William O’Mullan

ECE 3522: Stochastic Processing

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

# Problem Statement

This computer assignment focused on estimating variance in real time and comparing the pros and cons of different methods of estimation. There were three tasks in this assignment that accomplished this goal, to be followed for both the Google closing stock price data set and the speech recording we used in previous computer assignments. The first task was to simply find the global variance of the entire data set and plot it as a horizontal line. The second task addressed finding the variance in real time, we had to start with the first ten samples of the data set and find the variance for each $N$ samples of data where $N$ increased by one with each iteration. We then overlaid the plot of the variance calculated in this manner onto the line from task one. The third and final task was to estimate the variance using a frame/window analysis. For the Google data set we used a frame duration of one day and a window of thirty days and for the speech signal we used a frame duration of ten milliseconds and a window duration of thirty milliseconds. Like the previous task, we overlaid the plot of the variance found using this method onto the other variance plots.

# Approach and Results

Recall that variance is how far the values in any data set vary from the mean of that data set. This is represented as:

$$σ\_{x}^{2}=E\left[X^{2}\right]-\left\{E\left[X\right]\right\}^{2}$$

MATLAB’s $var(…)$ function automatically performs this calculation for whatever array is input to it. This function allows the bulk of thought to be focused on the format of our solution’s arguments and result rather than how to find the variance itself. For the first task, finding the global variance is as simple as inputting the entire data set into the $var(…)$ function and plotting its output as a vector that’s the same length as the original signal. Task two only required a single loop which starts at $N=10$ and iterates until $N$ is the length of the data set. We then use this value of $N$ to decide how many samples to parse from the data set for evaluation and the order in which the computations are fed into the solution array. For the third task we accomplished the frame/window analysis in the fashion that we have for the previous two assignments.

# MATLAB Code

% Computer Assignment no. 3
% Stochastic Processing
% Bill O'Mullan

## Data Set no. 1 -- Google Stock Prices

clear;close all;clc;

% Set up array for population with closing price data using MS Excel column
% designations from google\_v00.xlsx
close\_prices = 'E:E';

% Populate array with data from file
closing\_prices = xlsread('google\_v00',1,close\_prices);

% Estimate variance for entire data set
global\_variance = var(closing\_prices);

% Establish empty arrays for results
N\_variance = zeros(length(closing\_prices)-10,1);
WinFrm\_variance = zeros(length(closing\_prices),1);

% Starting with 10 samples, estimate variance with N varying from 0 to the
% max number of samples in file
for N = 10:1:length(closing\_prices)

 % Hand off number of samples to its own variable
 num\_samples = N;

 % Select number of samples from source file
 closing\_input = closing\_prices(1:N);

 % Perform variance operation on desired number of samples
 input\_var = var(closing\_input);

 % Populate answer array with solutions
 N\_variance((N-9),1) = input\_var;
end

% Set window duration of 30 days and frame duration of 2 days
N = 30;
M = 2;

% Determine the number of prices and the number of frames
tot\_prices = length(closing\_prices);
tot\_frames = round(tot\_prices/M);

% Estimate variance using Window-Frame analysis
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 input prices in case it exceeds the size
 % of the input data set
 if ((left\_bound < 0) || (right\_bound > tot\_prices))
 input\_prices = 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

 % Estimate variance of input\_prices
 input\_var = var(input\_prices);

 % Populate answer array with computations
 for c = 1:M
 index = left\_bound + (c - 1) + (N/2);
 if((index > 0) && (index <= tot\_prices))
 WinFrm\_variance(index,1) = input\_var;
 end
 end
end

% Generate x axis of plot for WinFrm\_variance
WinFrm\_x = zeros(length(closing\_prices),1);
for a = 1:length(closing\_prices)
 WinFrm\_x(a,1) = a;
end

% Generate x axis of plot for N\_variance
Nvar\_x = zeros(length(closing\_prices)-10,1);
for a = 10:length(closing\_prices)-9;
 Nvar\_x(a,1) = a;
end

% Generate x axis for global variance plot and establish global variance as
% a vector of that length
global\_x = zeros(length(closing\_prices),1);
GlobVar\_vector = zeros(length(closing\_prices),1);
for a = 1:length(closing\_prices)
 global\_x(a,1) = a;
 GlobVar\_vector(a,1) = global\_variance;
end

% Overlay different variance estimations on the same plot
figure(1);clf;
hold on;
p = plot(global\_x,GlobVar\_vector,WinFrm\_x,WinFrm\_variance, ...
 Nvar\_x,N\_variance);
p(1).LineWidth = 1;
p(1).Color = 'b';
p(1).LineStyle = '--';
p(2).LineWidth = 1;
p(2).Color = 'r';
p(3).LineWidth = 1;
p(3).Color = 'g';
title('Google Closing Stock Price Variance');
xlabel('Days');
ylabel('Variance');
legend('Global Variance','Window-Frame Variance','N Variance');



## Data Set no. 2 -- '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);

% Estimate variance for entire data set
global\_variance = var(speech\_data);

% Establish empty arrays for results
N\_variance = zeros(length(speech\_data)-10,1);
WinFrm\_variance = zeros(length(speech\_data),1);

% Starting with 10 samples, estimate variance with N varying from 0 to the
% max number of samples in file
for N = 10:1:length(speech\_data)

 % Hand off number of samples to its own variable
 num\_samples = N;

 % Select number of samples from source file
 speech\_input = speech\_data(1:N);

 % Perform variance operation on desired number of samples
 input\_var = var(speech\_input);

 % Populate answer array with solutions
 N\_variance((N-9),1) = input\_var;
end

% Set window duration of 30 ms and frame duration of 10 ms
N = 0.03\*8e3;
M = 0.01\*8e3;

% Determine the number of prices and the number of frames
tot\_samples = length(speech\_data);
tot\_frames = round(tot\_samples/M);

% Estimate variance using Window-Frame analysis
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 input samples in case it exceeds the size
 % of the input signal
 if ((left\_bound < 0) || (right\_bound > tot\_samples))
 input\_samples = zeros(N,1);
 end

 % Transfer speech data in window to input\_samples variable for
 % calculations
 for b = 1:N
 index = left\_bound + (b-1);
 if((index > 0) && (index <= tot\_samples))
 input\_samples(b,1) = speech\_data(index,1);
 end
 end

 % Estimate variance of input\_samples
 input\_var = var(input\_samples);

 % Populate answer array with computations
 for c = 1:M
 index = left\_bound + (c - 1) + (N/2);
 if((index > 0) && (index <= tot\_samples))
 WinFrm\_variance(index,1) = input\_var;
 end
 end
end

% generate x axis of plot for WinFrm\_variance, correcting for 8 kHz Fs
Fs = 8000;
WinFrm\_x = zeros(length(speech\_data),1);
for a = 1:length(speech\_data)
 WinFrm\_x(a,1) = a;
end
WinFrm\_x = WinFrm\_x/Fs;

% Generate x axis of plot for N\_variance, correcting for 8 kHz Fs
Nvar\_x = zeros(length(speech\_data)-10,1);
for a = 10:length(speech\_data)-9;
 Nvar\_x(a,1) = a;
end
Nvar\_x = Nvar\_x/Fs;

% Generate x axis for global variance plot and establish global variance as
% a vector of that length
global\_x = zeros(length(speech\_data),1);
GlobVar\_vector = zeros(length(speech\_data),1);
for a = 1:length(speech\_data)
 global\_x(a,1) = a;
 GlobVar\_vector(a,1) = global\_variance;
end
global\_x = global\_x/Fs;

% Overlay different variance estimations on the same plot
figure(1);clf;
hold on;
p = plot(global\_x,GlobVar\_vector,WinFrm\_x,WinFrm\_variance, ...
 Nvar\_x,N\_variance);
p(1).LineWidth = 1;
p(1).Color = 'b';
p(1).LineStyle = '--';
p(2).LineWidth = 1;
p(2).Color = 'r';
p(3).LineWidth = 1;
p(3).Color = 'g';
title('Speech Recording 01 Variance');
xlabel('Time (Seconds)');
ylabel('Variance');
legend('Global Variance','Window-Frame Variance','N Variance');



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

# Conclusions

The big question for this assignment is not how to find the variance in these different methods but rather *why* we would choose to find the variance in these fashions. No one method used in this assignment is better than another, they just tell the same information in different fashions. Naturally the method of analysis from task two allows us to estimate the variance for an incoming signal, eventually aligning with the global variance for the entire signal. With this knowledge we could perhaps estimate future values of a signal to help us eliminate noise since we know how far a value should vary from the mean. The frame/window approach shows us small “snap-shots” of the variance for portions of the signal and represents the energy present in different portions of the signal. We can see that there is equal area under the curve above and below the line of the global variance. The global variance of the data set is useful since, for this application, it allows us to diagnose the results from the previous two methods of variance computation.