Introduction to Monte Carlo Simulation

Bruna G. Palm

Fall, 2024

Outline

What is Monte Carlo Simulation?

- A computational technique that uses random sampling to estimate complex mathematical problems.
- Named after the Monte Carlo Casino in Monaco, reflecting the element of chance.
- Widely used in finance, physics, engineering, and more.

Key Concepts

- Random Variables
- Probability Distributions
- Expectation and Variance

Law of Large Numbers

- The average of the results obtained from a large number of trials should be close to the expected value.
- Formal Definition: $\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{P} \mu$

Basic Monte Carlo Algorithm

- Define the domain of possible inputs.
- Generate inputs randomly from the domain.
- Perform a deterministic computation on the inputs.
- Aggregate the results to estimate the desired quantity.

Importance Sampling – Bootstrap

- Technique to reduce variance by sampling from a distribution similar to the target distribution.
- Modify the basic Monte Carlo method to use this new distribution.

Mean and Bias

- **Mean:** The average of the simulation results.
- **Bias:** Difference between the expected value of an estimator and the true value of the parameter being estimated:

$$\text{Bias} = E[\hat{\theta}] - \theta$$

Relative Bias and Standard Error

- **Relative Bias:** Bias normalized by the true value:

$$\text{Relative Bias} = \frac{E[\hat{\theta}] - \theta}{\theta}$$

- **Standard Error (SE):** Measures the variability of the estimator:

$$\text{SE} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\hat{\theta}_i - \bar{\theta})^2}$$

Mean Square Error (MSE)

- **MSE:** Measures the average of the squares of the errors:

$$\text{MSE} = E[(\hat{\theta} - \theta)^2] = \text{Variance} + \text{Bias}^2$$

Skewness and Kurtosis

- **Skewness:** Measures the asymmetry of the probability distribution:

$$\text{Skewness} = \frac{E[(X-\mu)^3]}{\sigma^3}$$

- **Kurtosis:** Measures the "tailedness" of the probability distribution:

$$\text{Kurtosis} = \frac{E[(X-\mu)^4]}{\sigma^4} - 3$$

- Normal distribution skewness and kurtosis values are around 0 and 3, respectively.

Estimating Pi using Monte Carlo Simulation in R

- Generate random points within a square.
- Count the number of points inside the quarter circle.
- Estimate Pi using the ratio of points inside the circle to the total number of points.

R Code Example

```
# Monte Carlo Simulation to Estimate Pi
set.seed(100)
n <- 100000
# Number of points in each simulation
R <- 1000
 # Number of Monte Carlo simulations

# Vector to store pi estimates
pi_estimates <- numeric(R)
# Vector to store pi estimates
```

R Code Example

```
for (i in 1:R) {
points <- matrix(runif(2 * n), ncol = 2)
# Generate random (x, y) points
inside_circle <- sum(rowSums(points^2) <= 1)
# Count points inside the unit circle
#checking if the point lies within the unit circle
#by calculating the sum of the squares
#of the x and y coordinates and checking
#if it is less than or equal to 1
pi_estimates[i] <- (inside_circle / n) * 4
#Formula to estimate pi
}
```

Model Evaluation in R

```
# Model Evaluation
bias <- mean(pi_estimates) - pi
mse <- mean((pi_estimates - pi)^2)
se <- sd(pi_estimates)

# Compute skewness and kurtosis
library(moments)
skewness_value <- skewness(pi_estimates)
kurtosis_value <- kurtosis(pi_estimates)

# Output the results
results <- list(Bias = bias, MSE = mse, SE = se,
Skewness = skewness_value,
Kurtosis = kurtosis_value)
print(results)
```

Model Evaluation in R

```r
library(ggplot2)
# Create a data frame for the plot
pi_data <- data.frame(pi_estimates)

# Plot the density of estimated pi values
ggplot(pi_data, aes(x = pi_estimates)) +
geom_density(fill = "lightblue", alpha = 0.7) +
geom_vline(xintercept = pi,
color = "red", linetype = "dashed", size = 1) +
labs(title = "Density Plot of Estimated Pi Values",
x = "Estimated Pi",
y = "Density") +
theme_minimal()
```