RELIABLE QUASI-MONTE CARLO WITH CONTROL VARIATES

BY

DA LI

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Applied Mathematics
in the Graduate College of the
Illinois Institute of Technology

Approved ________________________
Advisor

Chicago, Illinois
Aug 01

CHAPTER 1

INTRODUCTION

what are we going to do? This article try to combine reliable Quasi-Monte Carlo methods with control variates based on digital sequences.

Why this is a good idea? Control Variates is proved to be able to increase efficiency of MC. If we can work it out with QMC, it will further improve the speed of cubSobol algorithm. And this is especially useful to problems where we can easily find good control variates.

What's the challenge? As is already shown[ref], one problem is that optimal solution of beta is different between MC and QMC. Hence, we need to find a new way computing beta for QMC.

Outline In chapter 2 we brief introduce MC,CV and QMC. In chapter 3 we show the theory and algorithm of this method. In chapter 4 we do several numerical experiments.

CHAPTER 2

BACKGROUND

2.1 QMC

2.1.1 Digital Sequence

Talk about the whole idea briefly.

2.1.2 QMC

Introduce QMC.

2.2 Control Variates

2.2.1 A Brief Review

Control variates has been know as variance reduction technique used in Monte Carlo methods.In this part we will brief review some crucial idea of this methods so we can see what's the problem for using it with QMC.

Suppose we have the following integration approximation problem:

$$I = \int_{[0,1]^d} f(x)dx$$

If we use Monte-Carlo method, the estimator should be:

$$\hat{I}(f) = \frac{1}{n}\sum_{i=1}^{n} f(X_i), X_i \sim \mathcal{U}[0,1)^d$$

Now suppose we have a known function h and its value on the interval $\int h(x)dx = \theta$. We construct a new estimator as the following:

$$\hat{I}_{cv}(f) = \frac{1}{n}\sum_{i=1}^{n}\Big(f(X_i) - \beta_{mc}(g(X_i) - \theta)\Big) \quad s.t. X_i \sim \mathcal{U}[0,1)$$

We can easily see it's an unbiased estimator.($\mathbb{E}(\hat{I}_{cv}) = I$)

Now we want to pick the right β_{mc} such that make the estimation more efficient. Base on previous MC error estimating formula (**??**), we know its achievable by minimizing the variance of the estimator, which is:

$$\begin{aligned}
&\mathrm{Var}_{mc}(\hat{I}_{cv})\\
=&\frac{\mathbb{E}\big([f(X_i) - \beta_{mc}(g(X_i) - \theta) - I]^2\big)}{n}\\
=&\frac{\mathbb{E}\big([(f(X_i) - I)^2 - 2\beta_{mc}(f(X_i) - I)(g(X_i) - \theta) - \beta_{mc}^2(g(X_i) - \theta)^2]\big)}{n}\\
=&\frac{\mathrm{Var}(f(X_i) - 2\beta_{mc}\mathrm{Cov}(f(X_i), g(X_i)) + \beta_{mc}^2\mathrm{Var}(g(X_i))}{n}\\
=&\frac{\mathrm{Var}(g(X_i)(\beta_{mc} - \frac{\mathrm{Cov}(f(X_i),g(X_i))}{\mathrm{Var}(g(X_i))})^2 + \mathrm{Var}(f(X_i) - \frac{\mathrm{Cov}^2(f(X_i),g(X_i))}{\mathrm{Var}(g(X_i))}}{n}
\end{aligned}$$

then the optimal β_{mc} is given by: $\beta_{mc} = \frac{\mathrm{Cov}(f(X_i),g(X_i))}{\mathrm{Var}(g(X_i))}$

In which case the variance become:

$$\mathrm{Var}_{mc}(\hat{I}_{cv}) = \frac{\mathrm{Var}(f(X_i)}{n}[1 - \mathrm{corr}^2(f(X_i), g(X_i))]$$

2.2.2 Control Variates with QMC

Suppose $X_1, \ldots, X_n$ are generated by QMC rule, the estimator stays the same. We can prove it is still unbiased:

$$\mathbb{E}(\hat{I}_{cv}) = \mathbb{E}\Big(\frac{1}{n}\sum_{i=1}^{n}\Big(f(X_i) - \beta(g(X_i) - \theta)\Big) = I$$

However, it's not the same case for β_{rqmc} since we do not have I.I.D. for X_i this time

$$\begin{aligned}
\mathrm{Var}_{rqmc}(\hat{I}_{cv}) &\neq \frac{\mathbb{E}\big([f(X_i) - \beta_{mc}(g(X_i) - \theta) - I]^2\big)}{n} \\
&= \mathrm{Var}_{rqmc}\Big(\hat{I} - \beta_{rqmc}\hat{G}\Big) \quad , \hat{G} = \sum_{i=1}^{n}(g(X_i) - \theta) \\
\beta^*_{rqmc} &= \mathrm{Var}(\hat{G})^{-1}\mathrm{Cov}(\hat{G}, \hat{I})
\end{aligned}$$

2.3 Reliable Adaptive QMC with digital sequence

2.3.1 Idea of adaptive cubature algorithm

One practical problem for QMC method is that how to get the sample size big enough for a required error tolerance. The idea in work of Hickernell and Jimnez Rugama(2014) is to construct a QMC algorithm with reliable error estimation on digital sequence. Here we briefly summarize their results. The error of QMC method on digital sequence can be expressed in terms of Walsh coefficients of the integrand on certain cone conditions.

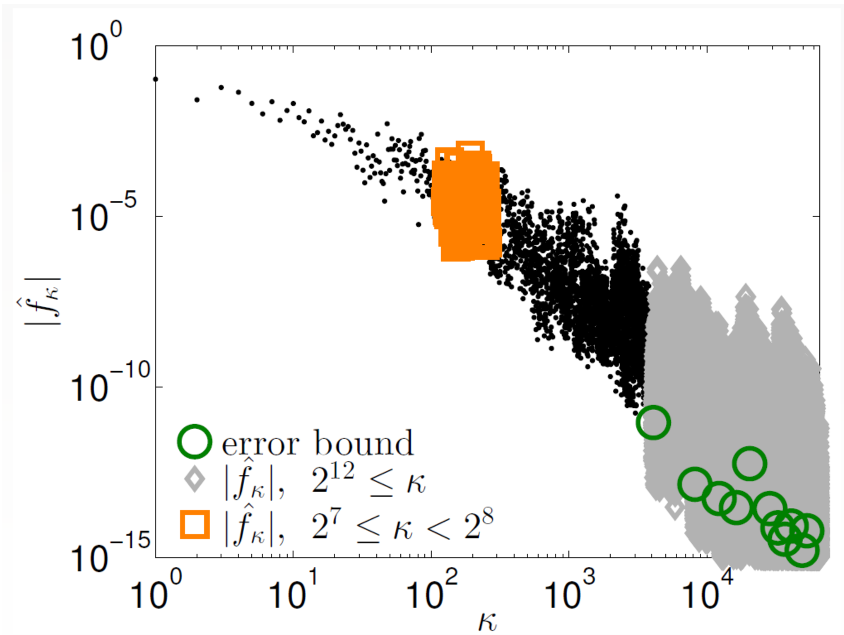
$$\begin{aligned}
&\Big|\int_{[0,1)^d} f(x)dx - \hat{I}_m(f)\Big| \leq a(r,m) \sum_{\lfloor 2^{m-r-1}\rfloor}^{2^{m-r}-1} |\hat{f}_{m,k}| \\
&\hat{I}_m(f) := \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(z_i \oplus \Delta) \\
&\hat{f}_{m,k} = \text{ discrete Fourier coefficients of } f \\
&a(r,m) = \text{ inflation factor that depends on } \mathcal{C}
\end{aligned}$$

Here is the cone condition.

$$\left| \int_{[0,1)^d} f(x)dx - \hat{I}_m(f) \right| \leq \sum \bigcirc \leq \sum \square \leq a(r,m) \sum_{\lfloor b^{m-r-1} \rfloor}^{b^{m-r}-1} |\hat{f}_{m,k}|$$

$$\bigcirc := \sum_{\lambda=1}^{\infty} |\hat{f}_{\lambda b^m}|, \quad \square := \sum_{\kappa=b^{l-1}}^{b^l-1} |\hat{f}_\kappa|, \quad \Diamond := \sum_{\kappa=b^m}^{\infty} |\hat{f}_\kappa|$$

$$\mathcal{C} := \left\{ \sum \bigcirc \leq \sum \Diamond \leq \sum \square \right\}$$



CHAPTER 3

RELIABLE ADAPTIVE QMC SOBOL WITH CV

3.1 Theory

3.1.1 Idea to add control variates to cubSobol

The idea is similar to traditional control variates technique for Monte-Carlo. If we know the integration of a function $\boldsymbol{h} = (h_1, \ldots, h_J)$ on the interval same as our f, say $\int_{[0,1)^d} h_j dx = \theta_j$, then we can define a new function g

$$\begin{aligned} & g := f - (\boldsymbol{h} - \boldsymbol{\theta})\boldsymbol{\beta} \\ & \text{s.t. } \boldsymbol{\theta} = (\theta_1, \ldots, \theta_J), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^T \end{aligned} \tag{3.1}$$

Then easily we can find that if we replace f with g, the integration stays the same.

$$\int_{[0,1)^d} g dx = \int_{[0,1)^d} f - (\boldsymbol{h} - \boldsymbol{\theta})\boldsymbol{\beta} dx = \int_{[0,1)^d} f dx$$

However, as we stated in chapter 3, we can not find optimal β by minimizing variance of estimator like Monte-Carlo. Instead we have a different way to estimate error with adaptive QMC method introduced in chapter 2.

Recall equation (**??**) our error bound for the new estimator of g still holds

$$\left| \int_{[0,1)^d} g dx - \hat{I}_m(g) \right| \leq a(r,m) \sum_{\lfloor 2^{m-r-1} \rfloor}^{2^{m-r}-1} |\hat{g}_{m,k}| \tag{3.2}$$

Naturally, the new estimator become

$$\hat{I}_m(g) := \frac{1}{b^m} \sum_{i=0}^{b^m-1} g(z_i + \Delta) \tag{3.3}$$

3.1.2 How to choose β?

From (3.2) it is clear that the optimal β is the one that minimize the error term.

$$
\begin{aligned}
\boldsymbol{\beta}^* &= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{g}_\kappa| && && (3.4)\\
&= \min_{\boldsymbol{\beta}} \sum_{\kappa=b^{m-r-1}}^{b^{m-r}-1} |\hat{f}_\kappa - (\hat{\boldsymbol{h}}_\kappa - \hat{\boldsymbol{\theta}})\boldsymbol{\beta}| && \hat{\boldsymbol{h}}_\kappa = (\hat{h}_{\kappa,1}, \ldots, \hat{h}_{\kappa,J}), \hat{\boldsymbol{\theta}} = (\hat{\theta}_{\kappa,1}, \ldots, \hat{\theta}_{\kappa,J}) && (3.5)\\
&= \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_1 && \hat{\boldsymbol{f}} = (\hat{f}_{b^{m-r-1}}, \ldots, \hat{f}_{b^{m-r}-1)})^T && (3.6)\\
&\approx \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_2 && \hat{\boldsymbol{H}} = (\hat{\boldsymbol{H}}_1, \ldots, \hat{\boldsymbol{H}}_J) && (3.7)\\
& && \hat{\boldsymbol{H}}_j = (\hat{h}_{b^{m-r-1},j} - \hat{\theta}_j, \ldots, \hat{h}_{b^{m-r}-1,j} - \hat{\theta}_j)^T
\end{aligned}
$$

The second equivalence is not hard to get, but the third one may not be so obvious. Let's consider it backwards. Suppose we have a vector A and it's $\mathcal{L}_1$-norm.

$$
A = \begin{pmatrix} f_1 - z_1 \\ f_2 - z_2 \\ \ldots \\ f_n - z_n \end{pmatrix}, \quad \|A\|_1 = \sum_{i=1}^{n} |f_i - z_i|, \quad z_i := (\boldsymbol{h}_i - \boldsymbol{\theta})
$$

If we replace the index,A is exactly what's inside the $\mathcal{L}_1$-norm in (3.6). Hence we justified the third equivalence. The reason we use an approximation instead, i.e. the $\mathcal{L}_1$-norm, is because there is no efficient way to solve it compared to existing least square methods.

3.1.3 Can we get rid of θ?

We noticed a problem in solution for optimal β, which is we do a lot of subtractions with θ. This could be a large cost when we have difficult functions which means b^{m-r}

could be very large number. Therefore we present a way to avoid that part.

The idea is form a observation that Walsh transform of θ in (**??**) is actually zero, since $\hat{h}_\kappa = \hat{g}_\kappa - \theta\delta_{\kappa,0}$ and the summation is not start from $\kappa = 0$.

This simplifies (3.6) to the following. Note that we only need the information of function f and h to calculate β^*, θ has been get rid of the optimization process.

$$\begin{aligned}
\boldsymbol{\beta}^* &= \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_1 & \hat{\boldsymbol{f}} &= (\hat{f}_{b^{m-r-1}}, \ldots, \hat{f}_{b^{m-r}-1)})^T & (3.8)\\
&\approx \min_{\boldsymbol{\beta}} \|\hat{\boldsymbol{f}} - \hat{\boldsymbol{H}}\boldsymbol{\beta}\|_2 & \hat{\boldsymbol{H}} &= (\hat{\boldsymbol{H}}_1, \ldots, \hat{\boldsymbol{H}}_J) & (3.9)\\
& & \hat{\boldsymbol{H}}_j &= (\hat{h}_{b^{m-r-1},j}, \ldots, \hat{h}_{b^{m-r}-1,j})^T &
\end{aligned}$$

The same problem happened with the estimator 3.3.We have the similar solution for that.

$$\begin{aligned}
\hat{I}_m(g) &= \frac{1}{b^m}\sum_{i=0}^{b^m-1} g(z_i + \Delta)\\
&= \frac{1}{b^m}\sum_{i=0}^{b^m-1} f(z_i + \Delta) - (\boldsymbol{h}(z_i + \Delta) - \boldsymbol{\theta})\boldsymbol{\beta}\\
&= \frac{1}{b^m}\sum_{i=0}^{b^m-1} [f(z_i + \Delta) - \boldsymbol{h}(z_i + \Delta)\boldsymbol{\beta}] + \boldsymbol{\theta}\boldsymbol{\beta} \qquad (3.10)
\end{aligned}$$

After organize it the in format of (3.10), θ is eliminated from the summation part. From these two parts of work on θ we managed to save $\frac{b-1}{b}b^{m-r} + b^m$ operations of subtraction.

3.2 Algorithm

Now combining the work from **??** and our previous work, we have the following algorithm for reliable adaptive QMC with control variates.

Algorithm 1: Reliable Adaptive QMC with control variates

Data: function f and $\boldsymbol{H}$; value of $\int_{[0,1)^d} h_j dx = \theta_j$; tolerance ε

Result: estimate of $\int_{[0,1)^d} f dx$; samples size; optimal β

begin

$m, r \longleftarrow$ start numbers

$x \longleftarrow$ grab 2^m sobolset points

1 $\tilde{\kappa} \longleftarrow$ get kappa map using **??**

2 $\hat{f}, \hat{\boldsymbol{H}} \longleftarrow$ get Walsh transform of $f, \boldsymbol{h}$ using **??**

3 $a \longleftarrow 2^{m-r-1}, b \longleftarrow 2^{m-r} - 1$

4 $\boldsymbol{\beta} \longleftarrow \hat{H}\big\{\tilde{\kappa}[x(a:b)]\big\} \backslash \hat{f}\big\{\tilde{\kappa}[x(a:b)]\big\}, \hat{g} \longleftarrow \hat{f} - \hat{\boldsymbol{H}}\boldsymbol{\beta}$

5 $\tilde{S}_{m-r,m}(g) \longleftarrow \sum_a^b \Big|\hat{g}\big\{\tilde{\kappa}[x(a:b)]\big\}\Big|,$

6 **if** $a(m,r)\tilde{S}_{m-r,m}(g) \leq \varepsilon$ **then**

  return $\hat{I}_m(g) = \sum_{i=0}^{2^m-1} f[x(i)] + \boldsymbol{\theta}\boldsymbol{\beta}$

  return $\boldsymbol{\beta}, n = 2^m$

7 **for** $m = m + 1 : mmax$ **do**

  $xnext \longleftarrow$ grab the next 2^{m-1} points in sobolset

  repeat step 3 with new m

  update kappa map(repeat step 1 with $(x : xnext)$)

  repeat step 2 on $(x : xnext)$

  repeat stet 4 with new $\tilde{\kappa}, a, b$

  repeat step 5 with updated $\tilde{\kappa}$

  repeat step 6 with new $\tilde{\kappa}, \boldsymbol{\beta}$

3.3 Theorem

CHAPTER 4

NUMERICAL EXPERIMENT

4.1 When beta is not accurate?

Examples to show for certain functions beta needs to be updated.

4.2 Option Pricing

Option Pricing has always been a challenging topic in financial mathematics. Add some reasons?

Hence it became a good application for our QMC algorithm. Here we are going to demonstrate several examples of pricing different options with ccontrol variates.

4.2.1 Asian Option

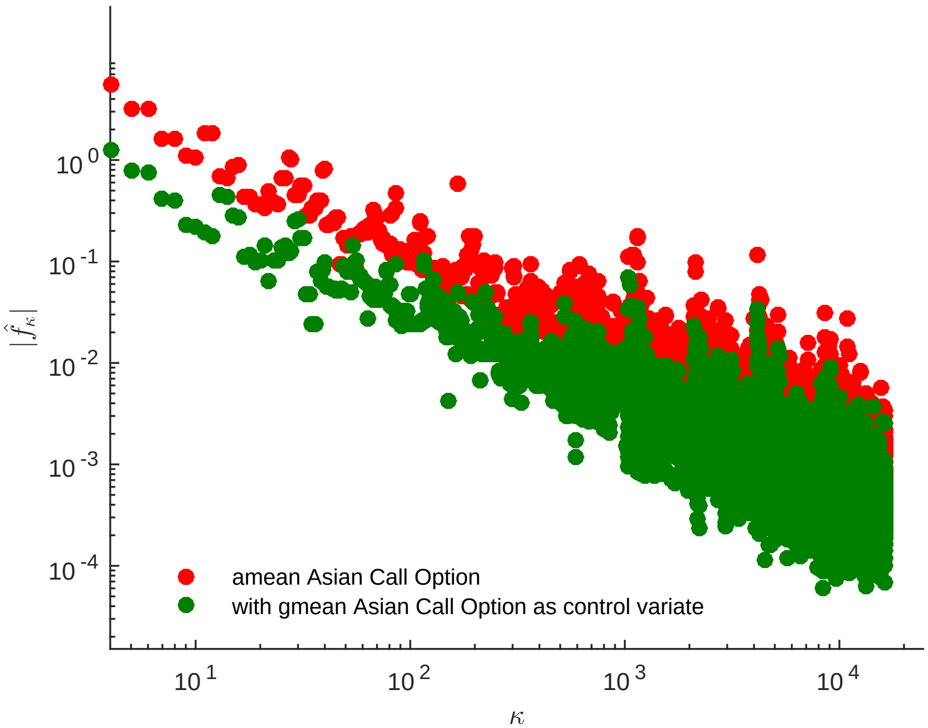
There are two types of asian options, depends on which types of mean you want to use.For this example we take arithematic mean asian call option as our target function, whose payoff function is:

$$C_T^{Amean} = \max\left(\frac{1}{d}\sum_{j=1}^{d} S(jT/d) - K, 0\right)$$

S0	K	TimeVector	r	volatility	abstol	reltol
120	130	1/52:1/52:16/52	0.01	0.5	1e-3	0

Table 4.1. Parameter Setup for Up and In Barrier Call Option

Sample Size			Time Cost		
cubSobol	cv_old	cv_new	cubSobol	cv_old	cv_new
65535	8192	9011	0.2783	0.1034	0.0673



4.2.2 Barrier Option

We will take up and in barrier call option as an example. Here is the payoff function for up and in barrier call option.

$$C_T^{U\&I} = (S_T - K)^+ 1_{\{\max S_t \geq Barrier\}}$$

From the payoff function it is naturally to consider european call option as control variates. Since if we take the barrier same as strike price, then this is just an european call option. Table ?? shows our setup for the barrier option.

S0	K	TimeVector	r	volatility	abstol	reltol
120	130	1/52:1/52:16/52	0.01	0.5	1e-3	0

Table 4.3. Parameter Setup for Up and In Barrier Call Option

Barrier	Sample Size			Time Cost		
	cubSobol	cv_old	cv_new	cubSobol	cv_old	cv_new
140	524288	78643	65535	1.874	0.5016	0.2743
135	524288	5802	6963	1.959	0.0781	0.0519
130	524288	1024	1024	1.876	0.0270	0.0199

Table 4.4. Results of cubSobol, cv_old and cv_new with Barrier Option

We then took three different barrier as listed in table ??, then we compared both oringinal cubSobol algorithm and the one with our modification as described in Chapter 4.

We can see from the results in table ??