Mathematical & Computational Finance II
Lecture Notes

Numerical Methods & Computational Finance

November 5 2015
Last update: December 4, 2017

1 Monte-Carlo Methods & Quasi Monte-Carlo Methods

Quasi Monte-Carlo methods (QMC) is extremely important for high dimensional problems. Instead of sampling/simulating random/pseudorandom numbers naively, the idea is to use a highly uniform point set (HUPS). Say we want to estimate μ as a function of some s-dimensional hypercube

$$\mu = \int_{[0,1)^s} f(\vec{u})\, du = \mathbb{E}\left[f(\vec{U})\right]$$

where $\vec{U}$ is a vector of $U \sim Unif(0,1)$ random variables. If $s = 2$ we have uniform points over the unit square and

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

However, we should note that using true random variables will produce some clumping of points in the unit square. Using the HUPS will be more diffuse across the box and cover the 2-dimensional square in a "more systematic" way. The key is that the empirical distribution induced by a point set P_n is closer to uniform than some true random uniform vector. Drawing some arbitrary box A in our unit square we want[1]

$$\mathbb{P}(A) \approx \mathrm{Vol}(A)$$

Consider all rectangular boxes in $[0,1)^s$ with a corner at the origin and $P_n =$ the HUPS. We count the fraction of points P_n in the box and we want the difference to be small. Taking the supremum over all boxes, let

$$D_n^* = \sup_{\vec{v}\in[0,1)^s} \left| \prod_{j=1}^{s} v_j - \frac{1}{n} \left| P_n \cap \prod_{j=1}^{n} [0, v_j) \right| \right|$$

[1] I'm not quite sure what we mean by this.

we say that D_n^* is the "starred discrepancy" and $\prod_{j=1}^s v_j$ is the volume of box size v_j in the j^{th} coordinate. Unless you have some real/clear structure this will be difficult to compute. The goal is to minimize D_n^*.

Definition 1. Given a sequence $\vec{u}_1, \vec{u}_2, ...$ in $[0,1)^s$ for which $P_n = \{\vec{u}_1, ...\vec{u}_n\}$ has $D_n^* \sim \mathcal{O}\left(\frac{(\log n)^s}{n}\right)$, then P_n is a low-discrepancy point set .

There's lots of ways to achieve $\mathcal{O}\left(\frac{(\log n)^s}{n}\right)$, but we should note that with random i.i.d. points

$$D_n^* \sim \mathcal{O}\left(\frac{\sqrt{\log\log n}}{n}\right)$$

We think of the HUPS as being "more" uniform (based on a specific definition of uniformity) than true uniform random points. The use of HUPS is for multivariate numerical integration problems where we wish to find a deterministic error bound for our problem. In true Monte-Carlo problems our error bounds are probabilistic.

1.1 Error Bounds

Consider

$$\hat{\mu}_{QMC} = \frac{1}{n}\sum_{i=1}^{n} f(\vec{u}_i)$$

to estimate

$$\mu = \int_{[0,1)^s} f(\vec{u})\, d\vec{u}$$

If f is of bounded variation then we can show that, letting $V(f)$ being the variation of f,

$$|\hat{\mu}_{QMC} - \mu| \leq D_n^* V(f)$$

This is good: We have an error of $\mathcal{O}\left(\frac{(\log n)^s}{n}\right)$ if D_n^* is a low-discrepancy point set. However, is this better than true Monte-Carlo? We have the Monte-Carlo error[2]

$$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \qquad \text{(note that this is a probabilistic error)}$$

But[3]

$$\frac{(\log n)^n}{n} = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

hence

$$\lim_{n\to\infty} \frac{\frac{(\log n)^s}{n}}{\frac{1}{\sqrt{n}}} = 0$$

[2]Where does this come from?
[3]Really?

But, if we have s large, say $s = 10$ then the convergence to $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ is slow. For $s = 10$ we would need need $n \geq 1.2144 \times 10^{39}$ to satisfy

$$\frac{(\log n)^1 0}{n} \leq \frac{1}{\sqrt{n}}$$

So, we have two problems

1. The conditions f must satisfy may be difficult to confirm.
2. Any asymptotic advantage of the deterministic estimator may take a while to kick in, for large s.

1.2 Randomized Quasi-Monte-Carlo

A solution to the two problems is to introduce randomized quasi-Monte-Carlo methods. We add some randomness to our point set P_n so that

1. We may compute error/variance estimators.
2. Improve the quality of P_n.

After adding some uniform noise to P_n we have a new point set $\tilde{P}_n$ so that

1. Each $\vec{U}_i \in \tilde{P}_n$ is $Unif\left([0,1)^2\right)$.
2. The HUPS property is preserved (i.e. points are still dependent on each other – "more" uniform than true uniform variates).

1.2.1 Cranley-Patterson (1976)

Example: "This is an example that you'd never use"

The idea is, given our point set $P_n = \{\vec{u}_1, ..., \vec{u_n}\}$, we have

$$\vec{u}_i + Unif\left([0,1)^s\right) \mod 1$$

More succinctly, letting $\vec{v} = Unif\left([0,1)^s\right)$,

$$\vec{u}_i + \vec{v}$$

where + represents elementwise addition. So

$$\begin{aligned}\vec{u}_i + \vec{v} \text{ mod } 1 &= \left((u_{i_1} + v_1) \text{ mod } 1, ..., (u_{i_s} + v_s \text{ mod } 1\right) \\ &=: \tilde{\vec{U}}_i\end{aligned}$$

Hence

$$\tilde{P}_n = \left\{\tilde{\vec{U}}_1, \tilde{\vec{U}}_2, ..., \tilde{\vec{U}}_n\right\}$$

Then, the randomized QMC estimator

$$\hat{\mu}_{RQMC} = \frac{1}{n}\sum_{i=1}^{n} f\left(\tilde{\vec{U}}_i\right)$$

is unbiased[4] and if we study its variance we can show that $\text{Var}\left[\hat{\mu}_{RQMC}\right] \leq \text{Var}\left[\hat{\mu}_{Crude}\right]$.

1.2.2 Comparison of Crude and RQMC Estimators

We have some theoretical results that permit us to meaningfully compare the variance of our RQMC estimator with the variance of the crude Monte-Carlo estimator, for some HUPS and randomizations.

1.2.3 Some Interesting HUPS

We have some noteworthy HUPS to think about:

1. Korobov Rule (1959)

   To generate our vector of points we pick some number a relatively prime to n. Then, take
   $$\vec{u}_i = \left[\frac{i}{n}\left(1, a, a^2 \bmod n, ...a^{s-1} \bmod n\right)\right] \bmod 1$$

2. Sobol Sequence

3. Halton Sequences

4. "Low Discrepancy Sequences"

1.3 Effective Dimensions

In practice, our problem may have some large "nominal" dimensionality s, but really only depends on a smaller subset of dimensions. QMC is particularly successful when a function f has large nominal dimensionality but small effective dimensionality. That this, there is some $d \in \mathbb{N}$ such that f can be well-approximated by a sum of d (or fewer) dimensional functions.

<u>Example</u>: Mortgages

Consider a function of 360 uniform variates (i.e. a mortgage with monthly payments over 30 years). Then,

$$f(u_1, ..., u_{360}) \approx u_1 + u_2 + ... + u_{360}$$

Here we have a 360-dimensional function approximated by a sum of 360 1-dimensional functions. That is, the effective dimensions of f is 1

[4]Proof left as an exercise to the reader.

Example: Asian Options

We may use a HUPS P_n to create a sample of n paths of a risky asset price that needs to be simulated to price some path-dependent derivative. One point, $\vec{u}_i$, corresponds to one path for an asset. We want to estimate

$$\mu = \mathbb{E}\left[e^{-rT} g_T\right]$$

for some payoff at time T, g_T. We want to rewrite this as an s-dimensional integration problem

$$\mu = \int_{[0,1)^s} f(\vec{u})\, d\vec{u}$$

So, for an Asian option under the Black-Scholes model, take $t_j = j\Delta, \Delta = \frac{T}{s}$, where $s =$ the number of monitoring points to compute the average. We have

$$g_T = \max\left\{0, \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right\}$$

and

$$S(t_j) = S_0 \exp\left\{\left(r - \frac{1}{2}\sigma^2\right) j\Delta + \sigma\sqrt{\Delta t}\,(Z_1 + ... + Z_j)\right\}$$

where $Z_j = \Phi^{-1}(u_j)$, and $u_1, ..., u_s \sim Unif(0,1)$ i.i.d. We can write the value of the call option

$$C_0^{Asian} = \int_{[0,1)^s} e^{-rT} \max\left\{0, \frac{1}{s}\sum_{j=1}^{s} S_0 \exp\left[\left(r - \frac{1}{2}\sigma^2\right) j\Delta + \sigma\sqrt{\Delta t}\left(\Phi^{-1}(u_1) + ... + \Phi^{-1}(u_j)\right)\right]\right\}$$

Now, suppose we use a HUPS

$$P_n = \{\vec{u}_1, ..., \vec{u}_n\}$$

and

$$\tilde{P}_{n,1}, ..., \tilde{P}_{n,m}$$

are m i.i.d. copies of a randomized version of P_n (i.e. generate m i.i.d. noise samples $\vec{v}_1, ..., \vec{v}_m$ from $Unif\left([0,1)^s\right)$). For brevity, let

$$\tilde{P}_{n,l} = \{\vec{w}_i = (\vec{u}_i + \vec{v}_l) \mod 1, l = 1, 2, ..., m\}$$

then the estimator

$$\hat{\mu}_l = \sum_{i=1}^{n} f(\vec{w}_i)$$

is the discount payoff estimator from a path generated using $\vec{w}_i$. We have the variance

$$\mathrm{Var}\left[\hat{\mu}_{RQMC}\right] = \frac{1}{m(m-1)}\sum_{l=1}^{m}\left(\hat{\mu}_l - \hat{\mu}_{RQMC}\right)^2$$

We may compare our variance with the variance from other techniques.[5]

[5]But we don't.