Exercise Walkthrough: Monte Carlo Integration

Justin Lanfermann

25. June 2025

Abstract

This document provides a detailed walkthrough for an exercise on Monte Carlo (MC) integration. We will explore how to approximate a definite integral over a two-dimensional domain—the unit disk—by leveraging the Law of Large Numbers. We will break down the problem into three main parts: sampling points from the domain, using these points to estimate the area of the domain, and finally, approximating the integral itself. Throughout this guide, we will connect the practical steps back to the theoretical foundations presented in the "Discrete Probability Theory" script.

1 Introduction: The Core Idea

Sometimes we encounter integrals that are difficult or even impossible to solve analytically. This is especially true for integrals over high-dimensional or complex-shaped domains. Monte Carlo integration offers a powerful, computer-friendly method to find numerical approximations for such integrals.

The central principle behind this method is the **Law of Large Numbers (LLN)** [1]. The LLN, in essence, states that the average of a large number of independent and identically distributed (iid) random samples will be very close to the expected value of the distribution they are drawn from.

How does this relate to integration? Let's say we want to compute the integral $I = \int_U g(x)\,\mathrm{d}x$ for some function g over a domain $U \subset \mathbb{R}^d$. If we consider a random variable X that is uniformly distributed on U (i.e., $X \sim \mathrm{Unif}(U)$ [2]), its probability density function (pdf) is $p(x) = \frac{1}{\lambda^d(U)}$ for $x \in U$, where $\lambda^d(U)$ is the volume (or area, in 2D) of U.

The expected value of the function g applied to our random variable X is then:

$$\mathbb{E}[g(X)] = \int_U g(x)p(x)\,\mathrm{d}x = \int_U g(x)\frac{1}{\lambda^d(U)}\,\mathrm{d}x = \frac{1}{\lambda^d(U)}\int_U g(x)\,\mathrm{d}x$$

Rearranging this gives us a direct link between the integral we want and an expectation we can estimate:

$$I = \int_U g(x)\,\mathrm{d}x = \lambda^d(U) \cdot \mathbb{E}[g(X)]$$

Now, the LLN (specifically, Theorem 2.61/2.62 in the script) tells us we can approximate the expectation $\mathbb{E}[g(X)]$ by drawing n iid samples $x_1, \dots, x_n$ from Unif(U) and computing their average:

$$\mathbb{E}[g(X)] \approx \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

Putting it all together, we get the master formula for Monte Carlo integration:

$$I = \int_U g(x)\,\mathrm{d}x \approx \lambda^d(U) \cdot \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

In this exercise, we will apply this formula to the function $f(x) = 10 - x_1^2 - x_2^2$ over the unit disk D^2.

2 Step-by-Step Solution

We follow the three steps outlined in the exercise statement.

2.1 Step (i): Sampling Uniformly from the Unit Disk

The Task: Draw n samples uniformly and independently from inside the unit disk, $D^2 = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\}$.

Reasoning and Method: A common mistake is to try to sample in polar coordinates by picking a radius $r \sim$ Unif$(0, 1)$ and an angle $\theta \sim$ Unif$(0, 2\pi)$. This will incorrectly cluster points near the origin. Why? Because the area of an annulus from radius r to $r + dr$ is $2\pi r dr$, which grows with r. A uniform sampling in r doesn't account for this.

A simple and correct method is **Rejection Sampling** [3]. The idea is to sample from a larger, simpler shape that contains our target domain and then "reject" the samples that fall outside.

1. **Define a bounding box:** The unit disk D^2 is fully contained within the square $S = [-1, 1] \times [-1, 1]$. The area of this square is $\lambda^2(S) = 2 \times 2 = 4$.
2. **Sample from the box:** Generate a candidate point (x_1, x_2) by drawing two independent samples from a uniform distribution: $x_1 \sim$ Unif$(-1, 1)$ and $x_2 \sim$ Unif$(-1, 1)$.
3. **Accept or Reject:** Check if the point lies inside the unit disk. A point is inside if its distance from the origin is at most 1, which is equivalent to checking if $x_1^2 + x_2^2 \leq 1$.
   - If $x_1^2 + x_2^2 \leq 1$, we **accept** the point as a valid sample from the disk.
   - If $x_1^2 + x_2^2 > 1$, we **reject** the point and go back to step 2 to generate a new one.
4. **Repeat:** We continue this process until we have collected n accepted points.

This procedure guarantees that the collected points are uniformly distributed within the disk because every point in the bounding square has an equal chance of being generated, and we are simply filtering out those not in our target domain.

2.2 Step (ii): Estimating the Area of the Disk (π)

The Task: Use the samples to estimate the area of the unit disk, $\lambda^2(D^2) = \pi$.

Reasoning and Method: The rejection sampling method from the previous step gives us a natural way to estimate π. The probability that a random point drawn uniformly from the square S also lies inside the disk D^2 is the ratio of their areas:

$$p = P(\text{point is in } D^2) = \frac{\lambda^2(D^2)}{\lambda^2(S)} = \frac{\pi}{4}$$

We can estimate this probability p by generating a large number of points, let's say N_{total}, inside the square and counting how many of them, N_{inside}, fall within the disk. The ratio $\frac{N_{inside}}{N_{total}}$ is our Monte Carlo estimate for p.

$$\frac{N_{inside}}{N_{total}} \approx p = \frac{\pi}{4}$$

By rearranging, we get an estimate for π:

$$\pi \approx 4 \cdot \frac{N_{inside}}{N_{total}}$$

For example, if we generate $N_{total} = 1,000,000$ points in the square $[-1,1]^2$ and find that $N_{inside} = 785,398$ of them are in the disk, our estimate would be:

$$\pi \approx 4 \cdot \frac{785,398}{1,000,000} \approx 3.141592$$

This is a very accurate approximation! The quality of the estimate improves as N_{total} increases, as predicted by the LLN.

2.3 Step (iii): Approximating the Integral

The Task: Use the n samples from Step (i) to approximate the integral $I = \int_{D^2}(10 - x_1^2 - x_2^2)\,\mathrm{d}x$ and compare it to the analytical value.

Reasoning and Method: We now use the main Monte Carlo integration formula derived in the introduction:

$$I \approx \lambda^2(D^2) \cdot \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

Here, $f(x_i) = 10 - x_{i1}^2 - x_{i2}^2$, the domain is D^2, and its area is $\lambda^2(D^2) = \pi$. The samples $x_i = (x_{i1}, x_{i2})$ are the n points we generated and accepted in Step (i).

1. **Generate Samples:** Using the rejection method from Section 2.1, we generate n points, say $n = 10,000$, that lie inside the unit disk.

2. **Evaluate the Function:** For each accepted point x_i, we calculate the value of the function $f(x_i) = 10 - (x_{i1}^2 + x_{i2}^2)$.

3. **Compute the Average:** We calculate the sample mean of these function values:

$$\overline{f_n} = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

4. **Estimate the Integral:** We multiply the average by the area of the domain:

$$I_n = \pi \cdot \overline{f_n} = \pi \cdot \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

(Note: For a "pure" Monte Carlo approach, you could use the estimate for π from Section 2.2, but using the true value is standard practice when it's known.)

Comparison with the Analytical Value: To verify our result, we can solve the integral analytically using a change of variables to polar coordinates [4]. In polar coordinates, $x_1 = r\cos\phi$, $x_2 = r\sin\phi$, so $x_1^2 + x_2^2 = r^2$. The differential area element becomes $\mathrm{d}x_1\,\mathrm{d}x_2 = r\,\mathrm{d}r\,\mathrm{d}\phi$. The domain D^2 corresponds to $r \in [0,1]$ and $\phi \in [0, 2\pi]$.

$$\begin{aligned}
I &= \int_{D^2} (10 - x_1^2 - x_2^2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \\
&= \int_0^{2\pi}\int_0^1 (10 - r^2)\cdot r\,\mathrm{d}r\,\mathrm{d}\phi \\
&= \int_0^{2\pi}\left[\int_0^1 (10r - r^3)\,\mathrm{d}r\right]\mathrm{d}\phi \\
&= \int_0^{2\pi}\left[5r^2 - \frac{r^4}{4}\right]_0^1 \mathrm{d}\phi \\
&= \int_0^{2\pi}\left(5 - \frac{1}{4}\right)\mathrm{d}\phi = \int_0^{2\pi}\frac{19}{4}\,\mathrm{d}\phi \\
&= \frac{19}{4}[\phi]_0^{2\pi} = \frac{19}{4}\cdot 2\pi = \frac{19\pi}{2}
\end{aligned}$$

The true value of the integral is $\frac{19\pi}{2} \approx 29.845$.[1]

When we implement the Monte Carlo estimation, we expect our result I_n to be close to this value. For $n = 10,000$, a typical result might be around 29.85, demonstrating the effectiveness of the method.

3 Summary and Takeaways

- **Monte Carlo integration** is a versatile numerical technique for approximating complex integrals, grounded in the **Law of Large Numbers**.

[1] The analytical solution provided in the exercise prompt seems to contain a calculation error. The steps shown here lead to the correct analytical value for the given integrand and domain.

- The core process involves **uniform sampling** from the integration domain, **averaging** the function values at these sample points, and scaling by the **domain's volume/area**.

- For non-standard domains like the unit disk, techniques like **rejection sampling** provide a simple and effective way to generate the required uniform samples.

- The accuracy of the approximation generally improves as the number of samples (n) increases, with the error typically decreasing proportionally to $1/\sqrt{n}$.

This exercise beautifully illustrates the bridge between abstract probability theory and practical computational science, showing how sampling from distributions can solve problems in calculus.

4 Deeper Explanations

[1] **The Law of Large Numbers (LLN):** As presented in the script (Theorems 2.61 and 2.62), the LLN comes in two main forms. The **Weak LLN** states that for a large number of iid random variables, the sample mean will be close to the true mean with high probability. The **Strong LLN** makes a more powerful statement: the sample mean will almost surely converge to the true mean as the number of samples approaches infinity. For Monte Carlo methods, either law provides the theoretical justification that our sample average $\frac{1}{n}\sum g(x_i)$ is a valid estimator for the true expectation $\mathbb{E}[g(X)]$.

[2] **Uniform Distribution:** The uniform distribution, $\mathrm{Unif}(a, b)$ (Definition 1.56.i), describes a scenario where every outcome in a continuous interval $[a, b]$ is equally likely. Its pdf is constant, $p(x) = \frac{1}{b-a}$ for $x \in [a, b]$ and 0 otherwise. For a 2D domain like the square $S = [-1, 1]^2$, the uniform distribution has a constant pdf $p(x_1, x_2) = \frac{1}{\lambda^2(S)} = \frac{1}{4}$ for points inside the square.

[3] **Rejection Sampling:** This is a general Monte Carlo method used to generate observations from a distribution. It is especially useful when the target distribution has a complex shape but is contained within a simpler shape from which we can easily sample. By sampling from the simple shape and rejecting points outside the target domain, we effectively sample from the target distribution. The efficiency of this method depends on the "rejection rate," i.e., the ratio of the volumes of the target and sampling domains. In our case, this is $\pi/4 \approx 78.5\%$, meaning we accept about 4 out of every 5 points on average.

[4] **Change of Variables to Polar Coordinates:** This is a standard technique in multivariable calculus for simplifying integrals over circular domains. A point (x_1, x_2) is represented by its distance from the origin, $r = \sqrt{x_1^2 + x_2^2}$, and its angle, ϕ. When transforming the integral, the area element $\mathrm{d}x_1\,\mathrm{d}x_2$ is replaced by $r\,\mathrm{d}r\,\mathrm{d}\phi$. The factor of r is the determinant of the Jacobian matrix of the transformation and accounts for the fact that a small patch of area gets larger as its distance from the origin (r) increases.