Statistical Learning for Engineers (EN.530.641)
Homework 6

Jin Seob Kim, Ph.D.
Senior Lecturer, ME Dept, LCSR, JHU

Out: 10/14/2022
due: 10/24/2022 (Monday) by midnight EST

1 Solve Exercise Problem 4.2 (a), (b), and (c) in p.135 of ESL. To repeat:
Suppose we have features $\mathbf{x} \in \mathbb{R}^p$, a two-class response, with class sizes N_1, N_2, and the target coded as $-N/N_1$, N/N_2.

(a) Show that the LDA rule classifies to class 2 if

$$\mathbf{x}^\top \hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > \frac{1}{2}(\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)^\top \hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) - \log\left(\frac{N_2}{N_1}\right),$$

and class 1 otherwise.

(b) Consider minimization of the least squares criterion

$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2.$$

Show that the solution $\hat{\boldsymbol{\beta}}$ satisfies

$$\left[(N-2)\hat{\Sigma} + N\hat{\Sigma}_B\right]\boldsymbol{\beta} = N\left(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1\right)$$

(after simplification), where $\hat{\Sigma}_B = \frac{N_1 N_2}{N^2}\left(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1\right)\left(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1\right)^\top$.

(c) Hence show that $\hat{\Sigma}_B\boldsymbol{\beta}$ is in the direction $(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)$ and thus

$$\hat{\boldsymbol{\beta}} \propto \hat{\Sigma}^{-1}\left(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1\right).$$

Therefore the least-squares regression coefficient is identical to the LDA coefficient, up to a scalar multiple.

2 In an earlier homework assignment, you considered k-nearest neighbors algorithm for the classification problem (Figure 2.2 and 2.3). In this problem, you will apply a linear regression algorithm for the classification. Use the same training dataset as in the previous homework assignment.

(a) Write your own python codes to generate a figure similar to Figure 2.1 in ESL.

(b) Use Scikit-Learn library (`sklearn`) to plot the figure. For example, you will need `LinearRegression` in `linear_model` in `sklearn`.

3 (related to Figure 2.11 in ESL)
In this problem, we want to generate a plot which looks like Figure 2.11 in ESL (p.38). There is a dataset on Canvas. It is basically about whether a person will subscribe to a loan or not.

(a) Perform the data processing so that you have well-defined inputs (denoted as $\mathbf{x}_i$) and the qualitative outputs (denoted as y_i) which are two-class categorical variables (i.e., 0-1 class). Hint: Look for "Integer coding" or "One-Hot Encoding" for data preprocessing.

(b) Build k-NN models with varying k values (you can use Scikit-learn to build up models). For now, let us divide the data into two sets: among all the data, you can randomly select about 25 % of the data which will be used as a test data. The remaining 75 % will be used as the training data. Now plot the training and test errors (see below) as functions of k, as in Figure 2.11. Compare accuracies and determine which models are Bias or Variance heavy. How can the models be made more complex or simpler to fix Bias or Variance?

As two kinds of the prediction error, again for now, let us use the following: for the training error,

$$\text{training error} = \frac{1}{N_1}\sum_{i=1}^{N_1}(y_i - \hat{y}_i)^2 .$$

where N_1 denotes the number of training data. On the other hand, the test error can be similarly calculated as

$$\text{test error} = \frac{1}{N_2}\sum_{i=1}^{N_2}(y_i - \hat{y}_i)^2$$

where N_2 denotes the number of points in the test set.

4 You will perform classification with the data in the following ESL website:
https://web.stanford.edu/ hastie/ElemStatLearn/.
Select "Data" on the left menus, then find "Vowel". You can find two sets of data: one is the training set, and the other is the test set. Perform the classification using:

(a) Linear discriminant analysis (LDA);

(b) Quadratic discriminant analysis (QDA);

(c) Logistic regression.

If you use `sklearn`, you will need:

- For LDA, `LinearDiscriminantAnalysis` from `sklearn.discriminant_analysis`;
- For QDA, `QuadraticDiscriminantAnalysis` from `sklearn.discriminant_analysis`;
- For logistic regression, `LogisticRegression` from `sklearn.linear_model`. Set `multi_class='auto'`. Try different options for `solver`, and see the results. Which gives the smallest test error?

For loading .txt files, you can use `pandas.read.csv` function. Train each model with the training data, and then use the test data to report expected error of each case, which can be computed as

$$\text{test error} = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}(y_i - \hat{y}_i)^2$$

where N_{test} denotes the number of data points in the test dataset $\{(\mathbf{x}_i, y_i)\}$, and $\hat{y}_i$ denotes the fitted output at $\mathbf{x}_i$.

Submission Guideline

- For analytic parts (e.g., Problem **1** and the results of computational problems, **2**, **3** and **4**), submit your homework answers in a single pdf format, including plots, to "HW6_analytical" on the gradescope.
- No more than two (2) homework problems may be on the same page. In other words, for each problem your answers should be on a separate set of pages. Then when submitting, you should assign the pages to each problem on Gradescope.
- *Show your work.*
- Submit all your python codes in a single .zip file that contains codes for each problem (name them by including the problem number). Name your single zip file submission as "YourName_HW6.zip". For example, "JinSeobKim_HW6.zip" for a single zip file. Submission will be done through "HW6_computational" on the gradescope.
- Just in case you have related separate files, please make sure to include *all the necessary files*. If TAs try to run your function and it does not run, then your submission will have a significant points deduction.
- Make as much comments as possible so that the TAs can easily read your codes.