Emilie Doyle

ECE 3522: Stochastic Process in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

The objective of this computer assignment is to gain a better understanding of the mean and the variance in a set of data. There is no real given data, as there has been in the past. The data instead is generated by MATLAB through the use of the random number generator function to create uniformly distributed random numbers. It is expected that the mean of these random numbers that range from 0 to 1 should be 0.5. For the first task, the first item to do was to create an array of random numbers from 0 to 1 that will be our “signal” for this analysis. The mean and variance had to be estimated through the employment of the given equations for the mean and the variance. Both had to be found continuously and then plotted against a semi-log scale. The next task focused on the idea of a probability density function (PDF). The PDF of the “signal” had to be found with 100 bins, each of size 0.01. The mean-square error then had to be calculated and plotted against a semi-log for comparison and analysis. Lastly, the process also had to be completed for other values of N.

# Approach and Results.

I started out by creating a lot of variables to allocate memory space in my program. In order to make this code reusable for the different values of N I made an array that contained the used values for N and put my main code in a loop that could easily be changed for each value of K (or the given N, really). For simplicity’s sake, it would be best to only do one value of K at a time. I could loop it through the different values and show all the plots, but that would involve making all of the titles into strings so that you could substitute in the value of K, and some other stuff, and I would like to hand this in on Sunday night so I didn’t add that stuff. But anyway, I used rand to generate an array of random numbers to be the signal x. I calculated the overall mean and the overall variance, as well as used the MATLAB functions to calculate the whole mean and variance, just as a point of comparison. Also I thought it was part of the assignment, which it may or may not be. To find the continuously changing variance and mean (by which I mean, the mean and the variance calculated at each point in x, using all the previous points) I used a loop to loop through the different values of x and calculate them at each step. Next I plotted the continuous variances and means against a semi-log.

Next, I used the histogram function to create a histogram of the data. I used 100 bins and normalized it to be a PDF. I then used the histc function to determine how many values were in each bin of the histogram. In order to find the mean squared error it was important to determine what the PDF a true uniform distribution would look like. The area of a PDF has to be equal to 1, so if there are 10 points in the data set then the height of it will be 1/10. I used this logic to determine the value of the true p(x). I then used a loop to loop through the values and create an array that matched the summation part of the equation for the mean squared error. I then used that to continually calculate the mean square error. After the loop, I plotted the total mean square error array versus a semi-log to see the overall reaction. The index of K could then be changed to do any of these for the other values given.



Figure : Continuous Mean of Original Signal



Figure : Continuous Variance of Original Signal



Figure : Normalized PDF of Original Signal



Figure : Continuous Mean Square Error of Original Signal



Figure : PDF When N=10



Figure : Mean Square Error when N=10



Figure : PDF when N=10e3



Figure : Mean Square Error when N=10e3

# MATLAB Code

%StochCA6

K=[10, 10e3, 10e6];

for q=3:3

 %Declare Variables

 N=[1,K(q)];

 v=[1,K(q)];

 MSE=[100,1];

 contVar=[1,K(q)];

 contMean=[1,K(q)];

 G=[100,1];

 countMean=[1,K(q)];

 countVar=[1,K(q)];

 x= rand(K(q),1);

 %Find Statistics

 overallMean= ((1/K(q))\*sum(x));

 for i=1:length(N)

 v=(x(i)- overallMean)^2;

 contVar(i)=(1/K(q))\*sum(v);

 contMean(i)= ((1/K(q))\*sum(x));

 end

 overallVariance= (1/K(q))\*sum(v);

 MTLBMean= mean(x);%to compare values

 MTLBVar= var(x);%to compare values

 %Plot

 figure(1)

 semilogx(N,contVar);

 title('Continuous Variance of N plotted against N');

 figure(2)

 semilogx(N, contMean);

 title('Continuous Mean of N plotted against N');

 %2

 figure(3)

 histogram(x, 100, 'Normalization', 'pdf');

 title('PDF of x[N]');

 g= histc(x, [0:0.01:0.99]);

 %a true uniform distribution of an array of 10e6 would have height of

 %1/10e6

 pTrue =1/K(q);

 for i=1:length(g)

 G(i)=(g(i)- pTrue)^2;

 MSE(i)= (1/100)\*sum(G);

 end

 p=linspace(1,100,100);

 figure(4)

 semilogx(p, MSE);

 title('Continuous Mean Square Error');

end

# Conclusions

The mean for the uniformly distributed random variables that range from 0 to 1 makes sense that it is 0.5. That does not need to be contested. The variance is the second movement of the data, which is interesting because in these graphs it looks like the variance data resembles a second order of the mean data. What I mean by that is that the mean is a 0th degree equation. It would just be y=0.5. However, the variance is a1st degree equation, in that I could be written in the form y=mx+b. This relationship is almost like a integral/differential relationship. It is almost like the variance is the integral of the mean and/or the mean is the derivative of the variance. This is not exactly true, it just resembles that relationship.

Additionally, it makes sense that the PDF that had 10e6 data points would nearly resemble a perfect uniform distribution. A perfect uniform distribution would have equal values across all of the bins, and the y-value would be 1/10e6, that way the total area under the PDF would be equal to 1. The plot created with 10e6 data points almost matches that. Of course, it is not perfect as the line is not perfectly straight at 1, instead there are small variances amongst the data, but that is to be expected because there are not infinite data points. Since the number of data points is larger than the number of bins, it is highly likely that every bin will have some contents. As the number of data points becomes smaller than the number of bins, it becomes clear that the points are randomly assigned, because the PDF becomes less uniform at the top. This is greatly evident in the PDF for when N=10. There are some bins that have absolutely no values in them, and then there are others that are extremely filled. The difference between the contents in bins is also very evident in the PDF for when N= 10e3. All of the bins have contents, but each bin does not have equal, or near equal, contents. The data is approaching perfect uniform, but is not quite there yet. All of these differences are then mirrored in the mean squared error (MSE) plots. When N=10e6 the MSE plot is very smooth, as a result of consistency amongst the contents of the bins. As N decreases, and the uniformity also decreases, the MSE plots become more erratic and choppy. The MSE graph when N=10e3 is still fairly smooth, with a few minor bumps or wavers, but the MSE graph for when N=10 is so choppy and dramatic.