Project #1 Description

Linear Regression + Classification

CSE583/EE552 Pattern Recognition and Machine Learning, Spring 2022

Release Date: Monday, January 17, 2022
Submission Due Date: Monday, January 31, 2022

Contents

1 Part 1: Linear Regression **2**
- 1.1 Introduction . . . 2
- 1.2 Requirements . . . 2
- 1.3 Extra credits . . . 2

2 Part 2: Classification **3**
- 2.1 Introduction . . . 3
- 2.2 Requirements . . . 4
- 2.3 Extra credits . . . 4

3 Submission **4**

4 Grading Criteria **5**

5 Common Issues **6**

1 Part 1: Linear Regression

1.1 Introduction

The first part of this project is about linear regression, which motivates a number of important key concepts covered in the book. You will generate noisy observations $(\mathbf{x}, \mathbf{t})$ (training data points), assuming Gaussian noise. Then you will need to introduce a prior distribution $\mathbf{p}(\mathbf{w}|\alpha)$ over the coefficients and solve the Bayesian linear regression problem by two different approaches:

1. The ML (maximal likelihood) estimator of the Bayesian approach (refer to Equation 1.62, page 29).

2. The MAP (maximum a posteriori) estimator of the Bayesian approach (refer to Equation 1.67, page 30 and Equation 3.55, page 153, you can use $\beta = 11.1$ and $\alpha = 0.005$ as shown in textbook)

After completing your program, you need to report a summary and comparison of these two methods. The purpose of this part 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.

Download the starter code in Matlab from CANVAS. Please check the **ReadMe.md** for more details about the starter code.

1.2 Requirements

Your report for Part 1 MUST include:

1. The derived equations of two Bayesian approaches

2. Visualization results of estimated regression models for $N = 50$ sample points. You should generate plots like Figure 1.3 (page 6) and Figure 3.8 (page 157) to make it very clear to see how good your model fits the data. *The plotting function is given in the starter code.*

3. A summary and comparison of these two Bayasian methods.

1.3 Extra credits

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

- Add to the plot of errors for the point $\ln \lambda = -18, -15$ and -13 (Figure 1.8), and you are welcome to use more lambda values. – 5pts.

- For a fixed number of sample point (e.g. $N = 50$), 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.
- Use your results to prove or disprove the curse of dimensionality (basically whether M and N have an exponential relationship) – 5pts.

2 Part 2: Classification

2.1 Introduction

The second part of this project is about classification on real data. You will learn how to do Fisher's linear discriminant on your data, and evaluate the performance on real data.

You will be working on two different datasets (both datasets are included in the starter code).

1. Wallpaper Group Dataset - This dataset consists of the features extracted from images containing the 17 Wallpaper Group. These features have already been extracted for you. Here is an example of multiple patterns from one such group (specifically P4M):
2. Taiji Pose Dataset - This is a dataset of the joint angles (in quaternions) of 35 sequences from 4 people performing Taiji in our motion capture lab. You will classify which MoCAP frames are transitional frames 7 different poses (non '0' labels) and the non-transitional frames (the '0' labels). Here is a sample video of one of the performances: link to the video (We are only using up to 1:30 in the video)

Your implementation should be as follows:

1. You will need to first implement a function to find the Fisher projection using the training features and labels (Bishop 4.1.4 and 4.1.6). You must implement the Fisher projection by yourself.
2. Then you will need to train a classifier to the Fisher projected training data. The classifier can be a linear discriminant(Bishop 4.1.1 - 4.1.3), a KNN classifier (Bishop 2.5.2), from Decision Theory (end of Bishop 4.1.4) using an optimum threshold, and etc. You can either use Matlab build-in function or implement one by yourself. This function returns the Fisher projection coefficients and the corresponding fitted classifier necessary for the testing function.
3. Finally you need to test the performance of your projection and classification method. You can either use Matlab build-in function or implement one by yourself. You have to quantitatively evaluate your classification method on **BOTH** datasets given.

2.2 Requirements

Your report for Part 2 MUST include:

1. Equations that define your Fisher projection and your estimated model parameters.
2. Classification results on **BOTH** the training and testing data for **BOTH** dataset. This must include:
   (a) **Confusion matrices** (a definition of confusion matrix can be found at https://en.wikipedia.org/wiki/Confusion_matrix). *The function of drawing confusion matrices is given in the starter code.*
   (b) **Classification rates for each class and an overall classification rate.**
3. Analyze your results in the report. The report should not be merely a stack of figure and tables, but also words to explain the meaningful observations behind the numbers. For example, you must compare the classification results on training and testing data, do you observe an over-fitting problem?
4. Do you observe any outliers in the data? If so, describe how do they affect the classification results and point out the outliers within your results. If not, display the data set to illustrate this point.

2.3 Extra credits

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

1. Choose samples from any two classes of a dataset and show that, for the two-class problem, Fisher criterion is a special case of least squares(Bishop 4.1.5) – 5pts.
2. Visualize the decision boundaries by projecting the features to 2 dimensions. *The function of drawing decision boundaries is given in the starter code.* – 5pts.
3. Plot the training/testing data points, with indicating the wrongly classified points based on your results from confusion matrix for Fisher projection + KNN (or other classifier you use). *You will need to modify the visualize function to achieve this.* – 5pts.

3 Submission

Your submission should include two parts: code and report. 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. Name the zip file as **yourFirstname_yourLastName_projectNo.zip**, for example, **Shimian_Zhang_1.zip**.

Your code must be reasonably commented and written in an understandable manner. A **ReadMe** document is needed to explain the function of each code file, and which data file is used to generate which model. Graders will read and run your code.

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

4 Grading Criteria

- Part 1 (**50 Points**)
  - Maximum likelihood (**15 points**)
    - Show the derived equations (and the deriving of the equations). (5 points)
    - Code implementation. (10 points)
  - Maximum a posteriori (**15 points**)
    Details the same.
  - Write-up the report to summarize, compare and contrast the results (**20 points**)
    - Figures and tables with nice visualization and description. (10 points)
    - Comparison with different methods and summary. (10 points)
  - Extra Credits
- Part 2 (**50 Points**)
  - Implementation of Fisher projection (**15 points**)
    - Provide close-form weight equation (providing derivation steps will also be sufficient) (5 points).
    - Code implementation. (10 points)
  - Use/Implement a classifier to do classification after Fisher projection (**5 points**)
  - Results on two datasets (**30 points**)
    For each dataset:
    - Confusion matrix. (5 points)
    - Classification rates for each class and overall classification rate. (5 points)
    - Conclusion (5 points)
  - Extra Credits

5 Common Issues

Familiarize yourself with Matlab by reviewing the online tutorials and Matlab help documents.
Don't try to do this at the last minute.
Good luck!

1. I run into error when running the starter code. How to fix it?
   Kindly check if Image Processing Toolbox has been installed to your Matlab.

2. How to solve the linear system of equations?
   You can use the backslash operator "\" rather than the inverse function to solve the linear system of equations. Kindly check `https://www.mathworks.com/help/matlab/ref/mldivide.html` for more reference.

3. How to compute eigenvalue & eigenvector?
   You can use Matlab build-in function eig for computation.

4. How to convert training label vector of length N into a $N \times K$ matrix?
   The idea is to use 1-of-K coding scheme, which is to use a vector $\mathbf{t}$ of length K to represent a label with K classes. Such that if the class is C_j, then all elements of $\mathbf{t}$ are set to zero except t_j set to one.

   For example, we have $K = 5$ classes. And the label is class 3, then its corresponding 1-of-K coding will be:

   $$\mathbf{t} = (\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0})^{\mathbf{T}}$$

   And finally, for training label vector of length N, we can use this scheme to expend it to a $N \times K$ matrix.

5. How to do classification by Fisher Projection?
   Notice that fisher projection can ONLY do dimension reduction but not classification. The goal of fisher's linear discriminant is to find a projection that maximized the class separation. Hence, after implementing Fisher Projection, you are required to train a classifier to the projected training data (in a reduced dimension). You are welcome to use any classifier from lecture or based on your own knowledge.

Notice that this section will be updated if more common issues are asked by the students.