IITM-CS5691 : Pattern Recognition and Machine Learning — Release Date: October 9, 2023
Assignment II — Due Date : October 23, 2023, 23:59

Roll No: CS23E001 — Name: Shuvrajeet Das
Collaborators (if any):
References/sources (if any):

- Use LaTeX to write-up your solutions (in the solution blocks of the source LaTeX file of this assignment), submit the resulting rollno.asst2.answers.pdf file at Crowdmark by the due date, and propery drag that pdf's answer pages to the corresponding question in Crowdmark (do this propery, otherwise we won't be able to grade!). (Note: **No late submissions** will be allowed, other than one-day late submission with 10% penalty or four-day late submission with 30% penalty.)
- Please upload to moodle a rollno.zip file containing three files: rollno.asst2.answers.pdf file mentioned above, and two code files for the programming question (rollno.ipynb file and rollno.py file). Do not forget to upload to Crowdmark your results/answers (including Jupyter notebook **with output**) for the programming question.
- Collaboration is encouraged, but all write-ups must be done individually and independently, and mention your collaborator(s) if any. Same rules apply for codes written for any programming assignments (i.e., write your own code; we will run plagiarism checks on codes).
- If you have referred a book or any other online material or LLMs (Large Language Models like ChatGPT) for obtaining a solution, please cite the source. Again don't copy the source *as is* - you may use the source to understand the solution, but write-up the solution in your own words (this also means that you cannot copy-paste the solution from LLMs!). Please be advised that *the lesser your reliance on online materials or LLMs* for answering the questions, *the more your understanding* of the concepts will be and *the more prepared you will be for the course exams*.
- Points will be awarded based on how clear, concise and rigorous your solutions are, and how correct your answer is. The weightage of this assignment is 12% towards the overall course grade.

1. (8 points) [SPECTRAL CLUSTERING - LAPLACIAN EIGENMAP] Consider a simple undirected graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges. Let A be the binary adjacency matrix of the graph (i.e., the symmetric 0-1 matrix where 1 indicates the presence of the corresponding edge; diagonal entries of A are zero). Let $x \in \mathbb{R}^n$ denote the node scores.

   Let the graph Laplacian matrix be $L = D - A$ seen in class, with D being the diagonal matrix of node degrees. Let $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$ denote the eigenvalues of the graph Laplacian L (sometimes also referred to as L_G to explicitly mention the graph).

   (a) (2 points) Show that $x^T L x = \sum_{(i,j)\in E}(x_i - x_j)^2$, and hence argue very briefly why $\lambda_i \geqslant 0$ for $i = 1, 2, \ldots, n$?

Solution: Given, $G = (V, E)$ with $|V| = n$, $|E| = m$ and $L = D - A$,
The laplacian matrix L is defined by $L = D - A$ also we,ve to find $x^T Lx$
To derive the expression $x^T Lx = \sum_{i=1}^{n} \deg(i)x_i^2 - \sum_{(i,j)\in E} x_i x_j = \sum_{(i,j)\in E}(x_i - x_j)^2$, we'll step through each term.
1. $x^T Lx$:

$$x^T Lx = x^T(D - A)x$$

2. $x^T Dx$:
The term $x^T Dx$ is a quadratic form representing the diagonal elements of the matrix product Dx. In this case, D is a diagonal matrix, so the product Dx simply scales each element of x by the corresponding degree of the node. This term becomes:

$$x^T Dx = \sum_{i=1}^{n} \deg(i)x_i^2$$

3. $x^T Ax$:
The term $x^T Ax$ is also a quadratic form representing the off-diagonal elements of the matrix product Ax. In this case, A is the adjacency matrix of the graph, and A_{ij} is 1 if there is an edge between nodes i and j, and 0 otherwise. This term becomes:

$$x^T Ax = \sum_{(i,j)\in E} a_{ij} x_i x_j$$

Combining all three terms:

$$x^T Lx = \sum_{i=1}^{n} \deg(i)x_i^2 - \sum_{(i,j)\in E} a_{ij} x_i x_j$$

Now, we want to express this in terms of the differences between x_i and x_j, also $a_{ij} = 2$

$$\begin{aligned}
&= \sum_{i=1}^{n} \deg(i)x_i^2 - \sum_{(i,j)\in E} a_{ij} x_i x_j \\
&= \sum_{i=1}^{n} \deg(i)x_i^2 - \sum_{(i,j)\in E} 2x_i x_j \\
&= \sum_{(i,j)\in E} x_i^2 + x_j^2 - \sum_{(i,j)\in E} 2x_i x_j \\
&= \sum_{(i,j)\in E} (x_i - x_j)^2
\end{aligned}$$

(b) (1 point) If G has 3 connected components, what is the multiplicity of the eigen value 0 of L_G, and what are the corresponding eigen vectors?

> **Solution:** The multiplicity is 3 and the corresponding eigenvectors are Certainly! The eigen vectors corresponding to the eigenvalue 0 of the Laplacian matrix L_G for a graph with 3 connected components can be written in LaTeX as follows:
>
> $$x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \; x_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \; x_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
>
> In these vectors, the 1 indicates the position of the vertex in the corresponding connected component, and all other entries are 0.

(c) (2 points) Let's add one edge to G to obtain a new graph G'. What can you say about the multiplicity of eigen value 0 of $L_{G'}$ relative to that of L_G? Will the sum of eigen values of $L_{G'}$ change compared to that of L_G; and if so, by what amount?

> **Solution:** The multiplicity of eigenvalue 0 of $L_{G'}$ will be the same as the multiplicity of eigenvalue 0 of L_G. The sum of eigenvalues of $L_{G'}$ will be greater than the sum of eigenvalues of L_G by 2.

(d) (3 points) If G is a complete graph on n nodes, we know that the multiplicity of eigen value 0 of L_G is 1; prove in this case that the multiplicity of eigen value n of L_G is $n-1$.
(Hint: Let v be an eigen vector of L_G orthogonal to the (all-ones) eigen vector of L corresp. to eigen value 0. Assume, without loss of generality, that $v(1) \neq 0$. Now compute the first coordinate of $L_G v$, and then divide by $v(1)$ to compute eigen value λ.)

> **Solution:** Given a complete graph on n nodes, the Laplacian matrix L_G can be written as:
>
> $$L_G = nI - J$$
>
> Where I is the identity matrix of size $n \times n$, and J is a matrix of all ones of size $n \times n$.
> We know that the multiplicity of eigenvalue 0 of L_G is 1. Let v be an eigenvector of L_G orthogonal to the all-ones vector $\mathbf{1}$ corresponding to eigenvalue 0. Without loss of generality, we can assume that $v(1) \neq 0$.
> Now, the first coordinate of $L_G v$:
>
> $$(L_G v)_1 = (nI - J)\, v_1 - \sum_{i=2}^{n} v_i = n v_1 - (v_2 + v_3 + \ldots + v_n)$$

Since ν is orthogonal to **1**, we know that $\sum_{i=1}^{n} \nu_i = 0$. Therefore, the first coordinate simplifies to:

$$(L_G\nu)_1 = n\nu_1$$

Now, divide by ν_1 to compute the eigenvalue λ:

$$\lambda = \frac{(L_G\nu)_1}{\nu_1} = n$$

This shows that the eigenvalue n has multiplicity $n-1$, as it arises from considering eigenvectors orthogonal to the all-ones vector, excluding the eigenvector with eigenvalue 0.

2. (8 points) [PRINCIPAL COMPONENT ANALYSIS - NUMERICAL] Consider the following dataset D of 8 datapoints:

data #	x	y
1	5.51	5.35
2	20.82	24.03
3	-0.77	-0.57
4	19.30	19.38
5	14.24	12.77
6	9.74	9.68
7	11.59	12.06
8	-6.08	-5.22

You need to reduce the data into a single-dimension representation. You are given the first principal component: $PC1 = (-0.694, -0.720)$.

(a) (2 points) What is the xy coordinate for the datapoint reconstructed (approximated) from data #2 (x=20.82, y=24.03) using the first principal component of D? What is the reconstruction error of this PC1-based approximation of data #2?

Solution:

$$\text{Projected value} = \text{data point} \cdot \text{principal component}$$

For data point 2:

Projected value $= (20.82, 24.03) \cdot (-0.694, -0.720) = -0.694 \times 20.82 - 0.720 \times 24.03 = -31.3708$

To reconstruct the original data point from its projection onto the first principal component, you can use the following formula:

$$\begin{aligned}
&\text{Reconstructed data point} = \text{Projected value} \times \text{Principal component}\\
&\text{Reconstructed data point} = -31.3708 \times (-0.694, -0.720)\\
&\text{Reconstructed data point} = (21.77849, 22.606656)
\end{aligned}$$

The reconstruction error is:

$$\begin{aligned}
\text{Reconstruction error} &= \sqrt{(20.82 - 21.78)^2 + (24.03 - 22.61)^2}\\
\text{Reconstruction error} &= \sqrt{(-0.96)^2 + (1.42)^2}\\
\text{Reconstruction error} &= \sqrt{2.1476 + 2.0164}\\
\text{Reconstruction error} &= \sqrt{4.164}\\
\text{Reconstruction error} &\approx 2.04
\end{aligned}$$

The reconstruction error of this PC1-based approximation of data #2 is approximately 2.04 units.

(b) (2 points) What is the second principal component of the dataset D? How will you represent data #2 as a linear combination of the two principal components? What is the reconstruction error of this $(PC1, PC2)$-based representation of data #2?

Solution: The second principal component is $(-0.720, 0.694)$.
The representation of data #2 as the linear combination as:

$$\begin{bmatrix} -0.69 & -0.72 \\ -0.72 & 0.69 \end{bmatrix} \begin{bmatrix} 20.82 \\ 24.03 \end{bmatrix} = \begin{bmatrix} -31.7499053 & 1.69 \end{bmatrix}$$

The point after reconstruction is: $(-31.74, 1.69)$ The reconstruction error is: 37.45

(c) (2 points) Let D' be the mean-subtracted version of D. What will be the first and second principal components PC1 and PC2 of D'? What is the xy coordinate of data #2 and its PC1-based reconstruction in D'? What is the associated reconstruction/approximation error of data #2?

Solution: The mean is: $[11.52625, 14.345]$

$$\begin{bmatrix} -0.69 & -0.72 \\ -0.72 & 0.69 \end{bmatrix} \begin{bmatrix} 11.53 \\ 14.35 \end{bmatrix} = \begin{bmatrix} -18.33 & 1.66 \end{bmatrix}$$

The point after reconstruction is: $(20.8224.03)$ The reconstruction error is: 0

(d) (2 points) Let D'' be a dataset extended from D by adding a third feature z to each datapoint. It so happens that this third feature is a constant value (3.5) across all 8 datapoints. Then, what will be the three principal components of D'', and what is the xyz coordinate of the PC1-based reconstruction of data #2 in D'' and the associated reconstruction error?

Solution: The three eigen vectors are:

$$EV_1 = \begin{bmatrix} -0.694 \\ -0.719 \\ 0 \end{bmatrix} \; EV_2 = \begin{bmatrix} -0.719 \\ 0.694 \\ 0 \end{bmatrix} \; EV_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The reconstruction error is 6.326

3. (8 points) [LINEAR REGRESSION]

(a) (4 points) The error function in the case of ridge regression is given by:

$$\tilde{E}(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2 + \frac{\lambda}{2}w^Tw$$

Show that this error function is convex and is minimized by:

$$w^* = (\lambda I + \phi^T\phi)^{-1}\phi^T t$$

Also show that $(\lambda I + \phi^T\phi)$ is invertible for any $\lambda > 0$.

(Note 1: To simplify and keep your solution concise, use vector/matrix format (e.g., gradient, Hessian, etc.,) for your expressions.

Note 2: Here, the target vector $t \in \mathbb{R}^N$ and the matrix $\phi \in \mathbb{R}^{N\times d'}$ represents all the N input datapoints after transformation by the feature-mapping function $\phi(.) : \mathbb{R}^d \to \mathbb{R}^{d'}$. For example, the $\phi(.)$ for performing k-degree polynomial regression on a d-dimensional input for $k = 2, d = 2$ is given by $\phi([x_1, x_2]) = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2]$.)

Solution:

$$\tilde{E}(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2 + \frac{\lambda}{2}w^Tw$$

First derivative w.r.t w is:

$$\frac{\partial E}{\partial w} = 2\|t_n - w^T\phi(x_n)\|_2\left(\frac{t_n - w^T\phi(x_n)}{\|t_n - w^T\phi(x_n)\|_2}\right)(-\phi(x_n)) + \lambda\|w\|_2\left(\frac{w}{\|w\|_2}\right)$$

$$\frac{\partial E}{\partial w} = 2\left(t_n - w^T\phi(x_n)\right)(-\phi(x_n)) + \lambda w$$

$$\frac{\partial E}{\partial w} = -2\phi(x_n)^T t_n + 2\phi(x_n)^T\phi(x_n)w + \lambda I w$$

Now for $\lambda > 0$

$$\frac{\partial^2 E}{\partial w^2} = \lambda I + 2\phi(x_n)^T\phi(x_n)$$

Thus we can conclude λI is a positive definite for any $\lambda > 0$ and $\phi(x_n)^T\phi(x_n)$ is a positive semidefinite and their total sum is positive definite then it is strictly convex.

Now for w^*, $\frac{\partial E}{\partial w} = 0$ Then, $-2\phi(x_n)^T t_n + 2\phi(x_n)^T\phi(x_n)w + \lambda I w = 0$

$$\begin{aligned}
-2\phi(x_n)^T t_n + 2\phi(x_n)^T\phi(x_n)w + \lambda I w &= 0 \\
2\phi(x_n)^T\phi(x_n)w + \lambda I w &= 2\phi(x_n)^T t_n \\
\left(\phi(x_n)^T\phi(x_n) + \lambda I\right) w &= 2\phi(x_n)^T t_n \\
w^* &= 2\left(\phi(x_n)^T\phi(x_n) + \lambda I\right)^{-1}\phi(x_n)^T t_n
\end{aligned}$$

Since the total resultant is positive definite that is its invertible $\left(\phi(x_n)^T\phi(x_n) + \lambda I\right)$ is invertible.

(b) (4 points) Given a dataset

$$X = \begin{bmatrix} -2 & 4 \\ -1 & 2 \end{bmatrix} t = \begin{bmatrix} 3 & -1 \end{bmatrix}$$

find all minimizers of w of $E(w) = \frac{1}{2}||Xw - t||^2$, and indicate the one with the smallest norm. How does your answer change if you are looking for minimizers of $\tilde{E}(w)$ instead (assuming λ = 1)?

Solution: Given $X = \begin{bmatrix} -2 & 4 \\ -1 & 2 \end{bmatrix}$ and $t = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$, let's calculate the gradient of $E(w)$ with respect to w:

$$E(w) = \frac{1}{2}||Xw - t||_2^2$$

$$E(w) = \frac{1}{2}(Xw - t)^T(Xw - t)$$

Now, differentiate $E(w)$ with respect to w and set it equal to zero:

$$\nabla E(w) = X^T(Xw - t) = 0$$

Solving for w:

$$X^T Xw - X^T t = 0$$

$$X^T X w = X^T t$$

Now, plug in the values for X and t:

$$\begin{bmatrix} -2 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ -1 & 2 \end{bmatrix} w = \begin{bmatrix} -2 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

Using the given values:

$$X^T X = \begin{bmatrix} -2 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 0 & 20 \end{bmatrix}$$

$$X^T t = \begin{bmatrix} -2 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \end{bmatrix} = \begin{bmatrix} -7 \\ 10 \end{bmatrix}$$

The equation $X^T X w = X^T t$ becomes:

$$\begin{bmatrix} 5 & 0 \\ 0 & 20 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -7 \\ 10 \end{bmatrix}$$

Solving this system of equations:

$$5w_1 = -7 \implies w_1 = -\frac{7}{5} = -\frac{1}{5} \times 7$$

$$20w_2 = 10 \implies w_2 = \frac{10}{20} = \frac{1}{2} \times 10$$

Therefore, the solution for w without the regularization term is $w = \left(-\frac{7}{5}, \frac{1}{2}\right)$.
Solving this equation will give you the minimizer w for the given dataset X and t.
Regarding the second part of your question, if you want to find the minimizers of the regularized function $E(w) = \frac{1}{2}||Xw - t||_2^2 + \frac{\lambda}{2}||w||_2^2$ with $\lambda = 1$, the minimizer can be found by differentiating the regularized function and setting it equal to zero, similar to the first case.

$$X^T X w - X^T t + \lambda w = 0$$

4. (8 points) [Life in lower dimensions...] You are provided with a dataset of 1797 images in a folder here - each image is 8x8 pixels and provided as a feature vector of length 64. You will try your hands at transforming this dataset to a lower-dimensional space, and clustering the images in this reduced space.

   Please use the template.ipynb file in the same folder to prepare your solution. Provide your results/answers in the pdf file you upload to Crowdmark, and submit your code separately in this moodle link. The code submitted should be a rollno.zip file containing two files: rollno.ipynb file (including your code as well as the exact same results/plots uploaded to Crowdmark) and the associated rollno.py file.

Write the code from scratch for both PCA and clustering. The only exception is the computation of eigenvalues and eigenvectors for which you could use the numpy in-bulit function.

(a) (4 points) Run the PCA algorithm on the given dataset. Plot the cumulative percentage variance explained by the principal components. Report the number of principal components that contribute to 90% of the variance in the dataset.

> **Solution:** Num Principal components with variance describing more than 90% is 21.
>
> `percentage_variance_desc.png`
>
> Figure 1: Graph explaining cumulative percentage variance explained vs number of components

(b) (4 points) Perform reconstruction of data using the small number of components: [2,4,8,16]. Report the Mean Square Error (MSE) between the original data and reconstructed data, and interpret the optimal dimension $\widehat{d}$ based on the MSE values.

Solution:

MSE of The original vs The reconstructed one is: 13.4210
MSE of The original vs The reconstructed one is: 9.6279
MSE of The original vs The reconstructed one is: 6.1217
MSE of The original vs The reconstructed one is: 2.8271

The optimal dimension is 16.