**Computer Assignment 2: Regression and Histograms**

John Lee

ECE 3522: Stochastic Processes in Signals

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

# Problem Statement

On this computer assignment, we will be continuing our work with the same signal files given from the past computer assignment. We will be asked to plot different types of functions to compare and contrast our results based on statistical data. This experiment will be distributed into two portions.

The first portion of the assignment is to plot a linear regression line of the google stock market file and compare its linear fit to standard plot which are set with a specific frame/window size.

The second portion of the experiment is to plot a histogram of the speech signal file. The purpose of this, is to understand how histograms function and their relevance with the cumulative distribution function.

# Approach and Results

The first part of this experiment relates to the comparison of two different plot distributions. As we managed to obtain both graphs that represent the google stock market data, we noticed that the signal plot from **Figure 1** represents the overall statistical data for the closing column. As we were given a set value for frame/window size, we can note that the RMS/Time data plot displays a rugged but increasing line of all values from the closing column. Once we managed to plot the first figure, we needed to create a function that uses the formula for linear regression. We know that the google stocks price could be determined as a linear formula (in terms of time)

$$\left[Y=mx+b\right] or [Stock price=a\*t+b] $$

Since we are looking for the linear regression of the stock price, our values of [a, b] will contain a set of formulas:

$$a= \frac{n\sum\_{}^{}xy-\sum\_{}^{}x\sum\_{}^{}y }{n\sum\_{}^{}x^{2}-\sum\_{}^{}x^{2}}$$

$$b= \overline{y}-a\overline{x}$$

**Equation 1.** Linear Regression Formula



**Figure 1.** Signal plot vs Linear regression

With these formulas, it can simulate and create a straight line which contains all of the values in increasing patterns. We could notice that the highest peak values for the first plot can be displayed through the highest values records on the linear regression.

The second part of this experiment, we continue working with the second signal, the speech file. We need to obtain the histogram of the signal file to obtain a set of histograms with their respective ranges. These histograms generate the repetition of values that are inside of the signal speech file. Note that this file is an array of 82425x1 values. The histogram will record all values into a bar graph, **Figures 2-10** represents ten plots for each bins, which contains accurate measurements obtained after plotting the histogram in different ranges. Each plot represents the ranges of increment/decrement of 10. We could notice that the bar graph displays the maximum value at higher ranges.

Notice that the change of ranges, affect the values retrieved from the program. Since we had 10 bins, each bin will contain a set of range values to observe the bar graph behavior as we select it on specific locations. **Figure 11** displays the cumulative distribution function of the .raw file. The graph represents the probability values taken from the speech file, we could observe that the probability gets close to 1, when the values are positive.



**Figure 2.** Range [-45:-35]



**Figure 3.** Range [-35:-25]



**Figure 4.** Range [-25:-15]



**Figure 5.** Range [-15:- 5]



**Figure 6.** Range [-5:5]



**Figure 7.** Range [5:15]



**Figure 8.** Range [15:25]



**Figure 9.** Range [25:35]



**Figure 10.** Range [35:45]



**Figure 11.** CDF of Speech File

# MATLAB Code

Just like the past assignment, **Figure 12** reveals the first part of this exercise, which is to simply plot the google stocks data within a frame size of 1 and a window size of 7. Note that this code contains a number of for loops to generate each window manually, instead of using function statements.

 

**Figure 12.** Part 1 Google Stocks Code

In order to plot the linear regression function of the google stocks, we will first need to build a function statement which contains the linear regression formula from **Equation 1**. Once we build a function, we will need set the variables [x,y] to their appropiate columns from the stockmarket data sheet. Plot both variables using the function formula we created from. This will generate the plot for the linear regression of the google stock market data sheet.



**Figure 13.** Part 1. Linear regression Google Stocks



**Figure 14.** Function (Linear regression formula)

The final portion of the experiment, will be to generate a histogram of the speech file given from the first computer assignment. We will need to load the audio file and set its parameters since is a 16 bit audio file. Once we complete this, we will need to plot a histogram of the audio file. We are given a 10 bins for the files, by which each bin will contain a unique number of range. Using histogram (‘FILE’,#:#) command, will highlight the desired range values from the entire histogram.



**Figure 15.** Histogram MATLAB code.

# Conclusions

We can reflect that the linear regression is a good tool to observe the line behavior of all unsorted data collected. It is a good way to observe the steady increase of the statistical data through the use of a line. It is crucial to create a separate function that contains the formulas of the linear regression, in order to obtain the correct plots. The use of the histograms are good ways to oversee the plots being made according to their ranges. Changing the ranges between the histogram will give us a better understanding in regards of where the higher values are located between specific ranges. Finally, plotting the cumulative distribution of the speech file shows the probability values taken from the array loaded from MATLAB.

.V. MATLAB CODE II

%% Part 1 Signal Data plot. No line regression here

clear;

clc;

%loading audio file onto matlab

a = xlsread('google\_v00.xlsx');

%defining M and N

M = [1];

N = [7];

%beginning of loop over signal M

for m = 1:length(M)

 Fig = figure('name', 'rms plot', 'numbertitle', 'off');

%beginning of loop over signal N

 for n = 1:length(N)

 start = 1;

 %Main loop over signal

 for main = 1:round(length(a)/M(m))

 %Pointer of frame location

 point = round(M(m)\*(main-1)+M(m)/2);

 p\_begin = round(point - N(n)/2);

 p\_end = round(point + N(n)/2-1);

 % when your point starts and ends, your values will always be positive.

 if(p\_begin < 1)

 p\_begin = 1;

 end

 if(p\_end > length(a))

 p\_end = length(a);

 end

 % Generating the windows (Beginning pointer vs Ending pointer)

 winvector = (p\_begin : p\_end);

 Average = sum(a(winvector).^2 / N(n));

 % Creating the repetitions of the M values.

 for main2 = 1:(M(m))

 RMS(start) = sqrt(Average);

 start = start + 1;

 % Since length contains numerous values, we will make sure it does

 % not overpass the values of start

 if start > length(a)

 break;

 end

 end

 end

 while start <= length(a)

 RMS(start) = 0;

 start = start + 1;

 end

 % Generating the plots for all values.

 figure(Fig);

 subplot(1 + length(N), 1, n);

 plot(RMS);

 ValDisp = sprintf('M = %d N = %d',M(m),N(n));

 title(ValDisp);

 xlabel('Time');

 ylabel('RMS');

 % Plot of input signal

 figure(Fig);

 subplot(1 + length(N), 1, n + 1);

 plot(a);

 title('Signal');

 xlabel('Time');

 ylabel('Amplitude');

 end

end

%% Part 1. Linear regression function

clear;

clc;

data = xlsread('google\_v00.xlsx');

x = data(:,1); %Data set from open column

y = data(:,4); %Data set from close column

[a0, a1] = func(x , y); %Recalling function

 %Linear regression formula

plot(x,y)

xlabel('Open Column');

ylabel('Close Column');

%% Part 2. Histrograms

clear;

clc;

%loading audio file onto matlab

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

b = fread(Faudio, inf, 'int16');

fclose(Faudio);

figure(1)

histogram(b,45:55) %Change range values here

xlabel('45:55'); %Change range values here

figure(2)

cdfplot(b) %Plotting CDF of speech file