MA3227 Numerical Analysis II

Lecture 8: Monte Carlo Methods

Simon Etter

Semester II, AY 2020/2021

Monte Carlo Methods

Problem statement

Estimate the expectation $\mathbb{E}[X]$ of a random variable X by computing the average of a large number of samples of X.

This problem statement raises several questions.

- *Why compute expectations?*

  According to basic probability theory, the expectation $\mathbb{E}[X]$ of a random variable X which assumes values x in a discrete or continuous set V with probability $p(x)$ is given by, respectively,

  $$\mathbb{E}[X] = \sum_{x \in V} x\, p(x) \qquad \text{or} \qquad \mathbb{E}[X] = \int_V x\, p(x)\, dx.$$

  Computing an expectation is hence the same as evaluating a particular sum or integral, and conversely any sum or integral can be reinterpreted as an expectation.

- *Why compute expectations through sampling?*

  This question is best answered by means of some examples; see the following slides.

Monte Carlo Methods

Example 1: Winning probabilities in m, n, k-games

Consider the *m, n, k-game* described under

`https://en.wikipedia.org/wiki/m,n,k-game`.

Assume we want to compute the probability P of a win for player 1 assuming random moves on behalf of both players.

This probability could be computed as the sum

$$P = \sum_{\text{all possible games}} [\text{ probability of game }] \times \begin{cases} 1 & \text{if player 1 wins,} \\ 0 & \text{otherwise,} \end{cases}$$

but this sum contains roughly $(mn)!$ terms and therefore cannot be evaluated except for very modest values of m and n.

For example, if we assume a runtime of just 1 nanosecond per game, then evaluating the above sum for $m = n = 4$ would take about 6 hours, and evaluating the sum for $m = n = 5$ would take about 500'000 years!

Monte Carlo Methods

Example 1: Winning probabilities in m, n, k-games (continued)

These ludicrous runtimes can be avoided if we rewrite the above sum as the expectation

$$P \approx \mathbb{E}[X] \qquad \text{where} \qquad X = \begin{cases} 1 & \text{if player 1 wins,} \\ 0 & \text{otherwise} \end{cases}$$

and then estimate this expectation as follows.

- Play out N random games.
- For each such game i, record in the variable $X_i \in \{0, 1\}$ whether player 1 won.
- Estimate $\mathbb{E}[X] \approx \frac{1}{N} \sum_{i=1}^{N} X_i$.

This approach is known as *Monte Carlo sampling*, and `mnk_probabilities()` shows that it leads to reasonably accurate estimates already for moderate values of N.

Monte Carlo Methods

Example 2: High-dimensional integrals

Assume we want to compute a d-dimensional integral

$$I = \int_0^1 \dots \int_0^1 f(x_1, \dots, x_d)\, dx_1 \, \dots \, dx_d.$$

Perhaps the most obvious way to approximate this quantity is to apply a one-dimensional quadrature rule $(x_k, w_k)_{k=1}^n$ to each of these d integrals, i.e. to compute

$$I \approx \sum_{k_d=1}^{n} \dots \sum_{k_1=1}^{n} f(x_{k_1}, \dots, x_{k_d})\, w_{k_1} \, \dots \, w_{k_d}.$$

We observe:

- The above approximation requires $N = n^d$ function evaluations.
- If the one-dimensional quadrature rule has an $O(n^{-p})$ error, then so does the high-dimensional quadrature approximation.

  Proof. See next slide.

Monte Carlo Methods

Proof that high-dimensional quadrature inherits the order of convergence of the one-dimensional quadrature rule.

Iteratively replacing integrals with quadrature approximations, we obtain

$$
\begin{aligned}
\int_0^1 \dots \int_0^1 f(x_1, \dots, x_d)\, dx_1 \, \dots \, dx_d &= \dots \\
&= \int_0^1 \dots \int_0^1 \left(\sum_{k_1=1}^{n} f(x_{k_1}, x_2, \dots, x_d)\, w_{k_1} + O\big(n^{-p}\big) \right) dx_2 \, \dots \, dx_d \\
&= \quad \dots \\
&= \sum_{k_d=1}^{n} \dots \sum_{k_1=1}^{n} f(x_{k_1}, \dots, x_{k_d})\, w_{k_1} \, \dots \, w_{k_d} + O\big(n^{-p}\big).
\end{aligned}
$$

Monte Carlo Methods

Example 2: High-dimensional integrals (continued)

Our observations on slide 5 imply that the error and number of function evaluations $N = n^d$ are related through

$$\text{error} \;=\; O\big(N^{-p/d}\big);$$

see `integral_via_quadrature()`.

The number of function evaluations required to meet a certain error tolerance is hence given by

$$N \;=\; O\big(\text{error}^{-d/p}\big),$$

i.e. N scales exponentially in the number of dimensions d and therefore becomes prohibitively large already for moderate values of d.

This phenomenon is known as the "curse of dimensionality".

Monte Carlo Methods

Example 2: High-dimensional integrals (continued)

As in the m, n, k-game example, it turns out that we can avoid excessive runtimes by replacing deterministic computations with random sampling.

In the case of high-dimensional integrals, the key to doing so is to observe that we can interpret such integrals as the expectation

$$\mathbb{E}[f(X)] = \int_0^1 \dots \int_0^1 f(x_1, \dots, x_d)\, dx_1 \, \dots \, dx_d$$

and hence

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{k=1}^{N} f(X_{k,1}, \dots, X_{k,d})$$

where

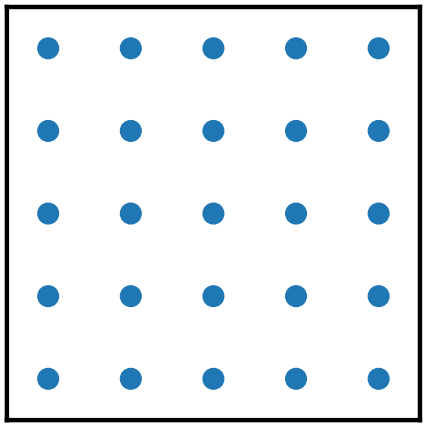
- X is a random variable uniformly distributed over $[0,1]^d$, and
- $X_k \in [0,1]^d$ is a sequence of samples of this random variable.

This trick is again known as *Monte Carlo sampling*.

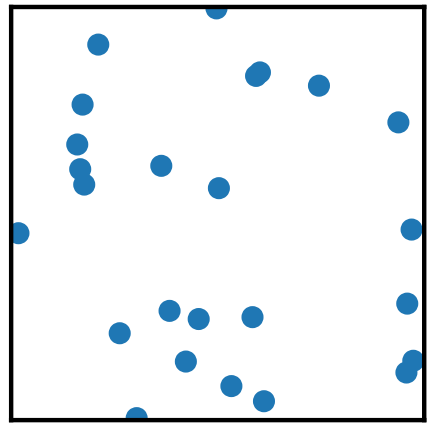
Monte Carlo Methods

Example 2: High-dimensional integrals (continued)

Illustration of quadrature vs. Monte Carlo sampling:



Quadrature



Monte Carlo

Blue dots denote quadrature / sampling points.

Example 2: High-dimensional integrals (continued)

`integral_via_monte_carlo()` shows that Monte Carlo sampling converges with rate $O(N^{-1/2})$ regardless of the dimension.

We therefore conclude:

- Monte Carlo sampling is slower than midpoint quadrature if

$$2/d > 1/2 \quad \Longleftrightarrow \quad d < 4.$$

- Monte Carlo sampling is faster than midpoint quadrature if

$$2/d < 1/2 \quad \Longleftrightarrow \quad d > 4.$$

In particular, Monte Carlo sampling avoids the curse of dimensionality and is therefore a powerful tool to tackle high-dimensional problems.

Monte Carlo Methods

Example 1: Winning probabilities in m, n, k-games (reprise)

It turns out that now that we know what to look out for, we can also observe the $O(N^{-1/2})$ convergence behaviour when applying Monte Carlo sampling to compute the m, n, k-winning probabilities: every time we run `mnk_probabilities()` with a 100x larger number of samples N, we gain roughly one extra digit of accuracy.

Introduction to Monte Carlo error estimation

The above examples indicate that the power of the Monte Carlo approach stems from the fact that

$$\mathbb{E}[X] \;=\; \tfrac{1}{N}\sum_{k=1}^{N} X_k + O(N^{-1/2})$$

for a very wide class of random variables X.

Our goal in the following will be to clarify and prove this claim. Doing so requires a solid background in probability theory; hence let us begin by recapitulating the basics.

Monte Carlo Methods

Def: Measure, probability space, random variable

- A *measure* is a function P which maps subsets of some set Ω to nonnegative real numbers such that

$$A, B \subset \Omega \quad \text{and} \quad A \cap B = \{\} \quad \Longrightarrow \quad P(A \cup B) = P(A) + P(B).$$

- A *probability measure* is a measure P such that $P(\Omega) = 1$. The number $P(A) \in [0, 1]$ is then called the *probability* of the event $A \subset \Omega$.
- A *probability space* is a pair (Ω, P) such that P is a probability measure on the set Ω.
- A *random variable* is a function $X : \Omega \to V$ defined on a probability space (Ω, P).
- The probability measure P_X on V given by

$$P_X(A) \;=\; P(\{\omega \in \Omega \mid X(\omega) \in A\})$$

  is called the *distribution* of the random variable $X : \Omega \to V$.
- Distributions are commonly expressed using the following convenience notation:

$$P(X_1 \in A_1, \ldots, X_n \in A_n) \;=\; P\big(\{\omega \in \Omega \mid X_1(\omega) \in A_1 \wedge \ldots \wedge X_n(\omega) \in A_n\}\big).$$

Example 1: Winning probabilities in m, n, k-games (continued)

The above definitions are fairly abstract, so let me illustrate them by means of the m, n, k-games example from the beginning of this lecture.

This example can be expressed in the above framework as follows.

- The probability space Ω is the set of all sequences of board states

$$\omega = B_0 \to B_1 \to \ldots \to B_g$$

  which represent a rules-conforming and complete game.

- Since this set is discrete, we can specify the probability measure P by specifying the probability $P(\{\omega\})$ of each sequence $\omega \in \Omega$ and then set

$$P(A) = \sum_{\omega \in A} P(\{\omega\}) \qquad \text{for all } A \subset \Omega.$$

Example 1: Winning probabilities in m, n, k-games (continued)

- According to the "random moves" assumption, the probability of a game of length g is given by

$$P\big(\{B_0 \to \ldots \to B_g\}\big) \;=\; \prod_{k=0}^{g} \big(mn-k\big)^{-1}.$$

- The winning probability for player 1 is then given by $P(W_1 = 1)$ where W_1 denotes the random variable $W_1 : \Omega \to \{0, 1\}$ given by

$$W_1\big(B_0 \to \ldots \to B_g\big) \;=\; \begin{cases} 1 & \text{if } B_g \text{ shows a win for player 1, and} \\ 0 & \text{otherwise.} \end{cases}$$

Monte Carlo Methods

Probability spaces vs random variables

An important and occasionally confusing subtlety of probability theory is that probability spaces can always be replaced by another layer of random variables.

To illustrate this point, assume we have

- a probability space (Ω, P), and
- a sequence of random variables $X_k : \Omega \to V_k$ where $k \in \{1, \ldots, n\}$.

We can then do the following:

- Choose another probability space $(\hat{\Omega}, \hat{P})$.
- Construct a random variable $W : \hat{\Omega} \to \Omega$ such that

$$\hat{P}(W \in A) = P(A).$$

- Replace $X_k : \Omega \to V_k$ with $\hat{X}_k : \hat{\Omega} \to V_k$ given by

$$\hat{X}_k(\hat{\omega}) = X_k(W(\hat{\omega})).$$

Monte Carlo Methods

Probability spaces vs random variables (continued)

These new random variables $\hat{X}_k$ are then indistinguishable from the original variables X_k in the sense that they have the same distribution:

$$\begin{aligned}
\hat{P}(\hat{X}_k \in A_k) &= \hat{P}(\{\hat{\omega} \in \hat{\Omega} \mid \hat{X}_k(\hat{\omega}) \in A_k\}) && \text{(def } \hat{P}(\hat{X}_k \in A_k)\text{)} \\
&\text{(def } \hat{X}_k\text{)} \quad = \hat{P}(\{\hat{\omega} \in \hat{\Omega} \mid X_k(W(\hat{\omega})) \in A_k\}) \\
&\text{(rewrite)} \quad = \hat{P}(\{\hat{\omega} \in \hat{\Omega} \mid W(\hat{\omega}) \in \{\omega \in \Omega \mid X(\omega) \in A_k\}\}) \\
&\text{(def } W\text{)} \quad = P(\{\omega \in \Omega \mid X_k(\omega) \in A_k\}) \\
&\text{(def } \hat{P}(X_k \in A_k)\text{)} \quad = P(X_k \in A_k).
\end{aligned}$$

The practical consequence of this is that it is possible and common practice to do probability theory exclusively in terms of random variables and without ever specifying the underlying probability space.

The following notations are relevant in this context:

- $X \in V$ indicates that X is a random variable $X : \Omega \to V$.
- $X \sim D$ indicates that X is a random variable with probability distribution D.

Monte Carlo Methods

Example 1: Winning probabilities in m, n, k-games (continued)

Let me once again illustrate the abstract theory by means of the m, n, k-game example.

On slide 14, I modeled such games by choosing the probability space to be the set of all sequences of board states. You can think of this as assuming that we have some means to directly generate such sequences.

This is of course not true: in our code, we generated the board state sequences using Julia's `rand()` function. One could therefore argue that it would be more accurate if we chose our probability space Ω to be the output of `rand()` and treat the sequence of board states as a function of this output (i.e. a random variable).

This line of reasoning can be continued forever: Julia's `rand()` must source its randomness from somewhere, e.g. thermal noise in your computer, so it would be even more accurate to let the probability space be the microscopic state of your hardware. This state in turn depends on some other variables, so we should update our choice of probability space once again, and then again and again until we conclude that our probability space should be the state of the entire universe.

Monte Carlo Methods

Example 1: Winning probabilities in m, n, k-games (continued)

Which of the above models we choose has of course no impact on our quantity of interest, namely the probability of a win for player 1. All that is needed to compute this quantity are the probabilities of sequences of board states; whether these probabilities come in the form of a probability space (Ω, P) or a random variable S with associated probability distribution $A \mapsto P(S \in A)$ is irrelevant.

Rather than going down the rabbit hole of seeking the most "accurate" model, it is therefore more productive to instead look for the simplest model which provides us with the aforementioned information.

At first sight, it may seem that this simplest model is the one presented on slide 14 where we treat the set of sequences of board states as the probability space and whether player 1 won as a random variable defined on that probability space.

However, on second though it becomes clear that this model has an important drawback, namely it forces us to deal with both probability spaces and random variables, i.e. two types of objects instead of one.

Example 1: Winning probabilities in m, n, k-games (continued)

The best way to do probability theory is hence to treat everything as random variables defined on some unspecified probability space and inject all the relevant bits of information in the form of statements regarding the distributions of these random variables.

This is the approach that I will follow for most of this lecture.

Monte Carlo Methods

Representing distributions

Another frequent source of confusion in probability is the fact that measures / distributions (I consider the two terms to be synonyms) can be represented in several distinct ways.

The three most frequently used representations are:

- Probability mass function (PMF),
- Probability density function (PDF),
- Cumulative distribution function (CDF).

Definitions of these representations will follow on slide 22.

All of these representation of course fulfill the basic purpose of assigning probabilities $P(A) \in [0, \infty)$ to every subset $A \subset \Omega$, but they differ in several other aspects:

- Each of the above representations applies only to some measures.
- Even if a representation applies to a given measure in principle, it may not be possible to explicitly write down this representation.
- Depending on what you want to achieve, working with one representation might be easier than working with another.

Representing distributions (continued)

Now that we understand why we have different measure representations, let me next formally define the three representations mentioned above.

Following the "treat everything as a random variable" approach motivated on slide 17, I will do so using the distribution-of-random-variable notation $P(X \in A)$ where $A \subset V$ rather than the technically simpler measure-on-probability-space notation $P(A)$ where $A \subset \Omega$, since the former notation is how we will describe measures throughout this lecture.

Monte Carlo Methods

Representing distributions (continued)

- The *probability mass function (PMF)* of a discrete random variable $X \in V$ is a function $p : V \to [0, 1]$ such that for all $A \subset V$ we have

$$P(X \in A) = \sum_{x \in A} p(x).$$

- The *probability density function (PDF)* of a continuous random variable $X \in V$ is a function $p : V \to [0, \infty)$ such that for all $A \subset V$ we have

$$P(X \in A) = \int_A p(x)\, dx.$$

- The *cumulative distribution function (CDF)* of a random variable $X \in \mathbb{R}$ is a function $C : \mathbb{R} \to [0, 1]$ such that for all $a \in \mathbb{R}$ we have

$$P(X \leq a) = C(a).$$

Monte Carlo Methods

Some more distribution terminology and notation

- I will write $X \sim \text{PMF}(p)$ in the following to indicate that X is distributed according to the distribution whose PMF is $p(x)$, and likewise for $X \sim \text{PDF}(p)$ and $X \sim \text{CDF}(C)$.
- I will use the term "PDF" to refer to both PDFs and PMFs if the continuous / discrete nature of the distribution is irrelevant in the given context.

Example: Uniform distribution

We say $X \sim \text{Uniform } V$ to indicate that X is "equally likely" to assume any value in V. This distribution is most easily described in terms of its probability mass / density function, which is given by

$$p(x) = \frac{1}{|V|}$$

where

- $|V|$ denotes the number of elements in V if V is discrete, and
- $|V|$ denotes the length / area / volume / etc of V if $V \subset \mathbb{R}^n$.

Monte Carlo Methods

Other important distributions

In addition to the uniform distribution, you should also be familiar with the following distributions.

Discrete distributions:

- Bernoulli(p)
- Binomial(n, p)
- Geometric(p)

Continuous distributions:

- Normal(μ, σ^2)
- Exponential(λ)

You can look up the PMFs / PDFs and some intuition for each of these distributions on their Wikipedia pages. (Click on the above list items; they are links.)

In particular, make sure you are familiar with the normal distribution since this distribution will feature prominently later on.

Now that we understand random variables and their distributions, let us continue our mission of recapitulating basic concepts from probability theory.

Monte Carlo Methods

Def: Expectation of random variables

The *expectation* of a random variable $X \in V$ with PMF / PDF $p(x)$ is given by

$$\mathbb{E}[X] = \sum_{x \in V} x\, p(x) \qquad \text{or} \qquad \mathbb{E}[X] = \int_V x\, p(x)\, dx$$

depending on whether V is continuous or discrete.

Def: Expectation of functions of random variables

The *expectation* of a function $f(X)$ of a random variable $X \in V$ with PMF / PDF $p(x)$ is given by

$$\mathbb{E}[f(X)] = \sum_{x \in V} f(x)\, p(x) \qquad \text{or} \qquad \mathbb{E}[f(X)] = \int_V f(x)\, p(x)\, dx$$

depending on whether V is continuous or discrete.

Monte Carlo Methods

General definition of expectation (not examinable)

The previous slide presented four different formulae for the expectations of four different types of random variables. Rather than definitions, these formulae are technically corollaries of just a single definition which goes as follows:

The *expectation* of a random variable $X : \Omega \to V$ is given by
$$\mathbb{E}[X] = \int_\Omega X(\omega)\, d\omega.$$

Let me point out the following.

- $\int_\Omega d\omega$ denotes integration with respect to the probability measure P of the underlying probability space. This notion of integral may not have been introduced to you yet, which is why I first presented the formulae on the previous slide as definitions and only now point out that all of these formulae actually derive from a single origin.
- The integral formulae on the previous slide derive from the above formula and the substitution $d\omega \to p(x)\, dx$ which may be interpreted as an instance of the integration by substitution formula.

Monte Carlo Methods

Def: Variance

The *variance* of a random variable X is given by

$$\mathrm{Var}[X] = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big].$$

Def: Independence

A collection of random variables $X_k \in V_k$ is called *independent* if for all $A_k \subset V_k$ we have

$$P\big(X_1 \in A_1, \ldots, X_n \in A_n\big) \;=\; \prod_{k=1}^{n} P(X_k \in A_k).$$

Def: Independent and identically distributed (iid)

We write

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} D$$

to indicate that the random variables $X_1, \ldots, X_n$ are independent and identically distributed according to a distribution D.

Lemma: Useful identities

- Let $X, Y : \Omega \to \mathbb{R}^n$ be random variables and $a, b \in \mathbb{R}$. Then,
  1. $\mathbb{E}[aX + bY] = a\,\mathbb{E}[X] + b\,\mathbb{E}[Y]$,
  2. $\mathrm{Var}[aX + b] = a^2\,\mathrm{Var}[X]$,
  3. $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$.
- The random variables $(X_k)_{k=1}^n$ are independent if and only if the PMFs / PDFs of $(X_k)_{k=1}^n$ and each X_k are related through

$$p(x_1, \ldots, x_n) \;=\; p_1(x_1) \ldots p_n(x_n)$$

- If $(X_k)_{k=1}^n$ are independent, then
  - $\mathbb{E}[X_1 \ldots X_n] \;=\; \prod_{i=1}^n \mathbb{E}[X_i]$, and
  - any set of functions $\big(f_k(X_k)\big)_{k=1}^n$ of $(X_k)_{k=1}^n$ are independent.

Proof. Try yourself or see any probability theory textbook.

Monte Carlo Methods

We have now reached the end of our probability theory recap. Let me next point out how the Monte Carlo idea fits into this framework.

Def: Monte Carlo estimate

The *Monte Carlo estimate* $\tilde{\mathbb{E}}_N[X]$ for the expectation of a random variable $X \sim D$ is given by

$$\tilde{\mathbb{E}}_N[X] = \frac{1}{N}\sum_{k=1}^{N} X_k \qquad \text{where} \qquad X_k \overset{\text{iid}}{\sim} D.$$

An important aspect of this definition is that the Monte Carlo estimate $\tilde{\mathbb{E}}_N[X]$ is itself a random variable $\tilde{\mathbb{E}}_N[X] : \Omega \to V$.

Concrete estimates can be derived by evaluating this random variable at a particular point $\omega \in \Omega$, which you can think of as a particular set of outputs of `rand()`; see slide 18.

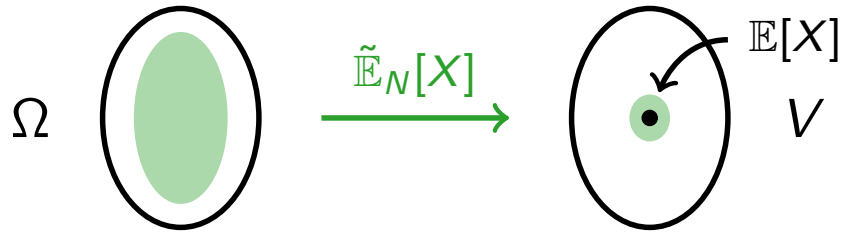
Monte Carlo Methods

Monte Carlo estimate as a random variable

Treating the Monte Carlo estimate as a random variable is on the one hand fairly intuitive since it reflects that this estimate depends on a "source of randomness" $\omega \in \Omega$.

On the other hand, it makes the Monte Carlo method look even more absurd: according to the above definition, the Monte Carlo method consists in assembling a function $\Omega \to \mathbb{V}$ and then evaluating this function at a particular point $\omega \in \Omega$; why on earth would this be a reasonable procedure for estimating $\mathbb{E}[X]$?

An important part of the answer to this question is that $\tilde{\mathbb{E}}_N[X](\omega)$ acts like a funnel in that it maps most of Ω to a region close to $\mathbb{E}[X]$.



This funnel property of Monte Carlo estimates is a consequence of the well-known central limit theorem presented on the next slide.

Monte Carlo Methods

Central limit theorem (CLT, sketch)

$$\tilde{\mathbb{E}}_N[X] \quad \xrightarrow{d} \quad \text{Normal}\Big(\mathbb{E}[X], \tfrac{1}{N}\text{Var}[X]\Big) \qquad \text{for } N \to \infty.$$

This formulation of the central limit theorem involves two components which may be confusing for you:

- $X_k \xrightarrow{d} D$ denotes convergence in distribution. Loosely speaking, this means that the distribution of X_k becomes increasingly indistinguishable from D as $k \to \infty$. A rigorous definition of this type of convergence is beyond the scope of this module.
- The limiting distribution $\text{Normal}\big(\mathbb{E}[X], \frac{1}{N}\,\text{Var}[X]\big)$ in the above formulation of the central limit theorem involves N and is therefore strictly speaking not a valid limit. The technically correct but less suggestive formulation of the central limit theorem is as follows:

$$\sqrt{\tfrac{N}{\text{Var}[X]}}\,\Big(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\Big) \quad \xrightarrow{d} \quad \text{Normal}(0,1).$$

Monte Carlo Methods

The central limit theorem is a rigorous mathematical result with a precise meaning, but many of its details are quite complicated.

For most of this module, we will therefore be working with the following oversimplified of the central limit theorem.

Engineer's version of the central limit theorem

$$\tilde{\mathbb{E}}_N[X] \;\tilde{\sim}\; \text{Normal}\Big(\mathbb{E}[X], \tfrac{1}{N}\text{Var}[X]\Big) \qquad \text{if } N \gtrsim 100.$$

Monte Carlo Methods

Interpretation of the central limit theorem

There are two ways how we can translate the central limit theorem into an $O(N^{-1/2})$ error estimate for Monte Carlo methods.

Option 1. We have

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} &= \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[\tilde{\mathbb{E}}_N[X]]\right)^2\right]} && \text{(CLT, expectation)} \\
&= \sqrt{\operatorname{Var}\left[\tilde{\mathbb{E}}_N[X]\right]} && \text{(def Var[}X\text{])} \\
&= \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}, && \text{(CLT, variance)}
\end{aligned}$$

i.e. the root of the expected squared error is $O(N^{-1/2})$.

This quantity is frequently used as a more easily analysable approximation to the expected error

$$\mathbb{E}\left[\left|\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right|\right] \approx \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} = \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}.$$

Monte Carlo Methods

Interpretation of the central limit theorem

There are two ways how we can translate the central limit theorem into an $O(N^{-1/2})$ error estimate for Monte Carlo methods.

Option 1. We have

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} &= \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[\tilde{\mathbb{E}}_N[X]]\right)^2\right]} && \text{(CLT, expectation)} \\
&= \sqrt{\mathrm{Var}\left[\tilde{\mathbb{E}}_N[X]\right]} && \text{(def Var[}X\text{])} \\
&= \sqrt{\tfrac{1}{N}\,\mathrm{Var}[X]}, && \text{(CLT, variance)}
\end{aligned}$$

i.e. the root of the expected squared error is $O(N^{-1/2})$.

This quantity is frequently used as a more easily analysable approximation to the expected error

$$\mathbb{E}\left[\left|\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right|\right] \approx \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} = \sqrt{\tfrac{1}{N}\,\mathrm{Var}[X]}.$$

Monte Carlo Methods

Interpretation of the central limit theorem

There are two ways how we can translate the central limit theorem into an $O(N^{-1/2})$ error estimate for Monte Carlo methods.

Option 1. We have

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} &= \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[\tilde{\mathbb{E}}_N[X]]\right)^2\right]} && \text{(CLT, expectation)} \\
&= \sqrt{\operatorname{Var}\left[\tilde{\mathbb{E}}_N[X]\right]} && \text{(def Var[}X\text{])} \\
&= \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}, && \text{(CLT, variance)}
\end{aligned}$$

i.e. the root of the expected squared error is $O(N^{-1/2})$.

This quantity is frequently used as a more easily analysable approximation to the expected error

$$\mathbb{E}\left[\left|\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right|\right] \approx \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} = \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}.$$

Monte Carlo Methods

Interpretation of the central limit theorem

There are two ways how we can translate the central limit theorem into an $O(N^{-1/2})$ error estimate for Monte Carlo methods.

Option 1. We have

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} &= \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[\tilde{\mathbb{E}}_N[X]]\right)^2\right]} && \text{(CLT, expectation)} \\
&= \sqrt{\operatorname{Var}\left[\tilde{\mathbb{E}}_N[X]\right]} && \text{(def Var[}X\text{])} \\
&= \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}, && \text{(CLT, variance)}
\end{aligned}$$

i.e. the root of the expected squared error is $O(N^{-1/2})$.

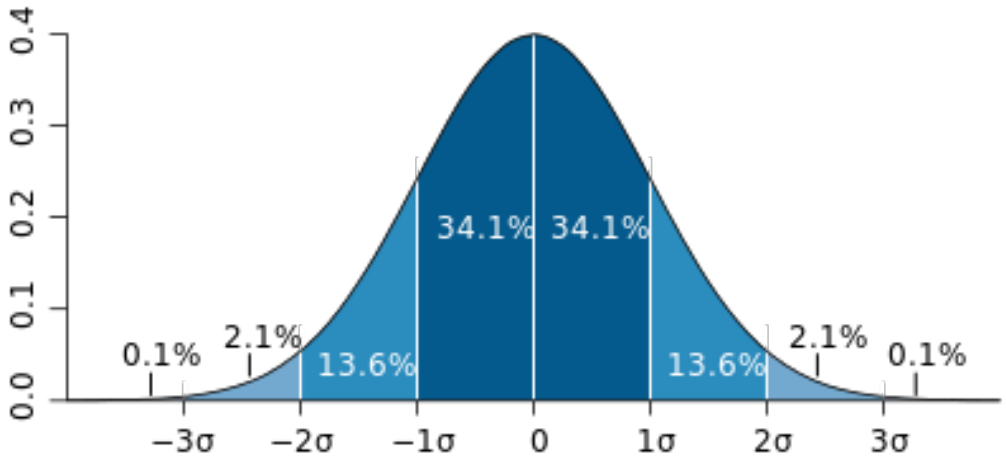
This quantity is frequently used as a more easily analysable approximation to the expected error

$$\mathbb{E}\left[\left|\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right|\right] \approx \sqrt{\mathbb{E}\left[\left(\tilde{\mathbb{E}}_N[X] - \mathbb{E}[X]\right)^2\right]} = \sqrt{\tfrac{1}{N}\operatorname{Var}[X]}.$$

Monte Carlo Methods

Interpretation of the central limit theorem (continued)

Option 2. A normally distributed random variable $X \sim \text{Normal}(\mu, \sigma^2)$ is highly unlikely to assume values more than about three standard deviations σ from its mean μ.



The engineer's version of the central limit theorem,

$$\tilde{\mathbb{E}}_N[X] \;\approx\; \text{Normal}\Big(\mathbb{E}[X], \tfrac{1}{N}\text{Var}[X]\Big),$$

thus implies that it is highly unlikely that the Monte Carlo estimate $\tilde{\mathbb{E}}_N[X]$ deviates from its mean $\mathbb{E}[X]$ by more than about $3\sqrt{\text{Var}[X]/N}$.

Image source: `https://en.wikipedia.org/wiki/File:Standard_deviation_diagram.svg`

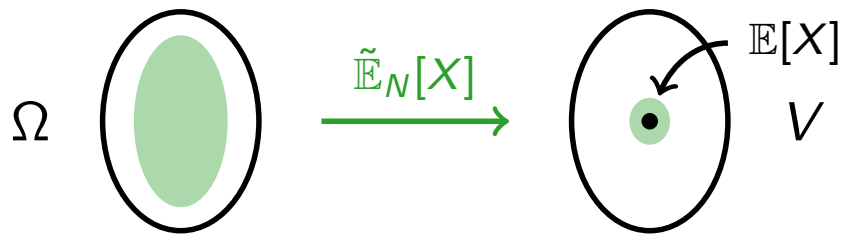
Monte Carlo Methods

Interpretation of the central limit theorem (continued)

We can summarise the above findings as follows.

1. The Monte Carlo estimate $\tilde{\mathbb{E}}_N[X]$ is *expected* to deviate from the true expectation $\mathbb{E}[X]$ by an $O(N^{-1/2})$ error.
2. It is *highly unlikely* that the Monte Carlo estimate $\tilde{\mathbb{E}}_N[X]$ deviates from the true expectation $\mathbb{E}[X]$ by more than an $O(N^{-1/2})$ error.

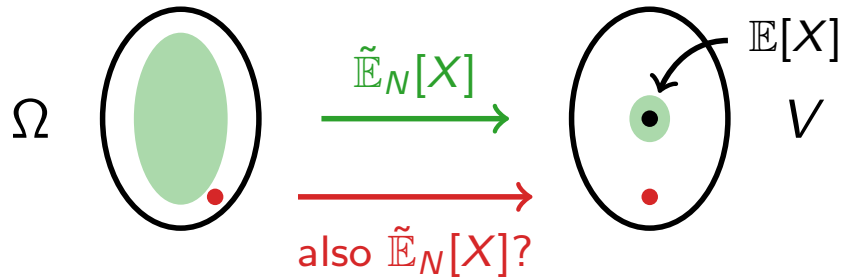
The Monte Carlo estimator $\tilde{\mathbb{E}}_N[X]$ hence indeed has the funnel property of mapping most of Ω to a region close to $\mathbb{E}[X]$.



Monte Carlo Methods

Monte Carlo as gambling with heavily biased odds

$\tilde{\mathbb{E}}_N[X]$ mapping *most* of Ω to a region close to $\mathbb{E}[X]$ of course does rule out that there might be some $\omega \in \Omega$ for which $\tilde{\mathbb{E}}_N[X](\omega)$ is far from $\mathbb{E}[X]$ even for large N.



It is therefore in principle possible that Monte Carlo sampling will give us a completely wrong result.

Monte Carlo Methods

Monte Carlo as gambling with heavily biased odds (continued)

However, the probability of obtaining a bad Monte Carlo estimate can be made vanishingly small simply by choosing N large enough. For example, it follows from the engineer's version of the central limit theorem that

$$P\left(|\tilde{\mathbb{E}}_{10'000}[X] - \mathbb{E}[X]| > 0.1\sqrt{\mathrm{Var}[X]}\right) \approx 10^{-23}.$$

With odds like these, you are more likely to get struck by lightning than to obtain a bad Monte Carlo estimate.

Julia code to determine the above probability:

```
julia> using Distributions
julia> N = 10_000
julia> 2*cdf(Normal(0,1/sqrt(N)), -0.1)
1.5239706048320995e-23
```

Monte Carlo Methods

Introduction to Monte Carlo algorithms

We have now established that Monte Carlo sampling is very likely to produce estimates with $O(N^{-1/2})$ errors.

This indicates that Monte Carlo sampling is a *promising* approach for evaluating complicated sums and integrals, but it is not enough to say that it is a *good* approach; to that end, we need not just error control but also algorithms for evaluating the Monte Carlo estimate $\tilde{\mathbb{E}}_N[X](\omega)$ efficiently.

Such algorithms must necessarily involve the following two steps.

- A way to generate random inputs $\omega \in \Omega$.

  This step is known as *random number generation*.
- A way to map this random input to a random variable $X \sim D$ with the desired distribution D.

  This step is known as *simulation of random variables*.

We will next look at each of these steps in turn.

Monte Carlo Methods

Random number generators (RNGs)

The standard choice for the "base layer" of random variables in Monte Carlo simulations is

$$\omega = (U_k)_{k=1}^{\infty}, \qquad U_k \overset{\text{iid}}{\sim} \text{Uniform}[0,1].$$

This choice is a compromise between between choosing ω which are on the one hand easy to generate and on the other hand easy to work with.

Julia allows us to generate random numbers uniformly distributed in $[0,1]$ using the `rand()` function.

```
julia> rand()
0.13065023345187532
julia> rand()
0.18678073841353582
julia> rand()
0.41031330399815924
```

Such functions are called *random number generators (RNGs)*.

Monte Carlo Methods

Pseudorandom number generators (pRNGs)

An important feature of `rand()` is that its output is not actually random; rather, its output is a deterministic function of some internal state which we can set using `Random.seed!()`.

```
julia> Random.seed!(42);    # Set internal state
julia> rand()
0.5331830160438613    # Random output
julia> rand()
0.4540291355871424    # Random output

julia> Random.seed!(42);    # Reset internal state
julia> rand()
0.5331830160438613    # Same as above!
julia> rand()
0.4540291355871424    # Same as above!
```

Such RNGs are called *pseudorandom number generators (pRNGs)*.

More on pRNGs

The notion of pRNGs raises three questions:

- How can we construct pRNGs?
- How can we test RNGs?
- Why use pRNGs and not "true" RNGs?

Let me address each of these questions in turn.

Monte Carlo Methods

How can we construct pRNGs?

Constructing good and fast pRNGs is a very challenging topic with many subtle pitfalls. It is therefore best left to domain experts who know what they are doing and who have the ability to accumulate the experience of millions of users worldwide.

The current state of the art pRNG is the *Mersenne Twister* algorithm proposed in 1997. This algorithm is the default in Julia, Matlab, Python, R, Microsoft Excel and virtually any other programming environment.

Some codes still use older pRNGs like *linear congruential* and *lagged Fibonacci generators*. These pRNGs are known to introduce subtle biases in your Monte Carlo simulation and should therefore be replaced with Mersenne Twister whenever possible.

Monte Carlo Methods

How can we test RNGs?

The purpose of an RNG is to produce a sequence of numbers $u_k \in [0, 1]$ which "look as if" they are samples of Uniform$[0, 1]$. It is clear that this characterisation leaves ample room for interpretation, but devising a more accurate characterisation turns out to be surprisingly difficult.

The current state of the art is to assess the quality of an RNG by subjecting its output to a large number of so-called *randomness tests*.

Well-known such test sets include:

- Diehard tests (1995)
- TestU01 (2007)

Mersenne Twister passes all of the Diehard tests but fails some of the TestU01 tests.

Monte Carlo Methods

Why use pRNGs and not "true" RNGs?

pRNGs offer two important advantages over "true" RNGs:

- They allow us to exactly reproduce the output of our code. This is useful for debugging and "reproducible science".
- pRNGs tend to be significantly faster than true RNGs; see `rng_benchmark()`.

Monte Carlo Methods

Simulating random variables

Now that we know how to sample $U \sim \text{Uniform}[0, 1]$, the next question is how we can construct a function $U \mapsto X$ which maps one or more such random variables U to a random variable X with a desired target distribution D.

There are three standard techniques for doing so, and the following slides will go through each one of them in turn.

Monte Carlo Methods

Thm: Inverse transform sampling

Let $C(x)$ be a cumulative distribution function, i.e. a monotonically increasing function $C : \mathbb{R} \to [0, 1]$ such that $C(-\infty) = 0$ and $C(\infty) = 1$. Then,

$$U \sim \text{Uniform}[0, 1] \qquad \Longrightarrow \qquad X = C^{-1}(U) \sim \text{CDF}(C).$$

Proof. We have

$$\begin{aligned} P\big(X \leq x\big) &= P\big(C^{-1}(U) \leq x\big) && \text{(def } X\text{)} \\ \text{(rewrite)} &= P\big(U \leq C(x)\big) \\ (U \sim \text{Uniform}[0, 1]) &= C(x). \end{aligned}$$

The CDF of $X = C^{-1}(U)$ is hence indeed equal to $C(x)$.

Monte Carlo Methods

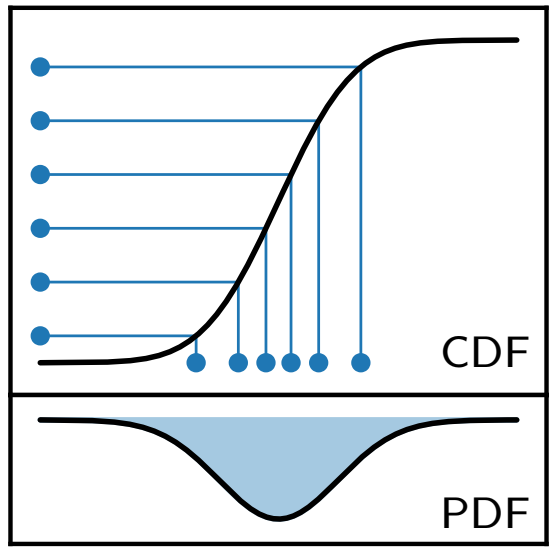
Intuition for inverse transform sampling

The intuition behind inverse transform sampling is as follows:

- If the PDF $p(x)$ is large on a certain interval $[a, b]$, then the CDF

$$C(x) = \int_{-\infty}^{x} p(x)\, dx$$

  is steep on this interval.
- A steep $C(x)$ exposes a larger cross section to the y-axis; hence $C([a, b])$ is more likely to be hit by $U \sim \text{Uniform}[0, 1]$, and hence we will obtain more samples in this interval $[a, b]$.



Monte Carlo Methods

Example 1

Consider the PDF $p(x) = 2x$ on $[0, 1]$. Its associated CDF is

$$C(x) = \int_0^x 2y \, dy = x^2 \qquad \text{for } x \in [0, 1].$$

Given samples of $U \sim \text{Uniform}[0, 1]$, we can hence generate samples of $X \sim \text{PDF}(p)$ by setting

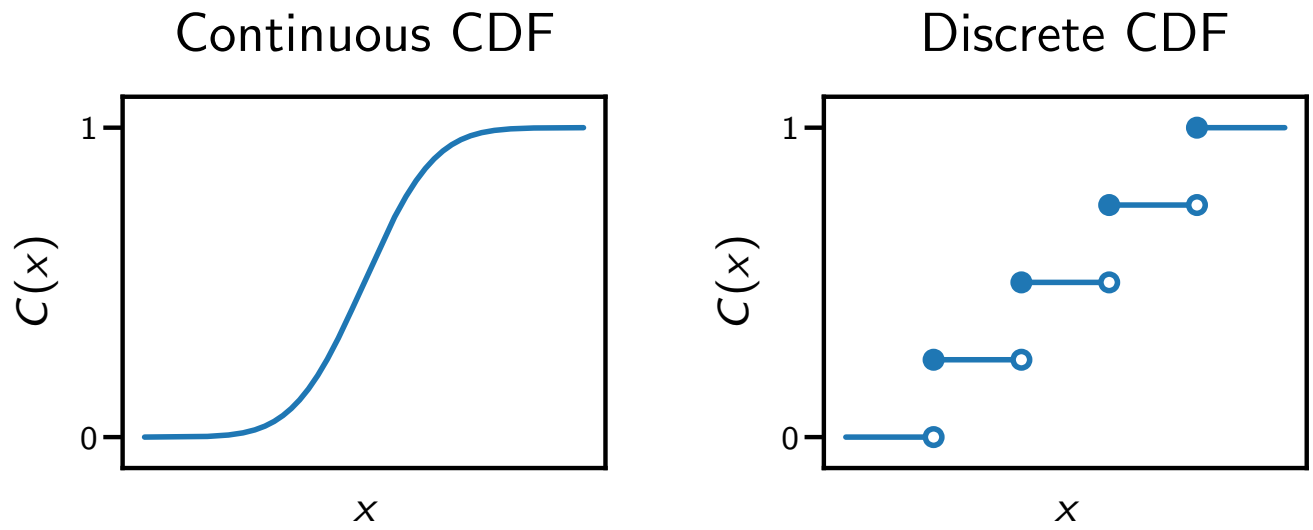
$$X = C^{-1}(U) = \sqrt{U};$$

see `inverse_transform()`.

Inverse cumulative distribution function (iCDF)

The above formulation of the inverse transform sampling theorem assumes that the CDF $C(x)$ has an inverse.

This is true if the target distribution is continuous since then $C(x)$ is a continuous and monotonically increasing function, but it is not true for discrete target distributions.



Inverse cumulative distribution function (iCDF, continued)

Fortunately, it turns out that we can use the inverse transform sampling theorem even for discrete distributions if we generalise the notion of inverse CDF to

$$C^{-1}(u) \;=\; \inf\{\, x \mid u \leq C(x) \,\}.$$

This fact can be shown by repeating the proof on slide 48 but using Property 5 of the following lemma in the (rewrite) step.

Lemma: Properties of iCDF

1. $C^{-1}(u)$ is the usual inverse if $C(x)$ is invertible.
2. $C^{-1}(u)$ is increasing.
3. $C^{-1}(C(x)) \;\leq\; x$
4. $C(C^{-1}(u)) \;\geq\; u$
5. $C^{-1}(u) \leq x \iff u \leq C(x)$

Monte Carlo Methods

Proof (not examinable).

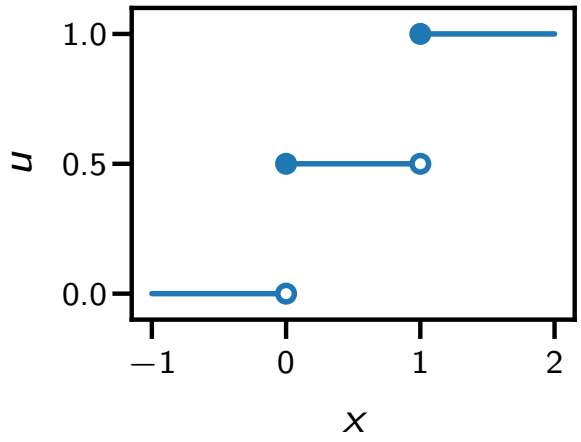
1. If $C(x)$ is invertible, then we can apply $C^{-1}(u)$ to the set condition to obtain
$$\inf\{\, x \mid u \leq C(x) \,\} \;=\; \inf\{\, x \mid C^{-1}(u) \leq x \,\} \;=\; C^{-1}(u).$$
2. $C^{-1}(u) \;=\; \inf\{\, x \mid u \leq C(x) \,\}$ is the infimum over a decreasing set; hence it is increasing.
3. We have
$$C^{-1}(C(x)) \;=\; \inf\{\hat{x} \mid C(x) \leq C(\hat{x})\} \;\leq\; x,$$
where in the second step I used that $x \in \{\hat{x} \mid C(x) \leq C(\hat{x})\}$.
4. Since $C(x)$ is increasing and right-continuous, we have
$$C(C^{-1}(u)) \;=\; \inf\{C(x) \mid u \leq C(x)\} \;\geq\; u.$$
5. Since $C(x)$ is increasing, we have according to Property 3 that
$$C^{-1}(u) \;\leq\; x \quad\Longrightarrow\quad u \;\leq\; C(C^{-1}(u)) \;\leq\; C(x).$$
Similarly, since $C^{-1}(x)$ is increasing, we have according to Property 2 that
$$u \;\leq\; C(x) \quad\Longrightarrow\quad C^{-1}(u) \;\leq\; C^{-1}(C(x)) \;\leq\; x.$$

Monte Carlo Methods

Example 2

The CDF of the Bernoulli(p) distribution is given by

$$C(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1-p & \text{if } 0 \leq x < 1, \\ 1 & \text{if } 1 < x. \end{cases}$$



Given samples of $U \sim \text{Uniform}[0, 1]$, we can hence generate samples of $X \sim \text{Bernoulli}(p)$ by setting

$$X = C^{-1}(U) = \begin{cases} 0 & \text{if } U < 1-p, \\ 1 & \text{if } U \geq 1-p. \end{cases}$$

See `bernoulli()`.

Monte Carlo Methods

Thm: Rejection sampling

Let $p(x)$, $q(x)$ be two PDFs such that

$$p(x) \;\leq\; M\, q(x) \qquad \text{for all } x \text{ and some } M > 0.$$

The random variable X generated by the below algorithm then satisfies

$$X \;\sim\; \text{PDF}(p).$$

Algorithm Rejection sampling

---

for $k = 1, 2, 3, \ldots$ **do**
  Sample $Q_k \sim \text{PDF}(q)$ (independently of all other Q_k and U_k)
  Sample $U_k \sim \text{Uniform}[0, 1]$ (independently of all other Q_k and U_k)
  if $U_k \leq \frac{p(Q_k)}{M\, q(Q_k)}$ **then**
    Return $X = Q_k$
  end if
end for

Monte Carlo Methods

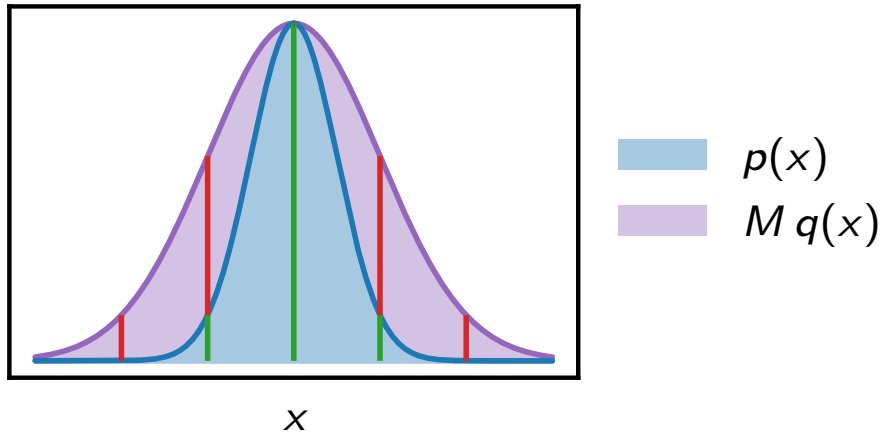
Rejection sampling terminology

- $p(x)$ is called the *target distribution* (as before).
- $q(x)$ is called the *proposal distribution*.
- Setting $X = Q_k$ (line 5) is called *accepting* the proposal Q_k.

Intuition for rejection sampling

The above algorithm can be put into words as follows.

Generate proposals according to PDF(q),
but accept them only with probability $\frac{p(Q_k)}{M\, q(Q_k)}$.



Monte Carlo Methods

The following result is useful both for understanding the correctness and performance of rejection sampling.

Thm: Acceptance probability

The probability that the rejection sampling algorithm accepts the next proposal is

$$P([\text{accept } Q_k]) = \tfrac{1}{M}.$$

Proof (not examinable). According to the law of total probability, we have

$$\begin{aligned} P([\text{accept } Q_k]) &= \int P\big([\text{accept } Q_k] \mid Q_k = x\big)\, q(x)\, dx \\ &= \int \tfrac{p(x)}{M\, q(x)}\, q(x)\, dx \\ &= \tfrac{1}{M} \int p(x)\, dx \\ &= \tfrac{1}{M}. \end{aligned}$$

Monte Carlo Methods

Proof of rejection sampling theorem from slide 55 (not examinable).

$$
\begin{aligned}
P(X \in A) &= P(Q_1 \in A, [\text{accept } Q_1]) + \dots && \text{(disjoint events)} \\
&\qquad P(Q_2 \in A, [\text{reject } Q_1, \text{ accept } Q_2]) + \dots \\
\text{(independence)} \quad &= \int_A q(x)\, \tfrac{p(x)}{M\, q(x)}\, dx + (1 - \tfrac{1}{M}) \int_A q(x)\, \tfrac{p(x)}{M\, q(x)}\, dx + \dots \\
&= \tfrac{1}{M} \int_A p(x)\, dx \left(\sum_{k=0}^{\infty} (1 - \tfrac{1}{M})^k \right) \\
&= \tfrac{1}{M} \int_A p(x)\, dx\, \frac{1}{1 - (1 - \tfrac{1}{M})} \\
&= \int_A p(x)\, dx.
\end{aligned}
$$

Monte Carlo Methods

Example

See `rejection_sampling()`.

Note that since

$$[\text{accept } Q_k] \sim \text{Bernoulli}(\tfrac{1}{M}),$$

we have

$$[\#\text{ proposals until acceptance}] \sim \text{Geometric}(\tfrac{1}{M})$$

and in particular

$$\mathbb{E}[\#\text{ proposals until acceptance}] = M.$$

Rejection sampling can hence become quite inefficient if $p(x)$ is large in regions where $q(x)$ is small (i.e. if M is large).

Inverse transform vs rejection sampling

Now that we have two algorithms for the same purpose, we should compare their relative strengths and weaknesses.

This comparison goes as follows.

	Inverse transform	**Rejection**
Pros	Fast if $C^{-1}(u)$ is fast	Fairly flexible
Cons	$C^{-1}(u)$ may be complicated. Only works in 1d.	Inefficient if M is large (i.e. requires good proposals)

Monte Carlo Methods

Intro to importance sampling

The purpose of inverse transform and rejection sampling is to generate samples $X_k \overset{\text{iid}}{\sim} D$ which allow us to evaluate the Monte Carlo estimate

$$\tilde{\mathbb{E}}_N[X] = \frac{1}{N}\sum_{k=1}^{N} X_k.$$

This is of course one way to approximate the expectation $\mathbb{E}[X]$, but it turns out that it is not the only one.

Specifically, we will see on slide 63 that given another random variable Y, we can construct a function $f(Y)$ such that

$$\mathbb{E}[X] = \mathbb{E}[f(Y)].$$

Instead of applying Monte Carlo sampling to $\mathbb{E}[X]$, we can hence use Monte Carlo sampling to estimate $\mathbb{E}[f(Y)]$.

Intro to importance sampling (continued)

Applying Monte Carlo sampling to $\mathbb{E}[f(Y)]$ instead of $\mathbb{E}[X]$ may result in either or both of the following advantages:

- Y may be easier to sample than X.
- $f(Y)$ may have lower variance than X and hence sampling $f(Y)$ may lead to a smaller Monte Carlo error.
  (Recall from the CLT that $\left|\mathbb{E}[X] - \tilde{\mathbb{E}}_N[X]\right| \lesssim \sqrt{\mathrm{Var}[X]/N}$.)

The first advantage requires no further comments. I will further elaborate on the second advantage after explaining how to construct $f(Y)$.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x\, \tfrac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x \, \frac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x\, \tfrac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x\, \frac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x\, \tfrac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x \, \frac{p(x)}{q(x)} \, q(x)\, dx = \mathbb{E}\Big[\, Y \, \frac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Thm: Importance sampling

Assume $p(x)$, $q(x)$ are two PDFs such that

$$x\, p(x) \neq 0 \quad \Longrightarrow \quad q(x) \neq 0.$$

Then,

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Proof.

$$\mathbb{E}[X] = \int x\, p(x)\, dx = \int x\, \tfrac{p(x)}{q(x)}\, q(x)\, dx = \mathbb{E}\Big[\, Y \, \tfrac{p(Y)}{q(Y)} \,\Big].$$

Importance sampling terminology

PDF(q) is called the *importance*, *reference* or *sampling* distribution.

Monte Carlo Methods

Example 1

Consider the PDFs $p(x) = 2x$ and $q(x) = 1$, both supported on $[0, 1]$. Assuming $X \sim \text{PDF}(p)$, $Y \sim \text{PDF}(q) = \text{Uniform}[0, 1]$, we then have

$$\mathbb{E}[X] \;=\; \mathbb{E}[\, Y \, \tfrac{p(Y)}{q(Y)} \,] \;=\; \mathbb{E}[\, Y \, \tfrac{2Y}{1} \,] \;=\; \mathbb{E}[\, 2Y^2 \,].$$

We can hence compute the expectation of $X \sim \text{PDF}(p)$ by sampling only $Y \sim \text{Uniform}[0, 1]$ and doing some trivial arithmetic on these samples; see `importance_sampling()`.

Monte Carlo Methods

Example 2

Example 1 above demonstrated how we can use importance sampling to estimate the expectation of a "complicated" random variable by taking a weighted mean of samples of a 'simple" random variable.

This is one of the two applications of importance sampling mentioned on slide 62, the other being that $Y \frac{p(Y)}{q(Y)}$ may have lower variance than X.

We can illustrate this second advantage of importance sampling simply by switching the roles of (X, p) and (Y, q): if we use samples of X to estimate the expectation of Y, then we obtain

$$\mathbb{E}[Y] \;=\; \mathbb{E}\left[X \, \frac{q(X)}{p(X)}\right] \;=\; \mathbb{E}\left[X \, \frac{1}{2X}\right] \;=\; \frac{1}{2},$$

i.e. the random variable $X \frac{q(X)}{p(X)}$ is simply a constant and hence Monte Carlo sampling produces the correct result for any number of samples N,

$$\tilde{\mathbb{E}}_N\left[X \, \frac{q(X)}{p(X)}\right] \;=\; \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} \;=\; \frac{1}{2} \;=\; \mathbb{E}[Y].$$

Example 2 (continued)

A formal way to see that Monte Carlo sampling applied to $X \frac{q(X)}{p(X)}$ must be exact is to note that since $X \frac{q(X)}{p(X)} = \frac{1}{2}$ is a constant, its variance is 0 and hence

$$\begin{aligned} \tilde{\mathbb{E}}_N\left[X \frac{q(X)}{p(X)}\right] &\sim \text{Normal}\left(\mathbb{E}\left[X \frac{q(X)}{p(X)}\right], \text{Var}\left[X \frac{q(X)}{p(X)}\right]\right)) \\ &= \text{Normal}\big(\mathbb{E}[Y], 0\big). \end{aligned}$$

Monte Carlo Methods

Summary

- Monte Carlo idea: estimate expectations by taking the mean of a large number of samples.
- Basic probability theory: probability space, random variables, distributions, expectation, variance, independence.
- Central limit theorem: if $X, X_1, \ldots, X_N \overset{\text{iid}}{\sim} D$, then

$$\tilde{\mathbb{E}}_N[X] = \frac{1}{N}\sum_{k=1}^{N} X_k \quad \xrightarrow[N\to\infty]{d} \quad \text{Normal}\Big(\mathbb{E}[X], \tfrac{1}{N}\text{Var}[X]\Big).$$

Monte Carlo Methods

Summary (continued)

- Pseudorandom number generators (pRNGs): deterministic sequence $u_k \in [0, 1]$ which "looks like" samples of $U \sim \text{Uniform}[0, 1]$.
- Inverse transform sampling:

$$U \sim \text{Uniform}[0, 1] \quad \Longrightarrow \quad C^{-1}(U) \sim \text{CDF}(C).$$

- Rejection sampling: generate proposals according to PDF(q) but accept them only with probability $\frac{p(Q_k)}{M\, q(Q_k)}$.
- Importance sampling:

$$X \sim \text{PDF}(p), \quad Y \sim \text{PDF}(q) \qquad \Longrightarrow \qquad \mathbb{E}[X] = \mathbb{E}\left[Y \, \frac{p(Y)}{q(Y)}\right].$$