[Robert Irwin]

ECE 3522: Stochastic Processes in Signals and Systems

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

# Problem Statement

In this assignment, we are analyzing the Google stock prices as well as an audio signal sampled at 8kHz. With each of these signals, we are to compute the variance of the entire data set and plot it as a horizontal line. Then, we are to compute the variance of the first 10 samples, followed by the first 11 samples, continuing the process until we compute the variance for the entire data set. This curve is to be plotted on the same plot as the variance of the entire set. Then we are to use the frame and window method to compute the variance for each data set. For the stock prices we were to use a frame size of 1 day and a window size of 30 days. For the audio signal we were to use a frame of 10 msec and a window of 30 msec.

# Approach and Results

To calculate the variance of the entire set, the Matlab macro var() was used. To calculate the variance using the extending window method, we needed to write the first n samples of the data into a temporary variable that we would use to compute the variance. This variance was stored in a vector, and the index of this vector was increased after each computation of the variance. For the frame and window method we simply defined the frame and window and used the same code from the previous class assignments. To achieve a frame of 10 msec for the audio signal we had to figure out how many samples were in 10 msec. To do this, we multiplied the desired time by the sampling frequency of 8000samples/sec. This gave us a frame of 80 samples and a window of 240 samples. The two resulting plots are shown below.



*Figure 1: Shows the Variance of the Stock Data*

The plot above shows the analysis of the Google Stock data. The dotted line shows the variance of the entire set, the magenta curve shows the variance calculated using the extending method, and the black curve shows the variance calculated using the frame and window method. It is important to compare this plot to the original data signal shown below.



*Figure 2: Google Stock Prices*

First we will analyze the variance curve from the extending window method. The first few calculations of the variance make sense as all the stock prices for that window are similar, and no values are too far from the mean of the window. As the window is extended the stock prices vary more, which results in the rise in the variance curve. When we are in the range of day 1500 – 2000 we notice that data oscillates around the same value. This is why the variance in this region flat-lines. The final sanity check is the fact that the final variance calculation, where we compute the variance with the entire data set is the same converges to the constant value of the variance computed earlier.

For the frame and window method, it makes sense that the variance is always much lower. With a window of only 30 days, all of the data points in the window are close to the mean of that data set. This is because of the minimal change in the price of stock from day to day.

Now we will analyze the audio signal in the same way.



*Figure 3: Shows the Variance of the Audio Signal*

The plot above shows the analysis of the audio signal. The dotted line shows the variance of the entire set, the magenta curve shows the variance calculated using the extending method, and the black curve shows the variance calculated using the frame and window method. It is important to compare this plot to the original data signal shown below.

**

*Figure 4: Audio Signal*

We notice that the variance using the extending window method is always close to the variance of the entire signal. This makes sense because the audio signal oscillates around 0 because it is a sum of sinusoids. We also notice that every time there are spikes in the audio signal, the variance plot also spikes. This can be seen when we compare the variance of the frame and window method to the extending window method. We also notice that our extending window variance converges to the variance of the entire signal.

It makes sense that the variance using the frame and window method is very large. Because the audio signal changes so rapidly, its variance is very high. This can be seen at around 4 seconds. This is the highest frequency content of the signal, and the variance plot corresponds with this.

# MATLAB Code



*Figure 5: Bring in the Stock Prices and Compute the Variance*

In this portion of code we are bringing in the stock data. We are also computing the variance of the data set. Once we have the variance, we create a vector with this variance for each point in the vector ‘days’. We have to do this in order to be able to plot the constant value of the variance.



*Figure 6: Computing the Variance (Extending Window Method)*

In this piece of code we are computing the variance using the extending window method. We must first create a vector with integer values that extends the length of the signal. Then, with an index starting at 10, we assign the first 10 values to a temporary vector, and compute the variance of those ten data points. We then do this for every value of N until N+10 is greater than the number of data points in the signal.



*Figure 7: Computng the Variance and Plotting*

In this portion of code we are computing the variance using the frame and window approach. We are simply calling a function that computes the variance incrementally. That function is described below.

****

*Figure 8: Variance Function: Defining Local Variables*

This is the function compute\_var. The arguments sig\_a, fdur\_a, and wdur\_a correspond to the data we are working with, a frame size, and a window size respectively. Then we define local variables. Sig\_wbuf is a matrix that will hold temporary values of the mean. Var\_full is the variable that will store the entirety of the mean calculations.

****

*Figure 9: Variance Function: Defining Left Right and Center Pointers*

In this portion of the code we are defining the pointers. As stated in the comment, the center pointer tells us where our frame is located, and the left and right pointers correspond to the reach of our window. These pointers had to be rounded because they will be used as indexes.

****

*Figure 10: Variance Function: Calculating the Variance of a Specific Window*

Because of the way the pointers are defined, we need to ensure that each window has the same amount of data. The uppermost if statement in this portion of code takes care of this. This if statement is executed if either the left or the right pointer is outside the data set. If either pointer is outside the data set, the vector, sig\_wbuf is zero stuffed. Then we are storing each value of the sampled signal in that window and assigning it to the temporary value sig\_wbuf. After we have all of the data in that window, we compute the variance.

****

*Figure 11: Variance Function: Assign the Variance of One Window to the Output Vector*

Here we are assigning the variance value of one window to the output of the function. It is important to note that we are assigning the same value for the variance to the output variable once for each unit length of the frame. This explains why larger frame sizes are much blockier than smaller frame sizes.

We now look at the code used to analyze the audio signal.



*Figure 12: Bring in the Audio Signal*

For simplicity the same code was used to analyze the audio signal, we just assigned the audio signal to the variable stock\_close. Taking the desired duration and multiplying by the sampling frequency calculated the frame and window sizes. It is also important to note that the time vector had to be normalized by the sampling frequency.

# Conclusions

In this we saw what computing variance in real time would look like, and why it is so difficult to keep things like volume on the TV constant. We can see that the variance of the audio signal is constantly changing. Because variance is essentially the energy of the signal, we can see that keeping a constant volume of the signal is a challenging task. We can also look at the variance of the stock prices and determine when a good time to buy and sell would be. For instance, a good time to buy Google stock would be around day 1500. A good time to sell would be around day 2500. However, this is hard to predict in real time.