Lab Assignment 2
Sampling Methods and the Central Limit Theorem

ACS II

Spring 2019

Assigned: January 17
Due: January 24

Introduction

In statistics and probability it is important to be able to sample a random number from a given distribution. In this lab, the method from Lab 1 known as inverse transform sampling will be will be revisited in more depth. It will also investigate another method known as the acceptance-rejection method. The lab concludes by looking at one of the key theorems from probability, the central limit theorem.

Inverse Transform Sampling

Suppose we are given a distribution $\rho(x)$, where $\rho(x) \geq 0$ for all x. If in addition $\int_{-\infty}^{\infty} \rho(x)\ \mathrm{d}x = 1$ then ρ is a valid probability density function (PDF). Inverse transform sampling requires knowledge of the cumulative distribution function (CDF). The CDF corresponds to a particular PDF, and is defined as

$$F(x) = \int_{-\infty}^{x} \rho(t)\ \mathrm{d}t.$$

If ρ is a valid PDF then $F(a)$ can be interpreted as $P(x \leq a)$, i.e. the probability that a number chosen randomly from ρ is less than or equal to a. A CDF must satisfy certain requirements. For instance it must always be positive, always less than or equal to 1 and it can never be decreasing. Given the CDF, the generation of a single sample y from the corresponding PDF is done as follows:

1. pick a uniform random number u between 0 and 1

2. find y such that $F(y) = u$

In general $F(y) = u$ may be nonlinear in y, so root finding methods may have to be used. Since by construction $F(x)$ cannot decrease, the equation $F(y) = u$ has only a single solution so Newton's method will work well. Create some objective function $H(y) = F(y) - u = 0$ and find the root. Note that to use Newton's method you will need the derivative of $H(x)$.

Deliverable

Write a `C++` or `Fortran` program to test this method. This main program should call a function `inverse_transform_sample` that takes as an input the CDF $F(x)$ and its derivative $F'(x)$, and returns a sample from the corresponding PDF. Test your code on the exponential distribution, which has a PDF $\rho(x) = \lambda e^{-\lambda x}$ and a CDF $F(x) = 1 - e^{-\lambda x}$. Take $\lambda = 1$ and make a histogram of 1000 samples (you may use any scripting language to make the histogram from your output data). Making sure the histogram is properly normalized, plot $\rho(x)$ over this histogram. Describe your findings, and briefly state what test(s) could be used to assess the quality of your sample.

Acceptance-Rejection Method

Suppose we are given a probability distribution $\rho(x)$, defined over some interval $c \le x \le d$, where $\int_c^d \rho(x)\,\mathrm{d}x = 1$, with $\rho(x) \ge 0$. We will assume that ρ has compact support (i.e. c and d are finite), or that ρ can be appropriately truncated without losing a significant amount of information. In general, drawing a sample from such a distribution cannot be done unless we know the corresponding CDF, however the acceptance-rejection method allows us to sample from any $\rho(x)$ without knowing its CDF by repeatedly sampling from a different distribution $g(x)$ that has a known CDF. The acceptance-rejection method works as follows:

1. choose a distribution $g(x)$ and determine a number M such that $\rho(x) \le Mg(x)$ for $c \le x \le d$
2. choose a random number y from the distribution $g(x)$ between c and d using inverse sampling
3. choose another uniform random number u between 0 and 1
4. if $u \le \rho(y)/(Mg(y))$ accept y; otherwise pick another y and u

There is a relation between the number of samples required from $g(x)$ to obtain an accepted value, and the parameter M.

Deliverable

Write a `C++` or `Fortran` program implementing the acceptance-rejection method. Test your implementation with the Rayleigh distribution,

$$\rho(x) = \frac{xe^{-x^2/(2a^2)}}{a^2}.$$

This distribution has mean $a\sqrt{\pi/2}$ and variance $a^2(4-\pi)/2$. Let $a = 1$ and truncate x to be between 0 and 10. With $g(x)$ as the uniform distribution between 0 and 10 and $M = 6.5$, take 10000 samples and compare the sample mean and variance to the theoretical mean and variance. What is the probability of a sample being accepted using a uniform $g(x)$ and $M = 6.5$? Repeat this experiment with

$$g(x) = \begin{cases} 0.19 & 0 \leq x \leq 5 \\ 0.01 & 5 < x \leq 10 \end{cases}$$

Construct the CDF $G(x)$ and use $M = 3.2$. What is the probability of a sample being accepted using this $g(x)$ and M?

Central Limit Theorem

The central limit theorem is a key theorem in probability. Imagine drawing n random samples from a given probability distribution, computing the mean of these samples, and doing this multiple times. The central limit theorem says that as $n \to \infty$ the mean of these samples will be distributed according to a normal distribution.

Specifically, if the probability distribution has mean μ and variance σ^2, and the sample mean $\hat{u}$, then the random variables $S_n = \sqrt{n}\,(\mu - \hat{u}) \sim N(0, \sigma^2)$, i.e. a normal distribution with mean 0 and variance σ^2.

Deliverable

Verify the central limit theorem. Using your rejection sampling code, draw $n = 100$ random samples from the Rayleigh distribution with $a = 1$. Compute the mean $\hat{u}$ of the distribution and the statistic S_n. Repeat this 1000 times and do a histogram of the computed S_n (you may use any scripting language to make the histogram). Making sure the histogram is properly normalized, add a plot of a normal distribution with mean 0 and variance σ^2. Repeat this experiment with $n = 1000$. What do you notice?

Submission and Grading

Please submit a folder containing the following in Canvas:

- your source code files
- Your report as a single file in '.pdf' format, including results from your work and relevant discussion of your observations, results, and conclusions.

This information must be received by 11:59pm, January 24 through Canvas. As stated in the course syllabus, late assignment submissions will be subject to a 10% point penalty per 24 hours past the due date at time of submission, to a maximum reduction of 50%, according to the formula:

*[final score] = [raw score] - min(0.5, 0.1 * [# of days past due]) * [maximum score]*