Miles A. Vendetti-Houser

ECE 3522: Stochastics

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122

# Problem Statement

This computer assignment is designed to reinforce concepts discussed in last problem on the first exam. This is accomplished by utilizing MATLAB’s random number generator to generate uniformly distributed random numbers between 0 and 1. We need to then estimate the mean, variance, and compute the error of the estimates. Next, we use a plot with a log scaled x to view what the data does over large quantities.

# Approach and Results

The first part of the computer assignment included the usage of N random numbers all within the range of [0, 1]. Using the following two equations defining the mean, , and the variance, , I iterated through range [1, 106] and stored them in their own respective arrays. After this was completed, the error of these estimates were computed using the following two equations;



Using the *semilogx()* function in MATLAB, I plotted both sets of data and show them below.



So, let’s talk a little about what we see here. When the number of random data points is low, the result of the computed mean does not seem to converge to any sort of value. It jumps around from 0.7 to 0.2, etc. However, as we take the expected value as N grows larger and larger towards infinity (stopped at 105 for computational time reasons…), the estimated mean approaches 0.5. This is amazing! Being able to visualize this phenomena as a simple plot as above. The same goes for the rolling variance of the N amount of random values. At small N, the value also jumps around without any direction in mind until the value of N grows larger and larger. The converging value for variance is 0.8 or 1/12.

We were also required, as stated before, to compute the error of the mean and variance and plot the results. The figure below contains these results…

The shape of both plots for the error could have been predicted by knowing the fact that if as N grows towards infinity, the computed mean and variance both get smaller and smaller differences between the estimated and true convergent values. Therefore, as the graphs show, the error for both the mean and variance tend towards 0 as the mean and variance tend towards the expected values of 0.5 and 0.8 respectively.

The second part of the assignment has us estimate the pdf of x[n] using a fixed bin size of 100. We were then required to plot the mean-squared error between the measured distribution and the actual distribution using the following equation: . Where B = bin size of 100, and p(x) is the true distribution of 0.01 or 1/100. Using the first of the three ranges of N, [101, 103, 106], I generated N(1) random values, found the pdf, and computed the mean-squared error. This process was repeated for N = 103 and 106. I have included below, the three pdfs for each value of N. Also, in order to view what the curve of mean-square error does as N becomes larger, I computed a value for MSE[N] from 1 to 10^6 random variables with the same parameters for the number of bins and plotted them on a log axis.

Figure 3: 106 Random Values

Figure 2: 103 Random Values

Figure 1: 101 Random Values

As you can see from the three graphs shown above, as we increase N, we see that the probability of each bin approaches 0.01. Which makes total sense due to the restraints formed at the beginning of the computer assignment where we were selecting the random values as uniformly distributed.

As for the mean-square error plot below, it shows that as we calculate with higher amounts of N, we get closer to the uniformly distributed value of probability, which seems to be the maximum value the random values can be selected from divided by number of bins .



# MATLAB Code

% Computer Assignment 06

% Stochastics [ECE 3522] Dr. Picone

% Mean and Variance Revisited

clear;clc;clf;close all;

N = [1,10^5];

% Part one

% Generate random numbers

x = rand(N);

% Estimate the mean and variance

est\_mu = 1/N(2)\*sum(x);

est\_sigma = 1/N(2) \* sum(( x - est\_mu).^2);

FASTmu = zeros(N);

mu = zeros(N);

tic

for n=1:N(2)

 if (n-1 <= 0)

 FASTmu(n) = (1/n) \* sum( x(:,1:n) );

 else

 FASTmu(n) = (( FASTmu(n-1)\*(n-1) ) + x(:,n) ) \* (1/n);

 end

end

toc

tic

for n=1:N(2)

 mu(n)= 1/n \* sum( x(:,1:n) );

end

toc

figure(1)

subplot(2,1,1)

semilogx(FASTmu)

subplot(2,1,2)

semilogx(mu)

% Compute the variance

FAST\_sigma = zeros(N);

sigma = zeros(N);

% WORK IN PROGRESS

% mean of square minus square of mean

% for n = N(1):N(2)

% sum( x(:,n)^2 ) - ( FAST\_mu() )^2

% FAST\_sigma(n) =

% end

for n = N(1):N(2)

 sigma(n)= 1/n \* sum(( x(:,1:n) - est\_mu).^2);

end

figure(2)

subplot(2,1,1)

semilogx(mu)

subplot(2,1,2)

semilogx(sigma)

E\_mu = zeros(N);

E\_sigma = zeros(N);

% Compute the error

for n = N(1):N(2)

 E\_mu(n) = mu(n) - est\_mu;

end

for n = N(1):N(2)

 E\_sigma(n) = sigma(n) - est\_sigma;

end

% Plot the error value using logscale plot (bode plot?)

figure(3)

subplot(2,1,1)

semilogx(E\_mu)

subplot(2,1,2)

semilogx(E\_sigma)

%% Part two

%

clear;clc;clf;close all;

bin\_size = 100;

for N = 1:10^4

 summation = 0;

 M = [1, N];

 x = rand(M);

 y = histcounts(x, bin\_size, 'Normalization', 'probability');

 for n = 1:bin\_size

 summation = summation + ( (y(n) - 0.01).^2 );

 end

 mse(N) = (1/bin\_size) \* (summation);

end

semilogx(mse)

xlabel('Number of random variables (N)'); ylabel('MSE of N');

title('Mean-squared Error');

# Conclusions

To conclude this assignment, I quickly talk about the pdf of the uniform distribution. The pdf is characterized as; , where a and b are its minimum and maximum values. If we were to integrate this function for pdf over time multiplied by or in other words…

We find that this is the Moment Generating Function (MGF) where we could have found that evaluating t = 1, we would arrive at or the mean, verifying even more so that the mean will converge to 0.5. Also, taking the MGF and evaluating at t = 2, we can find the variance by subtracting M(2) – M(1)2 as follows;

This tells us that using the MGF of uniform distributed variables, we can still derive at the fact that as N grows infinitely large, the variance becomes closer and closer to 1/12 or 0.8.