Monte Carlo Methods for Uncertainty Quantification

Mike Giles

Mathematical Institute, University of Oxford

ERCOFTAC course on Mathematical Methods and Tools in Uncertainty Management and Quantification

October 25, 2013

Lecture outline

Lecture 2: Variance reduction

- elementary manipulations
- control variates
- importance sampling
- stratified sampling
- Latin Hypercube
- randomised quasi-Monte Carlo

Elementary Manipulations

If a, b are random variables, and λ, μ are constants, then

$$\begin{aligned}
\mathbb{E}[a+\mu] &= \mathbb{E}[a]+\mu \\
\mathbb{V}[a+\mu] &= \mathbb{V}[a] \\
\mathbb{E}[\lambda\, a] &= \lambda\, \mathbb{E}[a] \\
\mathbb{V}[\lambda\, a] &= \lambda^2\, \mathbb{V}[a] \\
\mathbb{E}[a+b] &= \mathbb{E}[a]+\mathbb{E}[b]
\end{aligned}$$

where $$\mathbb{V}[a] \equiv \mathbb{E}\left[(a-\mathbb{E}[a])^2\right] = \mathbb{E}\left[a^2\right] - (\mathbb{E}[a])^2$$

Elementary Manipulations

In addition,

$$\mathbb{V}[a+b] = \mathbb{V}[a] + 2\,\mathrm{Cov}[a,b] + \mathbb{V}[b]$$

where

$$\mathrm{Cov}[a,b] \equiv \mathbb{E}\left[\,(a-\mathbb{E}[a])\,(b-\mathbb{E}[b])\right]$$

Since

$$|\mathrm{Cov}[a,b]| \;\leq\; \sqrt{\mathbb{V}[a]\;\mathbb{V}[b]}$$

it follows that

$$\begin{aligned} \mathbb{V}[a+b] &\leq \left(\sqrt{\mathbb{V}[a]}+\sqrt{\mathbb{V}[b]}\right)^2 \\ \Longrightarrow \quad \sqrt{\mathbb{V}[a+b]} &\leq \sqrt{\mathbb{V}[a]}+\sqrt{\mathbb{V}[b]} \end{aligned}$$

If a, b are independent then $\mathbb{V}[a+b] = \mathbb{V}[a] + \mathbb{V}[b]$, and more generally the variance of a sum of independents is equal to the sum of their variances.

Control Variates

Suppose we want to estimate $\mathbb{E}[f(X)]$, and there is another function $g(X)$ for which we know $\mathbb{E}[g(X)]$.

We can use this by defining a new estimator

$$\widehat{f} = \overline{f} - \lambda\,(\overline{g} - \mathbb{E}[g])$$

Again unbiased since $\mathbb{E}[\widehat{f}] = \mathbb{E}[\overline{f}] = \mathbb{E}[f]$

Control Variates

For a single sample,

$$\begin{aligned}\mathbb{V}[f - \lambda\,(g - \mathbb{E}[g])] &= \mathbb{V}[f - \lambda\, g] \\ &= \mathbb{V}[f] - 2\,\lambda\,\mathrm{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]\end{aligned}$$

For an average of N samples,

$$\mathbb{V}[\overline{f} - \lambda\,(\overline{g} - \mathbb{E}[g])] = N^{-1}\left(\mathbb{V}[f] - 2\,\lambda\,\mathrm{Cov}[f,g] + \lambda^2\,\mathbb{V}[g]\right)$$

To minimise this, the optimum value for λ is

$$\lambda = \frac{\mathrm{Cov}[f,g]}{\mathbb{V}[g]}$$

Control Variates

The resulting variance is

$$N^{-1}\ \mathbb{V}[f]\left(1-\frac{(\mathrm{Cov}[f,g])^2}{\mathbb{V}[f]\,\mathbb{V}[g]}\right) = N^{-1}\ \mathbb{V}[f]\left(1-\rho^2\right)$$

where $-1<\rho<1$ is the correlation between f and g.

The challenge is to choose a good g which is well correlated with f.
The covariance, and hence the optimal λ, can be estimated numerically.

Importance Sampling

Importance sampling involves a change of probability measure.

Instead of taking X from a distribution with p.d.f. $p_1(X)$, we instead take it from a different distribution with p.d.f. $p_2(X)$.

$$\begin{aligned}\mathbb{E}_1[f(X)] &= \int f(X)\; p_1(X)\; \mathrm{d}X \\ &= \int f(X)\; \frac{p_1(X)}{p_2(X)}\; p_2(X)\; \mathrm{d}X \\ &= \mathbb{E}_2[f(X)\; R(X)]\end{aligned}$$

where $R(X) = p_1(X)/p_2(X)$ is the Radon-Nikodym derivative.

Importance Sampling

We want the new variance $\mathbb{V}_2[f(X)\ R(X)]$ to be smaller than the old variance $\mathbb{V}_1[f(X)]$.

How do we achieve this? Ideal is to make $f(X)R(X)$ constant, so its variance is zero.

More practically, make $R(X)$ small where $f(X)$ is large, and make $R(X)$ large where $f(X)$ is small.

Small $R(X) \Longleftrightarrow$ large $p_2(X)$ relative to $p_1(X)$, so more random samples in region where $f(X)$ is large.

Particularly important for rare event simulation where $f(X)$ is zero almost everywhere.

Stratified Sampling

Idea is to achieve a less random sampling – given a 1D problem:

$$I = \int_0^1 f(x)\,\mathrm{d}x.$$

instead of taking N samples from a uniform distribution on $[0,1]$, we break the interval into M strata of equal width and take L samples from each.

Defining $X^{(ij)}$ to be the value of i^{th} sample from strata j,

$$\begin{aligned} \overline{f}_j &= L^{-1} \sum_i f(X^{(ij)}) = \text{average from strata } j, \\ \overline{f} &= M^{-1} \sum_j \overline{f}_j = \text{overall average} \end{aligned}$$

then $$\mathbb{E}[\overline{f}] = M^{-1} \sum_j \mathbb{E}[f(X) | X \in \text{strata } j] = \mathbb{E}[f(X)]$$

so the estimator is still unbiased.

Stratified Sampling

If we define

$$\sigma_j^2 = \mathbb{V}[f(X) \,|\, X \in \text{strata } j],$$

the variance is

$$\begin{aligned} \mathbb{V}[\overline{f}] = M^{-2} \sum_j \mathbb{V}[\overline{f}_j] &= M^{-2} L^{-1} \sum_j \sigma_j^2 \\ &= N^{-1} M^{-1} \sum_j \sigma_j^2 \end{aligned}$$

where $N = LM$ is the total number of samples.

Can prove this is less than usual variance, particularly when there is little variation within each stratum, but large differences between strata.

Stratified Sampling

For a d-dimensional application, can split each dimension of the $[0,1]^d$ hypercube into M strata producing M^d sub-cubes.

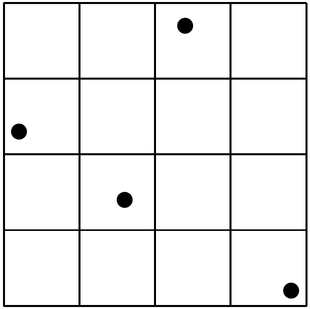
One generalisation of stratified sampling is to generate L points in each of these hypercubes

However, the total number of points is $L\,M^d$ which for large d would force M to be very small in practice.

Instead, use a method called Latin Hypercube sampling.

Latin Hypercube

Generate M points, dimension-by-dimension, using 1D stratified sampling with 1 value per stratum, assigning them randomly to the M points to give precisely one point in each stratum



Latin Hypercube

This gives one set of M points, with average

$$\overline{f} = M^{-1} \sum_{m=1}^{M} f(X^{(m)})$$

Since each of the points $X^{(m)}$ is uniformly distributed over the hypercube,

$$\mathbb{E}[\overline{f}] = \mathbb{E}[f]$$

The fact that the points are not independently generated does not affect the expectation, only the (reduced) variance.

Latin Hypercube

We now take L independently-generated set of points, each giving an average $\overline{f}_\ell$.

Averaging these

$$L^{-1} \sum_{\ell=1}^{L} \overline{f}_\ell$$

gives an unbiased estimate for $\mathbb{E}[f]$, and the empirical variance for $\overline{f}_\ell$ gives a confidence interval in the usual way.

Latin Hypercube

Note: in the special case in which the function $f(X)$ is a sum of one-dimensional functions:

$$f(X) = \sum_i f_i(X_i)$$

where X_i is the i^{th} component of X, then Latin Hypercube sampling reduces to 1D stratified sampling in each dimension.

In this case, potential for very large variance reduction by using large sample size M.

Much harder to analyse in general case.

Quasi Monte Carlo

Standard Monte Carlo approximates high-dimensional hypercube integral

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

by

$$\frac{1}{N}\sum_{n=1}^{N} f(x^{(i)})$$

with points chosen randomly, giving

- r.m.s. error proportional to $N^{-1/2}$
- confidence interval

Quasi Monte Carlo

Standard quasi Monte Carlo uses the same equal-weight estimator

$$\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

but chooses the points systematically so that

- error roughly proportional to N^{-1}
- no confidence interval

(We'll get the confidence interval back later by adding in some randomisation!)

Quasi-Monte Carlo

The key is to use points which are fairly uniformly spread within the hypercube, not clustered anywhere.

There is theory to prove that for certain point constructions, and certain function classes,

$$\text{Error} < C \, \frac{(\log N)^d}{N}$$

- for small dimension d, $(d < 10?)$ this is much better than $N^{-1/2}$ r.m.s. error for standard MC
- for large dimension d, $(\log N)^d$ could be enormous, so not clear there is any benefit

Rank-1 Lattice Rule

A rank-1 lattice rule has the simple construction

$$x^{(i)} = \frac{i}{N}\, z \quad \text{mod } 1$$

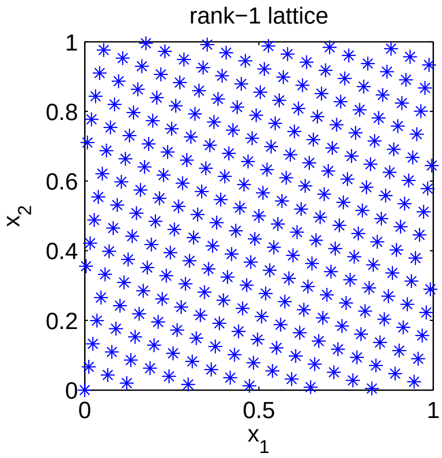
where z is a special d-dimensional "generating vector" with integer components co-prime with N (i.e. GCF is 1) and $r \mod 1$ means dropping the integer part of r

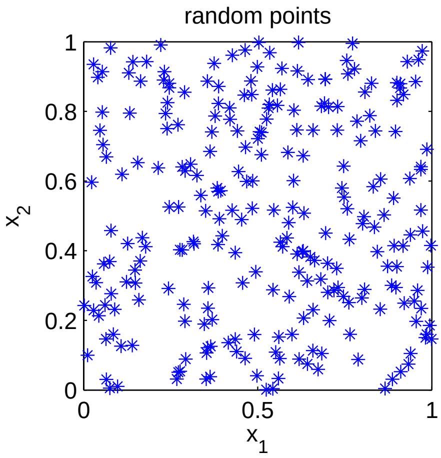
In each dimension k, the values $x_k^{(i)}$ are a permutation of the equally spaced points $0, 1/N, 2/N \ldots (N-1)/N$ which is great for integrands f which vary only in one dimension.

Also very good if $\quad f(x) = \sum_k f_k(x_k).$

Rank-1 Lattice Rule

Two dimensions: 256 points





Sobol Sequences

Sobol sequences $x^{(i)}$ have the property that for small dimensions $d < 40$ the subsequence $2^m \leq i < 2^{m+1}$ has precisely 2^{m-d} points in each sub-unit formed by d bisections of the original hypercube.

For example:

- cutting it into halves in any dimension, each has 2^{m-1} points
- cutting it into quarters in any dimension, each has 2^{m-2} points
- cutting it into halves in one direction, then halves in another direction, each quarter has 2^{m-2} points
- etc.

The generation of these sequences is a bit complicated, but it is fast and plenty of software is available to do it. MATLAB has `sobolset` as part of the Statistics toolbox.

Randomised QMC

In the best cases, QMC error is $O(N^{-1})$ instead of $O(N^{-1/2})$ but without a confidence interval.

To get a confidence interval using a rank-1 lattice rule, we use several sets of QMC points, with the N points in set m defined by

$$x^{(i,m)} = \left(\frac{i}{N}\, z \ + \ X^{(m)} \right) \quad \text{mod } 1$$

where $X^{(m)}$ is a random offset vector, uniformly distributed in $[0,1]^d$

Randomised QMC

For each m, let

$$\overline{f}_m = \frac{1}{N} \sum_{i=1}^{N} f(x^{(i,m)})$$

This is a random variable, and since $\mathbb{E}[f(x^{(i,m)})] = \mathbb{E}[f]$ it follows that $\mathbb{E}[\overline{f}_m] = \mathbb{E}[f]$

By using multiple sets, we can estimate $\mathbb{V}[\overline{f}]$ in the usual way and so get a confidence interval

More sets $\Longrightarrow$ better variance estimate, but poorer error.

Some people use as few as 10 sets, but I prefer 32.

Randomised QMC

For Sobol sequences, randomisation is achieved through digital scrambling:

$$x^{(i,m)} = x^{(i)} \veebar X^{(m)}$$

where the exclusive-or operation $\veebar$ is applied bitwise so that

$$\begin{array}{rl} & 0.1010011 \\ \veebar & 0.0110110 \\ = & 0.1100101 \end{array}$$

The benefit of the digital scrambling is that it maintains the special properties of the Sobol sequence.

MATLAB's `sobolset` supports digital scrambling.

Dominant Dimensions

QMC points have the property that the points are more uniformly distributed through the lowest dimensions. Consequently, important to think about how the dimensions are allocated to the problem.

Previously, have generated correlated Normals through $Y = L\,X$ with X i.i.d. $N(0,1)$ Normals.

For Monte Carlo, Y's have same distribution for any L such that $LL^T = \Sigma$, but for QMC different L's are equivalent to a change of coordinates and it can make a big difference.

Usually best to use a PCA construction $L = U\,\Lambda^{1/2}$ with eigenvalues arranged in descending order, from largest ($\Longrightarrow$ most important?) to smallest.

Final comments

- Control variates can sometimes be very useful – needs good insight to find a suitable control variate
- Importance sampling is very useful when the main contribution to the expectation comes from rare extreme events
- Stratified sampling is very effective in 1D, but not so clear how to use it in multiple dimensions
- Latin Hypercube is one generalisation – particularly effective when function can be almost decomposed into a sum of 1D functions

Final words

- quasi-Monte Carlo can give a much lower error than standard MC; $O(N^{-1})$ in best cases, instead of $O(N^{-1/2})$
- randomised QMC is important to regain confidence interval
- correct selection of dominant dimensions can also be important
- Hard to predict which variance reduction approach will be most effective
- Advice: when facing a new class of applications, try each one, and don't forget you can sometimes combine different techniques (e.g. stratified sampling with antithetic variables, or Latin Hypercube with importance sampling)