BIOE 198MI Biomedical Data Analysis. Spring Semester 2018.
Lab 3: Moving Average Filter

Assume we have a data set contains N data points, called $x_1, x_2 \dots x_N$. We can use the following equation to calculate the mean of the data set :

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

The mean is used to measure the center of a numerical data set. Sometimes, we want to look at the the trend of the data to get a basic idea of the data, and one single mean value can't help us. Therefore, the moving average filter is introduced.

The moving average filter is commonly used to smooth out the short-term data fluctuation (noise) and show the long shape pattern of the data. It is calculated by creating a series of averages of data points with specific window size. Sometimes it's call moving mean.

A. Understand the meaning of '*moving*' (signal shift)

We first generate a signal called $g[n] = 1$, with $n = 1{,}2$. Then let's shift it the the left by one point.

```
%% Visualize Signal Shift
g = [1 1];
ng = [1:length(g)];
Shift_direction = 'Left'; % shifting direction
Shift_N = 1; % # of shifting points
Figflag = 1; % plot figure yes--1, no—0

DiscreteShift(g,ng,Shift_direction,Shift_N,Figflag); % self-defined
function.
```

The shape of signal g is unchanged. The shifted signal can be write as $g[n+1]$

Note `DiscreteShift` is a self-defined function. What inputs should we use to generate $g[n-2]$?

B. Moving Average

Calculate the averages by hand

Here we have a discrete signal $f[n] = n$, where n = 1,2,3,4

We can calculate the average value of every two adjacent points by hand.

$$y[1] = \frac{1}{2}(f[0] + f[1]) = \frac{1}{2}(f[0]\times 1 + f[1]\times 1) = \frac{1}{2}(f[0]\times g[1] + f[1]\times g[1]) = 0.5$$

What is $f[0]$ here?

g[1]	**g[2]**			
f[0]	**f[1]**	f[2]	f[3]	f[4]
	y[1]			

$$y[2] = \frac{1}{2}(f[1] + f[2]) = \frac{1}{2}(f[1] \times 1 + f[2] \times 1) = \frac{1}{2}(f[1] \times g[0] + f[2] \times g[2]) = 1.5$$

	g[1]	**g[2]**		
f[0]	**f[1]**	**f[2]**	f[3]	f[4]
	y[1]	**y[2]**		

$$y[3] = \frac{1}{2}(f[2] + f[3]) = \frac{1}{2}(f[2] \times 1 + f[3] \times 1) = \frac{1}{2}(f[2] \times g[1] + f[3] \times g[2]) = 2.5$$

		g[1]	**g[2]**	
f[0]	f[1]	**f[2]**	**f[3]**	f[4]
	y[1]	y[2]	**y[3]**	

$$y[4] = \frac{1}{2}(f[3] + f[4]) = \frac{1}{2}(f[3] \times 1 + f[4] \times 1) = \frac{1}{2}(f[3] \times g[1] + f[4] \times g[2]) = 3.5$$

Thus, we can think the averaged $y[n]$ be a signal generated with the $g[n]$ shifted over the data $f[n]$, the $y[n]$ is called a moving average.

			g[1]	**g[2]**
f[0]	f[1]	f[2]	**f[3]**	**f[4]**
	y[1]	y[2]	y[3]	**y[4]**

The averaged signal can be expressed as:

$$f_a[n] = y[n] = \frac{1}{N}\sum_{m=n-N+1}^{n} f[m]g[m-n+N]$$

Calculate moving average using MATLAB

`%% Moving Average Filter`

Answer the following questions:

Line 38: `f0 = [zeros(1,N-1),f];`

Why did we do (N-1) points of zero padding?

Line 41: `fs = f0(n:n+N-1);`

Why don't we use `f(n-N+1:n)` as equation $f_a[n] = \frac{1}{N}\sum_{m=n-N+1}^{n} f[m]g[m-n+N]$ shows?

The key code of Line 34~60 (without all the plotting commands) are saved as a self defined function: `MovAvg(sig, window)`. Note that `sig` and `window` are general description of parameters `f` and `g`, respectively.

C. Moving average example

First, load the file `'EcgSig.csv'`, which contains one line of header and two columns of data (separated by comma).

```
fid = fopen(['EcgSig.csv']); % open a file
FileHeader = textscan(fid,'%s %s',1,'Delimiter',','); % read the file
header
% display the file header information
for i = 1:length(FileHeader)
    disp(['{' num2str(i) '} ' FileHeader{i}{1}]);
end

Data = textscan(fid,'%f %f','Delimiter',',','headerLines',1); % read the
data from file
fclose(fid); % close file

Ecg_sig = Data{1}';
Time = Data{2};
```

Second, let's play with different window size and see how that affects the averaged result. Four windows are generated with window size of 5, 10, 20 and 100 points.

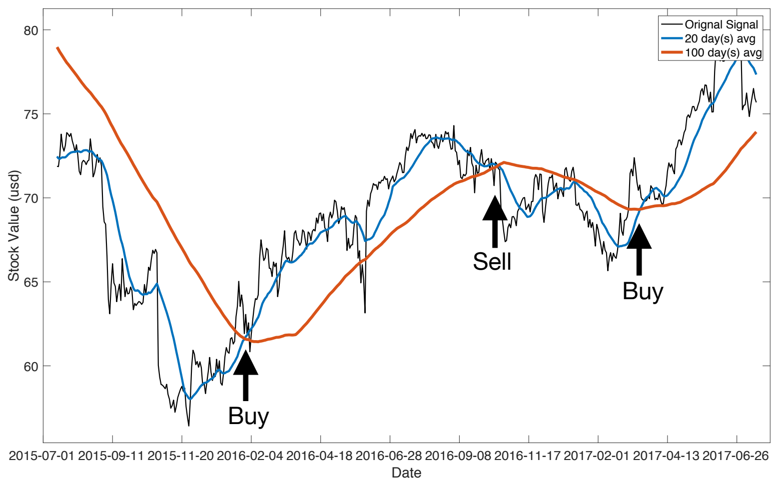
```
% define averaging window size
N = [5 10 20 100];

% define the moving averaging filter with different window size
for i = 1:length(N)
    F{i}= ones(1,N(i));
end
```

In lines 93~ 98: `Use` the defined function `'MovAvg'` to calculate the moving average with different window size. Compare the the plots and describe the difference.

Assignment

The simple moving average filter is commonly used to identify the trend direction of financial data, like stock price. The SMA is very simple but useful. One trading strategy is using the crossover point of long term and short term moving averages.

A short term MA will react faster to price changes than a long term MA. So when the short term MA crosses above the longer term MA, it's a buy signal as it indicates the trend is shifting up.This is known as a 'golden cross'. On the other hand, when the shorter MA crosses below the longer term MA, it's a sell signal as it indicates the trend is shifting down. This is known as a 'dead/death cross'.



You are given a .csv file with 3 years of historical stock data of one company. Assume you are given $5,000.00 as initial funding. How can you maximize your profit (or minimize the loss) using the crossover strategy of SMAs during 2 years (2015-07-01 to 2017-06-30)? Including first and last day.

To simplify the problem, here're some rules:

- Only use short term window of n = 5, 10, 20 (days).
- Only use long term window of n = 50, 100, 200 (days).
- Use the 'close' value as everyday stock price.
- Trading (sell/buy) cost is $5.00.
- When make purchases at golden cross, buy as much stock shares as possible.
- At dead cross, sell all the stock shares at once.
- On first day (2015-07-01), if short term MA is greater than long term MA, buy stocks.
- On last day (2017-06-30), sell all the left over stock shares.

Justify your answers with plots. Explain your choice long term and short term window size.

Suggested approach for Stock problem:

<u>PART 1: load data and compute SMA, LMA</u>

1. Load the csv file into Matlab
   a. Invest the csv file
      i. how many columns of data
      ii. how many lines of header
      iii. determine which columns of data do we need
   b. load the file header (hint: very similar to how the ecg data was loaded)
   c. load the rest of data
   d. Keep the data in ‘date’ and ‘close’ as stock date and stock value, respectively
      The date is sorted in descend order, use function ‘`flipud`’ to make it ascend.

2. Start with short time window size of 20 and long time window size 100
   a. generate the cell F, with different window sizes
   b. compute the moving average with different window sizes (Remember to save the averaged output)

3. Plot the original stock data, SMA, LMA on the same plot
   a. find the indices of starting and ending dates (hint: use the function ‘`strcmpi`’)
   b. plot the original stock value based on the indices
   c. plot the SMA and LMA on the same plot

<u>PART 2: Calculate the profit/loss based on crosses of MAs</u>

1. Create new variables (date duration, stock values, LMA, SMA) based on starting and ending dates (index)
2. Set up the initial fund and trade cost.
3. Calculate the profit/loss using function ‘`CalculateStockProfit`’ with all previously defined **6** inputs.

<u>PART 3: Vary the window sizes of short term and long term window.</u>