# PCA

Dimension Reduction-

- In pattern recognition, Dimension Reduction is defined as-

    -It is a process of converting a **data set having vast dimensions into a data set with lesser dimensions.**

- It ensures that the **converted data set conveys similar information conci**sely.

Example-

- Consider the following example-
- The following graph shows two **dimensions x1 and x2.**
- x1 represents the measurement of **several objects in cm**.
- x2 represents the measurement of **several objects in inches**.
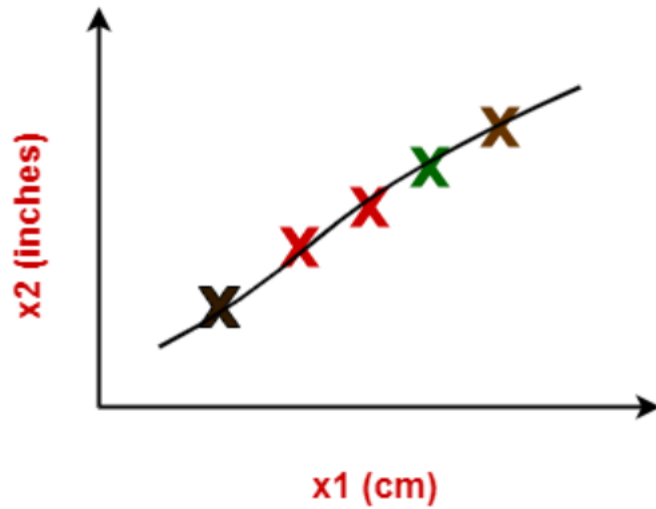
In machine learning,
- Using both these dimensions convey **similar information.**
- Also, they introduce a **lot of noise in the system**.
- So, it is better to use **just one dimension.**

Using dimension reduction techniques-
- We convert the **dimensions of data from 2 dimensions (x1 and x2) to 1 dimension (z1)**.
- It makes the data relatively **easier to explain**.

Benefits-

Dimension reduction offers several benefits such as-

- It compresses the **data and thus reduces the storage space** requirements.
- It reduces the **time required for computation** since less dimensions require less computation.
- It eliminates the **redundant features**.
- It improves **the model performance.**

Dimension Reduction Techniques-

- The two popular and well-known dimension reduction techniques are-

Dimension Reduction Techniques

Principal Component Analysis
(PCA)

Fisher Linear Discriminant Analysis
(LDA)
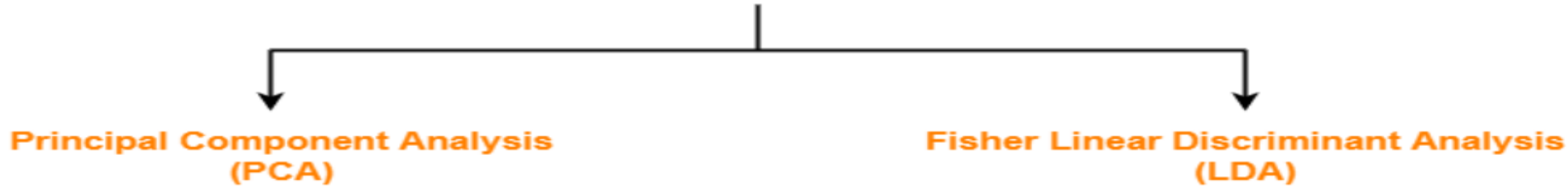
1. Principal Component Analysis (PCA)
2. Fisher Linear Discriminant Analysis (LDA)

Principal Component Analysis-

- Principal Component Analysis is a well-known **dimension reduction technique.**
- It transforms the variables into a **new set of variables called as principal components.**
- These principal components are **linear combination of original variables** and are orthogonal.
- The first principal component **accounts for most of the possible variation** of original data.
- The second principal component does its **best to capture the variance in the data.**
- There can be only two principal components for a **two-dimensional data set.**

PCA Algorithm-

- The steps involved in PCA Algorithm are as follows-

Step-01: Get data.

Step-02: Compute the **mean vector** (μ).

Step-03: Subtract **mean from the given data**.

Step-04: Calculate the **covariance matrix**.

Step-05: Calculate the **eigen vectors and eigen values** of the covariance matrix.

Step-06: Choosing components and forming a **feature vector**.

Step-07: Deriving the **new data set**.

Problem-01:

Given data = { 2, 3, 4, 5, 6, 7 ; 1, 5, 3, 6, 7, 8 }.

Compute the principal component using PCA Algorithm.

OR

OR

Consider the two dimensional patterns (2, 1), (3, 5), (4, 3), (5, 6), (6, 7), (7, 8).

Compute the principal component using PCA Algorithm.

Compute the principal component of following data-

CLASS 1

OR

X = 2 , 3 , 4

Y = 1 , 5 , 3

CLASS 2

X = 5 , 6 , 7

Y = 6 , 7 , 8

Solution-

We use the above discussed PCA Algorithm-

Step-01:

Get data.

The given feature vectors are-

- $x_1 = (2, 1)$
- $x_2 = (3, 5)$
- $x_3 = (4, 3)$
- $x_4 = (5, 6)$
- $x_5 = (6, 7)$
- $x_6 = (7, 8)$

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} \begin{bmatrix} 6 \\ 7 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Step-02:

Calculate the mean vector (μ).

Mean vector (μ)

$= ((2 + 3 + 4 + 5 + 6 + 7) / 6, (1 + 5 + 3 + 6 + 7 + 8) / 6)$

$= (4.5, 5)$

Thus,

$$\text{Mean vector } (\mu) = \begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$$

Step-03:

Subtract mean vector (μ) from the given feature vectors.

- $x_1 - \mu = (2 - 4.5, 1 - 5) = (-2.5, -4)$
- $x_2 - \mu = (3 - 4.5, 5 - 5) = (-1.5, 0)$
- $x_3 - \mu = (4 - 4.5, 3 - 5) = (-0.5, -2)$
- $x_4 - \mu = (5 - 4.5, 6 - 5) = (0.5, 1)$
- $x_5 - \mu = (6 - 4.5, 7 - 5) = (1.5, 2)$
- $x_6 - \mu = (7 - 4.5, 8 - 5) = (2.5, 3)$

Feature vectors (x_i) after subtracting mean vector (μ) are-

$$\begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$$

Step-04:

Calculate the covariance matrix.

Covariance matrix is given by-

$$\text{Covariance Matrix} = \frac{\sum (X_i - \mu)(X_i - \mu)^t}{n}$$

Now,

$$m_1 = (x_1 - \mu)(x_1 - \mu)^t = \begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -2.5 & -4 \end{bmatrix} = \begin{bmatrix} 6.25 & 10 \\ 10 & 16 \end{bmatrix}$$

$$m_2 = (x_2 - \mu)(x_2 - \mu)^t = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -1.5 & 0 \end{bmatrix} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}$$

$$m_3 = (x_3 - \mu)(x_3 - \mu)^t = \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} -0.5 & -2 \end{bmatrix} = \begin{bmatrix} 0.25 & 1 \\ 1 & 4 \end{bmatrix}$$

$$m_4 = (x_4 - \mu)(x_4 - \mu)^t \; = \; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \end{bmatrix} \; = \; \begin{bmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$m_5 = (x_5 - \mu)(x_5 - \mu)^t \; = \; \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 1.5 & 2 \end{bmatrix} \; = \; \begin{bmatrix} 2.25 & 3 \\ 3 & 4 \end{bmatrix}$$

$$m_6 = (x_6 - \mu)(x_6 - \mu)^t \; = \; \begin{bmatrix} 2.5 \\ 3 \end{bmatrix} \begin{bmatrix} 2.5 & 3 \end{bmatrix} \; = \; \begin{bmatrix} 6.25 & 7.5 \\ 7.5 & 9 \end{bmatrix}$$

Now,

Covariance matrix

$$= (m_1 + m_2 + m_3 + m_4 + m_5 + m_6) / 6$$

On adding the above matrices and dividing by 6, we get-

$$\text{Covariance Matrix} = \frac{1}{6} \begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix}$$

$$\text{Covariance Matrix} = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

Step-05:

Calculate the eigen values and eigen vectors of the covariance matrix.

λ is an eigen value for a matrix M if it is a solution of the characteristic equation |M − λI| = 0.

So, we have-

$$\begin{vmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{vmatrix} - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} = 0$$

$$\begin{vmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{vmatrix} = 0$$

From here,

$(2.92 - \lambda)(5.67 - \lambda) - (3.67 \times 3.67) = 0$

$16.56 - 2.92\lambda - 5.67\lambda + \lambda^2 - 13.47 = 0$

$\lambda^2 - 8.59\lambda + 3.09 = 0$

Solving this quadratic equation, we get λ = 8.22, 0.38

Thus, two eigen values are λ_1 = 8.22 and λ_2 = 0.38.

Clearly, the second eigen value is very small compared to the first eigen value.

So, the second eigen vector can be left out.

Eigen vector corresponding to the greatest eigen value is the principal component for the given data set.

So. we find the eigen vector corresponding to eigen value λ_1.

We use the following equation to find the eigen vector-

$$MX = \lambda X$$

where-

- M = Covariance Matrix
- X = Eigen vector
- λ = Eigen value

Substituting the values in the above equation, we get-

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} = 8.22 \begin{bmatrix} X1 \\ X2 \end{bmatrix}$$

Solving these, we get-

$2.92X_1 + 3.67X_2 = 8.22X_1$

$3.67X_1 + 5.67X_2 = 8.22X_2$

On simplification, we get-

$5.3X_1 = 3.67X_2$ ....(1)

$3.67X_1 = 2.55X_2$ ....(2)
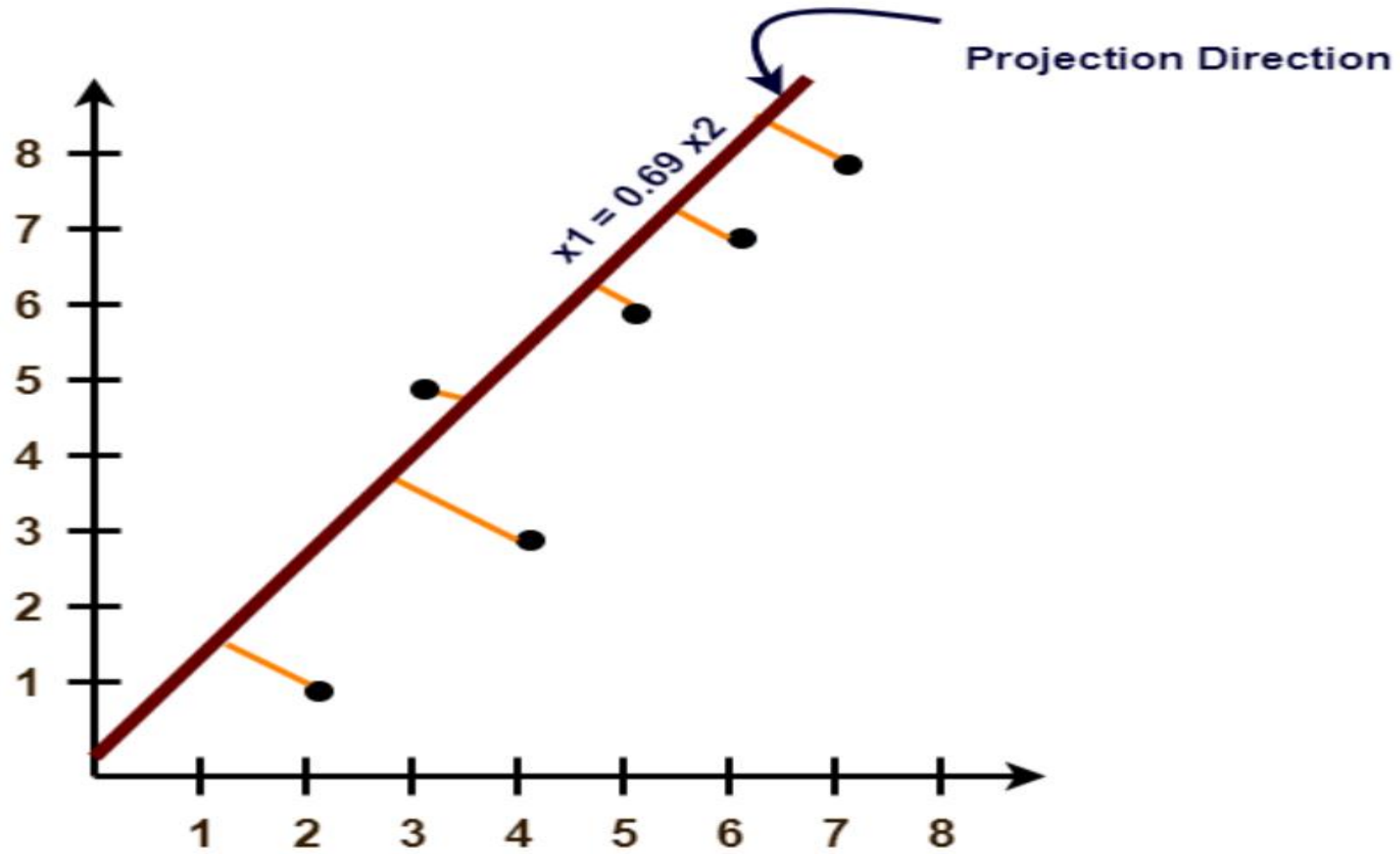
From (1) and (2), **$X_1 = 0.69X_2$**

From (2), the eigen vector is-

$$\textbf{Eigen Vector :} \quad \begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$

Thus, principal component for the given data set is-

$$\textbf{Principal Component :} \quad \begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$

Lastly, we project the data points onto the new subspace as-

Self-Organizing Map (SOM)

- **Purpose**: SOM is a type of artificial neural network used primarily for **dimensionality reduction, data visualization, and clustering**.

- It helps map **high-dimensional data into a lower-dimensional** (usually two-dimensional) grid while **preserving the topological properties** of the input space.

- **Structure**: SOMs consist of **a grid of nodes**, each with a **weight vector** of the **same dimension as the input data**. This grid can be 1D, 2D, or even 3D, but 2D grids are most common for visualization.

- SOM has two layers, one is the **Input layer** and the other one is the **Output layer.**

- The architecture of the Self Organizing Map with **two clusters and n input features of any sample is given below:**
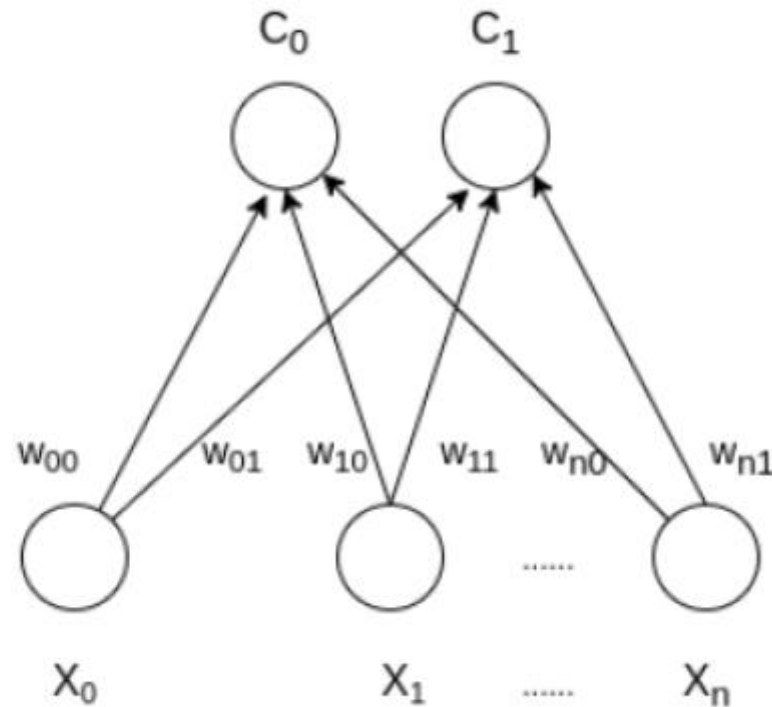
Process:

1.Initialization: Randomly initialize the weight vectors of the nodes.

2.Training:

- For each data point, find the node whose weight vector is closest to the data point. This node is **called the "Best Matching Unit" (BMU).**

- Update the **BMU and its neighboring nodes** to make them closer to the input vector.

- Repeat for all data points over multiple iterations, gradually reducing the **neighborhood size and learning rate.**

3.Output: After training, **similar data points are mapped to nearby nodes on the gri**d, enabling clustering and visualization.

- **Applications**: SOMs are commonly used in customer segmentation, anomaly detection, and visualizing complex datasets in a 2D space.