Machine Learning & Pattern Recognition

Xin Xin(辛鑫)

xinxin@sdu.edu.cn

https://xinxin-me.github.io/

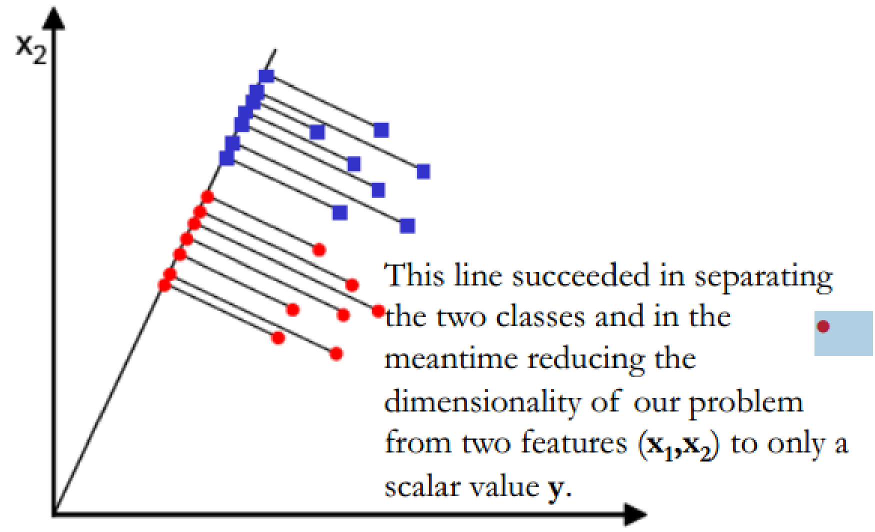
Supervised Feature Extraction

- **Linear Discriminant Analysis (LDA)**

Feature Extraction

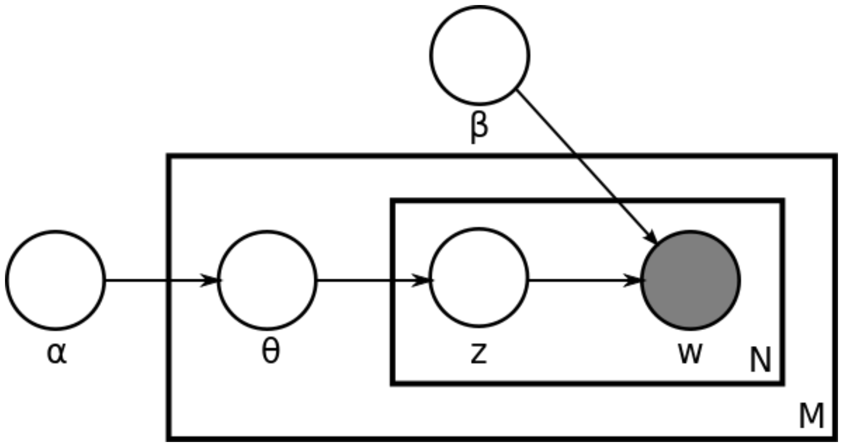
- Feature extraction (dimensionality reduction/feature reduction) refers to the mapping of the original **high-dimensional** data into a **low-dimensional** space.
- Criterion for feature reduction can be different based on different problem setting
  - ✓ Unsupervised setting: minimize the information loss
  - ✓ Supervised setting: maximize the class discrimination

Linear Discriminant Analysis



Linear Discriminant Analysis, a method to find a linear combination of features that separates two or more classes of objects.

Latent Dirichlet Allocation



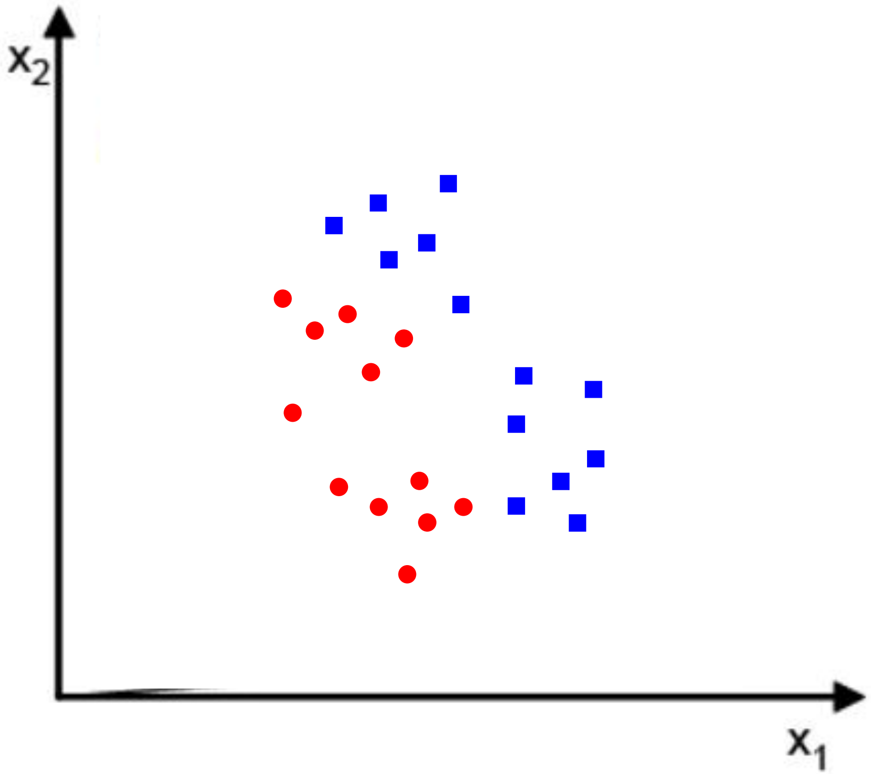
In natural language processing, latent Dirichlet allocation (LDA) is an example of a topic model.
https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

Linear Discriminant Analysis

- Linear Discriminant Analysis—2 Classes
- Linear Discriminant Analysis—C Classes

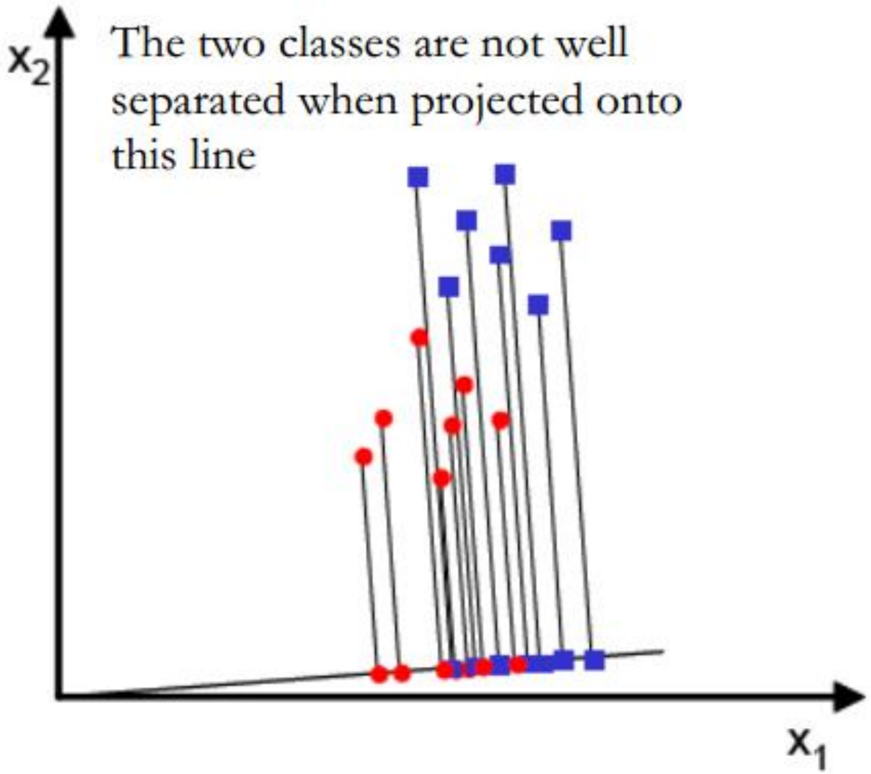
What is a Good Projection?

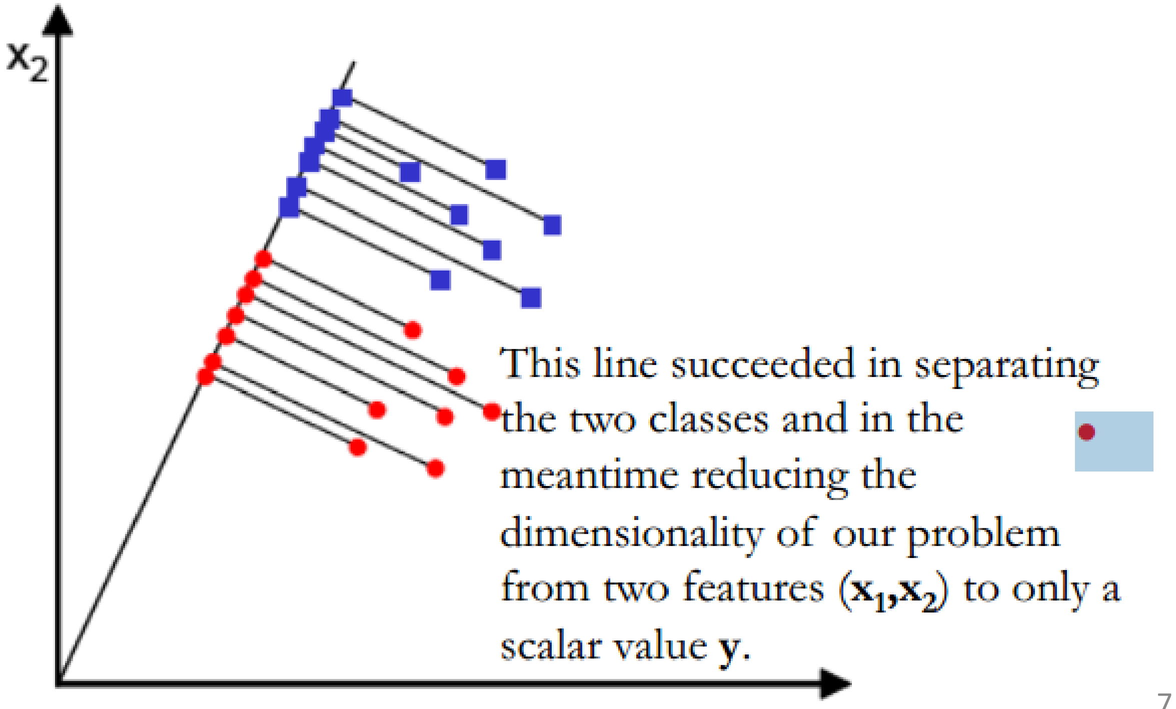
- Given a set of points (2-d) from two classes, we want to project them to a line that can well separate them.



What is a Good Projection?

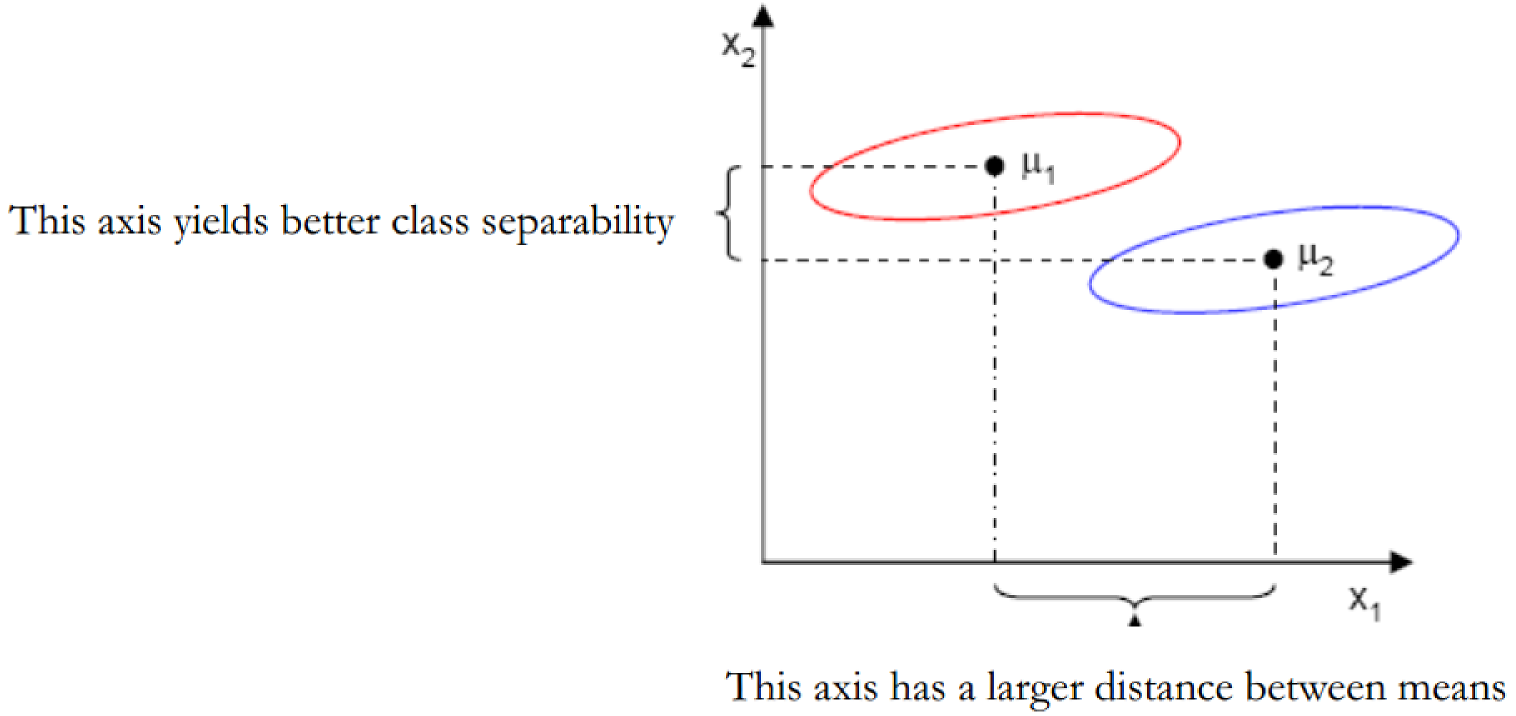
- What is a good criterion?
  - Maximize the between-class distance (means) Is it enough?





What is a Good Projection?

- What is a good criterion?
  - Maximize the between-class distance (means)
  - Minimize the within-class variability (scatter)



Linear Discriminant Analysis—Two Classes

- Assume we have d-dimensional samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, n_1 of which belong to C_1 and n_2 belong to C_2.
- We seek to obtain a transformation $\boldsymbol{\theta} \in \mathbb{R}^{d \times 1}$ that projects the samples $\boldsymbol{x}$ onto a line $(p = 1)$.

- $y_i = \boldsymbol{\theta}^T \boldsymbol{x}_i$, where $\boldsymbol{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{id} \end{bmatrix}$ and $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$

  where $\boldsymbol{\theta}$ is the projection vector used to project $\boldsymbol{x}$ to y.

Linear Discriminant Analysis—Two Classes

- The mean vector of each class in $\boldsymbol{x}$ and y feature space is:

$$\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{x \in C_i} \boldsymbol{x} \qquad \tilde{\mu}_i = \frac{1}{n_i}\sum_{y \in C_i} y = \frac{1}{n_i}\sum_{x \in C_i} \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{\theta}^T \boldsymbol{\mu}_i$$

- Projecting $\boldsymbol{x}$ to y will lead to projecting the mean of $\boldsymbol{x}$ to the mean of y.

- Choose $\boldsymbol{\theta}$ to maximize the distance between the projected means:

$$J_1(\boldsymbol{\theta}) = (\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (\boldsymbol{\theta}^T \boldsymbol{\mu}_1 - \boldsymbol{\theta}^T \boldsymbol{\mu}_2)^2 = \boldsymbol{\theta}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\theta} = \boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta}$$

Between-class scatter (类间散度矩阵): $\boldsymbol{S}_b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ $\quad \mathbf{S}_b \in \mathbb{R}^{d \times d}$

Linear Discriminant Analysis—Two Classes

- Meanwhile, to achieve a small variance within each class, i.e., minimizing the class overlap,
- We define the total within-class variance as $s_1^2 + s_2^2$.
- We want to choose $\boldsymbol{\theta}$ to minimize

$$\boxed{s_k^2 = \sum_{y \in C_k} (y - \tilde{\mu}_k)^2}$$

$$J_2(\boldsymbol{\theta}) = \sum_{y \in C_1} (y - \tilde{\mu}_1)^2 + \sum_{y \in C_2} (y - \tilde{\mu}_2)^2 = \boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta}$$

Within-class scatter (类内散度矩阵):

$$\boldsymbol{S}_w = \sum_{\boldsymbol{x} \in C_1} (\boldsymbol{x} - \boldsymbol{\mu}_1)(\boldsymbol{x} - \boldsymbol{\mu}_1)^T + \sum_{\boldsymbol{x} \in C_2} (\boldsymbol{x} - \boldsymbol{\mu}_2)(\boldsymbol{x} - \boldsymbol{\mu}_2)^T \qquad \mathbf{S}_w \in \mathbb{R}^{d \times d}$$

Linear Discriminant Analysis—Two Classes

- We can finally express the Fisher criterion in terms of $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$:

If $\boldsymbol{\theta}$ is one solution, then $\alpha\boldsymbol{\theta}$ would also be a solution.

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{J_1(\boldsymbol{\theta})}{J_2(\boldsymbol{\theta})} = \frac{\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta}}{\boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta}}$$

$$\min_{\boldsymbol{\theta}} -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta}$$

$$\text{s.t. } \boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} = 1$$

- Let λ be a **Lagrange multiplier**

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} - 1)$$

Linear Discriminant Analysis—Two Classes

- Let λ be a **Lagrange multiplier**

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} - 1)$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\boldsymbol{S}_b \boldsymbol{\theta} + 2\lambda \boldsymbol{S}_w \boldsymbol{\theta} = 0 \qquad \Longrightarrow \qquad \boldsymbol{S}_b \boldsymbol{\theta} = \lambda \boldsymbol{S}_w \boldsymbol{\theta}$$

- $\boldsymbol{\theta}$: the eigenvectors of $\boldsymbol{S}_w^{-1} \boldsymbol{S}_b$, and λ is the corresponding eigenvalue.

- How to choose $\boldsymbol{\theta}$?

Linear Discriminant Analysis—Two Classes

- Let λ be a **Lagrange multiplier**

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} - 1)$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\boldsymbol{S}_b \boldsymbol{\theta} + 2\lambda \boldsymbol{S}_w \boldsymbol{\theta} = 0 \qquad \Longrightarrow \qquad \boldsymbol{S}_b \boldsymbol{\theta} = \lambda \boldsymbol{S}_w \boldsymbol{\theta}$$

- Remember the objective function

$$\begin{cases} \min\limits_{\boldsymbol{\theta}} -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} \\ \text{s.t. } \boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} = 1 \end{cases} \xRightarrow{\boldsymbol{S}_b \boldsymbol{\theta}^* = \lambda \boldsymbol{S}_w \boldsymbol{\theta}^*} \qquad -\boldsymbol{\theta}^{*T} \boldsymbol{S}_b \boldsymbol{\theta}^* = -\lambda \boldsymbol{\theta}^{*T} \boldsymbol{S}_w \boldsymbol{\theta}^* = -\lambda$$

- How to choose? The eigenvector corresponds to the largest eigenvalue.

Linear Discriminant Analysis—Two Classes

- Let λ be a **Lagrange multiplier**

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} + \lambda(\boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} - 1)$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\boldsymbol{S}_b\boldsymbol{\theta} + 2\lambda \boldsymbol{S}_w \boldsymbol{\theta} = 0 \qquad \Longrightarrow \qquad \boldsymbol{S}_b\boldsymbol{\theta} = \lambda \boldsymbol{S}_w \boldsymbol{\theta}$$

- **Alternatively**, as $\boldsymbol{S}_b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$, $\quad \boldsymbol{S}_b\boldsymbol{\theta} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\underbrace{\boxed{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\theta}}}_{\lambda_{\boldsymbol{\theta}}}$
- Let $\boldsymbol{S}_b\boldsymbol{\theta} = \lambda_{\boldsymbol{\theta}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ then $\lambda_{\boldsymbol{\theta}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = \lambda \boldsymbol{S}_w \boldsymbol{\theta}$
- The scale of $\boldsymbol{\theta}^*$ does not matter, only direction matters.

$$\boldsymbol{\theta}^* = \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

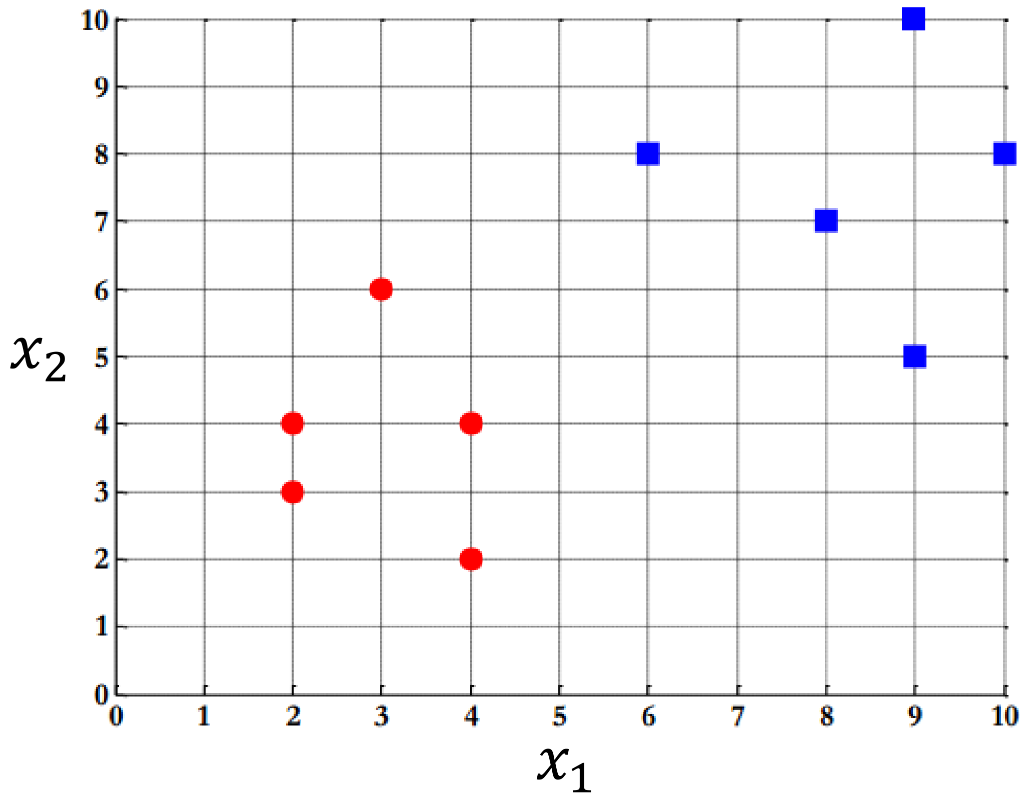
Linear Discriminant Analysis—Two Classes

- **Workflow of LDA for the binary classification**

1. Build $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ from the training set
2. Compute $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$
3. Compute $\boldsymbol{S}_w$
4. Compute $\boldsymbol{S}_w^{-1}$
5. Compute $\boldsymbol{\theta}^* = \boldsymbol{S}_w^{-1}\,(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
6. Given a testing sample, $y = {\boldsymbol{\theta}^*}^{\boldsymbol{T}}\boldsymbol{x}$
7. Set the threshold $\gamma = \frac{n_1{\boldsymbol{\theta}^*}^{\boldsymbol{T}}\boldsymbol{\mu}_1 + n_2{\boldsymbol{\theta}^*}^{\boldsymbol{T}}\boldsymbol{\mu}_2}{n_1+n_2}$.
8. Compare y with γ to determine the class.

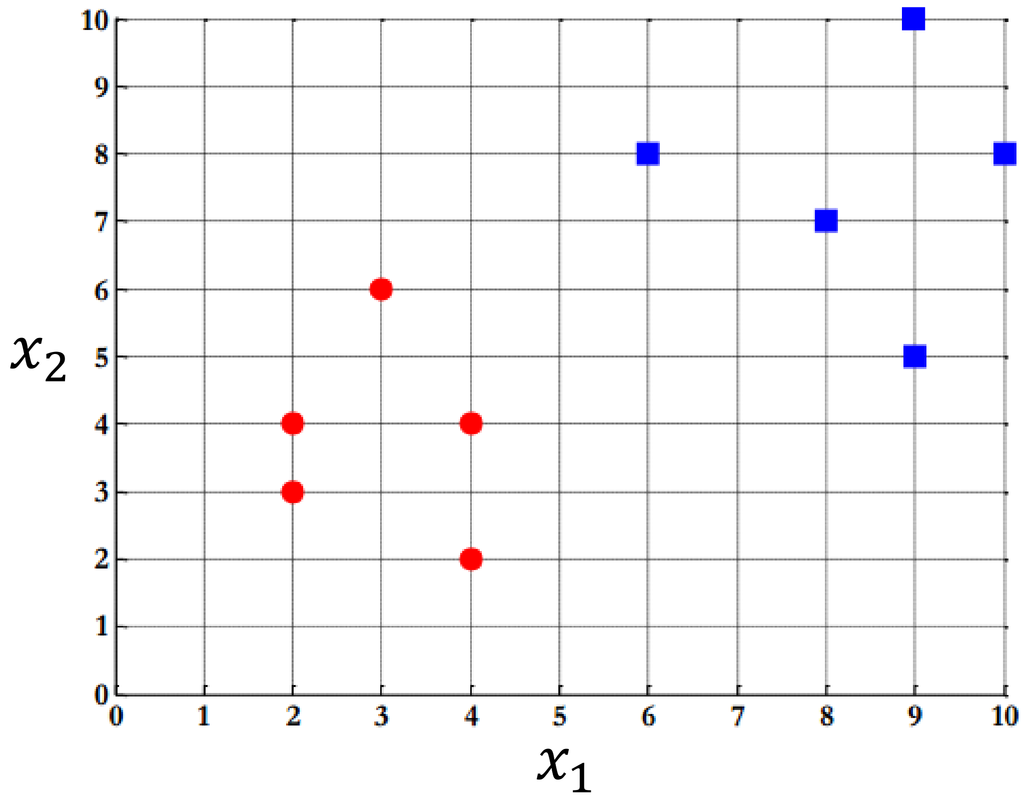
Example

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



Example

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



- Mean of each class:

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

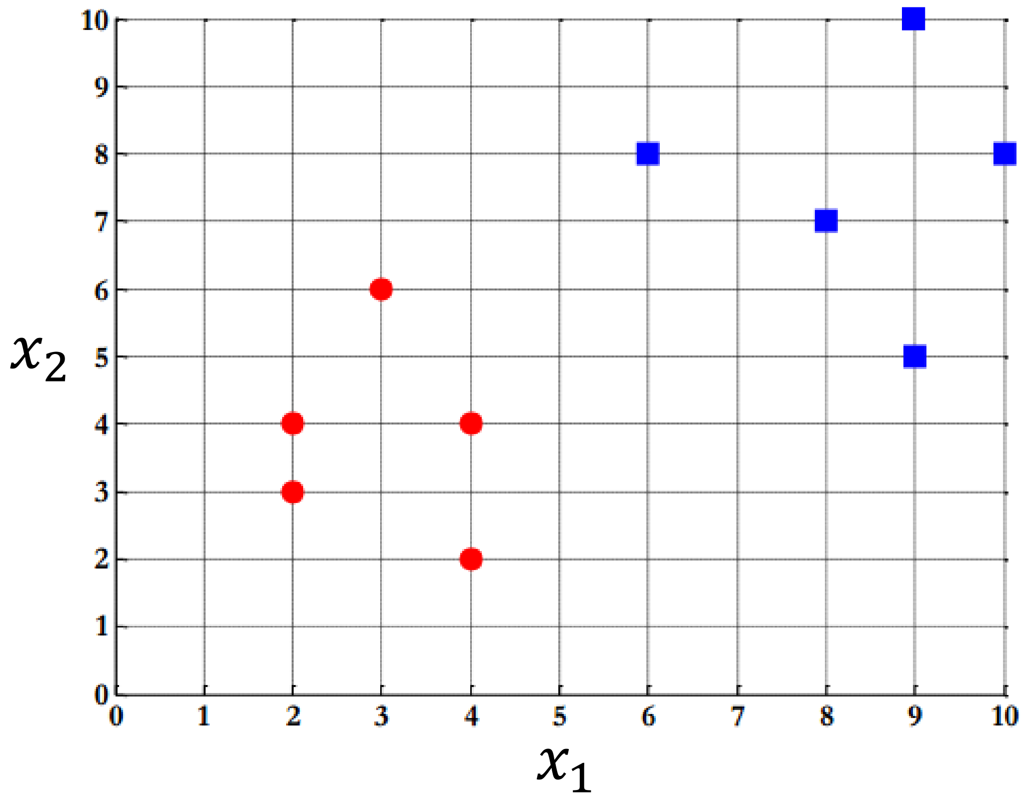
$$\mu_2 = \frac{1}{N_2} \sum_{x \in \omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix} + \begin{pmatrix}6\\8\end{pmatrix} + \begin{pmatrix}9\\5\end{pmatrix} + \begin{pmatrix}8\\7\end{pmatrix} + \begin{pmatrix}10\\8\end{pmatrix}\right] = \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Example

Compute the Linear Discriminant projection for the following two dimensional dataset.

– Samples for class $\boldsymbol{\omega}_1$: $\boldsymbol{X}_1 = (x_1, x_2) = \{(4,2), (2,4), (2,3), (3,6), (4,4)\}$

– Sample for class $\boldsymbol{\omega}_2$: $\boldsymbol{X}_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$

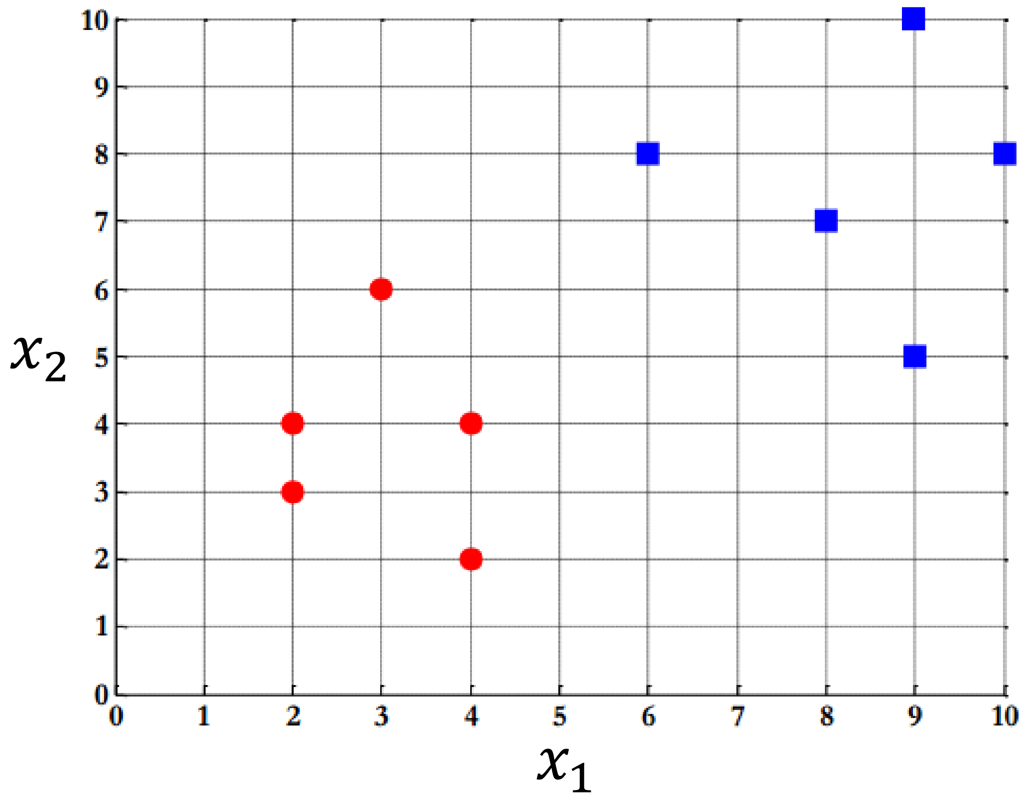


- Covariance matrix of the first class:

$$S_1 = \sum_{x \in \omega_1} (x - \mu_1)(x - \mu_1)^T = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix}$$

Example

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

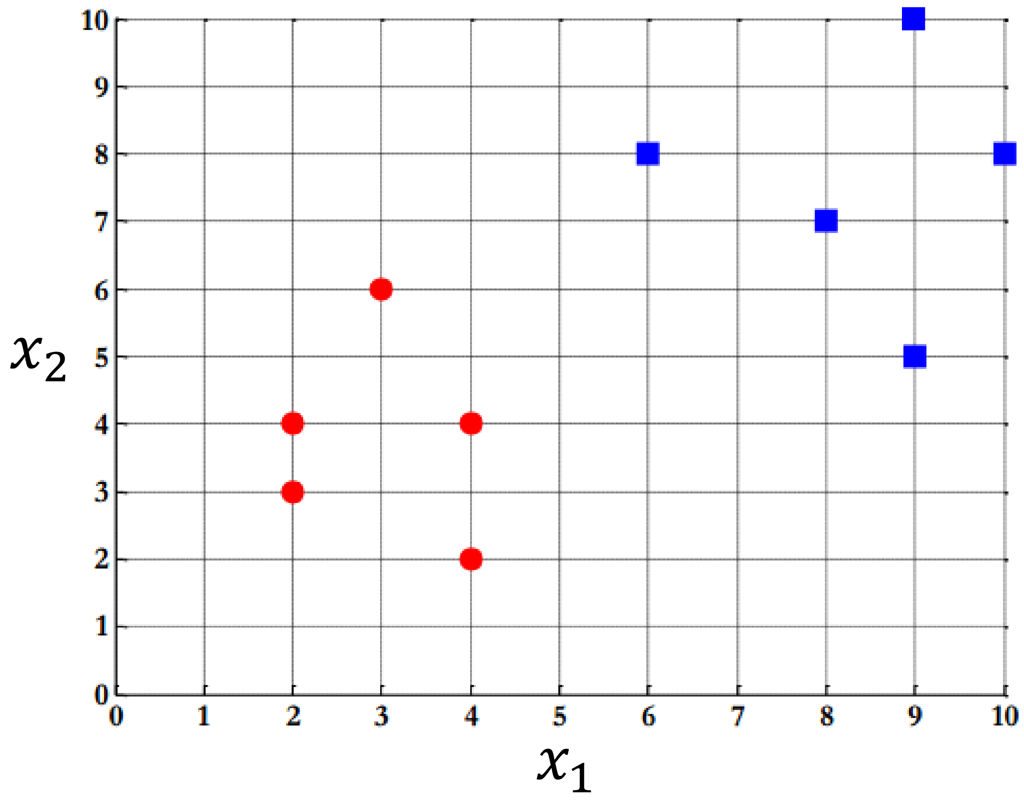


- Covariance matrix of the second class:

$$S_2 = \sum_{x \in \omega_2} (x - \mu_2)(x - \mu_2)^T = \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

Example

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

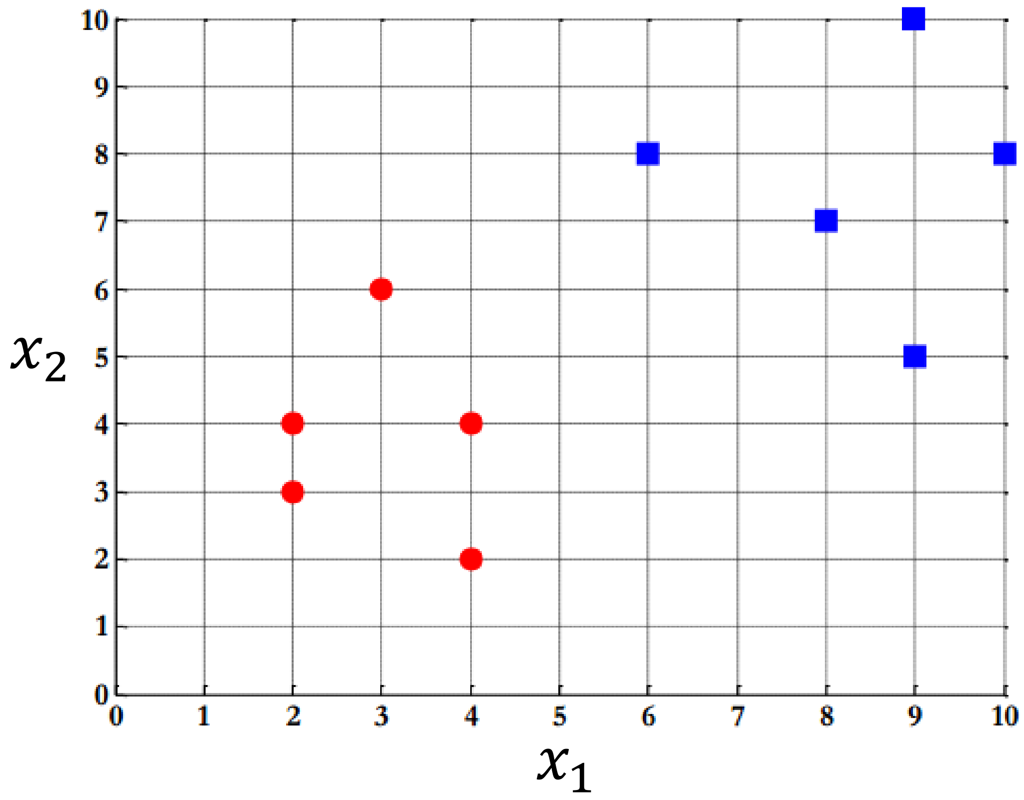


- Within-class scatter matrix:

$$S_w = S_1 + S_2 = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix} + \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$
$$= \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}$$

Example

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

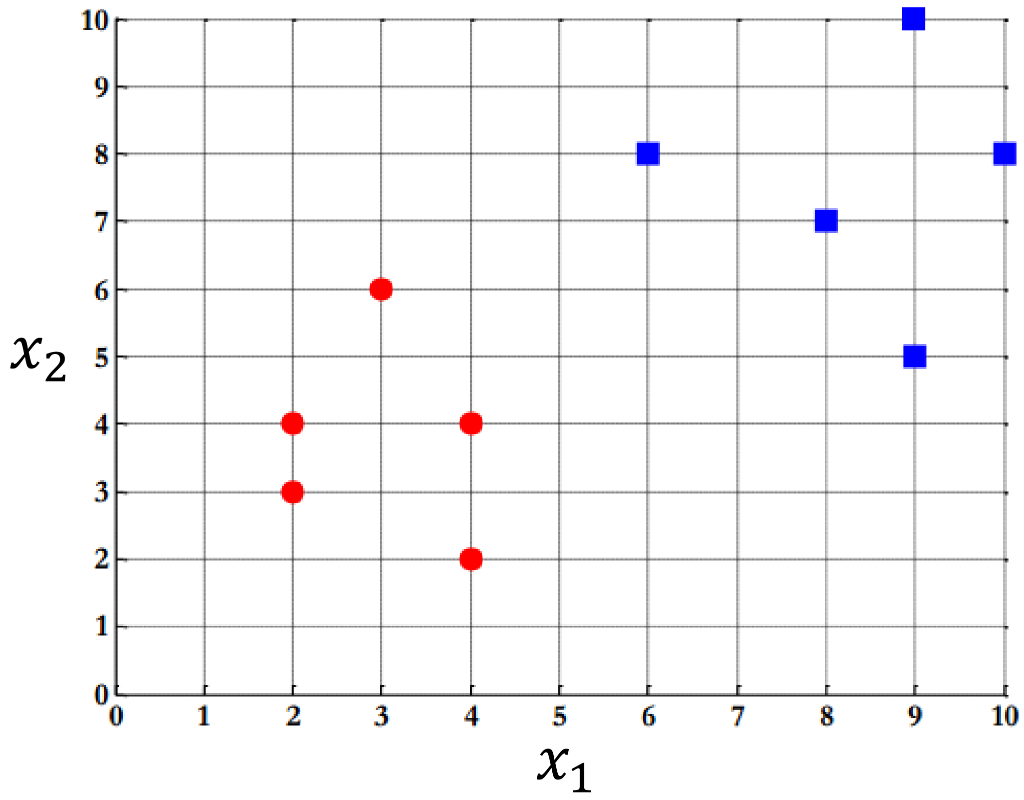


- Between-class scatter matrix:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$
$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]\left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$
$$= \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}\begin{pmatrix} -5.4 & -3.8 \end{pmatrix}$$
$$= \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix}$$

Example

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



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

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

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

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

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

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

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

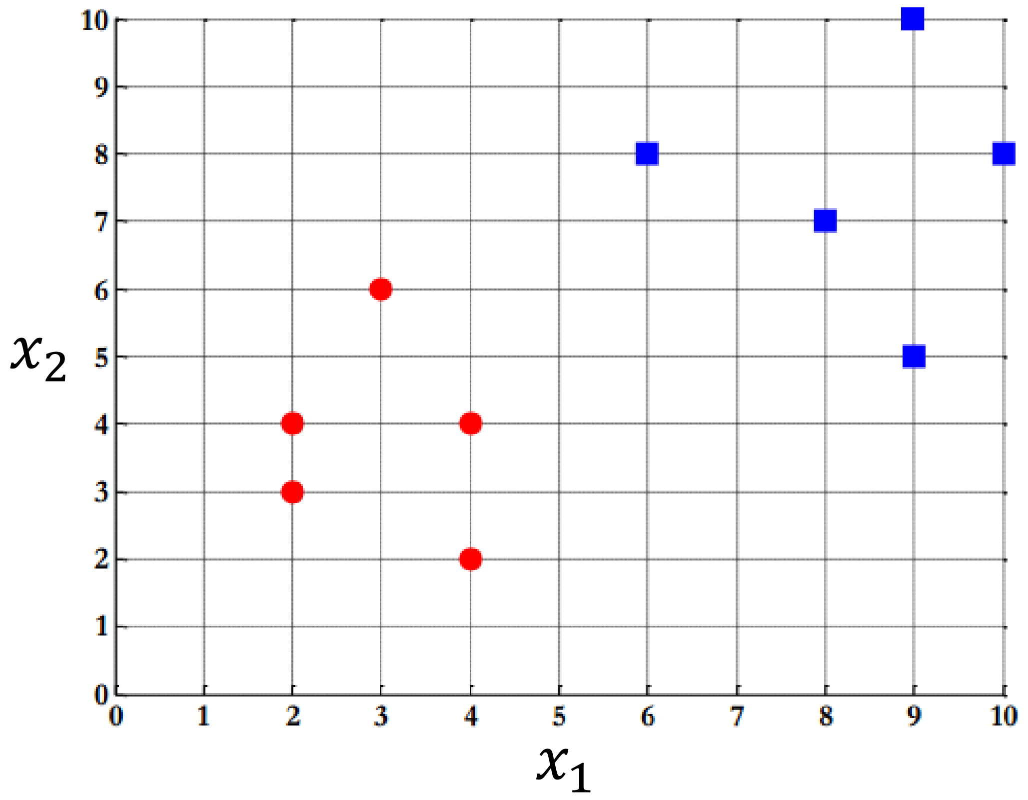
$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

Example

Compute the Linear Discriminant projection for the following two dimensional dataset.

– Samples for class $\boldsymbol{\omega}_1$: $\boldsymbol{X}_1 = (x_1, x_2) = \{(4,2), (2,4), (2,3), (3,6), (4,4)\}$

– Sample for class $\boldsymbol{\omega}_2$: $\boldsymbol{X}_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$



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

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

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

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

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

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

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

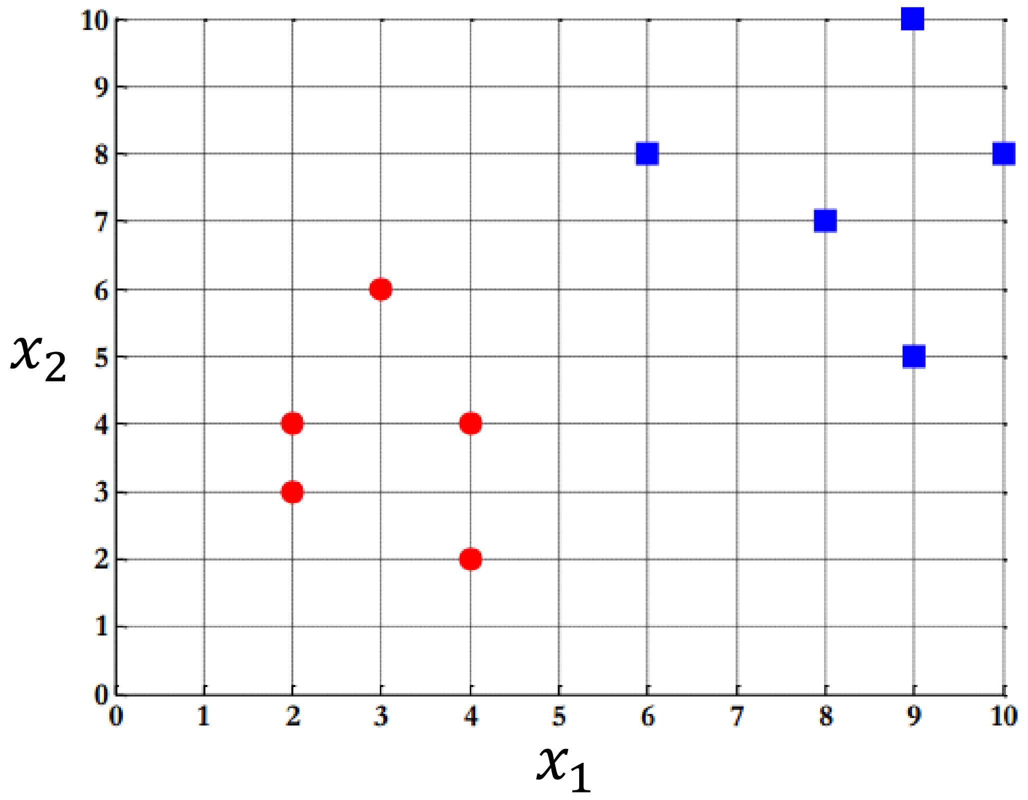
$$\Rightarrow \lambda_1 = 0, \boxed{\lambda_2 = 12.2007}$$

Example

Compute the Linear Discriminant projection for the following two dimensional dataset.

– Samples for class $\boldsymbol{\omega}_1$: $\boldsymbol{X}_1 = (x_1, x_2) = \{(4,2), (2,4), (2,3), (3,6), (4,4)\}$

– Sample for class $\boldsymbol{\omega}_2$: $\boldsymbol{X}_2 = (x_1, x_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$



- The optimal projection is the one that minimizes $\boldsymbol{J} = -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} = -\lambda$

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

and

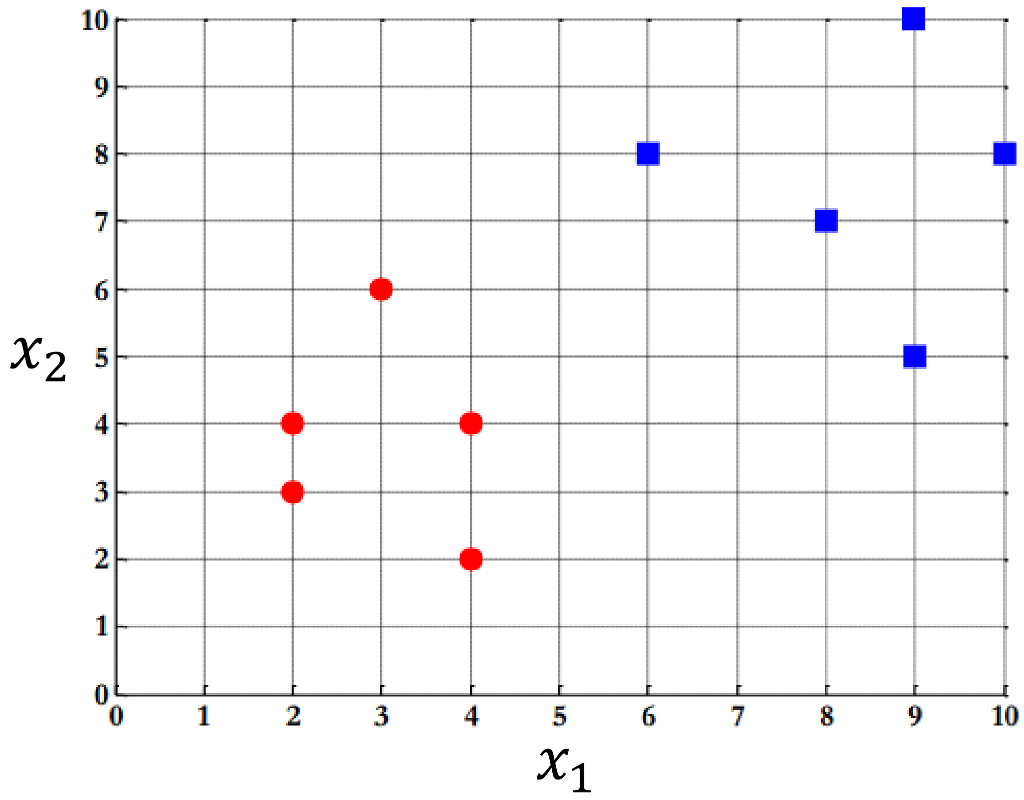
$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \boxed{\underbrace{12.2007}_{\lambda_2}} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Thus;

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

Example

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



- Or directly,

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

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

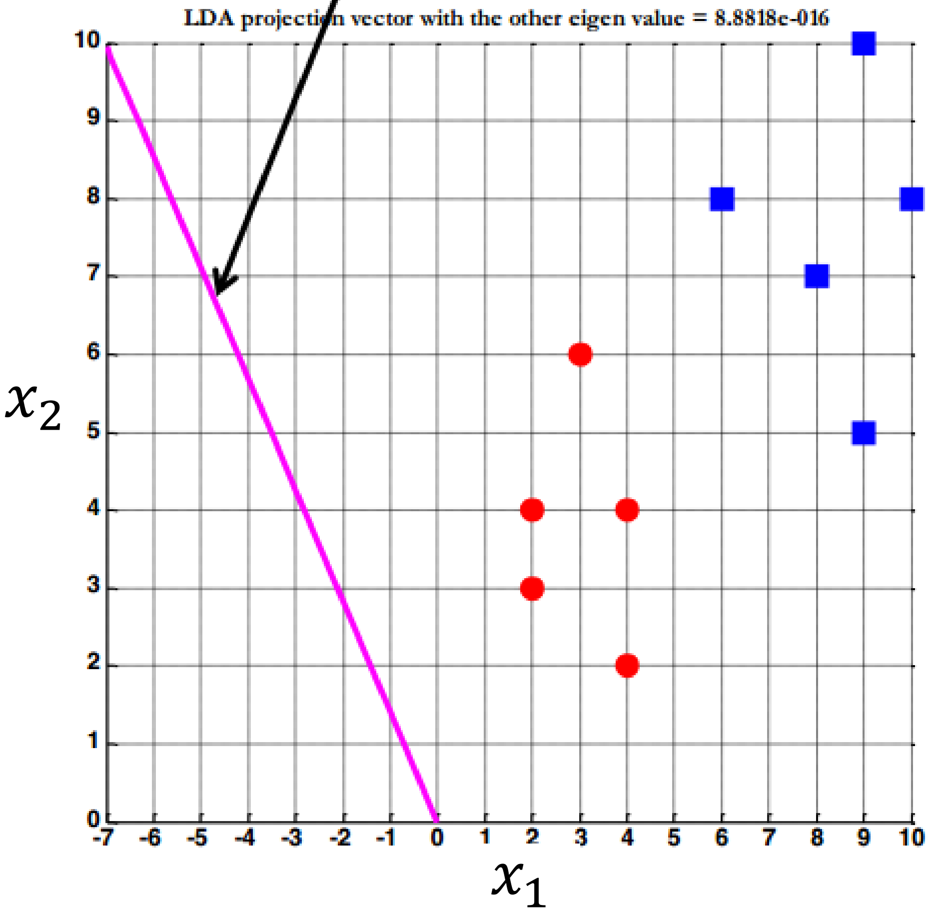
$$\propto \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$$

归一化

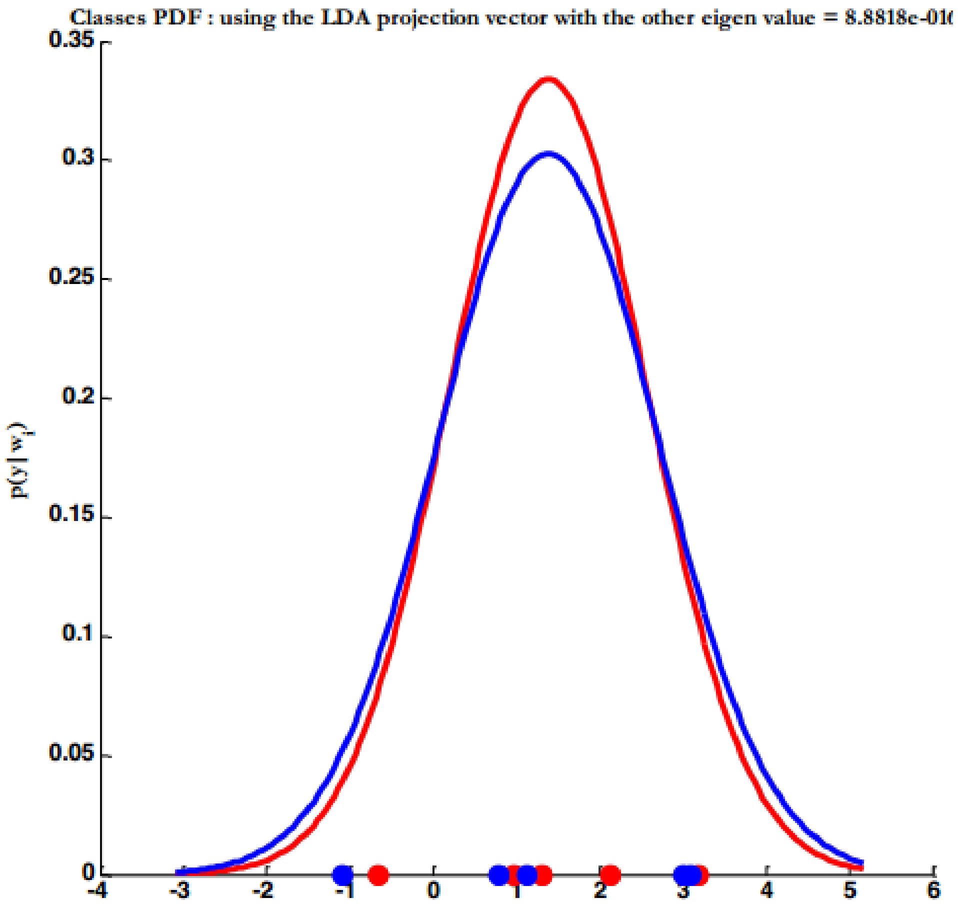
Example

LDA--Projection

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



概率密度函数

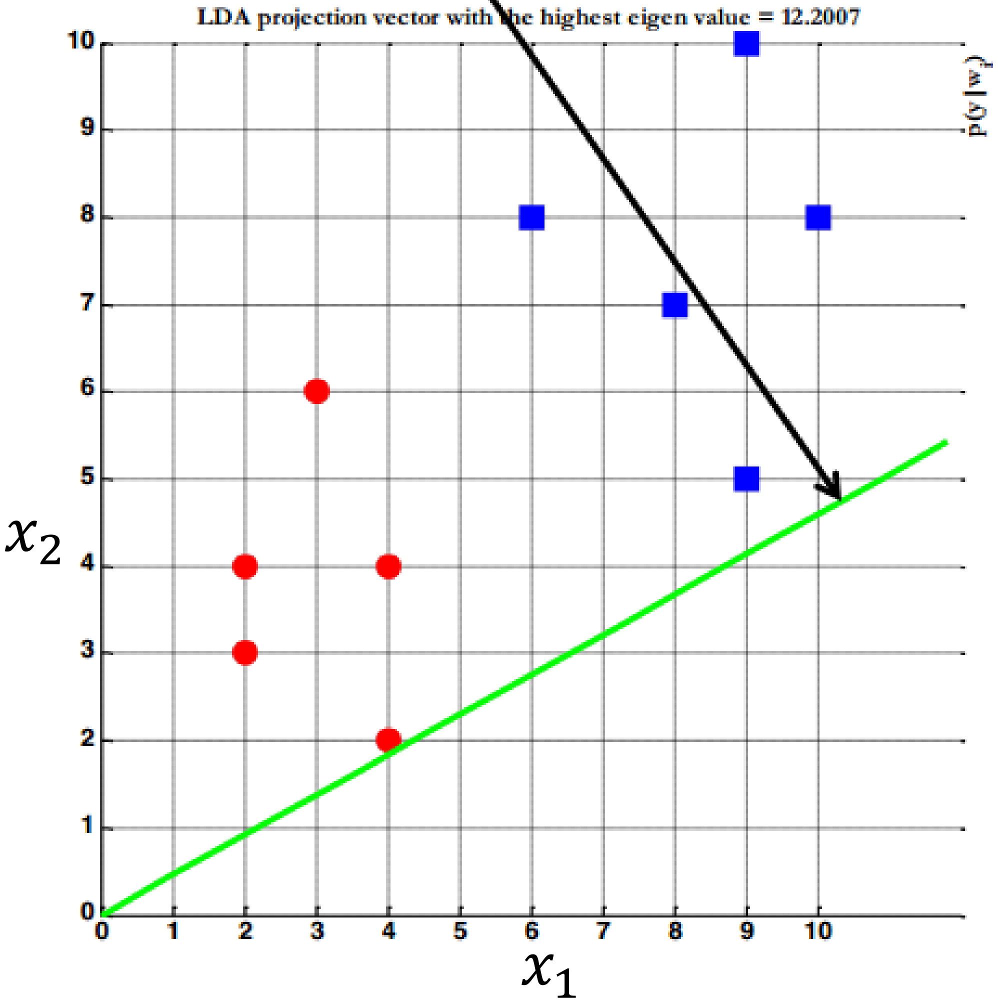


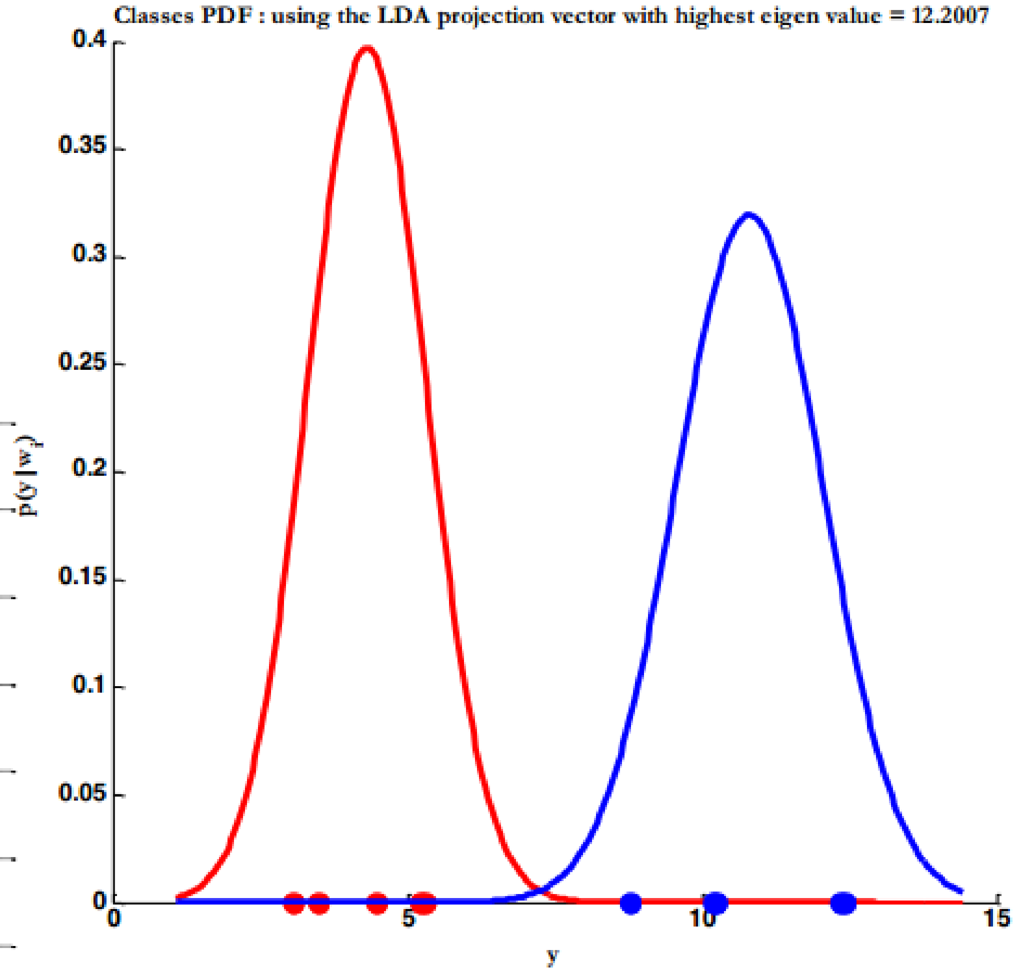
Using this vector leads to **bad separability** between the two classes

Example

LDA--Projection

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





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

Linear Discriminant Analysis—Two Classes

- **Workflow of LDA for the binary classification**

1. Build $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ from the training set
2. Compute $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$
3. Compute $\boldsymbol{S}_w$
4. Compute $\boldsymbol{S}_w^{-1}$
5. Compute $\boldsymbol{\theta}^* = \boldsymbol{S}_w^{-1}\,(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
6. Given a testing sample, $y = \boldsymbol{\theta}^{*\boldsymbol{T}}\boldsymbol{x}$
7. Set the threshold $\gamma = \frac{n_1\boldsymbol{\theta}^{*\boldsymbol{T}}\boldsymbol{\mu}_1 + n_2\boldsymbol{\theta}^{*\boldsymbol{T}}\boldsymbol{\mu}_2}{n_1+n_2}$.
8. Compare y with γ to determine the class.

Linear Discriminant Analysis—C Classes

- Assume we have C classes, each class has n_i d-dimensional samples, where $i = 1,2,\dots,C$
- A transformation $\boldsymbol{\Theta} \in \mathbb{R}^{d\times p}$: project the samples in $\boldsymbol{X}$ onto $\boldsymbol{Y}$ $(p \ll d)$. In fact, $p \leq C - 1$, we will see later.

$$\boldsymbol{y}_i = \boldsymbol{\Theta}^{\boldsymbol{T}} \boldsymbol{x}_i$$

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \qquad \boldsymbol{y}_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{ip} \end{bmatrix} \qquad \boldsymbol{\Theta} = \left[\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_p\right] \in \mathbb{R}^{d\times p}$$

Linear Discriminant Analysis—C Classes

- We have N d-dimensional samples from C classes, e.g., seabass, tuna, …
- Each class has n_i samples, where $i = 1,2, \dots, C$
- Stacking these samples from different classes into one big fat matrix $\boldsymbol{X} \in \mathbb{R}^{d\times N}$ such that each column represents one sample $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$.
- We aim to obtain a transformation $\boldsymbol{\Theta} \in \mathbb{R}^{d\times p}$ to project the d-dimensional samples in $\boldsymbol{X}$ onto a p-dimensional subspace $(p < d)$, such that after the projection we have:

In fact, $p \le C-1$, we will see later.

class means to be as **far apart** from each other as possible	➡	the **between-class** scatter to be **large**
samples from the same class to be as **close** to their mean as possible	➡	the **within-class** scatter to be **small**

Linear Discriminant Analysis—C Classes

The generalization of the within-class covariance matrix to the case of C classes.

Within-class scatter:

$$\boldsymbol{S}_w = \sum_{i=1}^{C} \boldsymbol{S}_{wi} \qquad \boldsymbol{S}_{wi} = \sum_{x \in \mathrm{C_i}} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\boldsymbol{T}} \qquad \boldsymbol{S}_w \in \mathbb{R}^{d \times d}$$

Class mean vector (sample): $\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x \in C_i} \boldsymbol{x},\ \boldsymbol{\mu}_i \in \mathbb{R}^{d \times 1}$

In order to find a generalization of the between-class covariance matrix, we follow Duda and Hart (1973) and consider the total covariance matrix first.

$$\boldsymbol{S}_t = \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\boldsymbol{T}} \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}$$

Linear Discriminant Analysis—C Classes

The total covariance matrix can be decomposed into

$$\boldsymbol{S}_t = \boldsymbol{S}_w + \boldsymbol{S}_b$$

Between-class scatter:

$$\boldsymbol{S}_b = \sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} = \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\boldsymbol{T}} \qquad \mathbf{S}_b \in \mathbb{R}^{d\times d}$$

Linear Discriminant Analysis—C Classes

$$\boldsymbol{S}_t = \sum_x (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^{\boldsymbol{T}} = \boldsymbol{S}_w + \boldsymbol{S}_b$$

$$\boldsymbol{S}_w = \sum_{i=1}^{C} \sum_{x\in \mathrm{C_i}} (\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\boldsymbol{T}}$$

$$\boldsymbol{S}_b = \sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^{\boldsymbol{T}}$$

$$\boldsymbol{S}_t = \sum_x (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^{\boldsymbol{T}} = \sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{x}_{ij}-\boldsymbol{\mu})(\boldsymbol{x}_{ij}-\boldsymbol{\mu})^{\boldsymbol{T}} \qquad \boldsymbol{x}_{ij}\in C_{\boldsymbol{i}}$$

$$= \sum_{i=1}^{C}\sum_{j=1}^{n_i}[(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)+(\boldsymbol{\mu}_i-\boldsymbol{\mu})][(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)+(\boldsymbol{\mu}_i-\boldsymbol{\mu})]^{\boldsymbol{T}}$$

$$= \sum_{i=1}^{C}\sum_{j=1}^{n_i}[(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)^{\boldsymbol{T}} + (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)^{\boldsymbol{T}} + (\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)(\boldsymbol{\mu}_i-\boldsymbol{\mu})^{\boldsymbol{T}} + (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^{\boldsymbol{T}}]$$

$$= \sum_{i=1}^{C}\sum_{j=1}^{n_i}\left[(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)^{\boldsymbol{T}} + (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^{\boldsymbol{T}}\right] = \boldsymbol{S}_w + \boldsymbol{S}_b$$

$$\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{x}_{ij}-\boldsymbol{\mu}_i)^{\boldsymbol{T}} = \sum_{i=1}^{C}(\boldsymbol{\mu}_i-\boldsymbol{\mu})\left(\sum_{j=1}^{n_i}\boldsymbol{x}_{ij} - \sum_{j=1}^{n_i}\boldsymbol{\mu}_i\right)^{\boldsymbol{T}} = 0$$

Linear Discriminant Analysis—C Classes

- Assume we have C classes, each class has n_i d-dimensional samples, where $i = 1,2,\dots,C$
- A transformation $\mathbf{\Theta} \in \mathbb{R}^{d\times p}$: project the samples in $\boldsymbol{X}$ onto $\boldsymbol{Y}$ $(p \ll d)$. In fact, $p \le C - 1$, we will see later.

$$\boldsymbol{y}_i = \mathbf{\Theta}^{\boldsymbol{T}}\boldsymbol{x}_i$$

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \qquad \boldsymbol{y}_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{ip} \end{bmatrix} \qquad \mathbf{\Theta} = \left[\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_p\right] \in \mathbb{R}^{d\times p}$$

$$\tilde{\boldsymbol{S}}_w = \boldsymbol{\Theta}^{\boldsymbol{T}}\boldsymbol{S}_w\boldsymbol{\Theta} \qquad \tilde{\boldsymbol{S}}_b = \boldsymbol{\Theta}^{\boldsymbol{T}}\boldsymbol{S}_b\boldsymbol{\Theta} \qquad \widetilde{\boldsymbol{\mu}}_i = \boldsymbol{\Theta}^{\boldsymbol{T}}\boldsymbol{\mu}_i \qquad \widetilde{\boldsymbol{\mu}} = \boldsymbol{\Theta}^{\boldsymbol{T}}\boldsymbol{\mu}$$

Linear Discriminant Analysis—C Classes

Popular objective function:

$$J_1(\mathbf{\Theta}) = \max_{\mathbf{\Theta}} \frac{tr(\tilde{\boldsymbol{S}}_b)}{tr(\tilde{\boldsymbol{S}}_w)} = \max_{\mathbf{\Theta}} \frac{tr(\mathbf{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_b \mathbf{\Theta})}{tr(\mathbf{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_w \mathbf{\Theta})}$$

$$J_2(\mathbf{\Theta}) = \max_{\mathbf{\Theta}} tr(\tilde{\boldsymbol{S}}_w^{-1} \tilde{\boldsymbol{S}}_b) = \max_{\mathbf{\Theta}} tr((\mathbf{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_w \mathbf{\Theta})^{-1} \mathbf{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_b \mathbf{\Theta})$$

$$J_3(\mathbf{\Theta}) = \frac{|\tilde{\boldsymbol{S}}_b|}{|\tilde{\boldsymbol{S}}_w|}$$

This technique was developed by R. A. Fisher (1936) for **the two-class case** and extended by C. R. Rao (1948) to handle **the multiclass case**.

Linear Discriminant Analysis—C Classes

In $J_1(\boldsymbol{\Theta})$, what is the meaning of "**trace**"?

$$J_1(\boldsymbol{\Theta}) = \max_{\boldsymbol{\Theta}} \frac{tr(\tilde{\boldsymbol{S}}_b)}{tr(\tilde{\boldsymbol{S}}_w)} = \max_{\boldsymbol{\Theta}} \frac{tr(\boldsymbol{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_b \boldsymbol{\Theta})}{tr(\boldsymbol{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_w \boldsymbol{\Theta})}$$

$$\boldsymbol{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_b \boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^{\boldsymbol{T}} \\ \vdots \\ \boldsymbol{\theta}_p^{\boldsymbol{T}} \end{bmatrix} \boldsymbol{S}_b [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_p] = \begin{bmatrix} \boldsymbol{\theta}_1^{\boldsymbol{T}} \\ \vdots \\ \boldsymbol{\theta}_p^{\boldsymbol{T}} \end{bmatrix} [\boldsymbol{S}_b \boldsymbol{\theta}_1, \boldsymbol{S}_b \boldsymbol{\theta}_2, \dots, \boldsymbol{S}_b \boldsymbol{\theta}_p]$$

$$tr(\boldsymbol{\Theta}^{\boldsymbol{T}} \boldsymbol{S}_b \boldsymbol{\Theta}) = \sum_{i=1}^{p} \boldsymbol{\theta}_i^{\boldsymbol{T}} \boldsymbol{S}_b \boldsymbol{\theta}_i \qquad tr(\boldsymbol{\Theta}^{T} \boldsymbol{S}_w \boldsymbol{\Theta}) = \sum_{i=1}^{p} \boldsymbol{\theta}_i^{\boldsymbol{T}} \boldsymbol{S}_w \boldsymbol{\theta}_i$$

Linear Discriminant Analysis—C Classes

Optimization $J_1(\mathbf{\Theta})$:

- Recall in two-classes case, we solved the eigenvalue problem.

$$\min_{\boldsymbol{\theta}} -\boldsymbol{\theta}^T \boldsymbol{S}_b \boldsymbol{\theta} \quad \text{s.t. } \boldsymbol{\theta}^T \boldsymbol{S}_w \boldsymbol{\theta} = 1 \qquad \Rightarrow \qquad \boldsymbol{S}_b \boldsymbol{\theta} = \lambda \boldsymbol{S}_w \boldsymbol{\theta}$$

- For C-classes case, we have p projection vectors,

$$\boldsymbol{S}_w^{-1} \boldsymbol{S}_b \boldsymbol{\theta}_i = \lambda \boldsymbol{\theta}_i, \qquad i = 1,2, \dots, p$$

Columns of $\mathbf{\Theta}^*$ are eigenvectors corresponding to the largest eigenvalues:

$$\boldsymbol{S}_w^{-1} \boldsymbol{S}_b \mathbf{\Theta}^* = \lambda \mathbf{\Theta}^* \qquad \mathbf{\Theta}^* = [\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \dots, \boldsymbol{\theta}_p^*]$$

$p \leq C - 1$, why?

Linear Discriminant Analysis—C Classes

- $\boldsymbol{S}_b$ has a maximum rank of $C-1$.
  - $\boldsymbol{S}_b$ is the sum of C $rank = 1$ matrices, and because only $C-1$ of these are independent,

$$\boldsymbol{S}_b = \sum_{i=1}^{C} \frac{n_i}{N} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

Given a matrix $\boldsymbol{A}_{m\times n}$ and $\boldsymbol{B}_{n\times k}$,

- $rank(\boldsymbol{AB}) = min(rank(\boldsymbol{A}), rank(\boldsymbol{B}))$
- $rank(\boldsymbol{A}+\boldsymbol{B}) \leq rank(\boldsymbol{A}) + rank(\boldsymbol{B})$

$$rank\left((\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}}\right) = rank(\boldsymbol{\mu}_i - \boldsymbol{\mu}) \leq 1 \qquad rank(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b) \leq rank(\boldsymbol{S}_b) \leq C-1$$

$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ has at most $C-1$ nonzero eigenvalues.

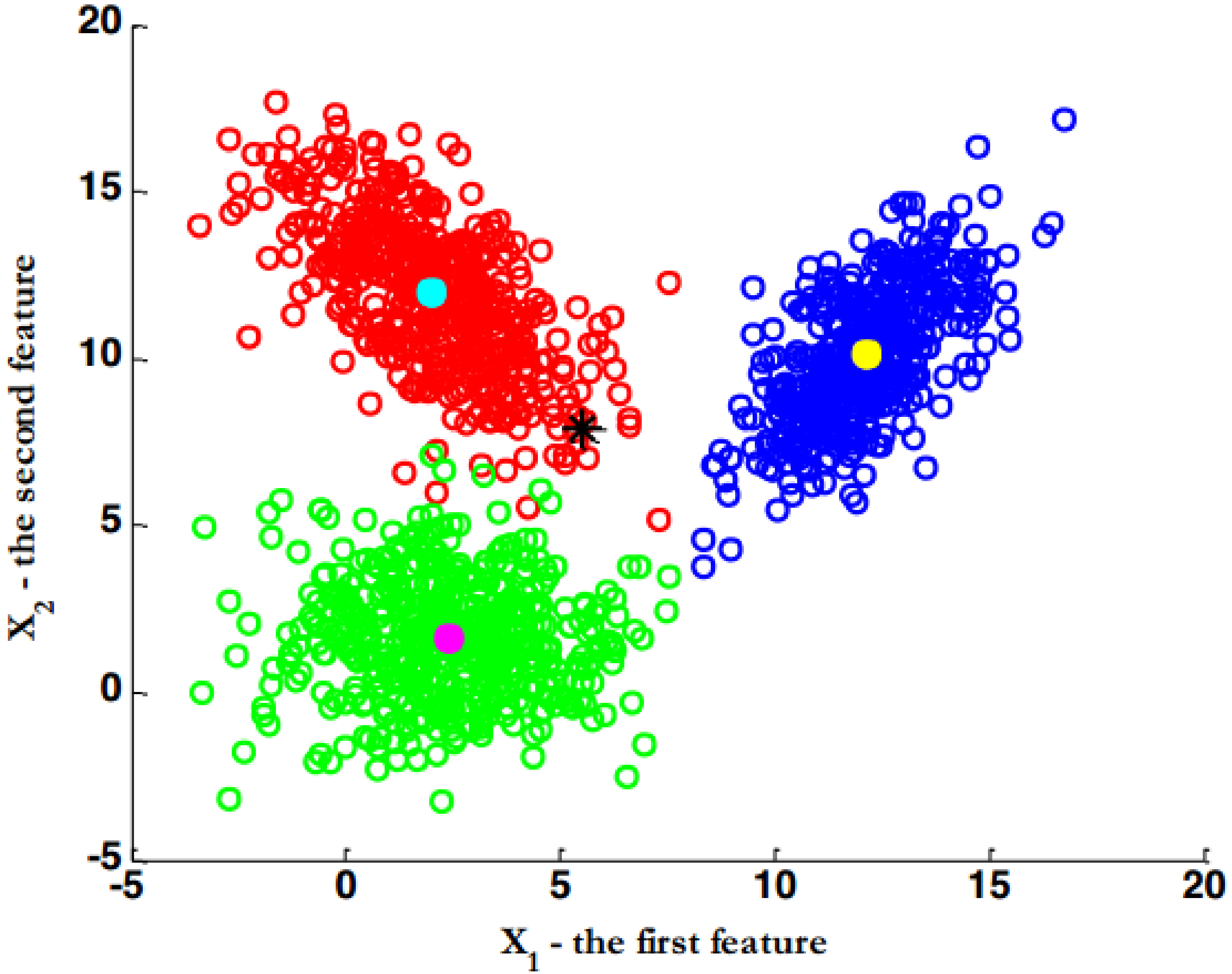
Zero eigenvalue does not alter the value of $J_1(\boldsymbol{\Theta})$.

Linear Discriminant Analysis—C Classes

- **Workflow of LDA for the C-classification**

1. Compute $\boldsymbol{\mu}_i$
2. Compute $\boldsymbol{S}_b$
3. Compute $\boldsymbol{S}_w^{-1}$
4. Compute the largest p eigenvalues of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ and the corresponding eigenvectors $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_p\}$.
5. Let $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_p]$, then $\boldsymbol{y}_i = \boldsymbol{\Theta}^T\boldsymbol{x}_i$

Illustration-3 Classes



```
%% computing the LDA
% class means
Mu1 = mean(X1')';
Mu2 = mean(X2')';
Mu3 = mean(X3')';

% overall mean
Mu = (Mu1 + Mu2 + Mu3)./3;

% class covariance matrices
S1 = cov(X1');
S2 = cov(X2');
S3 = cov(X3');

% within-class scatter matrix
Sw = S1 + S2 + S3 ;

% number of samples of each class
N1 = size(X1,2);
N2 = size(X2,2);
N3 = size(X3,2);

% between-class scatter matrix
SB1 = N1 .* (Mu1-Mu)*(Mu1-Mu)';
SB2 = N2 .* (Mu2-Mu)*(Mu2-Mu)';
SB3 = N3 .* (Mu3-Mu)*(Mu3-Mu)';

SB = SB1 + SB2 + SB3;

% computing the LDA projection
invSw = inv(Sw);
invSw_by_SB = invSw * SB;

% getting the projection vectors
%[V,D] = EIG(X) produces a diagonal matrix D of eigenvalues and a
%full matrix V whose columns are the corresponding eigenvectors
[V,D] = eig(invSw_by_SB);

% the projection vectors - we will have at most C-1 projection vectors,
% from which we can choose the most important ones ranked by their
% corresponding eigen values ... lets investigate the two projection
% vectors
W1 = V(:,1);
W2 = V(:,2);
```

$$S_W^{-1} S_B$$

Recall …

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

where $$S_i = \sum_{x \in \omega_i} (x-\mu_i)(x-\mu_i)^T$$

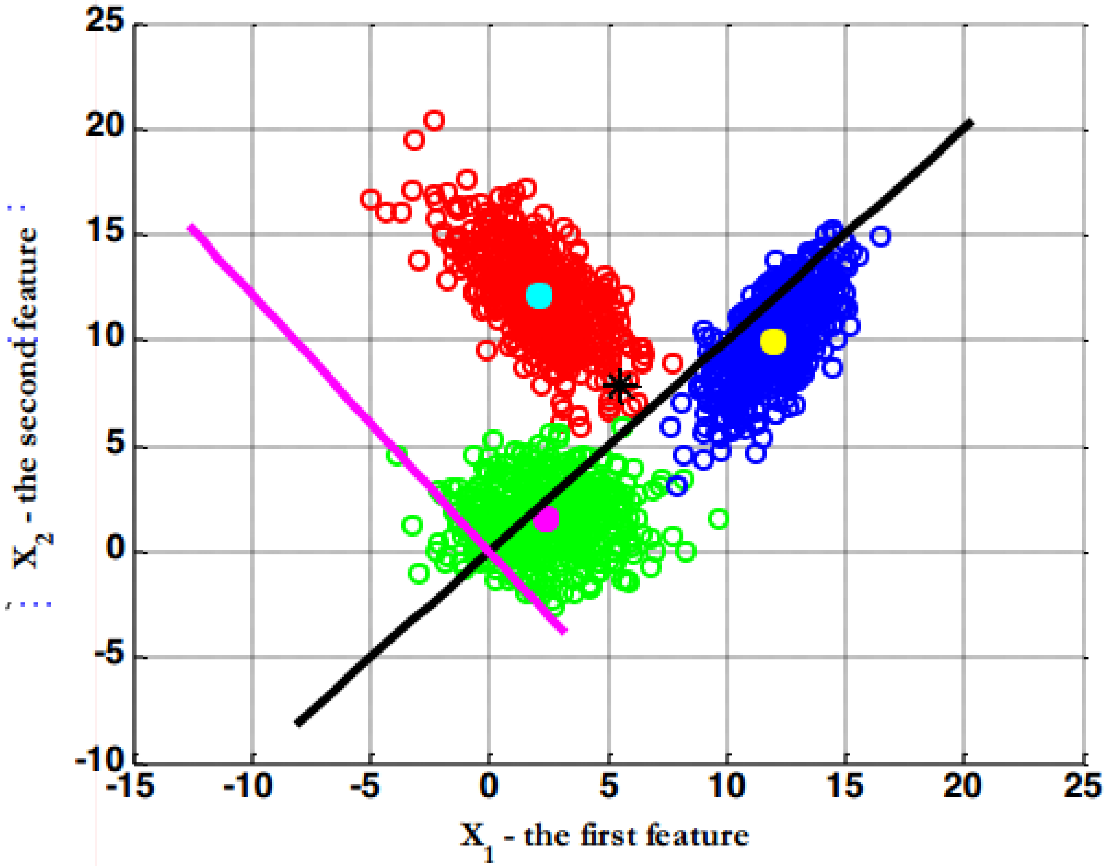
and $$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

where $$\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{\forall x} N_i \mu_i$$

and $$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

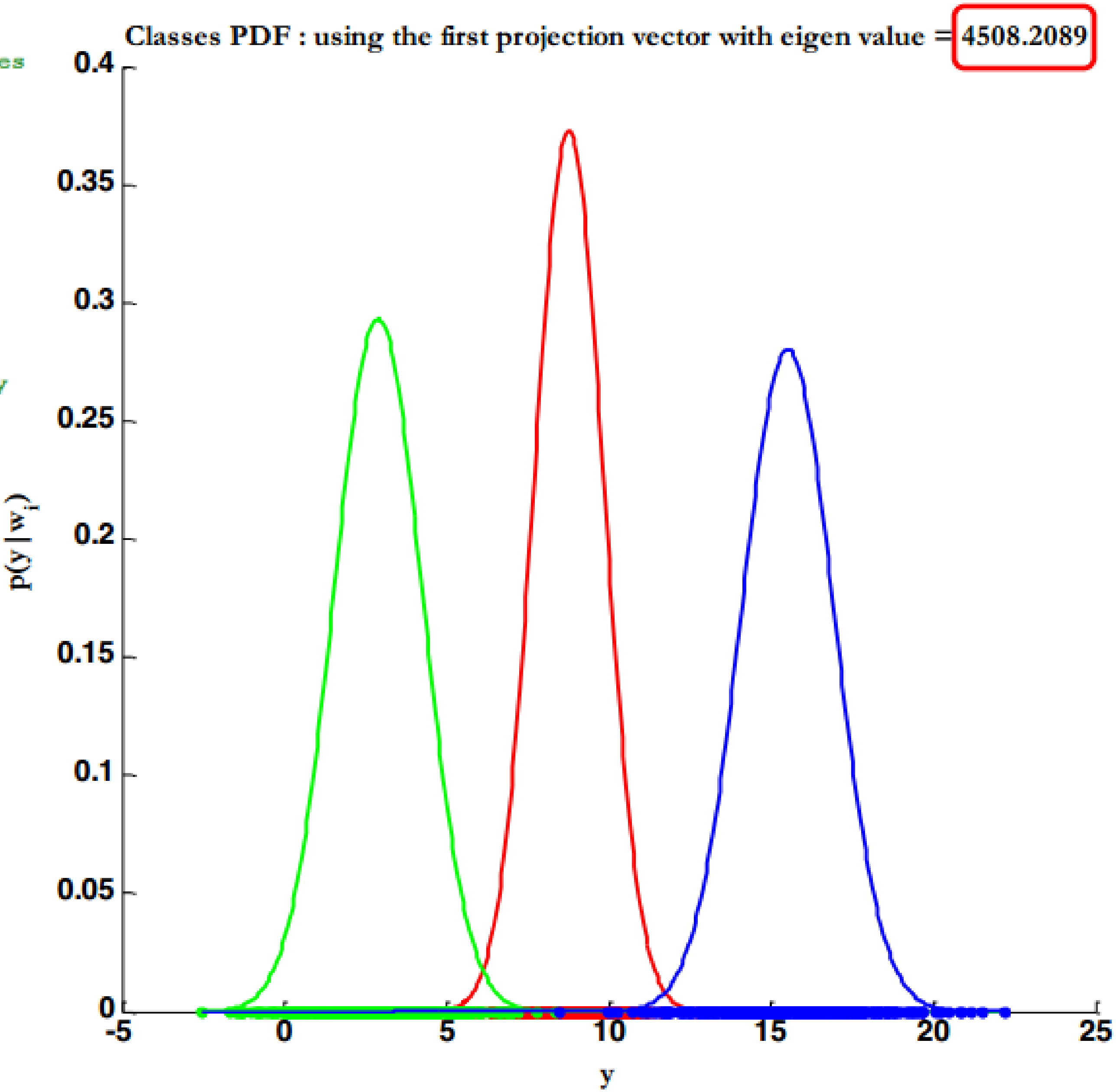
```
%% lets visualize them ...
% we will plot the scatter plot to better visualize the features
hfig = figure;
axes1 = axes('Parent',hfig,'FontWeight','bold','FontSize',12);
hold('all');

% Create xlabel
xlabel('X_1 - the first feature','FontWeight','bold','FontSize',12,...
    'FontName','Garamond');

% Create ylabel
ylabel('X_2 - the second feature','FontWeight','bold','FontSize',12,...
    'FontName','Garamond');

% the first class
scatter(X1(1,:),X1(2,:), 'r','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu1_est(1),Mu1_est(2),'co','MarkerSize',8,'MarkerEdgeColor','c',...
    'Color','c','LineWidth',2,'MarkerFaceColor','c','Parent',axes1);
hold on

% the second class
scatter(X2(1,:),X2(2,:), 'g','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu2_est(1),Mu2_est(2),'mo','MarkerSize',8,'MarkerEdgeColor','m',...
    'Color','m','LineWidth',2,'MarkerFaceColor','m','Parent',axes1);
hold on

% the third class
scatter(X3(1,:),X3(2,:), 'b','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu3_est(1),Mu3_est(2),'yo','LineWidth',2,'MarkerSize',8,'MarkerEdgeColor',...
    'y','Color','y','MarkerFaceColor','y','Parent',axes1);
hold on
```




```
% drawing the projection vectors
% the first vector
t = -10:25;
line_x1 = t .* W1(1);
line_y1 = t .* W1(2);

% the second vector
t = -5:20;
line_x2 = t .* W2(1);
line_y2 = t .* W2(2);

plot(line_x1,line_y1,'k-','LineWidth',3);
hold on
plot(line_x2,line_y2,'m-','LineWidth',3);
grid on
```

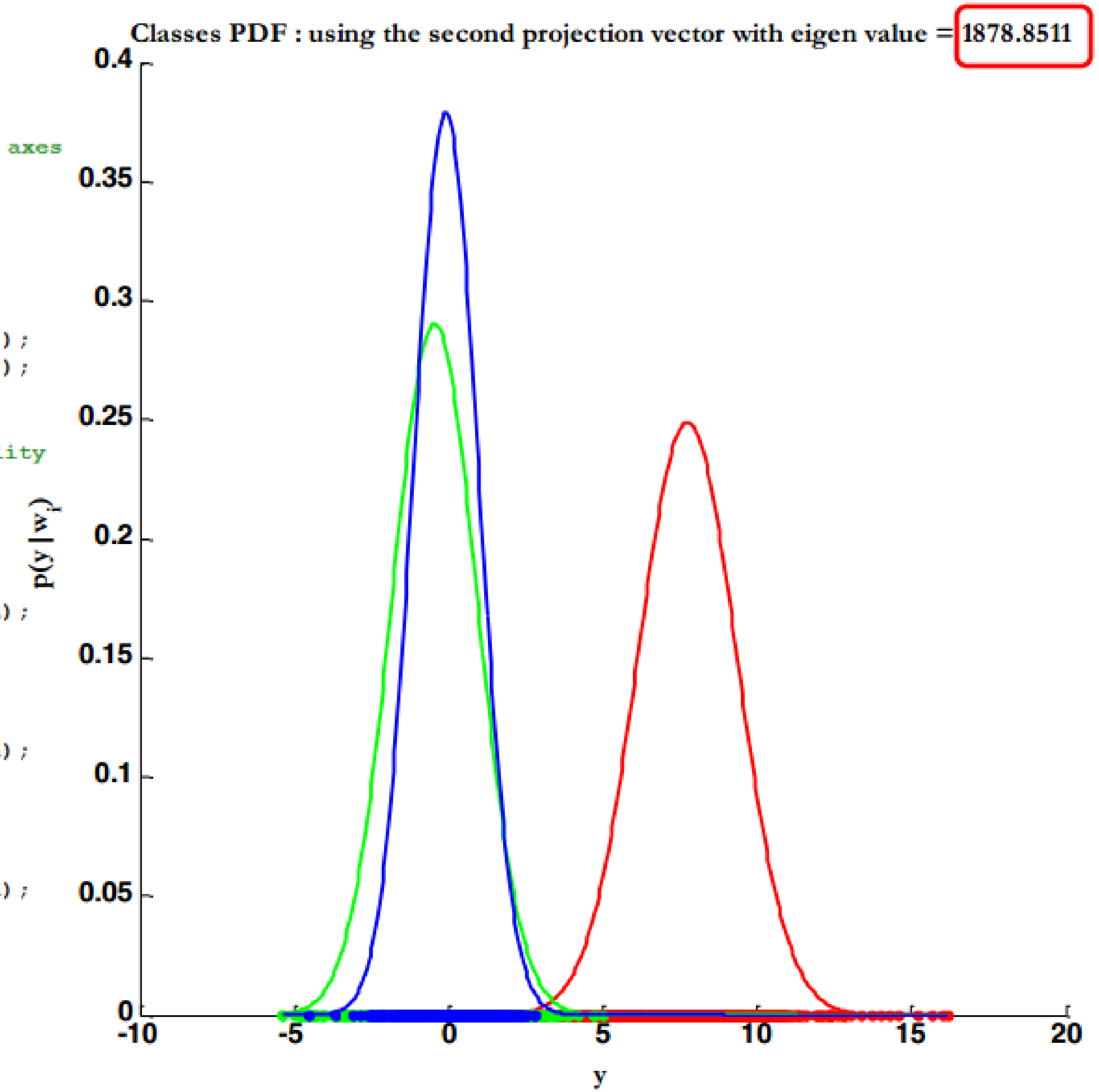
Along first projection vector $y = w_1^T x$

```
% project data samples along the projections axes
% the first projection vector
y1_w1 = W1'*X1;
y2_w1 = W1'*X2;
y3_w1 = W1'*X3;

% projection limits
minY = min([min(y1_w1),min(y2_w1),min(y3_w1)]);
maxY = max([max(y1_w1),max(y2_w1),max(y3_w1)]);
y_w1 = minY:0.05:maxY;

% for visualization lets compute the probability
% density function of the
% classes after projection
% the first class
y1_w1_Mu = mean(y1_w1);
y1_w1_sigma = std(y1_w1);
y1_w1_pdf = mvnpdf(y_w1',y1_w1_Mu,y1_w1_sigma);

% the second class
y2_w1_Mu = mean(y2_w1);
y2_w1_sigma = std(y2_w1);
y2_w1_pdf = mvnpdf(y_w1',y2_w1_Mu,y2_w1_sigma);

% the third class
y3_w1_Mu = mean(y3_w1);
y3_w1_sigma = std(y3_w1);
y3_w1_pdf = mvnpdf(y_w1',y3_w1_Mu,y3_w1_sigma);
```

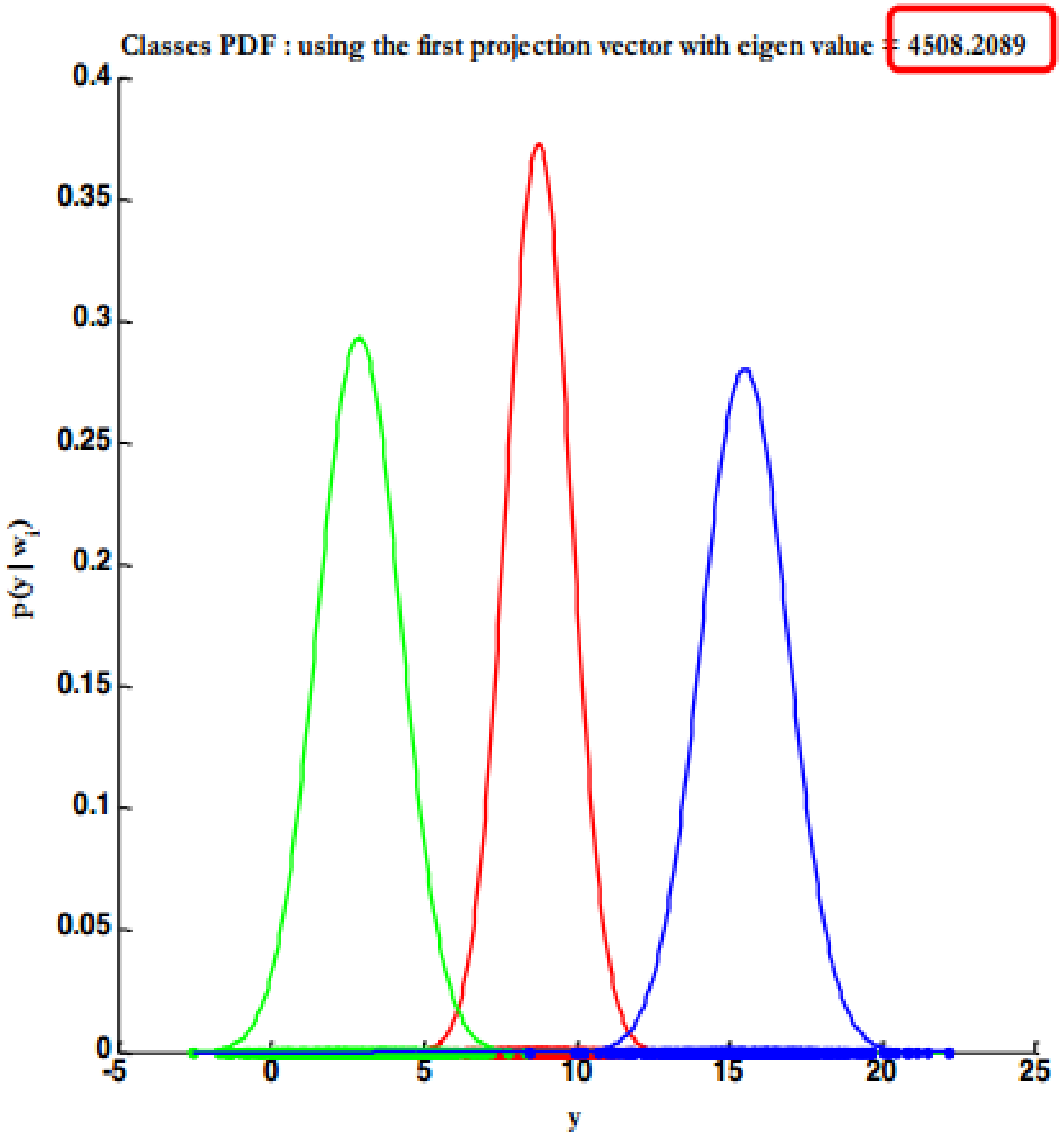


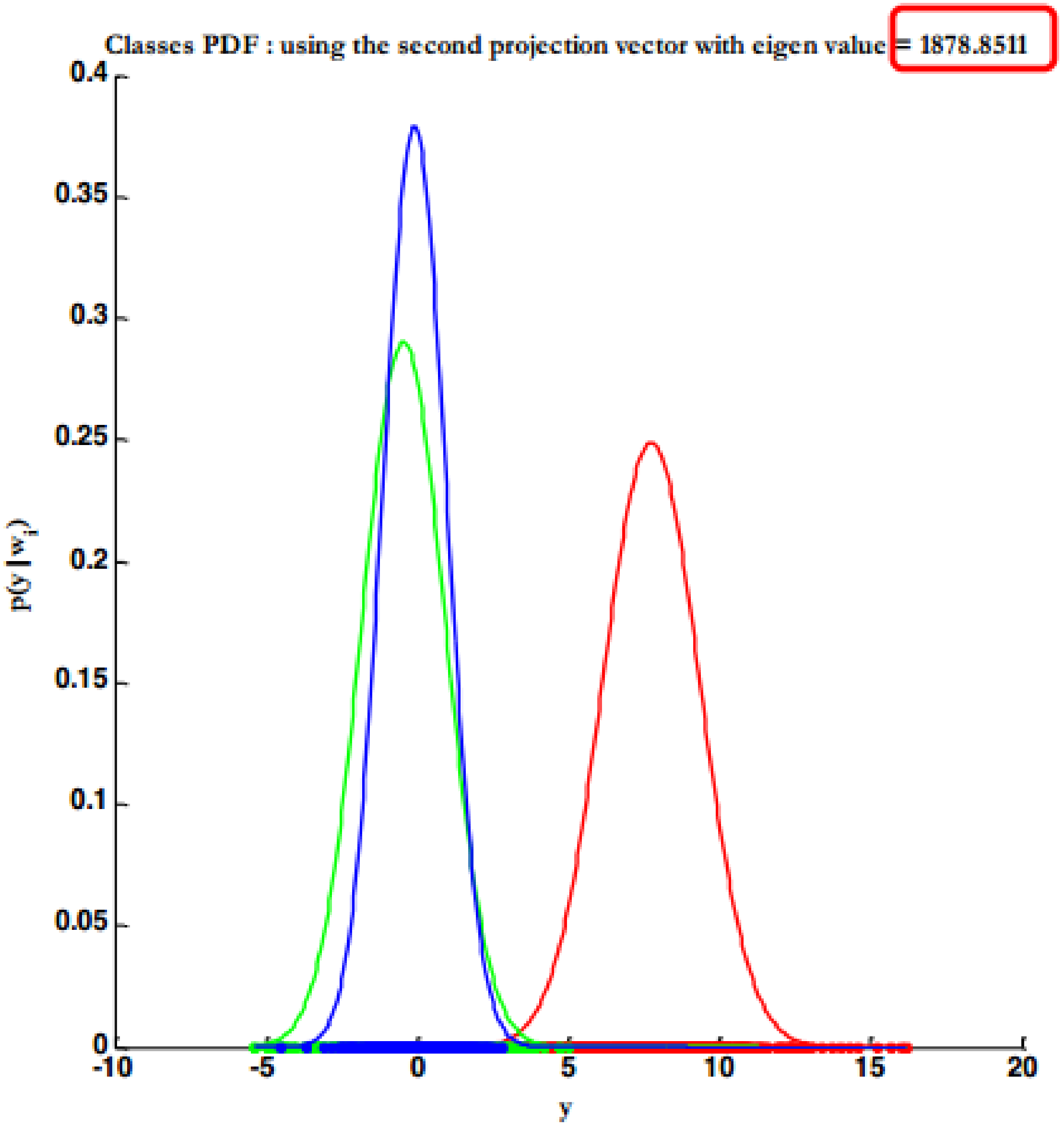
Along second projection vector $y = w_2^T x$

```
% project data samples along the projections axes
% the second projection vector
y1_w2 = W2'*X1;
y2_w2 = W2'*X2;
y3_w2 = W2'*X3;

% projection limits
minY = min([min(y1_w2),min(y2_w2),min(y3_w2)]);
maxY = max([max(y1_w2),max(y2_w2),max(y3_w2)]);
y_w2 = minY:0.05:maxY;

% for visualization lets compute the probability
% density function of the
% classes after projection
% the first class
y1_w2_Mu = mean(y1_w2);
y1_w2_sigma = std(y1_w2);
y1_w2_pdf = mvnpdf(y_w2',y1_w2_Mu,y1_w2_sigma);

% the second class
y2_w2_Mu = mean(y2_w2);
y2_w2_sigma = std(y2_w2);
y2_w2_pdf = mvnpdf(y_w2',y2_w2_Mu,y2_w2_sigma);

% the third class
y3_w2_Mu = mean(y3_w2);
y3_w2_sigma = std(y3_w2);
y3_w2_pdf = mvnpdf(y_w2',y3_w2_Mu,y3_w2_sigma);
```



Apparently, the projection vector that has the highest eigenvalue provides higher discrimination power between classes.





Summary

- Linear Discriminant Analysis—Two Classes
  - Minimize within-class scatter
  - Maximize between-class scatter
  - The eigenvector of the largest eigenvalue of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ (as $-\boldsymbol{\theta}^{*\boldsymbol{T}}\boldsymbol{S}_b\boldsymbol{\theta}^* = -\lambda\boldsymbol{\theta}^{*\boldsymbol{T}}\boldsymbol{S}_w\boldsymbol{\theta}^* = -\lambda$)
  - Or $\boldsymbol{\theta}^* = \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
- Linear Discriminant Analysis—C Classes
  - Dimension reduction. $\boldsymbol{\Theta} \in \mathbb{R}^{d \times p} : \boldsymbol{X} \to \boldsymbol{Y}$ $(p \ll d)$. In fact, $p \leq C - 1$.
  - Columns of $\boldsymbol{\Theta}^*$ are eigenvectors of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ corresponding to the p largest eigenvalues.

Backup Slides

Statistical Facts

Between-class scatter:

$$\boldsymbol{S}_b = \sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} = \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\boldsymbol{T}}$$

$$\frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\boldsymbol{T}} = \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j[(\boldsymbol{\mu}_i - \boldsymbol{\mu}) + (\boldsymbol{\mu} - \boldsymbol{\mu}_j)][(\boldsymbol{\mu}_i - \boldsymbol{\mu}) + (\boldsymbol{\mu} - \boldsymbol{\mu}_j)]^{\boldsymbol{T}}$$

$$= \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j[(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} + (\boldsymbol{\mu} - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu} - \boldsymbol{\mu}_j)^{\boldsymbol{T}} + (\boldsymbol{\mu} - \boldsymbol{\mu}_j)(\boldsymbol{\mu} - \boldsymbol{\mu}_j)^{\boldsymbol{T}}]$$

$$= \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j[(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} + (\boldsymbol{\mu} - \boldsymbol{\mu}_j)(\boldsymbol{\mu} - \boldsymbol{\mu}_j)^{\boldsymbol{T}}]$$

$$= \frac{1}{2}\sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} + \frac{1}{2}\sum_{j=1}^{C} n_j(\boldsymbol{\mu} - \boldsymbol{\mu}_j)(\boldsymbol{\mu} - \boldsymbol{\mu}_j)^{\boldsymbol{T}}$$

$$= \sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\boldsymbol{T}} = \boldsymbol{S}_b$$

Relation to Least Squares

- The least-squares approach: based on the goal of making the model predictions as close as possible to a set of target values.
- By contrast, the LDA (Fisher criterion) was derived by requiring maximum class separation in the output space.

It is interesting to see the relationship between these two approaches.

Relation to Least Squares

- We adopt a slightly different target coding scheme instead of {1,-1}.
  - Then the least-squares solution becomes equivalent to the Fisher solution (Duda and Hart, 1973).
- In particular, we shall take the targets (y_i) for class C_1 to be $\frac{N}{n_1}$.
- For class C_2, we shall take the targets (y_i) to be $-\frac{N}{n_2}$.
  - $(n_1 + n_2 = N)$

$$J = \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{n}} + b - y_n)^2$$

Relation to Least Squares

$$E = \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{w^T}\boldsymbol{x_n} + b - y_n)^2$$

$$\frac{\partial E}{\partial b} = 0 \qquad \qquad \sum_{n=1}^{N}\left(\boldsymbol{w^T}\boldsymbol{x_n} + b - y_n\right) = 0$$

$$\Longrightarrow$$

$$\frac{\partial E}{\partial \boldsymbol{w}} = 0 \qquad \qquad \sum_{n=1}^{N}\left(\boldsymbol{w^T}\boldsymbol{x_n} + b - y_n\right)\boldsymbol{x_n} = 0$$

$$\sum_{n=1}^{N} y_n = n_1\frac{N}{n_1} - n_2\frac{N}{n_2} = 0 \qquad \Longrightarrow \qquad \begin{aligned} \boldsymbol{b} &= -\boldsymbol{w^T}\boldsymbol{m} \\ \boldsymbol{m} &= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x_n} \\ &= \frac{1}{N}(n_1\boldsymbol{m_1} + n_2\boldsymbol{m_2}) \end{aligned}$$

Relation to Least Squares

$$\frac{\partial E}{\partial \boldsymbol{w}} = 0 \qquad\qquad \sum_{n=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x_n} + b - y_n\right)\boldsymbol{x_n} = 0$$

$$\boldsymbol{b} = -\boldsymbol{w}^T\boldsymbol{m} \quad \Downarrow \quad \boldsymbol{m} = \frac{1}{N}(n_1\boldsymbol{m_1} + n_2\boldsymbol{m_2})$$

$$\left(\boldsymbol{S}_w + \frac{n_1 n_2}{N}\boldsymbol{S}_b\right)\boldsymbol{w} = N(\boldsymbol{m_1} - \boldsymbol{m_2})$$

Leave for your homework

$\boldsymbol{S}_b\boldsymbol{w}$ is always in the direction of $(\boldsymbol{m_2} - \boldsymbol{m_1})$

$$w \propto \boldsymbol{S}_w^{-1}(\boldsymbol{m_2} - \boldsymbol{m_1})$$

Relation to Least Squares

This tells us that a new vector $\boldsymbol{x}$ should be classified as belonging to class C_1 if $y(\boldsymbol{x}) = \boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{m}) > 0$ and class C_2 otherwise.

For the two-class problem, LDA can be obtained as a special case of least squares.