Cours 1
Introduction
Codage
Réduction de dimension

Catherine ACHARD
Institut des Systèmes Intelligents et de Robotique

catherine.achard@sorbonne-universite.fr

Déroulement

14 h de cours + 2h de cc
15h de TP (6 séances de 2h + 1 séance de 3h)

2 cc de 1h (16 février et 13 avril)
1 examen de TP écrit de 1h (13 avril)

Introduction

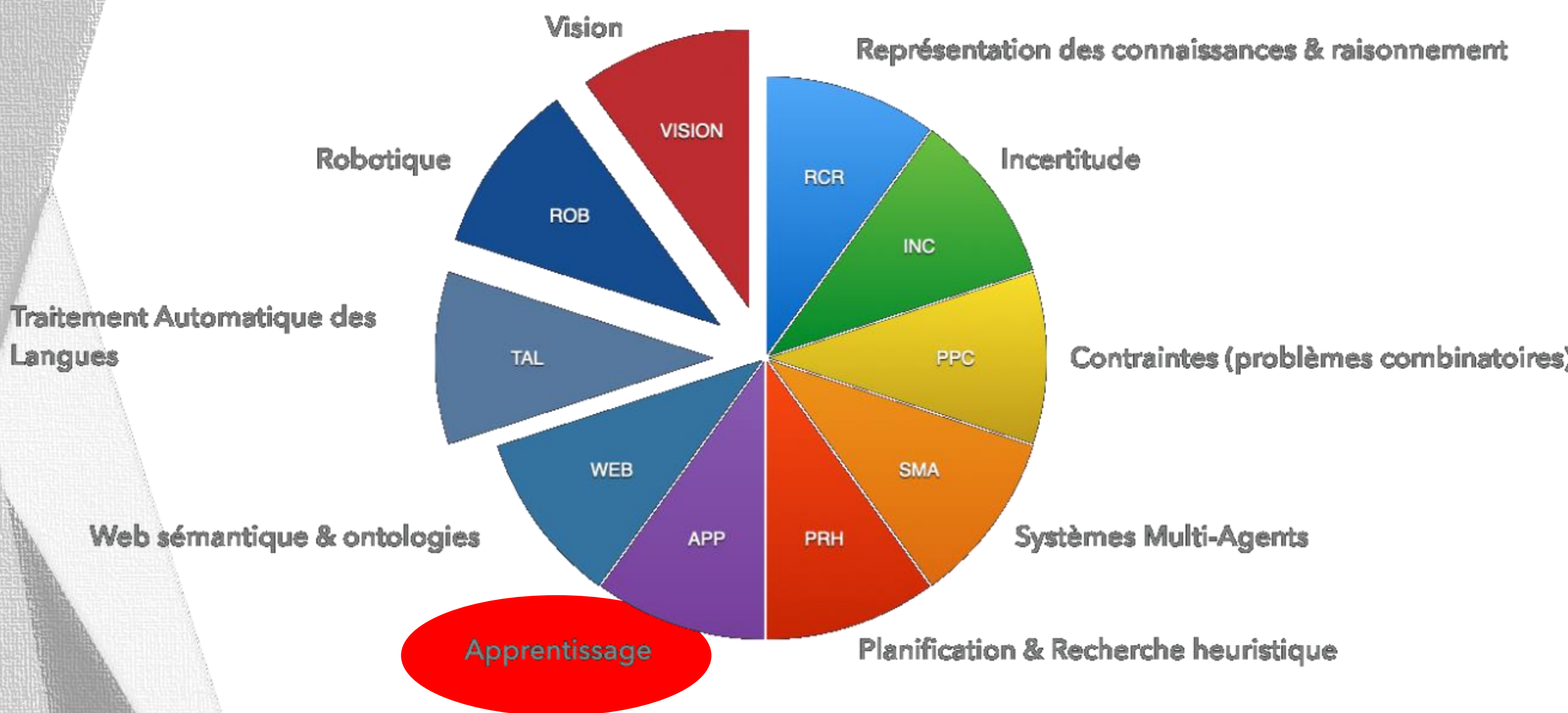
Références

- DUDA Richard, HART Peter STORK David, "**Pattern Classification** ". Wiley Sciences, 2nd edition, 2000.

- CHRISTOPHER M. BISHOP, “**Pattern Recognition and machine learning**”, springer, 2006

Introduction

Intelligence artificielle



Les différentes tâches

Classification

- But : L'algorithme doit dire à quelle catégorie appartient une entrée
- Exemple : L'image d'une personne ➔ son identité

Régression

- But : L'algorithme doit associer une valeur numérique à une entrée
- Exemple : Paramètres météorologique ➔ température à 24 heures

Retranscription

- But : A partir de l'observation d'une entrée, la transcrire sous une forme textuelle
- Exemple : A partir d'images de google street, donner les numéros de rue

Traduction

- But : convertir une séquence de symbole en une autre séquence de symbole dans un autre langage
- Exemple : traduction du français à l'anglais

Détection d'anomalie

- But : Trouver si une entrée est atypique ou non
- Exemple : détecter des voitures en sens contraire sur des images routières

Synthèse

- But : apprendre à générer de nouveaux exemples proches de ceux d'une base de données
- Application : Synthétiser des visages sous un autre point de vue

Débruitage

- But : Apprendre à générer des données non bruitées à partir de données bruitées

Applications

Média

- Facebook ou instagram personnalise la diffusion d'informations
- Linkedin propose des emplois pertinents, des nouveaux contacts personnalisés
- Détection de spam

Objets connectés/intelligents

- Reconnaissance de visages pour appareil photo
- Enceinte intelligente
- Véhicule intelligent : détection de véhicules, de piétons, prédiction de traffic
- Analyse de comportements anormaux
- Assistant personnel

Applications

Finance

- Détection de fraude à la CB
- Prédiction des valeurs financières

Commerce

- Recommandation personnalisée de produit sur internet
- Création de chatbots et de robots pour répondre aux appels téléphoniques des clients

Introduction

Applications

Transport

- Comment Uber fixe le prix d'une course?
- Comment Uber gère le co-voiturage
- Comment Uber gère le temps d'attente et affecte les véhicules aux clients

Voyage

- 90% des américains partagent leur expérience de voyage
- Tripadvisor reçoit 280 avis par minute
- Tripadvisor analyse les informations pour donner les plus pertinentes

Difficultés

Considérons le cas de la reconnaissance de visages (classification)

Problème de résolution

Peut-on définir une distance entre deux visages ?

Problème de pose

Problème d'expression faciale

Problème d'occultations

MARTINEZ, Aleix M. The AR face database. *CVC Technical Report24*, 1998.

Introduction

Raisonnons sur un cas simple

Reconnaître les truites et les saumons

Codage

Chaque poisson est représenté par un **vecteur de caractéristiques** aussi appelé **codage** ou **features**

Exemple

Supposons que l'on ait N poissons $\boldsymbol{x}_i, 1 \leq i \leq N$de classe $y_i, 1 \leq i \leq N$

$\boldsymbol{y_i}$ est la classe $\boldsymbol{x_i}$ $\in \{truite, saumon\}$ de l'exemple i (du poisson i)

$\boldsymbol{x_i}$ est le vecteur de caractéristiques de dimension $\boldsymbol{n}$:

- Si chaque poisson est représenté juste par sa taille, alors $n = 1$ et la base est constituée de N exemples $\{\boldsymbol{x}_i, y_i\}$ où $\boldsymbol{x}_i$ est de dimension 1
- Si chaque poisson est représenté par sa taille et sa teinte, alors $n = 2$ et la base est constituée de N exemples $\{x_i, y_i\}$ où $\boldsymbol{x}_i$ est de dimension 2

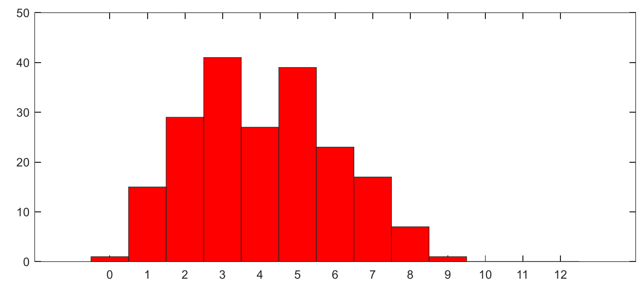
Introduction

Raisonnons sur un cas simple

Classification en 1 dimension : la taille des poissons
➔ Les x_i sont de dimension 1

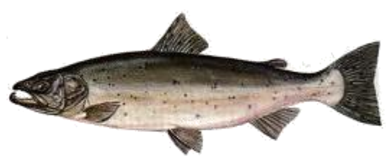
$\boxed{p(x/truite)}$
$\propto$ histogramme de la taille des truites

Nombre
de poissons

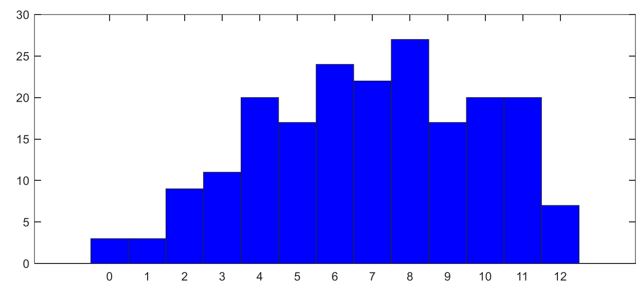


Taille des
poissons

$p(x/saumon)$
$\propto$ histogramme de la taille des saumons



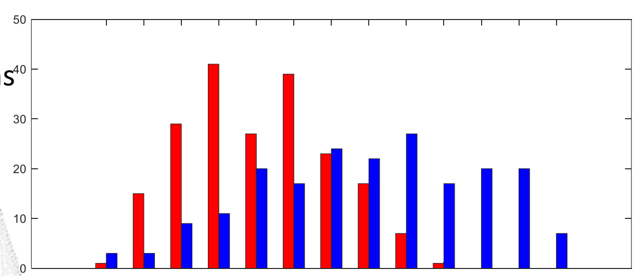
Nombre
de poissons



Taille des
poissons

Les deux superposés

Nombre
de poissons



Taille des
poissons

Aura-t-on une bonne classification ?

Seuil de décision

Introduction

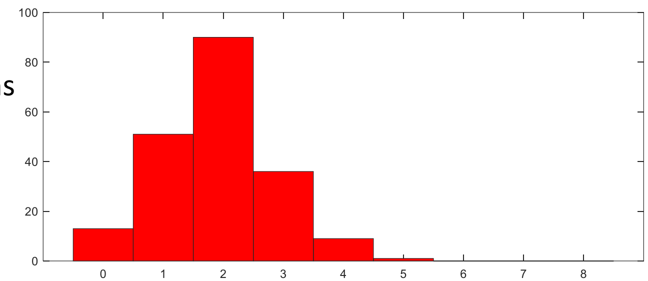
Raisonnons sur un cas simple

Classification en 1 dimension : la teinte des poissons
➔ Les x_i sont de dimension 1

$p(x/truite)$

∝ histogramme de la teinte des truites

Nombre
de poissons

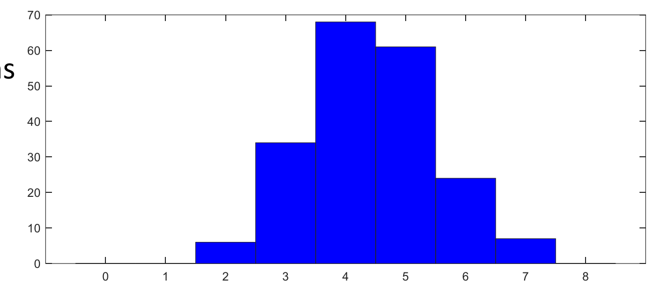


Teinte des
poissons

$p(x/saumon)$

∝ histogramme de la teintedes saumons

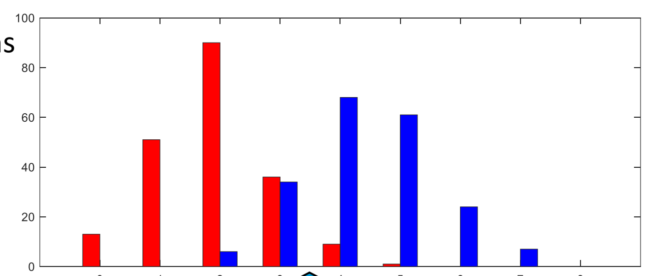
Nombre
de poissons



Teinte des
poissons

Les deux superposés

Nombre
de poissons



Teinte des
poissons

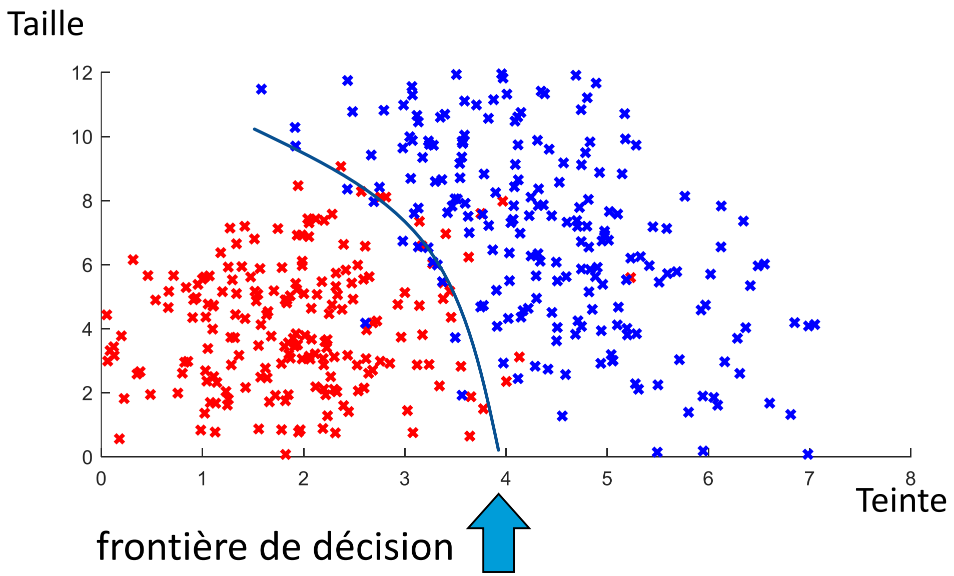
Est-ce mieux ?

Seuil de décision

Introduction
Raisonnons sur un cas simple

Classification en 2 dimensions : la teinte et la taille des poissons
➔ Les x_i sont de dimension 2

Classification avec la teinte et la longueur des poissons
➔ Vecteur de caractéristiques à 2 dimensions
➔ Chaque poisson est représenté par un point (dimension 2) dans le plan



Truite

Saumon

Il faut considérer les caractéristiques ensemble (et donc pas séparément)

Codage

Qu'est ce qu'un bon codage ?

Pouvoir discriminant

- Le codage doit être différent pour des exemples de classes différentes → **forte variance inter-classes**

Pouvoir unifiant

- Le codage doit être à peu près le même pour tous les exemples d'une même classe → **faible variance intra-classe**

Stabilité/invariance

- Codage le plus insensible possible au bruit
- En fonction des applications, invariance en translation, rotation, changement d'échelle

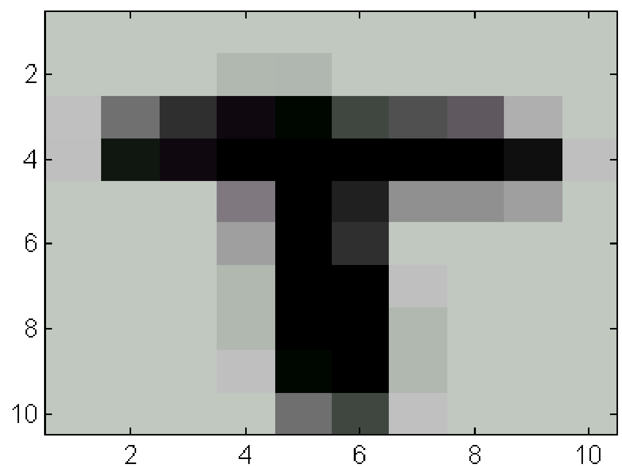
Faible dimension

- Plus le codage est de faible dimension, plus les temps de calcul seront faibles
- Augmenter la dimension peut détériorer les résultats de reconnaissance (malédiction des grandes dimensions→ compromis à trouver)

Codage
le codage rétinien

Mais on est vite en grande dimension

Exemple pour des lettres représentées pat un codage rétinien de taille 10x10
➔ **Vecteur de caractéristiques de dimension 100**



Codage
Fléau des grandes dimensions

Premier problème

- La complexité algorithmique des algorithmes de classification est souvent ***f*(*n*²)** **ou** ***f*(*n*³)** ou même, exponentielle

Second problème

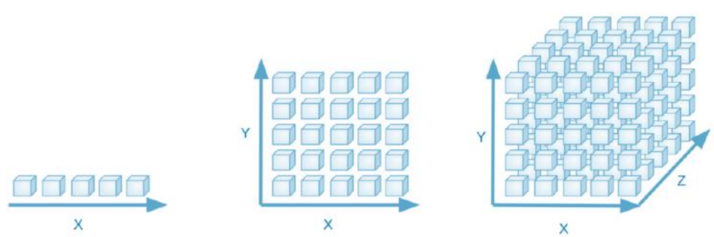
- Plus la dimension est élevée, plus il faut de données

Si on divise chaque dimension en 2,

Ex : avec deux exemples par dimension

- Si $n = 2$, $2^2=4$ données
- Si $n = 3$, $2^3=8$ données
- Si $n = 4$, $2^4=16$ données
- Si $n = 20$, $2^{20}=1.048.576$ données

➔ Si le nombre de données est fixe, plus on augmente la dimension, plus les données sont dispersées dans l'espace



Codage
Fléau des grandes dimensions

Troisième problème : la folie géométrique

Considérons des points uniformément répartis entre $[0,1]^n$

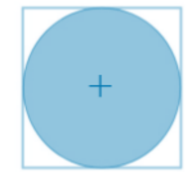
<u>Dimension 2</u>

Considérons un disque de rayon 0.5, inscrit dans un carré de rayon 1

Le disque ou le carré sont centrés sur la moyenne des points

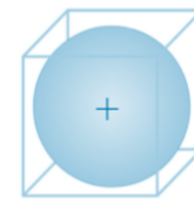
Tous les points à une distance inférieure à 0.5 du centre sont dans le cercle

Le rapport de surface est $\frac{\pi 0.5^2}{1} = 0.78$ ➔ **22% des points sont dans les coins**

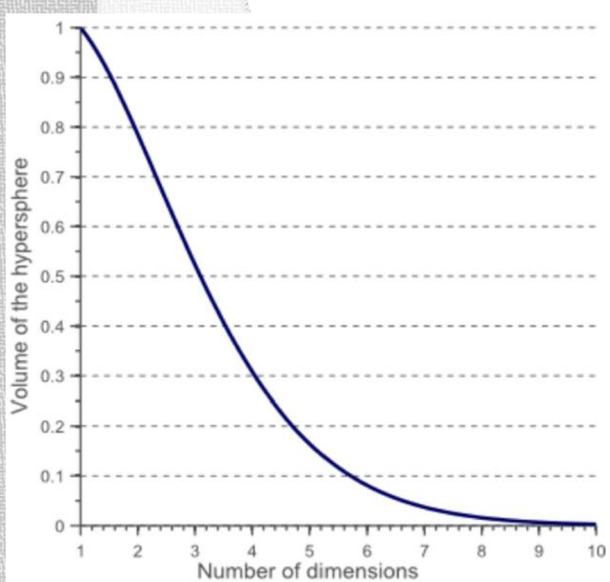


<u>Dimension 3</u>

Le rapport de surface est $\frac{4/3\pi 0.5^3}{1} = 0.17$ ➔ **83% des points sont dans les coins**



<u>Et en continuant</u>



Donc,

- la plupart des données va être dans les coins
- Ces points sont difficiles à classer car leurs valeurs diffèrent beaucoup des autres (ils sont dans les coins !)
- Plus la dimension augmente, plus il y a statistiquement de points dans les coins, difficiles à classer

Comment trouver un bon codage ?

Méthodes empiriques

- Choix des caractéristiques pertinentes

Méthodes exploratoires

- Algorithmes génétiques

Méthodes par apprentissage

- Boosting, forêt d'arbre aléatoire, deep learning

Méthodes statistiques pour réduire la dimension ou la taille des données

- Analyse en composantes principales
- Analyse discriminante linéaire
- Sélection/extraction de caractéristiques

Un bon codage aura une faible dimension

- Fléau des grandes dimensions

Matrice de Covariance

Rappel sur les matrices de covariance

Si on dispose de N vecteurs de données $\boldsymbol{x}_i$ de dimension n, leur matrice de covariance est estimée par :

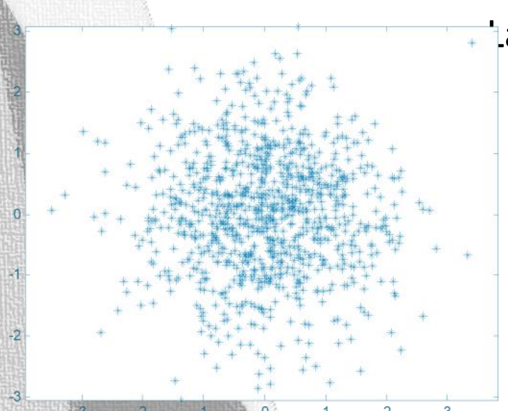
$$\boxed{\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \mu)(\boldsymbol{x}_i - \mu)^T}$$

où μ est le vecteur moyen des données $\boldsymbol{x}_i$

La matrice de covariance est de **dimension n x n**. Elle est **symétrique**. Ses valeurs représentent la **forme du nuage de points**

Exemple en dimension 2

On peut représenter chaque point $\boldsymbol{x}_i$ dans le plan.



La matrice de covariance est de dimension 2x2

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

Matrice de Covariance

Exercice

On dispose de 4 points de dimension 2 :

$$\boldsymbol{x}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad \boldsymbol{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Que valent N et n ?

Quelle est la moyenne du nuage de points ?

Quelle est la matrice de covariance du nuage de points ?

Matrice de Covariance

Exercice

On dispose de 4 points de dimension 2 :

$\boldsymbol{x}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $\boldsymbol{x}_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ $\boldsymbol{x}_3 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ $\boldsymbol{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Que valent N et n ?

Quelle est la moyenne du nuage de points ?

Quelle est la matrice de covariance du nuage de points ?

N=4 (nombre de points) et n=2 (dimension de chaque point)

La moyenne est :

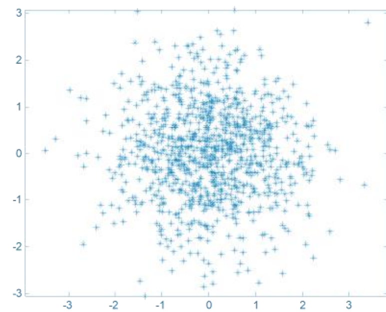
$\boldsymbol{\mu} = (\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3 + \boldsymbol{x}_4)/4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

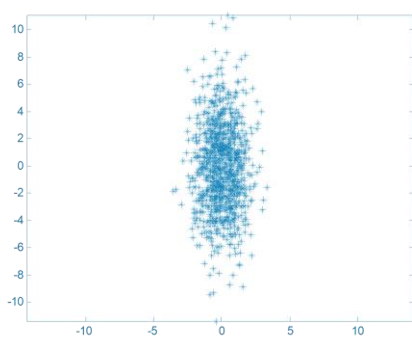
La matrice de covariance est définie par : $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$

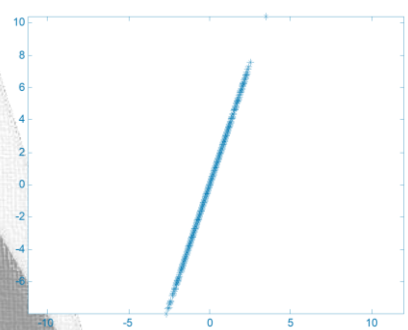
Donc, $\Sigma = \frac{1}{4}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & -1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix}\begin{bmatrix} -1 & -1 \end{bmatrix}\right)$

$\Sigma = \frac{1}{4}\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Matrice de Covariance







Exercice

Attribuer chaque ensemble de points à sa matrice de covariance

A B C

$$\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} \quad \begin{pmatrix} 1 & -0{,}02 \\ -0{,}02 & 9{,}46 \end{pmatrix} \quad \begin{pmatrix} 1{,}01 & -0{,}01 \\ -0{,}01 & 0{,}99 \end{pmatrix}$$

Analyse en composante principale (ACP)
Principal Component Analysis

- Projection des données depuis l'espace de dimension *n* vers un espace de dimension *d* (*d* < *n*) en gardant le maximum d'information

➔ Normalisation préalable des données

Les données $\{\boldsymbol{x_i}\}_{1 \leq i \leq \boldsymbol{N}}$ de dimension *n* sont rangées dans un tableau **X**

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	…	$\boldsymbol{x_N}$
dimension1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$		$x_{1,4}$
dimension2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$		$x_{2,4}$
dimension3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$		$x_{3,4}$
dimension4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$		$x_{4,4}$
dimension5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$		$x_{5,4}$

Le tableau contient des informations de natures différentes. Il faut
- **centrer les données** : on soustrait la moyenne de la ligne à chaque valeur
- **normaliser les données** : on divise chaque valeur par l'écart type de sa ligne

Analyse en composante principale (ACP)

Exercice :

On considère les points suivants :

$$\boldsymbol{x}_1 = \begin{bmatrix} 90 \\ 0.9 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 100 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 110 \\ 0.8 \end{bmatrix} \quad \boldsymbol{x}_4 = \begin{bmatrix} 110 \\ 2 \end{bmatrix} \quad \boldsymbol{x}_5 = \begin{bmatrix} 105 \\ 1.9 \end{bmatrix} \quad \boldsymbol{x}_6 = \begin{bmatrix} 115 \\ 2.1 \end{bmatrix}$$

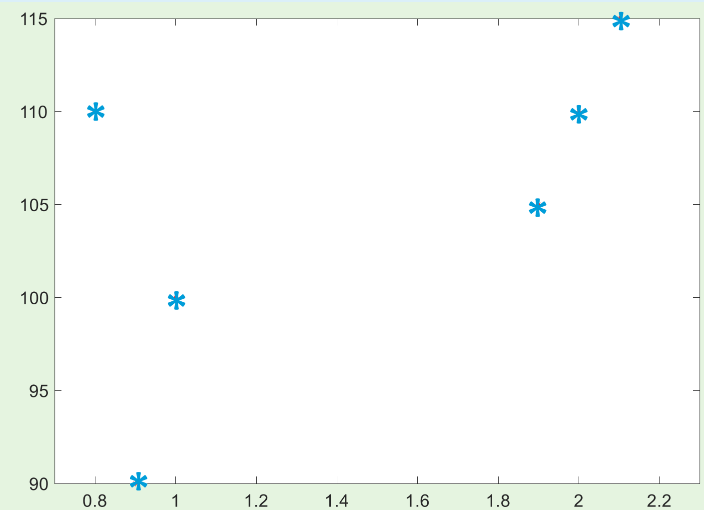
- Représenter ces points dans l'espace
- Est-ce qu'une distance euclidienne entre ces points semble pertinente ?
- Normaliser ces données

Exercice :

On considère les points suivants :

$$\boldsymbol{x}_1 = \begin{bmatrix} 90 \\ 0.9 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 100 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 110 \\ 0.8 \end{bmatrix} \quad \boldsymbol{x}_4 = \begin{bmatrix} 110 \\ 2 \end{bmatrix} \quad \boldsymbol{x}_5 = \begin{bmatrix} 105 \\ 1.9 \end{bmatrix} \quad \boldsymbol{x}_6 = \begin{bmatrix} 115 \\ 2.1 \end{bmatrix}$$

- Représenter ces points dans l'espace
- Est-ce qu'une distance euclidienne entre ces points semble pertinente ?
- Normaliser ces données



Non, la distance euclidienne n'est pas pertinente car les deux dimensions n'ont pas du tout le même ordre de grandeur

Analyse en composante principale (ACP)

Exercice :

On considère les points suivants :

$$\boldsymbol{x}_1 = \begin{bmatrix} 90 \\ 0.9 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 100 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 110 \\ 0.8 \end{bmatrix} \quad \boldsymbol{x}_4 = \begin{bmatrix} 110 \\ 2 \end{bmatrix} \quad \boldsymbol{x}_5 = \begin{bmatrix} 105 \\ 1.9 \end{bmatrix} \quad \boldsymbol{x}_6 = \begin{bmatrix} 115 \\ 2.1 \end{bmatrix}$$

- Représenter ces points dans l'espace
- Est-ce qu'une distance euclidienne entre ces points semble pertinente ?
- Normaliser ces données

La moyenne est :

$$\begin{bmatrix} 105 \\ 1.45 \end{bmatrix}$$

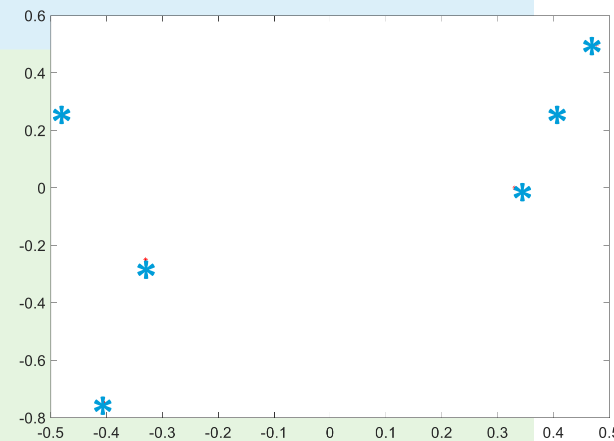
Et les écart-types sont :

$$\begin{bmatrix} 20 \\ 1.362 \end{bmatrix}$$

Les données normalisées dont donc :

$$\boldsymbol{x}_1 = \begin{bmatrix} (90-105)/20 \\ (0.9-1.45)/1.362 \end{bmatrix} = \begin{bmatrix} -0.75 \\ -0.4038 \end{bmatrix}$$

$$\boldsymbol{x}_2 = \begin{bmatrix} -0.25 \\ -0.3304 \end{bmatrix} \quad \boldsymbol{x}_3 = \begin{bmatrix} 0.25 \\ -0.47721 \end{bmatrix} \quad \boldsymbol{x}_4 = \begin{bmatrix} 0.25 \\ 0.4038 \end{bmatrix} \quad \boldsymbol{x}_5 = \begin{bmatrix} 0 \\ 0.3304 \end{bmatrix} \quad \boldsymbol{x}_6 = \begin{bmatrix} 0.51 \\ 0.4772 \end{bmatrix}$$

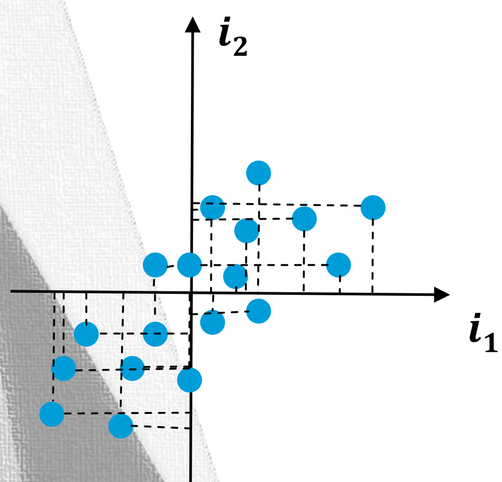


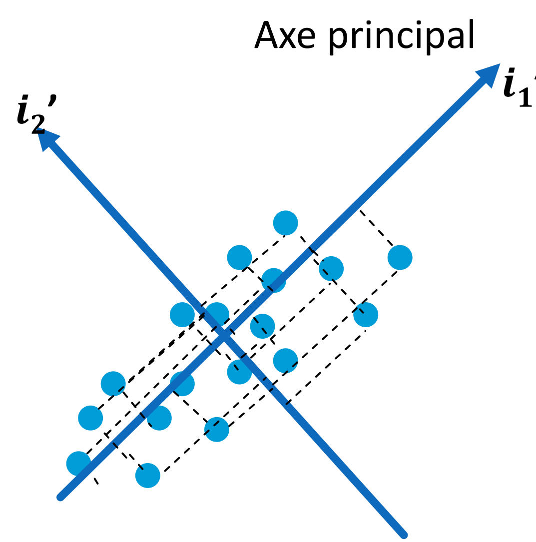
Analyse en composante principale (ACP)

L'idée va être de changer le système d'axe de manière à ce que le maximum d'information soit contenu sur les premiers axes.

Exemple

Pour des données en dimension 2 dans le repère $(O, \boldsymbol{i_1}, \boldsymbol{i_2})$, l'ACP va donner un nouveau repère $(O, \boldsymbol{i_1}', \boldsymbol{i_2}')$ tel que le maximum d'information soit porté par $\boldsymbol{i_1}'$.
$\boldsymbol{i_1}'$correspondra à **l'élongation maximale** du nuage de point.





De manière plus formelle, chaque point s'exprime dans le repère initial par:

$$x_{i1}\boldsymbol{i_1}+x_{i2}\boldsymbol{i_2}+\ldots+x_{in}\boldsymbol{i_n} \qquad \boldsymbol{x_i}=\begin{bmatrix} x_{i1} \\ \vdots \\ x_{in} \end{bmatrix}$$

Et dans le nouveau repère par :

$$x_{i1}'\boldsymbol{i_1}'+x_{i2}'\boldsymbol{i_2}'+\ldots+x_{in}'\boldsymbol{i_n}' \qquad \boldsymbol{x_i}'=\begin{bmatrix} x_{i1}' \\ \vdots \\ x_{in}' \end{bmatrix}$$

La projection des points sur l'axe $\boldsymbol{i_1}'$ (par exemple) amène à la coordonnée x_{i1}'et se fait avec:

$$\boxed{x_{i1}'=\boldsymbol{x_i}^T\boldsymbol{i_1}'} \text{avec } \boldsymbol{i_1}'^T\boldsymbol{i_1}'=1$$

La variance des données sur le premier axe par exemple sera:

$$\sigma=\frac{1}{N}\sum_{i=1}^{N}x_{i1}'^2=\frac{1}{N}\sum_{i=1}^{N}x_{i1}'^T x_{i1}'=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{i_1}'^T\boldsymbol{x_i}\boldsymbol{x_i}^T\boldsymbol{i_1}'=\frac{1}{N}\boldsymbol{i_1}'^T\left(\sum_{i=1}^{N}\boldsymbol{x_i}\boldsymbol{x_i}^T\right)\boldsymbol{i_1}'$$

$$\boxed{\sigma=\boldsymbol{i_1}'^T\,\boldsymbol{\Sigma}\,\boldsymbol{i_1}'} \quad \text{et} \quad \boxed{\boldsymbol{i_1}'^T\boldsymbol{i_1}'=1}$$

Où $\boldsymbol{\Sigma}$ est la matrice de covariance des données tq $\boldsymbol{\Sigma}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x_i}\boldsymbol{x_i}^T$

Analyse en composante principale (ACP)

➔ On recherche l'axe $\boldsymbol{i_1}'$ qui a la variance σ maximale, sous contrainte $\boldsymbol{i_1}'^T\boldsymbol{i_1}' = 1$

On maximise le lagrangien:

$$L=\boldsymbol{i_1}'^T\,\boldsymbol{\Sigma}\boldsymbol{i_1}' - \lambda\left(\boldsymbol{i_1}'^T\,\boldsymbol{i_1}' - 1\right)$$

Rappel sur les matrices

$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$

$(\boldsymbol{A}+\boldsymbol{B})^T = \boldsymbol{A}^T + \boldsymbol{B}^T$

$\frac{\partial a^T b}{\partial a} = \frac{\partial b^T a}{\partial a} = \mathrm{b}$

$\frac{\partial a^T \boldsymbol{B} a}{\partial a} = 2\mathbf{B}\mathrm{a}$

En annulant la dérivée de L par rapport à $\boldsymbol{i_1}'$, on obtient:

$$\frac{\partial L}{\partial\, \boldsymbol{i_1}'} = 0 \quad ➔ \quad \boldsymbol{\Sigma}\,\boldsymbol{i_1}' = \lambda\,\boldsymbol{i_1}'$$

➔ $\boldsymbol{i_1}'$ est un vecteur propre de la matrice $\boldsymbol{\Sigma}$ associé à la valeur propre λ

➔ En multipliant l'équation par $\boldsymbol{i_1}'^T$, on a $\boldsymbol{i_1}'^T\boldsymbol{\Sigma}\,\boldsymbol{i_1}' = \lambda\boldsymbol{i_1}'^T\boldsymbol{i_1}'$
Or $\boldsymbol{i_1}'^T\boldsymbol{i_1}' = 1$. Donc, $\boldsymbol{i_1}'^T\boldsymbol{\Sigma}\,\boldsymbol{i_1}' = \lambda$

Et comme on cherche à maximiser $\boldsymbol{i_1}'^T\boldsymbol{\Sigma}\,\boldsymbol{i_1}'$, on en déduit que

$\boldsymbol{i_1}'$ est le vecteur propre qui correspond à la plus grande valeur propre de $\boldsymbol{\Sigma}$ qui est la matrice de covariance des points

➔ **Le premier axe de projection est le premier vecteur propre de la matrice de covariance $\boldsymbol{\Sigma}$**
Le second axe correspondra au second vecteur propre de la matrice,…

Analyse en composante principale (ACP)

La base de projection correspond aux vecteurs propres rangés par ordre de valeurs propres décroissantes. Chaque point s'exprime dans cette base par :

$$x_{i1}'\boldsymbol{i_1}' + x_{i2}'\boldsymbol{i_2}' + \; \ldots . + x_{in}'\boldsymbol{i_n}' \qquad \boldsymbol{x_i}' = \begin{bmatrix} x_{i1}' \\ \vdots \\ x_{in}' \end{bmatrix}$$

Avec $x_{ij}' = \boldsymbol{x_i}^T . \boldsymbol{i_j}'$ (x_{ij}'scalaire)

La réduction de dimension consiste à ne garder que les premières composantes :

$$\begin{bmatrix} x_{i1}' \\ \vdots \\ x_{id}' \end{bmatrix} \quad \text{avec } d << n$$

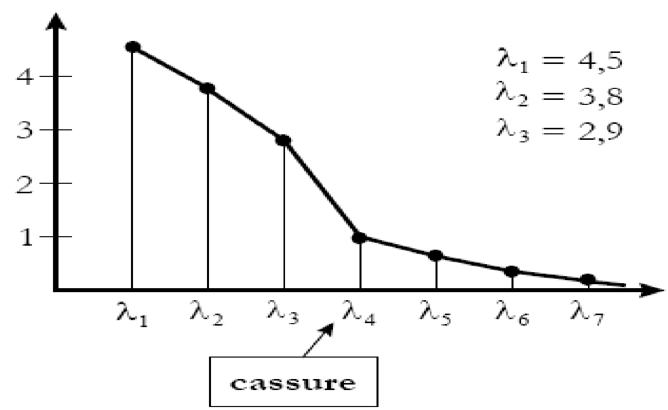
On définit **l'inertie** portée par les d premiers axes par

$$\frac{\lambda_1+\lambda_2+\cdots+\lambda_d}{\lambda_1+\lambda_2+\lambda_3+\cdots+\lambda_n}$$ où λ_j est la $j^{\text{ième}}$ valeur propre

Analyse en composante principale (ACP)

Comment connaître le nombre d'axes à conserver ?

1. Avec un pourcentage d'inertie souhaité *a priori*
2. On divise l'inertie totale par la dimension initiale pour connaître l'inertie moyenne par variable. On conserve tous les axes ayant une inertie supérieure à cette moyenne
3. On observe l'évolution des valeurs propres



4. En observant le taux de reconnaissance par exemple en fonction du nombre de dimensions gardées

Analyse en composante principale (ACP)

Exemples d'ACP sur des images de visages

On dispose d'une base de références composée de N=270 visages,
Chaque visage a pour dimension 38x38=1444 pixels → n=1444
On range tous ces visages dans une matrice de dimension 270x1444.

Quelle est la dimension de la matrice de covariance ?
Quelle est la dimension de chaque vecteur propre ?
Comment les transformer en eigen image

Analyse en composante principale (ACP)

Exemples d’ACP sur des images de visages

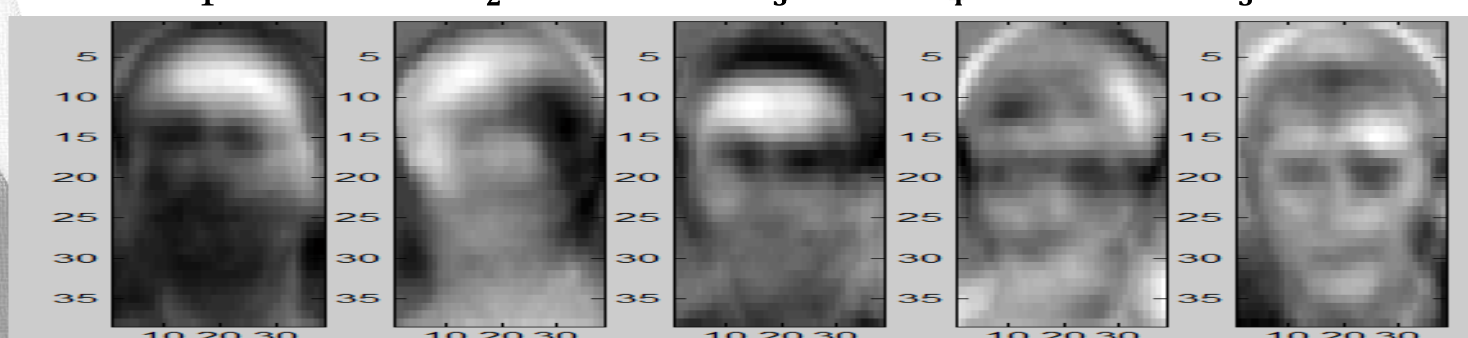
On dispose d’une base de références de 270 visages,
Chaque visage a pour dimension 38x38=1444 pixels ➔ n=1444

On range tous ces visages dans une matrice de dimension 1444x270.

On prétraite les données puis on calcule la matrice de covariance de cette grosse matrice. Elle est de dimension 1444x1444.

On calcule les valeurs propres et les vecteurs propres de cette matrice.
Chaque vecteur propre a pour dimension 1x1444. On peut remettre chacun d’eux sous la forme d’une matrice de dimension 38*38.
Les 5 premiers vecteurs propres (eigen image) :

$\boldsymbol{i_1}'$ $\boldsymbol{i_2}'$ $\boldsymbol{i_3}'$ $\boldsymbol{i_4}'$ $\boldsymbol{i_5}'$



Analyse en composante principale (ACP)

Si on ne conserve que ces 5 dimensions, chaque visage de la base s'exprime comme une combinaison linéaire de ces 5 'eigen-image'

$$x_{i1}'\boldsymbol{i_1}' + x_{i2}'\boldsymbol{i_2}' + \ \ldots . + x i_5' \boldsymbol{i_5}'$$

Ainsi, tous les exemples ne seront plus représentés que par un vecteur de dimension 5.

A partir de ce vecteur, on peut :

- Reconstruire les visages, on aura alors fait de la compression
- Reconnaitre les visages

Comment ?

Analyse en composante principale (ACP)

Si on ne conserve que ces 5 dimensions, chaque visage de la base s'exprime comme une combinaison linéaire de ces 5 'eigen-image'

$$x_{i1}'\boldsymbol{i_1}' + x_{i2}'\boldsymbol{i_2}' + \;\ldots. + xi_5'\boldsymbol{i_5}'$$

Ainsi, tous les exemples ne seront plus représentés que par un vecteur de dimension 5.

A partir de ce vecteur, on peut :

- Reconstruire les visages, on aura alors fait de la compression
- Reconnaitre les visages

Comment ?

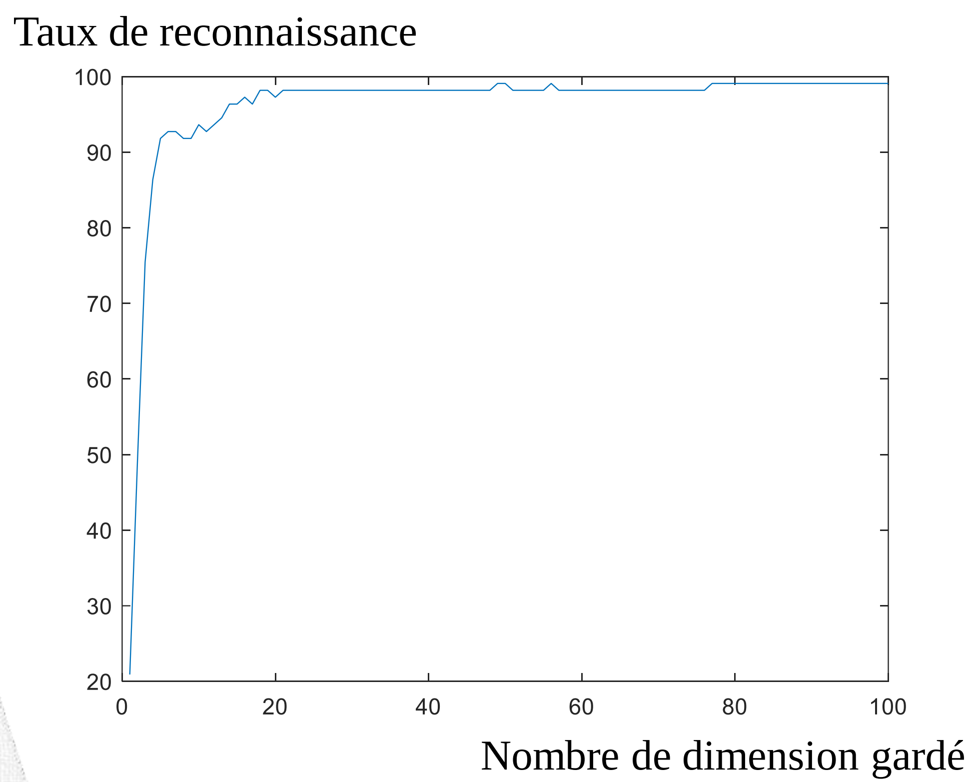
Si on garde l'exemple où on ne conserve que 5 dimensions, chaque visage est représenté uniquement par un vecteur de dimension 5 : $\boldsymbol{x_i}' = (x_{i1}'\; x_{i2}' \ldots xi_5')^T$

- Pour le reconstruire, à la décompression, on utilise les eigen images
$$x_{i1}'\boldsymbol{i_1}' + x_{i2}'\boldsymbol{i_2}' + \;\ldots. + xi_5'\boldsymbol{i_5}'$$
- Pour le reconnaitre, on utilise le vecteur $\boldsymbol{x_i}' = (x_{i1}'\; x_{i2}' \ldots xi_5')^T$ comme codage du visage

Analyse en composante principale (ACP)

Exemple sur une base de visages

270 images de visages de taille 38x38=1444
10 identités



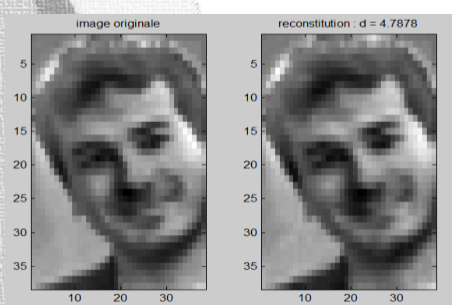
En gardant seulement 40 dimensions (2,7%), on est très proche du taux de reconnaissance maximal

Analyse en composante principale (ACP)

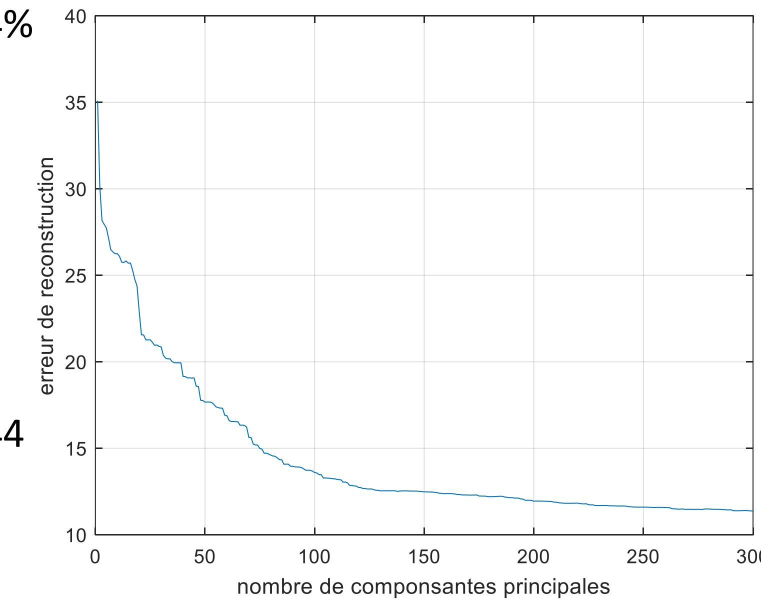
Sur la même base, en reconstruction

Reconstruction avec
5 « eigen images »
Compression = 1-5/1444%
= 99,6%

Reconstruction avec
49 « eigen images »
Compression = 1-49/1444
= 96,6%



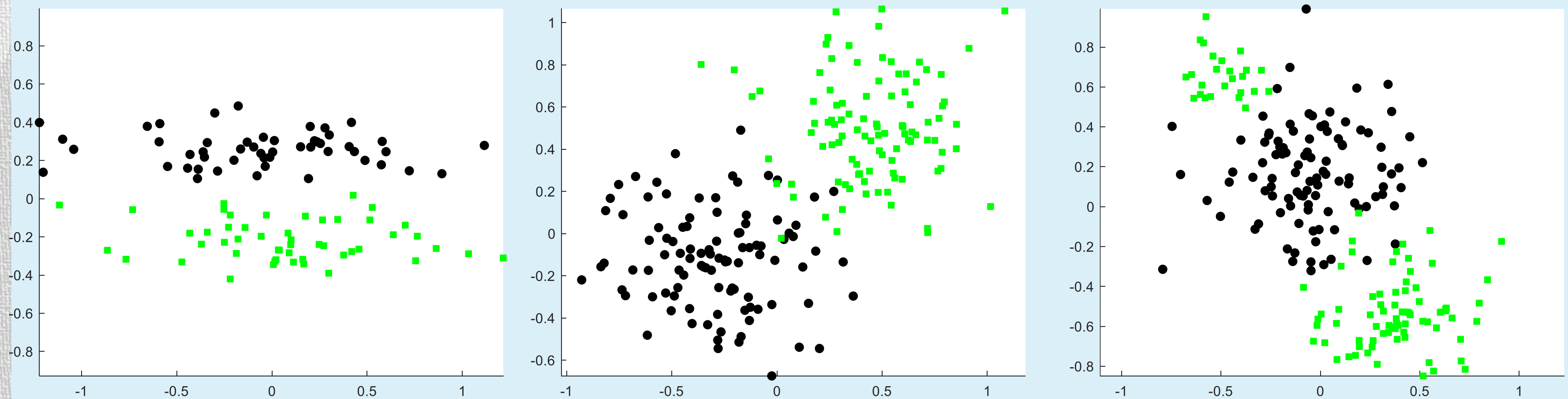
Reconstruction avec
144 « eigen images »
Compression = 1-144/1444
=90%



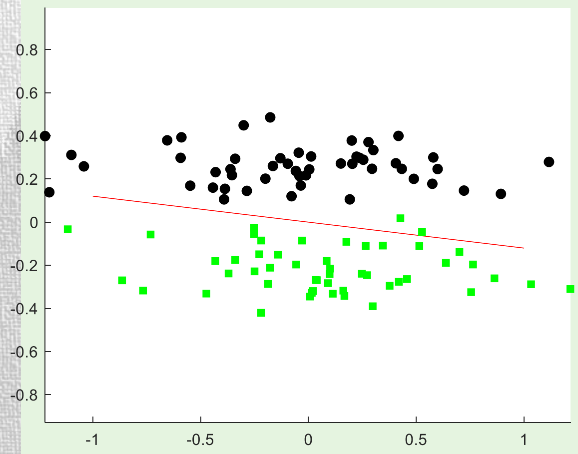
Analyse en composante principale (ACP)

Exercice

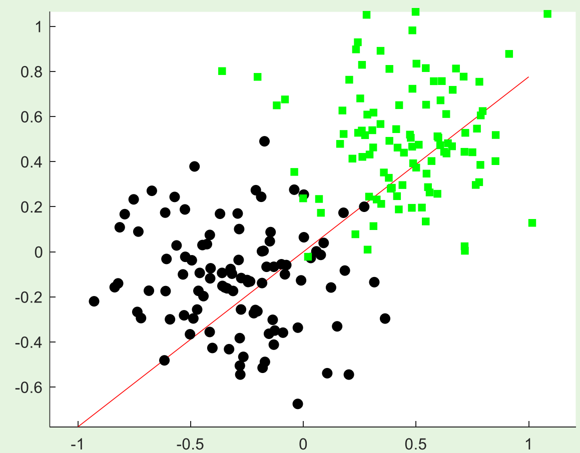
Pour ces trois ensembles de données, tracer l’axe de projection obtenu par ACP.
Conclusion



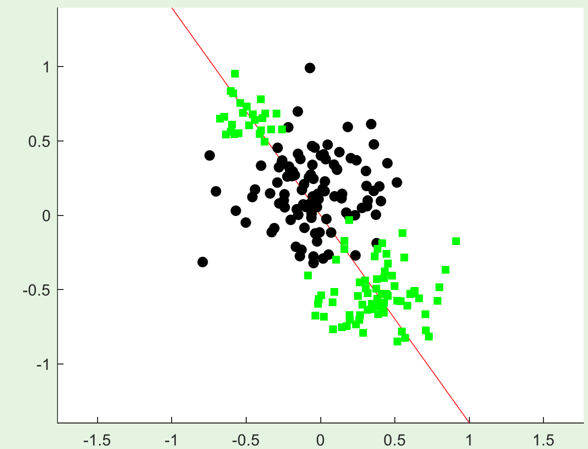
Analyse en composante principale (ACP)



Très mauvais



Bien



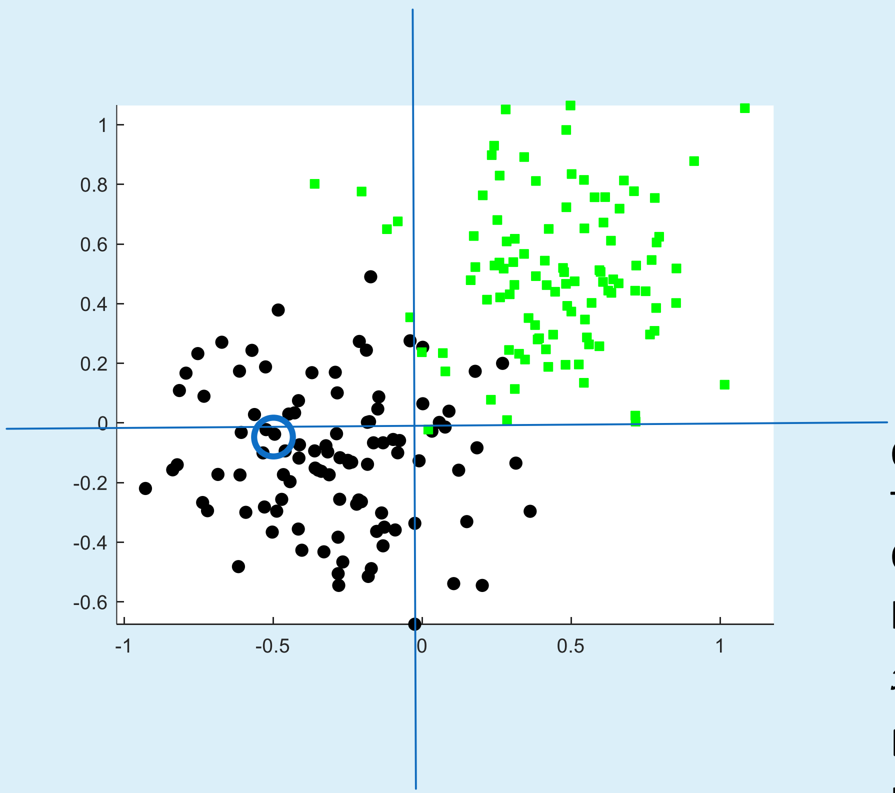
Bien

Analyse en composante principale (ACP)

Exercice

On considère le nuage de points ci-dessous dans un espace de dimension n=2.

On souhaite réduire la dimension par ACP



Matrice de covariance de tout le nuage :

0.2100	0.1119
0.1119	0.1528

Valeurs propres :

λ_1=59.0820 et λ_2=13.1152

Vecteurs propres :

$$v_1 = \begin{bmatrix} -0.7900 \\ -0.6132 \end{bmatrix} \text{ et } v_2 = \begin{bmatrix} 0.6132 \\ -0.7900 \end{bmatrix}$$

Quel sera le premier vecteur de projection ?

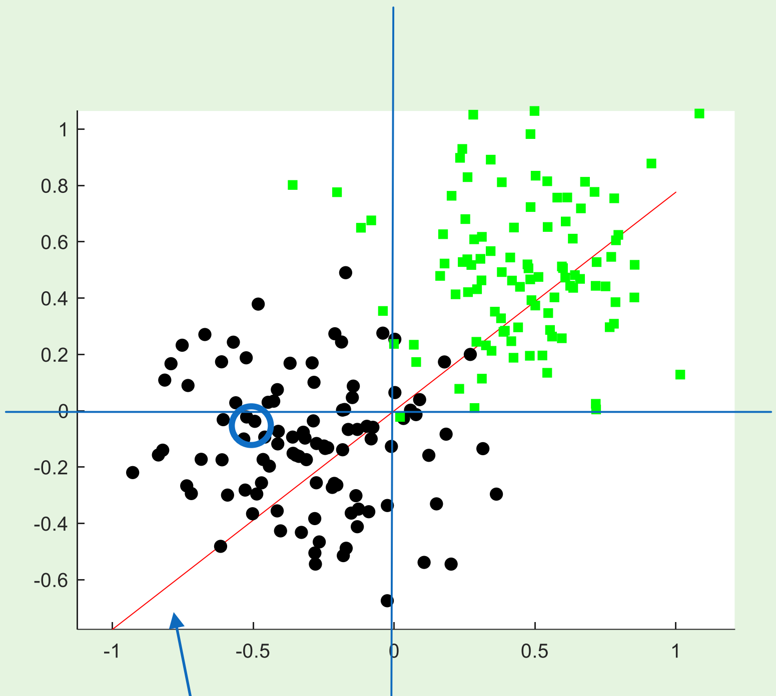
Tracer le

Quelle seront les nouvelles coordonnées du point

$$x = \begin{pmatrix} -0.5239 \\ -0.0222 \end{pmatrix}$$

Et donc, sa nouvelle coordonnée dans l'espace réduit ?

Analyse en composante principale (ACP)



Axe de projection

Matrice de covariance de tout le nuage :

0.2100	0.1119
0.1119	0.1528

Plus grande valeur propre

Valeurs propres :

λ_1=59.0820 et λ_2=13.1152

Vecteurs propres :

$$v_1 = \begin{bmatrix} -0.7900 \\ -0.6132 \end{bmatrix} \text{ et } v_2 = \begin{bmatrix} 0.6132 \\ -0.7900 \end{bmatrix}$$

$$x = \begin{pmatrix} -0.5239 \\ -0.0222 \end{pmatrix}$$

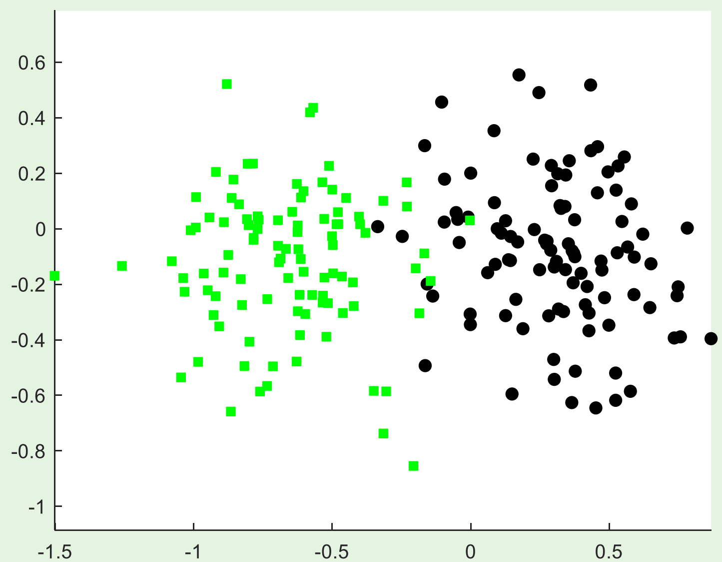
$$x_{i1}' = (-0.5239 \quad -0.0222)\begin{pmatrix} -0.7900 \\ -0.6132 \end{pmatrix} = 0.427$$

$$x_{i2}' = (-0.5239 \quad -0.0222)\begin{pmatrix} 0.6132 \\ -0.7900 \end{pmatrix} = -0.304$$

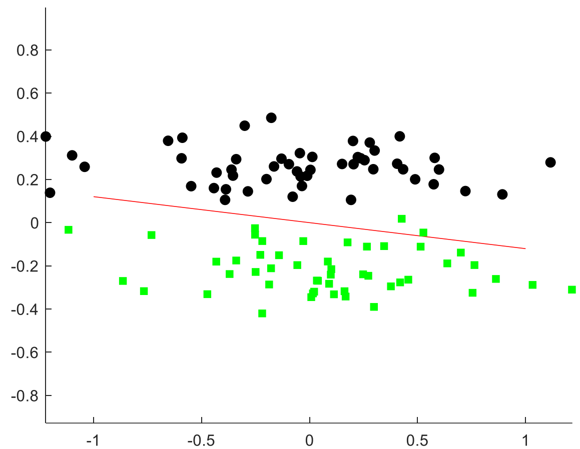
Analyse en composante principale (ACP)

Données après projection en gardant les deux axes

Le premier axe a bien la plus grande variance



Analyse en composante principale (ACP)



Mais comment gérer les cas où l'ACP mélange les données ?

➔ Il faut tenir compte des classes !
➔ Analyse discriminante linéaire

Analyse Discriminante Linéaire (LDA)
Linear Discriminant Analysis

But

Tenir compte de la répartition des points dans les classes

Dans le nouvel espace, on voudra

- Que chaque classe soit très compacte ➔ **petite variance intra-classe**
- Que 2 classes différentes soient très éloignées ➔ **grande variance inter-classe**

➔ **Maximise le ratio variance inter classe / variance intra classe.**

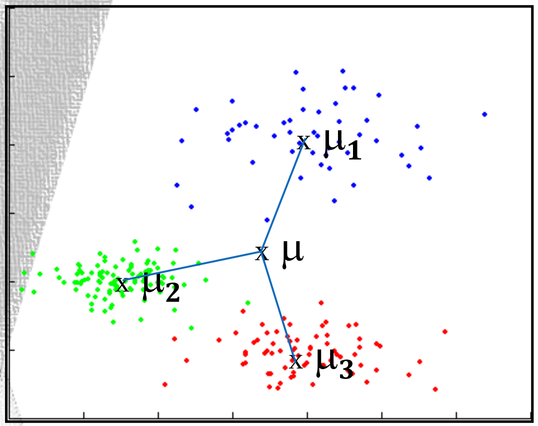
Problème

Comment définir les variances intra et inter classes ?

Supposons que l'on ait un problème à K classes où chaque point est représenté par un vecteur de dimension *n*:

$$x_{i1}\boldsymbol{i_1}+x_{i2}\boldsymbol{i_2}+\ldots+x_{in}\boldsymbol{i_n} \quad \textbf{et} \quad \boldsymbol{x_i} = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{in} \end{bmatrix}$$

Analyse Discriminante Linéaire (LDA)



On définit la moyenne de chaque classe et la moyenne globale

$$\mu_1, \mu_2, \dots, \mu_K,$$
$$\mu = p_1 * \mu_1 + p_2 * \mu_2 + \dots + p_K * \mu_K$$

Où $p_1, p_2, \dots\ p_K$ sont les probabilités de chaque classe : le nombre d'exemples de la classe divisé par le nombre d'exemples total

On définit également la matrice de covariance de chaque classe

$$\Sigma_1, \Sigma_2, \dots \Sigma_K$$

Dispersion intra classe

La matrice de covariance intra classe est tout simplement la moyenne pondérée des matrices de covariance

$$\Sigma_{intra} = p_1 * \Sigma_1 + p_2 * \Sigma_2 + \dots + p_K * \Sigma_K$$

Dispersion inter classe

La matrice de covariance inter classe est la matrice de covariance entre les centres des classes : elle mesure l'éloignement des classes. On tient également compte de la probabilité de chaque classe

$$\Sigma_{inter} = p_1(\mu_1 - \mu)(\mu_1 - \mu)^T + p_2(\mu_2 - \mu)(\mu_2 - \mu)^T + \dots + p_K(\mu_K - \mu)(\mu_K - \mu)^T$$

Analyse Discriminante Linéaire (LDA)

On cherche l'axe de projection $\boldsymbol{i_1}'$ qui maximise le rapport variance inter /variance intra

Avec la même démonstration que pour l'ACP, on trouve que :

➔ $\boldsymbol{i_1}'$ est le vecteur propre de $\boldsymbol{\Sigma_{intra}^{-1}\Sigma_{inter}}$ qui correspond à la plus grande valeur propre

Les autres axes de projection correspondent aux autres vecteurs propres de $\boldsymbol{\Sigma_{intra}^{-1}\Sigma_{inter}}$ classés par valeur propres décroissantes

Analyse Discriminante Linéaire (LDA)

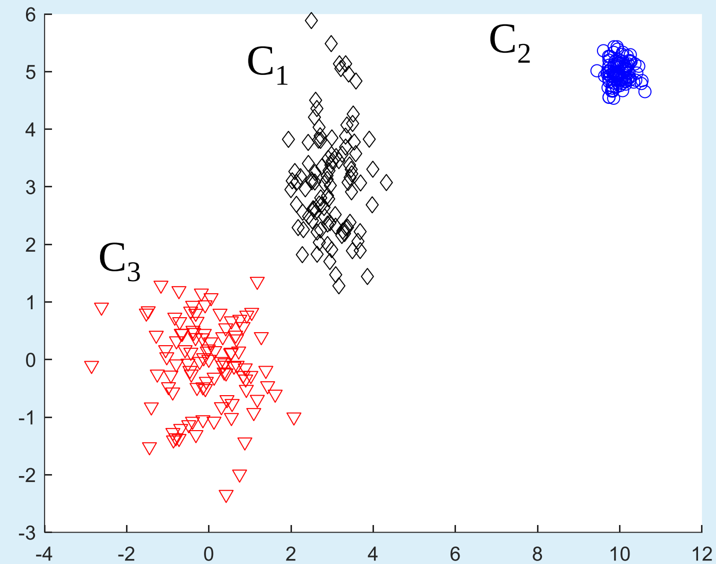
Exercice

On considère les données ci-contre
Les 3 matrices intra sont :

$$\begin{pmatrix} 0.72 & 0.04 \\ 0.04 & 0.74 \end{pmatrix} \begin{pmatrix} 0.22 & -0.03 \\ -0.03 & 0.91 \end{pmatrix}$$

$$\begin{pmatrix} 0.04 & 0.00 \\ 0.00 & 0.04 \end{pmatrix}$$

Réattribuer chaque matrice à chaque classe



Sachant qu'il y a 100 points dans chaque classe, quelle est la probabilité des classes ?
Retrouver $\boldsymbol{\Sigma}_{intra}$

Analyse Discriminante Linéaire (LDA)

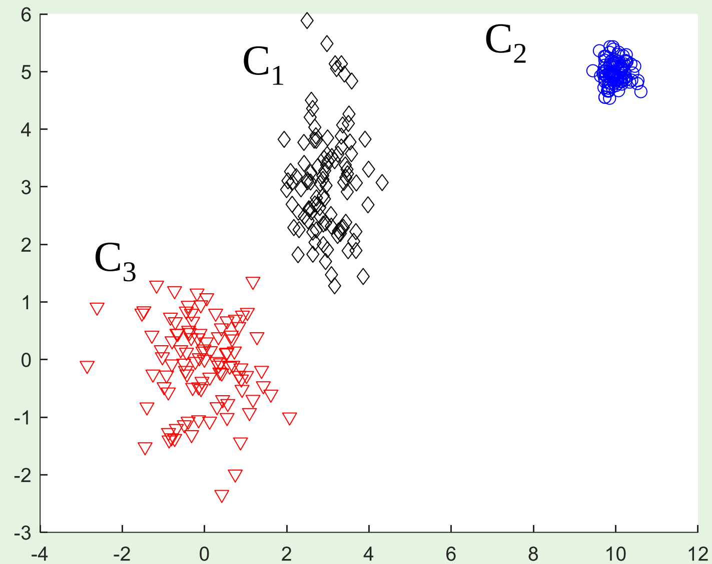
Exercice

On considère les données ci-contre
Les 3 matrices intra sont :

C3 $\begin{pmatrix} 0.72 & 0.04 \\ 0.04 & 0.74 \end{pmatrix}$ C1 $\begin{pmatrix} 0.22 & -0.03 \\ -0.03 & 0.91 \end{pmatrix}$

C2 $\begin{pmatrix} 0.04 & 0.00 \\ 0.00 & 0.04 \end{pmatrix}$

Réattribuer chaque matrice à chaque classe



Sachant qu'il y a 100 points dans chaque classe, quelle est la probabilité des classes ?
Retrouver $\boldsymbol{\Sigma_{intra}}$

Elles ont toutes la même probabilité : $p_1 = p_2 = p_3 = 1/3$

Et donc $\boldsymbol{\Sigma_{intra}} = \frac{1}{3}\begin{pmatrix} 0.22 & -0.03 \\ -0.03 & 0.91 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 0.04 & 0.00 \\ 0.00 & 0.04 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 0.72 & 0.04 \\ 0.04 & 0.74 \end{pmatrix}$

$$\boldsymbol{\Sigma_{intra}} = \begin{pmatrix} 0.33 & 0.00 \\ 0.00 & 0.56 \end{pmatrix}$$

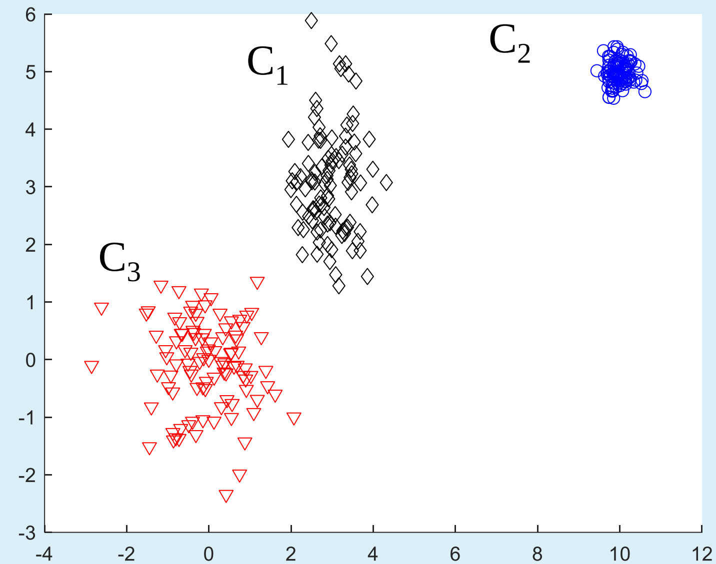
Analyse Discriminante Linéaire (LDA)

Exercice

On donne les moyennes des trois classes :

$$\mu_1 = \begin{pmatrix} -0.0542 \\ -0.0804 \end{pmatrix}, \mu_2 = \begin{pmatrix} 2.9606 \\ 3.0945 \end{pmatrix}, \mu_3 = \begin{pmatrix} 9.9930 \\ 4.9809 \end{pmatrix},$$

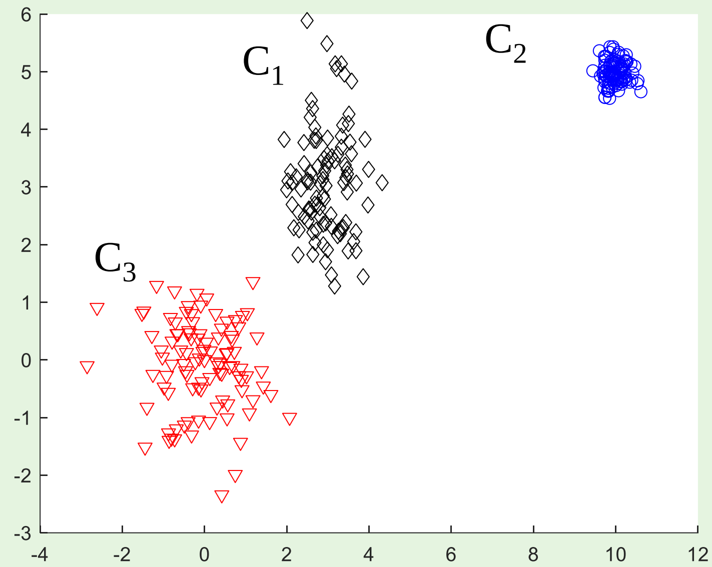
Que vaut $\boldsymbol{\Sigma_{inter}}$?



Exercice

On donne les vecteurs moyenne des
Trois classes :

$\mu_1 = \begin{pmatrix} -0.0542 \\ -0.0804 \end{pmatrix}, \mu_2 = \begin{pmatrix} 2.9606 \\ 3.0945 \end{pmatrix}, \mu_3 = \begin{pmatrix} 9.9930 \\ 4.9809 \end{pmatrix},$



Qua vaut $\boldsymbol{\Sigma_{inter}}$?
Il faut d'abord calculer la moyenne de tous les points :

$$\mu = p_1 * \mu_1 + p_2 * \mu_2 + p_2 * \mu_2 = \begin{pmatrix} 4.2998 \\ 2.6650 \end{pmatrix}$$

On a ensuite

$$\Sigma_{\boldsymbol{inter}} = p_1(\mu_{\mathbf{1}} - \mu)(\mu_{\mathbf{1}} - \mu)^T + p_2(\mu_{\mathbf{2}} - \mu)(\mu_{\mathbf{2}} - \mu)^T + p_3(\mu_{\mathbf{3}} - \mu)(\mu_{\mathbf{3}} - \mu)^T$$

$$\Sigma_{\boldsymbol{inter}} = \begin{pmatrix} 17{,}72 & 8{,}19 \\ 8{,}19 & 4{,}36 \end{pmatrix}$$

Analyse Discriminante Linéaire (LDA)

Exercice

Représenter sur la figure le premier axe obtenu par LDA sachant que les 2 vecteurs/valeurs propres de $\boldsymbol{\Sigma}_{intra}^{-1}\boldsymbol{\Sigma}_{inter}$ sont:

Valeurs propres :

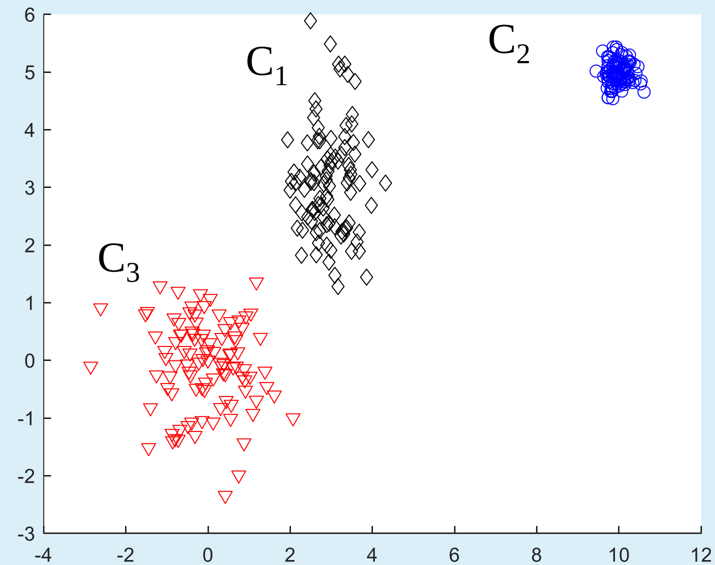
λ_1= 93.58 et λ_2-5.53

Vecteurs propres :

$v_1 = \begin{bmatrix} 0.96 \\ 0.28 \end{bmatrix}$ et $v_2 = \begin{bmatrix} -0.72 \\ 0.70 \end{bmatrix}$

Quel sera le premier axe de projection ?

Quelles sera la coordonnée du point $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ sur cet axe ?



Analyse Discriminante Linéaire (LDA)

Exercice

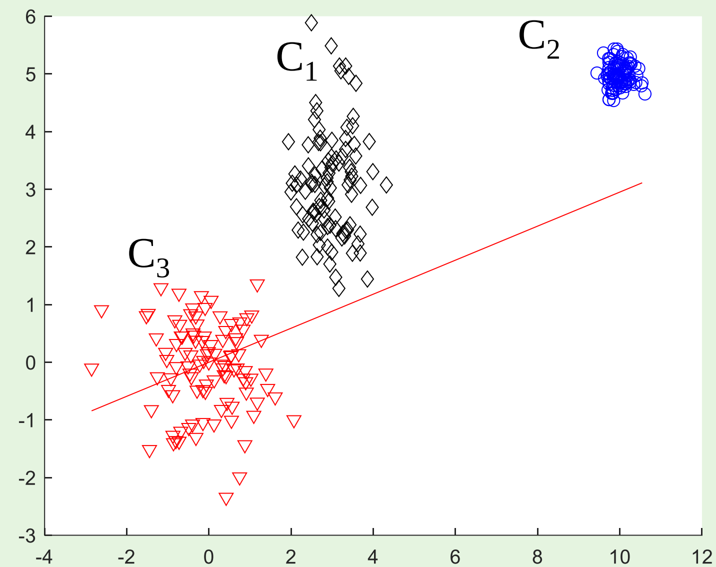
Représenter sur la figure le premier axe obtenu par LDA sachant que les 2 vecteurs/valeurs propre de $\boldsymbol{\Sigma}_{intra}^{-1}\boldsymbol{\Sigma}_{inter}$ sont:

Valeurs propres :
λ_1= 93.58 et λ_2-5.53

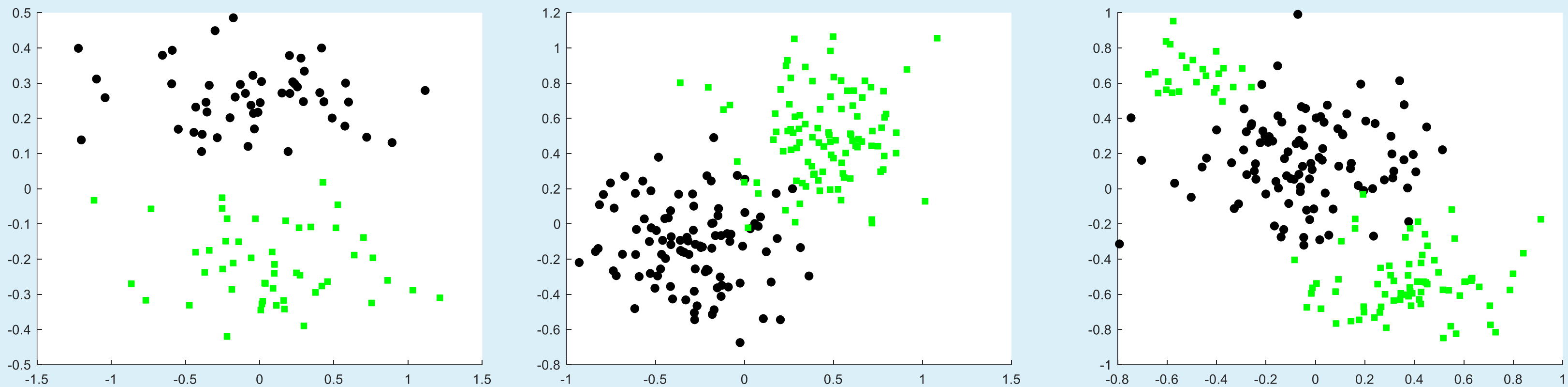
Vecteurs propres :
$v_1 = \begin{bmatrix} 0.96 \\ 0.28 \end{bmatrix}$ et $v_2 = \begin{bmatrix} -0.72 \\ 0.70 \end{bmatrix}$

Quelles sera la coordonnée du point $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ sur cet axe ?

$[4 \quad 3]\begin{bmatrix} 0.96 \\ 0.28 \end{bmatrix}$ =4.68

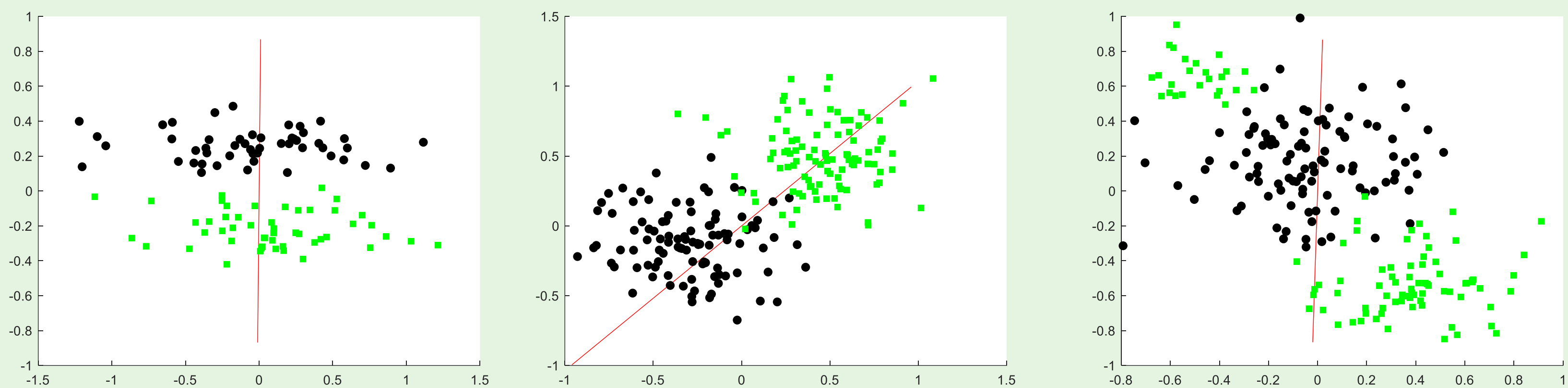


Reprenons les 3 exemples précédents



Donner le premier axe principal obtenu par LDA

Analyse Discriminante Linéaire (LDA)



Bien Bien Moyen

Sélection de caractéristiques

Les méthodes précédentes permettent de réduire la dimension des données en les projetant dans un nouvel espace dont on gardera seulement les premières dimensions

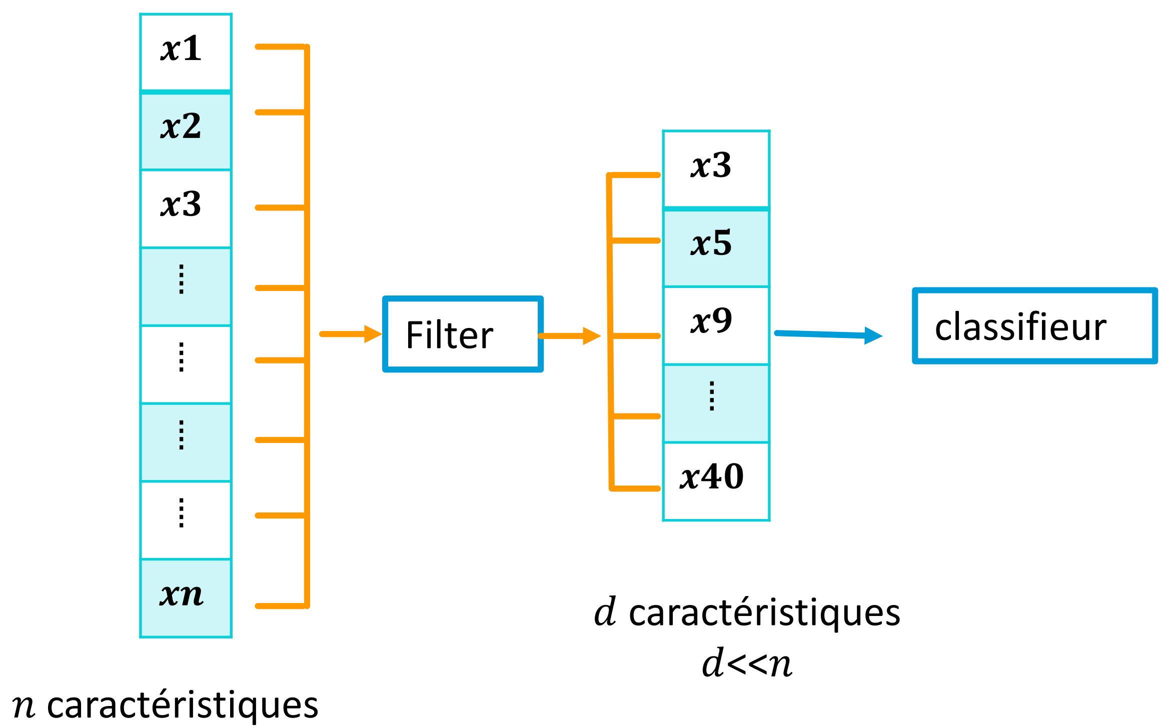
Inconvénient
Comme les nouvelles dimensions sont des combinaisons linéaires des anciennes, il faut quand même extraire toutes les dimensions de l'espace initial

Autre idée : la sélection de caractéristiques
On choisit certaines des dimensions de l'espace initial.
Comment ?

- Filter
- Wrapper

Sélection de caractéristiques

Les méthodes filter travaillent en amont de la classification : on étudie les *n* dimensions (ou caractéristiques) et on en sélectionne *d* en fonction d'un critère donné.
Par exemple, on garde les caractéristiques qui ont la plus forte corrélation possible avec les étiquettes.



Sélection de caractéristiques

Avantage des méthodes de type filter ?

Inconvénient des méthodes de type filter ?

Avantage des méthodes de type filter

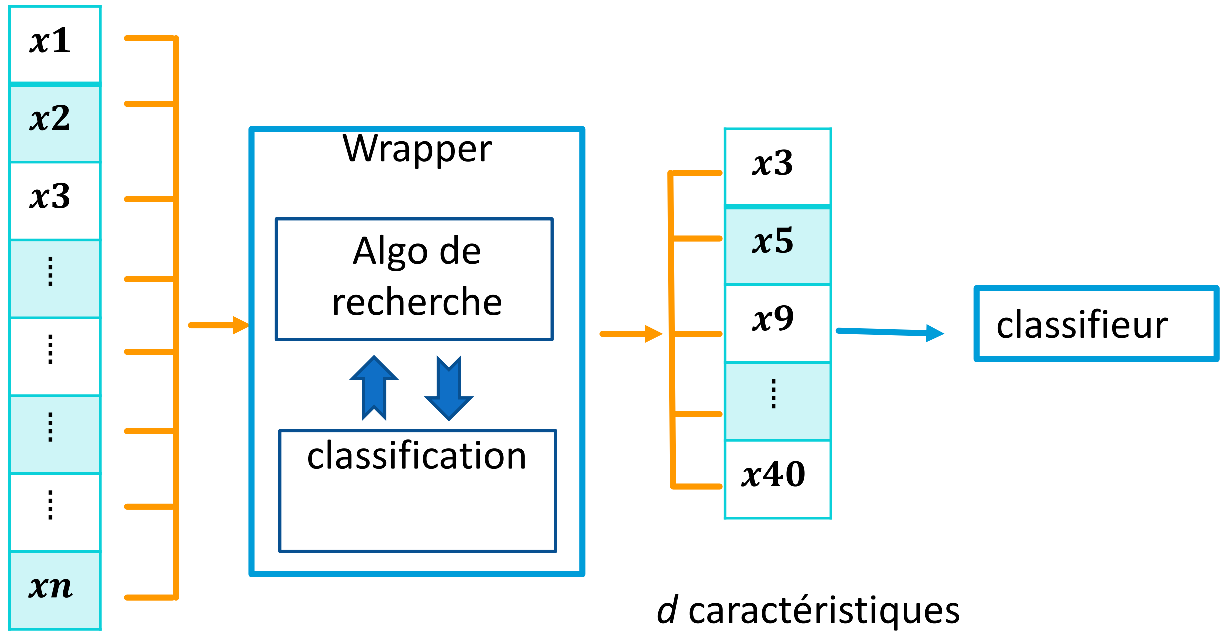
- efficacité calculatoire

Inconvénient des méthodes de type filter

- ne tiennent pas compte des interactions entre caractéristiques et tendent à sélectionner des caractéristiques redondantes plutôt que complémentaires.
- ne tiennent pas compte de la performance des méthodes de classification

Sélection de caractéristiques

Les méthodes wrapper évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage



n caractéristiques

d caractéristiques
d<<*n*

Sélection de caractéristiques

Exemple :
on commence par un ensemble vide de caractéristiques.
A chaque itération, la meilleure caractéristique parmi celles qui restent est sélectionnée

Avantage des méthodes de type wrapper ?

Inconvénient des méthodes de type wrapper ?

Avantage des méthodes de type wrapper
Capable de sélectionner des sous-ensembles de caractéristiques de petite taille qui sont performants pour le classificateur

Inconvénient des méthodes de type wrapper

- La complexité de l'algorithme d'apprentissage rend les méthodes "wrapper" très coûteuses en temps de calcul → stratégie de recherche exhaustive impossible

- Très longues en temps de calcul car beaucoup d'apprentissages nécessaires pour sélectionner le bon sous-ensemble de caractéristiques

- Sous-ensemble dépendant du classificateur choisi