CS550 Machine Learning: Assignment 1

Aryan Maurya
12240280

August 31, 2024

Contents

1 Solution 1 : Finding PCA for the dataset

Steps Involved in finding PCA:

1. Standardization
2. Finding Covariance Matrix
3. Eigenvalue Decomposition
4. Normalizing Eigenvectors
5. Projection on Principal Components (PC)

Given data:

Feature	**Example 01**	**Example 02**	**Example 03**	**Example 04**
x	2	6	10	14
y	5	4	11	14

Table 1: Data for PCA

1. Standardizing the given data:

For standardizing the given data, we use the following formula:

$$X' = \frac{x-\mu}{\sigma}$$

where,

- μ is the mean of the feature
- σ is the standard deviation of the feature

Formula for Mean:

$$\text{Mean} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Formula for Standard Deviation for sample:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2}$$

Feature	Example 01	Example 02	Example 03	Example 04
z_x	-1.34	-0.45	0.45	1.34
z_y	-0.84	-1.08	0.60	1.32

Table 2: Standardized Data

2. Finding Covariance Matrix:

To find the covariance matrix for the given data, we use the following formula:

$$\text{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

where,

- $\bar{X}$ is the mean of X
- $\bar{Y}$ is the mean of Y
- n is the number of data points in the data set

Covariance Matrix:

$$\begin{pmatrix} \text{Cov}(X,X) & \text{Cov}(Y,X) \\ \text{Cov}(X,Y) & \text{Cov}(Y,Y) \end{pmatrix} = \begin{pmatrix} 1.332 & 1.216 \\ 1.216 & 1.324 \end{pmatrix}$$

3. Eigenvalue Decomposition:

We need to find the eigen decomposition of the covariance matrix. The eigenvalue equation is:

$$(\mathbf{C} - \lambda\mathbf{I})\mathbf{v} = 0$$

where,

- $\mathbf{C}$ is the covariance matrix
- λ is the eigenvalue
- $\mathbf{v}$ is the eigenvector
- $\mathbf{I}$ is the identity matrix

Finding the Eigenvalues:

$$\begin{vmatrix} 1.332 - \lambda & 1.216 \\ 1.216 & 1.324 - \lambda \end{vmatrix} = 0$$

The characteristic equation formed is:

$$\lambda^2 - 2.656\lambda + 0.28 = 0$$

Solving this equation, we get:

$$\lambda_1 = 2.54, \quad \lambda_2 = 0.11$$

Corresponding Eigenvectors:

$$\mathbf{v}_1 = (0.708, -0.706), \quad \mathbf{v}_2 = (0.706, 0.708)$$

4. Normalizing Eigenvectors:

To normalize the eigenvectors, divide each component of the vector by its magnitude. After normalization:

$$\mathbf{n}_1 = (0.708, -0.706), \quad \mathbf{n}_2 = (0.706, 0.708)$$

(Note: The vectors remain the same after normalization.)

5. Projection on PC:

To transform our data set to 1D, we take the dot product between the principal component and the data points matrix.

Since we want to move to 1D space, we select the eigenvector corresponding to the highest eigenvalue ($\lambda_1 = 2.54$).

Principal Component 1:

$$\mathbf{PC}_1 = (0.708, -0.706)$$

After performing the dot product, our data set is transformed to 1D space:

$$\mathbf{PC}_1 : -0.355, 0.443, -0.104, 0.0169$$

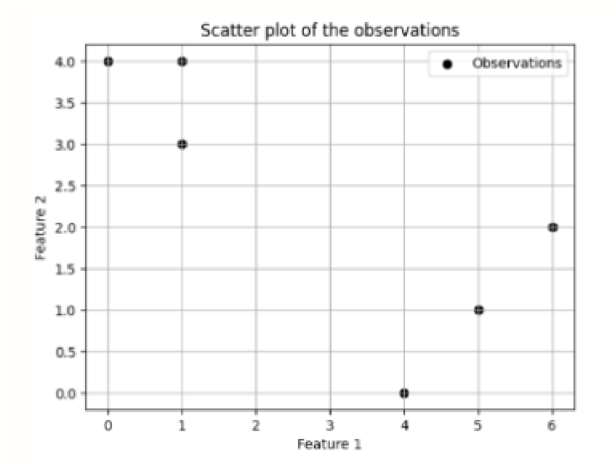
Conclusion: The axis on which maximum variance lies serves as a better principal component. Therefore, $\mathbf{PC}_1$ is more significant and must be used for further analysis as it explains more variance along the corresponding eigenvector.

2 Solution 2 : K-Means Clustering

Given Observation Table:

Observation	Feature 1	Feature 2
1	1	4
2	1	3
3	0	4
4	5	1
5	6	2
6	4	0

a) Plotting the given observation:



b) Randomly Assigning the Cluster Labels:

Observation	Feature 1	Feature 2	Initial Cluster
1	1	4	C1
2	1	3	C1
3	0	4	C2
4	5	1	C2
5	6	2	C1
6	4	0	C2

Let the two clusters be: C1 and C2

c) Computing the Centroids for each cluster:

To compute the centroid of each cluster, we calculate the mean of the observations in each cluster.

Centroid of C1:

C1: Observations: (1, 4), (1, 3), (6, 2)

$$\text{Mean in x-axis: } \frac{1+1+6}{3} = 2.67$$

$$\text{Mean in y-axis: } \frac{4+3+2}{3} = 3.00$$

$$\textbf{Centroid of C1: } (2.67, 3.00)$$

Similarly, we compute the centroid for the second cluster C2:
Centroid of C2:

$$\text{C2: Observations: (0, 4), (5, 1), (4, 0)}$$

$$\text{Mean in x-axis: } \frac{0+5+4}{3} = 3.00$$

$$\text{Mean in y-axis: } \frac{4+1+0}{3} = 1.67$$

$$\textbf{Centroid of C2: } (3.00, 1.67)$$

d & e) Assigning each observation to the closest centroid:

Now, we find the distance of each point from the given two centroids of the two clusters.
For finding the distance, we use the Euclidean formula for distance:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

For visibility convenience, I have made a table for the distances calculated and also listed the updated centroid for each observation.

Observation	Distance to C1	Distance to C2	Updated Centroid
(1, 4)	1.95	3.07	C1
(1, 3)	1.67	2.40	C1
(0, 4)	2.85	3.80	C1
(5, 1)	3.07	2.11	C2
(6, 2)	3.47	3.02	C2
(4, 0)	3.28	1.95	C2

Observations which changed their cluster:

- (0,4): From C2 to C1
- (6,2): From C1 to C2

2. Second iteration:

Updated values of centroids:

$$\text{C1: } (0.67, 3.67), \quad \text{C2: } (5.00, 1.00)$$

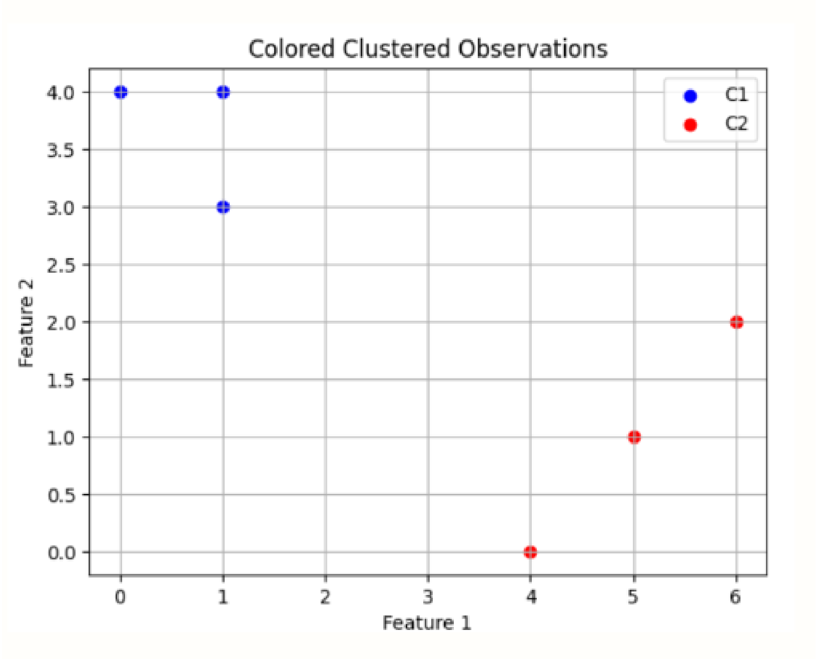
Again computing the distance from centroids and assigning updated centroids (if any):

Observation	**Distance to C1**	**Distance to C2**	**Closest Centroid**
(1, 4)	0.61	5.07	C1
(1, 3)	1.00	4.47	C1
(0, 4)	0.67	5.09	C1
(5, 1)	5.07	0.00	C2
(6, 2)	5.39	1.41	C2
(4, 0)	4.61	1.63	C2

This time, we see no change in the initial centroids assigned to each of the observations. Hence, we have obtained our clusters.

f) Plotting the clusters:

In your plot from (a), color the observations according to the cluster labels obtained.



3 Solution 3 : Performing LDA

a) Apply Linear Discriminant Analysis (LDA) for dimensionality reduction

Steps to apply LDA on a Dataset:

1. Find the mean for each of the given classes:

Mean of the classes are given below:

$$\mu_{X1} = \left(\frac{4+2+2+3+4}{5}, \frac{1+4+3+6+4}{5}\right) = (3, 3.6)$$

$$\mu_{X2} = \left(\frac{9+6+9+8+10}{5}, \frac{10+8+5+7+8}{5}\right) = (8.4, 7.6)$$

2. Computing within-class scatter matrix:

For computing the scatter matrix, we first calculate the covariance matrix for each class.

$$\mathbf{C} = \sum_{x\in\omega_1} \frac{(x-\mu_1)(x-\mu_1)^T}{\text{N-1}}$$

Finding covariance matrix for X_1:

$$C_1 = \left[(4,1)^T \cdot (4,1) + (2,4)^T \cdot (2,4) + (2,3)^T \cdot (2,3) + (3,6)^T \cdot (3,6) + (4,4)^T \cdot (4,4)\right]/4$$

Finding covariance matrix for X_2:

$$C_2 = \left[(9,10)^T \cdot (9,10) + (6,8)^T \cdot (6,8) + (9,5)^T \cdot (9,5) + (8,7)^T \cdot (8,7) + (10,8)^T \cdot (10,8)\right]/4$$

The resulting matrices are:

$$C_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 3.3 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 2.3 & 0.2 \\ 0.2 & 3.3 \end{pmatrix}$$

Now calculating **Within-class scatter matrix (S_w):**

$$S_w = C_1 + C_2 = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 6.6 \end{pmatrix}$$

3. Computing between-class scatter matrix (S_b):

For finding S_B, we use the formula:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

where,

- μ_1 is the mean of class X_1
- μ_2 is the mean of class X_2
- T denotes transpose of matrix

Given:

$$S_B = \left[\begin{pmatrix} 3 \\ 3.6 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right] \cdot \left[\begin{pmatrix} 3 \\ 3.6 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$

$$S_B = \begin{pmatrix} -5.4 \\ -4 \end{pmatrix} \cdot \begin{pmatrix} -5.4 & -4 \end{pmatrix}$$

Finally:

$$S_B = \begin{pmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{pmatrix}$$

4. Finding Eigenvalue and Eigenvector :

Equation we have to solve:

$$S_w^{-1} S_B \mathbf{v} = \lambda \mathbf{v}$$

Calculating S_w^{-1} : Now we have to calculate S_w^{-1}, since it is 2x2 matrix it is very easy to calculate the inverse which is as follows:

$$S_w^{-1} = \begin{pmatrix} \frac{6.6}{21.69} & \frac{0.3}{21.69} \\ \frac{0.3}{21.69} & \frac{3.3}{21.69} \end{pmatrix}$$

$$S_B = \begin{pmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{pmatrix}$$

Now, we have both S_w^{-1} and S_B. We can now begin our finding for eigenvectors and eigenvalues.

$$(S_w^{-1} S_B - \lambda \mathbf{I})\mathbf{v} = 0$$

$$(S_w^{-1} S_B - \lambda \mathbf{I}) = \begin{pmatrix} 9.178 - \lambda & 6.806 \\ 6.990 & 5.166 - \lambda \end{pmatrix} = 0$$

On solving this we get :

λ_1 = 14.348
λ_2 = -0.011

Calculating Eigenvectors: After calculations, eigenvectors are coming out to be :

$$v_1 = \begin{pmatrix}0.79590224 & -0.59518266\end{pmatrix}$$
$$v_2 = \begin{pmatrix}0.60542516 & 0.80359044\end{pmatrix}$$

since, λ_1 is greater, we will use v_1 for transformation.

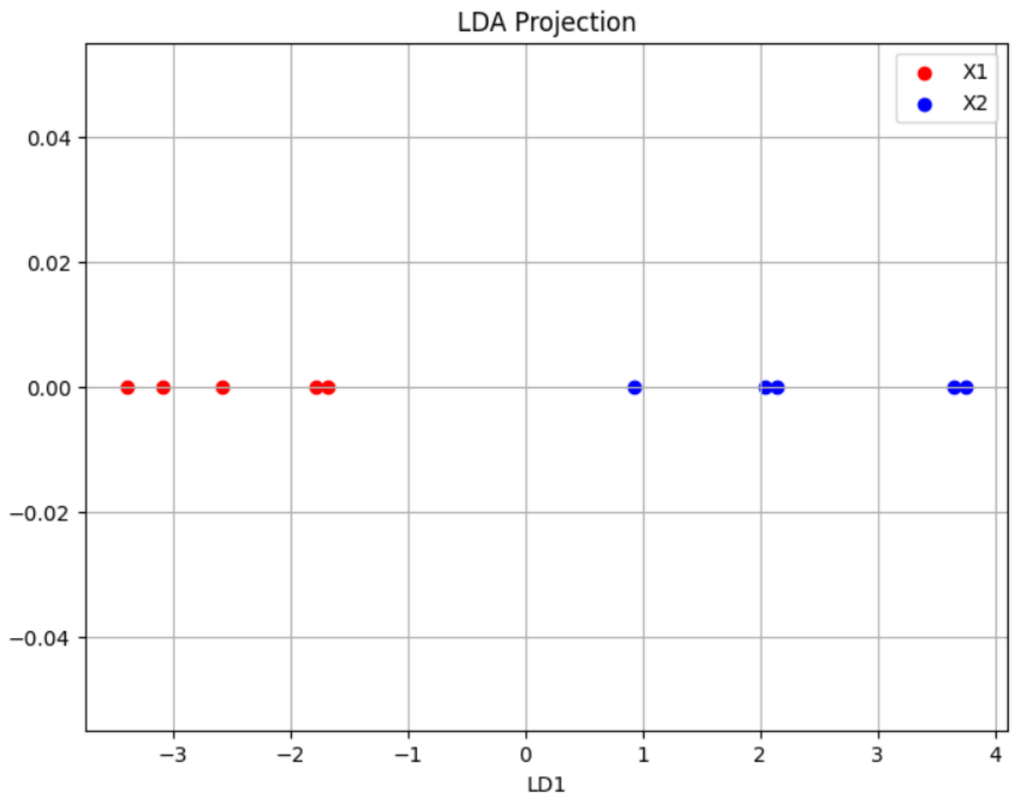
Transforming data : For this we have to take dot product between the eigenvector and the observations.

$$LDA_T = \mathbf{x}_i \cdot v_1$$

Transformed data :

$$-2.5807 \quad -3.0862 \quad -3.3871 \quad -1.7805 \quad -1.6782 \quad 3.6470 \quad 0.9332 \quad 2.1428 \quad 2.0404 \quad 3.7493$$

b) Plotting the graph



c) Writing advantages, disadvantages and applications of Linear Discriminant Analysis(LDA) :

Advantages :

- **Dimensionality Reduction:** Reduces the number of features while retaining the most important information.
- **Class Separability:** Enhances class separability by maximizing the ratio of between-class variance to within-class variance.

Disadvantages :

- **Linear relationship:** Works well if the their is linear relationship between data points, but there are many data-sets which has non-linear relations as well.
- **Normal Distribution :** LDA assumes that the data is in normal distribution, which in most case is not true as well. So in case of non-normal distribution LDA gives bad results.

Applications :

- **Image Compression:** It is helpful in reducing the dimensions of images which then uses less storage and maintains the class separability as well.
- **Face Recognition:** It can be used to reduce the dimension as well as retain the differentiable features in faces.

4 Solution 4 : Wine Quality Data Set

Code part is in the below given link for the ipynb file : CS550-ML Notebook

Why we use min-max scaling ?

Scaling is used because if in our dataset there are different scales of data. Let's say we have salary(range of Lakhs) and age(0-110). We can see how much is the scale difference. This may make the algorithms we are using to give more weightage to salary column and very less weightage to age.

To reduce this type of anomaly, we use scaling. This helps the algorithms to treat every column with equal weight.

5 Solution 5 : Model parameters and Model hyperparameters

a) Difference between model parameters and model Hyper-parameters

Model Parameters : These are the parameters which a model has after training, i.e. these parameters defines the ability to predict the values, which model has learned after being trained on the training data.

Hyper-parameters : These are the parameters which are set even before the model is trained onto the training data. Many algorithms have some parameters based on which they perform some actions, and if you change the parameter the algorithms will give different result.
It controls the training process as the complete algorithm works, based on hyper-parameters.

b) What is meant by hyperparameter tuning

In this we change the hyper-parameters of the algorithms in such a way that we get the optimal result from the algorithm on our data-set.
As I have told earlier, many algorithms have some hyper-parameters associated with them, which can be tuned such that it gives the max performance for our data-set.

c) Name some common hyperparameters used in clustering algorithms

As we have studied in the class :
There are hyper-parameters for clustering algorithms as well which is discussed below :

- **DBSCAN:** We have to define ϵ and minPs to define the maximum distance for a point to be considered in the same cluster and the minimum number of such points for a point to be considered as a core point.
- **K-Means Clustering** In K-Means clustering as well we have **k** as a hyper-parameter which defines the number of clusters for a given data-set

Because, changing these parameters changes the output of the algorithms as well and are given before the start of the training of the model, so they are considered as a hyper-parameter.

6 Solution 6 : Performing ICA

a) Using ICA to Estimate the Mixing Matrix and the source signal

$$\mathbf{X}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix}$$

where $\mathbf{X}(t)$ is the observed signal matrix. Mixing process equation :

$$\mathbf{X}(t) = \mathbf{A}\mathbf{s}(t)$$

Steps for estimating the mixing matrix:

1. Mean Centering the Data

For mean centering the data, we have to subtract mean from each observed signal:

$$\mathbf{X}_{\text{centered}}(t) = \mathbf{X}(t) - \mu$$

where,

$$\mu = \frac{1}{T}\sum_{t=1}^{T}\mathbf{X}(t)$$

2. Transforming the data such that it has unit variance (i.e. Whitening the Data)

a. Computing the Covariance Matrix
We have to first calculate the covariance matrix $\mathbf{C}$ of the centered data:
It's formula is given below :

$$\mathbf{C} = \frac{1}{T}\mathbf{X}\text{centered}\mathbf{X}_{\text{centered}}^T$$

where,

- $X_centered$ is the matrix containing all $X(t)$
- $X_centered^T$ is the transpose of the matrix containing all $X(t)$

b. Finding Eigenvalues and eigenvectors :
Now, we will find eigenvalues and eigenvectors for the covariance matrix:

$$\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

where:

- $\mathbf{E}$ is eigenvectors.
- $\mathbf{D}$ is diagonal eigenvalues matrix.

c. Applying the transformation

Equation for the whitening matrix is as follows :

$$\mathbf{W} = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$$

Transforming the mean centered data:

$$\mathbf{Z}(t) = \mathbf{W} \cdot \mathbf{X}\text{centered}(t)$$

3. Estimating independent components using ICA

Initializing the Weight Matrix

Initialize the weight matrix $\mathbf{W}$ randomly. Each row of $\mathbf{W}$ will be updated iteratively.

Maximizing the Non-Gaussianity Measure

ICA is used to maximize the component's non-Gaussianity.
non-Gaussianity $J(\mathbf{w})$ is defined as:

$$J(\mathbf{w}) = E[g(\mathbf{w}^\top \mathbf{Z}(t))]$$

where g is a non-linear function.
let's take it to be tanh(.).

$$J(\mathbf{w}) = E[tanh(\mathbf{w}^\top \mathbf{Z}(t))]$$

Update Rule

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \eta \left(E[tanh(\mathbf{w}^\top \mathbf{Z}(t))\mathbf{Z}(t)] - E[1 - tanh^2(\mathbf{w}^\top \mathbf{Z}(t))]\mathbf{w} \right)$$

where η is the learning rate.

To normalize the updated vector, we now have to divide it by it's norm. Equation is showed below :

$$\mathbf{w}_{\text{new}} = \frac{\mathbf{w}_{\text{new}}}{\|\mathbf{w}_{\text{new}}\|}$$

Repeat the process until the value stops changing.

Extracting the independent source signals

For extracting the independent source signals, we have to use the following equation:

$$\mathbf{s}(t) = \mathbf{W}\mathbf{Z}(t)$$

4. Estimating the Mixing Matrix

Now, since we have got all the ingredients to estimate the mixing matrix, use the following equation to do the same:

$$\mathbf{A} = \mathbf{X}(t)\hat{\mathbf{s}}(t)^\top (\hat{\mathbf{s}}(t)\hat{\mathbf{s}}(t)^\top)^{-1}$$

where:

- $\hat{\mathbf{s}}(t)$ is source matrix.
- $(\hat{\mathbf{s}}(t)\hat{\mathbf{s}}(t)^\top)^{-1}$ is the inverse of the covariance matrix of the estimated sources.

b) Discussion on challenges and the implications for recovering the original source signals:

- **Losing the Information:** Since a singular mixing matrix cannot span the entire space of observed signals, Hence, only a part of it will be recovered.Because of which we will get only a subset of the source signals, not all.
- **Uniqueness and Stability Issues:** Since, **A** is a singular matrix,Hence, it leads to unreliable source signals estimates. Since the inverse of matrix is not unique, hence, source signals are not recovered uniquely.

Implications:

- **Partial Recovery of sources:** Recovery can be made only for the number of independent sources equal to the rank of the matrix.
- **Discrepancy in Results:** Incorrect estimation of sources is one of the implications of this which will result in not getting the perfect source signals.