EE 24: Sampling from a distribution, Limit theorems

Lecturer: Shuchin Aeron, shuchin@ece.tufts.edu
TA: Changgyu Lee, changgyu.lee@tufts.edu Course Project

The aim of this project is to use concepts of probability for generating samples from a specified probability distribution and verify the Central Limit Theorem (CLT) via simulations.

Since we will be using histograms to visualize the distribution of the samples, please study what a histogram is from `https://en.wikipedia.org/wiki/Histogram`. Most numerical packages have this function in-built, for e.g. for python see
`https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.hist.html`
or
`https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html`

1 Sampling from a discrete distribution

In order sample from a distribution, one needs access to a *source of randomness.* We will assume that we have access to an *oracle* that can generate samples from a uniform distribution, uniform over $[0,1]$. In other words we have a machine that can, for any $n \in \mathbb{N}$, generate a set of independent random variables $U_1, \cdots, U_n$, each uniformly distributed between $[0,1]$. In python, e.g. see `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.uniform.html#scipy.stats.uniform`.

1. Suppose you would like to generate a random binary sequence, whose outcomes at each time are independent Bernoulli(p) random variables. Given independent $U_i \sim \text{Unif}([0,1])$ Let $X_i = 1$ if $U_i \leq p$ and $X_i = 0$ if $U_i > p$. Show that X_i are independent have Bernoulli p distribution.

2. Let X_i be defined as in the previous item. Let $X = \sum_{i=1}^{n}$. Show that X has binomial distribution.

3. Let P_X be a pmf of a random variable X taking values in $\{x_1, x_2, \cdots, x_n\}$. Without loss of generality $x_1 < x_2 < \cdots < x_n$ and $P_X(x_j) > 0$ for all $j = 1, \cdots, n$. Let $U \sim \text{Unif}([0,1])$. Define a random variable Y via:

$$Y = \min\{x_j : \sum_{k=1}^{j} P_X(x_k) \geq U\}. \tag{1}$$

   We will now show that Y is a random variable with pmf P_X as prescribed. Your task is to fully justify each step below.

   Proof. Clearly $P_Y(y) = 0$ if $y \notin \{x_1, x_2, \cdots, x_n\}$.

$$P(Y = x_j) = P(\{\sum_{k=1}^{j-1} P_X(x_k) < U\} \cap \{\sum_{k=1}^{j} P_X(x_k) \geq U\}) \tag{2}$$

$$= P(\{\sum_{k=1}^{j-1} P_X(x_k) < U \leq \sum_{k=1}^{j} P_X(x_k)\}) \tag{3}$$

$$= \sum_{k=1}^{j} P_X(x_k) - \sum_{k=1}^{j-1} P_X(x_k) \tag{4}$$

$$= P_X(x_j). \tag{5}$$

$\square$

Therefore, independent samples $u_i, i = 1, 2, ..$ from the Unif$([0,1])$ distribution, one can generate samples that are independent and identically distributed $\sim P_X$ $i = 1, 2, ..,$ via:

$$y_i = \min\{x_j : \sum_{k=1}^{j} P_X(x_k) \geq u_i\}. \tag{6}$$

4. Write a code (in your favorite programming language) to generate 100 independent rolls of a biased 4-faced dice with pmf $P_X(1) = 1/8, P_X(2) = 1/8, P_X(3) = 3/8, P_X(4) = 3/8$. Plot a histogram of the rolls, i.e. plot the observed frequencies for each of the 4 outcomes. Does it look close to the true pmf?

2 Sampling from a continuous distribution

1. Read Problem 10, Chapter 3, from the course textbook. You have already studied this problem in a previous HW. Using the method analyzed in that problem, write a code (in your favorite programming language) to generate samples from an exponential distribution with parameter $\lambda = 1$. Verify via plotting a histogram of 100 samples generated according to the problem.

2. Let $U_1 \sim$ Unif$([0,1])$ and $U_2 \sim$ Unif$([0,1])$ and let $U^{(1)}$ and $U^{(2)}$ be independent. Define $X = \sqrt{2\log(\frac{1}{U^{(1)}})}\cos(2\pi U^{(2)})$. Provide a *guess* so as to what the distribution of X can be, via plotting a histogram of independent samples generated via $x_i = \sqrt{2\log(\frac{1}{u_i^{(1)}})}\cos(2\pi u_i^{(2)})$, where $i = 1, 2, \cdots, 100$ and $u_i^{(1)}, u_i^{(2)}$ are all independent samples from the Unif$([0,1])$ distribution.

For more complicated and multivariate distributions that are often stated in an implicit form, there are other methods such as Markov Chain Monte Carlo (MCMC) methods. If you are interested you may find this link useful: `https://people.duke.edu/~ccc14/sta-663/MCMC.html`

3 Central Limit Theorem (CLT)

From the previous exercises, we can generate independent and identically distributed random samples for a given distribution. We will now use that to numerically observe the Central Limit Theorem (CLT). For this problem you may directly use in-built functions to generate samples for a specified distribution - e.g. `https://docs.scipy.org/doc/scipy/reference/stats.html`

1. Let us start with $X_1, \cdots, X_n, \cdots$ i.i.d. $\sim$ Unif$([0,1])$. Let $S_n = \sum_{i=1}^{n} X_i$ and let $Z_n = \frac{S_n - n\mu}{\sqrt{n}\sigma}$ where μ, σ are the mean and variance for the Unif$([0,1]$ distribution.

   For each $n = 2, 3, 5, 10, 20, 50, 100$, Generate $m = 100$ samples that are distributed according to Z_n and plot the histograms. At what value of n do you start to see CLT approximation to Z_n starting to be a good approximation?

   Now, to see the difference between CLT and Weak Law of Large Numbers (WLLN), also plot the corresponding histograms for $\tilde{Z}_n = \frac{S_n - n\mu}{n\sigma}$.

2. Repeat part 1. with $X_1, \cdots, X_n, \cdots$ i.i.d. $\sim$ Exp(2). You may use built-in functions for generating samples from Exp(2).

3. Repeat part 1. with $X_1, \cdots, X_n, \cdots$ i.i.d. $\sim$ Bernoulli(p) for $p = 1/2$ and for $p = 0.2$. You may use built-in functions for generating samples from the Bernoulli distribution.