Imperial College London

Lecture 15: Linear Discriminant Analysis

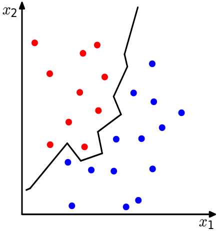
Recommended reading:
Bishop, Chapter 4.1
Hastie et al., Chapter 4.3

Duncan Gillies and Marc Deisenroth

Department of Computing
Imperial College London

February 22, 2016

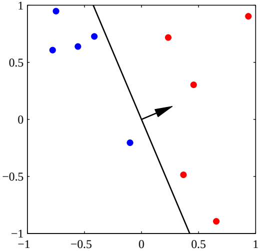
Classification



Adapted from PRML (Bishop, 2006)

- Input vector $\boldsymbol{x} \in \mathbb{R}^D$, assign it to one of K discrete classes $C_k, k = 1, \ldots, K$.
- Assumption: classes are disjoint, i.e., input vectors are assigned to exactly one class
- Idea: Divide input space into decision regions whose boundaries are called decision boundaries/surfaces

Linear Classification

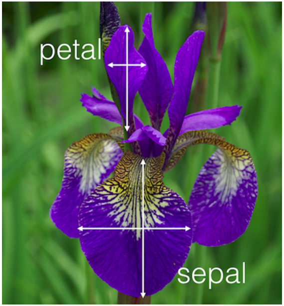


From PRML (Bishop, 2006)

- Focus on linear classification model, i.e., the decision boundary is a linear function of $\boldsymbol{x}$
  - ▸▸ Defined by $(D-1)$-dimensional hyperplane
- If the data can be separated exactly by linear decision surfaces, they are called linearly separable
- Implicit assumption: Classes can be modeled well by Gaussians

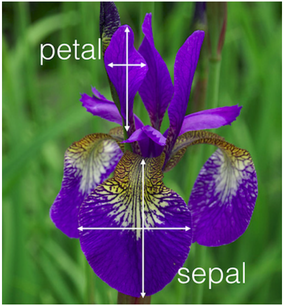
▸▸ Here: Treat classification as a projection problem

Example



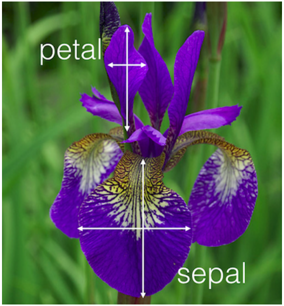
- Measurements for 150 Iris flowers from three different species.

Example



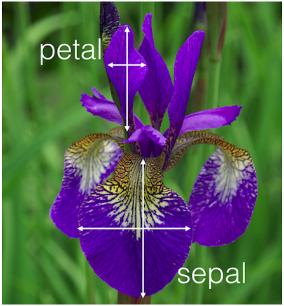
- Measurements for 150 Iris flowers from three different species.
- Four features (petal length/width, sepal length/width)

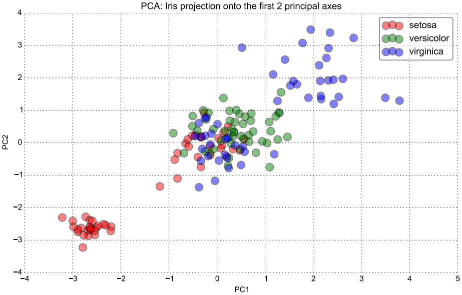
Example



- Measurements for 150 Iris flowers from three different species.
- Four features (petal length/width, sepal length/width)
- Given a new measurement of these features, predict the Iris species based on a projection onto a low-dimensional space.

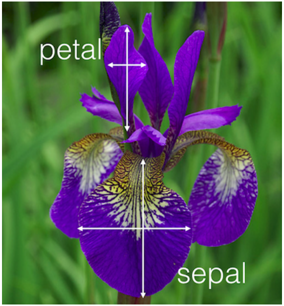
Example

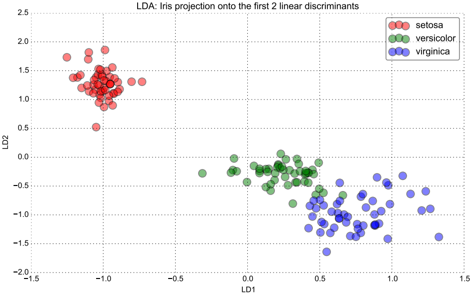




- Measurements for 150 Iris flowers from three different species.
- Four features (petal length/width, sepal length/width)
- Given a new measurement of these features, predict the Iris species based on a projection onto a low-dimensional space.
- PCA may not be ideal to separate the classes well

Example





- Measurements for 150 Iris flowers from three different species.
- Four features (petal length/width, sepal length/width)
- Given a new measurement of these features, predict the Iris species based on a projection onto a low-dimensional space.
- PCA may not be ideal to separate the classes well

Orthogonal Projections (Repetition)

- Project input vector $\boldsymbol{x} \in \mathbb{R}^D$ down to a 1-dimensional subspace with basis vector $\boldsymbol{w}$
- With $\|\boldsymbol{w}\| = 1$, we get

$\boldsymbol{P} = \boldsymbol{w}\boldsymbol{w}^\top$	Projection matrix, such that $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{p}$
$\boldsymbol{p} = y\boldsymbol{w} \in \mathbb{R}^D$	Projection point ⏩ Discussed in Lecture 14
$y = \boldsymbol{w}^\top \boldsymbol{x} \in \mathbb{R}$	Coordinates with respect to basis $\boldsymbol{w}$ ⏩ Today

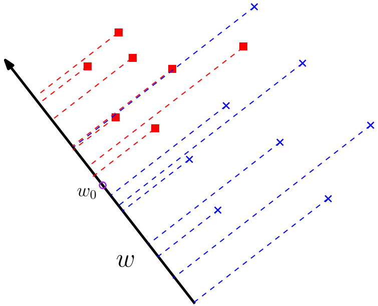
Orthogonal Projections (Repetition)

- Project input vector $\boldsymbol{x} \in \mathbb{R}^D$ down to a 1-dimensional subspace with basis vector $\boldsymbol{w}$
- With $\|\boldsymbol{w}\| = 1$, we get

$\boldsymbol{P} = \boldsymbol{w}\boldsymbol{w}^\top$	Projection matrix, such that $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{p}$
$\boldsymbol{p} = y\boldsymbol{w} \in \mathbb{R}^D$	Projection point ⏩ Discussed in Lecture 14
$y = \boldsymbol{w}^\top\boldsymbol{x} \in \mathbb{R}$	Coordinates with respect to basis $\boldsymbol{w}$ ⏩ Today

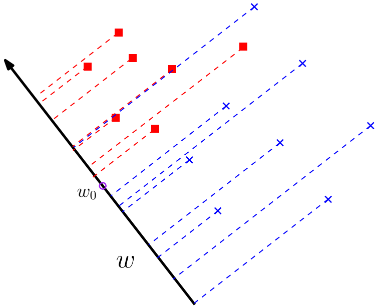
- We will largely focus on the coordinates y in the following
- Projection points equally apply to concepts discussed today
- Coordinates equally apply to PCA (see Lecture 14)

Classification as Projection



- Assume we know the basis vector $\boldsymbol{w}$, we can compute the projection of any point $\boldsymbol{x} \in \mathbb{R}^D$ onto the one-dimensional subspace spanned by $\boldsymbol{w}$

Classification as Projection



- Assume we know the basis vector $\boldsymbol{w}$, we can compute the projection of any point $\boldsymbol{x} \in \mathbb{R}^D$ onto the one-dimensional subspace spanned by $\boldsymbol{w}$
- Threshold w_0, such that we decide on C_1 if $y \geqslant w_0$ and C_2 otherwise

The Linear Decision Boundary of LDA

- Look at the log-probability ratio

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = \log \frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} + \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

  where the decision boundary (for $\mathcal{C}_1$ or $\mathcal{C}_2$) is at 0.

The Linear Decision Boundary of LDA

- Look at the log-probability ratio

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = \log \frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} + \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

  where the decision boundary (for $\mathcal{C}_1$ or $\mathcal{C}_2$) is at 0.
- Assume Gaussian likelihood $p(\boldsymbol{x}|\mathcal{C}_i) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{m}_i, \boldsymbol{\Sigma}\big)$ with the same covariance in both classes. Decision boundary:

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = 0$$

The Linear Decision Boundary of LDA

- Look at the log-probability ratio
$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = \log \frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} + \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$
where the decision boundary (for $\mathcal{C}_1$ or $\mathcal{C}_2$) is at 0.
- Assume Gaussian likelihood $p(\boldsymbol{x}|\mathcal{C}_i) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{m}_i, \boldsymbol{\Sigma}\big)$ with the same covariance in both classes. Decision boundary:

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = 0$$
$$\Leftrightarrow \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} - \frac{1}{2}(\boldsymbol{m}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{m}_1 - \boldsymbol{m}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{m}_2) + (\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} = 0$$

The Linear Decision Boundary of LDA

- Look at the log-probability ratio
$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = \log \frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} + \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$
where the decision boundary (for $\mathcal{C}_1$ or $\mathcal{C}_2$) is at 0.
- Assume Gaussian likelihood $p(\boldsymbol{x}|\mathcal{C}_i) = \mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{m}_i, \boldsymbol{\Sigma}\big)$ with the same covariance in both classes. Decision boundary:

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = 0$$
$$\Leftrightarrow \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} - \frac{1}{2}(\boldsymbol{m}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{m}_1 - \boldsymbol{m}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{m}_2) + (\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} = 0$$
$$\Leftrightarrow (\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} = \frac{1}{2}(\boldsymbol{m}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{m}_1 - \boldsymbol{m}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{m}_2) - \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

The Linear Decision Boundary of LDA

- Look at the log-probability ratio

$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = \log \frac{p(\boldsymbol{x}|\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)} + \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

  where the decision boundary (for $\mathcal{C}_1$ or $\mathcal{C}_2$) is at 0.
- Assume Gaussian likelihood $p(\boldsymbol{x}|\mathcal{C}_i) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{m}_i, \boldsymbol{\Sigma}\big)$ with the same covariance in both classes. Decision boundary:

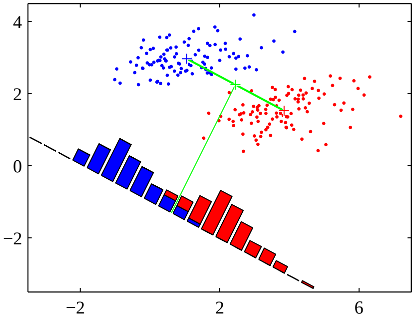
$$\log \frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})} = 0$$

$$\Leftrightarrow \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} - \frac{1}{2}(\boldsymbol{m}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{m}_1 - \boldsymbol{m}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{m}_2) + (\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} = 0$$

$$\Leftrightarrow (\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} = \frac{1}{2}(\boldsymbol{m}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{m}_1 - \boldsymbol{m}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{m}_2) - \log \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

▶▶ Of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ ▶▶ Decision boundary linear in $\boldsymbol{x}$

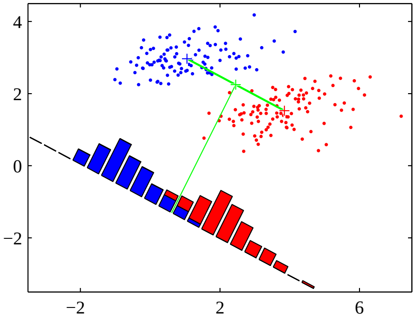
Potential Issues



From PRML (Bishop, 2006)

- Considerable loss of information when projecting

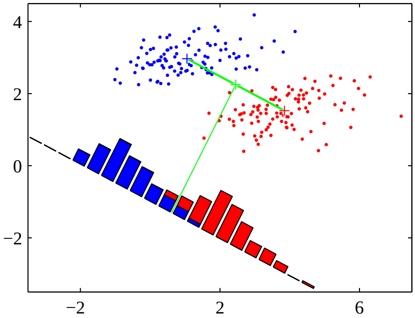
Potential Issues



From PRML (Bishop, 2006)

- Considerable loss of information when projecting
- Even if data was linearly separable in $\mathbb{R}^D$, we may lose this separability (see figure)

Potential Issues



From PRML (Bishop, 2006)

- Considerable loss of information when projecting
- Even if data was linearly separable in $\mathbb{R}^D$, we may lose this separability (see figure)

⏩ Find good basis vector $\boldsymbol{w}$ that spans the subspace we project onto

Approach: Maximize Class Separation

- Adjust components of basis vector $\boldsymbol{w}$
  - ⏩ Select projection that maximizes the class separation

Approach: Maximize Class Separation

- Adjust components of basis vector $\boldsymbol{w}$
  - Select projection that maximizes the class separation
- Consider two classes: C_1 with N_1 points and C_2 with N_2 points

Approach: Maximize Class Separation

- Adjust components of basis vector $\boldsymbol{w}$
  ⏩ Select projection that maximizes the class separation
- Consider two classes: C_1 with N_1 points and C_2 with N_2 points
- Corresponding mean vectors:

$$\boldsymbol{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \boldsymbol{x}_n \,, \qquad \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \boldsymbol{x}_n$$

Approach: Maximize Class Separation

- Adjust components of basis vector $\boldsymbol{w}$

  ▶▶ Select projection that maximizes the class separation
- Consider two classes: C_1 with N_1 points and C_2 with N_2 points
- Corresponding mean vectors:

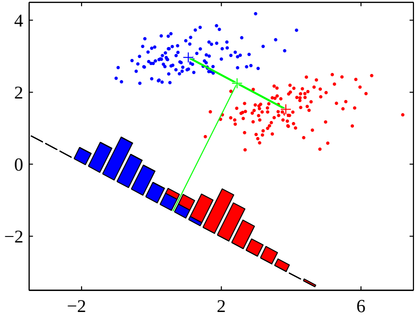
$$\boldsymbol{m}_1 = \frac{1}{N_1}\sum_{n\in C_1}\boldsymbol{x}_n\,, \qquad \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{n\in C_2}\boldsymbol{x}_n$$

- Measure class separation as the distance of the projected class means:

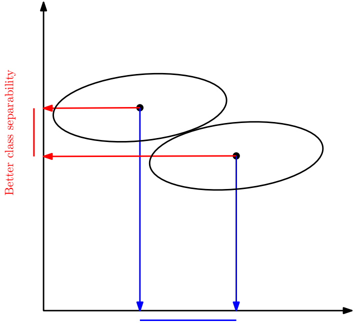
$$m_2 - m_1 = \boldsymbol{w}^\top\boldsymbol{m}_2 - \boldsymbol{w}^\top\boldsymbol{m}_1 = \boldsymbol{w}^\top(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

  and maximize this w.r.t. $\boldsymbol{w}$ with the constraint $\|\boldsymbol{w}\| = 1$

Maximum Class Separation

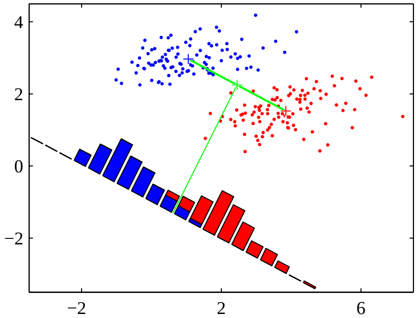


From PRML (Bishop, 2006)

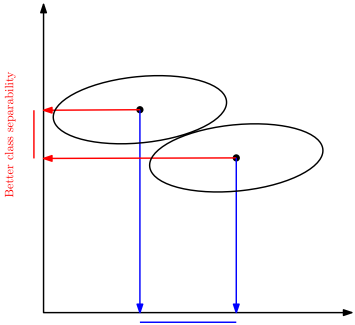


- Find $\boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1)$
- Projected classes may still have considerable overlap (because of strongly non-diagonal covariances of the class distributions)

Maximum Class Separation

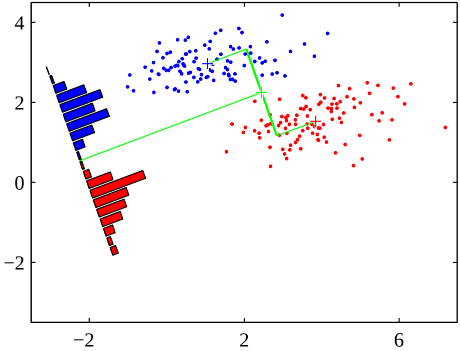


From PRML (Bishop, 2006)



- Find $\boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1)$
- Projected classes may still have considerable overlap (because of strongly non-diagonal covariances of the class distributions)
- LDA: Large separation of projected class means **and** small within-class variation (small overlap of classes)

Key Idea of LDA



From PRML (Bishop, 2006)

- Separate samples of distinct groups by projecting them onto a space that
  - Maximizes their between-class separability while
  - Minimizing their within-class variability

Fisher Criterion

- For each class C_k the within-class scatter (unnormalized variance) is given as

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 , \qquad y_n = \boldsymbol{w}^\top \boldsymbol{x}_n , \quad m_k = \boldsymbol{w}^\top \boldsymbol{m}_k$$

Fisher Criterion

- For each class C_k the within-class scatter (unnormalized variance) is given as

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 \,, \qquad y_n = \boldsymbol{w}^\top \boldsymbol{x}_n \,, \quad m_k = \boldsymbol{w}^\top \boldsymbol{m}_k$$

- Maximize the Fisher criterion:

$$J(\boldsymbol{w}) = \frac{\text{Between-class scatter}}{\text{Within-class scatter}} = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

$$\boldsymbol{S}_W = \sum_k \sum_{n \in C_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^\top$$

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^\top$$

Fisher Criterion

- For each class C_k the within-class scatter (unnormalized variance) is given as

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 \,, \qquad y_n = \boldsymbol{w}^\top \boldsymbol{x}_n \,, \quad m_k = \boldsymbol{w}^\top \boldsymbol{m}_k$$

- Maximize the Fisher criterion:

$$J(\boldsymbol{w}) = \frac{\text{Between-class scatter}}{\text{Within-class scatter}} = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

$$\boldsymbol{S}_W = \sum_k \sum_{n \in C_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^\top$$

$$\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^\top$$

- $\boldsymbol{S}_W$ is the total within-class scatter and proportional to the sample covariance matrix

Generalization to k Classes

For k classes, we define the between-class scatter matrix as

$$\boldsymbol{S}_B = \sum_k N_k(\boldsymbol{m}_k - \boldsymbol{\mu})(\boldsymbol{m}_2 - \boldsymbol{\mu})^\top, \qquad \boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

where $\boldsymbol{\mu}$ is the global mean of the data set

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$dJ/d\boldsymbol{w} = \boldsymbol{0} \Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right)\boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right)\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0}$$

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$\begin{aligned} dJ/d\boldsymbol{w} = \boldsymbol{0} &\Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right)\boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right)\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \end{aligned}$$

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$\begin{aligned} dJ/d\boldsymbol{w} = \boldsymbol{0} &\Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right)\boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right)\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_W^{-1}\boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{w} = \boldsymbol{0} \end{aligned}$$

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$\begin{aligned} dJ/d\boldsymbol{w} = \boldsymbol{0} &\Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right) \boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right) \boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_B \boldsymbol{w} - J \boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_W^{-1} \boldsymbol{S}_B \boldsymbol{w} - J \boldsymbol{w} = \boldsymbol{0} \end{aligned}$$

Eigenvalue problem $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B \boldsymbol{w} = J \boldsymbol{w}$

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$\begin{aligned} dJ/d\boldsymbol{w} = \boldsymbol{0} &\Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right)\boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right)\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_W^{-1}\boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{w} = \boldsymbol{0} \end{aligned}$$

- **Eigenvalue problem** $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B \boldsymbol{w} = J\boldsymbol{w}$
- The projection vector $\boldsymbol{w}$ is the eigenvector of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$.

Finding the Projection

Objective

Find $\boldsymbol{w}^*$ that maximizes

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

We find $\boldsymbol{w}$ by setting $dJ/d\boldsymbol{w} = \boldsymbol{0}$:

$$\begin{aligned} dJ/d\boldsymbol{w} = \boldsymbol{0} &\Leftrightarrow \left(\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}\right)\boldsymbol{S}_B \boldsymbol{w} - \left(\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}\right)\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{S}_W \boldsymbol{w} = \boldsymbol{0} \\ &\Leftrightarrow \boldsymbol{S}_W^{-1}\boldsymbol{S}_B \boldsymbol{w} - J\boldsymbol{w} = \boldsymbol{0} \end{aligned}$$

- **Eigenvalue problem** $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B \boldsymbol{w} = J\boldsymbol{w}$
- The projection vector $\boldsymbol{w}$ is the eigenvector of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$.
- Choose the eigenvector that corresponds to the maximum eigenvalue (similar to PCA) to maximize class separability

Algorithm

1. Mean normalization

- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample

Algorithm

1. Mean normalization
2. Compute mean vectors $\boldsymbol{m}_i \in \mathbb{R}^D$ for all k classes

- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample

Algorithm

1. Mean normalization
2. Compute mean vectors $\boldsymbol{m}_i \in \mathbb{R}^D$ for all k classes
3. Compute scatter matrices $\boldsymbol{S}_W, \boldsymbol{S}_B$

- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample

Algorithm

1. Mean normalization
2. Compute mean vectors $\boldsymbol{m}_i \in \mathbb{R}^D$ for all k classes
3. Compute scatter matrices $\boldsymbol{S}_W, \boldsymbol{S}_B$
4. Compute eigenvectors and eigenvalues of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$

- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample

Algorithm

1. Mean normalization
2. Compute mean vectors $\boldsymbol{m}_i \in \mathbb{R}^D$ for all k classes
3. Compute scatter matrices $\boldsymbol{S}_W, \boldsymbol{S}_B$
4. Compute eigenvectors and eigenvalues of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$
5. Select k eigenvectors $\boldsymbol{w}_i$ with the largest eigenvalues to form a $D \times k$-dimensional matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$

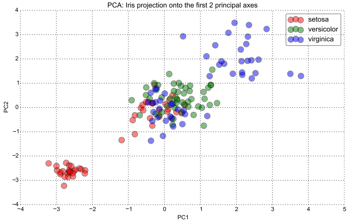
- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample

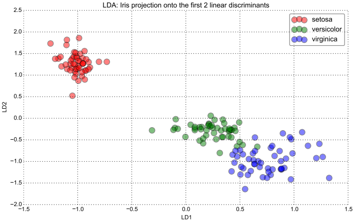
Algorithm

1. Mean normalization
2. Compute mean vectors $\boldsymbol{m}_i \in \mathbb{R}^D$ for all k classes
3. Compute scatter matrices $\boldsymbol{S}_W, \boldsymbol{S}_B$
4. Compute eigenvectors and eigenvalues of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$
5. Select k eigenvectors $\boldsymbol{w}_i$ with the largest eigenvalues to form a $D \times k$-dimensional matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k]$
6. Project samples onto the new subspace using $\boldsymbol{W}$ and compute the new coordinates as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}$

- $\boldsymbol{X} \in \mathbb{R}^{n \times D}$: ith row represents the ith sample
- $\boldsymbol{Y} \in \mathbb{R}^{n \times k}$: Coordinate matrix of the n data points w.r.t. eigenbasis $\boldsymbol{W}$ spanning the k-dimensional subspace

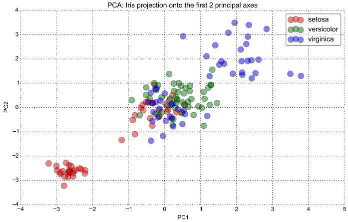
PCA vs LDA

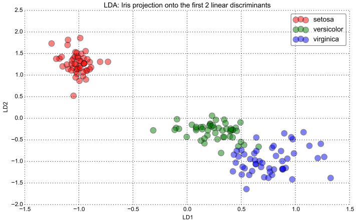




- Similar to PCA, we can use LDA for dimensionality reduction by looking at an eigenvalue problem

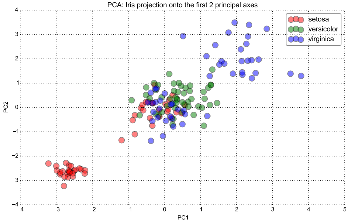
PCA vs LDA

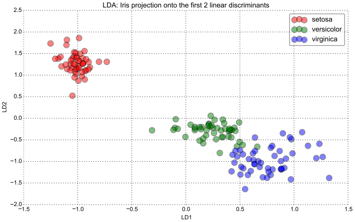




- Similar to PCA, we can use LDA for dimensionality reduction by looking at an eigenvalue problem
- LDA: Magnitude of the eigenvalues in LDA describe importance of the corresponding eigenspace with respect to classification performance

PCA vs LDA





- Similar to PCA, we can use LDA for dimensionality reduction by looking at an eigenvalue problem
- LDA: Magnitude of the eigenvalues in LDA describe importance of the corresponding eigenspace with respect to classification performance
- PCA: Magnitude of the eigenvalues in LDA describe importance of the corresponding eigenspace with respect to minimizing reconstruction error

Assumptions in LDA

- The true covariance matrices of each class are equal

Assumptions in LDA

- The true covariance matrices of each class are equal
- Without this assumption: Quadratic discriminant analysis (e.g. Hastie et al., 2009)

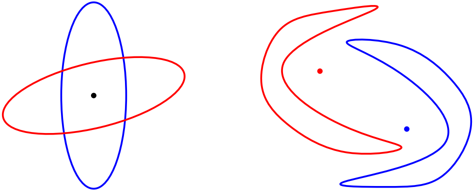
Assumptions in LDA

- The true covariance matrices of each class are equal
- Without this assumption: Quadratic discriminant analysis (e.g. Hastie et al., 2009)
- Performance of the standard LDA can be seriously degraded if there are only a limited number of total training observations N compared to the dimension D of the feature space.
  ▶▶ Shrinkage (Copas, 1983)

Assumptions in LDA

- The true covariance matrices of each class are equal
- Without this assumption: Quadratic discriminant analysis (e.g. Hastie et al., 2009)
- Performance of the standard LDA can be seriously degraded if there are only a limited number of total training observations N compared to the dimension D of the feature space.
  ⏩ Shrinkage (Copas, 1983)
- LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class

Limitations of LDA



- LDA's most disriminant features are the means of the data distributions
- LDA will fail when the discriminatory information is not the mean but the variance of the data.
- If the data distributions are very non-Gaussian, the LDA projections will not preserve the complex structure of the data that may be required for classification

⏩ Nonlinear LDA (e.g., Mika et al., 1999; Baudat & Anouar, 2000)

References I

[1] G. Baudat and F. Anouar. Generalized Discriminant Analysis using a Kernel Approach. *Neural Computation*, 12(10):2385–2404, 2000.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, 2006.

[3] J. B. Copas. Regression, Prediction and Shrinkage. *Journal of the Royal Statistical Society, Series B*, 45(3):311–354, 1983.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning—Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag New York, Inc., 175 Fifth Avenue, New York City, NY, USA, 2001.

[5] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher Discriminant Analysis with Kernels. *Neural Networks for Signal Processing*, IX:41–48, 1999.