Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels

Jiyan Yang[1] JIYAN@STANFORD.EDU
ICME, Stanford University, Stanford, CA 94305.

Vikas Sindhwani[1] VSINDHW@US.IBM.COM
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

Haim Avron[1] HAIMAV@US.IBM.COM
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

Michael W. Mahoney MMAHONEY@ICSI.BERKELEY.EDU
International Computer Science Institute and Dept. of Statistics, University of California at Berkeley, Berkeley, CA 94720

Abstract

We consider the problem of improving the efficiency of randomized Fourier feature maps to accelerate training and testing speed of kernel methods on large datasets. These approximate feature maps arise as Monte Carlo approximations to integral representations of shift-invariant kernel functions (e.g., Gaussian kernel). In this paper, we propose to use *Quasi-Monte Carlo* (QMC) approximations instead where the relevant integrands are evaluated on a low-discrepancy sequence of points as opposed to random point sets as in the Monte Carlo approach. We derive a new discrepancy measure called *box discrepancy* based on theoretical characterizations of the integration error with respect to a given sequence. We then propose to learn QMC sequences adapted to our setting based on explicit box discrepancy minimization. Our theoretical analyses are complemented with empirical results that demonstrate the effectiveness of classical and adaptive QMC techniques for this problem.

1. Introduction

Kernel methods (Schölkopf & Smola, 2002; Wahba, 1990; Cucker & Smale, 2001) offer a comprehensive collection of non-parametric modeling techniques for a wide range of problems in machine learning. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ denote a kernel function defined on an input domain $\mathcal{X} \subset \mathbb{R}^d$. The kernel k may be (non-uniquely) associated with an embedding of the input space into a high-dimensional Hilbert space $\mathcal{H}$ (with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$) via a feature map, $\Psi : \mathcal{X} \mapsto \mathcal{H}$, such that $k(\mathbf{x}, \mathbf{z}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{z}) \rangle_{\mathcal{H}}$. Standard regularized linear statistical models in $\mathcal{H}$ then provide non-linear inference with respect to the original input representation. The algorithmic basis of such constructions are classical Representer Theorems (Wahba, 1990; Schölkopf & Smola, 2002) that guarantee finite-dimensional solutions of associated optimization problems, even if $\mathcal{H}$ is infinite-dimensional.

However, there is a steep price of these elegant generalizations in terms of scalability. Consider, for example, least squares regression given n data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and assume that $n \gg d$. The complexity of linear regression training using standard least squares solvers is $O(nd^2)$, with $O(nd)$ memory requirements, and $O(d)$ prediction speed on a test point. Its kernel-based nonlinear counterpart, however, requires solving a linear system involving the Gram matrix of the kernel function (defined by $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$). In general, this incurs $O(n^3 + n^2 d)$ complexity for training, $O(n^2)$ memory requirements, and $O(nd)$ prediction time for a single test point – none of which are particularly appealing in "Big Data" settings.

In this paper, we revisit the randomized construction of a family of low-dimensional approximate feature maps proposed by Rahimi & Recht (2007) for scaling up kernel methods. These randomized feature maps, $\hat{\Psi} : \mathcal{X} \mapsto \mathbb{C}^s$, provide low-distortion approximations for (complex-valued) kernel functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$:

$$k(\mathbf{x}, \mathbf{z}) \approx \langle \hat{\Psi}(\mathbf{x}), \hat{\Psi}(\mathbf{z}) \rangle_{\mathbb{C}^s} \qquad (1)$$

where $\mathbb{C}^s$ denotes the space of s-dimensional complex numbers with the inner product, $\langle \alpha, \beta \rangle_{\mathbb{C}^s} = \sum_{i=1}^s \alpha_i \beta_i^*$,

[1] Equal contributors

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

with z^* denoting the conjugate of the complex number z. Though Rahimi & Recht (2007) define real-valued feature maps as well, our technical exposition is simplified by adopting the generality of complex-valued features. The approximation in (1) leads to scalable solutions, e.g., for regression we get back to $O(ns^2)$ training and $O(s + \mathrm{maptime})$ prediction speed where $\mathrm{maptime}$ is the time to generate random features for a test input, with $O(ns)$ memory requirements. In particular, the approximation in (1) is valid for an important class of kernel functions which are *shift-invariant*. A kernel function k on $\mathbb{R}^d$ is called shift-invariant if $k(\mathbf{x}, \mathbf{z}) = g(\mathbf{x} - \mathbf{z})$, for some complex-valued *positive definite function* g on $\mathbb{R}^d$. Positive definite functions are those that satisfy the property that given any set of m points, $\mathbf{x}_1 \ldots \mathbf{x}_m \in \mathbb{R}^d$, the $m \times m$ matrix $\mathbf{A}$ defined by $\mathbf{A}_{ij} = g(\mathbf{x}_i - \mathbf{x}_j)$ is positive semi-definite.

The starting point of Rahimi & Recht (2007) is a celebrated result that characterizes the class of positive definite functions:

Theorem 1 (Bochner (1933)). *A complex-valued function $g : \mathbb{R}^d \mapsto \mathbb{C}$ is positive definite if and only if it is the Fourier Transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$, i.e.,*

$$g(\mathbf{x}) = \hat{\mu}(\mathbf{x}) = \int_{\mathbb{R}^d} e^{-i\mathbf{x}^T\mathbf{w}} d\mu(\mathbf{w}), \quad \forall \mathbf{x} \in \mathbb{R}^d .$$

Without loss of generality, we assume henceforth that $\mu(\cdot)$ is a probability measure with associated probability density function $p(\cdot)$. The above result implies that a scaled shift-invariant kernel can therefore be put into one-to-one correspondence with a density p such that,

$$k(\mathbf{x}, \mathbf{z}) = g(\mathbf{x} - \mathbf{z}) = \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{z})^T\mathbf{w}} p(\mathbf{w}) d\mathbf{w} . \tag{2}$$

For the most notable member of the shift-invariant family of kernels – the Gaussian kernel: $k(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|_2^2}{2\sigma^2}}$, the associated density is again Gaussian, $\mathcal{N}(0, \sigma^{-2}\mathbf{I}_d)$.

The integral representation of the kernel (2) may be approximated as follows:

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{z})^T\mathbf{w}} p(\mathbf{w}) d\mathbf{w} & (3) \\ &\approx \frac{1}{s} \sum_{j=1}^{s} e^{-i(\mathbf{x}-\mathbf{z})^T\mathbf{w}_s} & (4) \\ &= \langle \hat{\Psi}_S(\mathbf{x}), \hat{\Psi}_S(\mathbf{z}) \rangle_{\mathbb{C}^s} , & (5) \end{aligned}$$

through the feature map,

$$\hat{\Psi}_S(\mathbf{x}) = \frac{1}{\sqrt{s}} \left[e^{-i\mathbf{x}^T\mathbf{w}_1} \ldots e^{-i\mathbf{x}^T\mathbf{w}_s} \right] \in \mathbb{C}^s . \tag{6}$$

The subscript S denotes dependence of the feature map on the sequence $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_s\}$. When elements of the sequence are drawn from the distribution defined by the density function $p(\cdot)$, the approximation in (4) may be viewed as a standard *Monte Carlo* (MC) approximation to the integral representation of the kernel. This simple observation is our point of departure from the work of Rahimi & Recht (2007).

We are now in a position to state the contributions of this paper:

- ○ We propose to use the low-discrepancy properties of *Quasi-Monte Carlo* (QMC) sequences to reduce the integration error in approximations of the form (4). A self-contained overview of Quasi-Monte Carlo techniques for high-dimensional integration problems is provided in Section 2. In Section 3, we describe how QMC techniques apply to our setting.
- ○ We provide an average case theoretical analysis of the integration error for any given sequence S (Section 4).
- ○ This bound motivates an optimization problem over the sequence S whose minimizer provides *adaptive QMC* sequences fine tuned to our kernels (Section 5).
- ○ Empirical results (Section 6) clearly demonstrate the superiority of QMC techniques over the MC feature maps (Rahimi & Recht, 2007), the correctness of our theoretical analysis and the potential value of adaptive QMC techniques for large-scale kernel methods.

2. Quasi-Monte Carlo Techniques: Overview

In this section we provide a self-contained overview of Quasi-Monte Carlo (QMC). Due to space limitation we restrict our discussion to background that is necessary for understanding subsequent sections. We refer the interested reader to the excellent reviews by Caflisch (1998) and Dick et al. (2013) for more detailed exposition.

Consider the task of computing an approximation of the following integral,

$$I_d[f] = \int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x} .$$

One can observe that if X_d is a random variable uniformly distributed over $[0,1]^d$ then $I_d[f] = \mathbb{E}[f(X_d)]$. An empirical approximation to the expected value can be computed by drawing a random point set $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_s\}$ independently from $[0,1]^d$, and computing:

$$I_S[f] = \frac{1}{s} \sum_{\mathbf{w} \in S} f(\mathbf{w}) .$$

This is the Monte Carlo (MC) method.

Define the integration error with respect to the point set S as

$$\epsilon_S[f] = |I_d(f) - I_S(f)|. \tag{7}$$

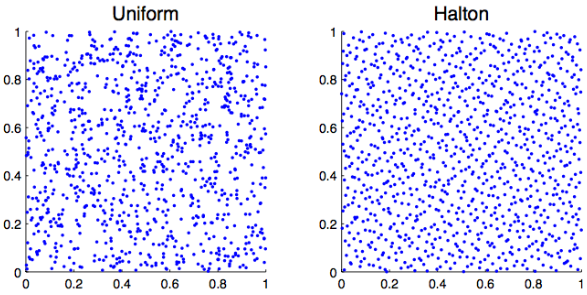

Figure 1. Comparison of MC and QMC sequences.

When S is drawn randomly, the Central Limit Theorem asserts that if $s = |S|$ is large enough then $\epsilon_S[f] \approx \sigma[f]s^{-1/2}\boldsymbol{\nu}$ where $\boldsymbol{\nu}$ is a standard normal random variable, and $\sigma[f]$ is the square-root of the variance of f; that is,

$$\sigma^2[f] = \int_{[0,1]^d} (f(\mathbf{x}) - I_d(f))^2 \, d\mathbf{x} \, . \tag{8}$$

In other words, the root mean square error of the Monte Carlo method is,

$$\left(\mathbb{E}_S \left[\epsilon_S[f]^2\right]\right)^{1/2} = \sigma[f]s^{-1/2}. \tag{9}$$

Therefore, the Monte Carlo method converges at a rate of $O(s^{-1/2})$.

The aim of QMC methods is to improve the convergence rate by using a deterministic *low-discrepancy sequence* to construct S, instead of randomly sampling points. The underlying intuition is illustrated in Figure 1, where we plot a set of 1000 two-dimensional random points (left graph), and a set of 1000 two-dimensional points from a quasi-random sequence (Halton sequence; right graph). In the random sequence we see that there is an undesired clustering of points, and as a consequence empty spaces. Clusters add little to the approximation the integral in those regions, while in the empty spaces the integrand is not sampled. This lack of uniformity is due to the fact that Monte Carlo samples are independent of each other. By carefully designing a sequence of correlated points to avoid such clustering effects, QMC attempts to avoid this phenomena, and thus provide faster convergence to the integral. A typical QMC sequence has a hierarchical structure: the initial points sample the integrand on a coarse scale while the latter points sample it more finely.

Informally, the integration error with respect to a sequence depends on a measure of variation of the integrand f over the integration domain, and a sequence-dependent term that typically measures the *discrepancy*, or degree of departure from uniformity, of the sequence S. For example, the expected Monte Carlo integration error decouples into a variance term, and $1/\sqrt{s}$ as in (9). A remarkable and classical result in QMC theory formalizes this intuition as follows.

Theorem 2 (Koksma-Hlawka inequality)**.** *For any function f with bounded variation, and sequence $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_s\}$, the integration error is bounded above as follows,*

$$\epsilon_S[f] \leq D^\star(S)V_{HK}[f] \, . \tag{10}$$

where V_{HK} is the variation of f in the sense of Hardy and Krause *(see Niederreiter (1992)) defined in terms of the following partial derivatives,*

$$V_{HK}[f] = \sum_{I \subset [d], I \neq \emptyset} \int_{[0,1]^{|I|}} \left| \frac{\partial f}{\partial \mathbf{u}_I} \bigg|_{u_j = 1, j \notin I} \right| d\mathbf{u}_I \ , \tag{11}$$

and $D^\star$ is the star discrepancy *defined by*

$$D^\star(S) = \sup_{\mathbf{x} \in [0,1]^d} |\mathrm{disr}_S(\mathbf{x})| \, , \tag{12}$$

where disr_S *is the* local discrepancy function

$$\mathrm{disr}_S(\mathbf{x}) = \mathrm{Vol}(J_\mathbf{x}) - \frac{|\{i \, : \, \mathbf{w}_i \in J_\mathbf{x}\}|}{s}$$

with $J_\mathbf{x} = [0, x_1) \times \cdots \times [0, x_d)$ *with* $\mathrm{Vol}(J_\mathbf{x}) = \prod_{j=1}^d x_j$.

An infinite sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots$ is defined to be a *low-discrepancy sequence* if, as a function of s, $D^\star(\{\mathbf{w}_1, \ldots, \mathbf{w}_s\}) = O((\log s)^d/s)$. It is conjectured that this decay rate of discrepancy is in fact optimal. It is outside the scope of this paper to describe these different constructions in detail. However we mention that notable members are *Halton sequences*, *Sobol' sequences*, *Faure sequences*, *Niederreiter sequences*, and more (see Dick et al. (2013), Section 2).

The classical QMC theory, which is based on the Koksma-Hlawka inequality and low discrepancy sequences, thus achieves a convergence rate of $O((\log s)^d/s)$. While this is asymptotically superior to $O(1/\sqrt{s})$ for a fixed d, it requires s to be exponential in d for the improvement to manifest. As such, in the past QMC methods were dismissed as unsuitable for very high-dimensional integration.

However, several authors noticed that QMC methods perform better than MC even for very high-dimensional integration (Sloan & Wozniakowski, 1998; Dick et al., 2013). Contemporary QMC literature explains and expands on these empirical observations, by leveraging the structure of the space in which the integrand function lives, to derive more refined bounds and discrepancy measures, even when classical measures of variation such as (11) are unbounded. This literature has evolved along at least two directions: one, where worst-case analysis is provided under the assumption that the integrands live in a Reproducing Kernel Hilbert Space (RKHS) of sufficiently smooth and well-behaved functions (see Dick et al. (2013), Section 3) and second, where the analysis is done in terms of *average-case* error, under an assumed probability distribution over the integrands, instead of worst-case error (Wozniakowski, 1991; Traub & Wozniakowski, 1994). We refrain from more details, as these are essentially the paths that the analysis in Section 4 follows for our specific setting.

3. QMC Feature Maps: Our Algorithm

We assume that the density function in (2) can be written as $p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j)$, where $p_j(\cdot)$ is a univariate density function. The density functions associated to many shift-invariant kernels, e.g. Gaussian, Laplacian and Cauchy, admit such a form.

The QMC method is generally applicable to integrals over a unit cube. So typically integrals of the form (2) are handled by first generating a discrepancy sequence $\mathbf{t}_1, \ldots, \mathbf{t}_s \in [0,1]^d$, and transformed it into a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_s$ in $\mathbb{R}^d$, instead of drawing the elements of the sequence from $p(\cdot)$ as in the MC method.

To convert (2) to an integral over the unit cube, a simple change of variables suffices. For $\mathbf{t} \in \mathbb{R}^d$, define

$$\Phi^{-1}(\mathbf{t}) = \left(\Phi_1^{-1}(t_1), \ldots, \Phi_d^{-1}(t_d)\right) \in \mathbb{R}^d, \qquad (13)$$

where Φ_j is the cumulative distribution function (CDF) of p_j, for $j = 1, \ldots, d$. By setting $\mathbf{w} = \Phi^{-1}(\mathbf{t})$, (2) can be equivalently written as

$$\int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{z})^T \mathbf{w}} p(\mathbf{w}) d\mathbf{w} = \int_{[0,1]^d} e^{-i(\mathbf{x}-\mathbf{z})^T \Phi^{-1}(\mathbf{t})} d\mathbf{t} \, . \qquad (14)$$

Thus, a low discrepancy sequence $\mathbf{t}_1, \ldots, \mathbf{t}_s \in [0,1]^d$ can be transformed using $\mathbf{w}_i = \Phi^{-1}(\mathbf{t}_i)$, which is then plugged into (6) to yield the QMC feature map. This simple procedure is summarized in Algorithm 1.

The main question is, of course, *which sequence to use?* One natural choice is the classical low-discrepancy sequences (e.g. Halton, Sobol'). Implementations of these sequences are provided by several scientific libraries (e.g. MATLAB and the GNU Scientific Library), so using these sequences is rather effortless. In Section 6 we give empirical evidence that these sequences produce better approximations to the kernel as compared to the MC approach of Rahimi & Recht (2007).

However, classical analysis (e.g., using Koksma-Hlawka inequality) is inapplicable as the variation of the integrand $e^{-i(\mathbf{x}-\mathbf{z})^T \Phi^{-1}(\mathbf{t})}$ is not bounded. Therefore, in the next section we develop a new discrepancy measure, which we call *box discrepancy*, that is specifically tuned for the problem at hand. We show that the square of the box discrepancy is equal to the expected integration error squared when $\mathbf{x} - \mathbf{z}$ is distributed uniformly. We give numerical evidence that several popular low-(star)-discrepancy sequences tend to have smaller box discrepancy values than random (MC) sequences, explaining why QMC feature maps are more effective. We also propose an adaptive QMC scheme that is based on minimizing the proposed box discrepancy measure.

Algorithm 1 Quasi-Random Fourier Features

```
Input: Shift-invariant kernel $k$, size $s$.
Output: Feature map $\hat{\Psi}(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{C}^s$.

Find $p$, the inverse Fourier transform of $k$.
Generate a low discrepancy sequence $\mathbf{t}_1, \ldots, \mathbf{t}_s$.
Transform the sequence: $\mathbf{w}_i = \Phi^{-1}(\mathbf{t}_i)$ by (13).
Set $\hat{\Psi}(\mathbf{x}) = \sqrt{\frac{1}{s}} \left[ e^{-i\mathbf{x}^T \mathbf{w}_1}, \ldots, e^{-i\mathbf{x}^T \mathbf{w}_s} \right]$.
```

4. Theoretical Analysis and Average Case Error Bounds

The goal of this section is to develop a framework for analyzing the approximation quality of QMC feature maps when used in (3)-(6). In particular, we derive a new discrepancy measure, *box discrepancy*, that characterizes integration error for the set of integrals defined with respect to the underlying data domain. Proofs for all the assertions can be found in supplementary material. Throughout this section we use the convention that $S = \{\mathbf{w}_1, \ldots, \mathbf{w}_s\}$. We also use the notation $\bar{\mathcal{X}} = \{\mathbf{x} - \mathbf{z} \mid \mathbf{x}, \mathbf{z} \in \mathcal{X}\}$.

We start with the observation that the classical Koksma-Hlawka inequality, cannot be immediately applied to the most important cases in for our setting.

Proposition 3. *For any $p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j)$, where $p_j(\cdot)$ is a univariate density function, define $\Phi^{-1}(\mathbf{t})$ by (13). For a fixed $\mathbf{u} \in \mathbb{R}^d$, variation $V_{HK}[\cdot]$ (11) is unbounded for $f_{\mathbf{u}}(\mathbf{t}) = e^{-i\mathbf{u}^T \Phi^{-1}(\mathbf{t})}$, $\mathbf{t} \in [0,1]^d$.*

Given a probability density function $p(\cdot)$ and S, we define the integration error $\epsilon_{S,p}[f]$ of a function f with respect to p and the s samples as,

$$\epsilon_{S,p}[f] = \left| \int_{\mathbb{R}^d} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} - \frac{1}{s} \sum_{i=1}^{s} f(\mathbf{w}_i) \right| \, . \qquad (15)$$

Next, we note that if integrands belong to a Reproducing Kernel Hilbert Space (RKHS), a worst-case integration error bound can be shown as below; see Cucker & Smale (2001) for the definition of RKHS.

Proposition 4 (Integration Error in an RKHS)**.** *Let $\mathcal{H}$ be a RKHS with kernel $h(\cdot,\cdot)$. Assume that $\kappa = \sup_{\mathbf{x} \in \mathbb{R}^d} h(\mathbf{x},\mathbf{x}) < \infty$. Then, for all $f \in \mathcal{H}$ we have,*

$$\epsilon_{S,p}[f] \leq \|f\|_{\mathcal{H}} D_{h,p}(S) \, , \qquad (16)$$

where

$$D_{h,p}(S)^2 = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} h(\omega,\phi)p(\omega)p(\phi)d\omega d\phi -$$
$$\frac{2}{s}\sum_{l=1}^{s}\int_{\mathbb{R}^d} h(\mathbf{w}_l,\omega)p(\omega)d\omega + \frac{1}{s^2}\sum_{l=1}^{s}\sum_{j=1}^{s} h(\mathbf{w}_l,\mathbf{w}_j)\ . \quad (17)$$

For a vector $\mathbf{b} \in \mathbb{R}^d$, let us define $\square\mathbf{b} = \{\mathbf{u} \in \mathbb{R}^d \mid |u_j| \leq b_j\}$. If $b_j = \sup_{\mathbf{u}\in\bar{\mathcal{X}}} |u_j|$ then $\bar{\mathcal{X}} \subset \square\mathbf{b}$, so the set of integrands

$$\mathcal{F}_{\square\mathbf{b}} = \left\{ f_{\mathbf{u}}(\mathbf{x}) = e^{-i\mathbf{u}^T\mathbf{x}},\ \mathbf{u} \in \square\mathbf{b} \right\} \quad (18)$$

is broader than the set of integrands we wish to approximate.

Now, consider the space of functions that admit an integral representation over $\mathcal{F}_{\square\mathbf{b}}$ of the form,

$$f(\mathbf{x}) = \int_{\mathbf{u}\in\square\mathbf{b}} \hat{f}(\mathbf{u})e^{-i\mathbf{u}^T\mathbf{x}}d\mathbf{u} \text{ where } \hat{f}(\mathbf{u}) \in L_2(\square\mathbf{b})\ . \quad (19)$$

This space is associated with bandlimited functions, i.e., functions with compactly-supported inverse Fourier transforms, which are of fundamental importance in the Shannon-Nyquist sampling theory. Under a natural choice of inner product, these spaces are called *Paley-Wiener spaces* and they constitute an RKHS.

Proposition 5 (Kernel of Paley-Wiener RKHS (Berlinet & Thomas-Agnan, 2004; Yao, 1967; Peloso, 2011))**.** *By $PW_{\mathbf{b}}$, denote the space of functions which admit the representation in* (19)*, with the inner product $\langle f, g\rangle_{PW_{\mathbf{b}}} = (2\pi)^{2d}\langle \hat{f}, \hat{g}\rangle_{L_2(\square\mathbf{b})}$. $PW_{\mathbf{b}}$ is an RKHS with kernel function,*

$$\mathrm{sinc}_{\mathbf{b}}(\mathbf{u},\mathbf{v}) = \pi^{-d}\prod_{j=1}^{d}\frac{\sin\left(b_j(u_j - v_j)\right)}{u_j - v_j}\ . \quad (20)$$

For notational convenience, in the above we define $\sin(0)/0$ *to be* 1*. Furthermore, $\langle f, g\rangle_{PW_{\mathbf{b}}} = \langle f, g\rangle_{L_2(\square\mathbf{b})}$.*

For the kernel function described above, the discrepancy measure $D_{h,S}$ defined in Proposition 4 can be expressed more explicitly.

Theorem 6 (Discrepancy in $PW_{\mathbf{b}}$)**.** *Suppose that p is a probability density function, and that we can write $p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j)$ where each p_j is a univariate probability density function. Let φ_j be the characteristic function associated with p_j. Then,*

$$D_{\mathrm{sinc}_{\mathbf{b}},p}(S)^2 = (\pi)^{-d}\prod_{j=1}^{d}\int_{-b_j}^{b_j}|\varphi_j(\beta)|^2 d\beta -$$
$$\frac{2(2\pi)^{-d}}{s}\sum_{l=1}^{s}\prod_{j=1}^{d}\int_{-b_j}^{b_j}\varphi_j(\beta)e^{iw_{lj}\beta}d\beta +$$
$$\frac{1}{s^2}\sum_{l=1}^{s}\sum_{j=1}^{s}\mathrm{sinc}_{\mathbf{b}}(\mathbf{w}_l,\mathbf{w}_j)\ . \quad (21)$$

This naturally leads to the definition of the *box discrepancy*, analogous to the star discrepancy described in Theorem 2.

Definition 7 (Box Discrepancy)**.** *The box discrepancy of a sequence S with respect to p is defined as,*

$$D_p^{\square\mathbf{b}}(S) = D_{\mathrm{sinc}_{\mathbf{b}},p}(S)\ .$$

For notational convenience, we generally omit the $\mathbf{b}$ from $D_p^{\square\mathbf{b}}(S)$ as long as it is clear from the context. The worse-case integration error bound for Paley-Wiener spaces is stated in the following as a corollary of Theorem 4.

Corollary 8 (Integration Error in $PW_{\mathbf{b}}$)**.** *For $f \in PW_{\mathbf{b}}$ we have*

$$\epsilon_{S,p}[f] \leq \|f\|_{PW_{\mathbf{b}}} D_p^{\square}(S).$$

The integrands we are interested in (i.e. functions in $\mathcal{F}_{\square\mathbf{b}}$) are not members of $PW_{\mathbf{b}}$. However, their damped approximations of the form $\tilde{f}_{\mathbf{u}}(\mathbf{x}) = e^{-i\mathbf{u}^T\mathbf{x}}\,\mathrm{sinc}(T\mathbf{x})$ are members of $PW_{\mathbf{b}}$ with $\|\tilde{f}\|_{PW_{\mathbf{b}}} = \frac{1}{\sqrt{T}}$. Hence, we expect $D_p^{\square}$ to provide a discrepancy measure for integrating functions in $\mathcal{F}_{\square\mathbf{b}}$.

More formally, the expected square error of an integrand drawn from a uniform distribution over $\mathcal{F}_{\square\mathbf{b}}$ is proportional to the square discrepancy measure $D_p^{\square}(S)$. This result is in the spirit of similar average case analysis in the QMC literature (Wozniakowski, 1991; Traub & Wozniakowski, 1994).

Theorem 9 (Average Case Error)**.**

$$\mathbb{E}_{f\sim\mathcal{U}(\mathcal{F}_{\square\mathbf{b}})}\left[\epsilon_{S,p}[f]^2\right] = \frac{(2\pi)^d}{\prod_{j=1}^{d} b_j} D_p^{\square}(S)^2\ . \quad (22)$$

We now give an explicit formula for the case that $p(\cdot)$ is the density function of the multivariate Gaussian distribution with zero mean and independent components. This is an important special case since this is the density function that is relevant for the Gaussian kernel.

Corollary 10 (Discrepancy for Gaussian Distribution)**.** *Let p be the d-dimensional multivariate Gaussian density function with zero mean and covariance matrix equal to* $\mathrm{diag}(\sigma_1^{-2},\ldots,\sigma_d^{-2})$*. We have,*

$$D_p^{\square}(S)^2 = \frac{1}{s^2}\sum_{l=1}^{s}\sum_{j=1}^{s}\mathrm{sinc}_{\mathbf{b}}(\mathbf{w}_l,\mathbf{w}_j) + C -$$
$$\frac{2}{s}\sum_{l=1}^{s}\prod_{j=1}^{d} c_{lj}\,\mathrm{Re}\left(\mathrm{erf}\left(\frac{b_j}{\sigma_j\sqrt{2}} - i\frac{\sigma_j w_{lj}}{\sqrt{2}}\right)\right)\ ,$$

where $c_{lj} = \left(\frac{\sigma_j}{\sqrt{2\pi}}\right)e^{-\frac{\sigma_j^2 w_{lj}^2}{2}}$ *and* $C = \prod_{j=1}^{d}\frac{\sigma_j}{2\sqrt{\pi}}\,\mathrm{erf}\left(\frac{b_j}{\sigma_j}\right)$.

In the above erf is the complex error function; see Weideman (1994) and Mori (1983) for more details.

5. Learning Adaptive QMC Sequences

For simplicity in this section we assume that p is the density function of Gaussian distribution with zero mean. We also omit the subscript p from $D_p^\square$. Similar analysis can be derived for other density functions.

Error characterization via discrepancy measures is typically used in the QMC literature to prescribe optimal sequences. Unlike the star discrepancy (12), the box discrepancy is a smooth function of the sequence with a closed-form formula. This allows us to both evaluate various candidate sequences, and select the one with the lowest discrepancy, as well as to *adaptively learn* a QMC sequence that is specialized for our problem setting via numerical optimization. This task is posed in terms minimization of the box discrepancy function (23) over the space of sequences of s vectors in $\mathbb{R}^d$:

$$S^* = \arg\min_{S=(\mathbf{w}_1 \dots \mathbf{w}_s)\in\mathbb{R}^{ds}} D^\square(S)\ . \qquad (23)$$

The gradient of $D^\square(S)$ is given by the following proposition.

Proposition 11 (Gradient of Box Discrepancy)**.** *Define the following scalar functions and variables,*

$$\begin{aligned}
\mathrm{sinc}'(z) &= \frac{\cos(z)}{z} - \frac{\sin(z)}{z^2}, \quad \mathrm{sinc}'_b(z) = \frac{b}{\pi}\,\mathrm{sinc}'(bz)\ ; \\
c_j &= \left(\frac{\sigma_j}{\sqrt{2\pi}}\right), j = 1, \dots, d\ ; \\
g_j(x) &= c_j e^{-\frac{\sigma_j^2}{2}x^2}\,\mathrm{Re}\left(\mathrm{erf}\left[\frac{b_j}{\sigma_j\sqrt{2}} - i\frac{\sigma_j x}{\sqrt{2}}\right]\right)\ ; \\
g'_j(x) &= -\sigma_j^2 x g_j(x) + \sqrt{\frac{2}{\pi}} c_j \sigma_j e^{-\frac{b_j^2}{2\sigma_j^2}} \sin(b_j x)\ .
\end{aligned}$$

byThen, the elements of the gradient vector of $D^\square$ are given

$$\frac{\partial D^\square}{\partial w_{lj}} = -\frac{2}{s} g'_j(w_{lj}) \left(\prod_{q\neq j} g_q(w_{lq})\right) +$$

$$\frac{2}{s^2} \sum_{\substack{m=1\\ m\neq l}}^{s} \left(b_j\,\mathrm{sinc}'_{b_j}(w_{lj}, w_{mj}) \prod_{q\neq j} \mathrm{sinc}_{b_q}(w_{lq}, w_{mq})\right)\ . \quad (24)$$

The gradient can be plugged into any first order numerical solver for non-convex optimization. We use non-linear conjugate gradient in Section 6.2.

The above learning mechanism can be extended in various directions. For example, QMC sequences for n-point rank-one Lattice Rules are integral fractions of a lattice defined by a single generating vector $\mathbf{v}$. This generating vector may be learnt via local minimization of the box discrepancy.

6. Experiments

In this section we report experiments with both classical QMC sequences and adaptive sequences learnt from box discrepancy minimization.

6.1. Experiments With Classical QMC Sequences

We examine the behavior of classical low-discrepancy sequences when compared to random Fourier features (i.e., MC). We consider four sequences: Halton, Sobol', Lattice Rules, and Digital Nets. For Halton and Sobol', we use the implementation available in MATLAB[1]. For Lattice Rules and Digital Nets, we use publicly available implementations[2]. For the low-discrepancy sequence, we use scrambling and shifting techniques recommended in the QMC literature (see Dick et al. (2013) for details). For Sobol', Lattice Rules and Digital Nets, scrambling introduces randomization and hence variance. For Halton sequence, scrambling is deterministic, and there is no variance. The generation of these sequences is extremely fast, and quite negligible when compared to the time for any reasonable downstream use. Therefore, we do not report running times as these are essentially the same across methods.

Quality of Kernel Approximation In our setting, the most natural and fundamental metric for comparison is the quality of approximation of the Gram matrix. We examine how close $\tilde{\mathbf{K}}$ (defined by $\tilde{\mathbf{K}}_{ij} = \tilde{k}(\mathbf{x}_i, \mathbf{x}_j)$ where $\tilde{k}(\cdot,\cdot) = \langle \hat{\Psi}_S(\cdot), \hat{\Psi}_S(\cdot)\rangle$ is the kernel approximation) is to the Gram matrix $\mathbf{K}$ of the exact kernel. In all comparisons, we work with a Gaussian kernel with bandwidth σ set by using cross-validation in favor of the Monte Carlo approach.

We examine four datasets: `cpu` (6500 examples, 21 dimensions), `census` (a subset chosen randomly with 5,000 examples, 119 dimensions), `USPST` (1,506 examples, 250 dimensions after PCA) and `mnist` (a subset chosen randomly with 5,000 examples, 250 dimensions after PCA). The reason we do subsampling on large datasets is to be able to compute the full exact Gram matrix for comparison purposes. The reason we do dimensionality reduction is that the maximum dimension supported by the Lattice Rules implementation we use is 250. To measure the quality of approximation we use $\|\mathbf{K} - \tilde{\mathbf{K}}\|_2 / \|\mathbf{K}\|_2$. The plots are shown in Figure 2.

We can clearly see that classical low-discrepancy sequences consistently produce better approximations to the Gram matrix. Among the four classical QMC sequences, the Digital Net, Lattice and Halton sequences yield much lower error. Similar results were observed for other

[1] http://www.mathworks.com/help/stats/quasi-random-numbers.html

[2] http://people.cs.kuleuven.be/ dirk.nuyens/qmc-generators/

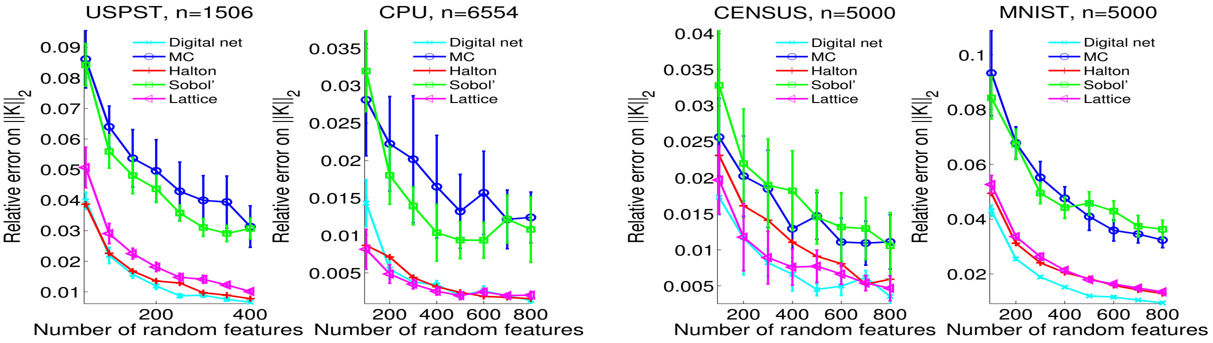

Figure 2. Relative error on approximating the Gram matrix, i.e. $\|\mathbf{K}-\tilde{\mathbf{K}}\|_2/\|\mathbf{K}\|_2$, for various s. For each kind of random feature and s, 10 independent trials are executed, and the mean and standard deviation are plotted.

datasets (not reported here). Although using randomized variants of QMC sequences may incur some variance, the variance is quite small compared to that of the MC random features. We have observed that randomized QMC sequences almost uniformly yield higher accuracies than non-randomized QMC sequences (results not reported).

Does better Gram matrix approximation translate to lower generalization errors? We consider two regression datasets, `cpu` and `census`, and we use (approximate) kernel ridge regression to build a regression model. The ridge parameter is set by the optimal values we obtain via cross-validation on the MC sequence. Table 1 summarizes the results.

	s	HALTON	SOBOL'	LATTICE	DIGIT	MC
CPU	100	**0.0367** (0)	0.0383 (0.0015)	0.0374 (0.0010)	0.0376 (0.0010)	0.0383 (0.0013)
	500	**0.0339** (0)	0.0344 (0.0005)	0.0348 (0.0007)	0.0343 (0.0005)	0.0349 (0.0009)
	1000	**0.0334** (0)	0.0339 (0.0007)	0.0337 (0.0004)	0.0335 (0.0003)	0.0338 (0.0005)
CENSUS	400	**0.0529** (0)	0.0747 (0.0138)	0.0801 (0.0206)	0.0755 (0.0080)	0.0791 (0.0180)
	1200	**0.0553** (0)	0.0588 (0.0080)	0.0694 (0.0188)	0.0587 (0.0067)	0.0670 (0.0078)
	1800	**0.0498** (0)	0.0613 (0.0084)	0.0608 (0.0129)	0.0583 (0.0100)	0.0600 (0.0113)

Table 1. Regression error, i.e. $\|\hat{\mathbf{y}}-\mathbf{y}\|_2/\|\mathbf{y}\|_2$ where $\hat{\mathbf{y}}$ is the predicted value and $\mathbf{y}$ is the ground truth. For each kind of random feature and s, 10 independent trials are executed, and the mean and standard deviation are listed.

As we see, for `cpu`, all the sequences behave similarly, with the Halton sequence yielding the lowest test error. For `census`, the advantage of using Halton sequence is significant (almost 20% reduction in generalization error) followed by Digital Nets and Sobol'. In addition, MC sequence tends to generate higher variance across all the sampling size. Overall, QMC sequences, especially Halton, outperform MC sequences on these datasets.

Behavior of Box Discrepancy Next, we examine if $D^{\square}$ is predictive of the quality of approximation. We compute the discrepancy values for the different sequences with different sample sizes s. Note that while the bounding box $\square\mathbf{b}$ is set based on observed ranges of feature values in the dataset, the actual distribution of points $\bar{\mathcal{X}}$ encountered inside that box might be far from uniform.

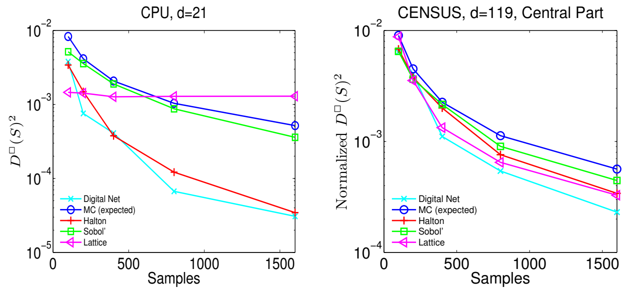

Figure 3. Discrepancy values ($D^{\square}$) for the different sequences on `cpu` and `census`. For `census` we measure the discrepancy on the central part of the bounding box (we use $\square\mathbf{b}/2$ in the optimization instead of $\square\mathbf{b}$).

In Figure 3, for `cpu` we see a strong correlation between the quality of approximation and the discrepancy values. Interestingly, Lattice Rules sequences start with low discrepancy, which does not decrease with increasing s. For `census`, using the original bounding box yielded very little difference between sequences (graph not shown). Instead, we plot the discrepancy when measured on the central part of the bounding box (i.e., $\square\mathbf{b}/2$), which is equal to the integration error averages over that part of the bounding box. Presumably, points from $\bar{\mathcal{X}}$ concentrate in that region, and they may be more relevant for downstream predictive task. Again, we see strong correlation between approximation quality and the discrepancy value.

6.2. Experiments With Adaptive QMC

In this subsection we provide a proof of concept for learning adaptive QMC sequences as described in Section 5. Sequences were optimized by applying non-linear Conjugate Gradient to optimize the normalized box discrepancy (i.e., $(2\pi)^d/(\prod_{j=1}^d b_j)D_p^{\square}(S)^2$). The Halton sequence is used as the initial setting of the optimization variables S.

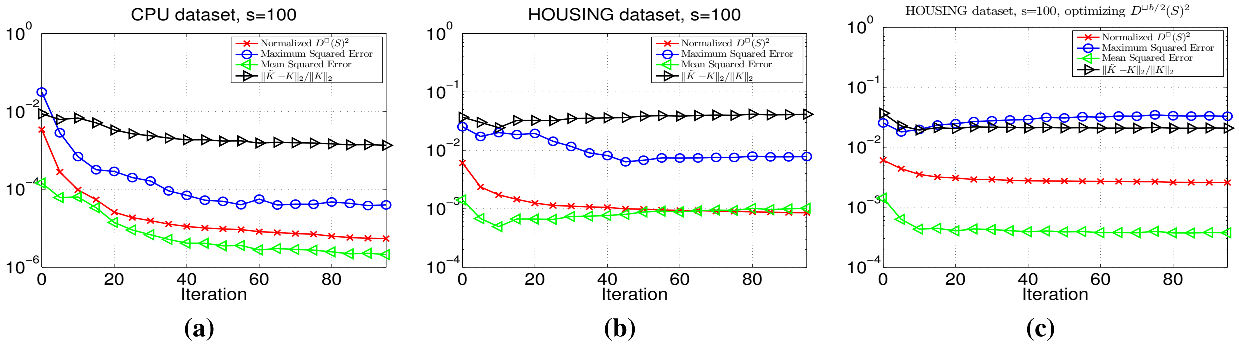

Figure 4. Examining the behavior of learning adaptive QMC sequences. Various metrics on the Gram matrix approximation are plotted.

It should be noted that adaptive QMC estimation is data-independent and as such a one-time expense given s, d, $\mathbf{b}$ and σ, with applicability to a variety of downstream applications of kernel methods.

In Figure 4 we examine how various metrics (discrepancy, maximum squared error, mean squared error, norm of the error) on the Gram matrix approximation evolve during the optimization. In Figure 4 (a) we examine the behavior on `cpu`. We see that all metrics go down as the iteration progresses. This supports our hypothesis that by optimizing the box discrepancy we can improve the approximation of the Gram matrix. We also see interesting behavior in Figure 4 (b), which examines the metrics on the scaled version of the `housing` dataset. Initially all metrics go down, but eventually mean error and the norm error start to go up (the maximum error continues to go down). One plausible explanation is that the integrands are not uniformly distributed in the bounding box, and that by optimizing the expectation over the entire box we start to overfit it, thereby increasing the error in those regions of the box where integrands actually concentrate. One possible way to handle this is to optimize closer to the center of the box (e.g., on $\Box\mathbf{b}/2$) under the assumption that integrands concentrate there. In Figure 4 (c) we try this on `housing` and see that now the mean error and the norm error are much improved, which supports the interpretation above. But the maximum error eventually goes up. This is quite reasonable as the outer parts of the bounding box are harder to approximate, so the maximum error is likely to originate from there. Subsequently, we stop the adaptive learning of the QMC sequences early, to avoid the actual error from going up due to averaging.

Next, we investigate the generalization error. We use the same learning algorithm as the previous subsection. The ridge parameter is set by the value which is near-optimal for both sequences in cross-validation. Table 2 summarizes the results. For `cpu`, the adaptive sequences can yield lower test error when the sampling size is small. When $s = 500$ or even larger (not reported here), the performance of the sequences are very close. For `census`, the adaptive sequences do not show any benefit until s is 1200. Afterwards we can see at least one of the two adaptive sequences can yield much lower error than Halton sequence for each sampling size. However, in some cases, adaptive sequences sometimes produce errors that are bigger than the unoptimized sequences. In most cases, the adaptive sequence on the central part of the bounding box outperforms the adaptive sequence on the entire box.

	s	HALTON	ADAPTIVE$_{\mathbf{b}}$	ADAPTIVE$_{\mathbf{b}/4}$
CPU	100	0.0304	0.0315	**0.0296**
	300	0.0303	**0.0278**	0.0293
	500	0.0348	**0.0347**	0.0348
CENSUS	400	**0.0529**	0.1034	0.0997
	800	**0.0545**	0.0702	0.0581
	1200	0.0553	0.0639	**0.0481**
	1800	0.0498	0.0568	**0.0476**
	2200	0.0519	**0.0487**	0.0515

Table 2. Regression error, i.e. $\|\hat{\mathbf{y}} - \mathbf{y}\|_2/\|\mathbf{y}\|_2$ where $\hat{\mathbf{y}}$ is the predicted value and $\mathbf{y}$ is the ground truth.

7. Conclusion and Future Work

This paper is the first to exploit high-dimensional approximate integration techniques from the QMC literature in large-scale kernel methods, with promising empirical results backed by rigorous theoretical analyses. Avenues for future work include incorporating stronger data-dependence in the estimation of adaptive sequences and analyzing how resulting Gram matrix approximations translate into downstream performance improvements for a variety of tasks.

Acknowledgements

The authors would like to thank Josef Dick for useful pointers to literature about improvement of the QMC sequences; Ha Quang Minh for several discussions on Paley-Wiener spaces and RKHS theory. This research was supported by the XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323. This work was done while J. Yang was a summer intern at IBM Research.

References

Berlinet, A. and Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

Bochner, S. Monotone funktionen, Stieltjes integrale und harmonische analyse. *Math. Ann.*, 108, 1933.

Caflisch, R. E. Monte Carlo and Quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1 1998. ISSN 1474-0508.

Cucker, F. and Smale, S. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39:1–49, 2001.

Dick, J., Kuo, F. Y., and Sloan, I. H. High-dimensional integration: The Quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 5 2013. ISSN 1474-0508.

Mori, M. A method for evaluation of the error function of real and complex variable with high relative accuracy. *Publ. RIMS, Kyoto Univ.*, 19:1081–1094, 1983.

Niederreiter, H. *Random number generation and Quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. ISBN 0-89871-295-5.

Peloso, M. M. Classical spaces of holomorphic functions. Technical report, Universit‘ di Milano, 2011.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Neural Information Processing Systems (NIPS)*, 2007.

Schölkopf, B. and Smola, A. (eds.). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

Sloan, I. H. and Wozniakowski, H. When are Quasi-Monte Carlo algorithms efficient for high dimensional integrals. *Journal of Complexity*, 14(1):1–33, 1998.

Traub, J. F. and Wozniakowski, H. Breaking intractability. *Scientific American*, pp. 102–107, 1994.

Wahba, G. (ed.). *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1990.

Weideman, J. A. C. Computation of the complex error function. *SIAM Journal of Numerical Analysis*, 31(5): 1497–1518, 10 1994.

Wozniakowski, H. Average case complexity of multivariate integration. *Bull. Amer. Math. Soc.*, 24:185–194, 1991.

Yao, K. Applications of Reproducing Kernel Hilbert Spaces - bandlimited signal models. *Inform. Control*, 11:429–444, 1967.