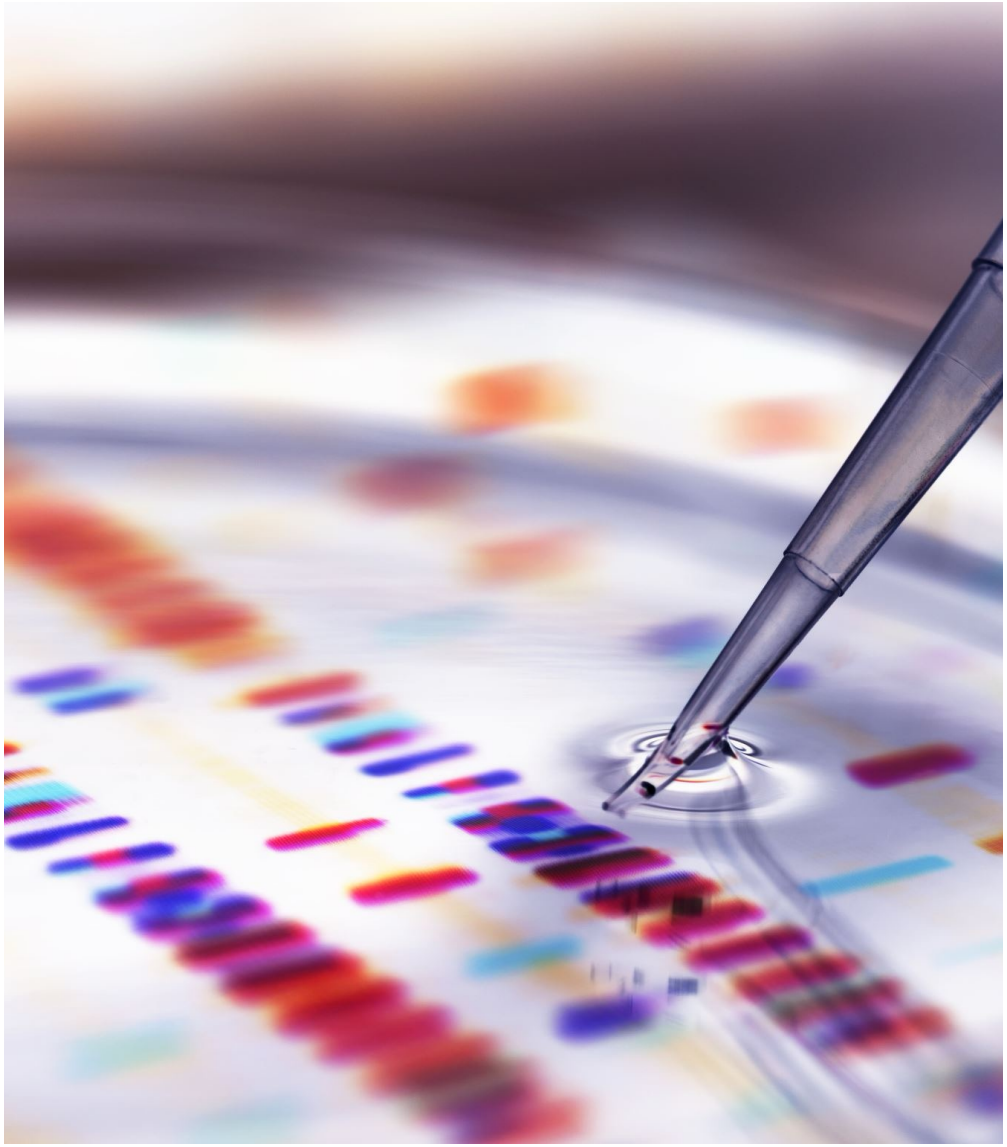


Linear Discriminant Analysis

K Kotecha

LDA vs. PCA

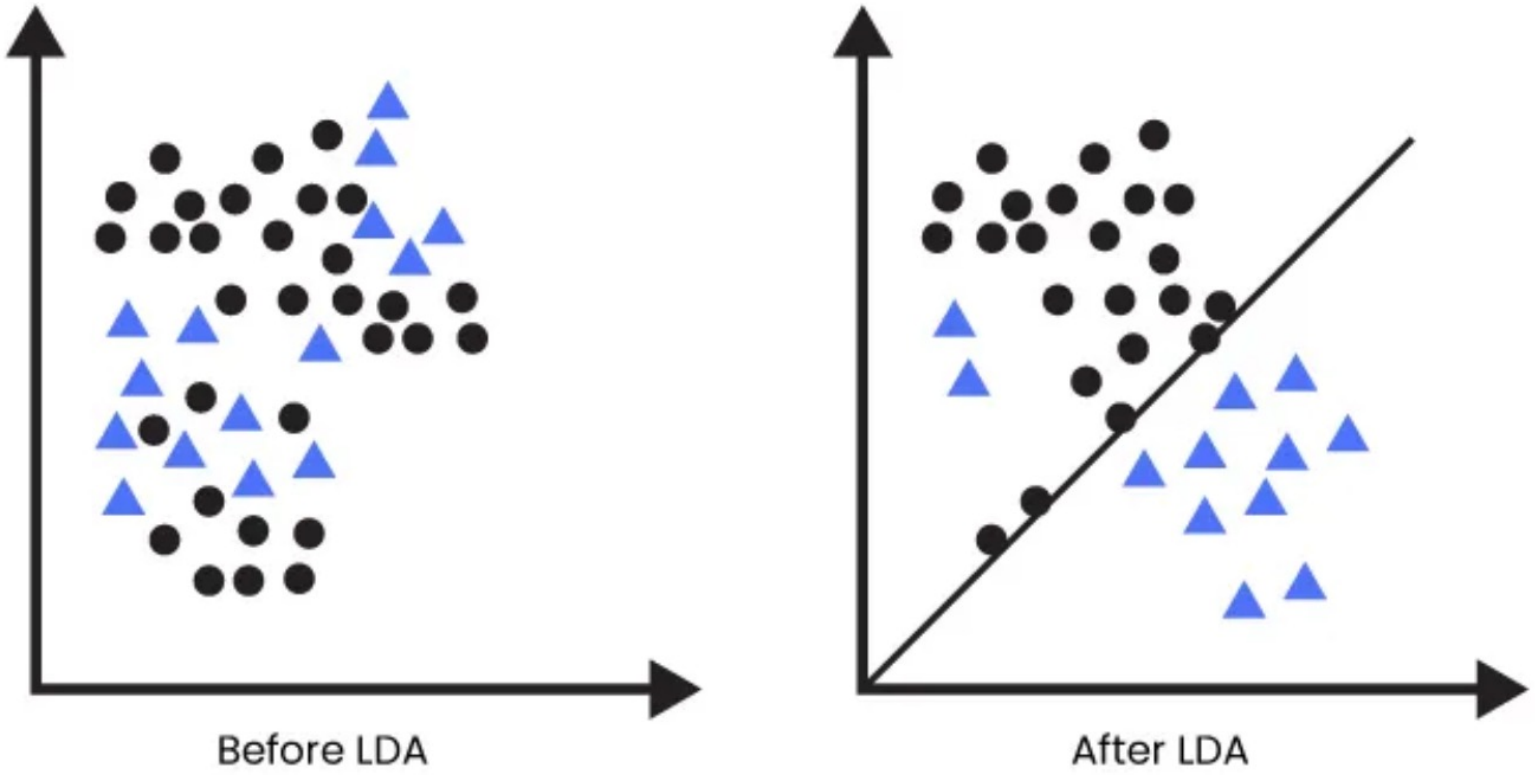
Linear discriminant analysis is very similar to PCA both look for linear combinations of the features which best explain the data.

The main difference is that the Linear discriminant analysis is a **supervised** dimensionality reduction technique that also achieves classification of the data simultaneously.

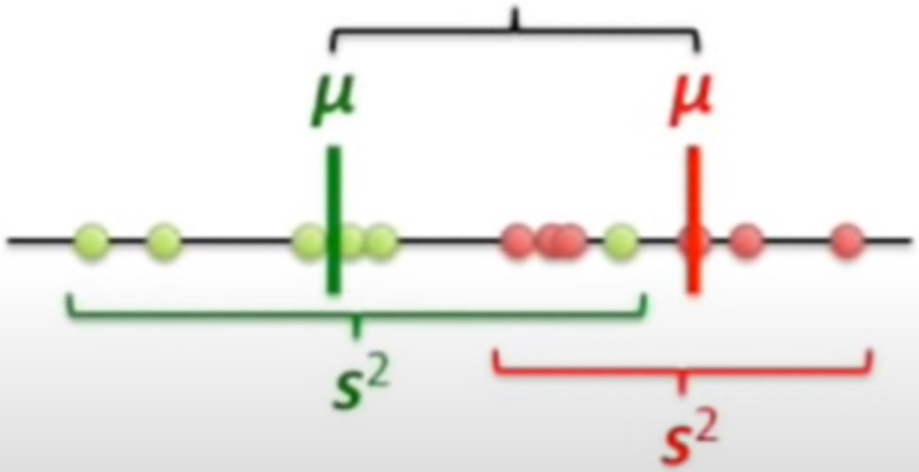
LDA Vs PCA

LDA focuses on finding a feature subspace that **maximizes the separability** between the groups.

PCA focuses on capturing the direction of **maximum variation** in the data set.


Before LDA
After LDA

1) Maximize the distance between means.



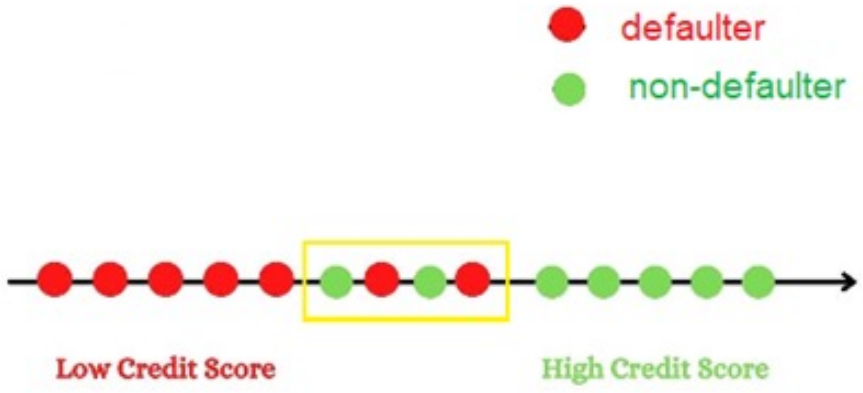
$$\frac{(\mu - \mu)^2}{s^2 + s^2}$$

2) Minimize the variation (which LDA calls "scatter" and is represented by s^2) within each category.

- Imagine you have a credit card loan dataset with a target label consisting of two classes defaulter and non-defaulter.

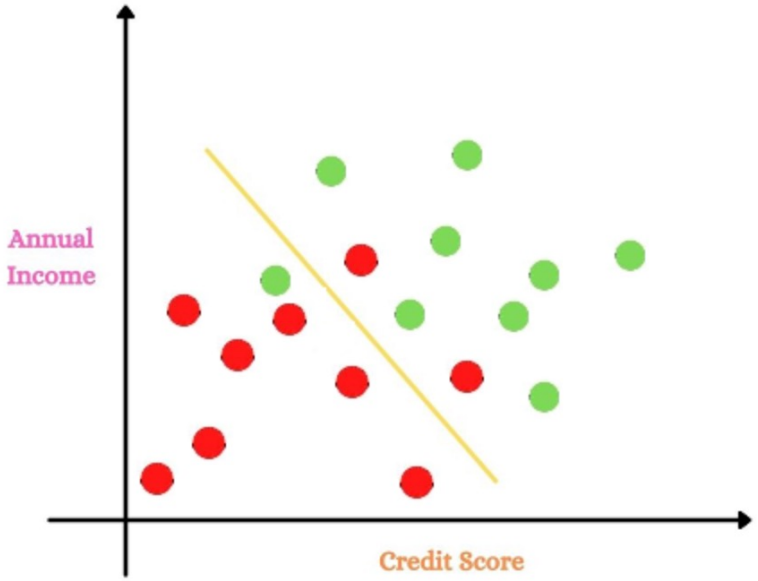
- **Class‘ 1’** is the **defaulter** and **class ‘0’** is the **non-defaulter**.

- When you have just **one attribute** say the **credit score** the graph would be a 1-D graph which is a number line.
-



Let's see what happens when we have two attributes, are we able to classify better?

Considering another attribute annual income along with the credit score.



Something interesting, we noticed is adding features we were able to reduce the number of overlapped points and distinguish better. But this becomes extremely difficult to visualize when we have high dimension dataset.

This is where the implementation of LDA plays a crucial role.

1.LDA uses information from both the attributes and projects the data onto the new axes.

2.It projects the data points in such a way that it satisfies the criteria of maximum separation between groups and minimum variation within groups simultaneously.

Step 1:

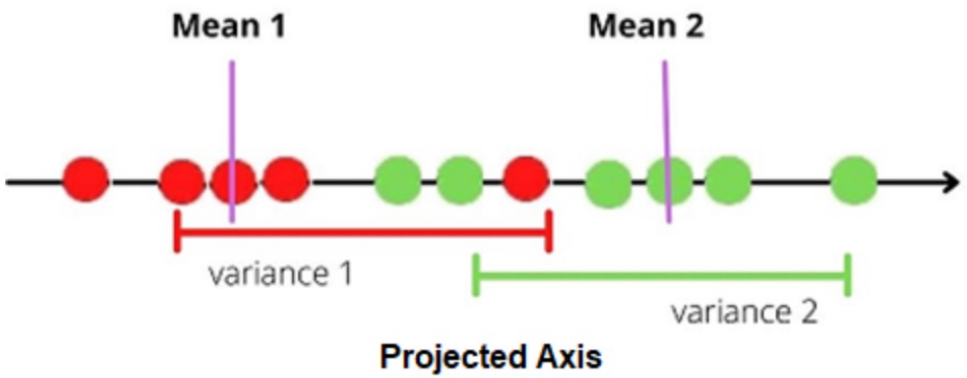
The projected points and the new axes

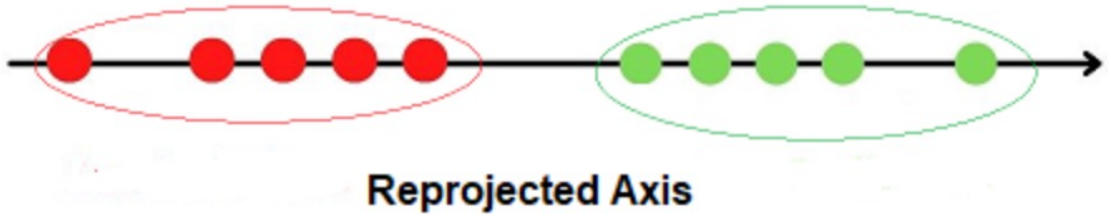
Step-2

Criterion LDA applies to the projected points is as follows.

1.It maximizes the distance between the means of each category.

2. It minimizes the variation or scatter within each category represented by s^2





Let the mean of the category 1 defaulter be mean 1 and mean 2 be the mean of the category non-defaulter.

1. Compute the d-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors ($\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$) and corresponding eigenvalues ($\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_d$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix $\boldsymbol{W}$ (where every column represents an eigenvector).
5. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}$ (where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

The iris dataset contains measurements for 150 iris flowers from three different species.

The three classes in the Iris dataset:

1. Iris-setosa (n=50)
2. Iris-versicolor (n=50)
3. Iris-virginica (n=50)

The four features of the Iris dataset:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

	sepal length in cm	sepal width in cm	petal length in cm	petal width in cm	class label
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

$$\boldsymbol{X} = \begin{bmatrix} x_{1_{\text{sepal length}}} & x_{1_{\text{sepal width}}} & x_{1_{\text{petal length}}} & x_{1_{\text{petal width}}} \\ x_{2_{\text{sepal length}}} & x_{2_{\text{sepal width}}} & x_{2_{\text{petal length}}} & x_{2_{\text{petal width}}} \\ \cdots & & & \\ x_{150_{\text{sepal length}}} & x_{150_{\text{sepal width}}} & x_{150_{\text{petal length}}} & x_{150_{\text{petal width}}} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \omega_{\text{setosa}} \\ \omega_{\text{setosa}} \\ \cdots \\ \omega_{\text{virginica}} \end{bmatrix}$$

Step 1: Computing the d-dimensional mean vectors

In this first step, we will start off with a simple computation of the mean vectors $\boldsymbol{m}_i$, $(i = 1, 2, 3)$ of the 3 different flower classes:

$$\boldsymbol{m}_i = \begin{bmatrix} \mu_{\omega_i}(\text{sepal length}) \\ \mu_{\omega_i}(\text{sepal width}) \\ \mu_{\omega_i}(\text{petal length}) \\ \mu_{\omega_i}(\text{petal width}) \end{bmatrix}, \quad \text{with} \quad i = 1, 2, 3$$

Step 2: Computing the Scatter Matrices

Now, we will compute the two *4x4*-dimensional matrices: The within-class and the between-class scatter matrix.

2.1 Within-class scatter matrix S_W

The **within-class scatter** matrix S_W is computed by the following equation:

$$S_W = \sum_{i=1}^{c} S_i$$

where

$$S_i = \sum_{\boldsymbol{x} \in D_i}^{n} (\boldsymbol{x} - \boldsymbol{m}_i)\ (\boldsymbol{x} - \boldsymbol{m}_i)^T$$

(scatter matrix for every class)

and $\boldsymbol{m}_i$ is the mean vector

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$

2.2 Between-class scatter matrix S_B

The **between-class scatter** matrix S_B is computed by the following equation:

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

where

$\boldsymbol{m}$ is the overall mean, and $\boldsymbol{m}_i$ and N_i are the sample mean and sizes of the respective classes.

Step 3: Solving the generalized eigenvalue problem for the matrix $S_W^{-1} S_B$

Next, we will solve the generalized eigenvalue problem for the matrix $S_W^{-1} S_B$ to obtain the linear discriminants.

A quick check that the eigenvector-eigenvalue calculation is correct and satisfy the equation:

$$\boldsymbol{Av} = \lambda \boldsymbol{v}$$

where

$\boldsymbol{A} = S_W^{-1} S_B$

$\boldsymbol{v} =$ Eigenvector

$\lambda =$ Eigenvalue

Step 4: Selecting linear discriminants for the new feature subspace

4.1. Sorting the eigenvectors by decreasing eigenvalues

4.2. Choosing *k* eigenvectors with the largest eigenvalues

After sorting the eigenpairs by decreasing eigenvalues, it is now time to construct our $d \times k$-dimensional eigenvector matrix $\boldsymbol{W}$ (here 4×2: based on the 2 most informative eigenpairs) and thereby reducing the initial 4-dimensional feature space into a 2-dimensional feature subspace.

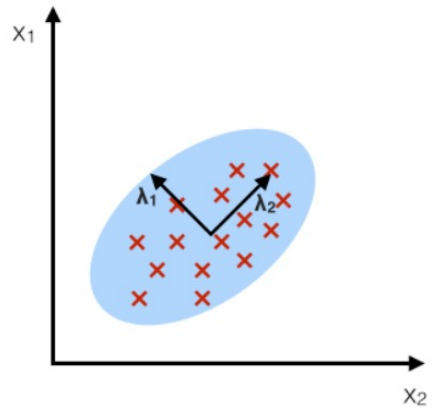
Step 5: Transforming the samples onto the new subspace

In the last step, we use the 4×2-dimensional matrix $\boldsymbol{W}$ that we just computed to transform our samples onto the new subspace via the equation

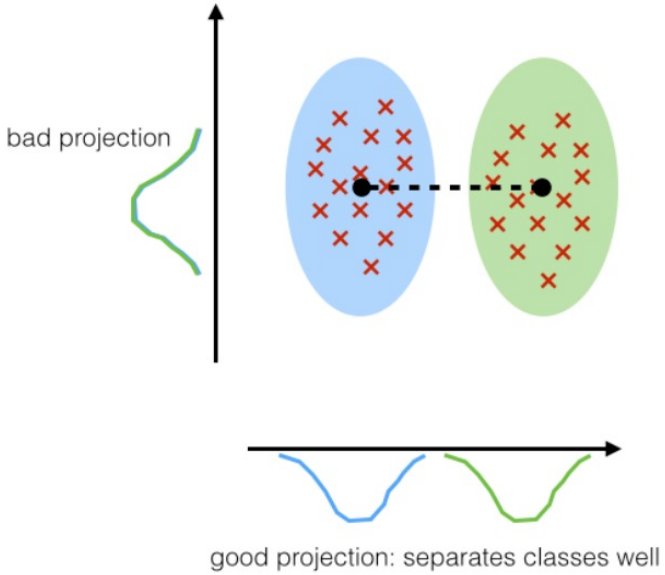
$$\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}.$$

(where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{Y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

PCA:
component axes that maximize the variance

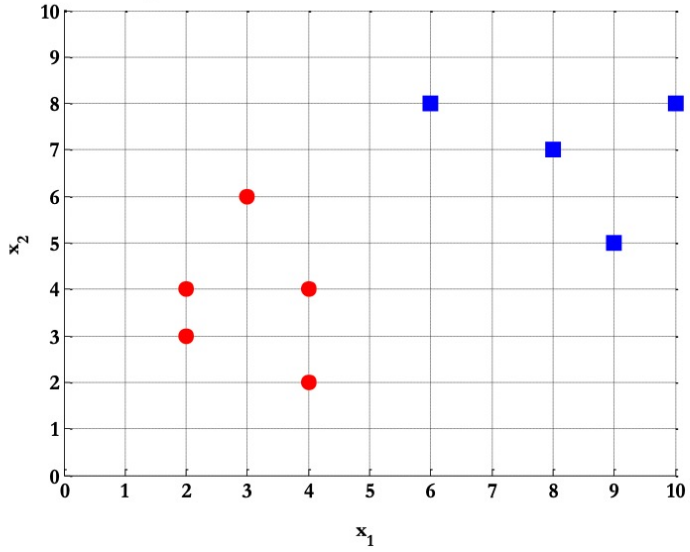


LDA:
maximizing the component axes for class-separation



LDA … Two Classes - Example

- Compute the Linear Discriminant projection for the following two-dimensional dataset.
  - Samples for class ω_1 : $\mathbf{X}_1=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$
  - Sample for class ω_2 : $\mathbf{X}_2=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$




```
% samples for class 1
X1 = [4,2;
      2,4;
      2,3;
      3,6;
      4,4];


% samples for class 2
X2 = [9,10;
      6,8;
      9,5;
      8,7;
      10,8];
```

LDA … Two Classes - Example

- The classes mean are :

$$\mu_1 = \frac{1}{N_1}\sum_{x\in\omega_1} x = \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right] = \begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2}\sum_{x\in\omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right] = \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

```
% class means
Mu1 = mean(X1)';
Mu2 = mean(X2)';
```

LDA … Two Classes - Example

- Covariance matrix of the first class:

$$S_1 = \sum_{x \in \omega_1} (x - \mu_1)(x - \mu_1)^T = \left[\begin{pmatrix} 4 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 2 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 3 \\ 6 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix}$$

```
% covariance matrix of the first class
S1 = cov(X1);
```

LDA … Two Classes - Example

- Covariance matrix of the second class:

$$S_2 = \sum_{x \in \omega_2} (x - \mu_2)(x - \mu_2)^T = \left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 6 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 8 \\ 7 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 10 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

```
% covariance matrix of the first class
S2 = cov(X2);
```

LDA … Two Classes - Example

- Within-class scatter matrix:

$$S_w = S_1 + S_2 = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix} + \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

$$= \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}$$

```
% within-class scatter matrix
Sw = S1 + S2 ;
```

LDA … Two Classes - Example

- Between-class scatter matrix:

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

$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]\left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$

$$= \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}\begin{pmatrix} -5.4 & -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix}$$

```
% between-class scatter matrix
SB = (Mu1-Mu2)*(Mu1-Mu2)';
```

LDA … Two Classes - Example

- The LDA projection is then obtained as the solution of the generalized eigen value problem

$$S_W^{-1} S_B w = \lambda w$$

$$\Rightarrow \left| S_W^{-1} S_B - \lambda I \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{pmatrix} \right|$$

$$= (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 \times 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

LDA … Two Classes - Example

```
% computing the LDA projection
invSw = inv(Sw);

invSw_by_SB = invSw * SB;

% getting the projection vector
[V,D] = eig(invSw_by_SB)

% the projection vector
W = V(:,1);
```

- Hence

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_1 = \underbrace{0}_{\lambda_1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

and

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \underbrace{12.2007}_{\lambda_2} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Thus;

$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \quad \textit{and} \quad w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*$$

- The optimal projection is the one that given maximum $\lambda = J(w)$

LDA … Two Classes - Example

Or directly;

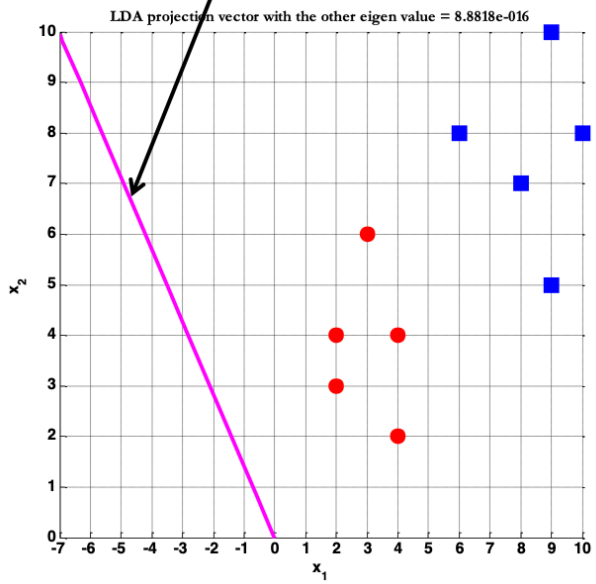
$$w^* = S_W^{-1}(\mu_1 - \mu_2) = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]$$

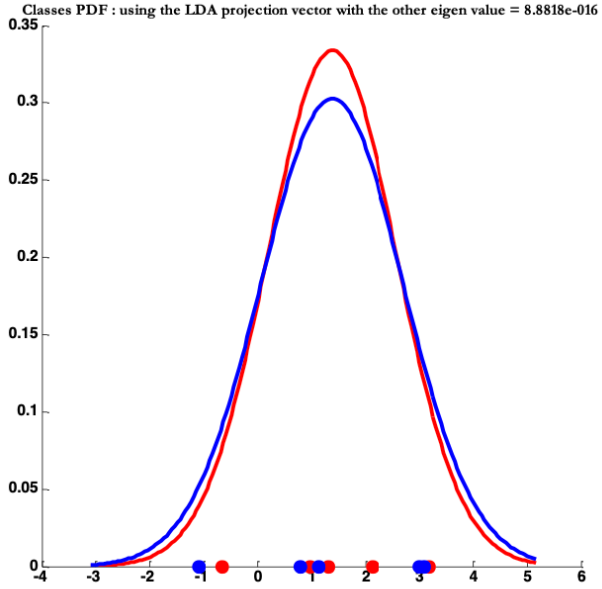
$$= \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$$

LDA - Projection

The projection vector corresponding to the **smallest** eigen value

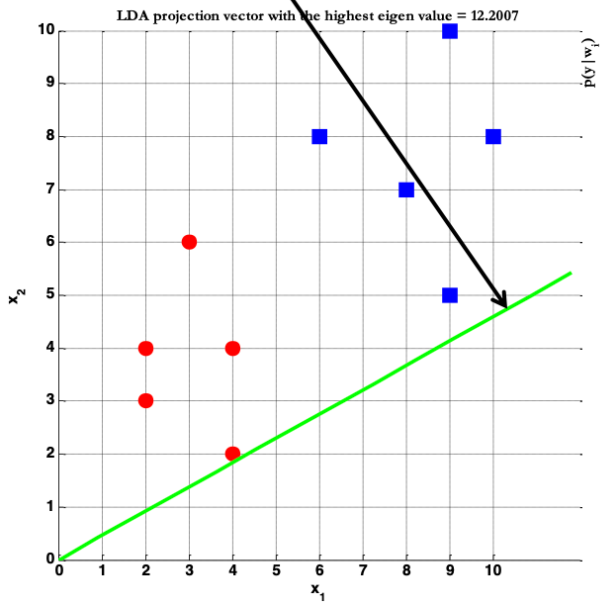


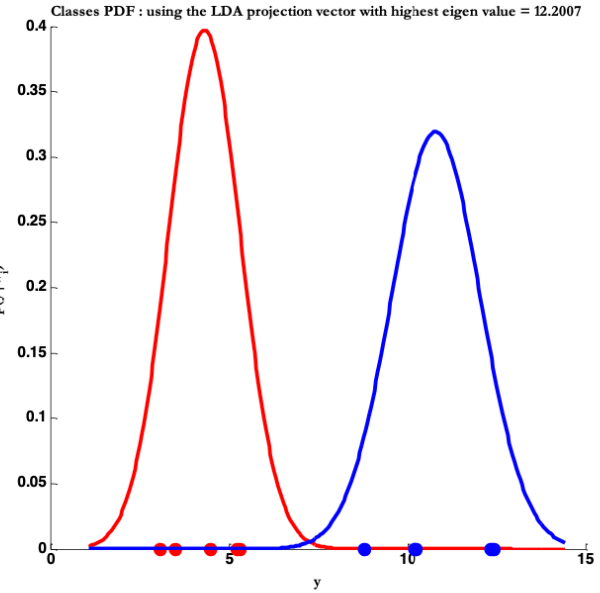


Using this vector leads to **bad separability** between the two classes

LDA - Projection

The projection vector corresponding to the **highest** eigen value





Using this vector leads to **good separability** between the two classes

Facial recognition

- In image processing (e.g., facial recognition), a high-resolution image may have thousands of features (pixels). LDA reduces these to a few key features that still differentiate between individuals' faces effectively, speeding up computation without losing classification accuracy.

Spam Email

In spam email detection, LDA extracts key features from text (like specific words) and reduces dimensionality, helping to separate spam and non-spam emails by identifying discriminative word patterns.

Fraud Detection

- **PCA**:

  Used for exploratory analysis, identifying clusters of transactions that might be anomalous, without focusing on whether they are fraud or not.

- **LDA**:

  Separates fraudulent and non-fraudulent transactions when labeled data is available, focusing on features (e.g., transaction amount, frequency) that best differentiate these classes.

Customer Segmentation

- PCA

  Groups customers into segments based on features like spending habits, age, and location, without focusing on predefined labels.

- LDA

  If labeled data is available (e.g., "high-value", "low-value" customers), LDA is used to classify and separate customers based on these categories.

Aspect	LDA	PCA
Objective	Maximizes **class separability** by preserving information that helps distinguish classes.	Maximizes **variance** in the dataset, irrespective of class labels.
Supervised/Unsupervised	Supervised: Requires class labels for computation.	Unsupervised: Does not require class labels.
Focus	Focuses on finding the axes that maximize separation between classes.	Focuses on finding the axes that capture the most overall variance in the data.
Applications	Best suited for **classification tasks**.	Best suited for **feature extraction** or when class labels are unavailable.


Thank you