Assignment Instructions

This assignment is *individual*. As given in Bishop, on page 31 (Section 1.2.6), using your favorite *programming* language (not Matlab!), implement the formula (1.69) for Bayesian curve fitting. [For this you will also need to implement the formulas (1.70) - (1.72).] As suggested in Bishop, page 31 (1st paragraph), assume that the parameters *alpha* and *beta* are fixed and known in advance. Also assume some reasonable values for the parameters that need to be specified, e.g., the number of input values *N*.

Excerpt from Bishop, pages 21-32 can be found in resources (will be added shortly).

The project involves some matrix operations (e.g., matrix inversion), and you may wish to look for existing libraries for numerical computation in the language that you're using. For example, for Java you can find a great deal of information here: Java Numerics: information on numerical computing in Java. There are many libraries listed there, and it appears that JAMA, Jampack, and many others provide all that you need for this project.

You may also consider calling Matlab from your program to perform numerical computations. Check below for more information.

Once you program the formula for curve fitting, test it using your own data set. Ideally, you could use a sequence of actual stock prices that your data collection module is already collecting (Project Phase - 1 of data collection module will be posted next week). Given a sequence of length *N*, run your program and compare the predicted price (at time *N*+1) with the actual price retrieved from the stock market. Repeat this 10 times for different input datasets. For performance evaluation, calculate both the absolute mean error and average relative error as shown in the lecture notes on page 52.

Submit the following:

1. Source code of the program
2. Short instructions on how to compile and run your program, including the allowed range of input parameters (the so called `README.txt` document)
3. An example set of test data to test your program—preferably a comma separated values (CSV) file
4. Brief summary (PDF document) of the performance evaluation (prediction errors)

Connecting Matlab and Programming Languages

You may wish to call Matlab from your program to perform numerical computations. To do it from Java, consider MATLAB Builder Java language, which lets you integrate MATLAB applications into your organizations's Java programs by creating MATLAB based Java classes ...
Check also Calling Matlab from Java or Connecting Java and Matlab.

To call Matlab from C, check Calling MATLAB Software from C and Fortran Programs
Also do Google search.

Going back to Java, once you get the C version running, you can write a small Java Native Interface (JNI) from Java to call the same from Java.

Also you can search for direct code for caling Matlab from Java, which uses a client/server mechanism. The advantage of using JNI over the client/server mechanism would be the speed, because the C engine uses pipes as compared to TCP/IP sockets.