Devin Trejo

ECE 3522: Stochastic Processes in Signals and Systems

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

I. PROBLEM STATEMENT

The purpose of this analysis is to take a set of data and perform statistical that will later allow us to predict future values. We again use Google's closing stock price for the past eleven years and a generic speech signal to perform our tests on.

The first task has us plot the average stock price for every week in those eleven years. We then fit a linear regression to the data to see the general trend of the stock price. What do these plots tell us about Google's stock? How do the two plots relate to one another and how can we use this analysis to help us better invest in Google?

In the next task we take a speech signal and plot its histogram. We set some parameters such as using a bin size of 10 and setting the range to be between ±32767 since our signal is stored in 16 bit integers. Then we normalize the data in each bin by dividing by the total number of samples in the entire signal. We compare this histogram to a cumulative distribution function (CDF). Again how can we use these histograms to better understand our speech signal?

II. APPROACH AND RESULTS

In the last assignment we learned how to load in external data into MatLab. We utilize this procedure to load in our Google data. We then perform similar functions we explore previously to find a mean value of the stock data in periods of 7 days at a time (window = 7) with 1 day intervals (frame = 1). Now we fit a linear regression to the data to see a trend. The resulting plot is seen below.

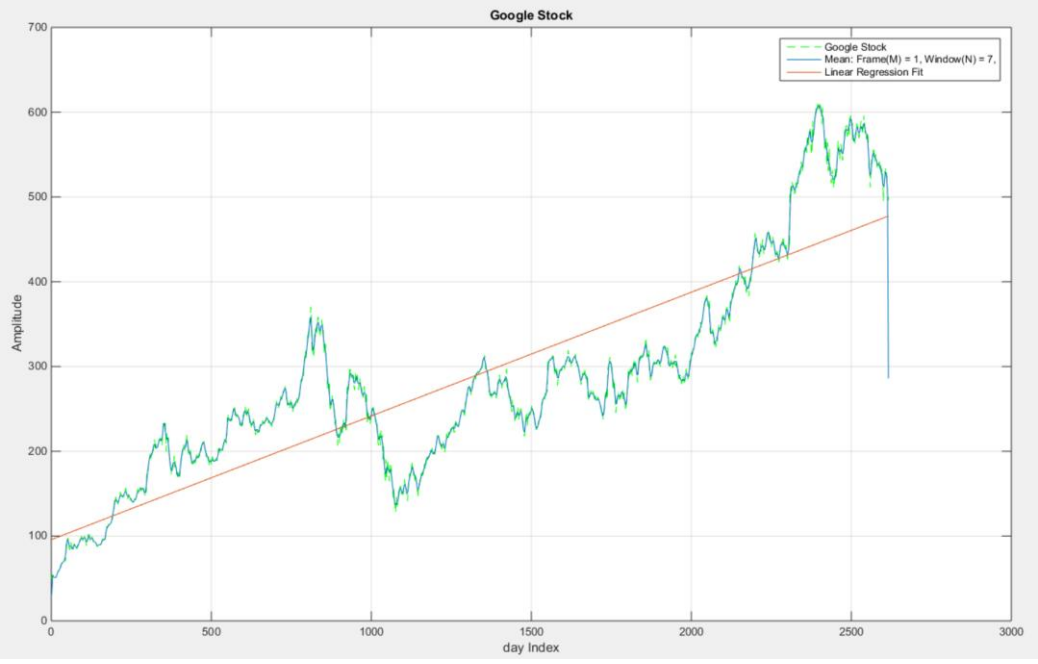

Figure 1: Google stock data, overlaid with its mean representation and a linear regression.

Google's stock in the past few years has overall had a positive trend. Comparing the trend line to the stock price we can determine certain time periods where it was beneficial to purchase stock. Anytime Google stock is below the

trends line we can say it is a good time to buy stock. Anytime the stock price is above the trend line we say you should sell. The problem with this thought process is that we can only compute this trend line if we know the actual data. We can interpret the trend line into the future but now we are guessing. The best model can never take into account outlier events that can drastically change the overall trend. You can have a prediction model that can take into account various parameters but no one model is perfect.

Next we will look at another prediction model using a speech signal. Histograms can tell us very useful information in a single plot. By breaking the data into larger groups (or bins) histograms are able to showcase the probability that a certain distribution will occur. Below is a histogram for our speech signal normalized to the total number of samples in our signal. Also, we plot a cumulative distribution of the speech data.

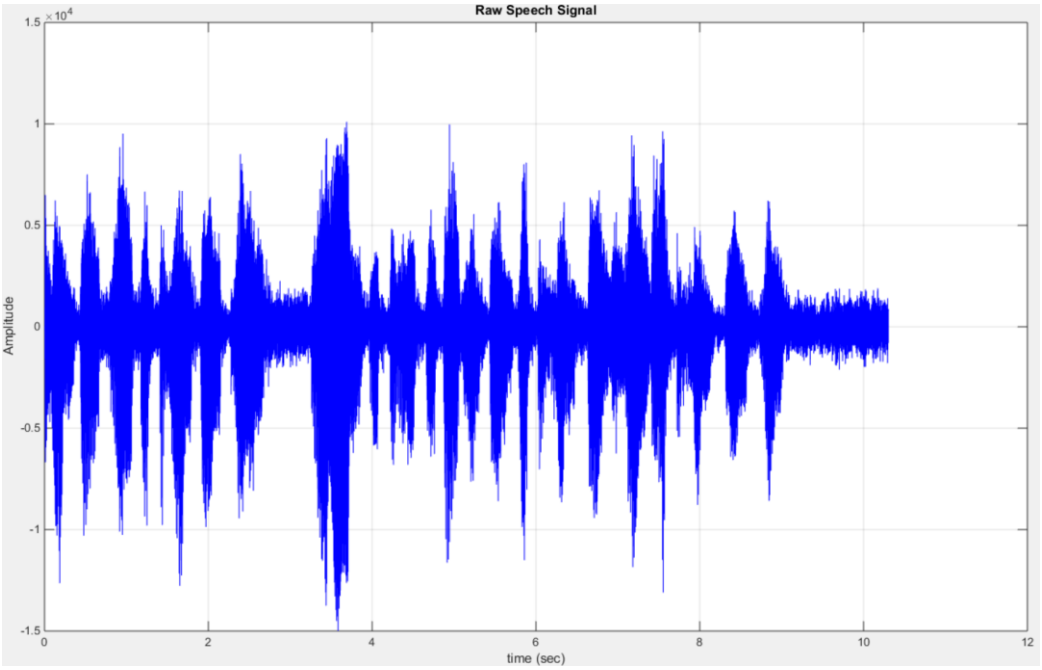

Figure 2: A graphical representation of our speech signal.

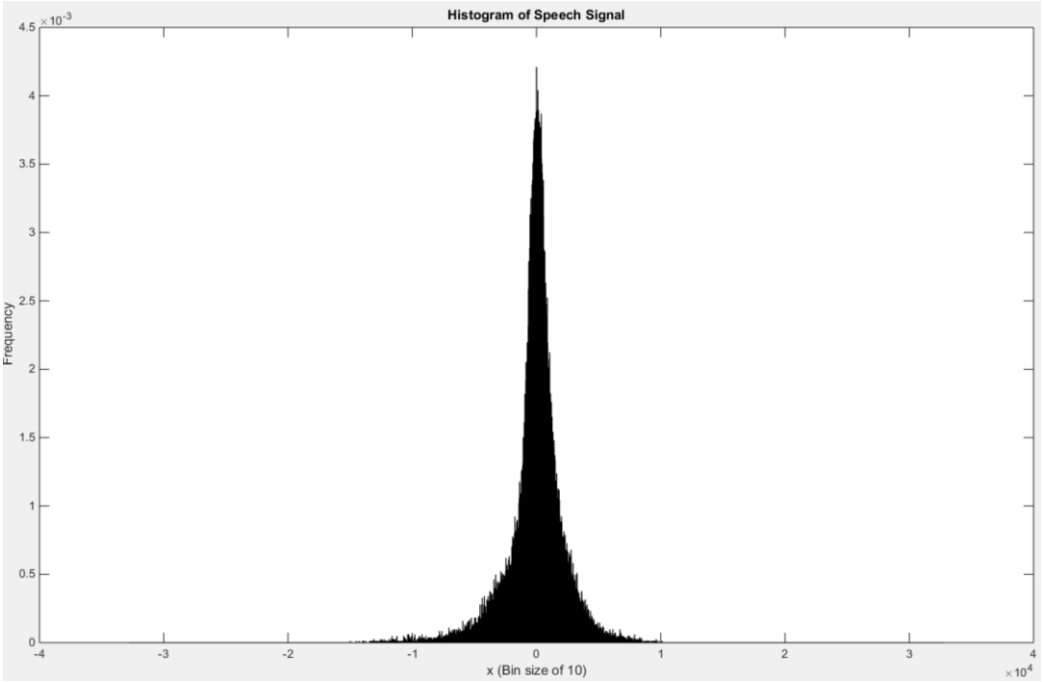

Figure 3: Histogram of our speech signal broken up into bin sizes of 10.

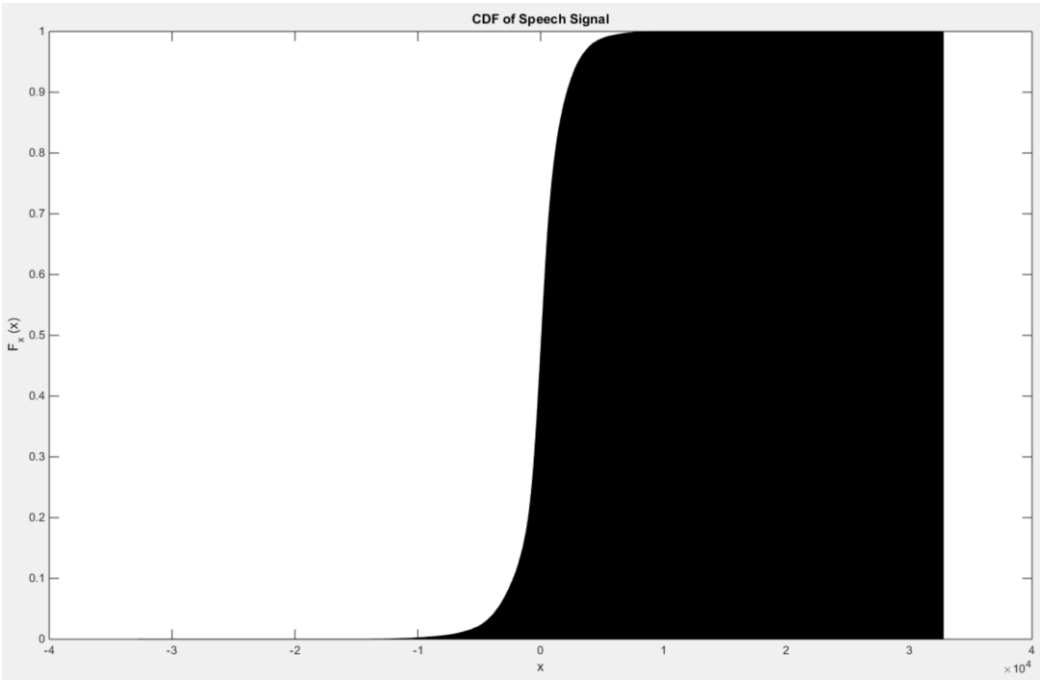

Figure 4: Cumulative Distribution of our speech signal.

At first glance we can see that our most common range of values occurs for values between our bin ranging from $[-5,5]$. The histogram tells us the range is most common from the peak in frequency at that range and in the last assignment we noted that the mean for this speech signal was zero. We also note that we have a bell curve centered around zero. The range of signal may be from ±32767 but those extreme values hardly occur. If we collectively take our entire sample space into account we note that the frequency that a values occurs becomes one.

The CDF plot also concurs with the same conclusion. Where our slope is steepest we can say we have low variance. The slope is steepest (by inspection) when $x = 0$. The probability that our speech signal takes on the amplitude of -32767 is practically zero. The slope of the CDF plot flattens out at these values which tells us there is high variance at these bins.

III. MATLAB CODE

```
%%
clear; clc; close all;

% Let's first open the raw speech data file and store its values in a
% vector fn
%
fp=fopen('rec_01_speech.raw', 'r');
% Test Sine Wave
    %fp=fopen('rec_01_sine.raw', 'r');
fn=fread(fp,inf,'int16');
fclose(fp);

L_speech = length(fn);

% Let's open the xls data file and store its values in avector fn
%
google_v00 = xlsread('google_v00.xlsx');
% google_open = google_v00(:,1);
% google_high = google_v00(:,2);
% google_low = google_v00(:,3);
google_close = google_v00(:,4);
L_googleClose = length(google_close);

clear google_v00

% Let us find the min/max val, mean, median, and variance
%
google_min = min(google_close);
google_max = max(google_close);
google_mean = mean(google_close);
google_median = median(google_close);
google_var = var(google_close);

% Print our findings
%

out = sprintf('Google data: min = %f, max = %f, mean = %f, median = %f, variance = %f
\n'...
    , google_min, google_max, google_mean, google_median, google_var);
disp(out);
```

1 We use the procedure from last assignment to load in the speech/Google stock price into MatLab. One important note needed for task 2 is that the speech signal is read in as 16 bit integers. When we construct a histogram later this information will be important.

```
% Now we separate into frames and windows.
%
M = [1];
N = [7];

num_frames = 1+round(L_googleClose / M);
windowMean = zeros(length(N),L_googleClose);
for i=1:num_frames
    sig_wbuf = zeros(1, N);
    Mp = M*(i-1)+M/2;
    windowStart = Mp-N/2;
    windowEnd = windowStart+N-1;


    windowStart = round(windowStart);
    windowEnd = round(windowEnd);


    % transfer the data to this buffer:
    %  note that this is really expensive computationally
    %
    for j = 1:N
        index = windowStart + (j - 1);
        if ((index > 0) && (index <= L_googleClose))
            sig_wbuf(j) = google_close(index);
        end
    end

    windowMean_t = mean(sig_wbuf);

   for j = 1:M
        index3 = Mp + (j - 1) - (M/2);
        if ((index3 > 0) && (index3 <= L_googleClose))
            windowMean(index3) = windowMean_t;
        end
   end
end
```

2 We break Google's stock into smaller windows of 7 days. We find the mean of this data at one day intervals. We store these means into an array to plot later.

```
% Let us plot
%

figure('name','[ECE 3522] Class Assignment [2]');
% We are given a sample frequency of 8 kHz
%
fs = 8000;
L_speech = length(fn);
timeL = L_speech/fs;
t= linspace(0, timeL, L_speech);
plot(t, fn, 'b');
grid on
xlabel('time (sec)');
ylabel('Amplitude');
title('Raw Speech Signal');


figure('name','[ECE 3522] Class Assignment [2]');
dDays = 1:1:L_googleClose;
plot(dDays, google_close, '--g');
hold on
xlabel('day Index');
ylabel('Amplitude');
title('Google Data');
plot(dDays, windowMean(:));
title('Google Stock');

grid on

% Linear Regression Operation
%
P = polyfit(dDays', google_close(:), 1);
G = polyval(P,dDays);
plot(dDays,G)
hold off

legend('Google Stock', sprintf('Mean: Frame(M) = %d, Window(N) = %d,',M ,N), 'Linear
Regression Fit')
```

3 Now we plot all the data we found. (See Figure 1)

```
% Spectrogram WRT bin Size
% Bin size = 10
%

bin_size = 10;

bounds = [round(-32767/5)*5:bin_size:round(32767/5)*5];
figure('name','[ECE 3522] Class Assignment [2]');
histogram(fn, bounds, 'Normalization', 'probability');
title('Histogram of Speech Signal');
xlabel(sprintf('x (Bin size of %d)',bin_size));
ylabel('Frequency');

figure('name','[ECE 3522] Class Assignment [2]');
h = histogram(fn, bounds, 'Normalization', 'cdf');
title('CDF of Speech Signal');
xlabel('x');
ylabel('F x (x)');
```

4 For task 2 we work with histograms. MatLab's built in histogram function has flexibility to accommodate for the parameters we are seeking. First we break up our range of possible values (16 bit integers) into bin sizes of 10. We can pass these bin intervals into the histogram function to get our desired plot. Our second figure plots normalizes our histogram into a cumulative density function by passing it the parameter 'cdf'.

IV. CONCLUSIONS

In bringing all the concepts gathered from the linear regression, probability mass functions, and cumulative distribution function we now have some tools to predict future. Understanding these concepts allows us to build models that focus on trends to anticipate things like weather, stock market, sports, and other seemingly random events. The more data we incorporate into our analysis permit us to build better models. For example, take this limited view of Google's stock from day 1410 to 1520.

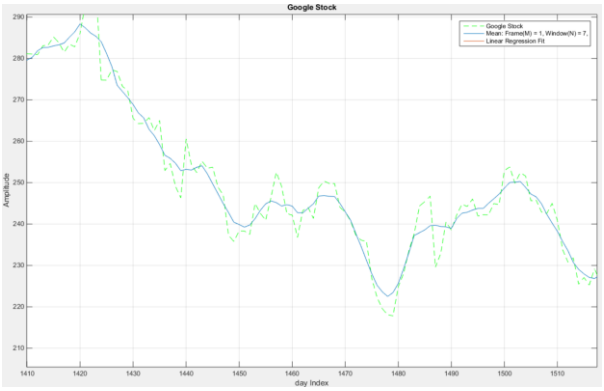

Figure 5: Google stock from day 1410 to 1520.

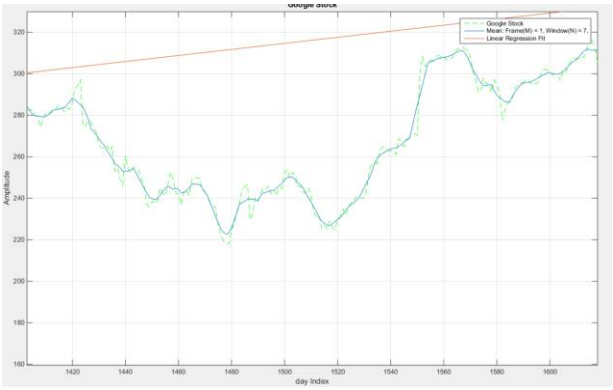

Figure 6: Google stock from day 1410 to 1610.

If we were to build a model on this limited range (Figure 5) of google stock we would say that Google has a stock trend that decreases in value. We know from our analysis that Google had an overall positive trend however. Building a model on just this range of data would produce bad prediction models. Expanding our view by another 100 days (Figure 6) will showcase how this range of days was a just a period of time where Google's stock decreased in value. We are still however below the overall trend of Google stock for the entire eleven years. No matter what day in this range you are in it would still be a good time to buy Google stock.