Homework 1

Machine Learning

Spring 2024

Due date: March 10, 2024

Upload your answers as a pdf file to your google drive directory. For the programming questions, in addition to your source code, include an example input and output, and include a short explanation of your code.

You can research your answers online or using textbooks, and you can discuss your solutions with your classmates; but you need to disclose all the resources that you used in your report. If you use tools like ChatGPT, include your prompt and the answer in your report.

1. Implement linear regression in python:
    a. Write a function that takes as input n (number of samples), p (number of predictors), and q (number of predictors related to target variable), and as output returns $\boldsymbol{y}$ ($n \times 1$ vector), X ($n \times p$ matrix), and $\boldsymbol{\beta}$ ($p \times 1$ vector) where $\boldsymbol{y}$ is the linear combination of q of columns of X and $\boldsymbol{\beta}$ is the coefficients used for that linear combination. You can generate the matrix X using normal distribution, select q of its columns randomly, generate $\boldsymbol{\beta}$ using normal distribution, and generate $\boldsymbol{y}$ using X and $\boldsymbol{\beta}$. You can also add a small noise $\boldsymbol{\epsilon}$ (normally distributed with $\mu = 0$ and a small variance).
    b. Write a function that takes as input a matrix X and a vector $\boldsymbol{y}$ and solves the linear regression using the closed form formula.
    c. Write a function that takes as input a matrix X, a vector $\boldsymbol{y}$, a learning rate α, a convergence threshold th, and a maximum number of iterations max_iter, and solves the linear regression using gradient descent algorithm.
    d. Compare the results of sections b and c to results of scikit-learn implementation of linear regression.

e. Write a function that takes as input a matrix X, a vector $\boldsymbol{y}$, a penalty hyperparameter of λ, a convergence threshold th, and a maximum number of iterations max_iter, and solves the LASSO linear regression using cyclic coordinate descent algorithm. For simplicity, you can normalize X so that its columns have zero mean and unit length, and normalize y to have zero mean and unit length. Compare your results to the results of scikit-learn implementation of LASSO linear regression.

2. Solution to Ridge and Elastic Net linear regressions.
   a. Consider the following two optimizations:

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2$$

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{y}^* - X^*\boldsymbol{\beta}||_2^2$$

   Where X is $n \times p$ matrix and $X^* = \begin{bmatrix} X \\ \lambda \mathrm{I} \end{bmatrix}$ is $(n+p) \times p$ matrix, y is $n \times 1$ vector and $y^* = \begin{bmatrix} y \\ 0 \end{bmatrix}$ is $(n+p) \times 1$ vector. Show that the solution to both problems is the same.

   b. Show that you can solve Elastic Net with LASSO using the same trick: Create a y^* and a X^* and show that the LASSO solution for them is the same as Elastic Net solution for the original y and X.