CS 573, Homework 1

Out: September 12, 2023, Due: September 24, 2023 at 11:59 pm, Total: 75

Note:

- This homework will carry 7.5 points towards your final score.
- Please answer all the questions below.
- Homeworks are individual work, please do not collaborate with others inside or outside of the class. Software will be used to determine code similarity, so do not take a chance.
- Start early and if you need help, post your questions on piazza or use instructor/TA's office hour.

Short Questions

a. (5 points) Say, projection of a vector $\mathbf{y}$ on $\mathbf{x}$ is $\mathbf{z}$. Show that $\mathbf{y}^T\mathbf{z}$ is equal to $\|\mathbf{z}\|^2$.

b. (10 points) In class, we have shown how to solve quadratic optimization with linear constraint by using Lagrangian function. In this problem, we will use Lagrangian for solving a simple quadratic optimization problem. Say, given a equation of circle in 2-D: $(x-2)^2+(y-5)^2=9$, find the point on this circle which is farthest from the origin.

c. (5 points) Consider the following set of points in 3-D. Project the points to the subspace spanned by the following two orthogonal vectors: $(2,1,2)^T$, and $(-1,0,1)$. What is the value of MSE (Mean square error) after projection?

$$D = \begin{pmatrix} 1 & -1 & 8 \\ 4 & 2 & 1 \\ 0 & 1 & 5 \\ 5 & -2 & -5 \\ -2 & 0 & -7 \\ 3 & 5 & 3 \end{pmatrix}$$

d. (5 points) You are given the matrix A, L, and Δ as below, such that L is left singular matrix, and Δ is singular value matrix. Find the Rt matrix without using a SVD program. You must need to show your work, just writing the content of Rt will not earn you any point.

$$A = \begin{pmatrix} 2 & 0 & -4 & 3 \\ -5 & 1 & 8 & 0 \\ 3 & -3 & 0 & 2 \\ 5 & 1 & 2 & 1 \\ 0 & 2 & 3 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} -0.4286 & -0.0026 & -0.1034 & 0.8939 & -0.0814 \\ 0.8338 & -0.1394 & -0.3332 & 0.3387 & -0.2441 \\ -0.2242 & -0.4303 & -0.7934 & -0.1709 & 0.3255 \\ -0.1063 & -0.8429 & 0.3315 & -0.0522 & -0.4069 \\ 0.2438 & -0.2914 & 0.3728 & 0.2332 & 0.8138 \end{pmatrix}$$

$$D = \text{Diag}([11.1826,\ 6.2933,\ 3.6019,\ 2.7147])$$

Programming Question

Download the Magic dataset from `https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope` The dataset has 10 real attributes, and one categorical attribute, which is the class label. For this assignment we will ignore the class label attribute. Now, answer the following questions:

a. (5 points) Write a script to apply z-normalization to this dataset. For the reamining questions, you should use z-normalized dataset instead of the original dataset.

b. (5 points) Compute the sample covariance matrix ($\mathbf{\Sigma}$) as sum of outer product of each centered point (see Equation 2.31). Verify that your answer matches with the one using `numpy.cov` function. For both the cases normalize by n instead of $n - 1$

c. (10 points) Compute the dominant eigenvalue and eigenvector of the covariance matrix $\mathbf{\Sigma}$ via the power-iteration method, which works as follows:

$$\mathbf{x_0} = \begin{pmatrix} x_{0,1} \\ x_{0,2} \\ \vdots \\ x_{0,d} \end{pmatrix} \tag{1}$$

Let, $\mathbf{x_0}$ be a starting vector in R^d, where d is a vector of appropriate dimensions. In each iteration i, we compute the new vector: $\mathbf{x_i} = \mathbf{\Sigma x_{i-1}}$. We then find the element of $\mathbf{x_i}$ that has the maximum absolute value, say at index m_i. For the next round, to avoid numerical issues with large values, we re-scale $\mathbf{x_i}$ by dividing all elements by x_{i,m_i}, so that the largest value is always 1 before we begin the next iteration. To test convergence, you may compute the norm of the difference between the scaled vectors from the current iteration and the previous one, and you can stop if this norm falls below some threshold, say 0.000001. That is, to stop, check if $\|\mathbf{x_i} - \mathbf{x_{i-1}}\| < 0.000001$ is true. For the final eigen-vector, make sure to normalize it, so that it has unit length. Also, the ratio $x_{i,m_i}/x_{i-1,m_{i-1}}$ gives you the largest eigenvalue. Verify your answer using the numpy `linalg.eig` function.

d. (5 points)

Use `linalg.eig` to find the first two dominant eigenvectors of $\mathbf{\Sigma}$, and compute the projection of data points on the subspace spanned by these two eigenvectors. Now, compute the variance

of the datapoints in the projected subspace (Do not print the projected datapoints on stdout, only print the value of the variance).

e. (5 points) Use `linalg.eig` to find all the eigenvectors, and print the covariance matrix $\mathbf{\Sigma}$ in its eigen-decomposition form ($\mathbf{U\Lambda U^T}$)

f. (5 points) Write a subroutine to implement MSE (mean square error). Compute MSE value when the data points are projected on the space spanned by the first two eigenvectors of the covariance matrix and show that it equals to the sum of the eigenvalues expect the first two.

g. (5 points) For this question, you should use the class label which we ignored in the other questions. Plot the datapoints after projecting those in the first two principal components. Use different colors for instances of different classes.

h. (10 points) Write a subroutine to implement PCA Algorithm. Use the program above and find the principle vectors that we need to preserve 95% of variance? Print the co-ordinate of the first 10 data points by using the above set of vectors as the new basis vector.

Deliverables

For short question, submit a hand-written pdf file in Canvas before due date. For programming question, submit a scriptfile and an output file. Write a script named assign1-iuUserName.py that takes as input the data filename, and prints the answers to the questions to stdout. Do not hard code the filename inside the script, but rather you should read the file name from the command line. Save your output to a textfile and name it assign1-iuUserName.txt, Submit your script and output file in Canvas before the due date.