Vaughn Wiernicki

ECE 3522: Stochastic Processes in Signals and Systems

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

# Problem Statement

For the first problem the goal was to take in the Google Stock Data, using only the Closing Prices, and first run a window and frame analysis. From the plot obtained in the window frame analysis I then had to compute a linear regression as well as the mean value of the data. These three functions were then plotted on the same graph for comparison functions. For the second problem I used the speech signal provided and created a histogram of the amplitude of the signal. From this I had to loop through the data and normalize each bin in the data and re-plot the histogram as a function of probability.

# Approach and Results

For the first problem I mainly reused the code from the first computer assignment to read in the excel file and then go through and do a window frame analysis of the data for just a frame size of 1 and a window size of 7. After this was done the mean value was easy to compute as I just used MATLAB’s mean function on the amplitude variable. The linear regression was the more challenging part, where I used an equation to find the slope of the regression and from there solved for the intercept of the line. These three functions were all plotted on the same graph to achieve the following:

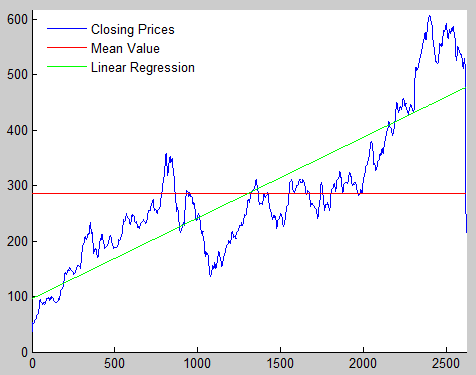


Figure : Google Closing Prices Analysis

We can see from the plot that the closing prices of the stock are identical to what was seen previously. The mean value is roughly half of the amplitude as the closing prices and bisects the linear regression into halves horizontally. This is what we would expect being the mean of all closing prices seen in the data. The linear regression also models the basic pattern of the closing prices signal. It increases with respect to time just as the stock prices do and is offset from the origin.

For the second problem I had to start fresh, aside from the few lines of code used to read in the raw file speech signal. I then decided to use MATLAB’s “accumarray” function to gather the amplitudes and determine which ones were most frequent in the signal. This however proved problematic initially as “accumarray” does not like to have subscript values lower than 0. I remedied this problem by giving the original signal an offset value so that its lowest value would be one. To then loop through the bins of the input signal I used a series of for loops in a similar fashion to the window frame analysis used in previous problems. The loops would go through and divide the number of occurrences of each amplitude by the total number of data points in the signal, giving us a representation of the probability of each of these amplitudes occurring. Before plotting I had to remember to factor in the original offset, which was done in the plotting functions. The following is the two plots received from the code thus far:

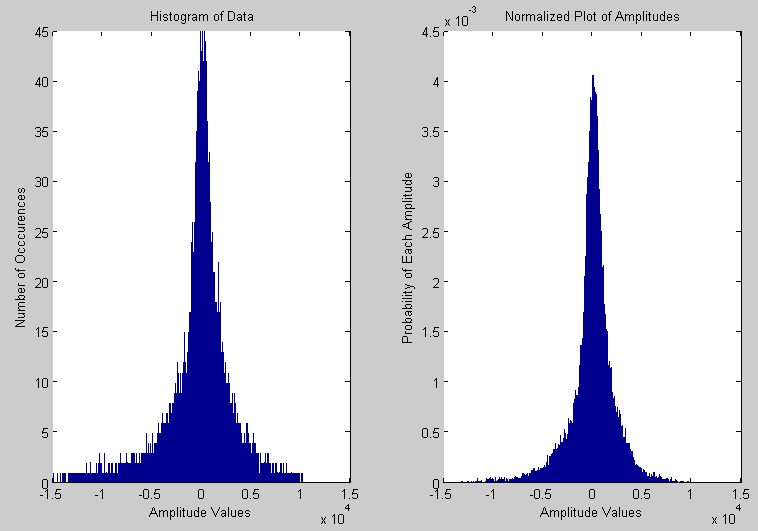


Figure : Histogram and Normalized Plot of the Speech Signal

The last part for this problem was to have a Cumulative Density Function of the signal. This was easily done by utilizing MATLAB’s “cdf” function. The following plot was received:

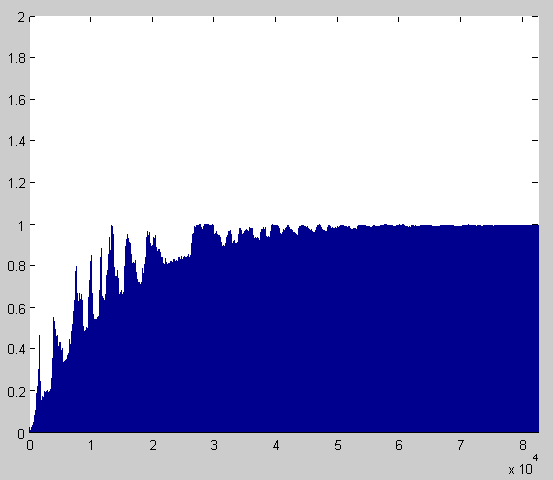


Figure : Cumulative Distribution Function

The biggest thing to take away from the histogram of the data is that it’s very close to a Gaussian distribution, which makes sense for something like a speech signal that is comprised of a number of periodic functions like sines and cosines. The amplitude of these functions center around 0, so we’re likely to see more amplitudes of 0, or values close to zero than we are to see extreme values in either direction. The cumulative distribution also matches what we would expect, the general trend of a logarithmic curve that tapers off as its amplitude gets close to one.

# MATLAB Code

clear; clc; clf; close all;

%Reading Google Stock Prices:

GOOG = xlsread('C:\Users\Vaughn II\Desktop\School\Stochastics\CA\_02\google\_v00.xlsx');

[R, C] = size(GOOG);

clse = zeros(1, R);

for l = 1:R

clse(1, l) = GOOG(l, 4);

end

%Loop through Data with Window of N, Frame of M:

M = 1;

N = 7;

for m = 1:1:length(M)

meandata = zeros(1, round(length(clse) / M(m)));

for n = 1:1:length(N)

data = zeros(1, N(n));

times = 1;

for j = 1:M(m):length(clse)

for k = 1:1:N(n)

x = M(m) - round(N(n)/2) + k + j;

if (x < 1)

data(1, k) = 0;

end

if (x > length(clse))

data(1, k) = 0;

end

if (1 <= x && x <= length(clse))

data(1, k) = clse(1, x);

end

end

%Analyzing the Windows:

meandata(1, times) = mean(data);

times = times + 1;

end

end

end

%Take the Mean value & Linear Regression of the data:

t = linspace(1, length(meandata), length(meandata));

meanV = ones(size(t))\*mean(meandata);

slope = (length(t)\*sum(t.\*clse)-sum(t)\*sum(clse) ) / (length(t)\*sum(t.^2)-(sum(t))^2);

inter = mean(clse)- slope \* mean(t);

B = slope .\* t + inter;

%Plot the graphs:

hold on

plot(t, meandata, 'b', 'DisplayName', 'Closing Prices');

plot(t, meanV, 'r','DisplayName', 'Mean Value');

plot(t, B', 'g','DisplayName', 'Linear Regression');

legend('-DynamicLegend', 'Location', 'NorthWest');

legend('boxoff');

axis([0, length(meandata), 0, (max(meandata) + 10)]);

--------------------------------------------------------------------------------------------------------------------------------------------

clear; clc; clf; close all;

%Reading Data:

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

speech = fread(fileID, inf, 'int16');

fclose(fileID);

%Create Histogram of Data:

binS = 10; %Bin Size.

shift = -14994;

Size = length(speech);

for m = 1:1:Size

speech(m, 1) = speech(m, 1) - shift;

end

A = accumarray(speech, 1);

B = A; %Create a copy of A.

%Go through Bins and Normalize each:

for n = 1:binS:length(A)

data = zeros(1, binS);

for o = 1:1:binS

pos = n + o -1;

if pos < length(A)

data(1, o) = A(pos, 1);

end

if pos > length(A)

break

end

end

per = sum(data) / Size;

for p = 1:1:binS

B((n + p - 1), 1) = per;

end

end

%Get the Cumulative Distribution Function:

q = linspace(1, Size, Size);

y = cdf('exp', q', speech);

%Plot the Result:

t = linspace(1, length(A), length(A));

subplot(1, 2, 1);

bar((t+shift), A);

title('Histogram of Data');

xlabel('Amplitude Values');

ylabel('Number of Occcurences');

u = linspace(1, length(B), length(B));

subplot(1, 2, 2);

bar((u+shift), B);

title('Normalized Plot of Amplitudes');

xlabel('Amplitude Values');

ylabel('Probability of Each Amplitude');

figure;

bar(q', y);

axis([0, Size, 0, (max(y)+1)]);

# Conclusions

Linear Regression models can reflect the general trend of data in a long scale of time, which is good for things like mutual funds that want to steadily grow with time. A histogram of data can also be helpful to see how an input signal trends. With the type of analysis we performed we can see how often a signal is likely to continue in either a positive or negative direction. Using these simple statistical analysis methods we can learn a lot about a time variant signal and how it behaves.