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 assignment is to further our MATLAB skill through the learning of modeling systems and fitting models to given data. The data given to us in this assignment are the Google stock prices since 2004 in the form of a table, and the audio signal given to us as a \*.raw file. The first task of the assignment is to create a histogram of the amplitude of the data and then to normalize it in order to determine the probability density function of the data. The next task was to estimate the mean and the variance of the data sets. Next a Gaussian model was to be fitted to the data and analyzed for the exactness of the model fit. The last task required the Google stock data to be fit to a different type of model, because a Gaussian model is not a particularly good fit for the data. The mean squared error then had to be calculated for all sets of data and all models used to describe them.

# Approach and Results.

For the first task, I read in the signals into arrays. I used the same process that I have used for the past couple of assignments. The Google data got read in from the Excel table and then converted to an array (‘Close’). I used the file commands to open and read the audio file. It got read right into an array. Both histograms were created just by using the ‘histogram’ function. I chose to normalize them right in the command, and even chose to make the data into a probability density function right in the histogram command. I specified the bin size for the Google histogram because the original histogram was not descriptive enough..



Figure : Google Stock Histogram PDF



Figure : Audio Signal Histogram PDF

For the second task, you had to find the mean and the variance. In order to find the mean, I used the sum command to add up all of the components in the arrays and then divided that value by the number of values in the array. In order to find the variance, or the second movement, I recreated the variance equation. The variance is equal to the mean of the squared array minus the square of the mean of the array. I accomplished this by squaring each element in the array and the summing them all up and dividing by the length, and then subtracting the square of the original mean from that value. In order to fit a Gaussian model to them I used different processes. For the Google data I used the ‘histfit’ function to fit a normal Gaussian curve to the data. I originally used this process for both, and it worked, but then a later task required me to change it for the Audio signal. For the audio signal I used the ‘fitdist’ command and set it to a normal, or Gaussian, distribution. This required a few other steps to set up and plot but was just as effective as ‘histfit’. Figure 4 was obtained through the use of ‘histfit’ instead of my given code because it is easier to see the model in relation to the points.



Figure : Google Stock Histogram PDF with Gaussian



Figure : Audio Signal Histogram with Gaussian Model

For the last task, I used the ‘fitdist’ to fir a Burr Model curve to the Google stock data. The Gaussian model is shown in red, while the Burr model is shown in green on the graph below. The graph below also is a zoomed in version of the overall produced graph. It shows a range from -100 to 700 points.



Figure : Google Stock Histogram PDF with Gaussian (Red) and Burr Model (Green)

In order to find the mean-squared error, I took the array that contained the amplitude points from the models and transpose it so that instead of N x M it because M x N. This was crucial in order for the matrix calculations to work, because the matrices needed to be of the same order. In order to find the mean-squared error, I just used the mean-square error command- ‘rse’. This command takes in the difference between the two arrays (which is why they needed to be oriented the same way) and calculates the mean-square error.



Figure : Mean-Square Error Calculated by MATLAB

# MATLAB Code

%Stoch CA 4

%1 Histogram- Normalized PDF

% Google

t= readtable('excelData.xlsx'); %read in data from excel

DateString= datestr(t.Date); %Convert the cell array of dates to strings

Close= table2array(t(:,5));

l=linspace(1,2616,2616);

figure(1)

histogram(Close, 'Normalization', 'pdf', 'BinWidth', 10);

title('Google Stock Data PDF');

%Audio Signal

FID= fopen('rec\_01\_speech.raw', 'r');

if FID<0

 error('Failed to Open File');

end

D=fread(FID, inf, 'int16');

fclose(FID);

figure(2)

histogram(D, 'Normalization', 'pdf');

title('Audio Signal Data PDF');

%2 Mean and Variance Gaussian Model

%Google

Gtotal= sum(Close);

Gmean= Gtotal/length(Close);

Gsquare= Close.^2;

Gvar= ((sum(Gsquare))/length(Gsquare))- (Gmean)^2;

figure(3)

histogram(Close, 'Normalization', 'pdf', 'BinWidth', 10);

title('Google Stock Data PDF with Gaussian Model');

hold on

histfit(Close)

%Audio

Atotal=sum(D);

Amean= Atotal/length(D);

Asquare=D.^2;

Avar= ((sum(Asquare))/length(Asquare))-(Amean)^2;

figure(4)

histogram(D, 'Normalization', 'pdf');

title('Audio Signal Data PDF With Gaussian Model');

hold on

apd=fitdist(D, 'Normal');

x= -41212:1:41212;

ay= pdf(apd, x);

plot(x, ay)

%3 Different Distribution on Google

gpd=fitdist(Close, 'Burr');

x\_values= -1307:1:1308;

y=pdf(gpd, x\_values);

figure(4)

histogram(Close, 'Normalization', 'pdf', 'BinWidth', 10);

title('Google Stock Data PDF with Gaussian Model and Burr Model');

hold on

plot(x\_values,y)

hold on

g2pd=fitdist(Close, 'Normal');

y2= pdf(g2pd, x\_values);

plot(x\_values, y2, 'g-')

%RMSE

A=y';

rmseNormal=rms(A-Close);

B= y2';

rmseBurr= rms(B-Close);

C= ay';

rmseAudio= rms(C-D);

# Conclusions

The histograms seem to be an accurate representation of the data. It was important to change the bin size for the Google stock histogram, because it allowed the data to be more accurately viewed. When it had larger bins, much of the variance in the data was not seen because of the large sample size. It is expected for the Google histogram to be all over the place and not evenly distributed, because the stock prices can vary so greatly and are not predictable. On the other hand, it makes sense that the audio signal presents a symmetric distribution, because it can only be present in a range of values since it is digitized. Because of these facts, it would make sense that the Gaussian, or normal, distribution does not fit well to the Google data, but does fit will to the audio signal data. The Burr Model may serve to more accurately fit the Google stock data, as shown in the graph with both the Gaussian and the Burr model. It covers a wider range of the data and is centered closer to the amplitude center of the data than the Gaussian is. The mean-squared error should be a good indicator of the accuracy of the models. However, it is identical for the Burr model and the Gaussian model. This surprised me. I am not sure if it an error that I made on my part, or that the models are equally accurate (or not accurate) to the data. However, both are much larger than that of the audio signal, and that may be an indicator that the Gaussian fit of the audio signal is still much more accurate and close than either the Gaussian or the Burr model to the Google data.