16

MONTE CARLO METHODS

16.1 Generating random variables

Assumption: we are able to generate random variables with uniform distribution $U([0,1])$.

16.1.1 Inverse transform method

Suppose we explicitly know the distribution, for example cdf $F(x)$ and for the target random variable X. Then we can generate a random sample $Z \sim U(0,1)$ and then apply the inverse transform $X = F^{-1}(Z)$ to get a sample whose cdf is $F(x)$.

Note the F^{-1} only exist for strictly increasing F. If $F(x)$ is not strictly increasing, then we define $F^{-1}(u) = \inf\{F^{-1}(x > u)\}$.

The procedures of inverse transform method is given by

Methodology 16.1.1 (inverse transform method). *Assume the random variable we want to generate has a cdf $F(x)$ strictly increasing. Then we have*

1. *Generate $X \sim U([0,1])$;*
2. *Set $Y = F^{-1}(x)$;*

then Y has cdf $F(x)$.

The working mechanism of the inverse transform method is explained in the following Lemma.

Lemma 16.1.1 (correctness of inverse transform method). *Let X be a random variable with cdf F_X. Let $U \sim U([0,1])$ be an uniform random variable. Then the transformed random variable $W = F_X^{-1}(U)$ has the same distribution cdf of F_X.*

Proof. Note that

$$F_W(w) = Pr(F_X^{-1}(U) < w) = Pr(U < F_X(w)) = F_X(w).$$

□

Example 16.1.1 (generating random sample from Laplace distribution). Given a random variable U drawn from the uniform distribution in the interval $(0,1]$, the random variable

$$X = \mu - b\ sgn(U - 0.5) \ln(1 - 2|U - 0.5|)$$

will have Laplace distribution with parameter (μ, b).

Note that we have used the fact that the inverse cdf of Laplace distribution is given by [Lemma 12.1.12]:

$$F^{-1}(p) = \mu - b \cdot sgn(p - 0.5) \ln(1 - 2|p - 0.5|).$$

We can further generalize the inverse transform method to transform random samples from distributions besides uniform distribution to random samples with target distribution.

Lemma 16.1.2 (generalized inverse transform method). *Let X be a random variable with cdf F_X. We can use the following ways to sample X:*

- *Let $U \sim U([0,1])$ be an uniform random variable. Then the transformed random variable $W = F_X^{-1}(U)$ has the same distribution of X.*
- *Let $Z \sim N(0,1)$ be a standard normal random variable with cdf ϕ. Then transformed random variable $W = F_X^{-1}(\phi(Z))$ has the same distribution of X.*
- *Let Y be a normal random variable with cdf F_Y. Then transformed random variable $W = F_X^{-1}(F_Y(Y))$ has the same distribution of X.*

Proof. (1)

$$F_W(w) = Pr(F_X^{-1}(U) < w) = Pr(U < F_X(w)) = F_X(w).$$

(2)

$$\begin{aligned} F_W(w) &= Pr(F_X^{-1}(\phi(Z)) < w) \\ &= Pr(\phi(Z) < F_X(w)) \\ &= Pr(Z < \phi^{-1}(F_X(w))) \\ &= \phi \circ \phi^{-1}(F_X(w))) \\ &= F_X(w) \end{aligned}$$

(3)

$$\begin{aligned} F_W(w) &= Pr(F_X^{-1}(F_Y(Y)) < w) \\ &= Pr(F_Y(Y) < F_X(w)) \\ &= Pr(Y < F_Y^{-1}(F_X(w))) \\ &= F_Y \circ F_Y^{-1}(F_X(w))) \\ &= F_X(w) \end{aligned}$$

□

Remark 16.1.1 (Box-Muller transformation method for generating standard normal). To generate standard normal variable, we can use Box-Muller transformation, see Lemma 16.1.3.

16.1.2 Box-Muller method for standard normal random variable

Note that in practice, we do not use inverse transform method for standard normal random variables.

Lemma 16.1.3 (Box Muller method for standard normal random variable). *Suppose we have U_1, U_2 being independent uniform on $[0,1]$. Define*

$$X = \sqrt{-2\ln(1-U_2)}\cos(2\pi U_1), Y = \sqrt{-2\ln(1-U_2)}\sin(2\pi U_1).$$

Then $X, Y \sim N(0,1)$ and X, Y be independent.

Proof. See Lemma 12.1.17. □

16.1.3 Acceptance-rejection method

The inverse transform method requires an explicit formula for F^{-1} which might not always be possible or simple. In this section, we look at another popular method, known as the Acceptance-Rejection method, which can be combined with inverse transform method to efficiently generate random numbers with complex distributions.

The basic idea in Acceptance-Rejection method is to find an alternative proposal probability distribution $q(x)$. $q(x)$ are selected from the ones we already have an efficient algorithm for generating random numbers (e.g., inverse transform method) and also $q(x)$ 'similar' to $f(x)$.

The algorithm is summarized below.

Algorithm 24: Accept-Reject algorithm for random number generation.

Input: the proposal distribution $q(x)$, the target distribution $p(x)$, and $M = \sup_x p(x)/q(x)$

1 Sample $x \sim q(x)$ and $u \sim U([0,1])$

2 If $u < \frac{p(x)}{Mq(x)}$, then accept x, otherwise reject.

3 Goto step 1.

Output: The sample x

Remark 16.1.2.

- The $q(x)$ and $p(x)$ are required to have the same support.
- The larger M, the lower the acceptance ratio and thus the simulation efficiency. Therefore, we should select $q(x)$ such that M is small.

Lemma 16.1.4 (correctness of algorithm).

In the Acceptance-Rejection algorithm, the probability to accept is $1/M$, and the average step needed to generate one accepted random sample x is M.

The Acceptance-Rejection algorithm will generate random number X with $p(x)$.

Proof. (1)

$$f(X = x \cap accept) = q(x)\frac{p(x)}{Mq(x)} = p(x)/M$$

$P(accept) = \int f(X = x \cap accept)dx = 1/M$. This is a geometric distribution with parameter $1/M$, which has an expected number of M.

(2)

$$\begin{aligned} P[X \le x|accept] &= P[X \le x \cap U < \frac{p(x)}{Mq(x)}]/P[U < \frac{p(x)}{Mq(x)}] \\ &= \frac{\int_{-\infty}^{x} \frac{p(y)}{Mq(y)} q(y)dy}{\int_{-\infty}^{\infty} \frac{p(y)}{Mq(y)} q(y)dy} \\ &= F_X(x) \end{aligned}$$

□

Example 16.1.2. Suppose we want to generate $p(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, x \ge 0$. We can choose $q(x) = e^{-x}, x \ge 0$. Then

$$M = \frac{p(x)}{q(x)} = e^{x-x^2}\sqrt{2/\pi}$$

will achieve maximum value at $x = 1$. Then the thresholding value is

$$\frac{p(x)}{q(x)M} = e^{(y-1)^2/2}.$$

16.1.4 Composition approach

If we can decompose the target distribution into several 'easier' distribution. Then we can synthesize the target random number via a composition approach.

The composition approach is summarized in the following Lemma.

Lemma 16.1.5 (composition method for random number generation). *Suppose we want to generate a random variable X with cdf F_X. If we can find decomposition such that*

$$F_X(x) = \sum_{i=1}^{N} p_i F_i(x),$$

where $p_i \geq 0, \forall i, p_1 + p_2 + \cdots + p_N = 1$. Or equivalently,

$$f_X(x) = \sum_{i=1}^{N} p_i f_i(x).$$

Then we can generate X via the following procedures:

- *Generate a positive random integer J such that $Pr(J = j) = p_j$.*
- *Generate X by generating a random variable with cdf F_j (using inverse transform or acceptance-rejection method).*

Proof. For a given fixed x, we have

$$\begin{aligned} Pr(X < x) &= \sum_{j=1}^{N} Pr(X \leq x | J = j) Pr(J = j) \\ &= \sum_{j=1}^{N} Pr(X \leq x | J = j) p_j \\ &= \sum_{j=1}^{N} F_j(x) p_j \\ &= F_X(x) \end{aligned}$$

where use the law of total probability in the first line [Theorem 11.2.1] □

Example 16.1.3. Suppose we want to generate a random variable X with density given by

$$f(x) = \begin{cases} x+1, -1 \leq x \leq 0 \\ -x+1, 0 \leq x \leq 1 \\ 0, otherwise \end{cases}.$$

We can have the decomposition as

$$\begin{aligned} f(x) &= (x+1)\mathbf{1}_{[-1,0]}(x) + (-x+1)\mathbf{1}_{[0,1]}(x) \\ &= 0.5 \times 2(x+1)\mathbf{1}_{[-1,0]}(x) + 0.5 \times 2(-x+1)\mathbf{1}_{[0,1]}(x) \\ &= p_1 f_1(x) + p_2 f_2(x) \end{aligned}$$

where $p_1 = p_2 = 0.5$ and $f_1 = 2(x+1)\mathbf{1}_{[-1,0]}(x)$ and $f_2 = 2(-x+1)\mathbf{1}_{[0,1]}(x)$.

We can integrate f_1 and f_2 to get

$$F_1(x) = x^2 + 2x + 1, F_2(x) = -x^2 + 2x,$$

and

$$F_1^{-1}(U) = \sqrt{U} - 1, F_2^{-1}(U) = 1 - \sqrt{1-U}.$$

Eventually, we can use the following algorithm to generate the desired X:

- Generate $U_1, U_2 \sim U(0,1)$ independently.
- If $U_1 < 0.5$, return $X = \sqrt{U_2} - 1$; else return $X = 1 - \sqrt{1-U_2}$.

Example 16.1.4. Suppose we want to generate a random variable X with density given by

$$f(x) = \begin{cases} 2 - a - 2(1-a), 0 \leq x \leq 1 \\ 0, otherwise \end{cases}.$$

We can have the decomposition as

$$\begin{aligned} f(x) &= a\mathbf{1}_{[0,1]}(x) + (1-a)2(1-x)\mathbf{1}_{[0,1]}(x) \\ &= p_1 f_1(x) + p_2 f_2(x) \end{aligned}$$

where $p_1 = a, p_2 = 1-a$ and $f_1 = \mathbf{1}_{[0,1]}(x)$ and $f_2 = (1-a)2(1-x)\mathbf{1}_{[0,1]}(x)$.

We can integrate f_1 and f_2 to get

$$F_1(x) = x, F_2(x) = -x^2 + 2x,$$

and

$$F_1^{-1}(U) = U, F_2^{-1}(U) = 1 - \sqrt{1-U}.$$

Eventually, we can use the following algorithm to generate the desired X:

- Generate $U_1, U_2 \sim U(0,1)$ independently.
- If $U_1 < a$, return $X = U_2$; else return $X = 1 - \sqrt{1-U_2}$.

16.1.5 Generate dependent continuous random variables

16.1.5.1 *Multivariate normal and lognormal distribution*

Note that Box-Muller method [Lemma 16.1.3] allows us to efficiently generate independent standard normal variables. By applying affine transformation [Theorem 14.1.1], we can generate multivariate random samples following distribution of $MN(\mu, \Sigma)$.

Methodology 16.1.2 (generate multivariate normal random vector). *Suppose we want to generate a random vector $X = (X_1, X_2, ..., X_n) \sim MN(\mu, \Sigma)$. We can use the following algorithm to generate X:*

- *Generate $Z_1, Z_2, ..., Z_n$ as iid $N(0,1)$, and let $Z = (Z_1, Z_2, ..., Z_n)$.*
- *Return $X = \mu + CZ$, where C is the Cholesky decomposition of Σ such that $\Sigma = CC^T$.*

Further, based on the definition of lognormal distribution [Definition 12.1.11, a random vector $X \sim LMN(\mu, \Sigma)$ if $\log X \sim MN(\mu, \Sigma)$. We can simply generate mulitvarate lognormal distribution in the following method.

Methodology 16.1.3 (generate multivariate lognormal random vector). *Suppose we want to generate a random vector $X = (X_1, X_2, ..., X_n) \sim LMN(\mu, \Sigma)$. We can use the following algorithm to generate X:*

- *Generate $Z = (Z_1, Z_2, ..., Z_n)$ as $MN(\mu, \Sigma)$.*
- *Return $X = (\exp(Z_1), \exp(Z_2), ..., \exp(Z_n))$.*

16.1.5.2 *Multivariate student t distribution*

We can similarly generate multivariate student t distribution based on the connection of t distribution to normal distribution [Definition 12.1.26] that

$$X = \mu + \sqrt{\frac{n}{W}} CZ,$$

where $W \sim \chi^2(n)$ and W is **independent** of Z, has a $t_n(\mu, \Sigma)$ multivariate distribution.

Methodology 16.1.4 (generate multivariate student t random vector). *Suppose we want to generate a random vector $X = (X_1, X_2, ..., X_n) \sim t_n(\mu, \Sigma)$. We can use the following algorithm to generate X:*

- *Generate $Z_1, Z_2, ..., Z_n$ as iid $N(0,1)$, and let $Z = (Z_1, Z_2, ..., Z_n)$.*
- *Generate a random $W \sim \chi^2(n)$ independent of Z.*
- *Return $X = \mu + \sqrt{\frac{n}{W}}CZ$, where C is the Cholesky decomposition of Σ such that $\Sigma = CC^T$.*

16.1.5.3 *General joint distribution*

If we have the joint cdf F_X for a random vector $X = (X_1, X_2, ..., X_N)$ and if conditional cdf can be easily computed, we can use the following method to generate random samples following a general joint distribution.

Methodology 16.1.5 (generate dependent continuous random variables from joint distribution). *Suppose we know the entire joint distribution F_X for a random vector $X = (X_1, X_2, ..., X_N)$.*

Then we can use the following algorithm to generate a random vector $X = (X_1, X_2, ..., X_N)$:

- *Generate X_1 from the marginal distribution F_{X_1} via inverse transform [subsection 16.1.1].*
- *Generate X_2 from the conditional distribution $F_{X_2|X_1}$via inverse transform.*
- *Generate X_3 from the conditional distribution $F_{X_3|X_1,X_2}$via inverse transform.*
- $\cdots$
- *Generate X_N from the conditional distribution $F_{X_N|X_1,X_2,...,X_{N-1}}$ via inverse transform.*
- *Finally, return $X = (X_1, X_2, ..., X_N)$.*

16.1.6 Generate discrete random variables

16.1.6.1 *Generate single discrete random variables*

Methodology 16.1.6 (generate discrete random variable). *Assume we are given the distribution function of random variable X,*

$$Pr(X = x_i) = p_i, i = 1, 2, ..., n.$$

We can divide the interval $[0,1]$ *into n buckets of*

$$[0,p_1),[p_1,p_1+p_2),...,[\sum_{i=1}^{n-1} p_i,1).$$

Then we generate a uniform random variable U: if U falls into bucket i, we will produce $X = x_i$.

Proof. It is easy to see that the probability for us to sample $X = x_i$ is p_i. □

16.1.6.2 *Generate correlated discrete random variables*

Methodology 16.1.7 (generate a pair of random variables with arbitrary joint distribution). *Assume we are given the* ***joint distribution*** *function for random variable X and Y as*

$$Pr(X = x, Y = y) = p_{xy}.$$

Then we have

$$Pr(X = x) = \sum_y p_{xy}$$

which can be used to generate X; and

$$Pr(Y = y|X = x) = p_{xy}/Pr(X = x)$$

which can be used to generate Y conditioned on the generated X.

Methodology 16.1.8 (generate two correlated random variables with arbitrary joint distribution). *Assume we are given the* ***marginal distribution function*** *for random variable X and Y as*

$$Pr(X = x) = p_x, Pr(Y = y) = p_y.$$

Denote the joint distribution by $Pr(X = x, Y = y) = a_{xy}$. Then we can solve a joint distribution with correlation ρ via the following equation system

$$\begin{aligned} \sum_x a_{xy} &= p_y, \forall y \\ \sum_y a_{xy} &= p_x, \forall x \\ \sum_{x,y} a_{xy} &= 1 \\ \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - E[X]^2}\sqrt{E[Y^2] - E[Y]^2}} &= \rho \end{aligned}$$

The solved the joint distribution can be used to generate correlated random variables via Methodology 16.1.7.

Proof. The equation system is constructed by requiring the satisfaction of margins, sum-to-unit, and correlation. □

Remark 16.1.3 (existence and uniqueness of joint distribution). Note that the joint distribution satisfying the specified correlation condition might not exist; even exists, that might not be unique. However, each joint distribution satisfying above condition will enable us to get the random variables with desired correlation.

Lemma 16.1.6 (generate two correlated Bernoulli random variables). *Suppose we want to generate two Bernoulli random variable X and Y with coefficients p and q, respectively. Further we want the two random variables have correlation coefficient ρ. That is,*

$$Pr(X = 1) = p, Pr(Y = 1) = q.$$

Then the joint distribution for (X, Y) can be uniquely determined, and the correlated random variables X and Y can be generated using Methodology 16.1.7.

Proof. Note that

$$\rho = \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - E[X]^2}\sqrt{E[Y^2] - E[Y]^2}} = \frac{Pr(X = 1, Y = 1) - pq}{\sqrt{p(1-p)}\sqrt{q(1-q)}}.$$

Denote the joint distribution

$$Pr(X = i, Y = j) = a_{ij}, i = 0, 1; j = 0, 1,$$

then we can solve the joint distribution via

$$\begin{aligned} a_{10} + a_{11} &= p \\ a_{01} + a_{11} &= q \\ a_{11} &= pq + \rho\sqrt{p(1-p)q(1-q)} \\ a_{00} + a_{01} + a_{10} + a_{11} &= 1 \end{aligned}$$

□

16.2 Monte Carlo integration

16.2.1 Naive approach

The goal of Monte Carlo integration is to evaluate

$$E[h(x)] = \int_{\mathcal{X}} h(x)f(x)dx.$$

The classic approach is to given by

Methodology 16.2.1 (Monte Carlo integration naive approach).

1. *Generate random samples* $X_1, ..., X_M$ *from pdf* $f(x)$
2. *Use*
$$\overline{h}_M = \frac{1}{M}\sum_{i=1}^{M} h(X_i)$$
to approximate the integral $\int_{\mathcal{X}} h(x)f(x)dx$.

Lemma 16.2.1 (properties of approximation). *[1]*

- *The estimator is unbiased:* $I_N(h) = E[\overline{h}_M] = I_h$;
- $\overline{h}_M$ *converges to* I_h *almost surely.*
- *The error distribution*
$$\sqrt{N}(\overline{h}_N - I_h) \sim N(0, \sigma_f^2) \text{ as } N \to \infty,$$
where $\sigma_f^2 = E[(\overline{h}_N - I_h)^2] = E[\overline{h}_N^2] - I_h^2$

Proof. (1) Linearity of expectation; (2) Strong law of large numbersTheorem 11.11.2; (3) Central limit theorem. □

Remark 16.2.1 (pros and cons of Monte Carlo integration).

- Ordinary numerical integration over n dimensions requires $O(m^n)$ if m is the number of divisions on each axis.
- Monte Carlo converges with speed $1/\sqrt{N}$ in standard deviation no matter how large dimensionality of the integral domain is.

16.2.2 Importance sampling

The naive approach is inefficient and has large error if there is a subset $\mathcal{Y} \subset \mathcal{X}$ has very low probability (i.e., rare events). The naive approach has to take a large number of samples in order to have enough samples in $\mathcal{Y}$. Importance sampling is a technique that changing the random sample generation probability so as to make the rare events happen often instead of rarely.

The idea is to generate these rare samples with a proposal distribution that puts larger probability weight on these rare samples and then properly discount the sample weight.

The importance sampling method is summarized in the following.

Algorithm 25: Importance sampling for Monte Carlo integration.

Input: the proposal distribution $q(x)$ and the target distribution $p(x)$, the number n of samples required, and the function $f(x)$ to be integrate

1 Sample $x \sim q(x)$ and calculate the corresponding weight $w(x) = p(x)/q(x)$
2 Goto step 1 until n samples are collected.
3 Evaluate

$$I(f) = \int_{\mathcal{X}} f(x)p(x)dx = \frac{1}{n}\sum_{i=1}^{n} f(x^i)w^i$$

Output: The sequence of sample pair$(x^1, w^1), ...(x^n, w^n)$, and evaluted result

Lemma 16.2.2 (property of importance sampling estimator). *Let* $I_n = \frac{1}{n}\sum_{i=1}^{n} f(x^i)w^i$*, then we have*

- *The approximiation is unbiased:*

$$E_q[\frac{1}{n}\sum_{i=1}^{n} f(x^i)w^i] = \int_{\mathcal{X}} f(x)p(x)dx = I(f)$$

  where $w(x) = p(x)/q(x)$*.*
- I_n *converges to* $\int_{\mathcal{X}} f(x)p(x)dx$ *almost surely as* $n \to \infty$*.*

Proof. (1) Use linearity of expectation:

$$E[\frac{1}{n}\sum_{i=1}^{n} f(x^i)w^i] = E[f(x)w(x)] = \int_{\mathcal{X}} f(x)w(x)q(x)dx = E[f(x)w(x)]$$
$$= \int_{\mathcal{X}} f(x)p(x)/q(x)q(x)dx = \int_{\mathcal{X}} f(x)p(x)dx$$

(2) Use strong law of large number [Theorem 11.11.2].Let $X_i = f(x^i)w^i = f(x^i)p(x)/q(x)$, then $E_q[X_i] = \mu = \int_{\mathcal{X}} f(x)p(x)dx$. □

Example 16.2.1 (tail probability estimation). [2, p. 93] Suppose $Z \sim N(0,1)$ and we want to estimate the tail probability as the integral of

$$I = P(Z > 4.5) \approx \frac{1}{M}\sum_{i=1}^{M} I(Z^i > 4.5)$$

If we have proposal distribution $q(z) = e^{-(x-4.5)} / \int_{4.5}^{\infty} e^{-x}dx$, then

$$I = P(Z > 4.5) \approx \frac{1}{M}\sum_{i=1}^{M} I(Z^i > 4.5)f(Z^i)/q(Z^i)$$

where $f(z)$ is the pdf of $N(0,1)$. In this importance sampling scheme, we can collect much more samples in regions $Z > 4.5$, but we will discount the sample contribution by the factor of $f(Z^i)/q(Z^i)$.

Example 16.2.2. Consider we evaluate $E[\max(S_T - K, 0)]$, where S_T is the random variable. If K is far from the mean value of S_T, then using Monte Carlo method will result in large variance. We can reduce the variance using importance sample via a proposal normal distribution with mean located near K.

The following theorem gives theoretical guidance on how to choose proposal distributions.

Theorem 16.2.1 (optimal proposal distribution). *[1][2, p. 95][3] In the importance sampling for Monte Carlo integration, we have*

$$I(f) = E_{p(x)}[f(x)] = E_{q(x)}[f(x)w(x)]$$

where $w(x) = p(x)/q(x)$. The variance of $f(x)w(x)$ with respect to $q(x)$ is given by

$$Var_{q(x)}[f(x)w(x)] = E_{q(x)}[f^2(x)w^2(x)] - I^2(f)$$

And the variance has the lower bound

$$E_{q(x)}[f^2(x)w^2(x)] \geq (E_{q(x)}[|f(x)|\, w(x)])^2$$

and the equality is achieved at

$$q^*(x) = \frac{\left|f(x)\right| w(x)}{\int \left|f(x)\right| w(x) dx}$$

Proof. Note that $E_{p(x)}[f(x)]$ is unbiased estimator of $I(f)$. □

Remark 16.2.2 (interpretation). Good sampling estimates can be achieved when we focus on sampling region from $p(x)$ where $\left|f(x)\right| p(x)$ is relatively large.

Remark 16.2.3 (convergence rate and optimal proposal distribution). Note that the optimal choice of the proposal distribution will not affect the convergence rate(still $1/\sqrt{N}$ in standard deviation), but will affect the prefactor.

16.3 Markov chain Monte Carlo

16.3.1 Basics

16.3.1.1 *Markov chain Monte Carlo (MCMC)*

Markov chain Monte Carlo (MCMC) methods are a class of algorithms to draw samples from a probability distribution P by construct and simulate an **irreducible and aperiodic Markov chain** that has the desired distribution P as the equilibrium distribution of the chain.

Given a discrete-time Markov chain characterized by transition probability $P(x|x')$, we can simulate a Markov chain in the following straight forward way:

Methodology 16.3.1 (Monte Carlo simulation of a Markov chain). *Given a discrete-time Markov chain characterized by transition probability $P(x|x')$. Let the initial state be $X_0 = x_0$. Set $n = 1$.*

1. *Generate next state $X_n =\sim P(x|x_{n-1})$.*
2. *Set $n = n + 1$ and go to step 1.*

Clearly Markov chains are relatively easy to simulate given transition distributions. Markov chain simulation therefore can be used to sample from an an unknown and possibly very complicated distribution if we can construct proper transition distribution.

Suppose our goal is to generate samples from distribution q but we do not have an explicit formula for the $q(x)$. Now suppose we can define transition probability $p(x|x')$ such that the q is the stationary distribution of the associated Markov chain. Such methods are particularly appealing when q is high-dimensional, with a complicated joint distribution structure.

16.3.2 Metropolis-Hasting algorithm

Metropolis-Hasting algorithm is one of the most well-known MCMC algorithm, particularly in the field of computational physics. In Metropolis-Hasting algorithm, we only need to know $p(x)$ up to the scale.

Algorithm 26: MCMC Metropolis-Hasting algorithm

Input: Initial state $x_0 \in \mathbb{R}^d$, $p(x)$ known upto the scale, and number of sampling steps

1 Given a previous state x, sample a new state x' based on **proposal distribution** $g(x'|x)$, a conditional distribution parameterized by two parameters x and x' .

2 Accepting the new state x' based on **accepting ratio function**, parameterized by two parameters x and x', $A(x'|x)$. One common choice of accepting ratio function is the Metropolis choice, given as

$$A(x'|x) = \min(1, \frac{P(x')}{P(x)} \frac{g(x|x')}{g(x'|x)})$$

3 Go to step 1 until n samples are collected.

Output: The full sequence of sampled states $x_0, x_1, ...$The low dimensional coordinate $y_i \in \mathbb{R}^p, i = 1, 2, ..., N$

Remark 16.3.1.

- To execute the algorithm, we have to know $P(x)$ up to scale; otherwise $A(x'|x)$ cannot be determined.For example, in Ising model of statistical physics, we have $P(x) \sim \exp(-U(x)/kT), x \in \mathbb{R}^{2N}$ based on Boltzmann distribution.
- There are different choices on g depending on the domain problem. In the most simple case, we can simply choose $g = 1$.

Theorem 16.3.1 (transition probability and detailed balance). *In the Metropolis-Hasting algorithm, the transition probability is given as*

$$P(x'|x) = g(x'|x)A(x'|x)$$

and detailed balance are satisfied between any pair of states when use Metropolis accepting ratio, that is

$$P(x'|x)P(x) = P(x|x')P(x').$$

Finally, the Metropolis-Hasting algorithm is constructing and sampling a Markov chain having an unique stationary distribution $P(x)$, provided that the Markov chain is irreducible, positive recurrent, and aperiodic.

Proof. (1) directly from the definition of proposal distribution and accepting ratio function; (2)

$$\frac{A(x'|x))}{A(x|x')} = \frac{\min(1, \frac{P(x')}{P(x)}\frac{g(x|x')}{g(x'|x)})}{\min(1, 1/\frac{P(x')}{P(x)}\frac{g(x|x')}{g(x'|x)})}$$

Note that if

$$\frac{P(x')}{P(x)}\frac{g(x|x')}{g(x'|x)} > 1,$$

we have

$$\frac{A(x'|x))}{A(x|x')} = \frac{P(x')}{P(x)}\frac{g(x|x')}{g(x'|x)},$$

which implies detailed balance given by

$$P(x)g(x'|x)A(x'|x) = P(x'|x)P(x) = P(x|x')P(x') = P(x')g(x|x')A(x|x')$$

If

$$\frac{P(x')}{P(x)}\frac{g(x|x')}{g(x'|x)} \leq 1$$

we get the same conclusion. (3)Note that the chain construct will satisfy detailed balance, and we can use results in section 20.5 to show that $P(x)$ is indeed the stationary distribution up to a scale. □

16.3.3 Gibbs sampling

Metropolis-Hasting method requires the full knowledge of the conditional distributions of the target distribution, which is not always possible. Gibbs sample method is an alternative method to generate multivariate samples, each component at a time.

Algorithm 27: MCMC Gibbs sampling algorithm

Input: Initial state $x_0 \in \mathbb{R}^d$, $p(x_i|x_{-i})$, and number of sampling steps

1 Given a previous state x^{s-1}, sample a new state x^s using the following procedure:

- sample $x_1^s \sim p(x_1^s|x_2^{s-1}...x_d^{s-1})$
- sample $x_2^s \sim p(x_2^s|x_1^s, x_3^{s-1}...x_d^{s-1})$
- ...
- sample $x_d^s \sim p(x_1^s|x_1^s...x_{d-1}^s)$

Go to step 1 until n samples are collected.

Output: The full sequence of sampled states $x_0, x_1, ...$

In Gibbs sampling method, only one component of the sample is updated at a time. And for a sample with d components, we need to update them sequentially. To understand that Gibbs sampling will also generate the desired stationary distribution, we see that Gibbs sampling is a special case of Metropolis-Hastings sampling with

$$g_k(x|x') = \begin{cases} p\left(x_k'|x_{-k}\right) & y'_{-k} = x_{-k} \\ 0 & \text{otherwise} \end{cases}$$

on updating the k component and that each component update will be accepted with probability 1.

Example 16.3.1. Consider the distribution

$$p(x,y) = \frac{n!}{(n-x)!x!} y^{(x+\alpha-1)}(1-y)^{(n-x+\beta-1)}, \quad x \in \{0,\dots,n\}, y \in [0,1]$$

It is difficult to directly simulate from $p(x,y)$ but the conditional distributions are easy to work with. We see that

$$p(x|y) \sim \text{Bin}(n,y), p(y|x) \sim \text{Beta}(x+\alpha, n-x+\beta$$

Since it's easy to simulate from each conditional, we can use Gibbs sampler to simulate samples from the joint distribution.

16.4 Monte Carlo for random processes

16.4.1 Simulating stochastic differential equations

16.4.1.1 *Simulating Brownian motion*

The building block of many stochastic process is Brownian motion. In this section, we first examine algorithms for generating sample paths of Brownian motion and geometric Brownian motion, then we examine more general cases.

Methodology 16.4.1 (exact simulation one dimensional Brownian motion). *[4, p. 705] Let $0 = t_0 < t_1 < ... < t_m$ be a set of dates.*

- *We can simulate a standard Brownian motion $W(t)$ on this set of dates using the following procedure:*
  - *set $W(t_0) = 0$.*
  - *generate independent $Z_j \sim N(0,1), j = 1, 2, ..., m$.*
  - *set $W(t_j) = W(t_{j-1}) + \sqrt{t_j - t_{j-1}}Z, j = 1, 2, ..., m$.*
- *We can simulate a drifting scaled Brownian motion*

$$W(t; \mu, \sigma, x_0) = x_0 + \mu t + \sigma W(t)$$

using the following procedure:
  - *set $W(t_0; \mu, \sigma, x_0) = x_0$.*
  - *generate independent $Z_j \sim N(0,1), j = 1, 2, ..., m$.*
  - *set $W(t_j; \mu, \sigma, x_0) = W(t_{j-1}; \mu, \sigma, x_0) + \mu(t_j - t_{j-1}) + \sigma\sqrt{t_j - t_{j-1}}Z, j = 1, 2, ..., m$.*

Note that the simulated sequence $W(t)$ and $W(t; \mu, \sigma, x_0)$ has the exact distribution desired.

16.4.1.2 *Simulating linear arithmetic SDE*

Methodology 16.4.2 (simulating linear arithmetic SDE). *[5, p. 104] Suppose we want to simulate*

$$dx(t) = \mu(t)dt + \sigma(t)dW_t, x(0) = x_0.$$

Let $0 = t_0 < t_1 < ... < t_m$ be a set of evenly space dates.

We can simulate a standard Brownian motion $W(t)$ on this set of dates using the following procedure:

- *Start from* $i = 1$.
- $$x(t_i) = x(t_{i-1}) + \mu(t_i)(t_i - t_{i-1}) + \sigma(t_i)\sqrt{t_i - t_{i-1}}Z$$
  where $Z \sim N(0,1)$.
- *Set* $i = i + 1$.
- *Repeat step (2)(3) until* $i = N$.

16.4.1.3 *Simulating linear geometric SDE*

Methodology 16.4.3 (simulating linear arithmetic SDE). *[5, p. 104] Suppose we want to simulate*

$$dx(t)/x(t) = \mu(t)dt + \sigma(t)dW_t, x(0) = x_0.$$

Let $0 = t_0 < t_1 < ... < t_m$ *be a set of evenly space dates.*

We can simulate a standard Brownian motion $W(t)$ *on this set of dates using the following procedure:*

- *Start from* $i = 1$.
- $$x(t_i) = x(t_{i-1}) \exp((\mu(t_i) - \frac{1}{2}\sigma(t_i)^2)(t_i - t_{i-1}) + \sigma(t_i)\sqrt{t_i - t_{i-1}}Z)$$
  where $Z \sim N(0,1)$.
- *Set* $i = i + 1$.
- *Repeat step (2)(3) until* $i = N$.

16.4.1.4 *Simulation mean-reversion(OU) process*

Methodology 16.4.4 (simulating a mean-reversion process). *[5, p. 104] Suppose we want to simulate*

$$dx(t) = (\theta(t) - kx_t)dt + \sigma(t)dW_t, x(0) = x_0.$$

Let $0 = t_0 < t_1 < ... < t_m$ *be a set of evenly space dates.*

We can simulate a standard Brownian motion $W(t)$ *on this set of dates using the following procedure:*

- *start from* $i = 1$.
- $$x(t_i) = x_{t_{i-1}} + (\theta(t_{i-1}) - kx(t_{i-1})(t_i - t_{i-1}) + \sigma(t_i)\sqrt{t_i - t_{i-1}}Z$$
  where $Z \sim N(0,1)$.

- *Set $i = i + 1$.*
- *Repeat step (2)(3) until $i = N$.*

16.4.2 Stochastic interpolation

16.4.2.1 *Interpolating Gaussian processes*

Methodology 16.4.5 (interpolating a Gaussian process). *Let X_t be a Gaussian process with mean function $\mu(t)$ and covariance function $cov(t_i, t_j)$.*

Consider a set of time points $t_a < t_1 < t_2 < ... < t_n < t_b$. Suppose we are given $X(t_a)$ and $X(t_b)$ and we want to sample $X(t_1), X(t_2), X(t_3), ..., X(t_n)$.

Construct a conditional mean vector and a conditional covariance matrix given by

$$m = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), V = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^T,$$

where

$$\Sigma_{22} = \begin{bmatrix} Cov(X_1, X_1) & \cdots & Cov(X_1, X_n) \\ \vdots & \cdots & \ddots \quad \vdots \\ Cov(X_n, X_1) & \cdots & Cov(X_n, X_n) \end{bmatrix}, \Sigma_{11} = \begin{bmatrix} Cov(X_a, X_a) & Cov(X_a, X_b) \\ Cov(X_a, X_b) & Cov(X_b, X_b) \end{bmatrix},$$

$$\Sigma_{12} = \begin{bmatrix} Cov(X_a, X_1) & \cdots & Cov(X_a, X_n) \\ Cov(X_b, X_1) & \cdots & Cov(X_b, X_n) \end{bmatrix}, \mu_2 = \begin{bmatrix} \mu(t_1) \\ \mu(t_2) \\ \cdots \\ \mu(t_n) \end{bmatrix}, x_2 = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_n \end{bmatrix}, \mu_1 = \begin{bmatrix} X_a \\ X_b \end{bmatrix}$$

Then we can generate a multivariate Gaussian random variable with mean m and covariance matrix V using Methodology 16.1.2.

Proof. □

16.4.2.2 *Interpolating one Dimensional Brownian motions*

Lemma 16.4.1 (interpolating a standard Brownian motion). *Let X_t be a standard Brownian motion. Consider three time points $t_a < t_1 < t_b$. Then conditioning on values of $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random variable with mean and variance*

$$\mu = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)), \sigma^2 = t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)}.$$

Proof. Note that the random vector $(X(t_a), X(t_1), X(t_b))$ will follow multivariate Gaussian distribution since X_t is a Gaussian process. Conditioning on $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random variable with conditional mean and variance given by [Theorem 14.1.2]

$$\mu = [t_a \; t_1] \begin{bmatrix} t_a & t_a \\ t_a & t_b \end{bmatrix}^{-1} \begin{bmatrix} X(t_a) \\ X(t_b) \end{bmatrix} = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)),$$

and variance

$$\sigma^2 = t_1 - [t_a \; t_1] \begin{bmatrix} t_a & t_a \\ t_a & t_b \end{bmatrix}^{-1} \begin{bmatrix} t_a \\ t_1 \end{bmatrix} = t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)}$$

□

Lemma 16.4.2 (interpolating a scaled Brownian motion). *Let Y_t be a Brownian motion. Let $X_t = vt + \sigma Y_t$. Consider three time points $t_a < t_1 < t_b$. Then conditioning on values of $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random variable with mean and variance*

$$\mu = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)), V^2 = \sigma^2(t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)}).$$

Proof. Note that the random vector $(X(t_a), X(t_1), X(t_b))$ will follow multivariate Gaussian distribution since X_t is a Gaussian process. Conditioning on $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random variable with conditional mean and variance given by [Theorem 14.1.2]

$$\mu = vt_1 + [\sigma^2 t_a \; \sigma^2 t_1] \begin{bmatrix} \sigma^2 t_a & \sigma^2 t_a \\ \sigma^2 t_a & \sigma^2 t_b \end{bmatrix}^{-1} \begin{bmatrix} X(t_a) - vt_a \\ X(t_b) - vt_b \end{bmatrix} = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)),$$

and variance

$$V^2 = \sigma^2 t_1 - [\sigma^2 t_a \; \sigma^2 t_1] \begin{bmatrix} \sigma^2 t_a & \sigma^2 t_a \\ \sigma^2 t_a & \sigma^2 t_b \end{bmatrix}^{-1} \begin{bmatrix} \sigma^2 t_a \\ \sigma^2 t_1 \end{bmatrix} = \sigma^2(t_1 - t_a) - \sigma^2 \frac{(t_1 - t_a)^2}{(t_b - t_a)}$$

□

Remark 16.4.1. **Note that homogeneous drift has no effect on the conditional mean.**

Methodology 16.4.6 (sequential interpolating a scaled Brownian motion). *Let X_t be a scaled Brownian motion with drift parameter v and variance parameter σ. Consider time points $t_a < t_1 < t_2 < \cdots < t_n < t_b$. Then conditioning on values of $X(t_a)$ and $X(t_b)$, we can sample $X(t_1), X(t_2), ..., X(t_n)$* ***as one sample trajectory*** *in the following way:*

- *sample $X(t_1)$ as a Gaussian random variable with mean and variance*

$$\mu = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)), V^2 = \sigma(t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)}).$$

- *sample $X(t_2)$ as a Gaussian random variable with mean and variance*

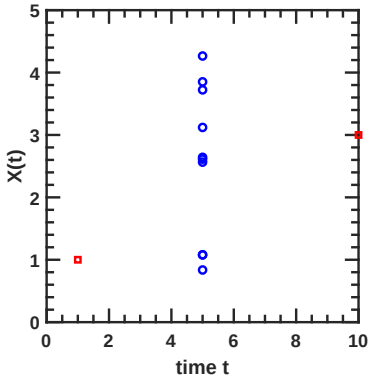
$$\mu = X(t_1) + \frac{t_2 - t_1}{t_b - t_1}(X(t_b) - X(t_1)), V^2 = \sigma(t_2 - t_1 - \frac{(t_2 - t_1)^2}{(t_b - t_1)}).$$

- *sample $X(t_i), i = 3, 4, ..., n$ as a Gaussian random variable with mean and variance*

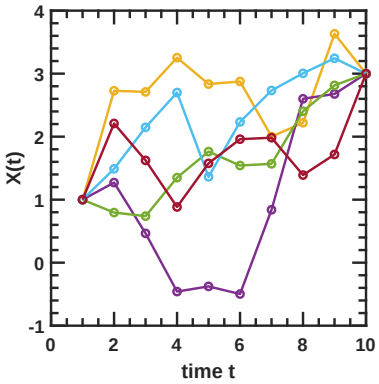
$$\mu = X(t_{i-1}) + \frac{t_i - t_{i-1}}{t_b - t_{i-1}}(X(t_b) - X(t_{i-1})), V^2 = \sigma(t_i - t_{i-1} - \frac{(t_i - t_{i-1})^2}{(t_b - t_{i-1})}).$$

Proof. Note that we use conditional independence such that $X(t_i)$ is independence of $X(t_{i-2})$ when conditioning on $X(t_{i-1})$. □

Example 16.4.1. In Figure 16.4.1(a) we sample $X(5)$ with $X(1) = 1, X(10) = 3$. In (b), we sample $X(2), ..., X(9)$ sequentially to produce a sample trajectory.



(a) Brownian motion interpolation at a single time point.



(b) Brownian motion interpolation at multiple time points to generate trajectories.

Figure 16.4.1: Brownian motion interpolation Demo.

16.4.2.3 *Interpolating multi-dimensional Brownian motions*

Lemma 16.4.3 (interpolating a standard multi-dimensional Brownian motion). *Let $X(t) = (X_1(t), ..., X_n(t))$ be a n-dimensional Brownian motion as the solution to an SDE governed by*

$$dX_i(t) = dW_i(t), i = 1, 2, ..., n,$$

where $E[dW_i(t)dW_j(s)] = \rho_{ij}\delta(s-t)$. Let Cholesky decomposition of the instantaneous correlation matrix $\rho \in \mathbb{R}^{n\times n}$ be given by $\rho = DD^T$. Now consider three time points $t_a < t_1 < t_b$. Then conditioning on values of $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random variable with mean and variance

$$\mu = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}(X(t_b) - X(t_a)), \Sigma = (t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)})\rho.$$

Proof. Note that [Definition 18.3.3] the covariance matrix for $X(t)$ as $Cov(X(t), X(s)) = \rho \min(s, t)$.

Note that the random vector $(X(t_a), X(t_1), X(t_b))$ will follow multivariate Gaussian distribution. Conditioning on $X(t_a)$ and $X(t_b)$, $X(t_1)$ is a Gaussian random vector with conditional mean and variance given by [Theorem 14.1.2]

$$\begin{aligned}
\mu &= [t_a\rho\ t_1\rho] \begin{bmatrix} t_a\rho & t_a\rho \\ t_a\rho & t_b\rho \end{bmatrix}^{-1} \begin{bmatrix} X(t_a) \\ X(t_b) \end{bmatrix} \\
&= D[t_a\ t_1] \begin{bmatrix} D^T & 0 \\ 0 & D^T \end{bmatrix} \left(\begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} t_a I_n & t_a I_n \\ t_a I_n & t_b I_n \end{bmatrix} \begin{bmatrix} D^T & 0 \\ 0 & D^T \end{bmatrix} \right)^{-1} \begin{bmatrix} X(t_a) \\ X(t_b) \end{bmatrix} \\
&= D[t_a\ t_1] \begin{bmatrix} t_a I_n & t_a I_n \\ t_a I_n & t_b I_n \end{bmatrix}^{-1} \begin{bmatrix} D^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} X(t_a) \\ X(t_b) \end{bmatrix} \\
&= DD^{-1}X(t_a) + \frac{t_1 - t_a}{t_b - t_a}\rho(X(t_b) - X(t_a)) \\
&= X(t_a) + \frac{t_1 - t_a}{t_b - t_a}\rho(X(t_b) - X(t_a))
\end{aligned}$$

and variance

$$\begin{aligned}
\sigma^2 &= t_1\rho - [t_a\rho\ t_1\rho] \begin{bmatrix} t_a\rho & t_a\rho \\ t_a\rho & t_b\rho \end{bmatrix}^{-1} \begin{bmatrix} t_a\rho \\ t_1\rho \end{bmatrix} \\
&= t_1\rho - D[t_a\ t_1]\left(\begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} t_a I_n & t_a I_n \\ t_a I_n & t_b I_n \end{bmatrix} \begin{bmatrix} D^T & 0 \\ 0 & D^T \end{bmatrix} \right)^{-1} \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} t_a \\ t_1 \end{bmatrix} D^T \\
&= D(t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)})D^T \\
&= (t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)})\rho
\end{aligned}$$

□

Methodology 16.4.7 (sequential interpolating a multidimensional Brownian motion). *Let $X(t)$ be a n dimensional Brownian motion with instantaneous correlation $\rho \in \mathbb{R}^{n\times n}$. Consider time points $t_a < t_1 < t_2 < \cdots < t_n < t_b$. Then conditioning on values of $X(t_a)$ and $X(t_b)$, we can sample $X(t_1), X(t_2), ..., X(t_n)$* ***as one sample trajectory*** *in the following way:*

- *sample* $X(t_1)$ *as a Gaussian random variable with mean and variance*

$$\mu = X(t_a) + \frac{t_1 - t_a}{t_b - t_a}\rho(X(t_b) - X(t_a)), \sigma^2 = (t_1 - t_a - \frac{(t_1 - t_a)^2}{(t_b - t_a)})\rho.$$

- *sample* $X(t_2)$ *as a Gaussian random variable with mean and variance*

$$\mu = X(t_1) + \frac{t_2 - t_1}{t_b - t_1}\rho(X(t_b) - X(t_1)), \sigma^2 = (t_2 - t_1 - \frac{(t_2 - t_1)^2}{(t_b - t_1)})\rho.$$

- *sample* $X(t_i), i = 3, 4, ..., n$ *as a Gaussian random variable with mean and variance*

$$\mu = X(t_{i-1}) + \frac{t_i - t_{i-1}}{t_b - t_{i-1}}\rho(X(t_b) - X(t_{i-1})), \sigma^2 = (t_i - t_{i-1} - \frac{(t_i - t_{i-1})^2}{(t_b - t_{i-1})})\rho.$$

Proof. Note that we use conditional independence such that $X(t_i)$ is indpendence of $X(t_{i-2})$ when conditioning on $X(t_{i-1})$. □

16.5 Monte Carlo variance reduction

16.5.1 Antithetic sampling

16.5.1.1 *Basic principles*

Definition 16.5.1 (antithetic variates).

- *Let X be a random variable $X \sim U(0,1)$, then $1-X$ is also $U(0,1)$.*
- *Let Y be a standard normal random variable $N(0,1)$, then $-Y$ is also $N(0,1)$ random variable.*
- *Let Z be a normal random variable $N(\mu, \sigma^2)$, then $2\mu - Z$ is also $N(\mu, \sigma^2)$ random variable.*

*The pairs X and $1-X$ or Y and $-Y$ are called **antithetic variates**.*

Remark 16.5.1 (negative correlation between antithetic variate pairs). The most important property of antithetic variates is that each pair is negatively correlated. For example, X and $1-X$; Y and $-Y$.

Theorem 16.5.1 (preserving negative correlation via monotone function). *[6, p. 209] If $h(x_1, ..., x_n)$ is a monotone function of each of its arguments, then for a set of independent uniform random variables $U_1, ..., U_n$,*

$$Cov[h(U_1, ..., U_n), h(1-U_1), ..., 1-U_n] \leq 0.$$

Similarly, if $Z_1, ..., Z_n$ is a set of independent standard Gaussian random variables, then

$$Cov[h(Z_1, ..., Z_n), h(1-Z_1), ..., 1-Z_n] \leq 0.$$

Proof. See reference. □

Lemma 16.5.1 (preserving negative correlation via linear function). *Let $X, \hat{X}$ be two iid random variables and negatively correlated,i.e., $Cov(X, \hat{X}) < 0$. Let $f(x) = ax + b$ be a linear function, then*

$$Cov(f(X), f(\hat{X})) < 0.$$

Proof.

$$Cov(f(X), f(\hat{X})) = a^2 Cov(X, \hat{X}) < 0.$$

□

Lemma 16.5.2 (generating antithetic variates of other distributions). *[5, p. 205] Let F be the cdf of the target distribution, let $F^{-1} = \inf\{F^{-1}(x > u)\}$. Then*

- *F^{-1} is a monotone function.*
- *$F^{-1}(U)$ and $F^{-1}(1-U)$ are a pair antithetic variates with negative correlations.*

Proof. (2) use Theorem 16.5.1. □

16.5.1.2 *Methods and analysis*

General principles
Let S and S' be a pair of antithetic variates random variables. If $C = g(S)$ (where g is a monotone function) is a calculation based on S, then $C' = g(S')$ is negatively correlated with C [Theorem 16.5.1].

Lemma 16.5.3. *Considering n iid random sample of S and n antithetic variate iid random sample of S'. Suppose we want to estimate the mean value of a function C,which is a monotone function of S. The estimator for the mean value is given as*

$$\hat{C} = \frac{1}{2n}(\sum_{k=1}^{n} C_k + \sum_{k=1}^{n} C'_k) = \frac{1}{n}\sum_{k=1}^{n} \frac{1}{2}(C'_k + C_k)$$

Then

- *$\hat{C}$ is an unbiased estimator.*
- *$Var[C]$ is smaller than the variance of $2n$ sample mean estimator C; that is, using antithetic variates, smaller estimation variance can be achieved when using the same size of random samples.*

Proof. Note that the n pairs of $\frac{1}{2}(C'_k + C_k)$ are independent with each other. Moreover,

$$\sigma^2(\frac{1}{2}(C'_k + C_k)) \leq \sigma^2(C_k)$$

since

$$\sigma^2(\frac{1}{2}(C'_k + C_k)) \leq \frac{1}{4}(\sigma^2(C_k) + \sigma^2(C'_k) + 2cov(C_k, C'_k)) \leq \frac{1}{2}\sigma^2(C_k)$$

since $cov(C_k, C'_k) \leq 0$.

Therefore, when using the same number of random samples, antithetic variate samples can have a smaller variance on estimating the expectation. □

Remark 16.5.2 (caution!). Antithetic method can only be used to **estimate the expectation rather than the variance.**

Example 16.5.1. Suppose we want to estimate the mean value $f(X) = X^2, X \sim N(2,1)$. We can use the following estimator

$$\frac{1}{n}\sum_{i=1}^{n}(f(X_i) + f(-X_i))/2.$$

Note that even though X^2 is not a monotone function, it is nearly 'monotone' in regions where probability weight concentrates.

Example 16.5.2 (option pricing).

- Consider we simulate a Brownian motion by adding up random samples from $N(0,1)$. Let $C = \sum_{i=1}^{n} S_i$. Then the antithetic path is given as $C' = \sum_{i=1}^{n} -S_i$. These antithetic paths can be used to reduce variance of Monte Carlo method based option pricing.
- Suppose we want to estimate

  $$C = E[e^{-rT}\max\{0, S_T - K\}],$$

  where $S_T = S_0 e^{(r-\sigma^2/2)T+\sigma\sqrt{T}X}, X \sim N(0,1)$. We can use standard normal antithetic variate to estimate C to reduce variance since S_T and C are both monotone functions.

Lemma 16.5.4 (variance reduction using normal random sample). *[5, p. 208] Let $Y = f(Z)$ with $Z = (Z_1, ..., Z_d) \sim N(0,1)$. Define the symmetric and antisymmetric parts of f, respectively, by*

$$f_0(z) = \frac{f(z) + f(-z)}{2}, f_1(z) = \frac{f(z) - f(-z)}{2}.$$

We have

- $$E[f_0(Z)f_1(Z)] = 0, E[f_1(Z)] = 0,$$
  $$Cov(f_0(Z), f_1(Z)) = E[f_0(Z)f_1(Z)] - E[f_0(Z)]E[f_1(Z)] = 0.$$
  that is $f_0(Z)$ and $f_1(Z)$ are not correlated.

- *variance decomposition*

$$Var[f(Z)] = Var[f_0(Z)] + Var[f_1(Z)].$$

- $$Var[\frac{1}{2}(f_1(Z) + f_1(-Z)] = 0$$

- *Antithetic sampling eliminates all variance if f is antisymmetric($f = f_1$);It eliminates no variance if f is symmetric($f = f_0$).*

Proof. (1) $E[f_1(Z)] = 0$ because of $f_1(x) = -f(-x)$ and Z is symmetric around 0.

$$E[f_0(Z)f_1(Z)] = \frac{1}{4}(E[f(Z)^2 + f(-Z)^2]) = 0$$

because Z is symmetric around 0. (2) use

$$Var[X + Y] = Var[X]^2 + Var[Y]^2 + 2Cov(X, Y).$$

(3) Note that $f_1(Z) = -f_1(-Z)$. (4) Due to the variance decomposition in (2) and (3). □

16.5.2 Control variates

16.5.2.1 *Basic principles*

General principles
The principle of control variates is to use estimation error of known quantities to improve the estimation of unknown quantities.

Lemma 16.5.5. *[5, p. 188] Let $X_1, ..., X_n$ be n samples of X, let $Y_1, ..., Y_n$ be n samples of Y with $E[Y]$ **known**. Then for any constant β, the statistic*

$$C(\beta) = \frac{1}{n}\sum_{i=1}^{n}(X_k - \beta(Y_k - E[Y]))$$

is a control variate estimator for $E[X]$. Then

- *$C(\beta)$ is an unbiased estimator.*
- *its variance is given as*

$$Var[C(\beta)] = \frac{1}{n}(Var[X] - 2\beta cov(X, Y) + \beta^2 Var[Y])$$

- *If β is chosen as*

$$\beta = \frac{Cov(X,Y)}{Var[Y]}$$

then $Var[C(\beta)] = Var[C] - \frac{Cov(X,Y)^2}{Var[Y]} \leq Var[C]$. *The ratio of the variance is given as*

$$\frac{Var[C(\beta)]}{Var[C]} = 1 - \rho_{XY}^2$$

Proof. (1)(2) Straight forward. (3) do optimization on β. □

Remark 16.5.3 (interpretation).

- The working of this method relies on that there exist correlations between two random variables.
- The amount of the variance reduction depends on the correlation coefficients of X and Y. From (3), if X and Y is uncorrelated, then zero reduction can be achieved; if perfectly correlated, we can achieve complete reduction.

Remark 16.5.4 (practical calculation of β). Note that the optimal β requires calculation of $Cov(X,Y)$(and $E[X]$), which might not be available. But we can estimate them from the simulation experiment itself. For example,

$$\hat{\beta} = \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^n (Y_i - \overline{Y})^2}.$$

Example 16.5.3 (underlying asset as control variate for derivative pricing). [5, p. 188] Consider an option on underlying asset $S(t)$. We know that $e^{-rt}S(t)$ is a martingale and

$$E[e^{-rt}S(t)] = S(0).$$

Let the **discounted** payoff of the option be Y, then the control variate estimator for the price will be

$$\frac{1}{n}\sum_{i=1}^n (Y_i - \beta(S_i(T) - e^{rT}S(0)).$$

Consider the case that $Y = e^{-rT}\max(S(T) - K, 0)$. Then the effectiveness of variance reduction depends on the correlation and Y and S, which further depends on the value of K. If $K = 0$, then we have perfect correlation to achieve large reduction; if K is large, then the correlation will be small.

16.5.2.2 *Multiple control variates*

Lemma 16.5.6. *[5, p. 196] Suppose each replication i of a simulation produces outputs Y_i and $X_i = (X_1^{(1)}, ..., X_i^{(d)})^T$ and suppose the vector of expectations $E[X]$ is known. We assume the covariance matrix $Cov(X, Y)$ is known as*

$$\begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{XY} & \sigma_Y^2 \end{bmatrix}$$

Then the estimator of Y via multiple control variates are given

$$\hat{Y}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T(X_i - E[X]))$$

where $\beta \in \mathbb{R}^d$. We have

- $E[\hat{Y}(\beta)] = E[Y]$.
- *The variance of $\hat{Y}(\beta)$ is*

$$Var[\hat{Y}(\beta)] = \frac{1}{n}(\sigma_Y^2 - 2\beta^T\Sigma_{XY} + \beta^T\Sigma_X\beta).$$

- *The optimal β achieving minimal variance is*

$$\beta^* = \Sigma_X^{-1}\Sigma_{XY},$$

with the minimum variance given as

$$\sigma_Y^2 - \Sigma_{XY}^T\Sigma_X^{-1}\Sigma_{XY}.$$

Proof. Straight forward. □

16.6 Notes on bibliography

Introductory level Monte Carlo methods, see [7]. Intermediate level Monte Carlo methods, see [8][2][9][3].

For an excellent review on MCMC, see [1][10].

For advanced Monte Carlo methods in financial applications, see [11].

For Quasi Monte Carlo methods, see [12][11].

BIBLIOGRAPHY

1. Andrieu, C., De Freitas, N., Doucet, A. & Jordan, M. I. An introduction to MCMC for machine learning. *Machine learning* **50,** 5–43 (2003).
2. Robert, C. & Casella, G. *Monte Carlo statistical methods* (Springer Science & Business Media, 2013).
3. Owen, A. B. *Monte Carlo theory, methods and examples* (2013).
4. Campolieti, G. & Makarov, R. N. *Financial mathematics: a comprehensive treatment* (CRC Press, 2016).
5. Glasserman, P. *Monte Carlo methods in financial engineering* (Springer Science & Business Media, 2003).
6. Ross, S. *Simulation* ISBN: 9780124158252 (Academic Press, 2013).
7. Dunn, W. L. & Shultis, J. K. *Exploring Monte Carlo Methods* (Elsevier, 2011).
8. Kroese, D. P., Taimre, T. & Botev, Z. I. *Handbook of monte carlo methods* (John Wiley & Sons, 2013).
9. Spall, J. C. *Introduction to stochastic search and optimization: estimation, simulation, and control* (John Wiley & Sons, 2005).
10. Gamerman, D. & Lopes, H. F. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference* (Chapman and Hall/CRC, 2006).
11. Jäckel, P. *Monte Carlo Methods in Finance* ISBN: 9780471497417 (Wiley, 2002).
12. Niederreiter, H. *Random Number Generation and Quasi-Monte Carlo Methods* ISBN: 9781611970081 (SIAM, 1992).

Part IV

DYNAMICS MODELING METHODS