[Tyler Castelli]

ECE 3512: Signals – Continuous and Discrete

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

# Problem Statement

The purpose of this assignment is to use different methods to estimate the variance in real-time of the data sets that were previously used in the computer assignments. We were asked to estimate the variance using three different methods, two of which were discussed previously in lectures or post assignments. The first method was to calculate the second central moment of the data sets. The second method was to calculate by finding the variance of the first 10 samples, then increasing the amount of samples we used until we reached the end of each file. The third method was to use the frame-window analysis used in previous assignments.

# Approach and Results

In order to calculate the variance by calculating the second central moment, I first needed to find the pmf of each set of data. This was done by creating a histogram of the data and normalizing it over the total amount of samples in each signal. Figures 1 and 2 show the histograms that were plotted:

Figure 1: PMF of Google Stock


Figure 2: PMF of Audio Signal

I loaded the values of each histogram into a respective vector and used this in the calculation of the second central moment. I also used the *var* function of MATLAB to check that the variance I was calculating. This resulted in a variance of 1.6191 x 104 for the Google stock data and 4.1393 x 106 for the speech signal. This is shown as the green lines in figures 3 and 4 below.

 For the second method, I created two vectors ranging from 10 to the length of e ach signal n increments of 1. I then used a loop to loop through the length of the vectors, and used the value of the loop number as to access the value at that index of the vectors. That value was used as the size of the sample set I was calculating, and a buffer vector containing that subset was created. I then found the mean of the data subsets and subtracted that value from the vectors. I estimated the variance by dividing the sum of the squared values of the subset by the number of samples in the subset. This was stored into a final variance vector, which is plotted against the number of samples of the subset that was used to generate the variance for that index value. This is shown as the blue line in figures 3 and 4.

 The final method was the same frame-window based analysis used in the previous assignment. I originally used a window of 30 for both signals, and a frame of 1 for the Google stock and 10 for the audio signal. However, for the google data, because the variance in this method is relative to the mean, the variance is hard to see using this window size when plotted along with the results from the other methods. I then changed the window size to 500 and re-plotted the results. These plots are shown as the orange lines in figures 3 and 4 below. Figure 5 shows the plot of the Google data with a window size of 30.

Figure 3: Variance of audio signal


Figure 4: Variance of Google stock using window size of 500


Figure 5: Variance of Google stock using window size of 30


# MATLAB Code

%%

%Part 1: Variance calculation of the data sets using central moment method

clc; clear all; close all;

%read in audio signal

f1 = fopen('rec\_01\_speech.raw');

a\_sig = (fread(f1,inf,'int16'))';

fclose(f1);

%read in google data

google\_data = xlsread('google\_v00');

data\_set = (google\_data(:,4))';

%Create histograms and normalize the data to get the pmf's of each data set

figure(271)

h1 = histogram(data\_set,'BinWidth',1);

title('PMF Google Stock');

ylabel('Frequeny of Stock Value');

xlabel('Stock Value')

h1.Normalization = 'pdf';

g\_pmf = h1.Values;

pause;

figure(271)

h2 = histogram(a\_sig,'Binwidth',1);

title('PMF of Speech Signal');

ylabel('Frequeny of Amplitude Value');

xlabel('Signal Amplitude')

h2.Normalization = 'pdf';

a\_pmf = h2.Values;

pause;

close all;

%Create vectors

%to hold values to be used in

%calculatons

g\_mean\_buffer = zeros(1,length(g\_pmf));

g\_cmoment = zeros(1,length(g\_pmf));

a\_mean\_buffer = zeros(1,length(a\_pmf));

a\_cmoment = zeros(1,length(a\_pmf));

%Fill vectors with appropiate values

for i = 1:length(g\_pmf)

 g\_mean\_buffer(i) = i\*g\_pmf(i);

end

for i = 1:length(a\_pmf)

 a\_mean\_buffer(i) = i\*a\_pmf(i);

end

%Calculate variance of Google Stock

g\_expect = sum(g\_mean\_buffer);

for i = 1:length(g\_pmf)

 g\_cmoment(i) = ((i-g\_expect).^2)\*g\_pmf(i);

end

g\_var2 = var(data\_set) %Variance function to have reference values

g\_var = sum(g\_cmoment);

%Calculate variance of audio signal

a\_expect = sum(a\_mean\_buffer);

for i = 1:length(a\_pmf)

 a\_cmoment(i) = ((i-a\_expect).^2)\*a\_pmf(i);

end

a\_var2 = var(a\_sig) %Variance function to have reference values

a\_var = sum(a\_cmoment);

%plot the data

gy = linspace(g\_var, g\_var,length(data\_set));

ay = linspace(a\_var,a\_var, length(a\_sig));

f1 = figure('name','Variance of Google Stock','numbertitle','off');

f2 = figure('name','Variance of Speech Signal','numbertitle','off');

figure(f1);

subplot(2,1,1);

hold on;

plot(1:length(data\_set),gy,'g-');

title('Variance of Google Stock');

xlabel('Time: Days');

ylabel('Variance');

axis([0 (length(data\_set)+100) 0 90000]);

subplot(2,1,2);

plot(1:length(data\_set),data\_set);

xlabel('Time: Days');

ylabel('Stock Value');

axis([0 (length(data\_set)+100) 0 900]);

figure(f2)

subplot(2,1,1);

hold on;

plot(1:length(a\_sig),ay,'g-');

title('Variance of Audio Signal');

ylabel('Variance');

xlabel('Time: ms');

axis([0 (length(a\_sig)+1000) 0 75000000]);

subplot(2,1,2);

plot(1:length(a\_sig),a\_sig);

xlabel('Time: ms');

ylabel('Amplitude');

axis([0 (length(a\_sig)+1000) -15000 15000]);

**This first section of code calculates the second central moment of the data sets. A histogram object is created using bin widths of 1. The data in the histogram is normalized by the total number of samples in the sets. This gives the pdf which is stored in a vector. The expectation, second moment, are found and then the variance is calculated from these values.**

%%

%Part 2: Variance calcultion of data sets using step method

%Set up vetors for sample points and variances to be plotted

Ng = 10:length(data\_set);

Na = 10:length(a\_sig);

g\_var\_est = zeros(1,length(Ng));

a\_var\_est = zeros(1,length(Na));

%calculate google stock variance

for i = 1:length(Ng)

 g\_buffer = data\_set(1:Ng(i));

 g\_mean = mean(g\_buffer);

 temp = g\_buffer-g\_mean;

 g\_var\_est(i) = (1/Ng(i))\*sum(temp.^2);

end

%overlay ontop of previous plot

figure(f1);

subplot(2,1,1);

plot(Ng,g\_var\_est);

%calculate audio signal variance

for i = 1:length(Na)

 a\_buffer = a\_sig(1:Na(i));

 a\_mean = mean(a\_buffer);

 temp = a\_buffer-a\_mean;

 a\_var\_est(i) = (1/Na(i))\*sum(temp.^2);

end

%overlay ontop of previous plot

figure(f2);

subplot(2,1,1);

plot(Na,a\_var\_est);

**This section of code uses a method where the variance is calculated within subsets of the data. Vectors containing the lengths of the subsets that will be taken ae created, and then, using a for loop, the subsets are created and filled with data, the mean and variance are calculated for each subset, and the variance for that subset is stored in a vector. The variances are then plotted against the number of samples of the subset that was used to calculate them.**

%%

%Part 3: Variance calculation using a centered window

Ng = 500;

Mg = 1;

Na = 30;

Ma = 10;

var\_valG(Mg,Ng,:) = get\_Var(Ng,data\_set,Mg);

var\_valA(Ma,Na,:) = get\_Var(Na,a\_sig,Ma);

figure(f1)

subplot(2,1,1);

plot(squeeze(var\_valG(Mg,Ng,:)));

str1 = sprintf('Central Moment Method');

str2 = sprintf('Increasing Sample Point Set method');

str3 = sprintf('Window Frame Method: Frame duration = 1 Window duration = 30');

legend(str1,str2,str3,'Location','northwest');

figure(f2)

subplot(2,1,1);

plot(squeeze(var\_valA(Ma,Na,:)));

str3 = sprintf('Window Frame Method: \n Frame duration = 10 Window duration = 30');

legend(str1,str2,str3,'Location','northeast');

**Here the same frame-window analysis as before is used. The frames are set to 1 and 10 for the Google stock and audio signal respectively and the windows are set t0 500 and 30 for the respective data sets. The get\_Var() function written in previous assignments is called to calculate the variance, which is then plotted on the same plot as the other variances that were calculated.**

# Conclusions

In the first problem to this assignment, the bins of the histograms needed to be set to 1 in order to capture all of the data and calculate accurate PMF’s for the data sets. The issue with this approach is that it created empty bins in the histogram because there may be no values that appear for that bin. However, when the values of the histogram are summed using a bin size of 1, the result is a 1, showing that the correct PMF has been found. Using the PMFs, the central second moment is easily calculated by applying the formula for finding the variance of a discrete PMF. For the second problem, the variance approaches that of the calculated value as more samples are taken and used to calculate the variance. This make sense because the variance will follow the same trend as the mean of the data. For the Google stock, the variance converges upwards to the calculated value because the trend of the mean of the data set is upwards. As more samples are used to find the variance, the variance will increase until it reaches the approximated value. From figure 4 and 5, one will also notice that the data set itself rises on an upward trend, and that the variance estimate roughly follows this trend. This can also be seen in the audio signal, as the variance oscillates around the variance value, eventually converging to the estimate from part 1. This trend is repeated in the audio signal which oscillates between a max and min value. The final problem to this assignment was covered mostly in previous assignments. Here we see that for the data, the variance is found based on the windows and frame sizes. The variance here is calculated based on the mean as usual, but the one will notice the plots do not follow a similar trend to that of the mean. This is because the variance is calculated on a window-by-window basis, not along the entire signal as the other two methods were. For the Google stock, the end of the third variance plot spikes because of zero stuffing at the end of the variance vectors; The vectors are zero stuffed to fill them to the appropriate size, which causes large numbers to be used in the variance computation, as well as a large number of zeros depending on the window size. The effects of the window size can be shown between figures 4 and 5, where a smaller window size causes a greater spike. This is due to the smaller window size using larger numbers in the final variance calculation as well as the zeros used to stuff that vector.