Homework 2, CSCE 240, Fall 2014

Overview

One of the classic problems in mining data is to determine whether or not the data is "normal". This often happens with data we are getting in real time, and what we want to be able to do is predict the future.

For example,

- If our network suddenly has a surge in incoming packets, perhaps it means that we have come under attack. It could also mean that we in the college just got an email sent to "all mail users" from the administration.
- If we have a stock that is rising in value, it could be at the peak of its value, and it's time to sell. Or it could rise even further, in which case by selling now we don't make as much money as we might. (Analogous arguments apply to falling values.)
- I am sure that many of us play the same game at the gas pump. When I pass the usually-best-price gas station on the way home, I notice the price. If that price seems lower today than it has been for a few days, and I need gas, then I buy gas. If it seems higher, and my tank isn't empty, then I hold off in hopes that it will go lower.
- If you have been charging a few hundred dollars a month on your credit card, and suddenly there are huge charges in foreign countries, it could mean your card was stolen. Or it could mean you have just gone on your dream vacation.

In all these instances, we try to use data from the past to determine a norm, and then we measure "today" against that norm in order to decide whether to ring a bell and take an action.

This Homework

This exercise is a simple version of a moving average for stock market buying and selling. Let's assume we have data for the price of a stock for the last 30 days. We can compute the average price of the stock for these days subscripted 0 through 29. Let's choose a "decision metric" of 5 percent. Now

if the price of the stock on day 30 is more than 5 percent higher than the average of the last 30 days, then we assume that the stock is doing well, and we sell shares. If the price is more than 5 percent lower than the average, we buy.

Then tomorrow we have data for 31 days. We recompute the average using the data from days subscripted 1 through 30, and continue.

We thus compute a moving average using the data in a window of fixed size, and make decisions based on whether the current price is "high" or "low" using our value of the decision metric as the definition of high or low.

Basically, this is a standard technique. Use past data to compute a "normal" value. Compare current data against the normal value, and if it's too far on the "abnormal" side, then do something.

For this assignment, you have 100 data items that represent the price of a stock. Read that data into a `vector` of `double` values. Then run a double loop to compute the moving averages of a window of width `width` from starting subscripts 0 up to the end. (If the ending subscript is beyong the length of the `vector` then quit. That is, if you have 5 data items, and the window width is 3, the last starting subscript would be 2, and you would do averages for 0-1-2, 1-2-3, 2-3-4.)

If the LAST VALUE IN YOUR WINDOW is more than `decisionMetric` away from the moving average, then issue a BUY or a SELL order as appropriate.

(Technically, you would look at the window values and use the NEXT value to dermine what to do, but that would require being more clever with subscripts, so don't worry about that. I am sure you could work that out, but it wouldn't teach you much because it would all be just detail.)

An Important Detail

I have given you a dot h file for the class to compute the moving average. Note that the function that does the average takes both the window size and the decision metric as parameters. These are hard-coded in the calls to the function from the main program.

I WILL USE AN IDENTICAL MAIN PROGRAM TO THIS EXAMPLE WHEN I RUN YOUR CODE FOR TESTING PURPOSES, EXCEPT THAT I WILL CHANGE THE VALUES OF THE WINDOW AND THE METRIC.

You should therefore not change the main program in any substantive way except to try different window and metric values.

The Real Computation Is Harder Than This Homework

In reality, you would not have all the data in one array all at once. You would get the data one day at a time, and you could throw away data earlier than your window. This could also be done, but it would also just be detail programming, so it is not required that you do this. Write this program more as an analysis of what you COULD have done rather than what you would do in real time.