A GRADUATE COURSE

IN

SIMULATION

A GRADUATE COURSE

IN

SIMULATION

DDA6104 Notebook

Lecturer

Prof. Xinyun Chen

Institute for Data and Decision Analytics

Tex Written By

Mr. Jie Wang

The Chinese University of Hongkong, Shenzhen

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Contents

Acknowledgments

This book is taken notes from the DDA6001 in spring semester, 2020. These lecture notes were taken and compiled in LaTeX by Jie Wang, an undergraduate student in Spring 2020. Prof. Masakiyo Miyazawa has not edited this document. Students taking this course may use the notes as part of their reading and reference materials. This version of the lecture notes were revised and extended for many times, but may still contain many mistakes and typos, including English grammatical and spelling errors, in the notes. It would be greatly appreciated if those students, who will use the notes as their reading or reference material, tell any mistakes and typos to Jie Wang for improving this notebook.

Notations and Conventions

sup	least upper bound
inf	greatest lower bound
$\overline{E}$	closure of E
$f \circ g$	composition
$\limsup(\liminf)$	upper (lower) limit
$L(\mathcal{P}, f), U(\mathcal{P}, f)$	Riemann sums
$\mathcal{R}[a, b]$	classes of Riemann integrable functions on $[a, b]$
$\underline{\int_a^b} f(x)\,\mathrm{d}x, \overline{\int_a^b} f(x)\,\mathrm{d}x$	Riemann integrals
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	inner product
$\omega(f; E)$	oscillation of f over set E
$\\| \cdot \\|$	norm
∇f	gradient
$\frac{\partial f}{\partial x_i}, f_{x_i}, f_i, \partial_i f, D_i f$	partial derivatives
$D_{\boldsymbol{v}} f$	directional derivative at direction $\boldsymbol{v}$
$\frac{\partial(y_1, \ldots, y_m)}{\partial(x_1, \ldots, x_n)}$	Jacobian
$\mathbb{S}^n$	set of real symmetric $n \times n$ matrices
$\succ$ $(\succeq)$	positive (semi)-definite
$\mathcal{C}^m$	classes of m-th order continuously differentiable functions
$\mathcal{C}(E; \mathbb{R}^m)$	set of $\mathcal{C}^1$ mapping from E to $\mathbb{R}^m$
$(\mathcal{H}, d)$	metric space

Chapter 1

Week1

1.1. Wednesday

1.1.1. Motivation

To evaulate a dynamic, stochastic system,

- the performance measure is not analytically tractable

Example:

- Expected waiting time in an emergency room;
- Price of option;
- Probability of failure of a power grid.

Applications:

- Small-sample inference. Bootstrap, or permulation test;
- Non-convex Optimization. SGD, or simulation annealing.

Basic Procedure for simulation:

$$\boldsymbol{U} \to \boldsymbol{X} \to \boldsymbol{Y}$$

with $\boldsymbol{U} \sim \mathcal{U}(0,1)$, $\boldsymbol{X}$ denotes the input random variable with specified distribution, and $\boldsymbol{Y}$ denotes the output random variable whose properties we wish to estimate. We will mainly talk about how to generate $\boldsymbol{X}$ from $\boldsymbol{U}$.

■ **Example 1.1** [Queuing Problem] Consider a single-server queue with

- infinite buffer size;
- FIFO;
- A_n: arrive time; (usually given)
- D_n: departure time, decomposed as:
  - $T_n = A_{n+1} - A_n$, inter-arrival time;
  - V_n: service time
- W_n: waiting time (before entering service):

$$W_n = (D_{n-1} - A_n)^+$$

Lindley recursion:

$$W_{n+1} = (W_n + V_n - T_n)^+$$

Performance measure:

- Mean waiting time: $\mathbb{E}[W_\infty] = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N W_k$;
- Tail of waiting time: $P(W_\infty > x) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N \mathbf{1}\{W_k > x\}$

There is no closed-form for general distribution. ■

(R) Simulation solution (Monte Carlo Method):

1. Generate i.i.d. samples of V_n and T_n.
2. Apply Lindley recursion to simulate W_∞.
   - Option 1: simulate a long sequence of W_n to approximate W_∞;
   - Option 2: apply advanced method to simulate exact W_∞.
3. Repeat the simulation procedure to get multiple W_∞
4. Output the empirical distribution of simulated samples.

Performance optimization: Denote μ as the service time.

$$\begin{aligned}\min_{\mu} \quad & \mathrm{Cost}(\mu) + \mathbb{E}[W_\infty(\mu)] \\ \text{s.t.} \quad & W_{n+1} = \left(W_n + \frac{V_n}{\mu} - T_n\right)^+\end{aligned}$$

Or

$$\begin{aligned}\min_{\mu} \quad & \mathrm{Cost}(\mu) \\ \text{s.t.} \quad & W_{n+1} = \left(W_n + \frac{V_n}{\mu} - T_n\right)^+ \\ & \mathbb{P}(W_\infty > x) < \varepsilon\end{aligned}$$

Simulation optimization:

1. Generate $Z_n(\mu)$ by Lindley recursion such that

$$\mathbb{E}[Z_n(\mu)] = \frac{\partial}{\partial \mu}\mathbb{E}[W_\infty(\mu)]$$

2. Apply first order method for optimization:

$$\mu_{k+1} = \mu_k - \delta_k \cdot \Big(\mathbb{E}[Z_n(\mu)] + \nabla_\mu \mathrm{Cost}(\mu)\Big)$$

Key Issues.

1. How to generate the needed random variables?
2. How to compute the limiting stationary distribution?
3. How to estimate the sensitivity?
4. How to use simulation to optimize?

Computational Efficiency:

1. The computational cost to obtain a good numerical solution;
2. How to exploit the problem structure to speed up the computation?

1.1.2. Generating Random Variables

We first discuss how to simulate a scalar random variable.

Generating Uniform Numbers. Two types of random number generate (RNG):

- Mathematical (Pseudo): multiple recursive generator;
- Physical: nuclear decay.

Multiple Recursive Generator (MRG):

1. Choose a large prime number m, and $a_1, \ldots, a_k$ are integers such that the following recursion has cycle length $m^k - 1$:

$$x_i \equiv (a_1 x_{i-1} + a_2 x_{i-2} + \cdots + a_k x_{i-k}) \pmod{m}.$$

   Output: $U_i = (x_i, x_{i-1}, \ldots, x_{i-k+1})$.
2. Example: $k = 1, m = 2^{31} - 1$.

Hot research topics in UNG:

- better design of UNG;
- test whether the generated sequence is i.i.d.;
- measrue the performance of a generator.

In most of the analysis, we assume that $\boldsymbol{U}$ is given.

Transfer RN to RV.

- Distribution function: $F_X(x) = \mathbb{P}(X \le x)$.
- Inverse function: $F_X^{-1}(u) = \inf\{x : \ F(x) \ge u\}$.

Theorem 1.1 — **Inverse Method.** Let $U \sim \mathcal{U}(0,1)$, and $F_X(x)$ the distribution function of X, then $F_X^{-1}(U)$ follows the same distribution of X.

Proof. It suffices to show that $F_X^{-1}(U) \stackrel{d}{=} X$. It suffices to check their cdf are the same:

$$\begin{aligned}\mathbb{P}(F^{-1}(U) \le x) &= \mathbb{P}(\inf\{y : F(y) \ge U\} \le x) \\ &= \mathbb{P}(F(x) \ge U) = F(x)\end{aligned}$$

■

■ **Example 1.2** Then we discuss how to generate i.i.d. exponential random variables with rate μ:

$$F(x) = 1 - e^{-\mu x} \implies F^{-1}(u) = -\log(1-u)/\mu.$$

Therefore, we can first generate U, and then output

$$X = -\log(1-U)/\mu \triangleq -\log U/\mu$$

■ **Example 1.3** The generation of i.i.d. discrete random variables is a little bit tricky:

$$\mathbb{P}(X = k) = p_k, \quad k = 1,2,3,\dots.$$

Sometimes we don't know the analytical form of the inverse distribution function, such as the normal distribution. Then statisticans consider the accpetance-rejection method.

Circle example.

We apply the similar idea to two distributions. We have a simple distribution $g(x)$ and a complicated distribution $f(x)$. First find $c > 0$ s.t. $f(x) \le g(x)$.

1. Generate $Y \sim g(x)$, which is our x-location;
2. Generate $U \sim \mathcal{U}(0,1)$. When $U \le f(x)/(cg(x))$, stop and return X; otherwise discard X and repeat step 1.

Here f is called the **target distribution**, and g the **proposed distribution**, the probability $\mathbb{E}_{x\sim g(x)}[f(x)/c(g(x))]$ is called the acceptance rate:

$$\mathbb{E}_{x\sim g(x)}[f(x)/c(g(x))] = \frac{1}{c}.$$

■ **Example 1.4** Consider the Beta distribution with density

$$f(x) = x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta), \quad x \in [0,1], \alpha,\beta > 1.$$

Since it is not easy to find the inverse distribution function, we select the proposed distribution $g \sim \mathcal{U}(0,1)$. We take $C = \max_{x\in[0,1]} f(x)$:

$$C = \max_{x\in[0,1]} x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta) \leq \frac{1}{B(\alpha,\beta)}$$

A simpler choice is $C = \frac{1}{B(\alpha,\beta)}$. The procedure is as follows:

1. Generate $U_1, U_2 \sim \mathcal{U}(0,1)$.
2. Compute the ratio $L = \frac{f(U_1)}{cg(U_1)} = U_1^{\alpha-1}(1-U_1)^{\beta-1}$.

■

■ **Example 1.5** Then we discuss how to generate normal distributions. Set $g(x) = ue^{-ux}$ and $f(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$. As a result,

$$C = \max_{0\leq x<\infty} \frac{1}{\sqrt{2\pi\mu}}\exp\left(-\frac{x^2}{2} + ux\right) = \frac{1}{\sqrt{2\pi\mu}}\exp\left(\frac{u^2}{2}\right)$$

with $u^* = \arg\min(\frac{1}{\sqrt{2\pi\mu}}e^{u^2/2})$. The ratio

$$L = \exp\left(-\frac{x^2}{2} + u^*x - \frac{(u^*)^2}{2}\right)$$

■

General rule: use a heavy tail distribution function.

More accpetance-rejection ideas.

$$f(x) \propto \tilde{f}(x).$$

Suppose that we can generate from density $h(x)$,

$$c^* = \max \frac{\tilde{f}(x)}{h(x)}$$

Squeezing. Suppose that the density $f(x)$ is difficult to evaluate. Suppose there exists two sequences of functions $h_n(x), g_n(x)$ such that

- $g_n(x)$ is a density and $f(x) \leq c g_n(x)$ for a fixed $c, \forall n$.
- $h_n(x) \leq f(x), \forall n$
- $g_n(x), h_n(x) \to f(x)$ pointwisely as $n \to \infty$.

$$\frac{\hat{h}_n(x)}{cg(x)} < \frac{f(x)}{cg(x)} < \frac{\bar{h}_n(x)}{cg(x)}$$

When $U < \frac{\hat{h}_n(x)}{cg(x)}$ or $U > \frac{\bar{h}_n(x)}{cg(x)}$; otherwise, $n \leftarrow n+1$.

Chapter 2

Week2

2.1. Wednesday

Box Muller Method. When $x_1, x_2 \sim \mathcal{N}(0,1)$, we find

$$r^2 = x_1^2 + x_2^2 \sim \mathcal{X}^2, \qquad \theta \sim \mathcal{U}(0,1).$$

It follows that

$$Y_1 \leftarrow \sqrt{-2\log U_1}\sin(2\pi U_2), \quad Y_2 \leftarrow \sqrt{-2\log U_1}\cos(2\pi U_2).$$

Alias. Let $X \in \{1,2,\dots,n\}$. The pre-computation process is to find n pairs $(x_{1,\ell}, x_{2,\ell}), (p_{1,\ell}, p_{2,\ell})$ for $\ell \in [n]$ s.t.

$$\sum_{\ell}\sum_{i=1}^{2} p_{i,\ell}\mathbf{1}\{x_{i,\ell} = k\} = np_k, \quad k \in [n].$$

These n pairs could be updated simply. Then we can view the distribution of X as a mixture of n two-point distributions.

- Generate a uniform $\ell \in [n]$;
- Generate a two point distribution s.t. $P(X = x_{i,\ell}) = p_{i,\ell}$;
- Return a sample for X.

2.1.1. Generating Multi-variate Random Variables

Simulating a multivariate normal distribution is simple. The random variable $\boldsymbol{X} \sim \mathcal{N}(0,\Sigma)$ can be obtained by

$$\boldsymbol{X} = \Sigma^{1/2}\boldsymbol{X}_0, \quad \text{with } \boldsymbol{X}_0 \sim \mathcal{N}(0,\boldsymbol{I})$$

The 2-dimension random variable can be obtained by

$$\begin{cases} X_1 = \sigma_1 Z_1 \\ X_2 = \rho\sigma_2 Z_1 + \sigma_2\sqrt{1-\rho^2}Z_2 \end{cases}$$

One way is to decompose the covariance matrix by Cholesky factorization.

Multi-nominal Distribution.

- Generate $X_1 \sim \text{binomial}(N, p_1)$
- Generate $X_2 \sim \text{binomial}(N - X_1, p_2/(1-p_1))$
- Keep the remaining proceed.

The idea is to sample the marginal distribution X_1, then sample the conditional distribution of X_i given $X_{1:i-1}$.

Copula. Characterize the dependence structure.

1. Generate a coupla $U = (U_1, \ldots, U_p)$
2. Compute $X_i = F_i^{-1}(U_i)$.

2.1.2. Simple Stochastic Process

Homogeneous Possion Process: $N(t) \sim \text{PP}(\beta)$ for $0 \le t \le T$.

- Use the inter-arrival time.
- First generate $N(T) \sim \text{Possion}(\beta T)$; then generate $N(t)$ uniformly at $[0,T]$.

Inhomogeneous Possion Process: $N(t) \sim \mathrm{PP}(\beta(t))$ for $\beta(t) \leq \beta$. By acceptance-rejection method, or

- $N(T) \sim \mathrm{Possion}(\int_0^T \beta(t)\,\mathrm{d}t)$;
- Generate $N(t)$ random variables by $c\beta(t)$ on $[0, T]$

The advantage of the second method is that it is easy to generalize into $[0, \infty)$.

Continuous Time Markov Chain. Let $J(t)$ be a Markov process with intensity matrix Λ.

- Simulate the holding time $-\lambda_{i,i}$
- Decide which state he aim to jump. For state j, w.p. $\lambda_{i,j}/-\lambda_{i,i}$.

2.2. Output Analysis

1. Make inference;
2. Access the performance of a simulation algorithm.

Normal Confidence Interval. We can use the sample average to approximate the expectation.

$$\sqrt{N}(\hat{z} - z) \to \mathcal{N}(0, \mathrm{Var}(z))$$

Then we can construct a $1 - \alpha$ confidence interval.

$$(\hat{z} - \phi_{1-\alpha/2}\frac{\sigma}{\sqrt{N}}, \hat{z} + \phi_{1-\alpha/2}\frac{\sigma}{\sqrt{N}}).$$