Emilie Doyle

ECE 3522: Stochastic Process in Signals and Systems

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

# Problem Statement

The objective of this assignment is to learn more about the concept of variance within two sets of data, the Google stock price data and the audio signal data. The given information is the Google stock information since 2004 as well as the \*.raw file of the audio signal. The objective for the first part of the assignment was to estimate the variance of the entire sets of data. The objective of the next part was to progressively take the variance, starting from 10 samples of data and then progressing all the way through the full value of the data set. The last objective was to determine the variance through the use of the windows and frames analysis technique.

# Approach and Results.

For the first part of the assignment, I read in the data from the Excel file that had the Google stock data and into an array. Next I calculated the variance of the entire data set and put it through a loop to make an array that has the variance value for each. Then I plotted it against a linearly spaced array. For the audio signal I read the signal in and into an array. I followed a similar process for the audio signal in which I took the variance of the entire array and then put it through a loop in order to plot it against a linearly spaced array. The variance of all the Google data is 1.6194e+04 and the total variance of the audio signal is 4.1394e+06.

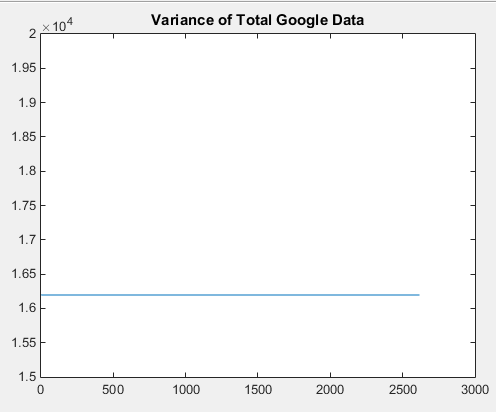


Figure : Total Variance of Google Data

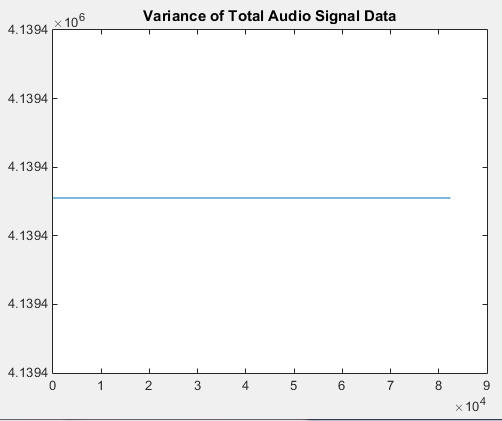


Figure : Total Variance of Audio Signal

For the second part of the assignment, I created a linearly spaced array to match the length that the progressive variances. The span of this is the span of the signal minus the first 10 that are included in the progressive variance. I then looped it through 10 through the length of the array and computed the partial variance- that is the variance at each progressive length of the signal. I then plotted the partial variance array versus a linearly spaced array. I held onto that and then plotted the horizontal line that is the variance of the total signal. I followed the exact same process for both the Google stock prices and the audio signal.

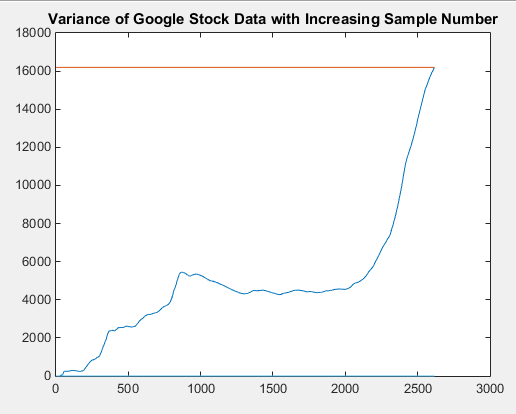


Figure : Progressive Variance of Google Stock Data (Blue) with Total Variance (Red)

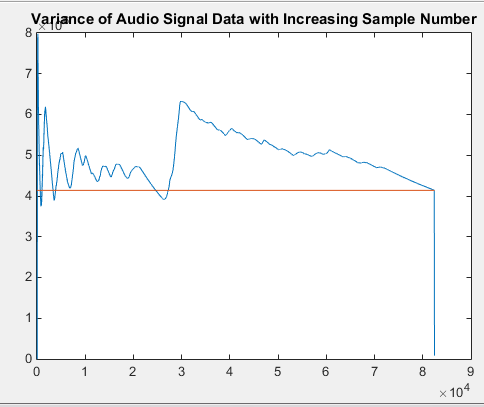


Figure : Progressive Variance of Audio Signal (Blue) with Total Variance (Red)

For the last part of the assignment, I reused code from previous assignments concerning windows and frames analysis in order to find the variance based on the windows and frames technique. The same process was used for both the Google stock data and the audio signal. Multiple loops were setup in order to sweep through the range of windows and frames, but since only one window and frame was specified, I just set the loops to those exact specifications. I used a function that I created to compute the variance based on window length. These were then plotted, and the progressive variance was plotted in comparison.

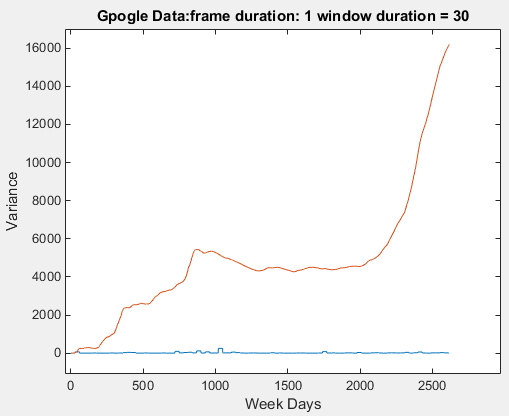


Figure : Google Progressive Variance (Red) and Windows and Frames Variance (Blue)

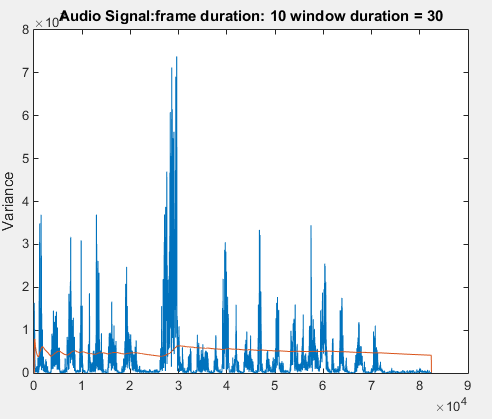


Figure : Audio Signal Progressive Variance (Red) and Windows and Frames Variance (Blue)

# MATLAB Code

%1

%Google

t= readtable('excelData.xlsx'); %read in data from excel

DateString= datestr(t.Date); %Convert the cell array of dates to strings

Close= table2array(t(:,5));

l=linspace(1,2616,2616);

gv= zeros(length(l));

gk= var(t.Close);

for i=1:2616

gv(i)= gk;

end

figure(1)

plot(l, gv)

title('Variance of Total Google Data');

ylim([1.5e4 2e4]);

%Audio Signal

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

if FID<0

error('Failed to Open File');

end

D=fread(FID, inf, 'int16');

fclose(FID);

m=linspace(1,82425, 82425); %number is obtained from length of array D

ak= var(D);

av=linspace(1,82425,82425);

for i=1:82425

av(i)=ak;

end

figure(2)

plot(m,av)

title('Variance of Total Audio Signal Data');

%%

%2

%Google

pgv=linspace(1,2607,2607);

n=linspace(10,2617,2607);

for i=10:length(t.Close)

pgv(i-9)= var(t.Close(1:i));

end

figure(3)

plot(n, pgv)

title('Variance of Google Stock Data with Increasing Sample Number');

hold on

plot(l, gv)

%Audio Signal

pav=linspace(1, 82425, 82425);

q= linspace(10, 82435, 82425);

for i=10:length(D)

pav(i-9)=var(D(1:i));

end

figure(4)

plot(q,pav)

title('Variance of Audio Signal Data with Increasing Sample Number');

hold on

plot(m,av)

%%

% %3

Google

GM= [1 7 14 30]; %frames

GN= [7 30]; %windows

GA=zeros(length(GM), length(GN), length(t.Close)); %store data

for M= 1:1

for N= 2:length(GN)

GA(M,N,:)= computeWVar(t.Close, GM(M), GN(N));

figure(5);

str = sprintf('Gpogle Data:frame duration: %d window duration = %d', GM(M), GN(N));

plot(squeeze(GA(M,N,:)));

title(str);

xlabel('Week Days');

ylabel('Variance');

hold on;

plot(n, pgv);

end

end

%Speech

SM= [5 10 20]; %frames

SN= [20 30]; %windows

SA=zeros(length(SM), length(SN), length(D)); %store data

for sgM= 2:2

for sgN= 2:length(SN)

SA(sgM,sgN,:)= computeWVar(D, SM(sgM), SN(sgN));

figure(6);

str = sprintf('Audio Signal:frame duration: %d window duration = %d', SM(sgM), SN(sgN));

plot(squeeze(SA(sgM,sgN,:)));

title(str);

ylabel('Variance');

hold on

plot(q, pav);

end

end

Compute Variance Function:

function windowVar = computeWVar( sig\_a, fdur, wdur )

for m = 1:fdur

for n = 1:wdur

% Create a matrix to store the output

windowVar = zeros(length(sig\_a),1);

% Find the center of the first frame

FC = (wdur+1)/2;

while (FC<length(sig\_a))

% Establish the bounds of the window subarray we're looking at

if (mod(wdur,2) == mod(fdur,2))

Mmin = FC - ((fdur-1)/2);

Mmax = FC + ((fdur-1)/2);

else

Mmin = FC - (fdur/2) - 1;

Mmax = FC + (fdur/2);

end

% Handle the beginning and end

if Mmin < 1

Mmin = 1;

end

if Mmax > length(sig\_a)

Mmax = length(sig\_a);

end

Window = sig\_a(Mmin:Mmax);

% Establish the bounds of the output subarray in which we're placing results

Nmin = FC - ((wdur-1)/2);

Nmax = FC + ((wdur-1)/2);

% Don't let your output matrix run larger than our input matrix

if (Nmax > length(windowVar))

Nmax = length(windowVar);

end

% Place the mean value of the window subarray in every value of the output

% frame subarray

for Ni = Nmin:Nmax

windowVar(Ni) = var(Window);

end

FC = FC + wdur; % Find the new window center

end

end

end

# Conclusions

The plots produced by part two of the assignment make sense because the progressive variance spans from 10 samples to the total amount of samples, and at the total amount of samples that is when the total variance matches the progressive or partial variance. This makes sense because the total variance includes all samples and when the progressive variance spans the total data set then it would include all of the samples so the values match. The windows and frames analysis of the variance appears to produce quite different results than the total variance, or the progressive variance. Both of the windows and frames approaches to analysis provided results that varied greatly than the progressive variance. The windows and frames provided values that were much larger than the progressive values. This may lead to an inaccurate representation of the variance of the data set. I think that the progressive variance is a better, more time-accurate representation of the variance of the data set.