Project 1 Description

Project 1: Curve Fitting / Linear Regression

Due on Friday, January 26, 2018 (Midnight)

An Overview

The curve fitting problem motivates a number of important key concepts covered in the book. It will recur in the later chapters. This project asks you to solve the linear regression problem by two different approaches: 1) direct error minimization and 2) Bayesian approach. After complete your program, you will write a report and compare the two different methods. The purpose of this project is to help you grasp the Bayesian modeling framework -- a building block of the course and later projects -- and get you familiar with Matlab.

Background Information

Bishop 1.1, 1,2

Your Implementation

You will generate noisy observations **(x, t)** (training data points), assuming Gaussian noise. Estimate the regression coefficients **w** by minimizing the sum-of-squares error. Re-formulate the problem in a Bayesian approach by introducing a prior distribution **p(w|α)** over the coefficients. Solve the Bayesian linear regression problem. Your code must be reasonably commented and written in an understandable manner--we will read your code. You cannot use matlab functions such as polyfit, you must use the equation you have derived.

The shaded error bars function from **www.mathworks.com/matlabcentral/fileexchange/26311-shadederrorbar** **(http://www.mathworks.com/matlabcentral/fileexchange/26311-shadederrorbar)** has been provided for you to visualize the predictive probabilities as well as export_fig from **https://github.com/altmany/export_fig** **(https://github.com/altmany/export_fig)** to generate decent looking figures.

Starter Code

You are given starter code which contains the following:

- Code to generate noisy sample points
- Code to save and load data
- Code to plot points and curves

Download the starter code in Matlab from [here (https://psu.instructure.com/courses/1930166/files/89758793/download?wrap=1)](https://psu.instructure.com/courses/1930166/files/89758793/download?wrap=1). Run generateData.m will create a file named data.mat in your current working directory. Then run curveFit.m. A window should appear with a plot looks like example_data_plot.png (included with the starter code). Implement your **OWN** code in curveFit.m (do NOT use the polyfit function / you must write the linear algebra yourself).

Grading Criteria

Your report must include the estimated the regression models for 10 sample points by

- error minimization (refer to Equation 1.2, page 5) - 20 pts
- error minimization with the regularization term (refer to Equation 1.4, page 10) - 20pts (You can generate plots similar to Figure 1.7, page 10).
- the ML (maximal likelihood) estimator of the Bayesian approach (refer to Equation 1.62, page 29) - 20 pts
- the MAP (maximum a posteriori) estimator of the Bayesian approach (refer to Equation 1.67, page 30 and Equation 3.55, page 153) - 20 pts
- write-up the report to summarize, compare and contrast the results - 20 pts

Use plots to visualize your results. For example, plots like Figure 1.3 (page 6) and Figure 3.8 (page 157) make it very clear to see how good your model fits the data.

Grading is mainly based upon:

- your implementation -- whether you know how to solve the regression problem by different methods.
- your understanding -- whether you understand different properties of each method. Make sure to show the derived equations (and the deriving of the equations) for each method.
- your presentation of results -- whether they are logically and clearly presented in the report, especially through visualization and tabulation. Make sure every figure and table has a descriptive caption.

Make sure to properly cite ALL resources you used for this project!

Submission

Please submit your code *along with your* data files that you used to estimate each of the regression model, and your written report in a zip file.

Create a readme.txt file to explain which data file is used to generate which model. Name the zip file as yourFirstname_yourLastName_projectNo.zip, for example, Weina_Ge_1.zip.

Tips

- Familiarize yourself with Matlab by reviewing the online tutorials and Matlab help documents.
- Don't try to do this at the last minute. This assignment is supposed to be fun and relatively easy, but time pressure has a way of ruining that notion.

- Use the backslash operator "\" rather than the inverse function to solve the linear system of equations. **https://www.mathworks.com/help/matlab/ref/mldivide.html** (https://www.mathworks.com/help/matlab/ref/mldivide.html)

Extras

You may choose to include any the following for in your report for extra credits.

- add to the plot of errors for the point ln **λ = -18, -15 and -13 (Figure 1.8)** - 5pts
- For a fixed number of sample point (50 points), vary the order of polynomial M (M = 0,1,3,6,9). Generate a table similar to Table 1.1 (page 8). - 5pts
- For a fixed degree of polynomial (M=9), vary the number of sample points N. Generate a plot similar to Figure 1.6 (page 9). - 5pts