Optimization Project

Ziyu Guo

Project Finished by 2020/02/07
Report Start 2020/02/11
Report Last Updated 2020/02/11

Overview

This is a small Machine Learning project. I have already finished all the works by 2020/02/07. There are still few things I need to update in my code and here is the report.

About this project, it includes 4 parts which focus on concepts from linear algebra and optimization theory, with emphasis on linear least sqare estimation. I also implement a linear classification algorithm, visualize its decision boundary, and test its accuracy.

Data files are provided, all my programming works are in the py file and the output files are png files.

Part 1

Objective

For the part 1, I loaded data via python which is divided into two classes based on gender. The data focus on two different categories for each individual: the height (in mm) and body mass index(BMI)

Code

```
male_train = []
female_train = []
with open("male_train_data.csv", "r") as csv_file:
    reader = csv.reader(csv_file, delimiter=',')
    csv_file.readline()
    for row in reader:
        male_train.append(row[1:])
csv_file.close()
with open("female_train_data.csv", "r") as csv_file:
    reader = csv.reader(csv_file, delimiter=',')
    csv_file.readline()
    for row in reader:
        female_train.append(row[1:])
csv_file.close()
```

Part 2

In this part, I build a two-class linear classifier via optimization tools. There is a dataset D with training samples $(\mathrm{x}_j, y_j)_{j=1}^{N}.The\ input\ vector\ x_j \in \mathbb{R}^d. The\ scalar\ y_j \in \{+1, -1\}.$
$The\ decision\ boundary\ of\ the\ linear\ classifier\ as\ the\ set\ \{x \in \mathbb{R}^d \mid g(x) = 0\}\ where:$
$g(x) = \omega^T \boldsymbol{x} + \omega_0.$

To minimize the error of the decision boundary, I solve the least square problem and by substituting:

$$\begin{bmatrix} \boldsymbol{\omega}^* \\ \omega_0^* \end{bmatrix} = \underset{\boldsymbol{\omega}, \omega_0}{argmin} \sum_{j=1}^{N} (\boldsymbol{\omega}^T \boldsymbol{x}_j + \omega_0 - y_j)^2 \tag{1}$$

Let

$$\boldsymbol{\theta} = (\boldsymbol{\omega}, \omega_0) \tag{2}$$

I can rewrite the (1) as

$$\boldsymbol{\theta^*} = \underset{\boldsymbol{\theta}}{argmin} \|\boldsymbol{A\theta} - \boldsymbol{b}\|^2 \tag{3}$$

Where

$$A = \begin{bmatrix} \boldsymbol{x}_1^T & 1 \\ \boldsymbol{x}_2^T & 1 \\ \dots & \dots \\ \boldsymbol{x}_N^T & 1 \end{bmatrix} \ and \ \ \boldsymbol{b} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

if $\mathbf{x}_0\ is\ a\ local\ minimum\ of\ g,\ then$

$$\nabla g(\boldsymbol{x}_0) = 0. \tag{4}$$

we can get

$$\boldsymbol{\theta^*} = (\boldsymbol{A^T A})^{-1} \boldsymbol{A^T b} \tag{5}$$

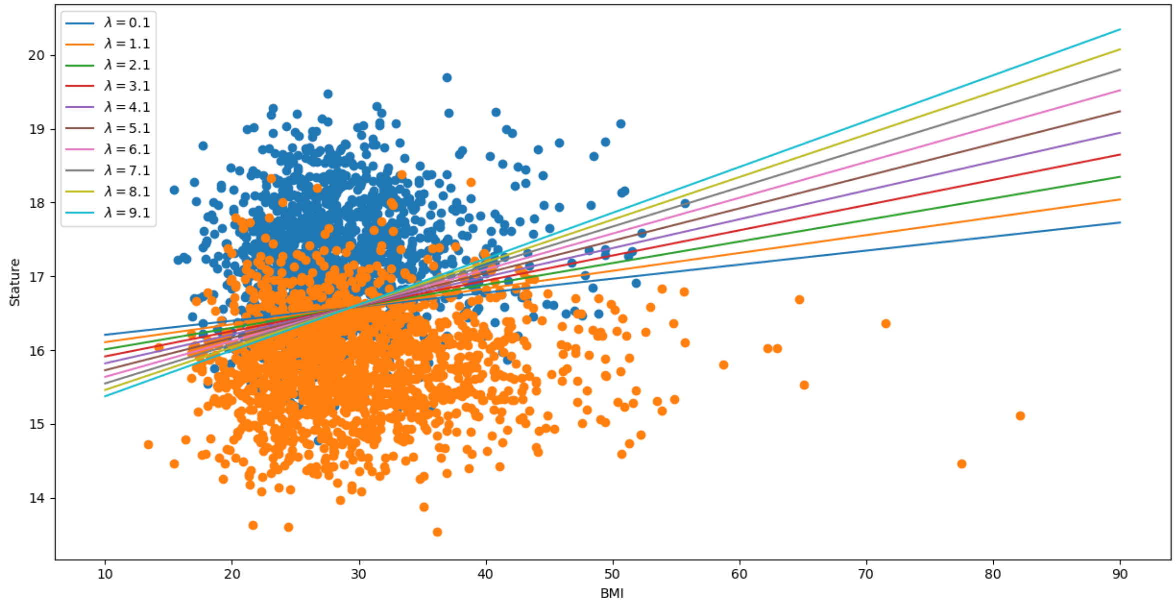
Now we need to consider whether $(\mathrm{A}^T A)^{-1}$exists or not,and A is singular or not if not exist.

Code

```
#get the size of training data
n_male = len(male_train)
n_female = len(female_train)

#get b: 1 for male, −1 for female.
class_one = np.ones((n_male))
class_mone = np.ones((n_female))* (−1)
y = (class_one, class_mone)
b = np.concatenate(y)

#get A
#Pre_A = np.array(np.concatenate((male_train,female_train)))
Pre_A = np.array((male_train + female_train), dtype=float)
ones = np.ones(((n_male + n_female), 1))
A = np.concatenate((Pre_A, ones),axis=1)


#calculate the weight vector
A_T = np.transpose(A)
ATA = np.dot(A_T,A)
invA = np.linalg.inv(ATA)
invAAT = np.dot(invA, A_T)
w_vector = np.dot(invAAT, b)

#Result: [−1.23396767e−02  6.67486843e−03 −1.07017505e+01]

#Another way to solve the problem by using CVX toolbox.
x = cp.Variable(len(A[0]))
objective = cp.Minimize(cp.sum_squares(A*x − b))
#constraints = [0 <= x, x <= 1]
prob = cp.Problem(objective)
# The optimal objective value is returned by  prob  .solve()  .
result = prob.solve()
print(x.value)
```

Contine Updating PART 3, 4

For PART 3, I tested the performance of the linear classifier and PART 4 is about regularization and solving different optimization problems with constrains. I have already finished the coding and it is in the py file. For the output figures, I provide exercise4.png as an example below. Feel free to check other png files. Figure1.png needs to be fixed later.



$\lambda = 0.1$
$\lambda = 1.1$
$\lambda = 2.1$
$\lambda = 3.1$
$\lambda = 4.1$
$\lambda = 5.1$
$\lambda = 6.1$
$\lambda = 7.1$
$\lambda = 8.1$
$\lambda = 9.1$
Stature
BMI