William O’Mullan

ECE 3522: Stochastic Processing

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

# Problem Statement

This computer assignment introduced us to linear regression functions and the use of histograms in statistical analysis. For the first task we needed to evaluate the mean of the Google stock closing prices from our first computer assignment with a frame duration of one day and a window duration of seven days and overlay that on the plot of the closing prices themselves. Next we computed a linear regression model for the whole data set and plotted it over both the means and the actual prices themselves. For the second task we needed to plot a histogram of the speech signal with a range of +/- 32767.0, a bin size of ten samples, and each bin should be a function of the middle value of the bin (e.g. the first bin is centered at zero and the rest are computed from that reference). For the histogram the number of samples in each bin was normalized by the total number of samples in the file giving us the probability density function. Separately, we plotted the cumulative distribution function as a comparison to the probability distribution function.

# Approach and Results

1. Task no. 1

Plotting the closing prices and overlaying it with the mean with respect to the window and frame durations given in the assignment was a simple matter of code recursion to the first computer assignment. The real challenge was in finding the linear regression to use for this data set. Since our data set had one explanatory variable, the stock’s closing prices, and one dependent variable, the mean values, I chose to go with a simple linear regression. Simple linear regression uses the equation:

$$f= α+ βx$$

Where:

$$x=explanatory variable=Google closing prices$$

And:

$$y=dependent variable=closing price mean values$$

The slope for the simple linear regression is the covariance between the explanatory and dependent variables divided by the variance of the explanatory variable:

$$β=\frac{\sum\_{i=1}^{n}(x\_{i}-\overbar{x})(y\_{i}-\overbar{y})}{\sum\_{i=1}^{n}(x\_{i}-\overbar{x})}=\frac{Cov[x,y]}{Var[x]}$$

The y-intercept for the simple linear regression is given by subtracting the mean of the dependent variable by the slope multiplied by the mean of the explanatory variable:

$$α=\overbar{y}-β\overbar{x}$$

Essentially the simple linear regression between the closing prices and the mean values we computed from our window-frame analysis gives us an equation of how closely our dependent and explanatory variables are related to one another adjusted by how far they vary from their average values. Solving for these values in MATLAB is simpler than it appears thanks to the inbuilt functions for mean and covariance.

1. Task no. 2

Finding the histogram of the speech signal is far simpler in MATLAB than computing a linear regression is. To generate the correct bin edges and widths, I concatenated two arrays that stretched from five and negative five to either end of the range in increments of ten into a single array. I utilized these custom bin edges and the inbuilt normalization functions for the probability density function (PDF) and cumulative distribution function (CDF) to generate the necessary histograms.

To understand our findings, a solid understanding of how the CDF and PDF are related is necessary. The CDF tells us how probable it is that any random variable $X$ falls in the range from $[-\infty ,x]$. Formally:

$$Cumulative Distribution Function=F\_{X}\left(x\right)=P[X\leq x]$$

Using this we can find that:

$$Probability Density Function= f\_{X}\left(x\right)=\frac{dF\_{X}\left(x\right)}{dx}$$

The derivative of the CDF is the PDF, this means that the PDF is how probable it is that a variable equals one value rather than a range of values.

# MATLAB Code

Task no. 1 -- The Mean and Simple Linear Regression of Google Stock Prices 1

Task no. 2 -- The Histogram and Cumulative Distribution of 'rec\_01\_speech.raw' 4

% Computer Assignment no. 2
% Stochastic Processing
% Bill O'Mullan

## Task no. 1 -- The Mean and Simple Linear Regression of Google Stock Prices

clear;close all;clc;

% load .xlsx file -- full sheet = raw, text = txt,
% numeric data = stock\_prices
[stock\_prices,txt,raw] = xlsread('google\_v00.xlsx');

% just evaluate the closing prices for each day
closing\_prices = zeros(length(stock\_prices),1);
for a = 1:length(stock\_prices)
 closing\_prices(a,1) = stock\_prices(a,4);
end

% generate x axis of plot for prices, mean, and linear regression
days = zeros(length(stock\_prices),1);
for a = 1:length(stock\_prices)
 days(a,1) = a;
end

% set window size of 8 and frame size of 2
N = 8;
M = 2;

% determine the number of prices, the number of frames, and set up
% answer arrays for population
tot\_prices = length(closing\_prices);
tot\_frames = round(tot\_prices/M);
prices\_mean = zeros(tot\_prices,1);
prices\_variance = zeros(tot\_prices,1);
mean\_variance = zeros(tot\_prices, 1);
prices\_mean\_covariance = zeros(tot\_prices, 1);
better\_beta\_input = zeros((tot\_prices/2),1);

% loop through the prices using specific window and frame sizes
for a = 1:tot\_frames

 % determine the size of the window about the center point of
 % the frame
 center\_pt = ((a-1)\*M) + (M/2);
 left\_bound = center\_pt - (N/2);
 right\_bound = center\_pt + (N/2);

 % zero stuff the variable for mean, variance, and covariance
 % calculation in case it exceeds the size of the input data set
 if ((left\_bound < 0) || (right\_bound > tot\_prices))
 input\_prices = zeros(N,1);
 input\_mean\_cov = zeros(N,1);
 end

 % transfer closing prices data in window to input\_prices
 % variable for calculations
 for b = 1:N
 index = left\_bound + (b-1);
 if((index > 0) && (index <= tot\_prices))
 input\_prices(b,1) = closing\_prices(index,1);
 end
 end

 % perform mean computations on input\_prices
 mean\_input = mean(input\_prices);

 % populate mean answer array with computations
 for c = 1:M
 index = left\_bound + (c - 1) + (N/2);
 if((index > 0) && (index <= tot\_prices))
 prices\_mean(index,1) = mean\_input;
 end
 end

 % establish input to compute covariance matrix of closing prices mean and
 % closing prices themselves
 for d = 1:N
 index = left\_bound + (d-1);
 if((index > 0) && (index <= tot\_prices))
 input\_mean\_cov(d,1) = prices\_mean(index,1);
 end
 end

 % compute covariance for closing prices and the mean values
 covariance\_matrix = cov(input\_mean\_cov, input\_prices);

 % split covariance matrix into its component values for evaluation
 prices\_variance\_input = covariance\_matrix(2,2);
 mean\_variance\_input = covariance\_matrix(1,1);
 prices\_mean\_covariance\_input = covariance\_matrix(1,2);

 % populate variance and covariance answer arrays with computations
 for e = 1:M
 index = left\_bound + (c - 1) + (N/2);
 if((index > 0) && (index <= tot\_prices))
 prices\_variance(index,1) = prices\_variance\_input;
 mean\_variance(index,1) = mean\_variance\_input;
 prices\_mean\_covariance(index,1) = prices\_mean\_covariance\_input;
 end
 end

 % determine slopes of all values for covariance and variance of the
 % mean
 beta\_input = prices\_mean\_covariance./mean\_variance;

 % account for zero stuffing in slope vector
 for f = 1:(length(beta\_input)/2)
 better\_beta\_input(f,1) = beta\_input(2\*f,1);
 end

 % find the average slope for simple linear regression
 beta = sum(better\_beta\_input)/length(better\_beta\_input);

 % determine y intercept for simple linear regression
 alpha = mean(closing\_prices) - (beta\*mean(prices\_mean));

 % determine the linear regression with respect to days since Google
 % stock inception
 linear\_regression = beta\*prices\_mean + alpha;
end

figure(1);clf;
hold on;
plot(days, closing\_prices);
plot(days, prices\_mean, 'g');
plot(days, linear\_regression, 'r');
title('Google Stock Prices: Frame Length 2 days, Window Length 8 days');
xlabel('Days (since inception)');
ylabel('Price (in USD)');
legend('Closing Prices', 'Mean', 'Linear Regression');



## Task no. 2 -- The Histogram and Cumulative Distribution of 'rec\_01\_speech.raw'

clear;close all;clc;

% open .raw file, load binary data as 16 bit integers, then close the
% file to avoid corruption
file = fopen('rec\_01\_speech.raw','r');
speech\_data = fread(file,inf,'int16');
fclose(file);

% set bin edges centered at zero for pdf and cdf with a range of +/- 32767
edges\_rightbound = -5:10:32767;
edges\_leftbound = -32767:10:-5;
edges = horzcat(edges\_leftbound, edges\_rightbound);

% plot histogram of speech\_data with a bin width of 10 samples, normalized
% by the total number of samples in the file (essentially determine the
% probability density function)
figure(2);clf;
h\_01 = histogram(speech\_data, 'Normalization', 'pdf', ...
 'DisplayStyle', 'stairs');
h\_01.BinEdges = edges;
title('Probability Density Function of Speech File');
ylabel('Probability');
xlabel('Sample Value');

% plot the cumulative density function with the same requirements as
% previous
figure(3);clf;
h\_02 = histogram(speech\_data, 'Normalization', 'cdf', ...
 'DisplayStyle', 'stairs');
h\_02.BinEdges = edges;
title('Cumulative Distribution Function of Speech File');
ylabel('Probability');
xlabel('Sample Value');





[*Published with MATLAB® R2014b*](http://www.mathworks.com/products/matlab)

# Conclusions

For task one, our simple linear regression function is essentially an extremely accurate line of best fit for the data. Where the mean plot (in green) traces the data fairly closely, the liner regression model (in red) shows the mean for the entire duration of the data set. In task two the PDF of the speech signal shows that most of the data in the signal has an amplitude between $-1∙10^{4}$ and $1∙10^{4}$ and the CDF corroborates this by showing that practically the entire signal is contained within the amplitude range of $[-\infty ,1∙10^{4}]$. These plots make sense because the speech waves are sinusoidal, every peak and trough must pass through the x axis and thus contain values close to zero.