IITM-CS5691 : Pattern Recognition and Machine Learning — Release Date: October 9, 2023
Assignment II — Due Date : October 26, 2023, 23:59

Roll No: MA22C043 — Name: Shubham Singh
Collaborators (if any):
References/sources (if any): Harish Guruprasad's Notes

- 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 eigen values 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:

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

with D being the diagonal matrix of node degrees, and A be the binary adjacency matrix of the graph.

$$x^T (D - A) x = \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{i,j} x_i x_j - A_{i,j} x_i x_j)$$

$$\begin{aligned} x^T (D - A) x &= \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{i,j} x_i x_j - A_{i,j} x_i x_j) \\ &= \sum_{i=1}^{n} D_{i,i} x_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} x_i x_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} x_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} x_i x_j \\ &= \frac{2}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \left(x_i^2 - x_i x_j \right) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \left(x_i - x_j \right)^2 \end{aligned}$$

now,

$$A_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases}$$

and $A_{i,j} = A_{j,i}$, we can write

$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \left(x_i - x_j \right)^2 = \sum_{(i,j) \in E} \left(x_i - x_j \right)^2$$

- L is symmetric matrix, so its eigenvectors will be pairwise orthogonal.
- if λ is an eigenvalue of L, then there exist an unit vector for which $Lx = \lambda x$ and $x^T L x = \lambda$ and $\sum_{(i,j) \in E} (x_i - x_j)^2 = \lambda$
- Hence, for any eigenvalue $\lambda_i \geqslant 0$ for $i = 1, 2, \ldots, n$

(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: If G has 3 connected components, the multiplicity of the eigen value 0 of L_G is 3.

Let C_1, C_2, C_3 are three connected components of L_G then eigenvector V_1 corresponding to eigenvalue 0 is,

$$V_1 = \begin{bmatrix} V_1^1 & V_1^2 & \dots & V_1^n \end{bmatrix}$$

$$V_1^{(i)} = \begin{cases} 1 & \text{if } i \in C_1 \\ 0 & \text{if } i \notin C_1 \end{cases}$$

similarly, for general case, V_i for $i \in 1, 2, 3$, where

$$V_i = \begin{bmatrix} V_i^1 & V_i^2 & \dots & V_i^n \end{bmatrix}$$

$$V_i^{(j)} = \begin{cases} 1 & \text{if } j \in C_i \\ 0 & \text{if } j \notin C_i \end{cases}$$

for j ∈ 1,2, . . . ,n

(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:

When we add an edge to a graph G to obtain a new graph G′, the multiplicity of the eigenvalue 0 of the Laplacian matrix $L_{G'}$ relative to that of L_G may or may not decrease by one, provided that G′ remains connected.

Here's why:

1. **Laplacian Eigenvalues:** The eigenvalues of the Laplacian matrix L_G are typically non-negative. The multiplicity of the eigenvalue 0 in L_G represents the number of connected components in the graph. If L_G has a zero eigenvalue with multiplicity k, it implies that the graph has k connected components.

2. **Adding an Edge:** When we add an edge to the graph G, it can creates a new connection between two previously disconnected components (or vertices). As a result, the graph G′ becomes connected if it wasn't already. This connectedness can reduce the connected component in G′ by one, which, in turn, means that the multiplicity of the eigenvalue 0 in $L_{G'}$ decreases by one.

3. **Change in Sum of Eigenvalues:** The sum of the eigenvalues of the Laplacian matrix increased by 1 when we add an edge to the graph.(suppose D is diagonal matrix of G and D′ be diagonal matrix of G′) then if the edge connects i_{th} and j_{th} vertices then $D'_{i,i} = D_{i,i}+1$ and $D'_{k,k} = D_{k,k}$ for $k \neq i$ hence Trace(D′) = Trace(D) +1,

$$Trace(D') = \sum_{i=1}^{n} \lambda'_i$$

where λ'_i are the eigenvalues of D′

$$Trace(D') = \sum_{i=1}^{n} \lambda'_i = Trace(D) + 1 = \sum_{i=1}^{n} \lambda_i + 1$$

where λ_i are the eigenvalues of D

$$\sum_{i=1}^{n} \lambda'_i = \sum_{i=1}^{n} \lambda_i + 1$$

In summary, when we add an edge to G to obtain G′, the multiplicity of the eigenvalue 0 of $L_{G'}$ relative to that of L_G may or may not decreases by one, and the sum of eigenvalues increases by one.

(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: If G is a complete graph on n nodes then, the adjacency matrix of G is A where each entry is 1. so, A can be written an $(XX^T - I)$, where $X_{(n,1)}$ is $\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

D can be written as $(n-1)I$, where I is identity matrix of n*n dimension. hence,

$$L = (nI - XX^T)$$

X is the eigenvector of L, as

$$LX = (nI - XX^T)X = (nX - (X^TX)X) = (nX - nX) = 0 = 0 * X$$

now, we know there are (n-1) linearly independent vectors orthogonal to X, and we can make them pairwise orthogonal using Gram-Schmidt process say, $Y_2, Y_3, \ldots, Y_n$ are the pairwise (n-1) orthonormal vectors orthogonal to X, then,

$$LY_i = (nI - XX^T)Y_i = (nY_i - X(X^TY_i))$$

and $X^TY_i = 0$

$$LY_i = (nI - XX^T)Y_i = nY_i$$

for all $i \in 2, 3, \ldots, n$

In summary, the multiplicity of eigen value n of L_G is $n - 1$.

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: the xy coordinate for the datapoint reconstructed (approximated) from data #2 (x=20.82, y=24.03) using the first principal component of D is given by,

$$\tilde{x_2} = (x_2^T PC1)PC1 + (mean^T PC2)PC2$$

$$\frac{\sum_{i=1}^{8}(x_i, y_i)}{n} = ((5.51, 5.35) + (20.82, 24.03) + (-0.77, -0.57) + (19.30, 19.38) + (14.24, 12.77) + (9.74, 9.68) + (11.59, 12.06) + (-6.08, -5.22))/n$$

$$mean = (9.29375, 9.684999)$$

$$\tilde{x_2} = ((20.82, 24.03)^T(-0.694, -0.720))(-0.694, -0.720) + ((9.29375, 9.684999)^T(-0.720, 0.694))(-0.720, 0.694)$$

$$\tilde{x_2} = (22.01345162, 22.88123278)$$

$$||x_2 - \tilde{x_2}||_2^2 = ((x_2^T PC2) - (mean^T PC2))^2$$

$$||x_2 - \tilde{x_2}||_2^2 = ((1.68642) - (0.0298893))^2$$

$$||x_2 - \tilde{x_2}||_2^2 = 2.744093940164124$$

(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 of the dataset D say PC2 is (-0.720,0.694). since we know principal components are orthonormal vectors and $\mathrm{PC1}^{\mathsf{T}}\mathrm{PC2} = 0$.

$$\tilde{x_2} = (x_2^{\mathsf{T}}\mathrm{PC1})\mathrm{PC1} + (x_2^{\mathsf{T}}\mathrm{PC2})\mathrm{PC2}$$

$$\begin{aligned}\tilde{x_2} = &((20.82, 24.03)^{\mathsf{T}}(-0.694, -0.720))(-0.694, -0.720)\\ &+((20.82, 24.03)^{\mathsf{T}}(-0.720, 0.694))(-0.720, 0.694)\end{aligned}$$

The reconstruction error of this $(\mathrm{PC1}, \mathrm{PC2})$-based representation of data #2 is 0. as $\tilde{x_2} = x_2$

$$||x_2 - \tilde{x_2}||_2^2 = 0$$

(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: $\text{Cov}(X, X) = \mathbb{E}[(X - \mathbb{E}(X))(X - \mathbb{E}(X))^T] = \mathbb{E}[XX^T] - \mathbb{E}[X]\mathbb{E}[X]^T$
Let D' be the mean-subtracted version of D, so
$\text{Cov}(D) = \mathbb{E}[XX^T] - \mathbb{E}[X]\mathbb{E}[X]^T$

$\text{Cov}(D') = \mathbb{E}[(X')(X')^T]$ where, $X' = X - \mathbb{E}(X)$

we can see that, $\text{Cov}(D') = \mathbb{E}[(X - \mathbb{E}(X))(X - \mathbb{E}(X))^T] = \mathbb{E}[XX^T] - \mathbb{E}[X]\mathbb{E}[X]^T$

$\text{Cov}(D') = \text{Cov}(D)$ so the Principal Components(eigenvectors) will be same for both the data. so,
$PC1 = (-0.694, -0.720)$
$PC2 = (-0.720, 0.694)$
$\mathbb{E}[X] = (9.29375, 9.684999)$

The xy coordinate of data #2 is [(20.82, 24.03) - (9.29375,9.684999)] =(11.52625,14.34501)
The associated reconstruction/approximation error of data #2.

$$\tilde{x_2'} = (x_2'^T PC1)PC1$$

$$||x_2' - \tilde{x_2'}||_2^2 = ||(x_2'^T PC2)PC2||_2^2 = (x_2'^T PC2)^2$$

we know, $x_2' = x_2 - \mathbb{E}[X]$

$$||x_2' - \tilde{x_2'}||_2^2 = ((x_2 - \mathbb{E}[X])^T PC2)^2$$

from previous,

$$||x_2' - \tilde{x_2'}||_2^2 = 2.744093940164124$$

(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:

$PC1 = (-0.694, -0.720, 0)$
$PC2 = (-0.720, 0.694, 0)$
$PC3 = (0, 0, 1)$

$$\tilde{x_2} = (x_2^T PC1)PC1 + (mean^T PC2)PC2 + (mean^T PC3)PC3$$

$$\begin{aligned}\tilde{x_2} = ((20.82, 24.03, 3.5)^T(-0.694, -0.720, 0))(-0.694, -0.720, 0)\\ +(9.29375, 9.684999, 3.5)^T(-0.720, 0.694, 0))(-0.720, 0.694, 0)\\ +(9.29375, 9.684999, 3.5)^T(0, 0, 1))(0, 0, 1)\end{aligned}$$

$$\tilde{x_2} = (22.01345162, 22.88123278, 3.5)$$

$$\|x_2 - \tilde{x_2}\|_2^2 = ((x_2^T PC2) - (mean^T PC2))^2 + ((x_2^T PC3) - (mean^T PC3))^2$$

$$\|x_2 - \tilde{x_2}\|_2^2 = ((x_2 - mean)^T PC2)^2 + ((x_2 - mean)^T PC3)^2$$

$$(x_2 - mean) = (11.52625, 14.345001, 0)$$
$$(x_2 - mean)^T PC3 = 0$$

hence,

$$\|x_2 - \tilde{x_2}\|_2^2 = ((x_2 - mean)^T PC2)^2 = 2.744093940164124$$

3. (8 points) [LINEAR REGRESSION]

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

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

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: The error function in ridge regression is given by:

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

Where:

- t is the target vector of shape $(N, 1)$.
- ϕ is the data matrix of shape (N, D), where D is the number of features.
- w is the parameter vector of shape $(D, 1)$.
- λ is the regularization parameter.

The gradient of the error function with respect to w is given by:

$$\nabla E(w) = -\phi^T(t - \phi w) + \lambda w$$

The Hessian matrix of E is $(\phi^T\phi + \lambda I)$, which is positive semi-definite. This shows that the function $E(w)$ is convex, and any stationary point will be its minimizer if it exists.

To find the minimum, we set the gradient equal to zero and solve for w:

$$\phi^T(\phi w - t) + \lambda w = 0$$

Solving for w, we get:

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

Where:

- I is the identity matrix of size $D \times D$.

The matrix $(\lambda I + \phi^T\phi)$ is invertible for any positive value of λ, ensuring numerical stability.

This represents the solution for the parameter vector w that minimizes the ridge regression error function.

To show the inverse of $(\phi^T\phi + \lambda I)$ exists, we can use the Singular Value Decomposition (SVD):

$$\phi^T\phi = V\Sigma V^T$$

where all the singular values are $\geqslant 0$, and V is an orthogonal matrix.

$$(\phi^T\phi + \lambda I) = (V\Sigma V^T + \lambda V I V^T)$$

$$(\phi^T\phi + \lambda I) = V(\Sigma + \lambda I)V^T$$

Here, $(\Sigma + \lambda I)$ is a diagonal matrix, and since $\lambda > 0$, all the diagonal values of $(\Sigma + \lambda I)$ are strictly greater than 0. This implies the inverse of $(\Sigma + \lambda I)$ exists, and we can write:

$$(\phi^T \phi + \lambda I)^{-1} = (V(\Sigma + \lambda I)V^T)^{-1}$$

$$(\phi^T \phi + \lambda I)^{-1} = (V^T(\Sigma + \lambda I)^{-1}V)$$

Hence, done.

(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 $\lambda = 1$)?

Solution:

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

we know the minimizers of w of $E(w) = \frac{1}{2}||Xw - t||^2$ is the solution of the system,

$$(X^T X)W = X^T t$$

$$(X^T X) = \begin{bmatrix} 5 & -10 \\ -10 & 20 \end{bmatrix}$$

$$X^T t = \begin{bmatrix} -5 \\ 10 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -10 \\ -10 & 20 \end{bmatrix} W = \begin{bmatrix} -5 \\ 10 \end{bmatrix}$$

$$R_2 \leftarrow R_2 + 2R_1$$

$$\begin{bmatrix} 5 & -10 \\ 0 & 0 \end{bmatrix} W = \begin{bmatrix} -5 \\ 0 \end{bmatrix}$$

$$R_1 \leftarrow \frac{R_1}{5}$$

$$\begin{bmatrix} 1 & -2 \\ 0 & 0 \end{bmatrix} W = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

solution of the system is $\{(w_1, w_2) : w_1 - 2W_2 + 1 = 0, W_1, w_2 \in \mathbb{R}\}$ or,

$$\begin{bmatrix} -1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} 2 \\ 1 \end{bmatrix} ; \alpha \in \mathbb{R}$$

Now, we have to find a solution that minimises the norm of W, i.e,

$$min||W||_2^2 \text{ subject to } w_1 - 2w_2 = -1$$

$$min||W||_2^2 \text{ subject to } \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} = -1$$

using lagrange multiplier,

$$f(w) = W^T W + \lambda(W^T \begin{bmatrix} 1 \\ -2 \end{bmatrix} + 1)$$

$$\nabla f(w) = 2W + \lambda \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$W = \frac{\lambda}{2}\begin{bmatrix}1\\-2\end{bmatrix}$$

$$W^{\mathsf{T}}\begin{bmatrix}1\\-2\end{bmatrix} = -1$$

$$\frac{\lambda}{2}\begin{bmatrix}1 & -2\end{bmatrix}\begin{bmatrix}1\\-2\end{bmatrix} = -1$$

$$\lambda = \frac{-2}{5}$$

$$W = \frac{-1}{5}\begin{bmatrix}1\\-2\end{bmatrix}$$

$$= \begin{bmatrix}-0.2\\0.4\end{bmatrix}$$

is the stationary point of f(w), by geometry we can easily see that it is indeed global minimum.

Minimizer of $\tilde{E}(w)$ (assuming $\lambda = 1$)

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

$$= (I + X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}t$$

$$= \begin{bmatrix}6 & -10\\-10 & 21\end{bmatrix}^{-1}\begin{bmatrix}-5\\10\end{bmatrix}$$

$$= \frac{1}{26}\begin{bmatrix}21 & 10\\10 & 6\end{bmatrix}^{-1}\begin{bmatrix}-5\\10\end{bmatrix}$$

$$= \frac{1}{26}\begin{bmatrix}-5\\10\end{bmatrix}$$

$$= \begin{bmatrix}-0.19231\\0.38462\end{bmatrix}$$

In summary, the solution from smallest norm and ridge regression are very close to each other.

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 PCA. 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.

   (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.