The Monte-Carlo Method

The Monte-Carlo method is an approach to solving a common problem that arises in many contexts. We have a stochastic model in which there is a random object X that takes values in some space $\mathcal{X}$. This space is typically some finite set, or the real line $\mathbb{R}$ or some finite-dimensional Euclidean space $\mathbb{R}^k$, but we will encounter a very natural situations in which we can talk about a random continuous path in the plane, or a random function on the real line.

Examples.

- **Coin tossing** - Here, in a single toss, the random object is X takes the value 0 or 1, so $\mathcal{X} = \{0, 1\}$. The probability distribution of X is determined by the constant $p = P[X = 1]$. If we toss a coin repeatedly, say n times then our random object $X = (X_1, \dots, X_n)$ is an element of $\mathcal{X} = \{0, 1\}^n$, the set of binary vectors of length n.
- **Shuffled deck of cards** - Here the random object is $X = (c_{\pi_1}, c_{\pi_2}, \dots, c_{\pi_{52}})$ where the c_i are the labels for the 52 cards and $(\pi_1, \dots, \pi_{52})$ is a random permutation of 1,2,...,52 e.g. uniformly distributed among the 52! possibilities. Here we can take $\mathcal{X} = S_{52}$ the space of permutations of $(1, 2, \dots, 52)$.
- **Stochastic differential equation solution** - $X = \{X_t, t \in [0, T]\}$ a random function with some generative distribution. (More on this soon.) Here we might take $\mathcal{X} = \mathcal{C}[0, T]$ the space of continuous functions on $[0, T]$.

Some standard types of questions we might ask in the examples above.

For coin tossing, what is the chance of seeing a *run* of 15 consecutive heads in 100 flips of a coin. On average, what is the length of the longest run?

For a shuffled deck of cards, how likely is it that when we split the deck into 4 bridge hands, all of the aces end up in one of the hands? On average, how many suits are represented in a given hand?

For a random function $\{X_t, t \in [0, T]\}$ what is the chance it exceeds some level τ_U and falls below some level τ_L? What is the expected value of

$$\int_{t=0}^{T} X_t^2 dt?$$

In each case above, we want to determine the value of $\mu = E[g(X)]$ for some $g : \mathcal{X} \to \mathbb{R}$. Note that as a special case, $g : \mathcal{X} \to \{0, 1\}$ i.e. g is the indicator function I_A where

$A = \{x \in \mathcal{X} \ : \ g(x) = 1\}$, in which case

$$E[g(X)] = P[g(X) = 1]1 + P[g(X) = 0] = P[g(X) = 1] = P[X \in A].$$

So determining the probability that X falls in some particular set A is a special case of determining an expectation.

The Monte-Carlo Method is based on the strong law of large numbers. In its simplest form we have a method for generating a sequence of iid random variables $X^{(1)}, \ldots, X^{(n)}$ whose distribution is the same as that of X. Then we estimate $\mu = E[g(X)]$ using

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} g(X^{(i)}).$$

The strong law of large numbers tells us that, assuming μ is finite, $\hat{\mu}_n \to \mu$ with probability 1 as $n \to \infty$. In addition, the central limit theorem allows us to quantify the error assuming $\sigma^2 = \mathrm{Var}(g(X)) < \infty$ we have

$$\sqrt{n}(\hat{\mu} - \mu) \overset{\mathcal{L}}{\to} N(0, \sigma^2),$$

so that the interval $\hat{\mu} \pm z_{\alpha/2}\sigma/\sqrt{n}$ will contain μ with probability approximately $1 - \alpha$ for large n. [1]

The interesting part of implementing the Monte-Carlo method is in coming up with iid *realizations* of X. It is typical in a computing environment to have a function, which, when called returns a value that can be viewed as a Uniform$(0, 1)$ random variable [2] and which, when called repeatedly, produces a sequence of values that can be viewed as and iid Uniform$(0, 1)$. We view the *random number generator* that generates pseudo-random iid Uniform$(0, 1)$'s as a primitive tool that can be utilized to produce more complex random objects.

Some random variables X lend themselves to a simple generation method. The most straightforward approach is the following.

Inversion method. Given a strictly increasing continuous cdf F_X if we want to generate $X \sim F_X$ we can take $X = F_X^{-1}(U)$. This works because

$$P[X \le x] = P[F_X^{-1}(U) \le x] = P[U \le F_x(x)] = F_X(x).$$

Example. Suppose we want to generate $X \sim$ Exponential(λ) so that $F_X(x) = 1 - e^{-\lambda x}$ for $x > 0$. Then $F_X^{-1}(u) = -\log(1 - u)/\lambda$ so $X = -\log(1 - U)/\lambda \sim$ Exponential(λ) if $U \sim$ Uniform$(0, 1)$.

[1] Here, the quantity $z_{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution, i.e. $z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2)$, where Φ denotes the standard normal cdf.

[2] Or a uniformly distributed integer K in some range $\{0, 1, \ldots, 2^d - 1\|$ which can be transformed to Uniform$(0, 1)$ by taking $U = K/2^d$.

While the inversion method applies in many situations, but it is not practical in others. For example if we want to generate $X \sim N(\mu, \sigma^2)$ the cdf and its inverse are not available in closed-form, and a different method for generating X using a uniform random number generator is required. Another very useful method is the Acceptance-Rejection method.

Acceptance-Rejection method. Suppose f is a probability density function and we want to sample $X \sim f$. Assume that for some probability density function g that it is easy to sample from we have $f(x) \leq cg(x)$ for all x where c is a constant. Then the following algorithm yields $X \sim f$.

```
Repeat
    Generate X distributed according to g
    Independently generate U distributed uniformly in (0,1)
Until cUg(X) <= f(X)
    Return X
```

it can be shown that the expected number of iterations of this algorithm is c so we try to find c as small as possible.

Example. Standard normal distribution. If we want to generate $X \sim N(0,1)$, the pdf to sample from is $f(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$. Define $g(x) = \frac{1}{2}\exp(-|x|)$, the double-exponential/Laplace distribution. This is easy to sample from. Take $Y \sim$ Exponential(1) using inversion, generate $V \sim$ Uniform$(0,1)$ independent of Y then take $X = Y$ if $V \leq 1/2$ and $X = -Y$ if $V > 1/2$. Then $X \sim g$.

Now to apply the acceptance-rejection method we need to find c so that $f(x) \leq cg(x)$ for all x. So we need

$$c \geq \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)/(\frac{1}{2}\exp(-|x|)).$$

To maximize the right-hand side in x we can, by symmetry, focus on $x > 0$ and take

$$c = \max_x \frac{2}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2 - x). = \max_x \frac{2}{\sqrt{2\pi}} \exp(-\frac{1}{2}(x-1)^2 + \frac{1}{2}) = \sqrt{\frac{2e}{\pi}} < 1.32.$$

Explanation of the Acceptance-Rejection Method

To explain why the acceptance-rejection method works, we rely on the following facts.

(1) If g is a probability density function, then to sample a point (X, Y) uniformly in the set $\mathcal{D}_{cg} = \{(x, y) : -\infty < x < +\infty, \text{ and } 0 \leq y \leq cg(x)\}$ we can take $X \sim g$ and

conditionally, given X, take Y uniformly distributed in the line segment $[0, cg(X)]$. [3]

(2) Suppose we have two domains $\mathcal{D}_1 \subseteq \mathcal{D}_2 \subseteq \mathbb{R}^2$ having positive area, and we repeatedly sample points $(X^{(n)}, Y^{(n})$ iid uniformly distributed in $\mathcal{D}_2$ until we get a point $(X^{(N)}, Y^{(N)})$ in $\mathcal{D}_1$ then $(X^{(N)}, Y^{(N)})$ is uniformly distributed in $\mathcal{D}_1$. In addition, N has a Geometric(p) distribution: $P[N = k] = p(1-p)^{k-1}$ for $k = 1, 2, \ldots,$ where $p = |\mathcal{D}_1|/|\mathcal{D}_2|$ so that $E[N] = 1/p$.

(3) If f is a probability density function and we sample a point (X, Y) uniformly in the region $\{(x, y) \ : \ -\infty < x < +\infty, \text{ and } 0 \leq y \leq f(x)\}$. Then $X \sim f$.

[3]A random vector (X, Y) taking values in $\mathbb{R}^2$ is said to be uniformly distributed in a domain $\mathcal{D} \subseteq \mathbb{R}^2$ if $P[(X, Y) \in \mathcal{D}] = 1$ and for any subset $A \subseteq \mathcal{D}$ we have $P[(X, Y) \in A] = |A|/|\mathcal{D}|$ where $|A|$ denotes the area of A.