Monte Carlo introduction

Brooks Paige

Week 2

Estimating average height (1/2)

How would you estimate the average height of everyone in COMP0171?

1. Get a ruler, and measure everyone's height $h_1, \ldots, h_M$, where $M \approx 80$ students
2. Compute $\bar{h} = \frac{1}{M} \sum_{i=1}^{M} h_i$

The only error is measurement error.

Estimating average height (2/2)

How would you estimate the average height of everyone in the UK?

1. Get a ruler, and measure everyone's height $h_1, \ldots, h_M$, where $M \approx 67,000,000$
2. Oops

Estimating average height (2/2)

How would you estimate the average height of everyone in the UK?

1. Get a ruler, and measure everyone's height $h_1, \ldots, h_M$, where $M \approx 67,000,000$
2. Oops

This is where **Monte Carlo** estimates make sense. Instead of measuring all M people, we measure a **sample** of $N \ll M$ people, drawn at random from the population, and average over them instead.

Estimating average height (2/2)

How would you estimate the average height of everyone in the UK?

1. Get a ruler, and measure everyone's height $h_1, \ldots, h_M$, where $M \approx 67,000,000$
2. Oops

This is where **Monte Carlo** estimates make sense. Instead of measuring all M people, we measure a **sample** of $N \ll M$ people, drawn at random from the population, and average over them instead.

Now, we have both measurement error, and "finite-sample" error (which decreases as $N \to M$).

Basic Monte Carlo identity

Suppose we are trying to compute the expectation of $f(\mathbf{x})$ under $p(\mathbf{x})$,

$$\mathbb{E}[f] = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int p(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}.$$

If we have a way of drawing samples from $p(\mathbf{x})$, we can approximate this with

$$\hat{f} = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}); \qquad \mathbf{x}^{(i)} \sim p(\mathbf{x})$$

Basic Monte Carlo identity

Suppose we are trying to compute the expectation of $f(\mathbf{x})$ under $p(\mathbf{x})$,

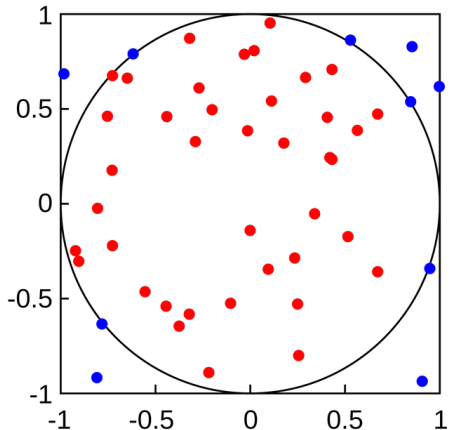
$$\mathbb{E}[f] = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int p(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}.$$

If we have a way of drawing samples from $p(\mathbf{x})$, we can approximate this with

$$\hat{f} = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}); \qquad \mathbf{x}^{(i)} \sim p(\mathbf{x})$$

This is an unbiased estimator, meaning $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$. It becomes exact as N approaches ∞.

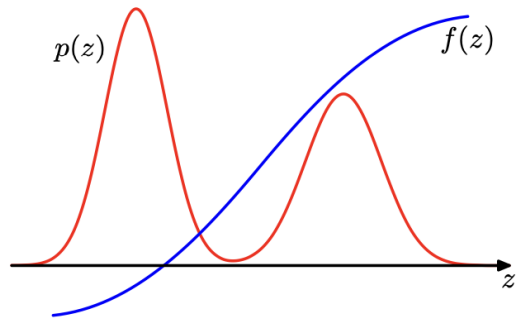
Monte Carlo example



Compute the area of the circle by counting the fraction of points landing inside

Figure: wikipedia

Monte Carlo example



We want to compute the expectation of $f(z)$ under the distribution $p(z)$

Figure: Bishop, Ch. 11

Monte Carlo unbiasedness

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

- This expectation is over all random choices in the estimator.

Monte Carlo unbiasedness

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{x}^{(i)})}\left[f(\mathbf{x}^{(i)})\right]$$

- This expectation is over all random choices in the estimator.
- In general, expectation is a **linear operator**:

$$\mathbb{E}_{p(\mathbf{x})}[a\mathbf{x} + b] = a\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}] + b$$

Monte Carlo unbiasedness

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{x}^{(i)})}\left[f(\mathbf{x}^{(i)})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[f]$$

- This expectation is over all random choices in the estimator.
- In general, expectation is a **linear operator**:

$$\mathbb{E}_{p(\mathbf{x})}[a\mathbf{x} + b] = a\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}] + b$$

- Each $\mathbf{x}^{(i)}$ is assumed to be drawn independently from $p(\mathbf{x})$

Monte Carlo unbiasedness

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{x}^{(i)})}\left[f(\mathbf{x}^{(i)})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[f] = \mathbb{E}[f]$$

- This expectation is over all random choices in the estimator.
- In general, expectation is a **linear operator**:

$$\mathbb{E}_{p(\mathbf{x})}[a\mathbf{x} + b] = a\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}] + b$$

- Each $\mathbf{x}^{(i)}$ is assumed to be drawn independently from $p(\mathbf{x})$

Variance

Variance is a measure of dispersion. It is the expected value of the squared difference of a random variable and its mean.

$$\mathrm{Var}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])^2\right] = \mathbb{E}[\mathbf{x}^2] - \mathbb{E}[\mathbf{x}]^2$$

(Note: the second form can be easier to work with mathematically, but can be numerically unstable due to floating-point issues!)

Properties of Variance

Some useful properties of variance, for a random variable $\mathbf{x}$ and scalar a:

$$\mathrm{Var}[\mathbf{x}] \geq 0 \qquad \mathrm{Var}[a] = 0$$

$$\mathrm{Var}[\mathbf{x} + a] = \mathrm{Var}[\mathbf{x}] \qquad \mathrm{Var}[a\mathbf{x}] = a^2 \,\mathrm{Var}[\mathbf{x}]$$

Properties of Variance

Some useful properties of variance, for a random variable $\mathbf{x}$ and scalar a:

$$\mathrm{Var}[\mathbf{x}] \geq 0 \qquad \mathrm{Var}[a] = 0$$
$$\mathrm{Var}[\mathbf{x} + a] = \mathrm{Var}[\mathbf{x}] \qquad \mathrm{Var}[a\mathbf{x}] = a^2 \,\mathrm{Var}[\mathbf{x}]$$

If we have two different random variables $\mathbf{x}, \mathbf{y}$, their **covariance** is defined as

$$\mathrm{Cov}[x, y] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])] = \mathbb{E}[\mathbf{xy}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}].$$

Note that $\mathrm{Cov}[x, x] = \mathrm{Var}[x]$. Also, $\mathrm{Cov}[x, y] = 0$ if $\mathbf{x} \perp \mathbf{y}$.

Properties of Variance

Some useful properties of variance, for a random variable $\mathbf{x}$ and scalar a:

$$\mathrm{Var}[\mathbf{x}] \geq 0 \qquad \mathrm{Var}[a] = 0$$
$$\mathrm{Var}[\mathbf{x} + a] = \mathrm{Var}[\mathbf{x}] \qquad \mathrm{Var}[a\mathbf{x}] = a^2 \,\mathrm{Var}[\mathbf{x}]$$

If we have two different random variables $\mathbf{x}, \mathbf{y}$, their **covariance** is defined as

$$\mathrm{Cov}[x, y] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])] = \mathbb{E}[\mathbf{xy}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}].$$

Note that $\mathrm{Cov}[x, x] = \mathrm{Var}[x]$. Also, $\mathrm{Cov}[x, y] = 0$ if $\mathbf{x} \perp \mathbf{y}$.

Sums and differences:

$$\mathrm{Var}[a\mathbf{x} + b\mathbf{y}] = a^2 \,\mathrm{Var}[\mathbf{x}] + b^2 \,\mathrm{Var}[\mathbf{y}] + 2ab \,\mathrm{Cov}[\mathbf{x}, \mathbf{y}]$$
$$\mathrm{Var}[a\mathbf{x} - b\mathbf{y}] = a^2 \,\mathrm{Var}[\mathbf{x}] + b^2 \,\mathrm{Var}[\mathbf{y}] - 2ab \,\mathrm{Cov}[\mathbf{x}, \mathbf{y}]$$

Monte Carlo variance

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

Monte Carlo variance

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N^2}\,\mathrm{Var}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

Monte Carlo variance

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N^2}\,\mathrm{Var}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

$$= \frac{1}{N^2} \times N\,\mathrm{Var}\left[f(\mathbf{x})\right]$$

Monte Carlo variance

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N^2}\,\mathrm{Var}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

$$= \frac{1}{N^2} \times N\,\mathrm{Var}\left[f(\mathbf{x})\right]$$

$$= \frac{1}{N}\mathbb{E}\left[(f - \mathbb{E}[f])^2\right]$$

Monte Carlo variance

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right] = \frac{1}{N^2}\,\mathrm{Var}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\right]$$

$$= \frac{1}{N^2} \times N\,\mathrm{Var}\left[f(\mathbf{x})\right]$$

$$= \frac{1}{N}\mathbb{E}\left[(f - \mathbb{E}[f])^2\right]$$

So, as $N \to \infty$, the variance goes to zero.

Why Monte Carlo?

1. Only a small number of specific integrals are analytically tractable. We can Monte Carlo estimate all sorts of stuff

2. We're all in a computer science department — we're probably better at debugging our code, than debugging our math

3. Simulation methods are interactive and allow for rapid prototyping and testing of models and ideas

4. The variance of the estimator doesn't depend on the dimensionality of $\mathbf{x}$

How to sample from a distribution

Most common distributions we will use as 'building blocks' have readily implemented sampling algorithms available for us.

e.g.: `dist.Beta(5, 4).sample()`

In general, assume there are a small number of "primitive" distributions we know how to sample from.

- *(that is, they have a* `.sample` *function defined in our software environment!)*

Pseudo-random numbers

Getting "real" random samples on a computer is hard. Instead, we use a pseudo-random number generator that creates a deterministic sequence of values that "looks random", given a "seed".

```
>>> np.random.seed(0)
>>> np.random.rand(5)
array([0.5488135 , 0.71518937, 0.60276338, 0.54488318, 0.4236548 ])
```

The upside is that this means results are repeatable.

These methods typically produce a sequence of integers, or (normalized) a sequence of floating point numbers on the range $[0, 1]$.

Transforming samples

If the pseudo-random number generator produces samples $x^{(i)} \sim \mathcal{U}([0, 1])$, how do we get other values?

If we have a target distribution $p(y)$, one option is to define some transformation $T(x)$ such that the values $y = T(x)$ are distributed according to $p(y)$.

Transforming samples

If the pseudo-random number generator produces samples $x^{(i)} \sim \mathcal{U}([0, 1])$, how do we get other values?

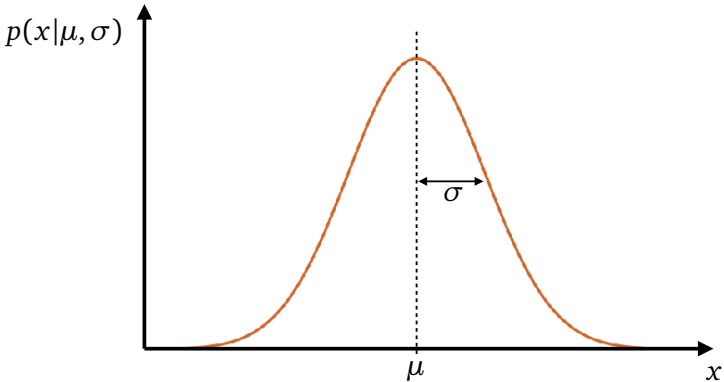
If we have a target distribution $p(y)$, one option is to define some transformation $T(x)$ such that the values $y = T(x)$ are distributed according to $p(y)$.

For some distributions this is easy: for example, if we want to sample $y \sim \mathcal{U}([-1, 1])$,

$$T(x) = 2x - 1$$

which just requires shifting and scaling.

Normal distribution



$$p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**.

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**. **Why does this work?**

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**. **Why does this work?**

$$p(F_X^{-1}(u) < x)$$

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**. **Why does this work?**

$$p(F_X^{-1}(u) < x) = p(u < F_X(x))$$

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**. **Why does this work?**

$$p(F_X^{-1}(u) < x) = p(u < F_X(x)) = F_X(x)$$

How do we sample from this?

Recall: the **CDF** F_X is a monotonic function that defines the probability $p(X < x)$.

For distributions that have a tractable **inverse CDF**, F_X^{-1}, we can use the following algorithm:

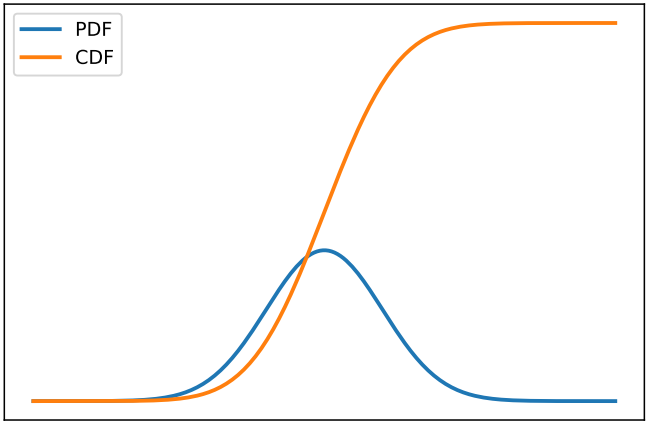
1. Sample $u^{(i)} \sim \mathcal{U}([0, 1])$
2. Compute $x^{(i)} = F_X^{-1}(u^{(i)})$

This is called **inverse transform sampling**. **Why does this work?**

$$p(F_X^{-1}(u) < x) = p(u < F_X(x)) = F_X(x)$$

That is, the CDF of the transformed uniform random variable is exactly the CDF of the target distribution.

Inverse transform sampling: visual aid



Inverse transform sampling: visual aid

Animation: wikipedia