-linear kernel learning. Particularly, Li and Li (2021) showed that in many cases, using about 4 bits suffices to match the performance of full-precision RFF, suggesting a substantial reduction of the memory/storage cost.

In Eq. (2), constructing RFFs can be viewed as a two-stage procedure: (i) random projection (RP): $w^T x$; (ii) applying non-linearity (sine and cosine functions). Using the quantization scheme developed in Li and Li (2021), one would have to store a set of quantized RFFs for each different tuning parameter γ. When the best tuning parameter is already known (e.g., from prior experience), then the methods in Li and Li (2021) should be adopted. In practice, however, the best tuning parameter might be unknown especially in the early stage of exploration. This means practitioners might have to store multiple (or many) sets of quantized RFFs. This motives us to develop alternative schemes to avoid the burden of storage.

In our proposed scheme, the quantization is applied **before the non-linearity**. That said, we first derive the quantized RP as $Q(w^T x)$ in step (i), which is then used to construct non-linear RFF in step (ii). We call it the "QRP-RFF" scheme. The title of our paper, inspired by Gilbert et al. (2007); Li et al. (2008), characterizes the key advantage of the proposed QRP-RFF approach. That is, it achieves "one-sketch-for-all" because we only need one set of highly compressed linear measurements (i.e., quantized RPs), for both linear and non-linear learning. In this paper, we provide the theoretical analysis on the QRP-RFF kernel estimator and the approximation error.

1.1 Practical significance

The method of random projections (RP) has become the standard tool in machine learning, data mining, and many other applications (Johnson and Lindenstrauss, 1984; 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). As mentioned before, our QRP-RFF framework allows one to only store the quantized random projections (QRPs) in the database, without requiring access to the full-precision RPs (FP-RPs). Note that, the "intermediate product", namely the QRP, is itself a useful tool in machine learning, with a wide range of applications in theory, linear learning, similarity search, compressed sensing, etc. (Goemans and Williamson, 1995; Charikar, 2002; Zymnis et al., 2010; Boufounos and Baraniuk, 2008; Datar et al., 2004; Plan and Vershynin, 2013; Gopi et al., 2013; Li et al., 2014; Li and Slawski, 2017; Li and Li, 2019b,a).

Consider the following practical scenario, where a server has collected RPs of massive data samples and

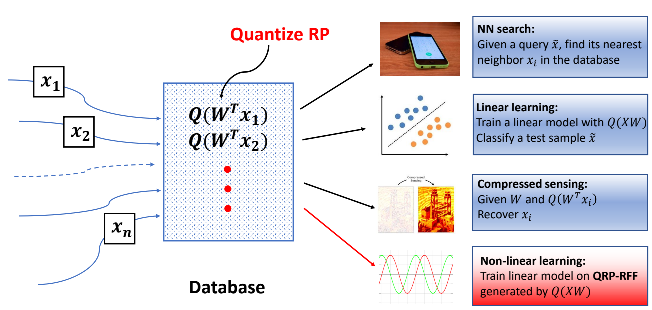

Figure 1: Applications of QRP in large-scale systems. Here, the full-precision RPs are never stored in memory. QRP can be used for linear learning and compressed sensing. The box in red is the new application studied in this paper—constructing non-linear random features for non-linear learning.

stored the quantized RPs (QRPs) in the database to save storage. In this procedure, we have lost access to the full-precision RPs (otherwise, quantization becomes meaningless). In order to achieve better learning performance, a data scientist wants to apply non-linear kernel learning. Typically, this can be done in a standard way by learning with RFFs generated by full-precision RPs (FP-RPs). Yet, they have been discarded after quantization, and re-collecting the data might be inconvenient or even impossible (e.g., due to data loss or privacy). Our QRP-RFF method exactly provides a solution in this case, by directly extracting non-linear sketches from QRPs. Therefore, QRP-RFF can be viewed as the first application of QRP to non-linear learning, arising from very practical settings.

2 Backgrounds: Compression for Linear Sketches

We denote the data matrix as $X \in \mathbb{R}^{n\times d}$, where we assume all the samples are normalized to the unit sphere to avoid keeping track of the norms in our analysis. Note that, instance normalization is a common pre-processing step for many learning algorithms. Recall that ρ is the correlation between sample x and y.

Random projection (RP), i.e., the linear sketch, is defined by $X_W = XW$, where $W \in \mathbb{R}^{d\times k}$ is a random matrix with i.i.d. from certain probability distribution (e.g., Rademacher, Gaussian, Cauchy). To derive the RFF for RBF kernel as in Eq. (2), we focus on the Gaussian random projection, i.e., the entries of W are i.i.d. $N(0,\gamma^2)$. X_W is called the full-precision random projection (FP-RP). For two samples $x, y \in \mathbb{S}^{d-1}$, it can be shown that when $\gamma = 1$, we have $\mathbb{E}[(w^T x)(w^T y)] = \rho$ where w is a column of W. In other words, the inner product (or cosine) between data samples is preserved in expectation by random projection.

Quantized RP. The QRP-RFF approach relies on the quantized random projections (QRPs). An m-level fixed quantizer is a map $Q : \mathcal{X} \mapsto \mathcal{C}$ with $\mathcal{X}$ the signal domain and $\mathcal{C}$ the codebook containing the reconstruction levels (or the codes) $\mu_1, ..., \mu_m$. Precisely,

$$Q(x) = \mu_i, \quad \text{if } t_{i-1} < x \le t_i,$$

with $t_0 < t_1 < ... < t_m$ the borders of quantizer Q. We assume $m = 2^b$ where $b \ge 1$ is the number of bit representation. As the projected signal $w^T x \sim N(0,\gamma^2)$ is supported on the real line, we consider quantizers symmetric about 0 and set two ends $t_0 = -\infty$ and $t_m = +\infty$. Next, we introduce the quantizer for QRP that will be discussed in this paper.

The *Lloyd-Max (LM) quantization (Lloyd, 1982)* is an important scheme constructed via purposeful design. For QRP, the LM quantization has been proved favorable for several learning tasks (Li and Slawski, 2017; Li and Li, 2019a). When the underlying signal z comes from a probability distribution $g(z)$, the LM quantizer minimizes the distortion defined as

$$D_Q = \mathbb{E}[(z - Q(z))^2] = \int (z - Q(z))^2 g(z) dz, \qquad (4)$$

which is the expected squared loss between the true signal and the quantized signal. For QRPs, we use Lloyd's algorithm for quantizer construction, which is summarized in Algorithm 1 with g set as $N(0,\gamma^2)$, the marginal distribution of the projected data $w^T x$.

Algorithm 1: Lloyd-Max (LM) quantization

```
Input: Signal distribution g ~ N(0, γ^2), bit b
Output: LM quantizer [t_0, ..., t_{2^b}], [μ_1, ..., μ_{2^b}]
While true
    For i = 1 to 2^b
        Update μ_i by μ_i = ∫_{t_{i-1}}^{t_i} x g(x) dx / ∫_{t_{i-1}}^{t_i} g(x) dx
    End For
    For i = 1 to 2^b − 1
        Update t_i by t_i = (μ_{i-1} + μ_i) / 2
    End For
Until Convergence
```

As introduced in Section 1, to strive for more storage efficiency, we define quantized random projection (QRP) as $X_Q = Q(X_W)$, where Q is a quantizing function defined above. In this paper, we will study the problem of using QRP to construct non-linear random features, which will be introduced in Section 3.

Stochastic Rounding. Before moving forward, we briefly introduce a compression method that will be mainly compared with our QRP-RFF in this paper. Stochastic rounding (StocQ) scheme applies standard probabilistic quantization to FP-RFF after it has been generated. A b-bit StocQ quantizer splits the support of trigonometric functions in RFF (i.e., $[-1, 1]$) into $2^b - 1$ equal bins with size $\triangle = \frac{2}{2^b-1}$, and quantize z to either of its two neighboring borders by

$$P(Q(z) = t^*) = \frac{z - t_*}{\triangle}, \quad P(Q(z) = t_*) = \frac{t^* - z}{\triangle},$$

where $[t_*, t^*]$ is the bin containing z. By this construction, the output $Q(z)$ is unbiased of z. However, the unbiasedness pays a cost of larger variance brought by the sampling process, especially with low bits. With moderate number of bits (e.g., $b = 4$ to 8), StocQ can achieve good learning performance, with reduced storage cost compared to full-precision RFF. Note that, following (Zhang et al., 2019; Li and Li, 2021), we apply StocQ to the RFF formulation $F(x) = \sqrt{2}\cos(w^T x + \tau)$, with $\tau \sim unif(0, 2\pi)$. Practically, we found no significant difference in learning performance between this form and Eq. (2). In this case, each StocQ feature has one element while Eq. (2) contains two items. Hence, throughout this paper, k QRF-RFFs will be compared to $2k$ StocQ-RFFs.

3 QRP-RFF Scheme

As shown in Eq. (2), RFF is built upon RP with one extra step of casting non-linearity. Given the popularity and wide application of QRP, one natural question arises: can we extract RFF from QRP for fast non-linear kernel learning? Recalling Figure 1, since one typically would like to discard the full-precision RPs to spare unnecessary storage once they have been quantized and stored in the database, this is a more practical setting worth studying. We call the proposed method QRP-RFF, as depicted in Figure 2. It consists of three steps:

1. Apply random projection $X_W = XW_\gamma$.
2. Quantize the projected data $X_Q = Q(X_W)$ which will be stored in database. We can discard X_W afterwards and use compressed X_Q for various subsequent linear learning tasks.
3. Extract QRP-RFFs from the quantized X_Q, which can then be fed into linear machines for fast approximate non-linear kernel learning.

In step 2, we require the use of "linear" quantizers that satisfy the following property.

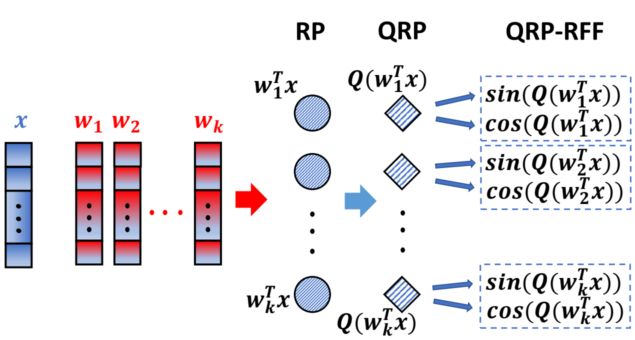

Figure 2: Illustration of QRP-RFF framework. QRP-RFF is constructed directly from QRP. Full-precision RPs can be discarded after QRPs are generated.

Definition 1. *In the context of Gaussian QRP, let Q be the quantizer w.r.t. $N(0, 1)$. The quantization scheme is called linear if for any $\gamma > 0$, γQ is the corresponding quantizer for $N(0, \gamma^2)$.*

In particular, it is easy to check that the LM quantizer falls into this category. The linearity of quantizer allows us to derive QRP for any γ, from the QRP with $\gamma = 1$. This has a crucial impact on the parameter tuning of QRP-RFF. We can simply store one set of compressed linear sketch (e.g., QRPs with $\gamma = 1$) in memory to tune QRP-RFF with any γ by scaling—This is the essential reason that QRP-RFF does not require FP-RP and achieves "one-sketch-for-all". On the contrary, if linearity does not hold, we will have to use the original FP-RP to re-construct quantized sketches for distinct γ, violating our problem setting.

Following step 3, we formally define the QRP-RFF as

$$F_Q(x) = [\sin(Q(w^T x)) \quad \cos(Q(w^T x))]^T, \qquad (5)$$

where $w \sim N(0, \gamma^2)$ and Q is a linear quantizer (Definition 1). Analogously, by k i.i.d. projections, we defined the QRP-RFF kernel estimator by

$$\hat{K}_Q(x, y) = \frac{1}{k}\sum_{i=1}^{k} F_{Q,i}(x)^T F_{Q,i}(y), \qquad (6)$$

where $F_{Q,i}(x)$ is the i-th QRP-RFF of x associated with projection w_i. We re-emphasize the significance of our QRP-RFF scheme: by directly retrieving non-linear random features from QRP, QRP-RFF does not need FP-RP or many sets of quantized RFFs (for different γ values). Instead, only one set of compressed QRP is needed for both linear and non-linear learning.

4 Analysis

In this section, we discuss properties of QRP-RFF kernel estimators and provide the theoretical analysis.

4.1 Equivalence in kernel learning when $b = 1$

In practice, 1-bit compression, e.g. 1-bit random projection, is an important special case of quantization because it achieves highest compression ratio. For linear RP, one can easily point out that all 1-bit quantization methods are equivalent, when the task is to estimate the cosine ρ. Every 1-bit quantizer can be written as $Q(w^T x) = \gamma c_Q \cdot sign(w^T x)$ with some quantizer-specific constant c_Q. That is, different 1-bit quantizers only differ by a constant scaling factor, which can be easily fixed by re-scaling when estimating ρ by the inner product $Q(w^T x)Q(w^T y)$. However, it is obvious that for QRP-RFF, linear scaling of the kernel estimate no longer holds due to the high non-linearity of sine and cosine functions. Nevertheless, we have a weaker statement of equivalence.

Claim 1. *For QRP-RFF, all 1-bit fixed linear quantizers are equivalent in non-linear kernel learning models, provided that γ is tuned properly.*

When $b = 1$, Eq. (5) can be written in the general form $F_Q(x) = [\sin(\gamma c_Q \cdot sign(w^T x))\ \cos(\gamma c_Q \cdot sign(w^T x))]^T$ with some quantizer-specific c_Q. Then, the QRP-RFFs generated by Q_1 with γ_1 can be produced by Q_2 with $\gamma_2 = \frac{c_1}{c_2}\gamma_1$. In words, the difference in c_Q can be eliminated in practice by tuning γ adequately. As a result, in principle, the learning performance of all 1-bit quantizers for QRP-RFF are expected to be the same with fine tuning.

4.2 Information loss of QRP-RFF

From now on, we will denote $K_Q(x, y) = \mathbb{E}[\hat{K}_Q(x, y)]$, or K_Q in short. This is sometimes referred as "expected kernel" in kernel approximation literature. One inevitable issue of extracting RFF directly from fixed quantized random projections, is the information loss in the transaction from linear projections to highly non-linear sine and cosine functions, especially for large γ and small bits b. The reason is that, when γ is large, the difference $|Q(z) - z|$ might be so large that $\sin(Q(z))$ and the FP-RFF $\sin(z)$ (and cosine) are very different, where $z = w^T x \sim N(0, \gamma^2)$ is the RP. When b is small, the deviation is even larger.

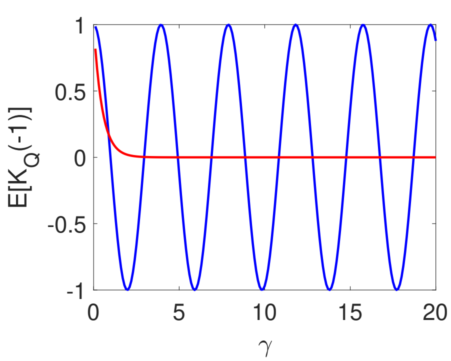

Figure 3: Information loss of QRP-RFF: Mean of 1-bit QRP-RFF estimate from LM quantized QRP, at $\rho = -1$. The red curve is the true kernel, and the blue curve is the 1-bit QRP-RFF mean.

We will use the estimation at a single point as an example. When $b = 1$, we can compute the LM quantizer as $Q(z) = sign(z) \times 0.7979\gamma$. Thus we can explicitly compute K_Q at $\rho = -1$ as $-\sin(0.7979\gamma)^2 + \cos(0.7979\gamma)^2$. This is a periodic function in γ that deviates significantly from the true kernel value at $\rho = -1$, as depicted in Figure 3. We see that the mean (at $\rho = -1$) is only reasonable with $\gamma \le 1$. With larger γ, the estimation becomes wild. Similar instability holds for other ρ. Unfortunately, this unstable behavior is caused by the nature of the problem, i.e., the information loss of coding with discrete Q in the "linear" QRP to "non-linear" RFF transaction. Nevertheless, as will be shown in Section 4.3, when $b \ge 3$, the information loss becomes acceptable as the mean estimation of QRP-RFF approaches the true RBF kernel.

4.3 Mean and variance

Theorem 1. *Let Q be a b-bit fixed quantizer with borders $-\infty = t_0 < t_1 < ... < t_{2^b} = +\infty$ and reconstruction levels $\mu_1 < ... < \mu_{2^b}$. Suppose $u, v \sim N\left(0, \gamma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$, and let $p_{ij} = P\big(u \in [t_{i-1}, t_i], v \in [t_{j-1}, t_j]\big)$ for $1 \le i, j \le 2^b$. Denote $s_i = \sin(\mu_i)$ and $c_i = \cos(\mu_i)$. For normalized data vectors x and y,*

$$K_Q := \mathbb{E}[\hat{K}_Q(x, y)] = \sum_{i=1}^{2^b}\sum_{j=1}^{2^b} (s_i s_j + c_i c_j)\, p_{ij},$$

$$Var[\hat{K}_Q(x, y)] = \frac{1}{k}\left\{\sum_{i=1}^{2^b}\sum_{j=1}^{2^b} (s_i s_j + c_i c_j)^2 p_{ij} - K_Q^2\right\}.$$

With the potentially severe instability of low-bit QRP-RFF estimate in mind, in Figure 4 we plot K_Q with different b and γ, when the QRP is quantized by LM quantizers. We observe the mild estimation when $b = 1, 2$ at some γ value. As expected, as b increases, K_Q converges to the true kernel K.

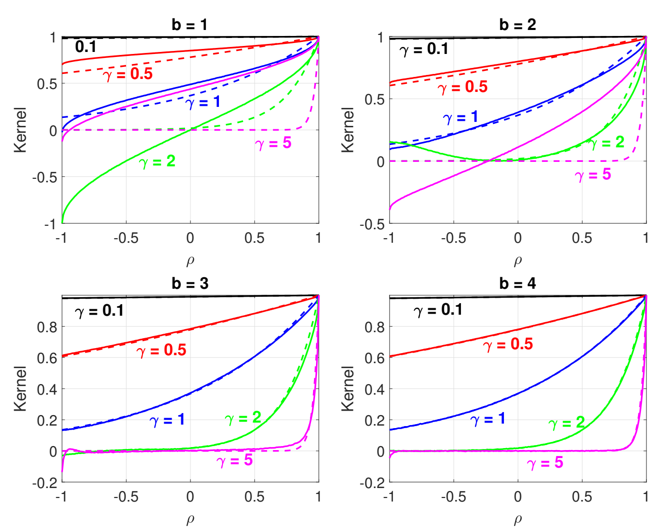

Figure 4: Solid curves: the mean of QRP-RFF estimate (Theorem 1). Dash curves: the true RBF kernel. We see some large deviations when $b = 1, 2$.

Remark 1. *It is important to understand that, K_Q deviating from the exact RBF kernel does not imply bad generalization performance of QRP-RFF. On the one hand, the performance of randomized algorithms also largely relies on the variance (e.g., the large variance of low-bit StocQ results in poor learning capacity, though it is unbiased). On the other hand, in some sense we can regard the QRP-RFF estimators as converging to some other kernel, and comparing the learning capacity of two non-linear kernels is non-trivial and in general data-dependent.*

Due to the information loss, the intrinsic instability of QRP-RFF estimator makes it difficult to obtain elegant theoretical results on the expected kernel K_Q (e.g., recalling Figure 3). Nonetheless, we still provide analytical bounds on K_Q measuring its concentration around the RBF kernel. The following is a general result holding for any quantizer Q.

Theorem 2. *For any fixed γ, let $z \sim N(0, \gamma^2)$, define $D_s = \mathbb{E}[(\sin(Q(z)) - \sin(z))^2]$, $\zeta_s = Cov\big(\sin(Q(z)) - \sin(z), \sin(z)\big)$, and D_c and ζ_c analogously for cosine function. Further denote $\triangle_c = \mathbb{E}[\cos(Q(z))] - e^{-\frac{\gamma^2}{2}}$ and $\tilde{D}_c = D_c - \triangle_c^2$. Denote $V_s^* = \frac{1}{2}\left[1 - e^{-2\rho^2\gamma^2}\right]$ and $V_c^* = \frac{1}{2}\left[1 + e^{-2\rho^2\gamma^2}\right] - e^{-\rho^2\gamma^2}$. Assume x, y are two normalized samples with correlation ρ. Then at γ, $K_Q(x, y)$ is lower and upper bounded respectively by*

$$K(x,y) - D_s - D_c + 2e^{-\frac{\gamma^2(1-\rho^2)}{2}}\big(C_{1-} + C_{2-}\big),$$
$$K(x,y) + D_s + D_c + 2e^{-\frac{\gamma^2(1-\rho^2)}{2}}\big(C_{1+} + C_{2+}\big),$$

where $C_{1\pm} = (C_1C_2 \pm \sqrt{(1-C_1^2)(1-C_2^2)})\sqrt{D_sV_s^*}$, $C_{2\pm} = \big[(C_3C_4 \pm \sqrt{(1-C_3^2)(1-C_4^2)})\sqrt{D_cV_c^*} + e^{-\frac{\gamma^2}{2}}\triangle_c\big]$, with

$$C_1 = \sqrt{\frac{2\zeta_s}{D_s(1-e^{-2\gamma^2})}}, \quad C_2 = \frac{e^{-\frac{\gamma^2(1-\rho)^2}{2}} - e^{-\frac{\gamma^2(1+\rho)^2}{2}}}{\sqrt{2(1-e^{-2\gamma^2})V_s^*}},$$

$$C_3 = \sqrt{\frac{\zeta_c}{\tilde{D}_c(\frac{1}{2}\left[1+e^{-2\gamma^2}\right] - e^{-\gamma^2})}},$$

$$C_4 = \frac{\frac{1}{2}\left[e^{-\frac{\gamma^2(1-\rho)^2}{2}} + e^{-\frac{\gamma^2(1+\rho)^2}{2}}\right] - e^{-\frac{\gamma^2(1+\rho^2)}{2}}}{\sqrt{(\frac{1}{2}\left[1+e^{-2\gamma^2}\right] - e^{-\gamma^2})V_c^*}}.$$

Theorem 2 gives a universal bound on the QRP-RFF mean for any Q at any γ and ρ, which states that K_Q "concentrates" around K with error no more than $\mathcal{O}(D_s + D_c + \sqrt{D_s} + \sqrt{D_c})$. Thus, smaller non-linear distortions lead to stronger concentration, as smaller D_s and D_c imply better approximation of QRP-RFF to RFF. As $b \to \infty$, the distortions go to 0 and K_Q converges to K.

Variance. Another important factor that affects the learning performance with randomized algorithms is the variance of the estimation. In Figure 5, we plot variances of full-precision RFF, QRP-RFF with LM quantization and StocQ estimators at representative γ levels. We see that at low bit constraint $b = 1, 2$, the stochastic StocQ has much larger variance than QRP-RFF estimators. Consequently, StocQ may perform worse than QRP-RFF in low-bit training. We omit the figures for more bits, since the variance of QRF-RFF converges to that of FP-RFF as expected.

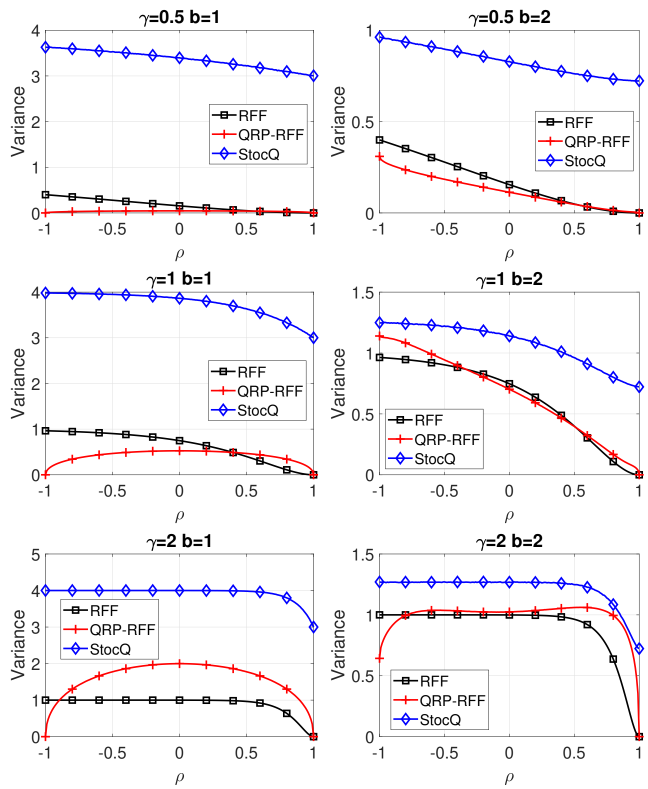

Figure 5: Variance of a random feature of FP-RFF, StocQ and QRP-RFF (Theorem 1, to be scaled by k). The variance of StocQ follows from Li and Li (2021).

4.4 Approximation error

In practice, it is favorable to produce and store as few RFFs as possible to achieve small approximation error to the true RBF kernel. Similarly, we are also interested in the sufficient number of QRP-RFFs to approximate K_Q within some pre-defined error. In this context, it is important to understand the sample complexity of QRP-RFF, measured by the uniform approximation error $\sup_{x,y\in\mathcal{X}} |\hat{K}_Q(x,y) - K_Q(x,y)|$. For full-precision RFF (Rahimi and Recht, 2007; Sutherland and Schneider, 2015), k is required to be at least $\mathcal{O}(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$ to guarantee ϵ-approximation w.h.p.. To proceed, we first introduce the following definition.

Definition 2. ***(Mean Smooth Quantizer)*** *We say a quantizer $Q(\cdot)$ is mean Lipschitz smooth w.r.t. distribution Γ and function f with constant L_Q^f, if for $\forall\delta > 0$, the following holds,*

$$\mathbb{E}_{t\sim\Gamma}\left[\sup_{|r|\le\delta}|f(Q(t+r)) - f(Q(t))|\right] \le L_Q^f\delta. \tag{7}$$

Basically, quantizer Q is mean smooth if the average maximal deviation of a function f applied to the quantized random measurements from Γ is bounded in a Lipschitz way. This is a an "averaged" version of Lipschitz continuity, which also works for discrete functions. Definition 2 is a generalisation of (Schellekens and Jacques, 2020) which was restricted to periodic functions. In our problem where f is sine or cosine, $f \circ Q$, when composited as one function, is no longer periodic. By extending the characterization to a more general setting, the uniform approximation error of QRP-RFF is given as below (with general quantizers).

Theorem 3. ***(Uniform Approximation Error)*** *Assume the sample space $\mathcal{S}$ is the unit sphere (normalized data). Let QRP-RFF estimators be defined as (6). Let $\Gamma \sim N(0,\gamma^2)$ in Definition 2. Suppose a quantizer Q is mean smooth w.r.t.* sin *and* cos *functions with Lipschitz constant L_Q^s and L_Q^c, respectively. Then for $\forall\ \epsilon > 0$, with probability at least $1 - 4e^{-k\epsilon^2/256}$,*

$$|\hat{K}_Q(x,y) - K_Q(x,y)| \le \epsilon, \qquad \textit{for } \forall x,y \in \mathcal{S},$$

when $k \ge \frac{512d}{\epsilon^2}\log(\frac{64\max\{L_Q^s,L_Q^c\}\gamma}{\epsilon}+1)$.

Theorem 3 says that to achieve ϵ-error, the sample complexity of QRP-RFF is the same as that of full-precision RFF, within constant factor. We now show that for our problem where f is the sin or cos function and $\Gamma \sim N(0,\gamma^2)$, every bounded quantizer with finite bits is mean smooth. Hence, the error bound in Theorem 3 holds for LM quantizer.

Proposition 1. *When f is* sin *or* cos *function and $\Gamma \sim N(0,\gamma^2)$ in Definition 2, every finite-bit bounded quantizer is mean Lipschitz smooth with $L_Q = \frac{4(2^b-1)}{\gamma\sqrt{2\pi}}$.*

Proof. We present the analysis of sine function. Assume the quantizer has b bits. Thus it contains 2^b-1 finite borders, denoted as $t_1^*,...,t_{2^b-1}^*$. For a fixed point t, the value $\sup_{|r|\le\delta}|\sin(Q(t+r)) - \sin(Q(t))|$ equals to 0 if $t_i^*+\delta \le t \le t_{i+1}^*-\delta$ for some i. Otherwise, $\sup_{|r|\le\delta}|\sin(Q(t+r)) - \sin(Q(t))|$ would be bounded by 2. Therefore, integrating over the domain of Γ gives

$$\begin{aligned}&\mathbb{E}_{t\sim N(0,\gamma^2)}\left[\sup_{|r|\le\delta}\left|f(Q(t+r)) - f(Q(t))\right|\right]\\ \le& 2P\left[t \in \cup_{i=1}^{2^b-1}[t_i^*-\delta, t_i^*+\delta]\right]\\ \le&\frac{4(2^b-1)}{\gamma\sqrt{2\pi}}\delta.\end{aligned}$$

The last line is due to the fact that for $t \sim N(0,\gamma^2)$, the property of normal density implies $P[t^*-\delta \le t \le t^*+\delta] \le 2\delta\cdot\frac{1}{\gamma\sqrt{2\pi}}$ for any t^*. Therefore, the mean smoothness constant L_Q^s is at most $\frac{4(2^b-1)}{\gamma\sqrt{2\pi}}$. Similar proof holds for cosine function. ∎

5 Experiments

In this section, we test the learning performance of QRP-RFF scheme in kernel classification problems. The main purpose is to show that (i) QRP-RFF performs better than StocQ with low bits; and (ii) when b is as large as 4, the performance of QRP-RFF is similar to the full-precision RFF.

Setting. We compare three randomized approximations: 1) the full-precision RFF; 2) QRP-RFF with underlying LM quantization; and 3) stochastic quantization (StocQ)[2]. For approaches involving quantization, after the FP-RFFs are generated, we process them with corresponding quantization strategy, then feed them into a linear SVM solver. We tune the parameters C for SVM and γ for RBF kernel over a wide range of values. We use public datasets from UCI machine learning repository (Dua and Graff, 2017). For all datasets, the samples are normalized to have unit norm. On each dataset, we randomly split the samples into 60% for training and 40% for testing. For each method, the best test accuracy among C and γ are reported, averaged over 10 independent repetitions.

Results. In Figure 6, we report the classification test accuracy against the number of RFFs used, with $b = 1, 2, 4$. We observe the following:

- **Low-bit training**. For $b = 1$, on all datasets, we observe significant higher accuracy of QRP-RFF over StocQ. On ISOLET, QRP-RFF with $b = 1$ almost achieves the same accuracy as full-precision RFF. For $b = 2$, QRP-RFF also compares favorably with StocQ, when the number of random sketches is moderate (i.e., more than 2^{10}). The poor performance of StocQ can be partially explained by its large variance with low bits (see Figure 5 and Li and Li (2021)).

[2] We implemented StocQ to both aforementioned formulations of RFFs, and found very similar performance.

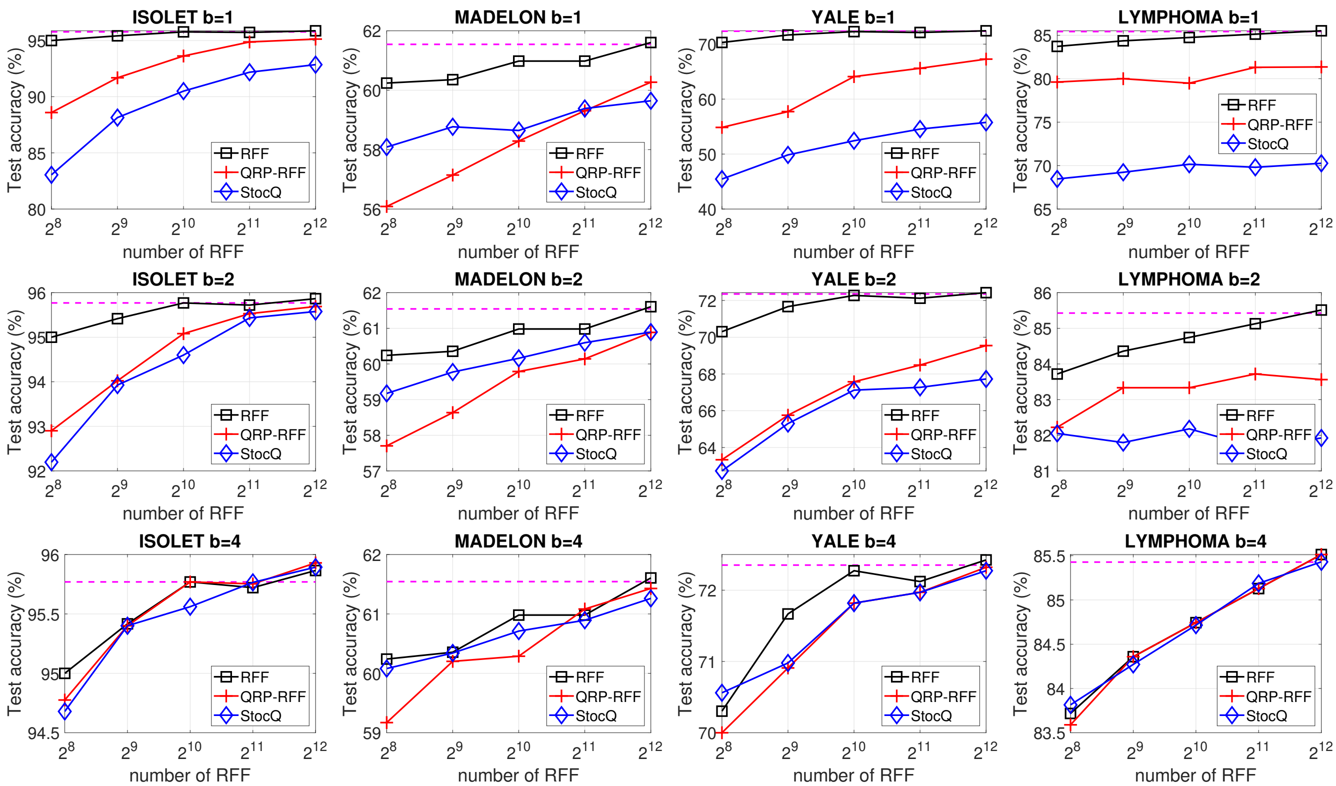

Figure 6: Test accuracy of (linearized) kernel SVM using RFF with different quantization strategies. The dash line is for standard RBF kernel SVM.

- **More bits**. As we use more bits, the test accuracy of both quantization methods gets improved. For QRP-RFF, on all datasets, $b = 4$ is sufficient to approach the performance of FP-RFF, with moderate number of random features.

Memory saving. The benefit of QRP-RFF in terms of storage saving becomes obvious given Figure 6. Since 4-bit QRP-RFF almost has same test accuracy as using FP-RFF, the storage can typically be reduced by at least $32/4 = 8$x or 16x, when FP-RFFs are represented by 32 bits or 64 bits, respectively.

6 Discussions and Conclusions

In this paper, we consider the problem of constructing random Fourier features (RFF) from quantized random projections (QRP). Our proposed QRP-RFF scheme is "one-sketch-for-all" in the sense that we only need to store one set of compressed linear sketches for both linear and non-linear learning (and for any γ parameter for the Gaussian kernel), which is convenient in practical scenarios where one would commonly discard full-precision RPs after deriving the QRPs from the original data and RPs. We provide general bounds on the mean and uniform approximation errors of the proposed kernel estimator and compare Lloyd-Max quantization with a stochastic rounding method. In the experiments, QRP-RFF outperforms stochastic rounding, in terms of the kernel SVM accuracy in the low-bit training scenario, which is important in practice, and approximates the performance of full-precision RFF with 4-bit quantization. Compared with Li and Li (2021), which directly optimized the quantized outputs on top of the RFFs, the proposed method would unavoidably lose certain accuracy. Nevertheless, QRP-RFF provides a feasible alternative in certain application scenarios in which practitioners could not afford to store multiple sets of RFFs for different tuning parameters (γ) of the Gaussian kernel. Finally, we should mention one additional price which the proposed scheme has to pay, that is, we will need to compute sine and cosine functions on the fly. The computations can be to an extent avoided by tabulations (i.e., one look-up table for each γ value).

In conclusion, for applications where we know the best tuning parameter γ, we should use the quantization scheme in Li and Li (2021). If the best γ is unknown (e.g., in an early stage of exploration), our proposed method provides a feasible alternative.

Acknowledgement

The authors sincerely thank the anonymous reviewers and area chairs of AISTATS 2021, for their constructive and encouraging comments.

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.

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.

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.

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

Petros Boufounos and Richard G. Baraniuk. 1-bit compressive sensing. In *42nd Annual Conference on Information Sciences and Systems (CISS)*, pages 16–21, Princeton, NJ, 2008.

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.

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.

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.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

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.

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.

Anna C. Gilbert, Martin J. Strauss, Joel A. Tropp, and Roman Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 237–246, San Diego, CA, 2007.

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

Sivakant Gopi, Praneeth Netrapalli, Prateek Jain, and Aditya V. Nori. One-bit compressed sensing: Provable support and vector recovery. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 154–162, Atlanta, GA3, 2013.

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.

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

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. 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, 2017.

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, Kenneth Ward Church, and Trevor Hastie. One sketch for all: Theory and application of conditional random sampling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 953–960, Vancouver, Canada, 2008.

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)*, pages 15124–15134, Vancouver, Canada, 2019a.

Xiaoyun Li and Ping Li. Random projections with asymmetric quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10857–10866, Vancouver, Canada, 2019b.

Xiaoyun Li and Ping Li. Quantization algortihms for random fourier features. Technical report, arXiv:2102.13079, 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.

Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Trans. Inf. Theory*, 59(1):482–494, 2013.

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.

Vincent Schellekens and Laurent Jacques. Breaking the waves: asymmetric random periodic features for low-bitrate kernel machines. *arXiv preprint arXiv:2004.06560*, 2020.

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.

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.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

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.