COMP7180 - Lecture 4 Dimensionality Reduction (Feature Extraction) – Part II

Dr. LIU Yang

Some contents are from "J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org"

Singular Value Decomposition (SVD)

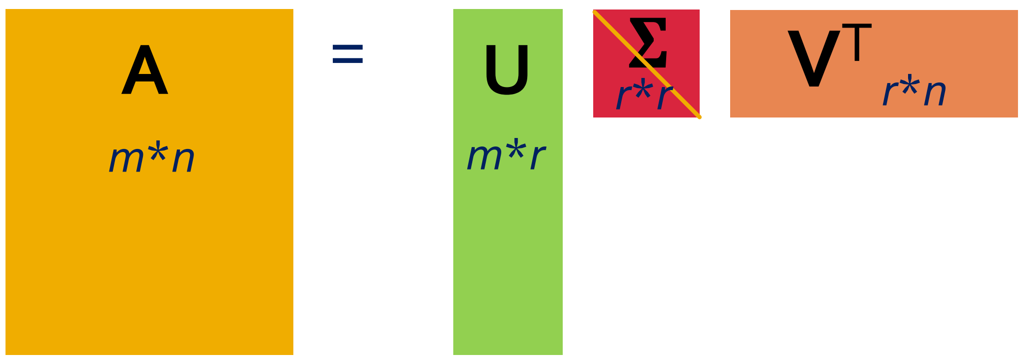
Singular Value Decomposition

- The eigenvalue decomposition requires square matrices. It would be useful to perform a decomposition on general matrices.

- The singular value decomposition (SVD) of a matrix is a central matrix decomposition method in linear algebra. It has been referred to as the "fundamental theorem of linear algebra" because it can be applied to all matrices, not only to square matrices, and it always exists.

SVD - Definition

- The Singular value decomposition (SVD) of a rectangular matrix $\mathbf{A}^{m*n}$



SVD - Properties

$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \Sigma_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

- **U** is the *m*r* matrix (**left singular matrix**) whose columns are orthonormal eigenvectors of **A*****A**^{T}: $\mathbf{A}\mathbf{A}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T$

SVD - Properties

$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \Sigma_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

- **V** is the $n*r$ matrix (**right singular matrix**) whose columns are orthonormal eigenvectors of $\mathbf{A}^T*\mathbf{A}$: $\mathbf{A}^T\mathbf{A} = \mathbf{V\Sigma U}^T\mathbf{U\Sigma V}^T = \mathbf{V\Sigma}^2\mathbf{V}^T$

SVD - Properties

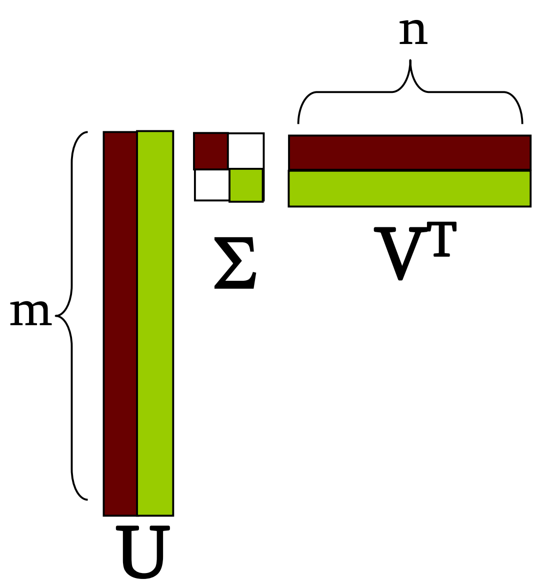
$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \Sigma_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

- **Σ** is an *r*r* diagonal matrix with non-negative numbers on the diagonal (These non-negative numbers are the square root of eigenvalues shared by $\mathbf{A}^*\mathbf{A}^T$ and $\mathbf{A}^T{}^*\mathbf{A}$) (also call **singular values** of matrix **A**)

SVD – Example: Users-to-Movies

- **$A = U\,\Sigma\,V^T$ - example: Users to Movies**

$$\begin{array}{cc} & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\ \text{Users} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \end{array} =$$



SVD – Example: Users-to-Movies

- $A = U \Sigma V^T$ - example: **Users to Movies**

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\text{Users} & \begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & 0.65 \\
0.07 & \mathbf{-0.73} & -0.67 \\
0.07 & \mathbf{-0.29} & 0.32
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & 1.3
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & -0.80 & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SVD – Example: Users-to-Movies

- $A = U \Sigma V^T$ - example: Users to Movies

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\text{Users} & \left[\begin{array}{ccccc}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{array}\right]
\end{array}
=
\left[\begin{array}{ccc}
\mathbf{0.13} & 0.02 & \cancel{-0.01} \\
\mathbf{0.41} & 0.07 & \cancel{-0.03} \\
\mathbf{0.55} & 0.09 & \cancel{-0.04} \\
\mathbf{0.68} & 0.11 & \cancel{-0.05} \\
0.15 & \mathbf{-0.59} & \cancel{0.65} \\
0.07 & \mathbf{-0.73} & \cancel{-0.67} \\
0.07 & \mathbf{-0.29} & \cancel{0.32}
\end{array}\right]
\times
\left[\begin{array}{ccc}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \cancel{1.3}
\end{array}\right]
\times
\left[\begin{array}{ccccc}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
\cancel{0.40} & \cancel{-0.80} & \cancel{0.40} & \cancel{0.09} & \cancel{0.09}
\end{array}\right]
$$

SVD – Example: Users-to-Movies

- $A = U \Sigma V^T$ - example:

U is "user-to-concept" similarity matrix

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\text{Users} & \begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
\approx
\begin{array}{c}
\begin{array}{cc} \text{SciFi-concept} & \text{Romance-concept} \end{array} \\
\begin{bmatrix}
\mathbf{0.13} & 0.02 \\
\mathbf{0.41} & 0.07 \\
\mathbf{0.55} & 0.09 \\
\mathbf{0.68} & 0.11 \\
0.15 & \mathbf{-0.59} \\
0.07 & \mathbf{-0.73} \\
0.07 & \mathbf{-0.29}
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 \\
0 & \mathbf{9.5}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69}
\end{bmatrix}
$$

SVD – Example: Users-to-Movies

- $A = U \Sigma V^T$ - example:

SciFi-concept

"strength" of the SciFi-concept

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\text{Users} & \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix}
\end{array}
\approx
\begin{bmatrix} \mathbf{0.13} & 0.02 \\ \mathbf{0.41} & 0.07 \\ \mathbf{0.55} & 0.09 \\ \mathbf{0.68} & 0.11 \\ 0.15 & \mathbf{-0.59} \\ 0.07 & \mathbf{-0.73} \\ 0.07 & \mathbf{-0.29} \end{bmatrix}
\times
\begin{bmatrix} \mathbf{12.4} & 0 \\ 0 & \mathbf{9.5} \end{bmatrix}
\times
\begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \end{bmatrix}
$$

SVD – Example: Users-to-Movies

- $A = U \Sigma V^T$ - example:

***V* is "movie-to-concept" similarity matrix**

SciFi-concept

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\text{Users} & \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix}
\end{array}
\approx
\begin{bmatrix} \mathbf{0.13} & 0.02 \\ \mathbf{0.41} & 0.07 \\ \mathbf{0.55} & 0.09 \\ \mathbf{0.68} & 0.11 \\ 0.15 & \mathbf{-0.59} \\ 0.07 & \mathbf{-0.73} \\ 0.07 & \mathbf{-0.29} \end{bmatrix}
\times
\begin{bmatrix} \mathbf{12.4} & 0 \\ 0 & \mathbf{9.5} \end{bmatrix}
\times
\begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \end{bmatrix}
$$

SciFi-concept

SVD - Interpretation

‘**movies**’, ‘**users**’ and ‘**concepts**’:

- ***U***: user-to-concept similarity matrix
- ***V***: movie-to-concept similarity matrix
- **Σ**: its diagonal elements:
  ‘strength’ of each concept

Case study: How to query?

- Q: Find users that like 'Matrix'
- A: Map query into a 'concept space' – how?

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{array}{c} \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & 0.65 \\
0.07 & \mathbf{-0.73} & -0.67 \\
0.07 & \mathbf{-0.29} & 0.32
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & 1.3
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & -0.80 & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

Case study: How to query?

- **Q: Find users that like 'Matrix'**
- **A: Map query into a 'concept space' – how?**

$$\mathbf{q} = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Project into concept space:
Inner product with each 'concept' vector $\mathbf{v}_i$

Case study: How to query?

Compactly, we have:

$$q_{concept} = q\,V$$

E.g.:

$$q = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \times \underset{\substack{\text{movie-to-concept} \\ \text{similarities (V)}}}{\begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix}} = \begin{array}{cc} \text{SciFi-concept} & \\ \downarrow & \\ \end{array} \begin{bmatrix} 2.8 & 0.6 \end{bmatrix}$$

Case study: How to query?

- **How would the user *d* that rated ('Alien', 'Serenity') be handled?**
  $\mathbf{d_{concept} = d\ V}$

E.g.:

$$
\begin{array}{ccccc}
\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}
\end{array}
$$

$$
\mathbf{d} = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \mathbf{x} \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{array}{cc} \text{SciFi-concept} & \\ \downarrow & \\ \end{array} \begin{bmatrix} 5.2 & 0.4 \end{bmatrix}
$$

movie-to-concept similarities (V)

Case study: How to query?

- **Observation:** User ***d*** that rated (‘*Alien*’, ‘*Serenity*’) will be **similar** to user **q** that rated (‘*Matrix*’), although ***d*** and **q** have **zero ratings in common**!

$$\begin{array}{rccccc} & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array}$$

$$\mathbf{d} = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{array}{cc} \text{SciFi-concept} & \\ \downarrow & \\ \end{array} \begin{bmatrix} 5.2 & 0.4 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{bmatrix} 2.8 & 0.6 \end{bmatrix}$$

Zero ratings in common

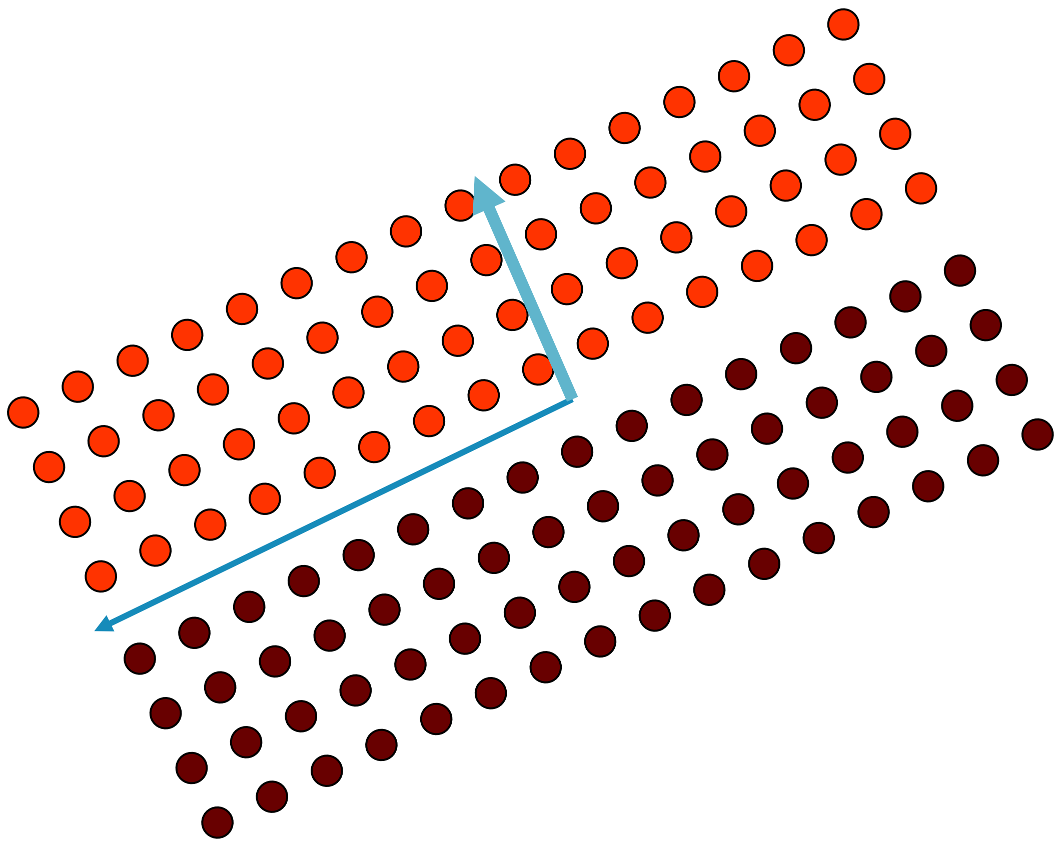
Similarity ≠ 0

Linear Discriminant Analysis (LDA)

The slides for this part are based on Dr. Guangliang Chen's slides at: https://www.sjsu.edu/faculty/guangliang.chen/Math251F18/lec3lda.pdf

Limitations of PCA

- PCA is not always an optimal dimensionality-reduction technique for classification purposes.

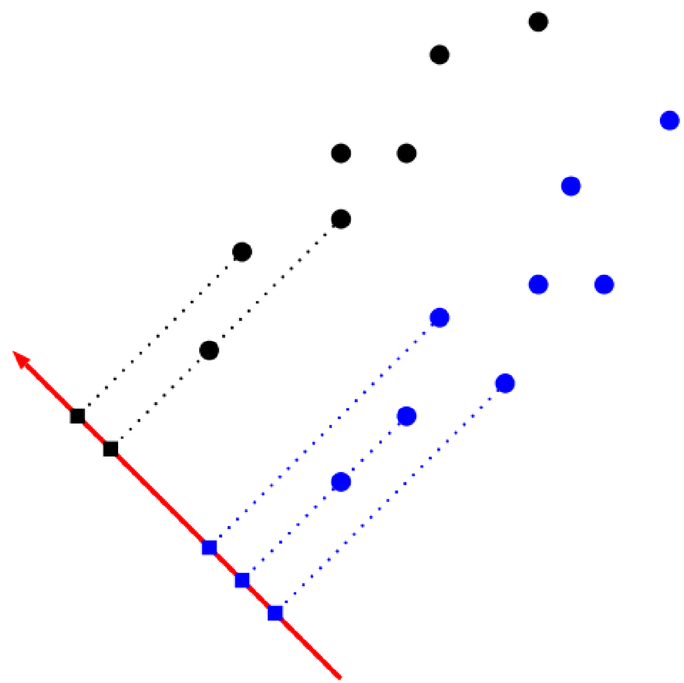


Objective of LDA

- Perform dimensionality reduction “while preserving as much of the class discriminative information as possible”.
- Seeks to find directions along which the classes are best separated, rather than the directions with the maximum variance.

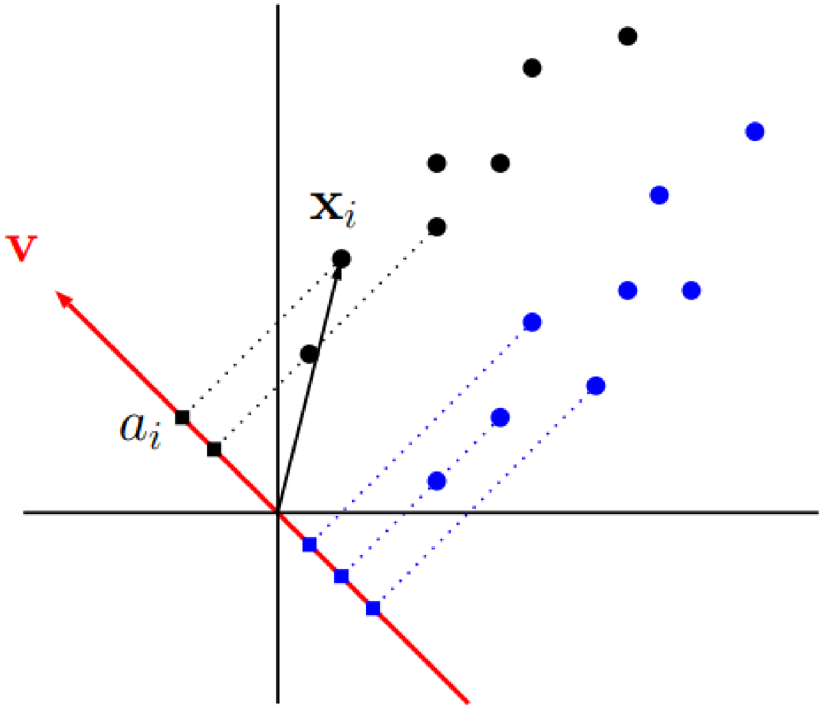
LDA: Two Classes

Given a training data set $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ consisting of two classes C_1, C_2, find a direction that "best" discriminates between the two classes.



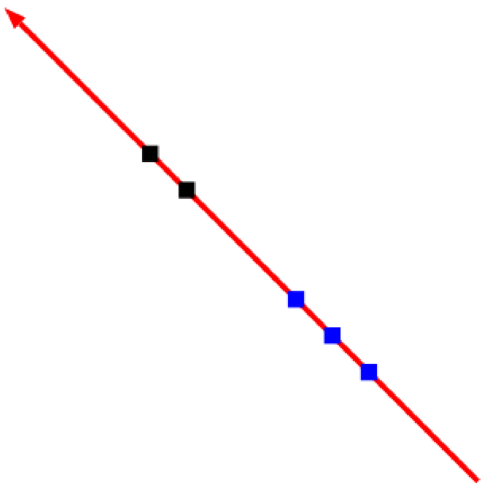
LDA: 1-D Projection

- Consider any unit vector $\mathbf{v} \in R^d$ as the projection direction
- The 1D projections of the points are:
  $a_i = \mathbf{v}^T\mathbf{x}_i$ (i = 1, . . . , n)



LDA: Initial Idea

Now the data look like this:



How do we quantify the separation between the two classes (in order to compare different directions $\mathbf{v}$ and select the best one)?

One (naive) idea is to measure the distance between the two class means in the 1D projection space: $|\mu_1 - \mu_2|$, where

$$\mu_1 = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} a_i = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} \mathbf{v}^T \mathbf{x}_i$$
$$= \mathbf{v}^T \cdot \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} \mathbf{x}_i = \mathbf{v}^T \mathbf{m}_1$$

and similarly,

$$\mu_2 = \mathbf{v}^T \mathbf{m}_2, \quad \mathbf{m}_2 = \frac{1}{n_2} \sum_{\mathbf{x}_i \in C_2} \mathbf{x}_i.$$

LDA: Problem of the Initial Idea

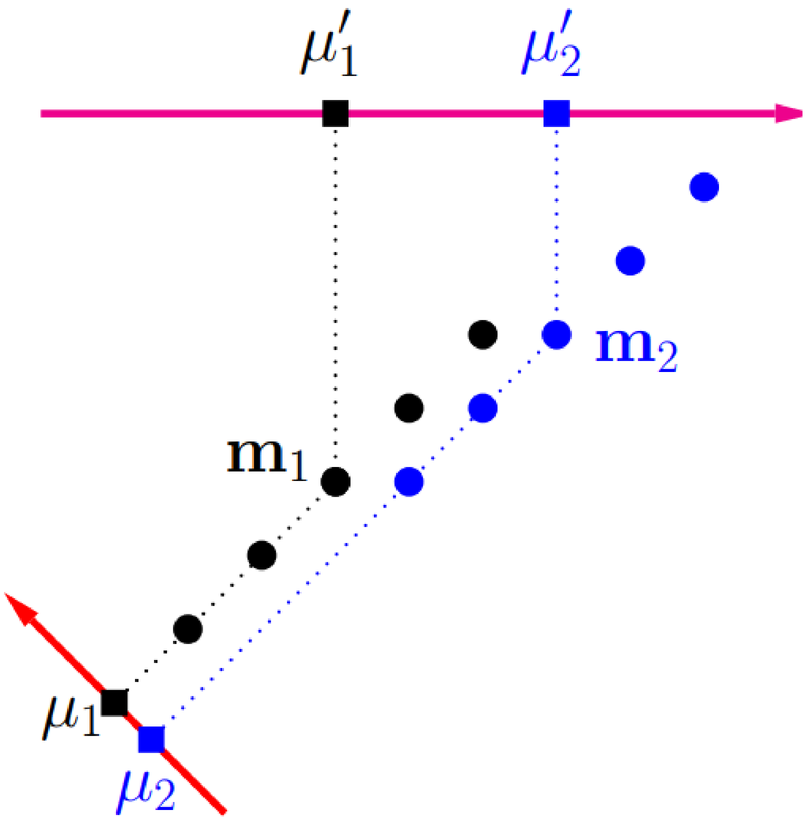
That is, we solve the following problem

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} |\mu_1 - \mu_2|$$

where

$$\mu_j = \mathbf{v}^T \mathbf{m}_j, \ j = 1, 2.$$

However, this criterion does not always work (as shown in the right plot).



What else do we need to consider?

LDA: Further Considerations

We should also consider the **variance** of each projected class:

$$s_1^2 = \sum_{\mathbf{x}_i \in C_1} (a_i - \mu_1)^2, \quad s_2^2 = \sum_{\mathbf{x}_i \in C_2} (a_i - \mu_2)^2$$

Ideally, the projected classes have both **faraway means** and **small variances**.

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}.$$

where $\mu_1 = \mathbf{v}^T \mathbf{m}_1, \quad \mu_2 = \mathbf{v}^T \mathbf{m}_2.$

LDA: Mathematical Derivation

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}.$$

where $\quad \mu_1 = \mathbf{v}^T \mathbf{m}_1, \quad \mu_2 = \mathbf{v}^T \mathbf{m}_2.$

First, we can rewrite the distance between the two centroids as follows:

$$\begin{aligned}
(\mu_1 - \mu_2)^2 &= (\mathbf{v}^T \mathbf{m}_1 - \mathbf{v}^T \mathbf{m}_2)^2 = (\mathbf{v}^T (\mathbf{m}_1 - \mathbf{m}_2))^2 \\
&= \mathbf{v}^T (\mathbf{m}_1 - \mathbf{m}_2) \cdot (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{v} \\
&= \mathbf{v}^T \mathbf{S}_b \mathbf{v},
\end{aligned}$$

where

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \in \mathbb{R}^{d \times d}$$

is called the **between-class scatter matrix**.

LDA: Mathematical Derivation

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}.$$

where $\quad \mu_1 = \mathbf{v}^T \mathbf{m}_1, \quad \mu_2 = \mathbf{v}^T \mathbf{m}_2.$

Next, for each class $j = 1, 2$, the variance of the projection (onto $\mathbf{v}$) is

$$\begin{aligned}
s_j^2 = \sum_{\mathbf{x}_i \in C_j} (a_i - \mu_j)^2 &= \sum_{\mathbf{x}_i \in C_j} (\mathbf{v}^T \mathbf{x}_i - \mathbf{v}^T \mathbf{m}_j)^2 \\
&= \sum_{\mathbf{x}_i \in C_j} \mathbf{v}^T (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^T \mathbf{v} \\
&= \mathbf{v}^T \left[\sum_{\mathbf{x}_i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^T \right] \mathbf{v} \\
&= \mathbf{v}^T \mathbf{S}_j \mathbf{v},
\end{aligned}$$

where

$$\mathbf{S}_j = \sum_{\mathbf{x}_i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^T \in \mathbb{R}^{d \times d}$$

is called the **within-class scatter matrix** for class j.

LDA: Mathematical Derivation

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}.$$

where $\quad \mu_1 = \mathbf{v}^T \mathbf{m}_1, \quad \mu_2 = \mathbf{v}^T \mathbf{m}_2.$

The total within-class scatter of the two classes in the projection space is

$$s_1^2 + s_2^2 = \mathbf{v}^T \mathbf{S}_1 \mathbf{v} + \mathbf{v}^T \mathbf{S}_2 \mathbf{v} = \mathbf{v}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{v} = \mathbf{v}^T \mathbf{S}_w \mathbf{v}$$

where

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 = \sum_{\mathbf{x}_i \in C_1} (\mathbf{x}_i - \mathbf{m}_1)(\mathbf{x}_i - \mathbf{m}_1)^T + \sum_{\mathbf{x}_i \in C_2} (\mathbf{x}_i - \mathbf{m}_2)(\mathbf{x}_i - \mathbf{m}_2)^T$$

is called the **total within-class scatter matrix** of the (original) training data.

Putting everything together, we have arrived at the following optimization problem:

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{\mathbf{v}^T \mathbf{S}_b \mathbf{v}}{\mathbf{v}^T \mathbf{S}_w \mathbf{v}}$$

LDA: Mathematical Derivation

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \frac{\mathbf{v}^T\mathbf{S}_b\mathbf{v}}{\mathbf{v}^T\mathbf{S}_w\mathbf{v}}$$

Let $J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_b\mathbf{v}}{\mathbf{v}^T\mathbf{S}_w\mathbf{v}}$, then $\frac{d}{d\mathbf{v}}J(\mathbf{v}) = 0$

$$\Longrightarrow \quad (\mathbf{v}^T\mathbf{S}_w\mathbf{v})\frac{d}{d\mathbf{v}}(\mathbf{v}^T\mathbf{S}_b\mathbf{v}) - (\mathbf{v}^T\mathbf{S}_b\mathbf{v})\frac{d}{d\mathbf{v}}(\mathbf{v}^T\mathbf{S}_w\mathbf{v}) = 0$$

$$\Longrightarrow \quad (\mathbf{v}^T\mathbf{S}_w\mathbf{v})2\mathbf{S}_b\mathbf{v} - (\mathbf{v}^T\mathbf{S}_b\mathbf{v})\mathbf{2}\mathbf{S}_w\mathbf{v} = 0$$

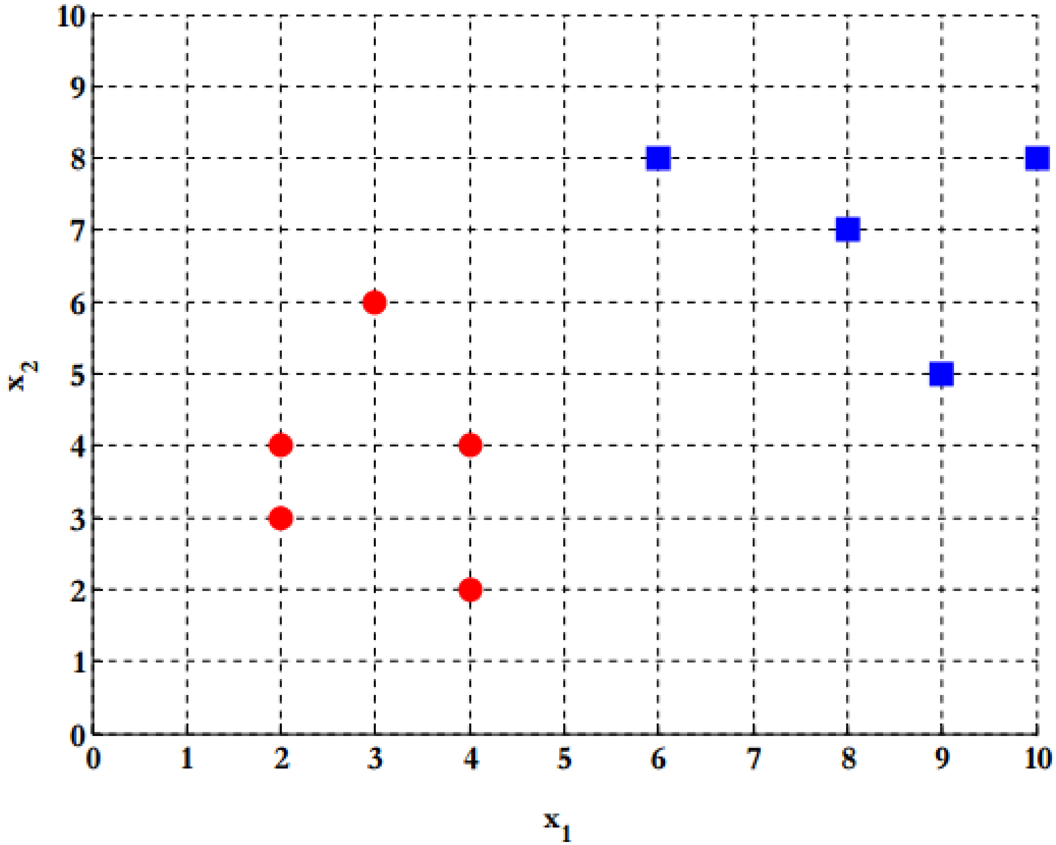
$$\Longrightarrow \quad \left(\frac{\mathbf{v}^T\mathbf{S}_w\mathbf{v}}{\mathbf{v}^T\mathbf{S}_w\mathbf{v}}\right)\mathbf{S}_b\mathbf{v} - \left(\frac{\mathbf{v}^T\mathbf{S}_b\mathbf{v}}{\mathbf{v}^T\mathbf{S}_w\mathbf{v}}\right)\mathbf{S}_w\mathbf{v} = 0$$

$$\Longrightarrow \quad \mathbf{S}_b\mathbf{v} - J(\mathbf{v})\mathbf{S}_w\mathbf{v} = 0 \quad \Longrightarrow \quad \left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\mathbf{v} = [J(\mathbf{v})]\,\mathbf{v}$$

$\mathbf{v}$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}_w^{-1}\mathbf{S}_b$

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

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

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

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

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

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

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

LDA: Two Classes - Example

- 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

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

Thus;

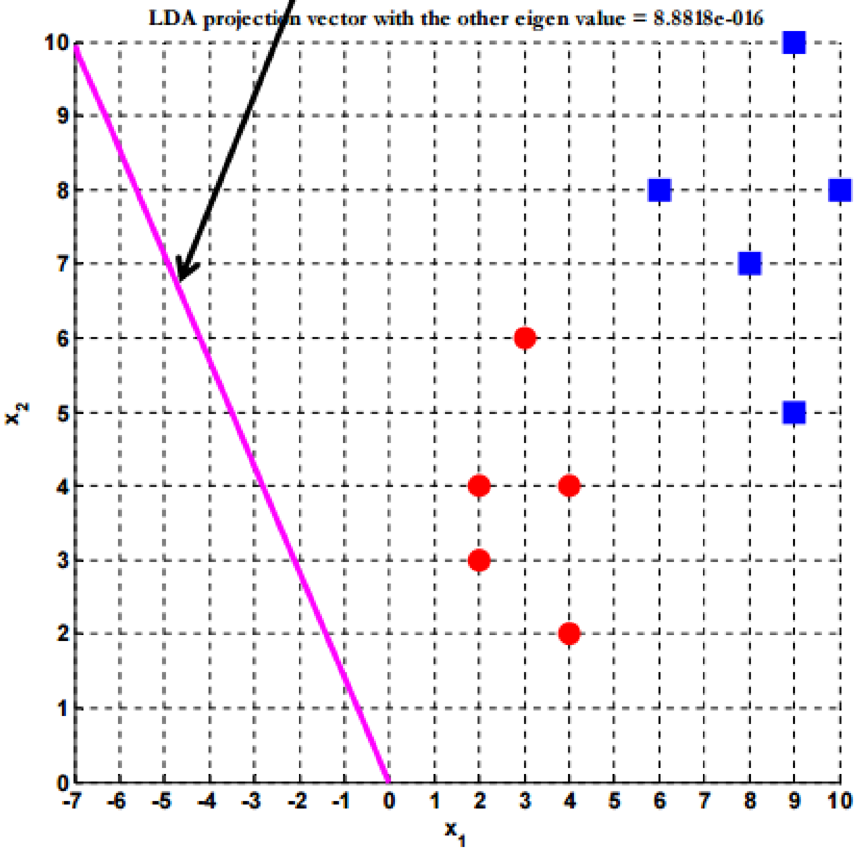
$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \quad \text{and} \quad \boxed{w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*}$$

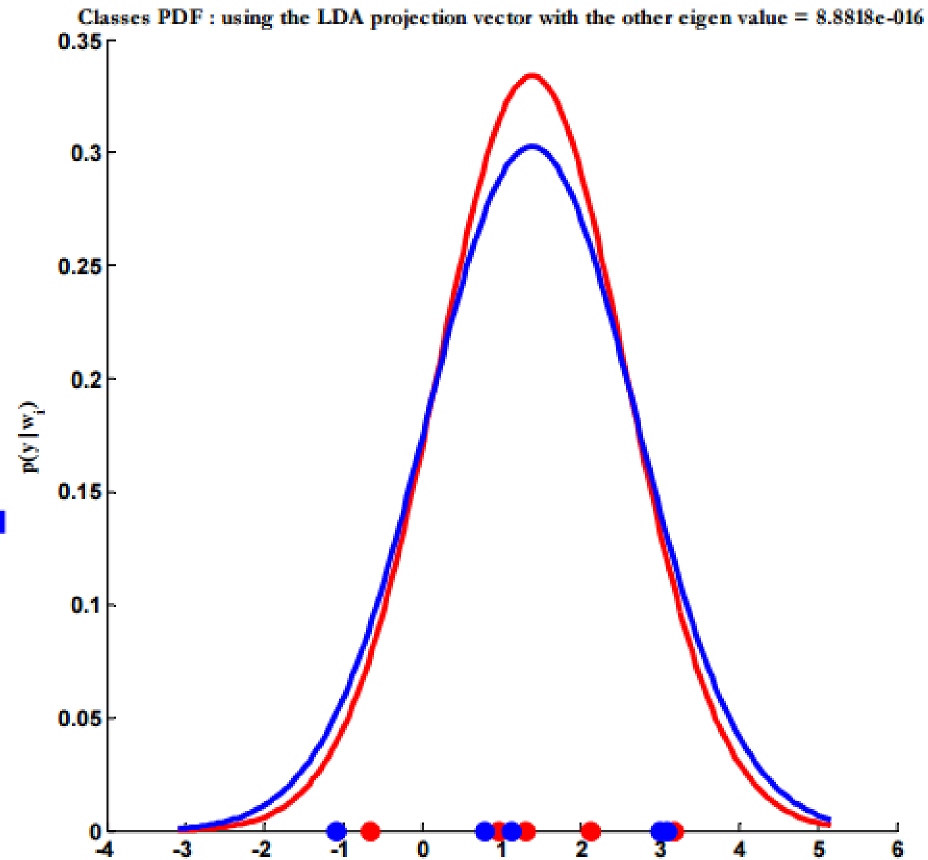
- The optimal projection is the one that given maximum $\lambda = J(w)$

```
% 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);
```

LDA: Two Classes - Example

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

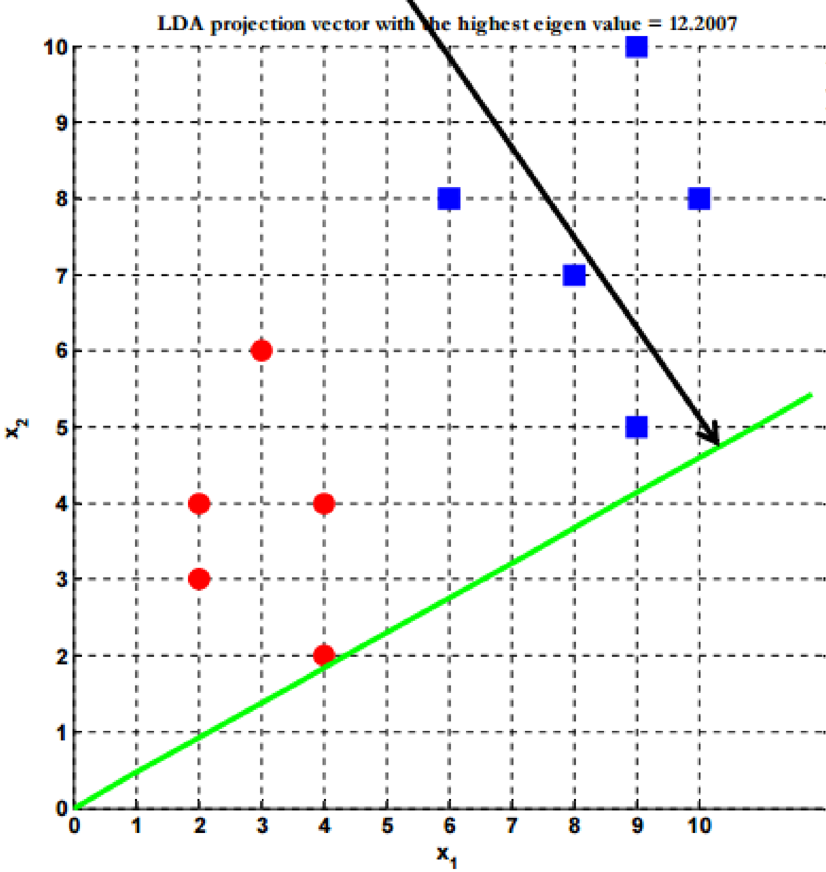


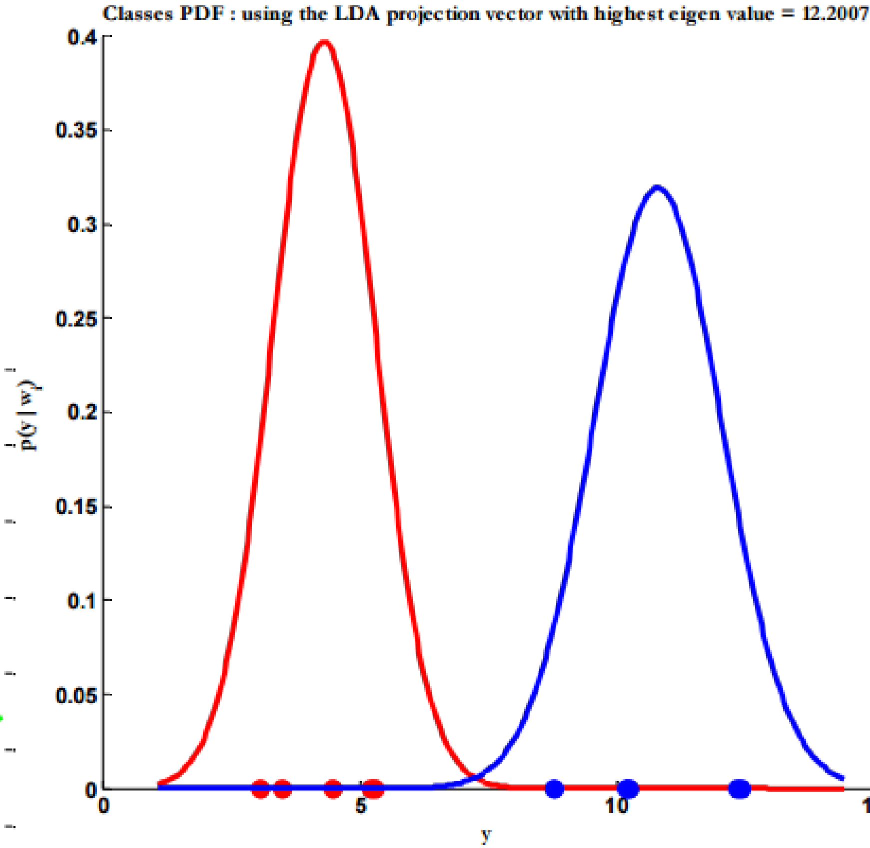


Using this vector leads to **bad separability** between the two classes

LDA: Two Classes - Example

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





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

References and Acknowledgement

- Slide from Jure Leskovec, Stanford CS246: Mining Massive Datasets https://web.stanford.edu/class/cs246/slides/06-dim_red.pdf

- Slide from Guangliang Chen, San José State University, Math 253: Mathematical Methods for Data Visualization https://www.sjsu.edu/faculty/guangliang.chen/Math253S20/lec11lda.pdf

- Slide from Shireen Elhabian and Aly A. Fara, University of Louisville, A Tutorial on Data Reduction https://www.sci.utah.edu/~shireen/pdfs/tutorials/Elhabian_LDA09.pdf

- https://slideplayer.com/slide/14520684/