Fundamentals of Image Processing

▶ Lecture 10: Introduction to pattern recognition ◀

Data analysis, image classification

Master of Computer Science

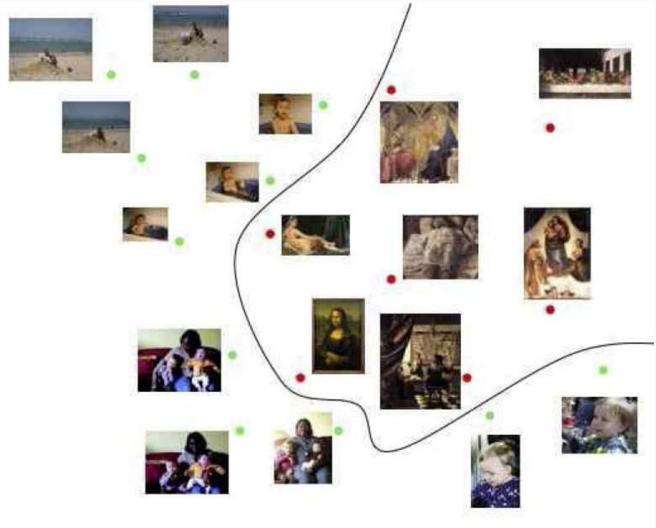
Sorbonne University

September 2022

Outline

Introduction

Introduction: classification



Visualization

Visualization

- Meta-data: sepal length and width, petal length and width for three species of Iris.

(a) Iris Setosa

(b) Iris Versicolor

(c) Iris Virginica

Iris database (Fisher, 1936)

$$X = \begin{pmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ 5.0 & 3.6 & 1.4 & 0.2 \\ \vdots & & & \\ 7.0 & 3.2 & 4.7 & 1.4 \\ 6.4 & 3.2 & 4.5 & 1.5 \\ 6.9 & 3.1 & 4.9 & 1.5 \\ 5.5 & 2.3 & 4.0 & 1.3 \\ 6.5 & 2.8 & 4.6 & 1.5 \\ \vdots & & & \\ 6.3 & 3.3 & 6.0 & 2.5 \\ 5.8 & 2.7 & 5.1 & 1.9 \\ 7.1 & 3.0 & 5.9 & 2.1 \\ 6.3 & 2.9 & 5.6 & 1.8 \\ 6.5 & 3.0 & 5.8 & 2.2 \\ \vdots & & & \end{pmatrix} \quad Y = \begin{pmatrix} \textit{setosa} \\ \textit{setosa} \\ \textit{setosa} \\ \textit{setosa} \\ \textit{setosa} \\ \vdots \\ \textit{versicolor} \\ \textit{versicolor} \\ \textit{versicolor} \\ \textit{versicolor} \\ \textit{versicolor} \\ \vdots \\ \textit{virginica} \\ \textit{virginica} \\ \textit{virginica} \\ \textit{virginica} \\ \textit{virginica} \\ \vdots \end{pmatrix}$$

Data visualization

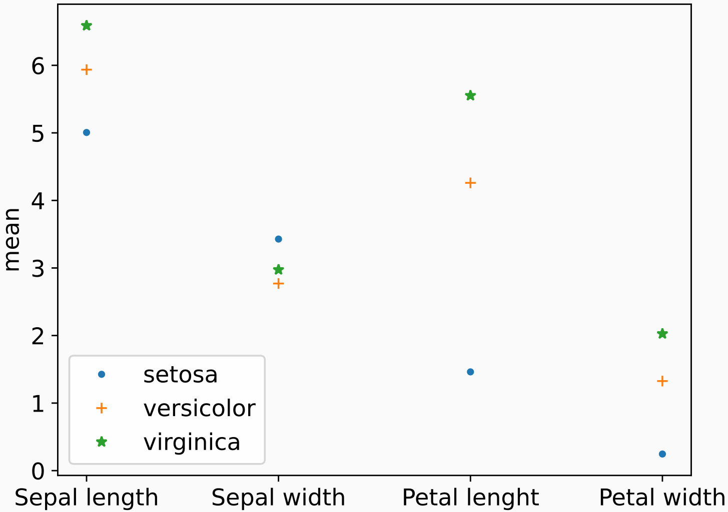
- Basic statistics: mean, standard deviation, median...
- Mean:

	'Sepal length'	'Sepal width'	'Petal length'	'Petal width'
'setosa'	5.006	3.428	1.462	0.246
'versicolor'	5.936	2.77	4.26	1.326
'virginica'	6.588	2.974	5.552	2.026

- Standard deviation:

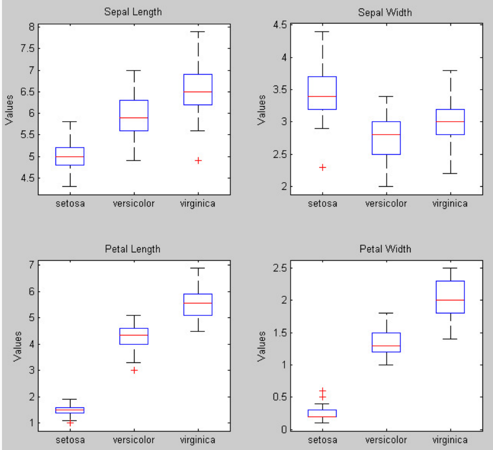
	'Sepal length'	'Sepal width'	'Petal length'	'Petal width'
'setosa'	0.349	0.375	0.172	0.104
'versicolor'	0.511	0.311	0.465	0.196
'virginica'	0.630	0.320	0.546	0.272

Data visualization



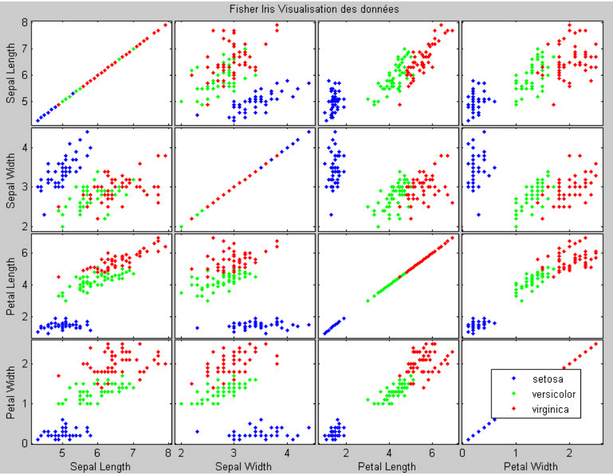
Data visualization

- Basic statistics: mean, standard deviation, median...
- Boxplot: min, max, first quartile, median, third quartile, outliers.



Data visualization

- Basic statistics: mean, standard deviation, median...
- Boxplot: min, max, first and third quartile, median, outliers.
- Crossed analysis: correlation, analysis of variance...



Data visualization

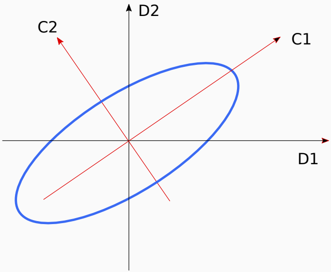
- Basic statistics: mean, standard deviation, median...
- Boxplot: min, max, first and third quartile, median, outliers.
- Crossed analysis: correlation, analysis of variance...
- Principal component analysis: fundamental process to reduce the size of data

Principal component analysis

Principal component analysis (PCA)

- Analysis of the structure of the variance-covariance matrix i.e. variability, dispersion of the data.
- Initially: n variables $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ (vectors of dimension $d \leq n$) to account for all data variability.
- PCA objective: describe most of this variability using $q < d$ components.
- Which allows:
  - a data reduction with a new set of descriptors
  - data visualization in 2 or 3 dimensions (if $q = 2$ or 3)
  - data interpretation: inter-variable links
- Preliminary step often used before further analysis!

PCA: geometric interpretation



- Components: $C_1, \cdots, C_q$
- $C_k =$ linear combination of vectors $D_1, \cdots, D_d$
- Coefficients $a_{ik} =< \mathbf{x}_i, C_k >$ projection of $\mathbf{x}_i$ on vector C_k.
- $(C_1, \cdots, C_q)$ determined such that:
  - $\langle C_k, C_{k'} \rangle = 0$ for $k \neq k'$
  - data projected on each C_k are of maximal variance
  - and sorted by decreasing importance of projected variance

Projection of variance

- $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \cdots, n$
- Variance: $\sigma^2 = \dfrac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{g})^T (\mathbf{x}_i - \mathbf{g})$

  with $\mathbf{g} = \dfrac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$
- Projection on vector $\mathbf{v}$: operator π such that $\pi = \mathbf{v}\mathbf{v}^T$ with $\mathbf{v}^T\mathbf{v} = 1$
- Variance of projected data on $\mathbf{v}$:
  $\sigma_{\mathbf{v}}^2 = \dfrac{1}{n-1} \sum_{i=1}^{n} (\pi(\mathbf{x}_i - \mathbf{g}))^T (\pi(\mathbf{x}_i - \mathbf{g}))$

Projection of variance

$$\begin{aligned}
\sigma_{\mathbf{v}}^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(\pi(\mathbf{x}_i-\mathbf{g}))^T(\pi(\mathbf{x}_i-\mathbf{g})) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{v}\mathbf{v}^T(\mathbf{x}_i-\mathbf{g}))^T(\mathbf{v}\mathbf{v}^T(\mathbf{x}_i-\mathbf{g})) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{g})^T(\mathbf{v}\mathbf{v}^T\mathbf{v}\mathbf{v}^T)(\mathbf{x}_i-\mathbf{g}) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{g})^T\mathbf{v}\mathbf{v}^T(\mathbf{x}_i-\mathbf{g}) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\mathbf{v}^T(\mathbf{x}_i-\mathbf{g})(\mathbf{x}_i-\mathbf{g})^T\mathbf{v} \\
&= \frac{1}{n-1}\mathbf{v}^T\left[\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{g})(\mathbf{x}_i-\mathbf{g})^T\right]\mathbf{v} = \mathbf{v}^T\Sigma\mathbf{v}
\end{aligned}$$

Maximization of projected variance

- $\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \Sigma \mathbf{v}$
- Σ matrix of covariance, symmetric and positive definite
- Maximum of σ_v w.r.t. $\mathbf{v}$ such that $\mathbf{v}^T \mathbf{v} = 1$?
- Use of multiplier of Lagrange method (optimization with constraints):

$$\mathcal{L}(\mathbf{v}, \lambda) = \mathbf{v}^T \Sigma \mathbf{v} + \lambda(1 - \mathbf{v}^T \mathbf{v})$$

- Necessary condition:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = 0$$

- leads to :

$$\Sigma \mathbf{v} = \lambda \mathbf{v}$$

Maximization of projected variance

- Search of $\mathbf{v}$ such as $\Sigma\mathbf{v} = \lambda\mathbf{v}$ with Σ matrix of covariance of data $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ (see tutorial work)
- By definition, λ and $\mathbf{v}$ are respectively eigenvalues and eigenvectors of Σ.
- Σ definite positive $\Rightarrow \lambda \geq 0$
- Variance of projected data:

$$\begin{aligned} \sigma_{\mathbf{v}}^2 &= \mathbf{v}^T\Sigma\mathbf{v} \\ &= \mathbf{v}^T\lambda\mathbf{v} \\ &= \lambda \end{aligned}$$

- PCA: projection of data on eigenvectors associated with the covariance matrix.
- Principal components: eigenvectors having the highest eigenvalues.

PCA: summary

1. Center data on mean : $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{g}$
2. Determine the variance-covariance matrix :
   $\Sigma = (n-1)^{-1}\mathbf{X}\mathbf{X}^T$, $\mathbf{X} = (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n)$
3. Diagonalization of Σ and sort according to decreasing eigenvalues
4. Selection of the first q eigenvectors C_k $(q \leq d)$ of Σ
5. Determination of vectors $\mathbf{a}_i$, $a_{ik} =< \mathbf{x}_i, C_k >$ of length q replacing the vectors $\mathbf{x}_i$

PCA: summary

- The PCA replaces the d original variables $\mathbf{x}_i$ with new q components ($q \leq d$) $\mathbf{a}_k$
  - which are pair-wise uncorrelated i.e. $\mathrm{cov}(C_k, C_{k'}) = 0 \quad \forall k \neq k'$
  - which have maximum variances, with $V(C_1) \geq V(C_2) \geq \cdots \geq V(C_q)$
- The maximum number of principal components $q \leq d$ becomes $q < d$ as soon as one of the original variables is a linear combination of the others!
  - highlight linear relationships in the data
  - the data actually belong to a subspace of reduced dimensions ($q < d$) i.e. maximum number of principal components = intrinsic dimension of the data

Choice for q

- $q \ll d$ reducing data size, obtaining uncorrelated data.
- Objective: keep as much information as possible from the initial data, which corresponds to the explained variance.
- Explained variance: $\sum_{k=1}^{q} V(C_k)$
- Ratio of information explained (inertia): $I = \frac{\sum_{k=1}^{q} V(C_k)}{\sum_{k=1}^{d} V(C_k)}$. For instance, choose q to keep 95% of total variance.
- Geometrically: this is equivalent to project the data into a sub-space of dimension q, centered on $\mathbf{g}$, keeping the q first main axes.

Example of PCA application: face recognition

$\xrightarrow[\text{Detection}]{}$

$\xrightarrow[\text{Recognition}]{}$ "Sally"

Applications of face recognition

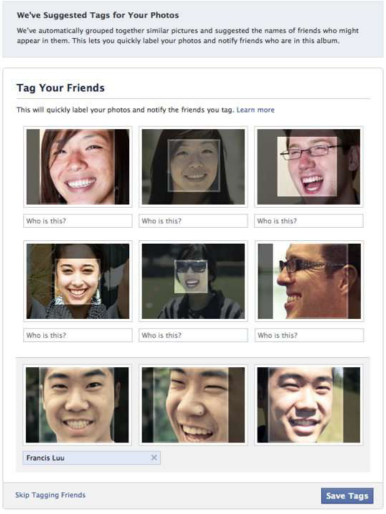
- Video surveillance

Applications of face recognition

- Album organization: iPhoto 2009

Applications of face recognition

- Facebook friend-tagging with auto-suggestion



Issues with face recognition

- Things iPhoto thinks are faces

Typical face recognition scenarios

- Verification: a person is claiming a particular identity; verify whether this is true
  - e.g. security.
- Closed-world identification: assign a face to one person among a known set of persons.
- General identification: assign a face to a known person or to "unknown".

Simple idea for face recognition

1. Treat face image as a vector of intensities

$$\longrightarrow \mathbf{x}$$

2. Recognize face by nearest neighbors in a database

$$\longrightarrow \mathbf{y}_1, \cdots, \mathbf{y}_n$$

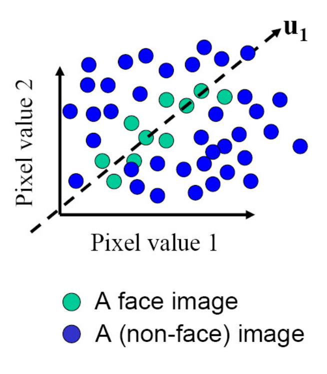
$$k = \underset{k}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{x}\|$$

The space of all face images

- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100×100 image $=$ 10,000 dimensions
  - Slow computation and high storage/memory cost
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

The space of all face images

- Eigenface idea: construct a low-dimensional linear subspace that best explains the variation in the set of face images



- Use PCA analysis to determine a relevant subspace

Eigenfaces (PCA on face images)

1. Compute the principal components ("eigenfaces") of the covariance matrix $\mathbf{X}\mathbf{X}^T$ with $\mathbf{X} = (\mathbf{x}_1 - \mathbf{g}, \cdots, \mathbf{x}_n - \mathbf{g})$ where $\mathbf{x}_i$ is an image of a face flatten into a vector
2. Keep q eigenvectors with largest eigenvalues
3. Represent all face images in the dataset as linear combinations of eigenfaces
    - Perform nearest neighbor on these coefficients

Eigenfaces: Implementation issue

- Covariance matrix is huge (d^2 for d pixels)
- But typically the number of examples $n \ll d$
- Simple trick:
  - $\mathbf{X}\mathbf{X}^T$ is $d \times d$ matrix of normalized training data
  - Solve for eigenvectors $\mathbf{u}$ of $\mathbf{X}^T\mathbf{X}$ ($n \times n$ matrix) instead of $\mathbf{X}\mathbf{X}^T$
  - Then $\mathbf{v} = \mathbf{X}\mathbf{u}$ is eigenvector of covariance $\mathbf{X}\mathbf{X}^T$
  - Need to normalize each vector of $\mathbf{X}\mathbf{u}$ into unit length

Eigenfaces example

- training images
- $\mathbf{x}_1, \cdots, \mathbf{x}_n$

Eigenfaces example

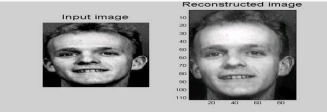
Mean: $\mathbf{g}$

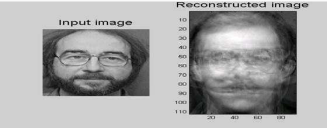
Top eigenvectors (eigenvectors of Σ):

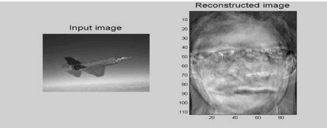
$C_1, \ldots, C_q$

Eigenfaces example

- Reduction of the number of vectors in the base
- Allows for a fast search of similar faces in the database
- Dedicated to face detection if trained on faces database !

known face


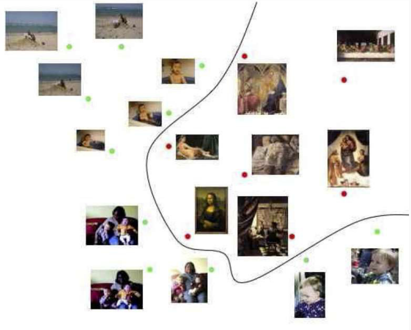
unknown face


not a face


- To experiment during practical works

PCA: conclusion

- A method for data analysis that allows for:
  - a reduction of the data to q descriptors
  - an easy data visualization if $q = 2$ or 3
  - data interpretation (linear inter-variable links)
- Intermediate step often used before further analysis
- A method that does not allow to take into account classes:



$\Rightarrow$ Classification: LDA, SVM...

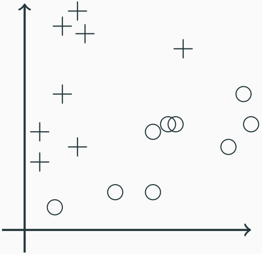
Linear Discriminant Analysis

LDA: Linear Discriminant Analysis

- Objective: to highlight differences between classes, i.e. between observations belonging to different classes.
- Description of the links between the “class” variable and the quantitative variables: do the q classes differ on all the numerical variables?
- Method close to PCA: linear transformation of the variables (change of basis) but taking into account the classes of individuals.

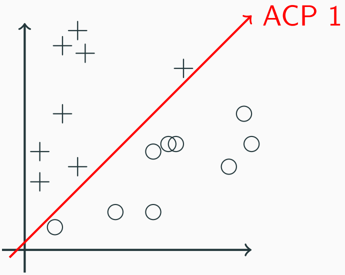
LDA: principle

- Determine *discriminating* factors as linear combinations of original descriptive variables, such that:
    1. values of a same class are the closest possible,
    2. values of different classes are the furthest possible.



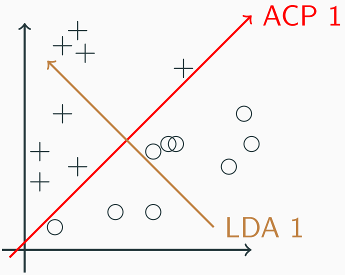
LDA: principle

- Determine *discriminating* factors as linear combinations of original descriptive variables, such that:
    1. values of a same class are the closest possible,
    2. values of different classes are the furthest possible.



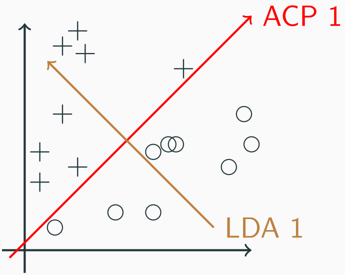
LDA: principle

- Determine *discriminating* factors as linear combinations of original descriptive variables, such that:
  1. values of a same class are the closest possible,
  2. values of different classes are the furthest possible.



LDA: principle

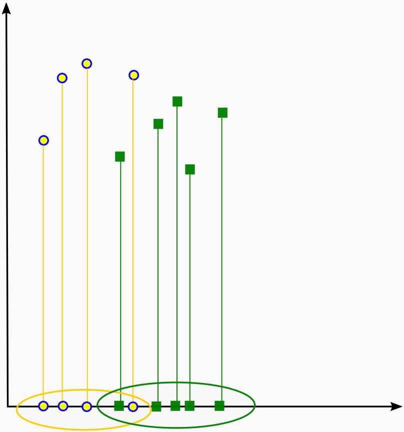
- Determine *discriminating* factors as linear combinations of original descriptive variables, such that:
    1. values of a same class are the closest possible,
    2. values of different classes are the furthest possible.



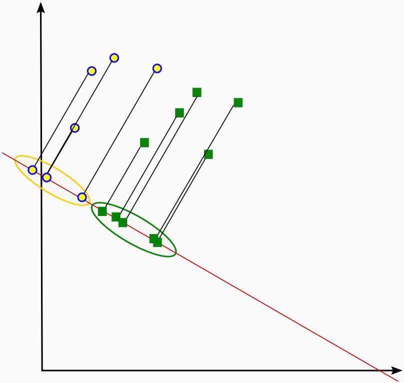
- Data projected on the first LDA axis have:
    - a minimal within class (intra-class) variance,
    - a maximum between class (inter-class) variance.

LDA: principle

- Using two classes as example:



(a) Poor projection



(b) Good projection

LDA: some notations

- Data: $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$, $\mathbf{x}_i \in \mathbb{R}^d$
- Classes: $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_q) \in \Omega$
- Ω a finite set of size q
- C_k = set of data belonging to class $\mathbf{y}_k$:
  $\Rightarrow \mathbf{x}_i$ is of class $\mathbf{y}_k \Leftrightarrow i \in C_k$
- $n_k = |C_k|$, $\displaystyle\sum_{i=1}^{q} n_k = n$
- Center of gravity: $\mathbf{g} = \dfrac{1}{n}\displaystyle\sum_{1}^{n} \mathbf{x}_i$
- Center of gravity of class k: $\mathbf{g}_k = \dfrac{1}{n_k}\displaystyle\sum_{i \in C_k} \mathbf{x}_i$

Decomposition of total variance

$$\begin{aligned}
\sigma^2 &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbf{g})^T(\mathbf{x}_i-\mathbf{g}) \\
&= \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in C_k}(\mathbf{x}_i-\mathbf{g})^T(\mathbf{x}_i-\mathbf{g}) \\
&= \frac{1}{n}\sum_{k=1}^{q} SS(k)
\end{aligned}$$

$$\begin{aligned}
SS(k) &= \sum_{i\in C_k}(\mathbf{x}_i-\mathbf{g})^T(\mathbf{x}_i-\mathbf{g}) \\
&= \sum_{i\in C_k}(\mathbf{x}_i-\mathbf{g}_k+\mathbf{g}_k-\mathbf{g})^T(\mathbf{x}_i-\mathbf{g}_k+\mathbf{g}_k-\mathbf{g})
\end{aligned}$$

Decomposition of total variance

$$SS(k) = \sum_{i \in C_k} \left(\|\mathbf{x}_i - g_k\|^2 + \|\mathbf{g}_k - \mathbf{g}\|^2 + 2(\mathbf{x}_i - \mathbf{g}_k)^T(\mathbf{g}_k - \mathbf{g}) \right)$$

$$\sum_{i \in C_k} 2(\mathbf{x}_i - \mathbf{g}_k)^T(\mathbf{g}_k - \mathbf{g}) = 0$$

$$\begin{aligned} SS(k) &= \sum_{i \in C_k} \left(\|\mathbf{x}_i - \mathbf{g}_k\|^2 + \|\mathbf{g}_k - \mathbf{g}\|^2 \right) \\ &= \underbrace{\sum_{i \in C_k} \|\mathbf{x}_i - \mathbf{g}_k\|^2}_{\text{within}} + \underbrace{n_k \|\mathbf{g}_k - \mathbf{g}\|^2}_{\text{between}} \end{aligned}$$

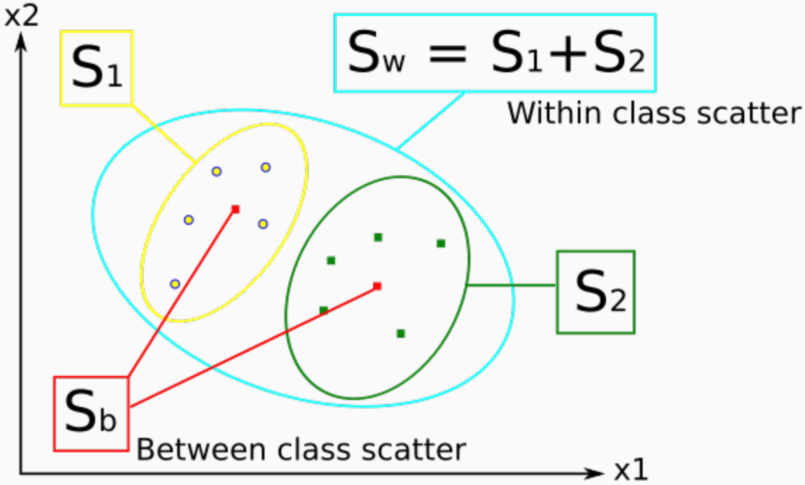
Decomposition of total variance

$$
\begin{aligned}
\sigma^2 &= \frac{1}{n}\left[\sum_{k=1}^{q}\sum_{i\in C_k}\|\mathbf{x}_i-\mathbf{g}_k\|^2+\sum_{i=1}^{q}n_k\|\mathbf{g}_k-\mathbf{g}\|^2\right]\\
&= \frac{1}{n}\left[\sum_{k=1}^{q}n_k\frac{1}{n_k}\sum_{i\in C_k}\|\mathbf{x}_i-\mathbf{g}_k\|^2+\sum_{k=1}^{q}n_k\|\mathbf{g}_k-\mathbf{g}\|^2\right]\\
&= \frac{1}{n}\sum_{k=1}^{q}n_k\left[\frac{1}{n_k}\sum_{i\in C_k}\|\mathbf{x}_i-\mathbf{g}_k\|^2+\|g_k-\mathbf{g}\|^2\right]\\
&= \frac{1}{n}\sum_{k=1}^{q}n_k\left[\sigma^2_{(\mathrm{w})}+\sigma^2_{(\mathrm{b})}\right]=\sigma^2_{(\mathrm{w})}+\sigma^2_{(\mathrm{b})}
\end{aligned}
$$

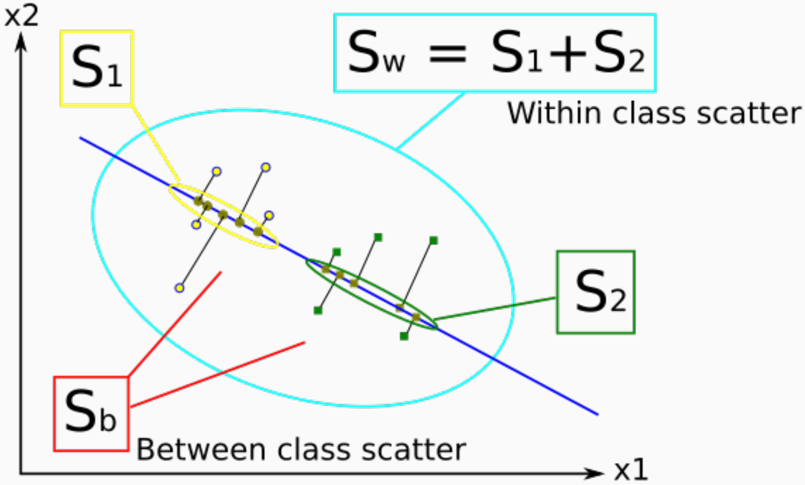
Variance projection

$$
\begin{aligned}
\sigma^2_{\mathbf{v}(\mathrm{w})} &= \frac{1}{n}\sum_{k=1}^{q} n_k \left(\frac{1}{n_k} \sum_{i \in C(k)} (\pi \mathbf{x}_i - \pi \mathbf{g}_k)^T (\pi \mathbf{x}_i - \pi \mathbf{g}_k) \right) \\
&= \mathbf{v}^T \left[\frac{1}{n}\sum_{k=1}^{q} n_k \left(\frac{1}{n_k} \sum_{i \in C(k)} (\mathbf{x}_i - \mathbf{g}_k)(\mathbf{x}_i - \mathbf{g}_k)^T \right) \right] \mathbf{v} \\
&= \mathbf{v}^T \left[\frac{1}{n}\sum_{k=1}^{q} n_k \mathbf{W}_k \right] \mathbf{v} = \mathbf{v}^T \mathbf{W} \mathbf{v} \\
\sigma^2_{\mathbf{v}(\mathrm{b})} &= \frac{1}{n}\sum_{k=1}^{q} n_k (\pi \mathbf{g}_k - \pi \mathbf{g})^T (\pi \mathbf{g}_k - \pi \mathbf{g}) \\
&= \mathbf{v}^T \left[\sum_{k=1}^{q} n_k \frac{(\mathbf{g}_k - \mathbf{g})(\mathbf{g}_k - \mathbf{g})^T}{n} \right] \mathbf{v} = \mathbf{v}^T \mathbf{B} \mathbf{v} \\
\sigma^2_{\mathbf{v}} &= \mathbf{v}^T \Sigma \mathbf{v}
\end{aligned}
$$

Illustration



Illustration



Optimization

$$\begin{aligned}\sigma_{\mathbf{v}}^2 &= \sigma_{\mathbf{v}(\mathsf{w})}^2 + \sigma_{\mathbf{v}(\mathsf{b})}^2 \\ 1 &= \frac{\sigma_{\mathbf{v}(\mathsf{w})}^2}{\sigma_{\mathbf{v}}^2} + \frac{\sigma_{\mathbf{v}(\mathsf{b})}^2}{\sigma_{\mathbf{v}}^2} \\ 0 &< \frac{\sigma_{\mathbf{v}(\mathsf{b})}^2}{\sigma_{\mathbf{v}}^2} < 1\end{aligned}$$

- LDA: find $\mathbf{v}$ such that the variance between classes of projected data, $\sigma_{\mathbf{v}(\mathsf{b})}$, is maximal :

$$\underset{\mathbf{v}}{\operatorname{argmax}}\left(\frac{\sigma_{\mathbf{v}(\mathsf{b})}^2}{\sigma_{\mathbf{v}}^2}\right) = \underset{\mathbf{v}}{\operatorname{argmax}}\left(\frac{\mathbf{v}^T\mathbf{B}\mathbf{v}}{\mathbf{v}^T\Sigma\mathbf{v}}\right)$$

Optimization

- Necessary condition:

$$\frac{\partial}{\partial \mathbf{v}} \left(\frac{\mathbf{v}^T \mathbf{B} \mathbf{v}}{\mathbf{v}^T \Sigma \mathbf{v}} \right)$$

- Then:

$$\begin{aligned}
2(\mathbf{v}^T \Sigma \mathbf{v}) \mathbf{B} \mathbf{v} - 2(\mathbf{v}^T \mathbf{B} \mathbf{v}) \Sigma \mathbf{v} &= 0 \\
\mathbf{B} \mathbf{v} &= \underbrace{\left(\frac{\mathbf{v}^T \mathbf{B} \mathbf{v}}{\mathbf{v}^T \Sigma \mathbf{v}} \right)}_{\lambda} \Sigma \mathbf{v} \\
\Sigma^{-1} \mathbf{B} \mathbf{v} &= \lambda \mathbf{v}
\end{aligned}$$

- LDA: projection of data on the eigenvector of $\Sigma^{-1}\mathbf{B}$ having the highest eigenvalue.

LDA: summary

1. Center data on mean: $\tilde{\mathbf{x}} = (\mathbf{x}_i - \mathbf{g})$
2. Determine the variance-covariance matrix:
   $\Sigma = (n-1)^{-1}\mathbf{X}\mathbf{X}^T$, $\mathbf{X} = (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n)$
3. Determine the matrix of between classes variance B:
   $$\mathbf{B} = \sum_{k=1}^{q} \frac{n_k}{n}(\mathbf{g}_k - \mathbf{g})(\mathbf{g}_k - \mathbf{g})^T$$
4. Diagonalization of $\Sigma^{-1}\mathbf{B}$ and sort according to decreasing eigenvalues.

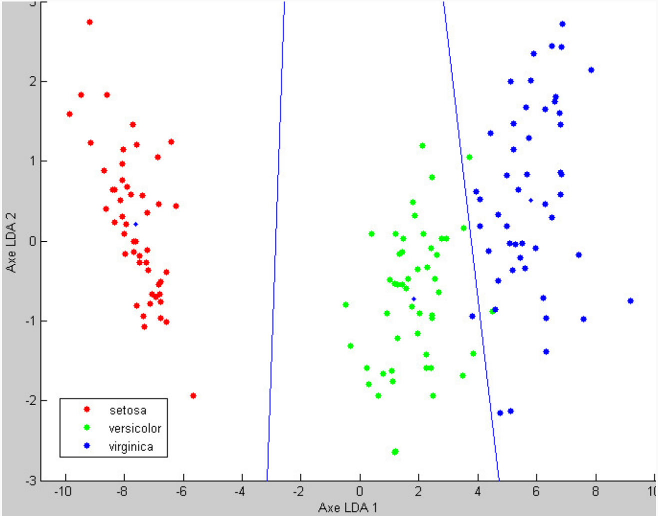
Classification with LDA

- How to assign a class to a new data item?
- The discriminant factors give the best representation of the separation of the q class centroids (in an orthonormal space).

$\Rightarrow$ for an individual $\mathbf{x}$ projected in factor space: assign the class whose center is nearest (in the sense of the Euclidean distance)

$\Rightarrow$ linear separation surfaces = median hyperplanes between the class centres

Example of LDA classification of IRIS database

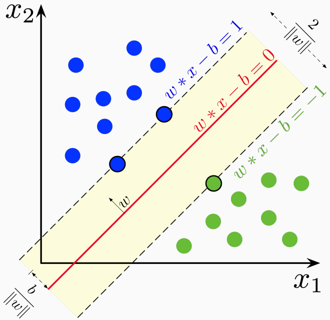


Other machine learning methods

- Conclusion: LDA remains a quick and easy way to visualize and classify separable data.
- Many other approaches to consider “difficult” cases
    - Quadratic discriminant analysis
    - Mix of probabilistic models (Gaussian)
    - Boosting
    - SVM
    - Perceptron, neural networks, convolutional neural networks...
    - ...

SVM: Support vector machine [Cortes and Vapnik, 1995]

- SVM is a discrimination technique which consists in separating sets of points (or classes) by a hyperplane maximizing the margin between these classes.



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors. Credit: Wikipedia

$$\begin{cases} \min \frac{1}{2}\|\mathbf{w}\|^2 \\ \forall i \;\; y_i(\mathbf{w}.x_i + \mathbf{w}_0) - 1 \geq 0 \end{cases}$$

$\Rightarrow$ Constrained convex optimization: use of Lagrange multiplier technique.

Decision function:

$$f(u) = \mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i y_i x_i.u + b\right)$$

Supplements on supervised learning

- Data: $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^d$, $y_i \in \Omega$
- Pattern recognition via a discriminant function (or model) f:

$$\begin{aligned} f: \quad & \mathbb{R}^d \to \Omega \\ & x \mapsto f(x) \end{aligned}$$

- Risk:

$$R(f) = \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i)}_{\text{empirical risk}} + \underbrace{c\gamma(f)}_{\text{regularization}}$$

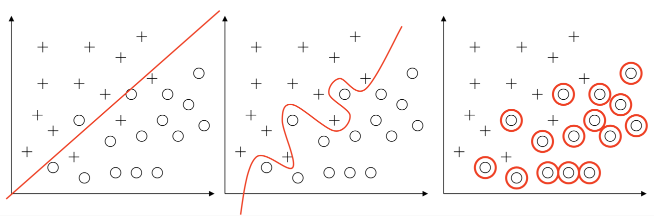
and L loss function

Learning error

- During learning, the performance of model f should be evaluated to:
  - Compare different models
  - Select relevant variables
  - To have an idea of the probability of correctly classifying a new data item (generalization error)
- To be banned: train and evaluate on the same set of data!
  - this introduces a bias because the algorithm is specialized for the train set

Example of under/over-fitting

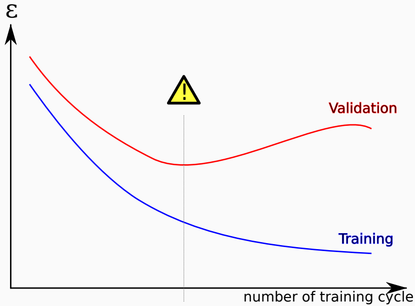
- How to select a good model f?
  - Too few parameters lead to under-fitting
  - Too many parameters for f compared to dataset size lead to over-fitting



- First idea:
  - Separation of the available data into 2 sets
  - Training / Test Calculation of the generalization error with the test set

Happy curve

- Under training/over training with neural networks

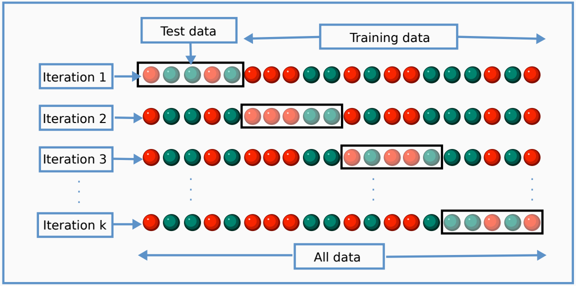


Credit: Wikipedia

- If the validation error increases (positive slope) while the training error steadily decreases (negative slope) then a situation of overtraining occurs. The best predictive and fitted model is where the validation error has its global minimum.

Cross validation

- What can we do with small datasets? Cross-validation
  - Divide the available data into K groups
  - For each group k: train on $K-1$ groups and test on group k
  - Generalization error = average of test errors



Credit: Wikipedia

- To go further with supervised learning: see chapter 5 of Goodfellow's book [2], and Bishop's book [1].

C. M Bishop.
Pattern Recognition and Machine Learning.
Springer, 2006.

I. Goodfellow, Y. Bengio, and A. Courville.
Deep Learning, chapter Machine learning basics.
MIT Press, 2016.
https://www.deeplearningbook.org/contents/ml.html.