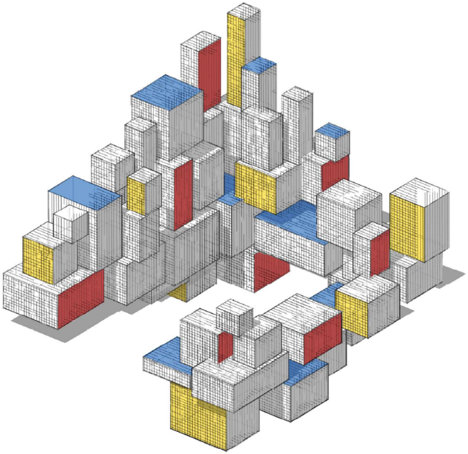
Variance Minimization



Purpose

In this lecture we discuss methods to main techniques for increasing the efficiency of Monte Carlo estimators:

- Control Variables
- Importance Sampling

Variance Reduction

Monte Carlo simulation can be made more efficient by utilizing known information about the simulation model.

Variance reduction techniques include:

- Antithetic variables
- Control variables
- Importance sampling
- Conditional Monte Carlo
- Stratified sampling

Suppose Y is the output of a simulation experiment. A random variable $\widetilde{Y}$, obtained from the same simulation run, is called a control variable for Y if Y and $\widetilde{Y}$ are correlated (negatively or positively) and the expectation of $\widetilde{Y}$ is known. The use of control variables for variance reduction is based on the following theorem.

Control Variables

Theorem: Control Variable Estimation

Let $Y_1, \ldots, Y_N$ be the output of N independent simulation runs and let $\widetilde{Y}_1, \ldots, \widetilde{Y}_N$ be the corresponding control variables, with $\mathbb{E}\widetilde{Y}_k = \widetilde{\mu}$ known. Let $\varrho_{Y,\widetilde{Y}}$ be the correlation coefficient between each Y_k and $\widetilde{Y}_k$. For each $\alpha \in \mathbb{R}$ the estimator

$$\widehat{\mu}^{(c)} = \frac{1}{N} \sum_{k=1}^{N} \left[Y_k - \alpha \left(\widetilde{Y}_k - \widetilde{\mu} \right) \right] \tag{1}$$

is an unbiased estimator for $\mu = \mathbb{E}Y$. The minimal variance of $\widehat{\mu}^{(c)}$ is

$$\mathbb{V}\mathrm{ar}\, \widehat{\mu}^{(c)} = \frac{1}{N} (1 - \varrho^2_{Y,\widetilde{Y}})\, \mathbb{V}\mathrm{ar}\, Y, \tag{2}$$

which is obtained for $\alpha = \varrho_{Y,\widetilde{Y}} \sqrt{\mathbb{V}\mathrm{ar} Y / \mathbb{V}\mathrm{ar} \widetilde{Y}}$.

Control Variables

From (2) we see that, by using the optimal α in (1), the variance of the control variate estimator is a factor $1 - \varrho^2_{Y,\widetilde{Y}}$ smaller than the variance of the crude Monte Carlo estimator.

Thus, if $\widetilde{Y}$ is highly correlated with Y, a significant variance reduction can be achieved.

The optimal α is usually unknown, but it can be easily estimated from the sample covariance matrix of $\{(Y_k, \widetilde{Y}_k)\}$.

Monte Carlo Integration (cont.)

We estimate the multiple integral

$$\mu = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sqrt{|x_1 + x_2 + x_3|}\, e^{-(x_1^2+x_2^2+x_3^2)/2}\, dx_1\, dx_2\, dx_3$$

using control variables. Defining $Y = |X_1 + X_2 + X_3|^{1/2}(2\pi)^{3/2}$, with $X_1, X_2, X_3 \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$, we can write $\mu = \mathbb{E}Y$.

The random variable Y is positively correlated with the random variable $\widetilde{Y} = X_1^2 + X_2^2 + X_3^2$, for the same choice of $X_1, X_2, X_3 \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. As $\mathbb{E}\widetilde{Y} = \mathbb{V}\text{ar}(X_1 + X_2 + X_3) = 3$, we can use it as a control variable to estimate the expectation of Y.

Monte Carlo Integration with Control Variables

The following Python program imports the earlier crude Monte Carlo sampling code.

mcintCV.py

```python
from mcint import *

Yc = np.sum(x**2, axis=1) # control variable data
yc = 3  # true expectation of control variable
C = np.cov(y,Yc) # sample covariance matrix
cor = C[0][1]/np.sqrt(C[0][0]*C[1][1])
alpha = C[0][1]/C[1][1]

est = np.mean(y-alpha*(Yc-yc))
RECV = np.sqrt((1-cor**2)*C[0][0]/N)/est  #relative error

print('Estimate = {:3.3f}, CI = ({:3.3f},{:3.3f}), Corr = {:3.3f}'.
      format(est, est*(1-z*RECV), est*(1+z*RECV),cor))
```

```
Estimate = 17.045, CI = (17.032,17.057), Corr = 0.480
```

Monte Carlo Integration with Control Variables

A typical estimate of the correlation coefficient $\varrho_{Y,\widetilde{Y}}$ is 0.48, which gives a reduction of the variance with a factor $1 - 0.48^2 \approx 0.77$ — a simulation speed-up of 23% compared with crude Monte Carlo.

Although the gain is small in this case, due to the modest correlation between Y and $\widetilde{Y}$, little extra work was required to achieve this variance reduction.

Importance Sampling

One of the most important variance reduction techniques is importance sampling.

This technique is especially useful for the estimation of very small probabilities.

The standard setting is the estimation of a quantity

$$\mu = \mathbb{E}_f H(\boldsymbol{X}) = \int H(\boldsymbol{x})\, f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x},$$

where H is a real-valued function and f the probability density of a random vector $\boldsymbol{X}$, called the nominal pdf. The subscript f is added to the expectation operator to indicate that it is taken with respect to the density f.

Importance Sampling

Let g be another probability density such that $g(\boldsymbol{x}) = 0$ implies that $H(\boldsymbol{x})\, f(\boldsymbol{x}) = 0$. Using the density g we can represent μ as

$$\mu = \int H(\boldsymbol{x})\, \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}\, g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \mathbb{E}_g \left[H(\boldsymbol{X})\, \frac{f(\boldsymbol{X})}{g(\boldsymbol{X})} \right].$$

Consequently, if $\boldsymbol{X}_1, \dots, \boldsymbol{X}_N \sim_{\text{iid}} g$, then

$$\widehat{\mu} = \frac{1}{N} \sum_{k=1}^{N} H(\boldsymbol{X}_k)\, \frac{f(\boldsymbol{X}_k)}{g(\boldsymbol{X}_k)}$$

is an unbiased estimator of μ. This estimator is called the importance sampling estimator and g is called the importance sampling density. The ratio of densities, $f(\boldsymbol{x})/g(\boldsymbol{x})$, is called the likelihood ratio.

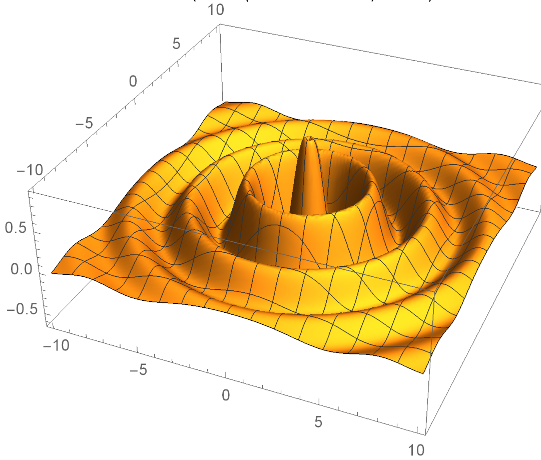
Algorithm 1: Importance Sampling Estimation

```
input: Function $H$, importance sampling density $g$ such that
       $g(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ for which $H(\boldsymbol{x}) f(\boldsymbol{x}) = 0$, sample size $N$,
       confidence level $1 - \alpha$.
output: Point estimate and approximate $(1 - \alpha)$ confidence
        interval for $\mu = \mathbb{E} H(\boldsymbol{X})$, where $\boldsymbol{X} \sim f$.
1 Simulate $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N \overset{\text{iid}}{\sim} g$ and let
        $Y_i = H(\boldsymbol{X}_i) f(\boldsymbol{X}_i) / g(\boldsymbol{X}_i), \ i = 1, \ldots, N.$
2 Estimate $\mu$ via $\widehat{\mu} = \overline{Y}$ and determine an approximate $(1 - \alpha)$
  confidence interval as
        $\mathcal{I} := \left( \widehat{\mu} - z_{1-\alpha/2} \frac{S}{\sqrt{N}}, \ \widehat{\mu} + z_{1-\alpha/2} \frac{S}{\sqrt{N}} \right),$
  where $z_\gamma$ denotes the $\gamma$-quantile of the $\mathcal{N}(0, 1)$ distribution and $S$
  is the sample standard deviation of $Y_1, \ldots, Y_N$.
3 return $\widehat{\mu}$ and the interval $\mathcal{I}$.
```

Example: Area Estimation via Importance Sampling

Let us examine the workings of importance sampling by estimating the area, μ say, under the graph of the function

$$M(x_1, x_2) = \mathrm{e}^{-\frac{1}{4}\sqrt{x_1^2+x_2^2}}\left(\sin\left(2\sqrt{x_1^2+x_2^2}\right)+1\right), \quad (x_1, x_2) \in \mathbb{R}^2.$$



Crude Monte Carlo

A natural approach to estimate the area is to truncate the domain to the square $[-b, b]^2$, for large enough b, and to estimate the integral

$$\mu_b = \int_{-b}^{b}\int_{-b}^{b} \underbrace{(2b)^2 M(\boldsymbol{x})}_{H(\boldsymbol{x})} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}_f H(\boldsymbol{X})$$

via crude Monte Carlo, where $f(\boldsymbol{x}) = 1/(2b)^2, \boldsymbol{x} \in [-b, b]^2$, is the pdf of the uniform distribution on $[-b, b]^2$.

Following is the Python code which does just that.

impsamp1.py

```
import numpy as np
from numpy import exp, sqrt, sin, pi, log, cos
from numpy.random import rand

b = 1000
H = lambda x1, x2: (2*b)**2 * exp(-sqrt(x1**2+x2**2)/4)*(sin(2*sqrt(
        x1**2+x2**2))+1)*(x1**2 + x2**2 < b**2)
f = 1/((2*b)**2)
N = 10**6
X1 = -b + 2*b*rand(N,1)
X2 = -b + 2*b*rand(N,1)
Z = H(X1,X2)
estCMC = np.mean(Z).item()  # to obtain scalar
RECMC = np.std(Z)/estCMC/sqrt(N).item()
print('CI = ({:3.3f},{:3.3f}), RE = {: 3.3f}'.format(estCMC*(1-1.96*
     RECMC), estCMC*(1+1.96*RECMC),RECMC))
```

```
CI = (82.663,135.036), RE =  0.123
```

Crude Monte Carlo

For a truncation level of $b = 1000$ and a sample size of $N = 10^6$, a typical estimate is 108.8, with an estimated relative error of 0.123.

We have two sources of error here:

- The first is the error in approximating μ by μ_b. However, as the function H decays exponentially fast, $b = 1000$ is more than enough to ensure this error is negligible.
- The second type of error is the statistical error, due to the estimation process itself. This can be quantified by the estimated relative error, and can be reduced by increasing the sample size.

Importance Sampling

Let us now consider an importance sampling approach in which the importance sampling pdf g is radially symmetric and decays exponentially in the radius, similar to the function H.

In particular, we simulate (X_1, X_2) by first generating a radius $R \sim \mathsf{Exp}(\lambda)$ and an angle $\Theta \sim \mathcal{U}(0, 2\pi)$, and then returning $X_1 = R\cos(\Theta)$ and $X_2 = R\sin(\Theta)$.

By the Transformation Rule of probability, we then have

$$g(\boldsymbol{x}) = f_{R,\Theta}(r,\theta)\frac{1}{r} = \lambda\,\mathrm{e}^{-\lambda r}\,\frac{1}{2\pi}\,\frac{1}{r} = \frac{\lambda \mathrm{e}^{-\lambda\sqrt{x_1^2+x_2^2}}}{2\pi\sqrt{x_1^2+x_2^2}}, \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \{\boldsymbol{0}\}.$$

The following code, which imports the one given above, implements the importance sampling steps, using the parameter $\lambda = 0.1$.

impsamp2.py

```python
from impsamp1 import *

lam = 0.1;
g = lambda x1, x2: lam*exp(-sqrt(x1**2 + x2**2)*lam)/sqrt(x1**2 + x2
     **2)/(2*pi);
U = rand(N,1); V = rand(N,1)
R = -log(U)/lam
X1 = R*cos(2*pi*V)
X2 = R*sin(2*pi*V)
Z = H(X1,X2)*f/g(X1,X2)
estIS = np.mean(Z).item()  # obtain scalar
REIS = np.std(Z)/estIS/sqrt(N).item()
print('CI = ({:3.3f},{:3.3f}), RE = {: 3.3f}'.format(estIS*(1-1.96*REIS
     ), estIS*(1+1.96*REIS),REIS))
```

```
CI = (100.723,101.077), RE =  0.001
```

Importance Sampling

A typical estimate is 100.90 with an estimated relative error of $1 \cdot 10^{-4}$, which gives a substantial variance reduction.

In terms of approximate 95% confidence intervals, we have (82.7,135.0) in the CMC case versus (100.7,101.1) in the importance sampling case.

Of course, we could have reduced the truncation level b to improve the performance of CMC, but then the approximation error might become more significant.

For the importance sampling case, the relative error is hardly affected by the threshold level, but does depend on the choice of λ. We chose λ such that the decay rate is slower than the decay rate of the function H, which is 0.25.

Optimal Importance Sampling Pdf

A main issue in importance sampling is how to choose the importance sampling distribution. A poor choice of g may seriously affect the accuracy of both the estimate and the confidence interval.

The theoretically optimal choice g^* for the importance sampling density minimizes the variance of $\widehat{\mu}$ and is therefore the solution to the functional minimization program

$$\min_g \mathbb{V}\mathrm{ar}_g \left(H(\boldsymbol{X}) \frac{f(\boldsymbol{X})}{g(\boldsymbol{X})} \right).$$

If either $H(\boldsymbol{x}) \geqslant 0$ or $H(\boldsymbol{x}) \leqslant 0$ for all $\boldsymbol{x}$, then the optimal importance sampling pdf is

$$g^*(\boldsymbol{x}) = \frac{H(\boldsymbol{x})\, f(\boldsymbol{x})}{\mu}.$$

Namely, then $\mathbb{V}\mathrm{ar}_{g^*}\widehat{\mu} = \mathbb{V}\mathrm{ar}_{g^*}(H(\boldsymbol{X}) f(\boldsymbol{X})/g(\boldsymbol{X})) = \mathbb{V}\mathrm{ar}_{g^*}\mu = 0$, so that the estimator $\widehat{\mu}$ is *constant* under g^*.

Choosing a Good Importance Sampling Pdf

An obvious difficulty is that the evaluation of the optimal importance sampling density g^* is usually not possible, since $g^*(\boldsymbol{x})$ depends on the unknown quantity μ.

Nevertheless, one can typically choose a good importance sampling density g "close" to the minimum variance density g^*.

One of the main considerations for choosing a good importance sampling pdf is that the IS estimator should have finite variance. This is equivalent to the requirement that

$$\mathbb{E}_g\left[H^2(\boldsymbol{X})\frac{f^2(\boldsymbol{X})}{g^2(\boldsymbol{X})}\right] = \mathbb{E}_f\left[H^2(\boldsymbol{X})\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}\right] < \infty.$$

This suggests that g should not have lighter tails than f and that, preferably, the likelihood ratio, f/g, should be bounded.