Computer Assignment (CA) No. 3: Variance

Tyler Berezowsky

February 2, 2015

1 Problem Statement

The purpose of this assignment was to explore methods to calculate variance in real time, and create a distinction between those methods. Comparison was also made to the global variance. This was accomplished through MATLAB, the previously used data sets and the following tasks:

1. Estimate the variance (second central moment) from the entire data set. Plot this as a horizontal dotted line.
2. Starting with the first 10 samples, estimate the variance from the first N samples of the signal, letting N vary from 0 to a maximum of the number of samples in the file. Overlay a plot of this on the plot from (1). Describe what you observe.
3. Now estimate the variance using a frame/window approach. For the Google stock price, set the frame to 1 day and the window to 30 days. For the speech signal, set the frame duration to 10 msec and the window duration to 30 msec. Overlay this plot on the above plot and describe what you observe. Which approach makes sense?

2 Approach and Results

The data sets were loaded via code from previous computer assignments. For task one, the global variance was determined through the `var()` command. For task two, the rolling calculation was performed through a `for loop` which selected a subset of data always starting at the origin. The subset was increased by 10 samples per iteration until the subset was the length of the original set. At each iteration, the variance of the subset was calculated (via `var()`) developing a rolling calculation of variance for the data set. For task two, frame and windowing was accomplished through code from CA: 01. Variance was calculated for the audio signal with a frame size of 10 msec and a window size of 30msec. Variance of the google data set was calculated with a frame size of 1 day and a window size of 30 days.

For each respective data set, the global, rolling, and framed variance was plotted together. The plots can be seen in figures 2.1 and 2.2. The trace legend for both plots is identical. The blue trace represents the rolling variance, the red trace represents the framed variance, and the orange trace represents the global variance.

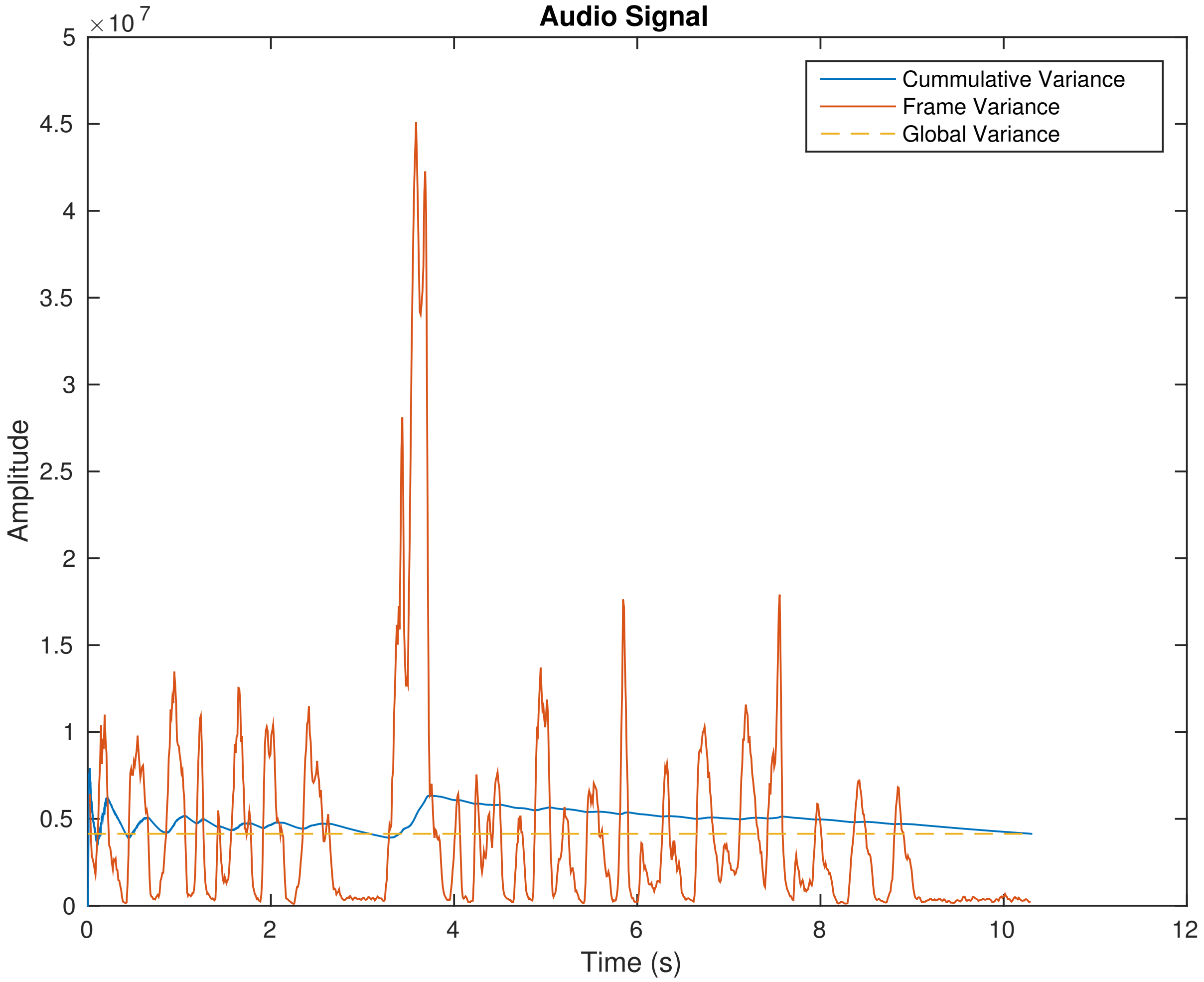

Figure 2.1: Plot of global, rolling, and framed and windowed variance overlaid for the audio signal.

Interesting to note for the audio plot, is the self leveling behavior of the rolling variance with the global variance. When the subset is small, at the beginning of the signal, ripples of a diminishing nature can be observed. This of course correlates to the spikes of variance or energy (due to a near zero mean) present in the audio signal. The energy spikes, relatively constant in amplitude and period, between 0 and 3 seconds begin to level the rolling variance as the carnality of the sample set increases until the large energy spike between 3 and 4 seconds upsets the rolling variance. As the carnality of the sample set increases again with energy spikes of a more uniform amplitude, but less than then those near the origin of the signal, the rolling variance begins to diminish and level again with the global mean between the 4 and 10 second mark.

The rolling calculation is self leveling with the global variance because it is constantly approaching it with each increase of the subset. As the subset increases, the spikes in variation also become more illusive. This behavior can be illustrated through the variance calculation:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{2.1}$$

As the set increases, the large number of relatively constant sums and increasing denominator N, level any variation in variance. This effect is directly correlated to the size of the set. Therefore, a rolling variance calculation would not be effective at determining local variation in signal amplitude. Frame and windowing is a more suitable technique as it permits sections of the signal to be examined which then can be compared to the global variance or rolling variance for real time applications.

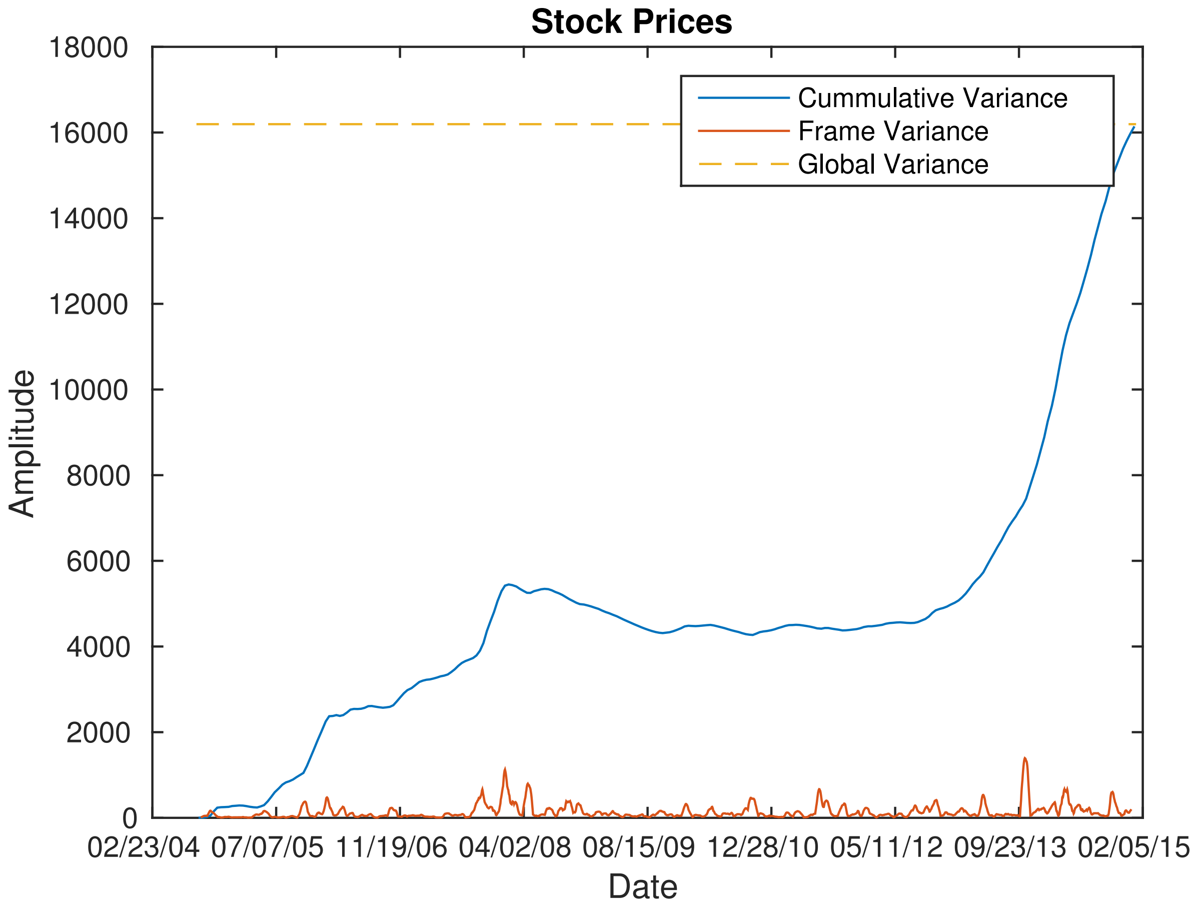

Figure 2.2: Plot of global, rolling, and framed and windowed variance overlaid for the Google stock.

The Google stock plot illustrates an increasing rolling variance which ends at the global variance, with minimal deviation in variance per frame compared to the audio signal. This is distinct in comparison to the audio signal's rolling variance which tracked the global variance. This behavior is due to the completely random nature of the samples and positive non-irregular trend in the stock data set.

In layman's terms, the samples can be seen to be regularly increasing, and therefore generally never identical. In perspective of the variance calculation, a data set with a positive trend will be constantly increasing in variation as each sample is different from the last. This is also true for a signal with a negative trend. A flat lined signal would produce no variation as the mean would always cancel the numerator. And as with the audio signal, a signal with samples that illustrate a high degree of variance in a predicable manner, will illustrate a uniform amount of variance.

3 MATLAB CODE

Listing 1: MATLAB solution for CA: 03

```
%% Computer Assignment 03
% 1) Estimate the variance (second central moment) from the entire
% data set. Plot this as a horizontal dotted line.
% 2) Starting with the first 10 samples, estimate the variance from the
% first N samples of the signal, letting N vary from 0 to a maximum of
% the number of samples in the file. Overlay a plot of this on the plot
% from (1). Describe what you observe.
% 3) Now estimate the variance using a frame/window approach. For the
% Google stock price, set the frame to 1 day and the window to 30 days.
% For the speech signal, set the frame duration to 10Âămsec and the window
% duration to 30Âămsec. Overlay this plot on the above plot

%% Import the Audio and Google datasets
% Import the excel spreadsheet into an array

clear; clc; clear all;

filename = 'google_v00.xlsx'
[data, header, raw] = xlsread(filename);

high = data(:,3);
low = data(:,4);
close = data(:,5);
open = data(:,2);
dates = x2mdate(data(:,1), 1);

% Import the .raw file into an array

filename = 'rec_01_speech.raw';
file = fopen(filename, 'r');
audio = fread(file, inf,'short');

%audio = audio(1:10000); % cut down audio!

sampleRate = 8e3
time = (0:length(audio)-1)*1/(sampleRate);

%% Calculate Variance

aVar = var(audio);
gVar = var(close);
aVar = linspace(aVar, aVar, length(audio));
gVar = linspace(gVar, gVar, length(close));

%% Generate Vector for Change in Variance
% increment by 10 samples

% audio signal
aDeltaVar = [];  % change in variance of entire signal per 10 samples
aDeltaTime = [];
for i= 10:10:length(audio) % increment through signal 10 samples at time
    aDeltaTime = [aDeltaTime; time(i)];
    aDeltaVar = [aDeltaVar; var(audio(1:i))];
end

% google stock
gDeltaVar = []; % change in variance...again
gDeltaTime = [];
for i= 10:10:length(close) % increment through signal 10 samples at time
    gDeltaVar = [gDeltaVar; var(close(1:i))];
    gDeltaTime = [gDeltaTime; dates(i)];
end

%% Frame and Window Data and Generate Vector of Window Variance

% audio signal M = 10msec N = 30ms
M = 10e-3*sampleRate;
N = 30e-3*sampleRate;

sigLength = length(audio);
```

```
frameSize = M;
windowSize = N;

% initialize arrays for windows or dates
windows = [];
windowTimes = [];

for z = 1:frameSize:sigLength
    % calculate the frame center, and then the right and left window indexes
    frameCenter = floor( z + frameSize/2 ) ;
    windowLeft = floor( (frameCenter - 1) - 0.5*windowSize );
    windowRight = windowLeft + windowSize - 1;

    % insure the window never exceeds signal
    if (windowLeft >= 1) && (windowRight <= sigLength)
        windows = [windows; audio(windowLeft : windowRight)'];
        windowTimes = [windowTimes; time(windowLeft : windowRight)];
    end
end

audioFwVars = var(windows, 0, 2);
audioFwTime = mean(windowTimes, 2);

% google stock
sigLength = length(data);

frameSize = 1;
windowSize = 30;

% initialize arrays for windows or dates
windows = []
windowDates = []

for z = 1:frameSize:sigLength
    % calculate the frame center, and then the right and left window indexes
    frameCenter = floor( z + frameSize/2 ) ;
    windowLeft = floor( (frameCenter - 1) - 0.5*windowSize );
    windowRight = windowLeft + windowSize - 1;

    % insure the window never exceeds signal
    if (windowLeft >= 1) && (windowRight <= sigLength)
        windows = [windows; data(windowLeft : windowRight, 5)'];
        windowDates = [windowDates; dates(windowLeft: windowRight)'];
    end
end

googleFwDates = mean(windowDates, 2);
googleFwVar = var(windows, 0, 2);

%% Plot Everything!

% audio signal
figure()
plot(aDeltaTime, aDeltaVar, audioFwTime, audioFwVars, time, aVar, '--')
legend('Cummulative Variance', 'Frame Variance', 'Global Variance')
xlabel('Time (s)')
ylabel('Amplitude')
title('Audio Signal')
print -depsc plot_01_1

% google signal
figure()
```

```
plot(gDeltaTime, gDeltaVar, googleFwDates, googleFwVar, dates, gVar, '--')
datetick('x',2 ,'keeplimits', 'keepticks');
legend('Cummulative Variance', 'Frame Variance', 'Global Variance')
xlabel('Date')
ylabel('Amplitude')
title('Stock Prices')
print -depsc plot_02_1
```

4 Conclusions

This assignment illustrated two methods of calculating variance in "real-time": rolling and framed. Comparison was also made to global variance for each method.

The following conclusions were made. First, rolling variance calculations are not effective for determining local changes in variance. As the size of a sample set increases, small bursts of variance will become washed out due to averaging effect of the large set. Frame and windowing the data set, and then calculating the variance of the window is a more sensitive approach. Secondly, a digital audio signal contains a finite amount of possible values. Therefore, it is possible to predict the probability of the next sample which is ironic (to me atleast) as it contains no discernible trend besides averaging to zero. Last, and even weirder, while the Google stock data contains a trend, it is completely unpredictable through a classic probability spaces (As far as I currently know!?) as the Ω set increases with each sample. This is illustrated in the accumulating rolling variance due to the positive trend in the Google stock price present in figure 2.2.