Edward Moore

ECE 3512: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19121

# Problem Statement

In the real world data is often coming in continuously and variance will need to be calculated in real time. With this approach the value of variance will change as more data is acquired, opposed to calculating a single variance value for an entire set of data. Here we will calculate the variance in a variety of ways for the set of Google stock prices since inception and for the data from an audio signal.

# Approach and Results

The set of Google stock data since inception and the audio signal are shown below in figures 1 and 2. First, the variance of the entire set of Google stock data is calculated to be 1619. Next, starting with the first 10 samples and incrementing by 1 sample each time, the variance is calculated for the entire set with each iteration. Plotting these results along the way results in a plot of how the total variance value has changed with each new point of data for the stock price. This plot ends up growing from a small variance value to converge with the original variance calculated for the entire set of data. This is shown below in figures 3 and 4 with the red line representing the overall variance value and the blue line representing the variance value since the inception of Google stock.



Figure 1



Figure 2



Figure 3



Figure 4

Opposed to the resulting variation graph from increasing the amount of samples taken every time for Google stock, the audio signal’s variation was very small and not growing nearly exponentially. Instead it oscillates around zero. This is due to the fact that the Google stock price is always increasing where the audio signal oscillates around zero. Intuitively, if a signal is always increasing, the variance experienced over the entirety of the signal should be much greater than one that oscillates around a point.

Next, a window/ frame analysis approach was used to continuously measure variance of Google stock price and amplitude of the audio signal. This was done by calculating the variance for 30 points of data at a time, incrementing by one point of data at a time for the entire set of stock data. For the audio signal a frame of 10msec and window of 30ms was used. The stock price values for variance are much lower than in the previous methods since their accounting for a much shorter set of data with each calculation. This gives a much better idea of the short term variances the stock price is experiencing.



Figure 5

Overlaying the two plots illustrates how compounding the number of samples being calculated grows the variance at almost an exponential rate since that’s been the general trend of the stock price since its’ inception. Conversely, the variance calculated using the window/ frame analysis is significantly less since from each window the total change experienced can never be as much when looking at the whole set of data.



Figure 5

In figure 6, the window frame analysis of an audio signal we can see the variance is now much (a magnitude) greater than the previous variance calculations for the audio signal. This makes sense since the deviation between samples will be smaller when taken from the oscillating set of original audio data.



Figure 6

# MATLAB Code

filename = 'google\_v00.xlsx';

close = xlsread(filename, 'E:E');

plot(close, 'b')

title('Google Stock Price Since Inception');

xlabel('Days Since Inception');

ylabel('Stock Price');

v = var(close)

len = length(close)

for i = 0:len

 plot(i, v, 'r-')

 hold on

end

title('Variance Since Google Stock Inception');

xlabel('Days Since Stock Inception');

ylabel('Variance');

for i = 0:len-10

 for k = 10:len

 newclose = wkeep(close, k, 1);

 vnew = var(newclose);

 plot(k, vnew, 'b-')

 end

end

M = [1]; %(days) frame

N = [30]; %(days) window

 startPt = (N)/ 2; %picks a start point by moving over half the size of a window, avoids zero stuffing the ends

 newLength = floor((2616 - (N))/M); %calculates the array length of variances based on window & frame size

 windowVar = zeros(1, newLength); %creates an empty array to hold the variances

 for l = startPt:newLength

 windowVar(l) = var(close( (l+1-((N)/2)): (l+1+((N)/2))));%finds the variance over the specified window

 end

 plot(windowVar, 'b-') %plots the new mean values after the window/ frame analysis

xlabel('Days Since Stock Inception');

ylabel('Variance (Frame = 1, Window = 30)');

title('Window/ Frame Variance Analysis Google Stock Price')

%%

[y, Fs] = audioread('rec\_01\_speech.wav');

y = abs(y); %just want the absolute values of amplitude since that indicates the magnitude of the signal

len = length(y);

v = var(y);

for i = 0:len

 plot(i, v, 'r-')

 hold on

end

title('Variance Since Start of Signal');

xlabel('1/10,000''s of a second');

ylabel('Variance');

for i = 0:len-10

 for k = 10:len

 newy = wkeep(y, k, 1);

 vnew = var(newy);

 plot(k, vnew, 'b-')

 end

end

M = 80; %(days) frame ~8 second audio, 82425 samples, 10 samples = 1 millisecond

N = 240; %(days) window

startPt = (N)/ 2; %picks a start point by moving over half the size of a window, avoids zero stuffing the ends

newLength = 1+round(82425/M);

shortSig = zeros(1, newLength); %creates an empty array to hold the variances(1, 1027)

 for len = 1:newLength %(120, 1027)

 middle = M\*(len-1)+M/2; %centers frame

 rt = N/2+middle;

 rt = round(rt);

 lft = rt-N;

 lft = round(lft);

 if((lft<0)||(rt>82425)) %%avoids overflowing the new array of variances

 shortSig = zeros(1, N); %%creates placeholder array of zeros

 end

 for x = 1:N

 sbscpt = lft+(x-1);

 if ((sbscpt>0)&&(sbscpt<=82425))

 shortSig(x) = y(sbscpt);

 end

 end

 windowVar = var(shortSig); %finds the variance value for the window/frame iteration

 for x = 1:M

 sbscpt = middle+(x-1)-(M/2);

 if((sbscpt>0)&&(sbscpt<=82425))

 vars(sbscpt) =windowVar; %%puts variance values into array

 end

 end

 end

plot(vars, 'g-') %plots the new mean values after the window/ frame analysis

xlabel('Length of Audio Signal');

ylabel('Variance (Frame = 10ms, Window = 30ms)');

title('Window/ Frame Variance Analysis Audio Sample');

# Conclusions

For data sets like the audio signal, we can expect the variance over the whole signal to be extremely low (negligible) since it oscillates relatively evenly around 0. This means we need a window/ frame analysis to complement what frequencies we want to look for in the signal. Variance for data sets like the Google stock can be calculated over the whole signal to see how much it has increased, but that gives little more than does looking at the actual signal. A window/ frame analysis will indicate what the behavior has been like over shorter periods of time, which will normally be more useful to see how the stock has been performing lately. One other thing to be noted is that when computing the variance continually and continuously increasing the number of values considered for the variance calculation, a LOT of memory will be sucked up. Figure 7 below shows the state of my memory usage just after canceling one of those calculations.



Figure 7