EE 131A | Class Project
Probability and Statistics | Tuesday, February 19, 2019
Instructor: Lara Dolecek | Due: Monday, March 18, 2019 by 11:59 pm PDT via CCLE
TA: Ruiyi Wu | ruiyiwu@g.ucla.edu

Reading: Chapters 2 through 8 of *Probability, Statistics, and Random Processes* by A. Leon-Garcia
100 points total

In this project we will further analyze random variables and their various properties. Each part will have a combination of MATLAB programming, mathematical analysis and technical writing. You will be graded on all three components.

When producing your plots **clearly indicate** the x-axis, the y-axis and what is being plotted (using legends, title etc.). You may need to rescale x-axis to ensure that your plot is showing the right quantity.

Make sure to attach in the appendix of your project report **all MATLAB programs** that you used to generate the data.

1. (25 pts) *Probability estimator.* Consider the binary sequence of length 50000 in the file '`data.txt`', available on the course website. In the sequence, **1** denotes an error event and **0** denotes an error-free event. Suppose that these are generated i.i.d., with probability of an error event denoted by p. Based on this data, estimate the probability of an event being in error using the following method.
   A natural estimator is an emperical estimate of p from these samples. That is, find the fraction of error events in these N random samples and use that for estimating p. We will refer to this estimate as $\hat{p}_N$ since it is based on N random samples.

   (a) Take a single pass over the entire sequence and find the fraction of events in error. This is our estimate $\hat{p}_N$ for $N = 50000$, which is based on the entire data.

   (b) Repeat the probability estimation for $N = 10, 100, 200, 300, 500, 1000, 2000, 10000,$ 20000 and plot the corresponding $\hat{p}_N$ versus N. What can you say about this empirical estimator from this plot?

   The data file can be read using MATLAB's `dlmread` (i.e., use `A = dlmread('data.txt')` to read the entire sequence into an array `A`). You can also use any other program to do this problem.

2. (30 pts) *Tossing a fair and unfair die.* Suppose you have a 4-sided die.

   (a) Write a MATLAB program to simulate the tossing of a 4-sided fair die, with sides numbered 1, 2, 3, and 4, for t =10, 50, 100, 500 and 1000 tosses. Based on the simulation, what is the probability of obtaining an odd number?

(b) Suppose X is a random variable denoting the outcome of a die toss. Based on the analysis, what is the probability that X has odd value?

(c) Refer back to part (a). Does it agree with the theoretical result in (b)?

(d) Repeat parts (a), (b), and (c) if even sides are twice as likely as odd sides.

You may find useful the MATLAB function `rand` that generates a uniform random value in the $(0, 1)$ interval.

3. (45 pts) *Central Limit Theorem* Let X_1,X_2,... be a sequence of iid random variable with finite mean μ and finite variance σ^2, and let S_n be the sum of the first n random variables in the sequence:

$$S_n = X_1 + X_2 + ... + X_n.$$

(a) Let X_i be a uniform continuous random variable taking values in the interval $(1, 4)$. Write a MATLAB program to plot the pdf and cdf of S_n. Consider $n = 1, 3, 5, 10, 20, 30$ and compare your results.

(b) Calculate analytically the mean and the variance of X_i and of S_n in part (a).

(c) Write a MATLAB program to generate a Gaussian random variable with the same mean and variance as S_n. Superimpose this plot on the plots from part (a).

(d) Repeat parts (a), (b), and (c) with X_i representing a toss of a fair 4-sided die (see Problem 2(a)).

(e) Repeat parts (a), (b), and (c) with X_i representing a toss of an unfair 4-sided die with even sides twice as likely as odd sides (see Problem 2(d)).

Use $t = 10^4$ samples in the above.