Stochastic Simulation - Concise Notes

Arnav Singh

April 14, 2024

Contents

1 Introduction

Definition 1.1 (Probability Mass Functions)**.** For a discrete random variable we define

$$p(x) = \mathbb{P}(X = x)$$

where $x \in X$.

Definition 1.2 (Measure and density)**.** Assume $X \subset \mathbb{R}$ and $X \in X$. Given random variable X we define measure of X as

$$\mathbb{P}(x_1 \leq X \leq x_2) = \mathbb{P}(X \in (x_1, x_2)) = \int_{x_1}^{x_2} f(x)\,dx$$

Definition 1.3 (Discrete Joint Probability Mass function)**.** Let X, Y random variables, and $\mathcal{X}, \mathcal{Y}$ the sets they live on, they are at most countable sets. The joint Probability Mass Function is

$$p(x, y) = \mathbb{P}(X = x, Y = y)$$

Definition 1.4 (Continuous Joint Probability Density Function)**.** Let X, Y random variables and $\mathcal{X}, \mathcal{Y}$ their ranges. The joint Probability Density Function is

$$\mathbb{P}(X \in A, Y \in B) = \int_A \int_B f(x, y)\,dx\,dy$$

Definition 1.5 (Discrete Conditional Probability Mass Function)**.** Let X, Y be random variables and $\mathcal{X}, \mathcal{Y}$ their ranges respectively. The conditional Probability Mass Function is

$$p(x|y) = \mathbb{P}(X = x|Y = y)$$

Definition 1.6 (Continuous Conditional Probability Density Function)**.** Let X, Y be random variables and $\mathcal{X}, \mathcal{Y}$ their ranges respectively. The conditional Probability Density Function is

$$p(y \mid x) = \frac{p(x, y)}{p(x)}$$

Here we have the conditional probability density function of Y given X

2 Exact Generation of Random Variates

Definition 2.1. A sequence of psuedo-random numbers $u_1, u_2, \ldots$ is a deterministic sequence of numbers whose statistical properties match a sequence of random numbers from a desired distribution.

2.1 Generating Uniform Random Variates

Definition (Linear Congruential Generator (LCG))**.** This method generates random numbers using a linear recursion

$$x_{n+1} \equiv ax_n + b \mod m$$

where x_0 is the seed, m the **modulus** of recursion, b the **shift** and a the **multiplier**.
If $b = 0$ then the generator is called a **multiplicative congruential generator**, and if $b \neq 0$ then it is called a **mixed congruential generator**.
We set m an integer and choose $a, b, x_0 \in \{0, \ldots, m-1\}$ and so we have $x_n \in \{0, 1, \ldots m-1\}$.
We then get the uniform numbers:

$$u_n = \frac{x_n}{m} \in [0, 1) \quad \forall n$$

2.2 Transformation Methods

Given pseudo-uniform random numbers, we can generate random numbers from other distributions using the following methods:

2.2.1 Inverse Transform Method

Theorem 2.1. *Consider random variable X with CDF F_X. Then the random variable $F_X^{-1}(U)$ where U is a uniform random variable on $[0, 1)$ has the same distribution as X.*

Algorithm 1: Psuedocode for inverse transform sampling

```
1. Input: number of samples $n$
2. for $i = 1, \ldots, n$ do
3.     Generate $U_i \sim U(0, 1)$
4.     Set $X_i = F_X^{-1}(U_i)$
5. end for
```

2.2.2 Tranformation Method

Algorithm 2: Psuedocode for transformation method

```
1. Input: number of samples $n$
2. for $i = 1, \ldots, n$ do
3.     Generate $U_i \sim U(0, 1)$
```

```
    Set $X_i = g(U_i)$
end for
```

Here choosing g is the crucial point.

2.2.3 Box-Muller Method

Box-Muller transfrom is a related transform to above, but provides a way to sample Gaussians directly from uniforms. In this case we just provide the algorithm.

Let $U_1, U_2, \sim U(0,1)$ be independent. Then the Box-Muller transform is

$$Z_1 = \sqrt{-2\log U_1}\cos(2\pi U_2)$$
$$Z_2 = \sqrt{-2\log U_1}\sin(2\pi U_2)$$

are independent standard normal random variables.

2.3 Rejection Sampling

Theorem 2.2 (Fundamental Theorem of Simulation)**.** *Drawing samples from one dimensional random variable X with density $\overline{p}(x) \propto p(x)$ is equivalent to sampling uniformly on the two dimensional region defined by*

$$A = \{(x, y) \in \mathbb{R}^2 : 0 \leq y \leq \overline{p}(x)\}$$

i.e. if (x', y') uniformly distributed on A then x' a sample from $p(x)$

2.3.1 Rejection Samples

Algorithm 3: Psuedocode for rejection sampling

```
Input: number of iterations $n$, and scaling factor $M$
for $i = 1, \ldots, n$ do
    Generate $X' \sim q(x')$
    Generate $U \sim U(0,1)$
    if $U \leq \frac{p(X')}{Mq(X')}$ then
        Accept $X'$
    end if
end for
return accepted samples
```

Definition. Denote the unnormalised density associated to $p(x)$ as $\overline{p}(x)$, we write

$$p(x) = \frac{\overline{p}(x)}{Z}, \quad Z = \int \overline{p}(x)\, dx$$

Algorithm 4: Psuedocode for rejection sampling without normalising constants

```
Input: number of iterations $n$, and scaling factor $M$
for $i = 1, \dots, n$ do
    Generate $X' \sim q(x')$
    Generate $U \sim U(0, 1)$
    if $U \leq \frac{\overline{p}(X')}{Mq(X')}$ then
        Accept $X'$
    end if
end for
return accepted samples
```

2.3.2 Acceptance Rate

Proposition 2.1. *When the target density $p(x)$ is normalised and M is prechosen, the acceptance ratio is given by*

$$\hat{a} = \frac{1}{M}$$

where $M > 1$ in order to satisfy the requirement that q covers p. For an unnormalised target density $\overline{p}(x)$ with the normalising constant $Z = \int \overline{p}(x)dx$ the acceptance rate is given as

$$\hat{a} = \frac{Z}{M}$$

2.3.3 Desigining the Optimal Rejection Sampler

Choosing M We see that we should choose M such that $Mq(x) \geq p(x) \forall x$. To choose smallest such M we should find M^* such that

$$M^* = \sup_x \frac{p(x)}{q(x)}$$

Optimising the proposal We optimise for the parameter θ of the proposal distribution q_θ.

$$\theta^* = \arg\min_\theta \log M_\theta$$

Use the log space as we obtain more tractable quantities

2.4 Composition

2.4.1 Sampling from Discrete Mixture Densities

Algorithm 5: Sampling Discrete Mixtures

```
1. The number of samples $n$
2. for $i = 1, \dots, n$ do
3.     Generate $k \sim p(k)$
4.     Generate $X_i \sim q_k(x)$
5. end for
```

Where we have

$$p(x) = \sum_{k=1}^{W} w_k q_k(x), \quad p(k) = w_k, \sum_{k=1}^{K} p(k) = 1$$

2.5 Sampling Multivariate Densities

2.5.1 Sampling a Multivariate Gaussian

Define $x \in \mathbb{R}^d$ a multivariate Gaussian

$$p(x) = (2\pi)^{-d/2} \det(\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

where $\mu \in \mathbb{R}^d$ is the mean and $\Sigma \in \mathbb{R}^{d \times d}$ is a $d \times d$ symmetric positive definite matrix. In univariate case, $Y = \mu + \sigma X$ gave us samples from $\mathcal{N}(\mu, \sigma^2)$, we now generalise this to the multivariate case.

$$Y = \Sigma^{1/2} X + \mu$$

Computing $\Sigma^{\frac{1}{2}}$ using Cholesky decomposition.

Algorithm 6: Sampling Multivariate Gaussian

```
Input: number of samples $n$,
for $i = 1, \dots, n$ do
    Compute $L$ such that $\Sigma = LL^T$ (Cholesky decomposition)
    Draw $d$ univariate independent normals $\nu_k \sim \mathcal{N}(0,1)$ to form vector $\nu = (\nu_1, \dots, \nu_d)$
    Generate $x_i = \mu + L\nu$
end for
```

3 Probabilistic Modelling and Inference

3.2 The Bayes Rule and it's Uses

Definition 3.1 (Bayes Theorem)**.** Let X, Y be random variables, with associated densities $p(x), p(y)$ respectively. Bayes rulse is given by

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$$

3.3 Conditional Independence

Definition 3.2. Let X, Y and Z be random variables. Say that X and Y are conditionally independent given Z if

$$p(x, y \mid z) = p(x \mid z)p(y \mid z)$$

Corollary 3.1. *If X, Y are conditionally independent given Z then*

$$p(x \mid y, z) = p(x \mid z) \quad \text{and } p(y \mid x, z) = p(y \mid z)$$

Proposition 3.1. *Let X, Y and Z be random variables. If X and Y are conditionally independent given Z then*

$$p(x, y, z) = p(x \mid z)p(y \mid z)p(z)$$

Proposition 3.2. *Given X, Y, Z without any conditional independence assumptions, the conditional Bayes rules is*

$$p(x \mid y, z) = \frac{p(y \mid x, z)p(x \mid z)}{p(y \mid z)}$$

Definition (Marginal Likelihood)**.** The marginal likelihood is given by

$$p(y) = \int p(y \mid x)p(x)\, dx$$

4 Monte Carlo Integration

Given a probability density function $p(x)$ we are interested in computing expectations of the form

$$\overline{\varphi} = \mathbb{E}_p[\varphi(x)] = \int \varphi(x)p(x)\,dx$$

where φ called a **test function**.

Definition (Dirac Delta Measure)**.** We define it as

$$f(y) = \int f(x)\delta_y(x)\ dx, \quad \delta_y(x) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

We can think of the dirac as a point mass at y

Proposition 4.1. *Let $X_1, \dots, X_n$ be i.i.d samples. Then the Monte Carlo estimator*

$$\hat{\varphi}^N = \frac{1}{N}\sum_{i=1}^{N} \varphi(X_i)$$

is unbiased, i.e.

$$\mathbb{E}[\hat{\varphi}^N] = \overline{\varphi}$$

Proposition 4.2. *Let $X_1, \dots, X_n$ be iid samples from p. Then the Monte Carlo estimator*

$$\hat{\varphi}^N = \frac{1}{N}\sum_{i=1}^{N} \varphi(X_i)$$

has variance

$$Var[\hat{\varphi}^N] = \frac{1}{N}\left(var_p[\varphi(X)]\right)$$

where

$$var_p[\varphi(X)] = \int (\varphi(x) - \overline{\varphi})^2 p(x)\,dx$$

4.2 Error Metrics

Definition (Bias)**.** The bias of an estimator is defined as

$$\text{Bias}[\hat{\varphi}^N] = \mathbb{E}[\hat{\varphi}^N] - \underbrace{\overline{\varphi}}_{\text{True value}}$$

Definition (Mean Squared Error)**.** The mean squared error of an estimator is defined as

$$\mathrm{MSE}[\hat{\varphi}^N] = \mathbb{E}[(\hat{\varphi}^N - \overline{\varphi})^2]$$

we have that

$$\mathrm{MSE}[\hat{\varphi}^N] = \mathrm{Var}[\hat{\varphi}^N] + \mathrm{Bias}[\hat{\varphi}^N]^2$$

and also the Root Mean Squared Error is

$$\mathrm{RMSE}[\hat{\varphi}^N] = \sqrt{\mathrm{MSE}[\hat{\varphi}^N]}$$

Definition (Relative Absolute Error)**.** The relative absolute error is defined as

$$\mathrm{RAE}[\hat{\varphi}^N] = \frac{|\hat{\varphi}^N - \overline{\varphi}|}{|\overline{\varphi}|}$$

4.3 Importance Sampling

Algorithm 7: Basic Importance Sampling

```
1. Input: number of samples $N$
2. for $i = 1, \dots, N$ do
3.     Generate $X_i \sim q(x)$
4.     Compute importance weights $w_i = \frac{p(X_i)}{q(X_i)}$
5. end for
6. Compute the estimate
       $$\hat{\varphi}^N = \frac{1}{N}\sum_{i=1}^{N} w_i \varphi(X_i)$$
```

Proposition 4.3. *The estimator $\hat{\varphi}_{IS}^N$ is unbiased, i.e.*

$$\mathbb{E}[\hat{\varphi}_{IS}^N] = \overline{\varphi}$$

Proposition 4.4. *Variance of estimator $\hat{\varphi}_{IS}^N$ is given by*

$$Var[\hat{\varphi}_{IS}^N] = \frac{1}{N}\left(\mathbb{E}_q[w^2(X)\varphi^2(X)] - \overline{\varphi}^2\right)$$

Psuedocode for self-normalised importance sampling

```
Input: number of samples $N$
for $i = 1, \dots, N$ do
    Generate $X_i \sim q(x)$
    Compute importance weights $W_i = \frac{\overline{p}_(X_i)}{q(X_i)}$
    Normalise:
        $\bar{w}_i = \frac{W_i}{\sum_{i=1}^N W_i}$
end for
Compute the estimate
    $\hat{\varphi}^N_{SNIS} = \sum_{i=1}^N \bar{w}_i \varphi(X_i)$
```

Common numerical trick is to use the log-sum-exp trick to avoid numerical instability.

$$\begin{aligned}
\log W_i &= \log \overline{p}(X_i) - \log q(X_i) \\
\log \widetilde{W}_i &= \log \overline{p}(X_i) - \log q(X_i) - \max \log W_i \\
\bar{w}_i &= \frac{\exp(\log \widetilde{W}_i)}{\sum_{i=1}^N \exp(\log \widetilde{W}_i)}
\end{aligned}$$

Proposition 4.5. *The marginal likelihood estimator given by*

$$p^N(y) = \frac{1}{N} \sum_{i=1}^N W_i$$

is an unbiased estimator of the marginal likelihood $p(y)$

Definition 4.1 (Effective Sample Size)**.** To measure the sample efficiency, one measure that is used in the literature is the effective sample size (ESS) which is given by

$$ESS_N = \frac{1}{\sum_{i=1}^N \overline{w}_i^2}$$

for the SNIS estimator.

5 Markov Chain Monte Carlo

Definition 5.1 (Markov Chain)**.** A discrete Markov Chain is a sequence of random variables $X_1, X_2, \ldots$ such that

$$\mathbb{P}(X_{n+1} = x_{n+1} \mid X_n = x_n, \ldots, X_1 = x_1) = \mathbb{P}(X_{n+1} = x_{n+1} \mid X_n = x_n)$$

Definition 5.2 (Transition Matrix)**.** The transition matrix of a Markov Chain is a matrix M such that

$$M_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i)$$

Definition (Chapman-Kolmogorov Equation)**.** The Chapman-Kolmogorov equation is given by

$$\mathbb{P}(X_{n+1} = j \mid X_1 = i) = \sum_k \mathbb{P}(X_{n+1} = j \mid X_n = k)\mathbb{P}(X_n = k \mid X_1 = i)$$

$$M^{m+n} = M^m M^n$$

Definition (Reccurent and Transient States)**.** A state $i \in X$ is **recurrent** if for

$$\tau_i = \inf\{n \geq 1 : X_n = i\} \quad \text{(the return time)}$$

we have

$$\mathbb{P}(\tau_i < \infty \mid X_0 = i) = 1$$

A state is **transient** if it is not recurrent.
We say i positively recurrent if

$$\mathbb{E}[\tau_i \mid X_0 = i] < \infty$$

If a chain recurrent but not positive recurrent, it is null recurrent.

Definition (Stationary Distribution)**.** A distribution π is stationary for a Markov Chain if

$$\pi = \pi M$$

Also called the invariant distribution.

Theorem 5.1. *If M is irreducible, then M has a unique invariant distribution if and only if it is positive recurrent.*

Definition (Periodicity)**.** A state i is aperiodic if

$$\{n > 0 : \mathbb{P}(X_{n+1} = i \mid X_1 i) > 0\}$$

has greatest common divisor 1.
A Markov Chain is aperiodic if all states are aperiodic.

Definition (Ergodicity)**.** A Markov Chain is ergodic if it is irreducible, aperiodic and positive recurrent.
If a chain $(X_n)_{n\in\mathbb{N}}$ is ergodic with initial distribution p_0 and invariant distribution $p^\star$ then

$$\lim_{n\to\infty} \mathbb{P}(X_n = i) = p^\star(i)$$

Moreover, for $i, j \in X$

$$\lim_{n\to\infty} \mathbb{P}(X_n = i \mid X_1 = j) = p^\star(i)$$

5.2 Continuous State Space Markov Chains

Definition. A continuous state space Markov Chain is a sequence of random variables $X_1, X_2, \dots$ such that

$$\mathbb{P}(X_{n+1} \in A \mid X_n = x_n, \dots, X_1 = x_1) = \mathbb{P}(X_{n+1} \in A \mid X_n = x_n)$$

where X an uncountable set, and denote by $K(x \mid x')$ the transition kernel.

Definition 5.3 (K-Variance)**.** Probability measure $p_\star$ is called K-invariant if

$$p_\star(x) = \int_X K(x \mid x') p_\star(x') \, dx'$$

Definition 5.4 (Detailed Balance)**.** A transition kernel K satisfies detailed balance with respect to a probability measure $p_\star$ if

$$K(x' \mid x) p_\star(x) = K(x \mid x') p_\star(x')$$

Proposition 5.1 (Detailed balance implies stationarity)**.** *If K satisfies detailed balance, then $p_\star$ is the invariant distribution*

5.3 Metropolis-Hastings Algorithm

Algorithm 9: Metropolis-Hastings Algorithm

```
1. Input: number of samples $N$
2. for $i = 1, \dots, N$ do
3.     Propose sample $X' \sim q(x' \mid X_{i-1})$
4.     Accept sample $X'$ with probability
           $\alpha(X_{n-1}, X') = \min\left(1, \frac{p(X')q(X_{n-1} \mid X')}{p(X_{n-1})q(X' \mid X_{i-1})}\right)$
```

```
    Otherwise reject sample and set $X_n = X_{n-1}$
end for
Discard first burn-in samples and return the rest
```

Definition. Define the acceptance ratio as

$$r(x, x') = \frac{p(x')q(x \mid x')}{p(x)q(x' \mid x)}$$

Proposition 5.2 (Metropolis-Hastings satisfies detailed balance)**.** *The Metropolis-Hastings algorithm satisfies detailed balance with respect to the target distribution $p_\star$ i.e.*

$$p_\star(x)K(x \mid x') = p_\star(x')K(x' \mid x)$$

where K is the kernel defined by the Metropolis-Hastings algorithm.

Algorithm 10: Metropolis-Hastings method for Bayesian Inference

```
Input: number of samples $N$, and starting point $X_0$
for $i = 1, \ldots, N$ do
    Propose sample $X' \sim q(x' \mid X_{i-1})$
    Accept sample $X'$ with probability
        $\alpha(X_{n-1}, X') = \min\left(1, \frac{\overline{p}_\star(x')q(x_{n-1} \mid x')}{\overline{p}_\star(x_{n-1})q(x' \mid x_{n-1})}\right)$
    Otherwise reject sample and set $X_n = X_{n-1}$
end for
Discard first burn-in samples and return the rest
```

Algorithm 11: Gibbs Sampler

```
Input: number of samples $N$, and starting point $X_0$
for $i = 1, \ldots, N$ do
```

```
    Sample
        $X_{n,1} \sim p_{1,\star}(X_{n,1} \mid X_{n-1,,2}, \dots, X_{n-1,d})$
        $X_{n,2} \sim p_{2,\star}(X_{n,2} \mid X_{n,1}, X_{n-1,3}, \dots, X_{n-1,d})$
        $\vdots$
        $X_{n,d} \sim p_{d,\star}(X_{n,d} \mid X_{n,1}, \dots, X_{n,d-1})$
end for
Discard first burn-in samples and return the rest
```

Proposition 5.3. *The Gibbs kernel K leaves the target distribution $p_\star$ invariant.*

Algorithm 12: Random Scan Gibbs Sampler

```
Input: number of samples $N$, and starting point $X_0$
for $i = 1, \dots, N$ do
    Sample $j \sim \{1, \dots, d\}$
        $X_{n,j} \sim p_{j,\star}(X_{n,j} \mid X_{n,1}, \dots, X_{n,j-1}, X_{n,j+1}, \dots, X_{n,d})s$
end for
Discard first burn-in samples and return the rest
```