A Taste of Quasi Monte Carlo

1 Introduction

Quasi Monte Carlo (QMC) methods attempt to approximate intractable integrals. The Monte Carlo method to estimate the integral $\int f dP$ for $f : \mathbb{R}^s \to \mathbb{R}$ is to take n independent samples from the probability distribution P and average them: $\frac{1}{n} \sum_{x_i \sim p(X)} f(x_i)$. As the estimator is unbiased, the expected squared error $E[(\frac{1}{n} \sum_{x_i \sim P(X)} f(x_i) - E[f(X)])^2]$ is just the variance of $f(X)$ divided by n, so the root mean square error will decay as $O(n^{-1/2})$. QMC methods try to choose samples x_i which lead to a faster decay than $n^{-1/2}$.

It turns out we can get an error that decays as $O(n^{-1})$. But there's a catch. Unlike standard Monte Carlo methods, the error in QMC methods can also grow exponentially with the dimension s of the space we're integrating in. Fortunately, many functions we might want to integrate in $\mathbb{R}^s$ don't make full use of all the available space; they concentrate most of their variance along a much smaller number of dimensions. QMC scales exponentially in these *effective dimensions*. For example, the authors in Paskov and Traub 1996 use Quasi Monte Carlo to price financial derivatives in a 360 dimensional space; because the effective dimension was low, this performed far better than the Monte Carlo estimator, shocking many in the community who assumed that QMC methods were inappropriate in high dimensions. This summary will lead up to a proof of this result, roughly following the approaches in Dick et al. 2013 and L'Ecuyer and Lemieux 2002.

For the much of this summary, I will restrict the discussion to integrals over a unit hypercube in s dimensions using Lebesgue measure: $\int_{[0,1]^s} f(x)\, dx$. This can be done without loss of generality. For one dimensional integrals with respect to a distribution P with an invertible CDF Φ, we can express $\int f \, dP$ as $\int_0^1 f(\Phi^{-1}(x))\, dx$. In s dimensions, factor $\Phi(x) = \Phi_1(x_1) \prod_{i \in [s]} \Phi_{i+1}(x_{i+1} | x_{\leq i})$ and integrate auto-regressively:

$$\int_{[0,1]^s} f(\Phi_1^{-1}(x_1), \Phi_2^{-1}(x_2 | \Phi_1^{-1}(x_1), \dots)\, dx$$

2 Techniques

Intuitively, Monte Carlo samples are less effective when they are clumped closely together; we'd like the samples to be well dispersed throughout the unit cube. All QMC techniques can be seen as ways to spread the particles out. There are two main families: lattice rules and digital nets.

2.1 Lattice Rules

Sampling points on a lattice is an natural way to prevent them from clumping together. Choose a *generator* $z \in \{1, 2, \dots n-1\}^s$. Then choose the n samples as $\{\frac{kz}{n} \bmod 1 : k \in [0, \dots n-1]\}$. To ensure that every one dimensional projection of these points has n distinct values, we can require that $\gcd(z_i, n) = 1$ for each coordinate z_i of the generator z. A simple way to achieve this is to make n prime; I will assume prime n for the rest of this summary.

To choose the generator z, we can optimize each component sequentially. Specifically, let $z_1 = 1$. At each subsequent step, having fixed $z_1, \dots, z_{i-1}$, we can pick z_i to minimize the worst case integration error using the samples this generator would produce over any function of i dimensions in a given class. As we will see in later sections, this worst case error is easy to compute, and finding a minimal z_i for each of the s steps can be done in time $n \log n$. This is known as the *component by component* (CBC) construction.

2.2 Digital Nets

Digital nets have a different way of spacing out sampled points. In this paradigm, we subdivide the unit hypercube in $\mathbb{R}^s$ into a size-b^{-ds} grid of boxes $\{\prod_{j=1}^{s} [\frac{j}{b^d}, \frac{j+1}{b^d}) : j \in [b^d]\}$ for some base b and precision d. We then pick b^t points per box, giving $b^{t+ds} = b^m$ samples. The result is called a (t, m, s) *net*. The general procedure goes as follows:

- Make a vector from the digits in the base b representation of each integer up to b^m. You get a $m \times 2^m$ matrix A.

- Multiply s different matrices $C_1, C_2, \dots C_s$ by A to get embeddings $B_1, B_2, \dots B_s$.

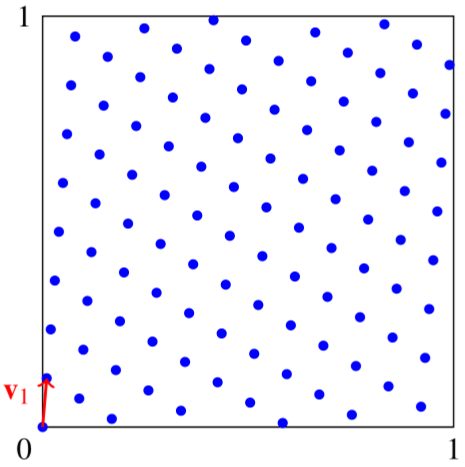

Figure 1: Example of samples from a lattice rule from L'Ecuyer 2018. The vector v_1 indicated on the plot is the generator.

- Interpret the ith column of B_j as the jth coordinate for sample i in base b^{-1}.

Theorem 1. *As long as the first d rows of each matrix C_i are all linearly independent, this procedure will produce exactly b^t samples in each box.*

Proof. Consider an arbitrary box v. All the points in v will have ith coordinates with the same first d coefficients in base b^{-1}. Call this vector of shared digits $u_{v,i}$, and concatenate them to get u_v, a vector representation of box v. Let $\bar{C}_i$ indicate the matrix containing the first d rows of C_i. The digital net procedure creates samples in dimension i by applying C_i to every vector in $\mathbb{Z}_b^m$ where $\mathbb{Z}_b$ is the finite field of integers mod b. This means that the number of samples we generate within v is the number of solutions z to the equation

$$u_v = \begin{bmatrix} \bar{C}_1 \\ \cdots \\ \bar{C}_s \end{bmatrix} z$$

This will be b^{m-t} if and only if the rows are linearly independent.

□

For the rest of this review, I will focus on lattice rules rather than digital nets. That said, the same techniques can be generalized to apply to digital nets as well.

2.3 Randomization

While the deterministic constructions outlined so far prevent particles from clumping, they will not result in unbiased estimators as in standard Monte Carlo. We can fix this by adding back some randomization. Sample a vector $\Delta \in [0, 1]^s$ uniformly and shift all the points by this offset mod 1. The result is unbiased, as $(\Delta + x_i)$ mod 1 is still distributed uniformly on the unit cube for any deterministic offset x_i. For the rest of the paper, I will indicate addition modulo 1 by the notation $\oplus$; this makes the shifted samples $x_i \oplus \Delta$. To decrease variance further, we can draw multiple Δ samples and average the associated randomized lattice estimates.

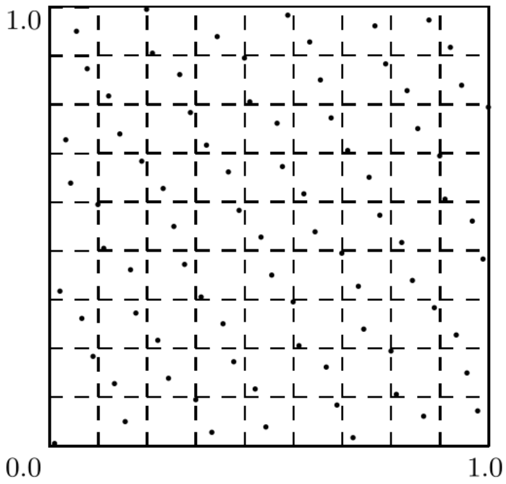

Figure 2: Example of digital net from L'Ecuyer and Lemieux 2002. By Theorem 1, each box must contain exactly one sample.

3 Worst Case Error

We will measure the quality of a set of points used for QMC integration by its worst case integration error. We can afford to be more abstract in this section, so rather than just considering the uniform distribution over the unit hypercube, I will consider sampling from an arbitrary probability measure μ.

3.1 RKHS Formulation

The worst case integration error for the points $x_1, \ldots x_n \in \mathbb{R}^s$ over the set $\mathcal{F}$ of functions we might want to integrate is

$$L_{n,\mathcal{F},s} := \sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i \in [n]} f(x_i) - \int f(x)\, d\mu(x) \right)$$

Calculating this supremum for arbitrary such sets is difficult in general, but when $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS), the analysis becomes much simpler. Recall that a reproducing kernel Hilbert space with kernel function $k(x, y)$ is a complete inner product space over functions such that $f(x) = \langle f, k(x, \cdot)\rangle$.

Theorem 2. *Let $\mathcal{F}$ be a unit ball in RKHS H with kernel k. The worst case squared integration error for points $x_1, \dots x_n \in \mathbb{R}^s$ is*

$$\begin{aligned} L^2_{n,\mathcal{F},s} = &\int\int k(x, y)\, d\mu(x)\, d\mu(y) \\ &- \frac{2}{n}\sum_{i\in[n]}\int k(x_i, y)\, d\mu(y) \\ &+ \frac{1}{n^2}\sum_{i\in[n]}\sum_{j\in[n]} k(x_i, x_j) \end{aligned}$$

Proof.

$$\begin{aligned} L_{n,\mathcal{F},s} &= \sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i\in[n]}\langle f, k(x_i, \cdot)\rangle - \int\langle f, k(x, \cdot)\rangle\, d\mu(x)\right) \\ &= \sup_{f\in\mathcal{F}}\left\langle f, \sum_{i\in[n]} k(x_i, \cdot) - \int k(x, \cdot)\, d\mu(x)\right\rangle \\ &= \|\xi\|_H \end{aligned}$$

where $\xi = \sum_{i\in[n]} k(x_i, \cdot) - \int k(x, \cdot)\, d\mu(x)$. Taking $\langle \xi, \xi\rangle$ gives the stated bound for L^2_n. □

The function ξ is called the *representer* of the QMC error. Theorem 2 tells us that to minimize the worst case integration error for any function with bounded norm in the RKHS, we can minimize the norm of the representer instead.

3.2 Shift Averaged Worst Case Error

We can extend the worst case error formula for an arbitrary RKHS to average over the sampled shift values Δ used to make QMC estimates unbiased. The procedure is specialized to integration over the uniform distribution on the unit hypercube.

Theorem 3. *The average squared representer norm of shifted samples is*

$$-\int_{[0,1]^s}\int_{[0,1]^s} k(x, y)\, dx\, dy + \frac{1}{n^2}\sum_{i\in[n]}\sum_{j\in[n]} k_{sh}(x_i, x_j)$$

where $k_{sh}(x, y) = \int_{[0,1]^s} k(x \oplus \Delta, y \oplus \Delta)\, d\Delta$

Proof. Start by substituting the point set in Theorem 2 with its shifted version.

$$\begin{aligned} &\int_{[0,1]^s}\int_{[0,1]^s} k(x, y)\, dx\, dy \\ &- \frac{2}{n}\sum_{i\ \in[n]}\int_{[0,1]^s}\int_{[0,1]^s} k(x_i \oplus \Delta, y)\, dy\, d\Delta + \\ &\frac{1}{n^2}\sum_{i\ \in[n]}\sum_{j\ \in[n]}\int_{[0,1]^s} k(x_i \oplus \Delta, x_j + \Delta)\, d\Delta \end{aligned}$$

In the second term, we can perform a change of variables so that $x = x_i \oplus \Delta$, making its double integral equal to that of the first term. Applying the definition of k_{sh} to the last term yields the result. □

This shows that the average worst case error for a randomly shifted QMC scheme is the same as the worst case error for the un-shifted scheme, but using the shift invariant kernel k_{sh} instead of the base kernel k. More precisely, let $\mathcal{F}_{\text{sh}}$ be the RKHS with kernel k_{sh}. The shift averaged worst case error for a set of n points in $\mathbb{R}^s$ is $L_{\mathcal{F}_{\text{sh}},n,s}$. We look at the integration error only over the space $\mathcal{F}_{\text{sh}}$ of shift invariant functions. Note that if the base kernel k is already shift invariant such that $k(x \oplus \Delta, y \oplus \Delta) = k(x, y)$, then $k_{\text{sh}} = k$ and $\mathcal{F}_{\text{sh}} = \mathcal{F}$.

4 Connections to Other Distributional Distances

In Section 3, we evaluated how close the finite sample approximation chosen by QMC was to the target distribution (uniform on the unit hypercube) with a distance of the form

$$\sup_{f\in\mathcal{G}}\left(\int f(x)d\nu(x) - \int f(x)\, d\mu(x)\right)$$

for some choice of RKHS $\mathcal{G}$. In general, this is known as the *maximum mean discrepancy* between the distributions. It is worth spending a moment to relate this choice of metric to other common distances between distributions; we will see that other plausible characterizations of distributional mismatch are really instances or upper bounds on the MMD.

4.1 Relationship to Kolmogorov Distance

Traditionally, QMC algorithms were analyzed using what is known as the *anchored Sobolev space*, with kernel $k(x, y) = 2 - \max(x, y)$ and inner product $\langle g, h\rangle = g(1)f(1) - \int_0^1 g'(x)h'(x)\, dx$. It includes all continuous functions on the unit interval with first derivatives in L^2. While we will not be using this space when we analyze multidimensional integrals in future sections, I will discuss it briefly here, as it shows how the MMD can be seen as a lower bound of the Kolmogorov distance when the associated RKHS is the anchored Sobolev space. Recall that the Kolmogorov distance has a similar form to the definition of MMD above, but with $\mathcal{G} = \{1_{\cdot \leq y} : y \in \mathbb{R}\}$. When $\mathcal{G}$ is not a RKHS, the form is known as an *integral probability metric.*

Theorem 4. *The maximum mean discrepancy between the sample and target distributions using the anchored Sobolev kernel is a lower bound on their Kolmogorov distance.*

Proof. Expand the associated MMD (which by Theorem 2 is the norm of the representer).

$$\langle \xi, \xi \rangle = \xi(1)^2 - \int_0^1 (\xi'(y))^2 \, dy$$

$$\xi'(y) = \frac{-1}{n} \sum_{i \in [n]} 1_{y \geq x_i} + \int_0^1 1_{y \geq x} \, dx$$

$$\xi(1) = \frac{1}{n} \sum_{i \in [n]} 1 - \int_0^1 1 \, dx = 0$$

$$\langle \xi, \xi \rangle = \int \left(\frac{1}{n} \sum_{i \in [n]} 1_{y \geq x_i} - \int_0^1 1_{y \geq x} \, dx \right) dy$$

$$= \int D^*(y) \, dy \leq \|D^*\|_\infty$$

where $D^*(y) = -\xi'(y)$ is called the *local discrepancy* at y, giving the difference between the fraction of samples in a region and the true probability of the region. The quantity $\|D^*\|_\infty$ precisely the Kolmogorov distance between the empirical and target distributions. □

The Kolmogorov distance between the sample and true distributions is also called the *star discrepancy* of the sampled points. Note that in this space, $\langle f, \xi \rangle = \int f' D^* d\mu$. Working from line 2 of Theorem 2, Holder's inequality tells us that $L_n \leq \|f'\|_1 \|D^*\|_\infty$, which is known as the Koksma-Hlawka inequality.

4.2 Relationship to Stein Discrepancy

In addition to Kolmogorov distance, the MMD also generalizes kernelized Stein Discrepancy. The kernelized Stein Discrepancy between distributions p and q over $\mathcal{F}$, a unit ball in a RKHS, is defined as

$$\sup_{f \in \mathcal{F}} E_{x \sim q}[A_p f(x)]$$

where $\mathcal{A}_p$ is the linear *Stein* operator which maps $f \mapsto \nabla \log p(x) f(x) + \nabla f(x)$. If the functions in the RKHS satisfy $\lim_{\|x\| \to \infty} p(x) f(x) = 0$ then this distance is zero if and only if the distributions are the same. The class of functions with this property is called the *Stein Class* of p.

Liu et al. 2016 notes that we can interpret this definition as an instance of maximum mean discrepancy over the RKHS $\mathcal{G} = \{\mathcal{A}_p f : f \in \mathcal{F}\}$. The term $\int f(x) \, d\mu(x)$ in the MMD is zero for functions in the Stein class, making $L_{\mathcal{G},n} = \|\frac{1}{n} \sum_{i \in [n]} k(x_i, \cdot)\|_H$.

5 Weighted Unanchored Sobolev Spaces

To work in multiple dimensions, we will follow Sloan and Wozniakowski 2002 and turn to a different RKHS: the *weighted unanchored Sobolev space.* Let $\eta(x_j, y_j) = \frac{1}{2} B_2(|x_j - y_j|) + (x_j - \frac{1}{2})(y_j - \frac{1}{2})$ where $B_2 = x^2 - x + \frac{1}{6}$ is the Bernoulli polynomial of degree 2. In $\mathbb{R}^s$, define the kernel $k(x, y) = \prod_i (1 + \gamma_i \eta(x_i, y_i))$ for some sequence $\gamma_1, \ldots, \gamma_s \leq 1$ of constants. Intuitively, each γ_i captures how much the values at dimension i can effect the outcome of the function. Using the identity that $\prod_{j \in [s]} (1 + a_j) = \sum_{u \subseteq [s]} \prod_{j \in u} a_j$ and letting $\gamma_u = \prod_{i \in u} \gamma_i$ and $\gamma_\emptyset = 1$, we can rewrite this definition as a sum over subsets:

$$k(x, y) = \sum_{u \subseteq \{1:s\}} \prod_{j \in u} \gamma_u \eta(x_j, y_j)$$

5.1 Integrating the Kernel

The nice thing about B_2 is it has the property that $\int_0^1 B_2(|x - y|) dy = \int_0^1 B_2(y) \, dy = 0$. This makes it easy to integrate the kernel function for unanchored Sobolev spaces.

Theorem 5. *For any fixed x, $\int_{[0,1]^s} k(x, y) dy = 1$.*

Proof.

$$\int_{[0,1]^s} k(x, y) dy = \sum_{u \subseteq \{1:s\}} \gamma_u \int_0^1 \prod_{j \in u} \eta(x_j, y_j) \, dy$$

$$= \sum_{u \subseteq \{1:s\}} \gamma_u \prod_{j \in u} \int \eta(x_j, y_j) dy_j$$

$$= \sum_{u \subseteq \{1:s\}} \gamma_u \prod_{j \in u} \Bigg[$$

$$\frac{1}{2} \int B_2(|x_j - y_j|) dy_j + \left(x_j - \frac{1}{2} \right) \int \left(y_j - \frac{1}{2} \right) dy_j \Bigg]$$

$$= 1$$

Both inner integrals are zero, leaving a product over the empty set. □

5.2 Shift Averaged Kernels for Unanchored Sobolev Spaces

As we discovered in Theorem 3, when a random shift is applied to a set of points, the integration error for functions in a RKHS with kernel k can be described by the original formula, but using a shift invariant kernel k_{sh}. For weighted unanchored Sobolev spaces, this shift invariant kernel is even simpler than the original.

Theorem 6. *The shift invariant kernel for weighted unanchored Sobolev spaces is*

$$k_{sh}(x, y) = \prod_j (1 + \gamma_j B_2(|x_j - y_j|))$$

Proof. Using the independence of the random shift components in each dimension, the shift invariant kernel can be written as

$$\sum_{u \in \{1:s\}} \gamma_u \prod_{j \in u} \left(\frac{1}{2} \beta_2(|x_j - y_j|) + I(x_j, y_j) \right)$$

where

$$I(a, b) = \int_0^1 ((a \oplus \Delta) - \frac{1}{2})((b \oplus \Delta) - \frac{1}{2}) d\Delta$$

It remains to show that $I(a, b) = \frac{1}{2} B_2(|a - b|)$. As I is symmetric, assume without loss of generality that $a \leq b$. We can remove the modular arithmetic by considering three cases: either adding the shift will keep b below one, or it will take b above 1 but not a, or it will take both above 1. The first case will be true whenever $\Delta < 1 - b$, the second whenever $1 - b \leq \Delta < 1 - a$, and the third when $1 - a \leq \Delta$. Therefore,

$$\begin{aligned} I(a, b) = &\int_0^{1-b} (a + \Delta - \frac{1}{2})(b + \Delta - \frac{1}{2}) d\Delta \\ &+ \int_{1-b}^{1-a} (a + \Delta - \frac{1}{2})(b + \Delta - 1 - \frac{1}{2}) d\Delta \\ &+ \int_{1-a} (a + \Delta - 1 - \frac{1}{2})(y + \Delta - 1 - \frac{1}{2}) d\Delta \end{aligned}$$

Integrating gives $\frac{1}{2}(b - a)^2 - \frac{1}{2}(b - a) + \frac{1}{12}$, which simplifies to $\frac{1}{2} B_2(|b - a|)$. □

Note that the shift invariant kernel, like the original kernel, has the property that $\int_{[0,1]^s} k(x, y) dy = 1$ for fixed x; the proof from Theorem 5 still applies. By combining Theorems 2, 5 and 6, we get

Corollary 7. *The shift averaged worst case error for weighted unanchored Sobolev space $\mathcal{F}$ is*

$$L^2_{\mathcal{F}_{sh}, n, s} = -1 + \frac{1}{n^2} \sum_{i \in [n]} \sum_{j \in [n]} \prod_{d \in [s]} (1 + \gamma_d B_2(|x_d - y_d|))$$

5.3 A Lower Bound on L_n

This expression for shift averaged worst case error provides a disheartening result: that integration error will grow exponentially in $\sum_i \gamma_i$, no matter what points are chosen by QMC. We can see this by finding a lower bound on L_n^2.

Theorem 8.

$$L^2_{n,s} \geq -1 + \frac{1}{n} \exp\left(\frac{1}{7} \sum_{d=1}^{s} \gamma_d \right)$$

Proof. Starting from corollary 7 we find

$$L_n^2 = -1 + \frac{1}{n^2} \sum_{i \in [n]} \sum_{j \in [n]} \prod_{d=1}^{s} (1 + \gamma_j B_2(|x_j - y_j|))$$

Break the sum into the diagonal and off-diagonal elements. As $B_2(0) = \frac{1}{6}$ and B_2 is always above $-\frac{1}{6}$, we get the lower bound

$$-1 + \frac{1}{n^2} \sum_{j=1}^{n} \prod_{d=1}^{s} \left(1 + \frac{\gamma_d}{6}\right) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j \neq i, j \in [n]} \prod_{d=1}^{s} \left(1 - \frac{\gamma_d}{6}\right)$$

The last term is positive, so we can drop it. This means

$$L^2_{\mathcal{F}, n} \geq -1 + \frac{1}{n} \exp\left(\sum_{j=1}^{s} \log(1 + \frac{\gamma_j}{6}) \right)$$

As $\log(1 + x) \geq \frac{x}{1+x}$ for all x and $\gamma_j \leq 1$, we can write this as

$$\begin{aligned} &\geq -1 + \frac{1}{n} \exp\left(\sum_{j=1}^{s} \frac{\gamma/6}{(6 + \gamma)/6} \right) \\ &\geq -1 + \frac{1}{n} \exp\left(\sum_{j=1}^{s} \frac{\gamma}{7} \right) \end{aligned}$$

□

We will never be able to get rid of this $\exp(\sum_{j=1}^{s} \gamma_j)$ factor. This means that QMC methods will only work well for integrating functions in spaces for which this sum of the weights (or *effective dimension*) is small. Intuitively, this requires the functions to concentrate their variances within a small number of dimensions.

6 Beating Monte Carlo with Lattice Rules

For a fixed set of samples, we now have an expression for the worst case integration error. We can use this to show that the lattice points produced by the CBC algorithm from Section 2.1 give root mean square error that decays as $O(n^{-1})$ with the number of samples: far faster than the $O(n^{-1/2})$ scaling of Monte Carlo.

6.1 Worst Case Lattice Error

When the sample set $x_1, \dots x_n$ are lattice points generated by the vector z, we can simplify the formula for worst case error. Let $\mathcal{L}(z_1, \dots, z_s)$ be the worst case shift averaged squared error for the n lattice points derived from generating vector $(z_1, \dots z_s)$.

Lemma 9.

$$\mathcal{L}(z_1, \dots z_s) = -1 + \frac{1}{n}\sum_{l=0}^{n-1}\prod_{d=1}^{s}(1 + \gamma_d \bar{B}_2(lz_d/n))$$

where $\bar{B}_2(x) = B_2(x \bmod 1)$*.*

Proof. By corollary 7,

$$\mathcal{L}(z_1, \dots z_s) = -1 + \frac{1}{n^2}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\prod_{d\in[s]}(1 + \gamma_d \bar{B}_2((j-k)z_d/n))$$

Performing a change of variables $l := j - k$ yields the claim. □

This gives the worst case error for lattice points using a given generator z. The next step is to find the worst case for whatever generator results from applying the CBC construction algorithm.

6.2 Preliminaries for CBC Error

We will prove a bound on the error for the CBC algorithm by induction. In one dimension, Lemma 9 tells us that

$$\mathcal{L}(z_1, \dots z_s) = -1 + \frac{1}{n}\sum_{l=0}^{n-1}(1 + \gamma_1 \bar{B}_2(lz_1/n))$$

To handle this average of Bernoulli polynomials over a lattice, we will need a few tools from Fourier analysis. When a function $f : [0, 1] \to \mathbb{R}$ has an absolutely convergent Fourier series, it can be written in the form

$$f(x) = \sum_{h\in\mathbb{Z}} \hat{f}(h)e^{2\pi ihx}$$
$$\hat{f}(h) = \int_{[0,1]} f(x)e^{-2\pi ihx}\,dx$$

As complex sinusoids of different frequencies are orthogonal, we can rely on

Theorem 10 (Character Property)**.**

$$\frac{1}{n}\sum_{k\ =0}^{n-1} e^{2\pi ikh/n} = \begin{cases} 1 \textit{ if } h = 0 \mod n \\ 0 \textit{ otherwise} \end{cases}$$

We will also need an expression for the Fourier transform of the Bernoulli polynomial, given here without proof.

Theorem 11. *For* $x \in [0, 1]$

$$B_2(x) = \frac{1}{2\pi^2}\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{e^{2\pi ihx}}{h^2}$$

Armed with these results, we can prove

Lemma 12. *For all* $n > 1$ *and* $1 \leq k \leq n - 1$*,*

$$\frac{1}{n-1}\sum_{l=1}^{n-1}\bar{B}_2(kz/n) = \frac{-1}{6n}$$

Proof. Start with the Fourier transform of $\frac{2\pi^2}{n-1}\sum_{l=1}^{n-1}\bar{B}_2(kz/n)$.

$$\frac{2\pi^2}{n-1}\sum_{z_s\ =1}^{n-1}\bar{B}_2(x) = \frac{1}{n-1}\sum_{z_s=1}^{n-1}\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{e^{2\pi ihz/n}}{h^2}$$
$$= \sum_{\substack{h\in\mathbb{Z}\setminus\{0\} \\ h=0 \bmod n}}\frac{1}{h^2} - \frac{1}{n-1}\sum_{\substack{h\in\mathbb{Z}\setminus\{0\} \\ h\neq0 \bmod n}}\frac{1}{h^2}$$
$$= \frac{n}{n-1}\sum_{\substack{h\in\mathbb{Z}\setminus\{0\} \\ h=0 \bmod n}}\frac{1}{h^2} - \frac{1}{n-1}\sum_{h\in\mathbb{Z}\setminus\{0\}}\frac{1}{h^2}$$

As $\sum_{h=1}^{\infty} 1/h^2 = \pi^2/6$, this simplifies to $\frac{-2\pi^2}{6n}$ and $\frac{1}{n-1}\sum_{z_s=1}^{n-1} B_2(z_s/n) = \frac{-1}{6n}$. □

This will provide the base case for the inductive argument. For the inductive steps, we will need an auxiliary lemma about concave functions.

Lemma 13. *For* $f(x) = x^{\lambda}$ *when* $\lambda \leq 1$*,*

$$f\left(\sum_{i=1}^{n} x_i\right) \leq \sum_{i=1}^{n} f(x_i)$$

Proof. Clearly this holds when $n = 0$. Assume the inductive hypothesis that the claim is true for $n - 1$. Let $a = \sum_{i=1}^{n-1} x_i$ and $b = x_n$. Note that f' is decreasing because f is concave. This means that

$$\begin{aligned} f(a+b) - f(a) &= \int_a^{a+b} f'(x)dx \\ &= \int_0^b f'(x+a)dx \\ &\leq \int_0^b f'(x)dx \\ &= f(b) \end{aligned}$$

Therefore,

$$f\left(\sum_{i=1}^{n} x_i\right) \leq f\left(\sum_{i=1}^{n-1} x_i\right) + f(x_n)$$

Apply the inductive hypothesis for the result. □

6.3 CBC Error

Now we area ready for the main proof. Let $z_1^* \dots z_s^*$ be the components of the generating vector chosen by the CBC algorithm. We will show the following result:

Theorem 14. *For* $\lambda \in (\frac{1}{2}, 1]$

$$\mathcal{L}^\lambda(z_1^*, \dots z_s^*) \leq \frac{1}{n-1} \sum_{u \subseteq [s] \setminus \{0\}} \gamma_u^\lambda K(\lambda)^{|u|}$$

where $K(\lambda) = \frac{2\zeta(2\lambda)}{(2\pi^2)^\lambda}$ and the Zeta function $\zeta(x) = \sum_{h=1}^\infty 1/h^x$.

Proof. Returning to the base case of $s = 1$, we can rewrite the inequality from Lemma 9 to make it easy to apply Lemma 12.

$$\begin{aligned} \mathcal{L}(1) &= \frac{\gamma_1}{n}\left(B_2(0) + \frac{n-1}{n-1}\sum_{l=1}^{n-1} B_2(k/n)\right) \\ &= \frac{\gamma_1}{n}\left(\frac{n}{6n} - \frac{n-1}{6n}\right) \\ &= \frac{\gamma_1}{6n^2} \end{aligned}$$

When $\lambda > \frac{1}{2}$,

$$\begin{aligned} L^\lambda(1) &\leq \frac{1}{n}\left(\frac{\gamma_1}{6}\right)^\lambda \\ &\leq \frac{1}{n}(1 + K(\lambda)\gamma_1^\lambda) \end{aligned}$$

This proves the base case.

For general s, start by expressing the error in the sum-of-subsets representation of the kernel.

$$\begin{aligned} &\mathcal{L}(z_1, \dots z_s) \\ &\quad = -1 + \sum_{u \subseteq \{1:s\}} \prod_{j \in u} \gamma_u \left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j \in u} \bar{B}_2(kz_j/n)\right) \end{aligned}$$

Separate these terms into those over subsets containing s and those not containing s. The terms without s can be collapsed into $\mathcal{L}(z_1, \dots z_{s-1})$.

$$\mathcal{L}(z_1^*, \dots z_s^*) = \mathcal{L}(z_1^*, \dots z_{s-1}^*) + \theta$$

where $\theta = \sum_{u \subseteq \{1:s\}} \left(\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j \in u} \bar{B}_2(kz_j/n)\right)$.

Take the Fourier transform of B_2 in θ and apply Theorem 10. Here, z_u^* indicates the subvector of z^* given by the indices in u.

$$\begin{aligned} \theta &= \sum_{s \in u \subseteq \{1:s\}} \frac{\gamma_u}{(2\pi^2)^{|u|}} \left(\frac{1}{n}\sum_{k=0}^{n-1} \sum_{h \in (\mathbb{Z}-\{0\})^{|u|}} \frac{e^{2\pi i k h^T z_u^*/n}}{\prod_{j \in u} h_j^2}\right) \\ &= \sum_{s \in u \subseteq \{1:s\}} \frac{\gamma_u}{(2\pi^2)^{|u|}} \sum_{\substack{h \in (\mathbb{Z}-\{0\})^{|u|} \\ h^T z_u^* = 0 \bmod n}} \frac{1}{\prod_{j \in u} h_j^2} \\ &= \sum_{s \in u \subseteq \{1:s\}} \frac{\gamma_u}{(2\pi^2)^{|u|}} \sum_{h_1 \in \mathbb{Z} \setminus \{0\}} \frac{1}{h_1^2} \cdots \sum_{\substack{h_s \in \mathbb{Z}-\{0\} \\ h^T z_u^* = 0 \bmod n}} \frac{1}{h_s^2} \end{aligned}$$

Apply Theorem 13 recursively to find that

$$\begin{aligned} &\theta^\lambda \\ &\leq \sum_{s \in u \subseteq \{1:s\}} \left(\frac{\gamma_u}{(2\pi^2)^{|u|}}\right)^\lambda \sum_{h_1 \in \mathbb{Z} \setminus \{0\}} \frac{1}{h_1^{2\lambda}} \cdots \sum_{\substack{h_s \in \mathbb{Z}-\{0\} \\ h^T z_u^* = 0 \bmod n}} \frac{1}{h_s^{2\lambda}} \end{aligned}$$

As the CBC algorithm picks the value for z_s^* that minimizes the error, we know $\mathcal{L}(z_1^*, \dots z_s^*) \leq E_{z_s}[\mathcal{L}(z_1^*, \dots z_s)]$ (the average error over all possible values of z_s). Similarly, $\theta^\lambda \leq E_{z_s}[\theta^\lambda]$. Let the vector $z_u(z_s)$ be z_u^*, but with the last element replaced with z_s. By linearity of expectation, we can write

$$\begin{aligned} \theta^\lambda \leq &\sum_{s \in u \subseteq \{1:s\}} \left(\frac{\gamma_u}{(2\pi^2)^{|u|}}\right)^\lambda \\ &\sum_{h_1 \in \mathbb{Z} \setminus \{0\}} \frac{1}{h_1^{2\lambda}} \cdots \frac{1}{n-1}\sum_{z_s=1}^{n-1} \sum_{\substack{h_s \in \mathbb{Z}-\{0\} \\ h^T z_u(z_s) = 0 \bmod n}} \frac{1}{h_s^{2\lambda}} \end{aligned}$$

$$\leq \sum_{s\in u\subseteq\{1:s\}} \left(\frac{\gamma_u}{(2\pi^2)^{|u|}}\right)^{\lambda}$$
$$\sum_{h_1\in\mathbb{Z}\setminus\{0\}} \frac{1}{h_1^{2\lambda}} \cdots \frac{1}{n-1} 2 \sum_{c=1}^{\infty} \frac{1}{h_s^{2\lambda}}$$

Using the definition of $K(\lambda)$, this simplifies to

$$\theta \leq \sum_{u\subseteq\{1:s\}} \frac{1}{n-1} \gamma_u^{\lambda} K(\lambda)^{|u|}$$

Apply the induction hypothesis to get the claim.

7 The Baker's Transformation

While shift averaged lattice rules let the error decay as $O(n^{-1})$ in the number of samples, we can apply an additional function to the samples to bring the decay rate to $O(n^{-2})$.

$$\phi(t) := 1 - |2t-1|$$

This is known as the *baker's transformation*, as it mimics the way a baker stretches and folds bread back on itself.

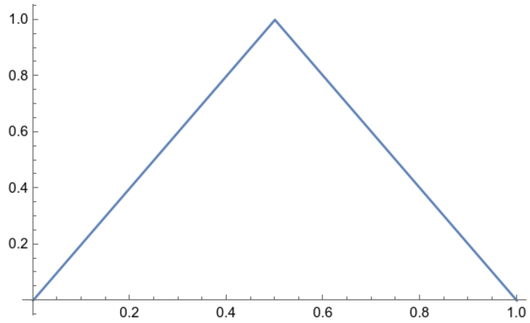

Figure 3: The Baker's Transformation folds each coordinate of a sample about the center of unit interval.

The shift averaged worst case error for samples that have undergone the Baker's transformation can be derived from a change of variables in Theorem 3 to give

$$-\int_{[0,1]^s}\int_{[0,1^s]} k(x,y)dx\,dy + \frac{1}{n^2}\sum_{i\in[n]}\sum_{j\in[n]} k_{\mathrm{b}}(x_i,y_j)$$

where $k_b(x,y) = \int k(\phi(x\oplus\Delta),\phi(y\oplus\Delta))\,d\Delta$.

A proof that this expression applied to lattice points leads to error decaying as $O(n^{-2})$ is beyond the scope of this introduction. Interested readers should turn to Hickernell 2002.

8 Computing Lattice Rules

The inner loop of the CBC algorithm in Section 2.1 requires computing the worst case squared error for each possible value of $z_s \in [1,n]$. From Theorem 9, this value is $-1+\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{s}(1+\gamma_j\bar{B}_2(kz_j/n))$. Naively, it seems like this computation requires $O(n^2 s)$ time. Fortunately, Nuyens and Cools 2006 describe a much faster approach.

To start, we will use a recursive definition of $L^2_{n,s}$ similar to the one derived in Theorem 14:

$$L^2_{n,s} = L^2_{n,s-1} + \frac{\gamma_s}{n}\sum_{k=0}^{n-1}\bar{B}_2(kz_s/n)\prod_{j=1}^{s}(1+\gamma_j\bar{B}_2(kz_j/n))$$

This can be more economically expressed as the inner product

$$L^2_{n,s} = L^2_{n,s-1} + \frac{\gamma_s}{n}\langle\Omega_z\, p^s\rangle$$

between n dimensional vectors Ω_z and p^s where

$$\Omega_{z,k} = B_2((kz \bmod n)/n)$$

$$p^s_k = \prod_{j=1}^{s-1}(1+\gamma_j B_2(kz_j/n \bmod 1))$$

An advantage of this vectorized representation is that we can represent the squared error in s dimensions for all $n-1$ possible values of z_s as the squared error in $s-1$ dimensions plus the matrix vector product Ωp^s.

For example, when $n=5$, if row i of Ω represents the error when $z_s = i$, then

$$\Omega = B_2\left(\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 1 & 3 \\ 0 & 3 & 1 & 4 & 2 \\ 0 & 4 & 3 & 2 & 1 \end{bmatrix}/n\right)$$

The order of the rows and columns of Ω is arbitrary. We can choose row i to represent the error when $z_s = 2^i$ in the multiplicative group of integers (mod n). Similarly, we can order columns so that $\Omega_{i,j} = B_2(2^{(i-j)} \bmod n/n)$ when $j < n$, and $\Omega_{i,n} = 0$ for all i. This makes a submatrix of Ω a circulant matrix.

When $n=5$, the z_s values for rows 1 through 5 in this permutation are $(2,4,1,3)$. This means

$$\Omega = B_2\left(\begin{bmatrix} 1 & 3 & 4 & 2 & 0 \\ 2 & 1 & 3 & 4 & 0 \\ 4 & 2 & 1 & 3 & 0 \\ 3 & 4 & 2 & 1 & 0 \end{bmatrix}/n\right)$$

Multiplication by a circulant matrix corresponds to discrete convolution (in our case, by the first column of Ω, cyclically extended in each direction). Convolution is multiplication in the Fourier domain, which takes linear time once p^s is in the Fourier basis. To put it in this basis, we only need to spend $O(n \log n)$ time taking the FFT. This means that the usual $O(n^2)$ matrix multiplication cost is unnecessary. The fast version of the CBC construction, therefore, goes as follows. Here, $\circ$ indicates elementwise product and $\hat{\Omega}$ indicates the filter against which multiplication by the submatrix of Ω implicitly convolves.

Algorithm 1 CBC Construction of $z \in \mathbb{Z}^s$

```
for each $d \in [s]$ do
    $l \leftarrow l + \frac{\gamma_s}{n}(\mathrm{FFT}^{-1}(\hat{\Omega} \circ \mathrm{FFT}(p[1\colon n-1])) + \frac{p[n-1]}{6})$
    $z_d \leftarrow \arg\min l$
    $p \leftarrow p \circ (1 + \gamma_d \Omega[\log_2 z_d \bmod n, :])$
end for
```

9 Conclusion

This short introduction to Quasi Monte Carlo methods gives you a taste both of their benefits — sample efficiency — and their drawbacks – error growing exponentially in $\sum_{d=1}^{s} \gamma_d$ – when it comes to approximating a distribution with a finite collection of samples. That this overview barely scratches the surface of the research area should come as no surprise. I have worked exclusively with unanchored Sobolev spaces here, but similar analysis is possible for anchored Sobolev spaces, like the one used in Section 4.1 to define star discrepancy. The lattice technique I described only used a single generator (producing a so called *rank 1* lattice). In general, lattices can be generated from multiple elements, and generalizing the CBC construction is necessary. I neglected to provide any kind of analysis for digital nets; the analysis is similar to what we showed for lattice rules, but uses *Walsh* functions, the Fourier basis's discontinuous cousin. To learn more, I highly recommend the second half of Dick et al. 2013, in addition to the other references listed below.

References

[1] Josef Dick et al. “High-dimensional integration: The quasi-Monte Carlo way”. en. In: *Acta Numerica* 22 (May 2013), pp. 133–288.

[2] Fred J. Hickernell. “Obtaining O(N- 2+) Convergence for Lattice Quadrature Rules”. en. In: *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Ed. by Kai-Tai Fang et al. Berlin, Heidelberg: Springer, 2002, pp. 274–289.

[3] Pierre L’Ecuyer. “Randomized Quasi-Monte Carlo: An Introduction for Practitioners”. en. In: *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Art B. Owen and Peter W. Glynn. Vol. 241. Series Title: Springer Proceedings in Mathematics & Statistics. Cham: Springer International Publishing, 2018, pp. 29–52.

[4] Pierre L’Ecuyer and Christiane Lemieux. “Recent Advances in Randomized Quasi-Monte Carlo Methods”. en. In: *Modeling Uncertainty*. Ed. by Moshe Dror et al. Vol. 46. Series Title: International Series in Operations Research & Management Science. New York, NY: Springer US, 2002, pp. 419–474.

[5] Qiang Liu et al. “A Kernelized Stein Discrepancy for Goodness-of-fit Tests”. en. In: *International Conference on Machine Learning*. ISSN: 1938-7228. PMLR, June 2016, pp. 276–284.

[6] Dirk Nuyens and Ronald Cools. “Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces”. en. In: *Mathematics of Computation* 75.254 (Jan. 2006), pp. 903–920.

[7] Spassimir Paskov and Joseph F. Traub. “Faster valuation of financial derivatives”. en. In: (1996).

[8] Ian Sloan and Henryk Wozniakowski. “Tractability of Integration in Non-periodic and Periodic Weighted Tensor Product Hilbert Spaces”. en. In: *Journal of Complexity* 18.2 (June 2002), pp. 479–499.