Samantha, Morales
CSCI 3203: Tashfeen's Machine Learning I

Homework 4

You're *not allowed* to use any libraries other than Numpy or Matplotlib.

We start by reminding ourselves of the variables from the class where $x_i^T \in \mathbb{R}^p$ is a single data point in the data set and $y_i \in \mathbb{R}$ is the corresponding label.

$$x_i = \begin{bmatrix} t_1 \\ \vdots \\ t_{p-1} \\ 1 \end{bmatrix}, \quad X_{(n,p)} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}, \quad \beta_{(p,1)} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad y_{(n,1)} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Therefore, the linear regression model under the appropriate assumptions mentioned in class is,

$$f(x) = x^T\beta$$

Using the quadratic error function we get,

$$\begin{aligned} J(\beta) &= \sum_{i=1}^{n}(f(x_i) - y_i)^2 \\ &= \sum_{i=1}^{n}(x_i^T\beta - y_i)^2 \\ &= (X\beta - y)^T(X\beta - y) \\ &= (\beta^T X^T - y^T)(X\beta - y) \\ &= \beta^T X^T X\beta - \beta^T X^T y - y^T X\beta - y^T y \end{aligned}$$

Note that,

$$\beta^T X^T y = (y^T X\beta)^T \text{ with dimensions } (1,p)\cdot(p,n)\cdot(n,1) = (1,1)$$

Therefore $\beta^T X^T X\beta - \beta^T X^T y - y^T X\beta - y^T y$ is equal to,

$$\beta^T X^T X\beta - 2\beta^T X^T y - y^T y$$

In order to minimise this error function in terms of β we take the partial derivative.

$$\begin{aligned} \frac{\partial J(\beta)}{\partial \beta} &= \frac{\partial}{\partial \beta}\left(\beta^T X^T X\beta - 2\beta^T X^T y - y^T y\right) \\ &= 2X^T X\beta - 2X^T y - 0 \\ &= 2X^T X\beta - 2X^T y \end{aligned}$$

Set the derivative to zero and solve for β,

$$\begin{aligned} \frac{\partial J(\beta)}{\partial \beta} &= 2X^T X\beta - 2X^T y \\ 2X^T X\beta - 2X^T y &= 0 \\ 2X^T X\beta &= 2X^T y \\ X^T X\beta &= X^T y \\ \beta &= (X^T X)^{-1} X^T y \end{aligned}$$

Question 1. What is wrong with the following approach of solving the derivative of the quadratic error function $J(\beta)$ for β after setting it to zero?

$$\begin{aligned} 2X^TX\beta - 2X^Ty &= 0 \\ 2X^TX\beta &= 2X^Ty \\ X^TX\beta &= X^Ty && \text{Multiply by } (X^T)^{-1} \\ X\beta &= y \\ \beta &= X^{-1}y \end{aligned}$$

Question 2. Code listing 1 generates and plots some linear data. Using the equation,

$$\beta = (X^TX)^{-1}X^Ty$$

plot the line of best fit in the same figure. Place the code for this question in a file called `bestfit.py` and give the plot.

Question 3. Read the two discussions bellow,

1 racist data destruction?
2 Boston housing price dataset was removed from scikit-learn 1.2

Do you think the data set should have been removed even if it was unethical? Do you think the original authors were actually pushing systematic racism or just accounting for it in their data? Discuss your thoughts in one paragraph.

Question 4. Download the `BoustonHousing.csv`. For the description of each column, visit,
`http://lib.stat.cmu.edu/datasets/boston`
For this question, we will try to predict the "Median value of owner-occupied homes in $1000's."

1 Plot each column on x–axis against the last column on y–axis. Label the axis and title the plots appropriately. Give the two plots that you find the most linearly correlated. Which columns are these plots graphing?
2 Using the two columns you picked above for $p = 2$ and the last column as labels, split the data into training and testing sets. The test set should contain 100 records. Use the remaining training set to calculate the vector β. With this β report the root mean squared error (RMSE) on the 100 records of the training set. Note that,

$$\text{RMSE} = \sqrt{\left(\frac{J(\beta)}{n}\right)}$$

Don't use any loops for other than drawing plots in the first part. Put all the code relevant to this question in a file called `regression.py`.

Submission Instructions

1 Submit a PDF that answers the questions and contains all the plots that the assignment asks for.
2 Submit your `bestfit.py`.
3 Submit your `regression.py`.