Institute of Biochemistry and Biophysics, Department of Bioinformatics

Homework 2

1. 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.

2. 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

3. Comparative Clustering of Shape and S-Set Datasets

Datasets:

- From the "Shape Sets" (from here: https://cs.joensuu.fi/sipu/datasets/) download Pathbased, Spiral, Jain, and Flame
- From the "S-Sets" download S1 and S4 datasets

For each dataset:

a. Cluster the data using the following methods. Use the same number of clusters as specified in the dataset. For each method, describe what distance or similarity metric you have used.

   i. K-means
   ii. Hierarchical clustering (average linkage)
   iii. Hierarchical clustering (single linkage)
   iv. Hierarchical clustering (complete linkage)
   v. Spectral clustering. Describe how you defined the graph.

b. Visualize the resulting clustering (use the scatter plot of the data points and color the points by cluster assignment).

c. For datasets from the "Shape Set" where the true cluster is known, evaluate each clustering result using the purity index.

For each dataset, compare the results of each pair of clustering methods using the Rand index. Visualize the results in a 5x5 heatmap.

4. **PCA on MNIST dataset**

Dataset:

- Load MNIST dataset (could be accessed using from keras.datasets in Python)
- Separate them by label into 10 smaller sets

For each set:

a. Flatten the pictures and apply PCA
b. Plot first PC vs. Second PC
c. Assume the points in this scatter plot are spread between (x_1, x_2) and (y_1, y_2) (which are the min and max of PC1 and PC2). Split this space into a 5x5 grid, and for each cell select a point that is closest to the center of that cell. Highlight these points in the scatter plot from the previous step.
d. Draw the original pictures corresponding to the 25 selecting points.
(See Figure 14.23 of Element of Statistical Learning).

5. **PCA and Clustering on gene expression data**

Dataset: Download the expression data from here.

The data is collected from 3 different SRA datasets and samples corresponding to two tissues (Leaf and Root) are combined into one expression matrix. The CPM normalization and log transformation were applied to the gene expression data, and batch effect was removed using the combat function in sva package in R. This zip file should include:

- The information about each sample: Leaf_Root_annotation.csv
  This should include Project label (accession ID of the original dataset) and Tissue label (tissue of each of the samples).
- The log-transformed raw expression matrix: Leaf_Root_raw_data.csv
- The normalized and batch effect corrected expression matrix: Leaf_Root_normalized_data.csv

For each expression matrix:

a. Perform clustering (with 2 and 3 clusters) on the samples and compare the results to Project and Tissue label in the annotation file.
b. Perform PCA, and color the samples once with Project label and once with the Tissue label.

Compare the results of raw and normalized data.

Keynotes:

For the programming questions, in addition to your source code, include an example input and output and a short explanation of your code.

Send your answers as a zip file via email to Mahdi.Anvari7@ut.ac.ir by **January 10, 2025**. Make sure to include your name and student number in the email subject line.

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.

Good luck