Marc Jurchak

ECE 3522: Stochastic Processes in Signals and Systems

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

# Problem Statement

Summarize the problem statement in one paragraph. Clearly state what the knowns are and what unknowns you must find.

 We will compute the variance of the speech signal and the Google stock prices three different ways. First we will compute a single value of the variance by computing the first and second moments of the data. Next we will compute the variance from the start of the signal to each point to see how the variance changes. Finaly we compute the variance on a window and frame basis so we can see the variance at each section of the signal.

# Approach and Results

Describe your approach to finding the unknowns. Use numbered figures, tables and equations where necessary.

To compute the constant value of variance $σ^{2}\_{X}$, we use the relationship $σ^{2}\_{X}=E\left[X^{2}\right]- \left(E[X]\right)^{2}$. Thus we need an expression for the first and second moments. We use:

$$E\left[X\right]= \sum\_{i}^{}x^{n}\_{i}p\_{X}(x\_{i})$$

From our previous project, we already have the PMF $p\_{X}(x\_{i})$ made from the normalized histogram. Thus we only need to multiply by $x^{n}\_{i}$ and sum the result. We plug the moments in the above equation to obtain the variance.

 For the second part, we use the following equation to compute variance on a rolling basis:

$$σ^{2}\_{X}=\frac{1}{N}\sum\_{i}^{}\left(x\_{mean}- x\_{i}\right)^{2}$$

This definition of the variance is more accurate, especially for when there are less data points. We plot this result on top of the constant variance we computed above in figure 1 and 2.



Figure 1: Constant and rolling variance of Google stock. Constant variance = 1.6191e4



Figure 2: Constant and rolling variance of speech signal. Constant variance = 4.1393e6

 In part three we calculate the variance by chunks of windows and frames. For the Google stock we used a frame size of 1 and a window size of 30, and for the speech signal we used a frame size of 30 and a window size of 240. We essentially use the code started in computer assignment 1 to compute variance in this way. Results are shown in figures 3 and 4.



Figure 3: Google stock signal variances.



Figure 4: Speech signal variances

# MATLAB Code

**Part 1**

clc;

clear;

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

speech = fread(fp,inf,'int16');

fclose(fp);

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

stock = google\_stock(:,4)';

% Use MATLAB var function to check result

matlab\_var\_speech = var(speech);

matlab\_var\_stock = var(stock);

% build Google stock histogram same as before

posedge\_stock = 1:1:2620;

posmiddle\_stock = 1:1:2619;

N\_stock = histcounts(stock,posedge\_stock);

N\_stock\_norm = N\_stock ./ length(stock);

% make sure PMF sums to 1

sum\_stock = sum(N\_stock\_norm)

% compute first and second moments of stock

%

PMF = N\_stock\_norm;

x = 1:length(posmiddle\_stock);

moment\_1\_stock = 0;

moment\_2\_stock = 0;

for n = 1:length(PMF)

 moment\_1\_stock = moment\_1\_stock + x(n).\*PMF(n);

 moment\_2\_stock = moment\_2\_stock + ((x(n)).^2).\*PMF(n);

end

% plug moments into variance equation to compute Google stock variance

variance\_stock = moment\_2\_stock - (moment\_1\_stock)^2;

% build speech signal histogram same as before

all\_speech = -32767:1:32767;

allmiddle\_speech = -32767:1:32767;

N\_speech = histcounts(speech,all\_speech);

all\_norm = N\_speech./length(speech);

% make sure PMF sums to 1

sum\_speech = sum(all\_norm)

% compute first and second moments of stock

%

PMF = all\_norm;

x = 1:length(allmiddle\_speech);

moment\_1\_speech = 0;

moment\_2\_speech = 0;

for n = 1:length(PMF)

 moment\_1\_speech = moment\_1\_speech + x(n).\*PMF(n);

 moment\_2\_speech = moment\_2\_speech + ((x(n)).^2).\*PMF(n);

end

% plug moments into variance equation to compute Google stock variance

variance\_speech = moment\_2\_speech - (moment\_1\_speech)^2;

% display results. Calculated variance should match the var() function.

matlab\_var\_speech

variance\_speech

matlab\_var\_stock

variance\_stock

**Part 2**

clear;

clc;

clf;

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

speech = fread(fp,inf,'int16');

fclose(fp);

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

stock = google\_stock(:,4)';

% compute rolling variance

stock\_variance = compute\_variance(stock);

speech\_variance = compute\_variance(speech);

figure(1)

plot(stock\_variance)

hold on

plot([1 length(stock)],[var(stock) var(stock)],'k-.')

xlabel('days')

ylabel('variance')

title('Google stock variance')

figure(2)

plot(speech\_variance)

hold on

plot([1 length(speech)],[var(speech) var(speech)],'k-.')

xlabel('sampled at 12.4 msec')

ylabel('variance')

title('speech signal variance')

**Compute\_variance Function**

function [ variance ] = compute\_variance( signal )

variance = 0;

sum = 0;

variance = zeros(1,length(signal) - 9); % computing variance from 10 to N

for i = 1:(length(signal) - 9)

 % buffer is the section of signal to compute the variance

 buffer = signal(1:(i + 9));

 buffer\_avg = mean(buffer);

 buffer = buffer - buffer\_avg;

 % sum the result

 for j = length(buffer)

 sum = sum + buffer(j)^2;

 end

 % final value for variance

 variance(i) = (sum.\*(1/length(buffer)));

end

end

**Part 3**

clear;

clc;

clf;

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

speech = fread(fp,inf,'int16');

fclose(fp);

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

stock = google\_stock(:,4)';

stock\_variance = compute\_variance(stock);

speech\_variance = compute\_variance(speech);

w\_stock\_variance = wfvariance(stock,1,30);

w\_speech\_variance = wfvariance(speech,80,240);

legend('-DynamicLegend')

figure(1)

hold on

plot(stock\_variance, 'DisplayName','rolling variance')

plot([1 length(stock)],[var(stock) var(stock)],'k-.', 'DisplayName','overall variance')

plot(w\_stock\_variance, 'DisplayName','variance by window')

hold off

title('Google Stock Variance')

xlabel('days')

ylabel('variance')

legend('-DynamicLegend')

figure(2)

hold on

plot(speech\_variance, 'DisplayName','rolling variance')

plot([1 length(speech)],[var(speech) var(speech)],'k-.', 'DisplayName','overall variance')

plot(w\_speech\_variance, 'DisplayName','variance by window')

hold off

title('speech signal Variance')

xlabel('msec')

ylabel('variance')

legend('-DynamicLegend')

**wfvariance Function**

function frame\_variance = wfvariance(sig\_a, fdur\_a, wdur\_a);

% declare local variables

%

sig\_wbuf = zeros(1, wdur\_a);

num\_samples = length(sig\_a);

num\_frames = 1+round(num\_samples / fdur\_a);

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

frame\_variance = zeros(length(sig\_a),1);

% loop over the entire signal

%

for i = 1:num\_frames

 % generate the pointers for how we will move through the data signal.

 % the center tells us where our frame is located and the ptr and right

 % indicate the reach of our window around that frame

 %

 n\_center = (i - 1) \* fdur\_a + floor(fdur\_a / 2);

 n\_left = n\_center - floor(wdur\_a / 2);

 n\_right = n\_left + wdur\_a - 1;

 % when the pointers exceed the index of the input data we won't be

 % adding enough samples to fill the full window. to solve this zero

 % stuffing will occur to ensure the buffer is always full of the same

 % number of samples

 %

 if( (n\_left < 0) || (n\_right > num\_samples) )

 sig\_wbuf = zeros(1, wdur\_a);

 end

 % transfer the data to this buffer:

 % note that this is really expensive computationally

 %

 for j = 1:wdur\_a

 index = n\_left + (j - 1);

 if ((index > 0) && (index <= num\_samples))

 sig\_wbuf(j) = sig\_a(index);

 end

 end

 % square the signal. divide it by the number of samples used and sum

 % the result to build the value for that frame

 %

 sum = 0;

 x = sig\_wbuf - mean(sig\_wbuf);

 for j = 1:length(x)

 sum = sum + x(j)^2;

 end

 win\_var = sum\*(1/length(sig\_wbuf));

 for j = 1:fdur\_a

 index = n\_center + (j - 1) - floor(fdur\_a/2);

 if ((index > 0) && (index <= num\_samples))

 frame\_variance(index) = win\_var;

 end

 end

end

# Conclusions

We found the most useful way to understand the variance of the signal is to plot variance on a rolling basis (part 2) since it has the information of the other 2 plots within it. From figures 1 and 2, we see that the final value matches our previous calculation of variance, and from figures 3 and 4 the slope approximates the windowed and framed variance at any small region. For example in figure 4, there is a sharp spike in variance at the point where the plot has a large slope for a period of time.

We can see why stock prices have a very subdued window and frame variance compared to the speech signal: At any small period of time, the stock price does not change much, while the speech signal goes from a high value to a low value in a very short period of time. We also notice that the rolling variance plot moves up and down, even with the Google stock prices. This is because when the signal dips towards the mean, the variance plot will decrease. In the case of the Google stock prices, it makes sense that the variance ultimately increases as stocks go up, since it is moving further from the mean ultimately.