Classical & Linear Methods #2: Classification & Unsupervised Learning

Seul-Ki Yeom
Technische Universität Berlin - Machine Learning Group
Beginners Machine Learning Workshop 2019

Agenda

Principal Component Analysis

Linear Classification & LDA

Agenda

Principal Component Analysis

Linear Classification & LDA

Unsupervised learning

Supervised algorithms

- Generate a function that maps input data into output data
- Classification and regression problems
- Labels for training

Unsupervised learning

Supervised algorithms

- Generate a function that maps input data into output data
- Classification and regression problems
- Labels for training

Often there is no label information available

- Ongoing neural activity
- Mixtures of different speakers in an audio recording
- Complex artefacts in experimental recordings

Unsupervised learning

Supervised algorithms

- Generate a function that maps input data into output data
- Classification and regression problems
- Labels for training

Often there is no label information available

- Ongoing neural activity
- Mixtures of different speakers in an audio recording
- Complex artefacts in experimental recordings

Unsupervised algorithms

- Find structure in data sets
- Allow partitioning of data in *meaningful* parts
- No labels for training

Dimensionality Reduction

Furthermore, in many applications, we have

- high-dimensional data
- known that computation efficiency will be increased by lower dimensional subspace while retaining most of the information

$\rightarrow$ Fewer parameters that have strong correlations between variables

Dimensionality Reduction

Furthermore, in many applications, we have

- high-dimensional data
- known that computation efficiency will be increased by lower dimensional subspace while retaining most of the information

$\rightarrow$ Fewer parameters that have strong correlations between variables

Examples:

- you want to classify high resolution images
- you want to make a predictive model based on hundreds of customer attributes
- you want to analyze high dimensional neural data
- you want to detect trends in new data

Why Dimensionality Reduction

- **Intuitive visualization**:
  Insights into high-dimensional structures in the data
- **Better generalization**:
  Fewer dimensions $\longrightarrow$ less chances of overfitting
- **Speeding up** learning algorithms:
  Most algorithms scale badly with increasing data dimensionality
- **Data compression**:
  Less storage requirements

The mathematical model for dimensionality reduction

We have

- high-dimensional data $X = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$

The mathematical model for dimensionality reduction

We have

- high-dimensional data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$

Goal: Reduce the dimensions of a D-dimensional dataset by projecting it onto a (k)-dimensional subspace (where $k < D$) while minimizing the loss.
The smaller dimensional subspace, $Y \in \mathbb{R}^{k \times N}$ can be expressed by mixing an projection matrix, ‘$W \in \mathbb{R}^{k \times D}$’ and input, ‘$X$’:

$$Y \approx WX$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\mathrm{Var}(Y) = \mathrm{Var}(\mathbf{w}^T X)$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\begin{aligned} \mathrm{Var}(Y) &= \mathrm{Var}(\mathbf{w}^T X) \\ \mathrm{Var}(\mathbf{w}^T X) &= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{s}_n - \mathbf{w}^\top \bar{\mathbf{s}}_n)^2 \end{aligned}$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$
\begin{aligned}
\mathrm{Var}(Y) &= \mathrm{Var}(\mathbf{w}^T X) \\
\mathrm{Var}(\mathbf{w}^T X) &= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{s}_n - \mathbf{w}^\top \bar{\mathbf{s}}_n)^2 \\
&= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top (\mathbf{s}_n - \bar{\mathbf{s}}_n))^2
\end{aligned}
$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$
\begin{aligned}
\mathrm{Var}(Y) &= \mathrm{Var}(\mathbf{w}^T X) \\
\mathrm{Var}(\mathbf{w}^T X) &= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{s}_n - \mathbf{w}^\top \bar{\mathbf{s}}_n)^2 \\
&= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top (\mathbf{s}_n - \bar{\mathbf{s}}_n))^2 \\
&= \frac{1}{N-1} \sum_{n=1}^{N} \mathbf{w}^\top (\mathbf{s}_n - \bar{\mathbf{s}}_n) \cdot (\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top \mathbf{w}
\end{aligned}
$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\mathrm{Var}(Y) = \mathrm{Var}(\mathbf{w}^T X)$$

$$\begin{aligned}
\mathrm{Var}(\mathbf{w}^T X) &= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{s}_n - \mathbf{w}^\top \bar{\mathbf{s}}_n)^2 \\
&= \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{w}^\top (\mathbf{s}_n - \bar{\mathbf{s}}_n))^2 \\
&= \frac{1}{N-1} \sum_{n=1}^{N} \mathbf{w}^\top (\mathbf{s}_n - \bar{\mathbf{s}}_n) \cdot (\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top \mathbf{w} \\
&= \mathbf{w}^\top \underbrace{\left(\frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{s}_n - \bar{\mathbf{s}}_n) \cdot (\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top \right)}_{\text{Covariance matrix S}} \mathbf{w}
\end{aligned}$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\mathrm{Var}(Y) = \mathbf{w}^\top \underbrace{\left(\frac{1}{N} \sum_{n=1}^{N} (\mathbf{s}_n - \bar{\mathbf{s}}_n) \cdot (\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top \right)}_{\text{Covariance matrix S}} \mathbf{w}$$

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N] \in \mathbb{R}^{D \times N}$

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\begin{aligned} \mathrm{Var}(Y) &= \mathbf{w}^\top \underbrace{\left(\frac{1}{N} \sum_{n=1}^{N} (\mathbf{s}_n - \bar{\mathbf{s}}_n) \cdot (\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top \right)}_{\text{Covariance matrix S}} \mathbf{w} \\ &= \mathbf{w}^\top X X^\top \mathbf{w} \end{aligned}$$

where we assume centered data

Principal Component Analysis

We obtained some data $X = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{D\times N}$

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\begin{aligned} \mathrm{Var}(Y) &= \mathbf{w}^\top \underbrace{\left(\frac{1}{N}\sum_{n=1}^{N}(\mathbf{s}_n - \bar{\mathbf{s}}_n)\cdot(\mathbf{s}_n - \bar{\mathbf{s}}_n)^\top\right)}_{\text{Covariance matrix S}} \mathbf{w} \\ &= \mathbf{w}^\top X X^\top \mathbf{w} \end{aligned}$$

where we assume centered data

And: we need to constrain $\mathbf{w}$

Principal Component Analysis

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\arg\max_{w} \mathrm{Var}(Y)$$

This objective function should be independent of the scaling of $\mathbf{w}$. Therefore, we can define constrained optimization w.r.t. $\mathbf{w}$:

$$\arg\max_{w} \ \mathbf{w}^\top S \mathbf{w}$$

$$\text{subject to} \quad \mathbf{w}^\top \mathbf{w} = 1$$

Principal Component Analysis

PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data Y is maximal

$$\arg\max_{w} \mathrm{Var}(Y)$$

This objective function should be independent of the scaling of $\mathbf{w}$. Therefore, we can define constrained optimization w.r.t. $\mathbf{w}$:

$$\arg\max_{w} \; \mathbf{w}^\top S\mathbf{w}$$
$$\text{subject to} \quad \mathbf{w}^\top \mathbf{w} = 1$$

Note the similarity to the objective of Linear Discriminant Analysis!

$\rightarrow$ Different covariance matrices, different problem, but: same maths solve it

Principal Component Analysis

$$L(\mathbf{w}) = \mathbf{w}^\top S\mathbf{w} + \lambda(1 - \mathbf{w}^\top \mathbf{w}) \tag{1}$$

Set the derivative w.r.t **w** to zero:

$$\begin{aligned}
\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial(\mathbf{w}^\top S\mathbf{w} + \lambda(1 - \mathbf{w}^\top \mathbf{w}))}{\partial \mathbf{w}} \\
2S\mathbf{w} - 2\lambda\mathbf{w} &= 0 \qquad (2) \\
S\mathbf{w} &= \lambda\mathbf{w} \qquad (3)
\end{aligned}$$

This is the standard eigenvalue problem. Once calculate covariance matrix S, and we can calculate its eigenvector that can maximize **w**

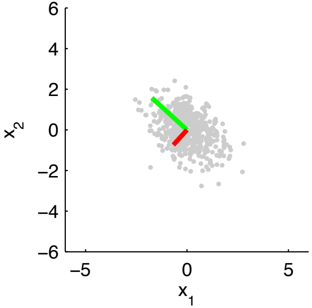
Principal Component Analysis

Algorithm 1 Principal Component Analysis

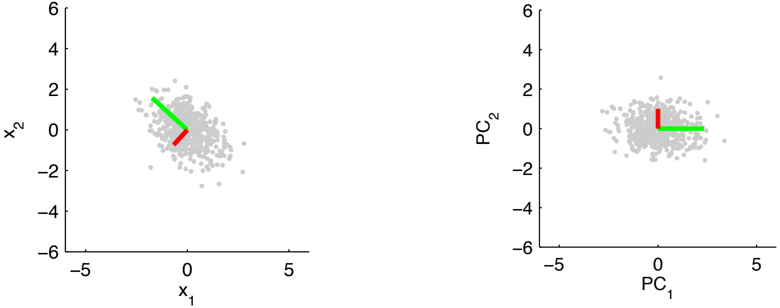
Require: data $\mathbf{s}_1, \ldots, \mathbf{s}_N \in \mathbb{R}^d$, number of principal components k

```
# Center Data
X = X − 1/N ∑_i s_i
# Compute Covariance Matrix
S = 1/(N − 1) X X^⊤
# Compute eigenvectors corresponding to the k largest eigenvalues
W = eig(S)
# Project data onto W
Y = W^⊤ X
return W, Y
```

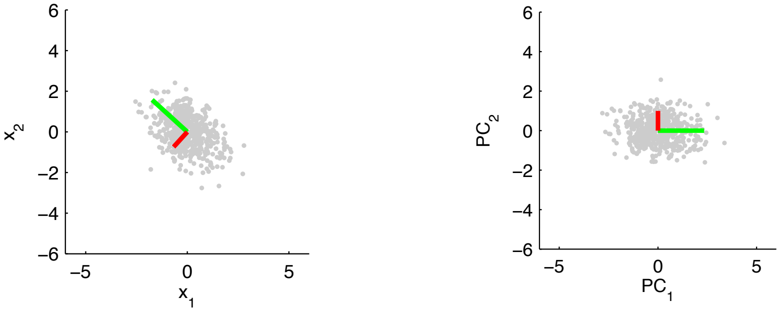
Principal Component Analysis



Principal Component Analysis



Principal Component Analysis



PCA finds the component axes that maximize the variance of our given data

PCA aligns maximum variance directions with standard basis

$\rightarrow$ Variance along each dimension is **uncorrelated**

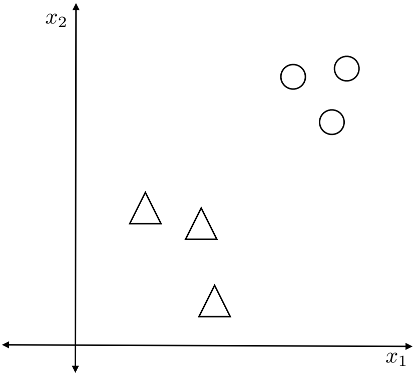
$\rightarrow$ Now we can remove each dimension separately

Agenda

Principal Component Analysis

Linear Classification & LDA

Prototypes: Psychological Models of Abstract Ideas



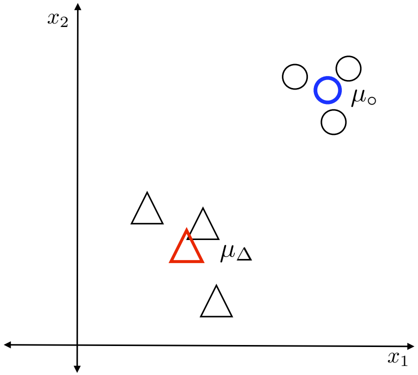
Psychologists postulated that we learn **prototypes**.

Toy data example:

Neuron receives two dimensional input $x \in \mathbb{R}^2$

Two *classes* of data, Δ and $\circ$

Prototypes: Psychological Models of Abstract Ideas



Prototypes μ_Δ and μ_o can be the class means

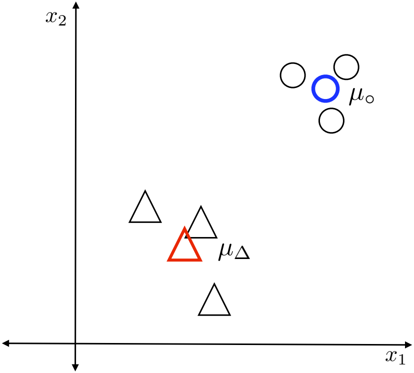
$$\mu_\Delta = \frac{1}{N_\Delta} \sum_{n=1}^{N_\Delta} \mathbf{x}_{\Delta,n}$$

$$\mu_o = \frac{1}{N_o} \sum_{n=1}^{N_o} \mathbf{x}_{o,n}$$

Distance from μ_Δ to new data $\mathbf{x}$

$$\|\mu_\Delta - \mathbf{x}\| = \sqrt{\sum_{j=1}^{2} (\mu_{\Delta j} - x_j)^2}$$

Prototypes: Psychological Models of Abstract Ideas



For new data *x* check:

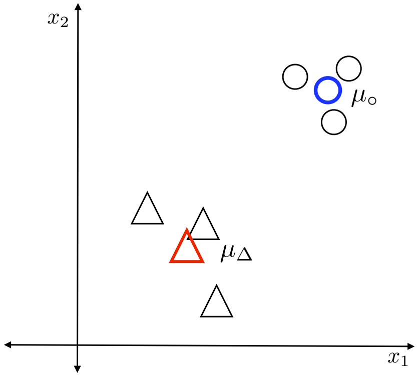
Is x more similar to μ_o?

$$\|\mu_\Delta - \mathbf{x}\| > \|\mu_o - \mathbf{x}\|$$

yes? $\rightarrow$ **x** belongs to *o*

no? $\rightarrow$ **x** belongs to Δ

Prototypes: Psychological Models of Abstract Ideas



For new data *x* check:

Is x more similar to μ_o?

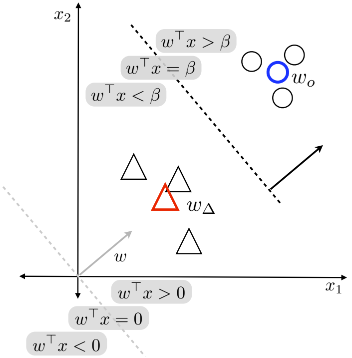
$$\|\mu_\Delta - \mathbf{x}\| > \|\mu_o - \mathbf{x}\|$$

yes? $\rightarrow$ **x** belongs to *o*

no? $\rightarrow$ **x** belongs to Δ

How does the classification boundary look like?

Linear Classification



Comparison of distance to class means is equivalent to linear classification

$$\|\mathbf{x} - \mu_\Delta\| > \|\mathbf{x} - \mu_o\|$$
$$\Leftrightarrow 0 < \mathbf{w}^\top \mathbf{x} - \beta$$

where

$$\mathbf{w} = \mu_o - \mu_\Delta$$

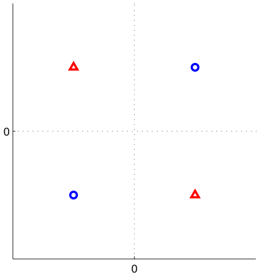
and

$$\beta = 1/2 \cdot \mathbf{w}^\top (\mu_o + \mu_\Delta)$$

β is computed by the centroid of intra-class variances. This simple linear classification rule is often called **Nearest Centroid Classifier**.

Problems with Nearest Centroid Classification

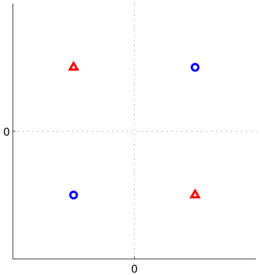
Non-linear Data



Solutions
Non-linear features,
Non-linear classification methods

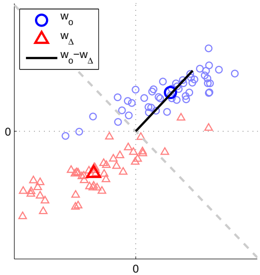
Problems with Nearest Centroid Classification

Non-linear Data



Solutions
Non-linear features,
Non-linear classification methods

Correlated Data



Solution
(Fisher's) Linear Discriminant Analysis

Ronald A. Fisher

R.A. Fisher (1890 - 1962)

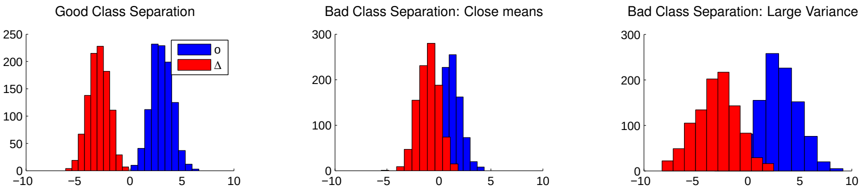
Founder of modern statistics

Interested in Biology

Formulated *Linear Discriminant Analysis* (LDA) in 1936

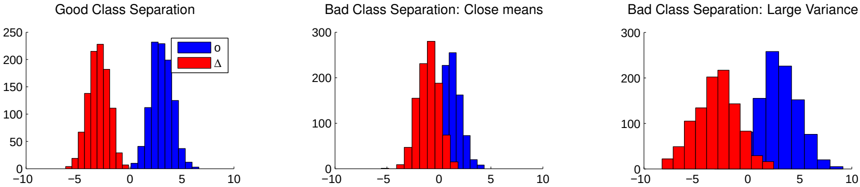
The Fisher Criterion - measure for class separability

Consider one dimensional data and two classes



The Fisher Criterion - measure for class separability

Consider one dimensional data and two classes



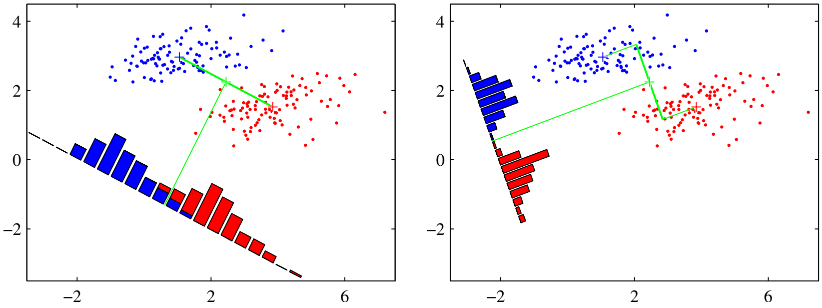
The fisher criterion:

$$\frac{\text{between class mean}}{\text{within class variance}} = \frac{(\mu_o - \mu_\Delta)^2}{\sigma_o^2 + \sigma_\Delta^2}$$

where $\mathbf{x}_{1o}, \ldots, \mathbf{x}_{N_o o} \in \mathbb{R}^D$ and
$\mu_o = \frac{1}{N_o} \sum_{i=1}^{N_o} \mathbf{x}_{io}$ and $\sigma_o^2 = \frac{1}{N_o} \sum_{i=1}^{N_o} (\mathbf{x}_{io} - \mu_o)^2$.

Linear Discriminant Analysis

View classification in terms of dimensionality reduction



Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximize mean class difference, and

Minimize variance in each class

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximize (projected) difference in class mean

$$(\mathbf{w}^\top \mu_o - \mathbf{w}^\top \mu_\Delta)^2 = \mathbf{w}^\top \underbrace{(\mu_o - \mu_\Delta)(\mu_o - \mu_\Delta)^\top}_{S_B\text{ — "between class scatter"}} \mathbf{w} \tag{4}$$

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximize (projected) difference in class mean

$$(\mathbf{w}^\top \mu_o - \mathbf{w}^\top \mu_\Delta)^2 = \mathbf{w}^\top \underbrace{(\mu_o - \mu_\Delta)(\mu_o - \mu_\Delta)^\top}_{S_B - \text{"between class scatter"}} \mathbf{w} \tag{4}$$

Minimize (projected) variance in each class

$$\frac{1}{N_o} \sum_{i=1}^{N_o} \left(\mathbf{w}^\top (\mathbf{x}_{oi} - \mu_o)\right)^2 + \frac{1}{N_\Delta} \sum_{j=1}^{N_\Delta} \left(\mathbf{w}^\top (\mathbf{x}_{\Delta j} - \mu_\Delta)\right)^2$$

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w} \in \mathbb{R}^D$ that

Maximize (projected) difference in class mean

$$(\mathbf{w}^\top \mu_o - \mathbf{w}^\top \mu_\Delta)^2 = \mathbf{w}^\top \underbrace{(\mu_o - \mu_\Delta)(\mu_o - \mu_\Delta)^\top}_{S_B \text{ — "between class scatter"}} \mathbf{w} \tag{4}$$

Minimize (projected) variance in each class

$$\frac{1}{N_o}\sum_{i=1}^{N_o}\left(\mathbf{w}^\top(\mathbf{x}_{oi} - \mu_o)\right)^2 + \frac{1}{N_\Delta}\sum_{j=1}^{N_\Delta}\left(\mathbf{w}^\top(\mathbf{x}_{\Delta j} - \mu_\Delta)\right)^2$$

$$= \mathbf{w}^\top \underbrace{\left(\frac{1}{N_o}\sum_{i=1}^{N_o}(\mathbf{x}_{oi} - \mu_o)(\mathbf{x}_{oi} - \mu_o)^\top + \frac{1}{N_\Delta}\sum_{j=1}^{N_\Delta}(\mathbf{x}_{\Delta j} - \mu_\Delta)(\mathbf{x}_{\Delta j} - \mu_\Delta)^\top\right)}_{S_W \text{ — "within class scatter"}} \mathbf{w}$$

Technische Universität Berlin

Linear Discriminant Analysis

Goal: Find a (normal vector of a linear decision boundary) $\mathbf{w}$ that

Maximizes mean class difference, $\mathbf{w}^\top S_B \mathbf{w}$ and

Minimizes variance in each class, $\mathbf{w}^\top S_W \mathbf{w}$

$\rightarrow$ maximize the *Fisher criterion*

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}} \tag{5}$$

Linear Discriminant Analysis

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

To optimize the Fisher criterion, we set its derivative w.r.t **w** to 0

$$\frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2} = 0$$

Linear Discriminant Analysis

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

To optimize the Fisher criterion, we set its derivative w.r.t **w** to 0

$$\begin{aligned}
\frac{(\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} - (\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w}}{(\mathbf{w}^\top S_W \mathbf{w})^2} &= 0 \\
(\mathbf{w}^\top S_B \mathbf{w}) S_W \mathbf{w} &= (\mathbf{w}^\top S_W \mathbf{w}) S_B \mathbf{w} \\
S_W \mathbf{w} &= S_B \mathbf{w} \underbrace{\frac{\mathbf{w}^\top S_W \mathbf{w}}{\mathbf{w}^\top S_B \mathbf{w}}}_{\textit{scalar}}
\end{aligned}$$

Linear Discriminant Analysis

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

$$\rightarrow S_W \mathbf{w} = \lambda S_B \mathbf{w}$$

Linear Discriminant Analysis

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

$$\rightarrow S_W \mathbf{w} = \lambda S_B \mathbf{w}$$

Note that

$$S_B \mathbf{w} = (\mu_o - \mu_\Delta) \underbrace{(\mu_o - \mu_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

Linear Discriminant Analysis

$$\arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}$$

$$\rightarrow S_W \mathbf{w} = \lambda S_B \mathbf{w}$$

Note that

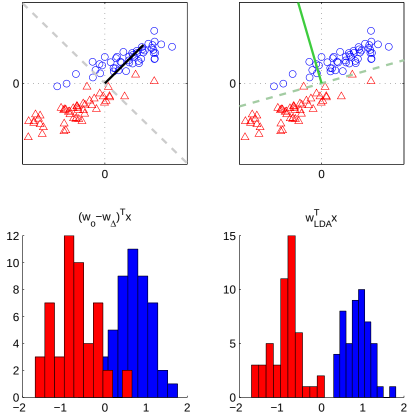
$$S_B \mathbf{w} = (\mu_o - \mu_\Delta) \underbrace{(\mu_o - \mu_\Delta)^\top \mathbf{w}}_{\text{scalar}}$$

thus left multiplying with S_W^{-1} yields

$$\mathbf{w} \propto S_W^{-1}(\mu_o - \mu_\Delta).$$

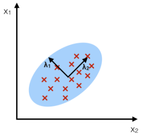
($\propto$ denotes proportional)

Linear Discriminant Analysis vs Nearest Centroid Classifier

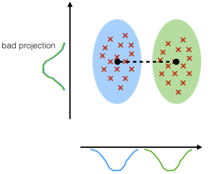


PCA vs. LDA

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation

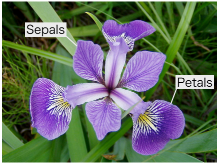


- PCA and LDA are linear transformation techniques that are commonly used for dimensionality reduction
- PCA can be described as an “unsupervised” algorithm, since it “ignores” class labels and its goal is to find the directions (the so-called principal components) that maximize the variance in a dataset
- LDA is “supervised” and computes the directions (“linear discriminants”) that will represent the axes that maximize the separation between multiple classes

Ipython Exercise: The *Iris* Flower Dataset

Iris Setosa

Iris Versicolor



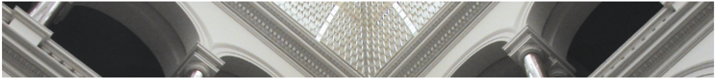
Iris Virginica

http://en.wikipedia.org/wiki/Iris_flower_data_set

50 flowers of each species were collected
"all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus"

Petal and Sepal length and width were measured

Very popular benchmark data set



Technische Universität Berlin

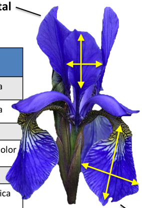
Ipython Exercise: The *Iris* Flower Dataset

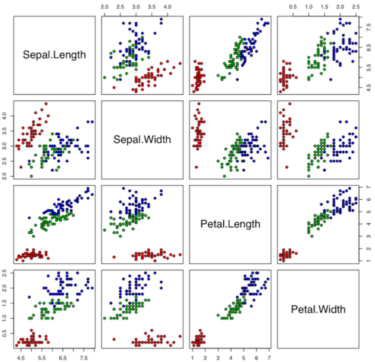
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)





http://en.wikipedia.org/wiki/Iris_flower_data_set

Thank you!