Cours 1
Introduction
Codage
Réduction de dimension

Catherine ACHARD
Institut des Systèmes Intelligents et de Robotique

catherine.achard@upmc.fr

Cc1 26 février
Cc2 22 avril
Examen de TP 22 avril

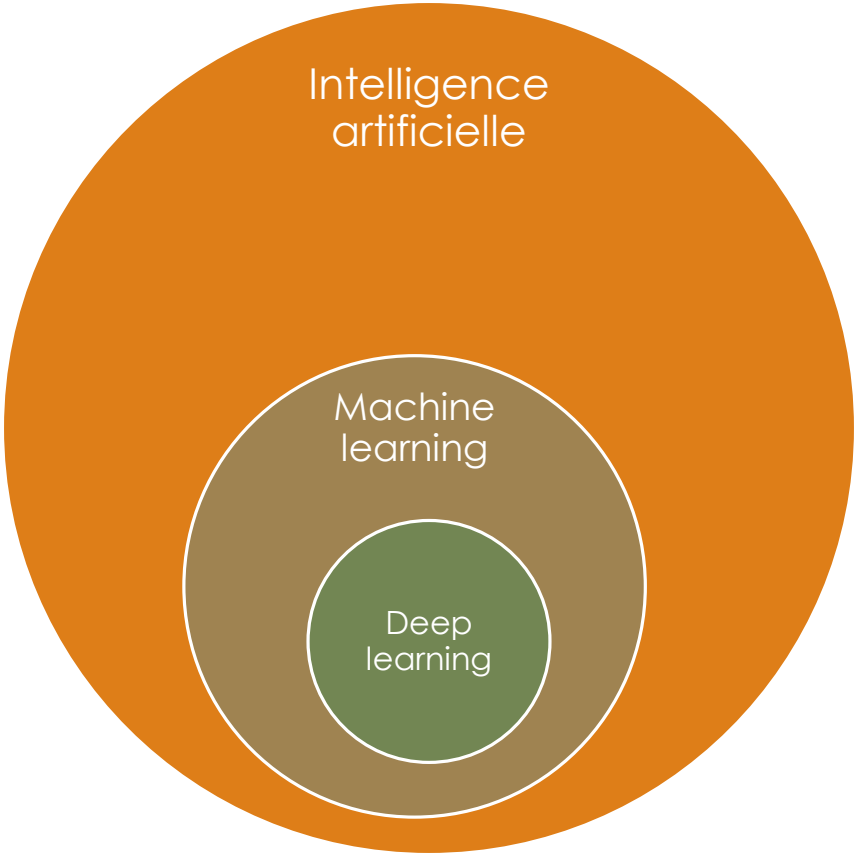
Références

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

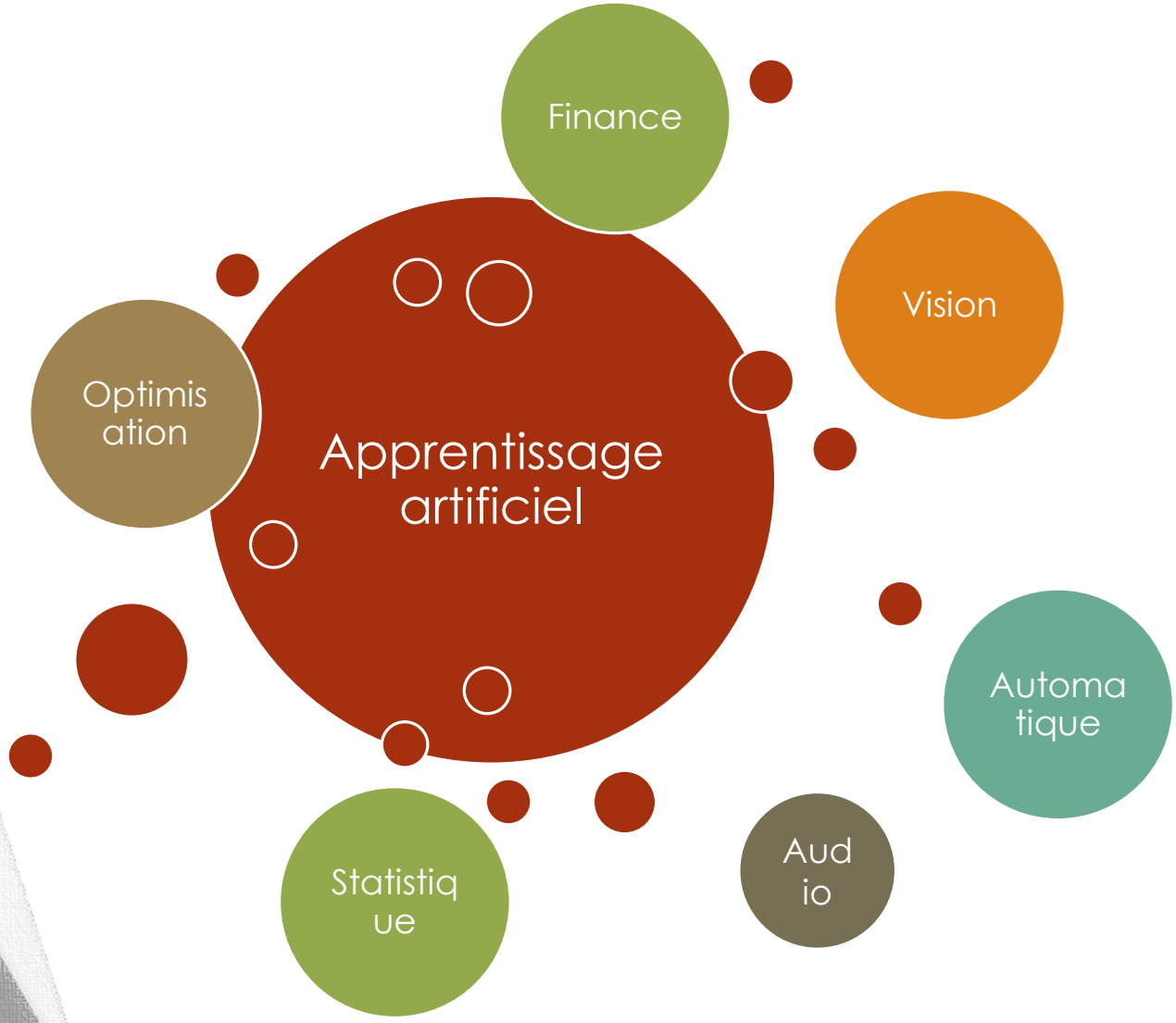
- CHRISTOPHER M. BISHOP, "**Pattern Recognition and machine learning**", springer, 2006

Introduction

Introduction



Introduction



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

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

Les difficultés sur un cas simple

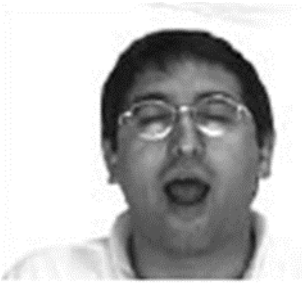
Considérons le cas de la reconnaissance de visages (classification)

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.

Importance du codage
Exemple sur un cas simple

Reconnaître les truites et les saumons

Codage

Chaque poisson est représenté par

- un vecteur de caractéristiques
- Un codage
- Des features

Ici

- La taille
- La couleur

Importance du codage
Exemple sur un cas simple

Codage

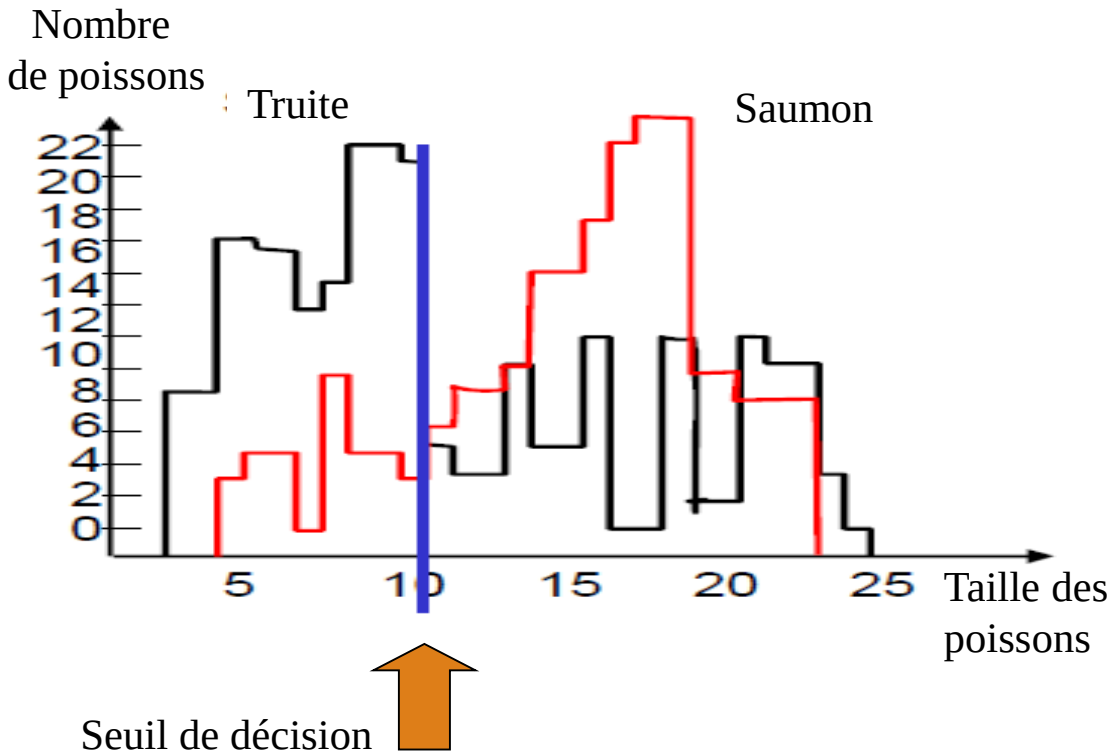
- Extraire de la forme un **vecteur de mesure** aussi appelé **vecteur de caractéristiques** ou **codage** ou **features**

Exemple en 1 dimension

- la taille des poissons

Importance du codage
Exemple sur un cas simple

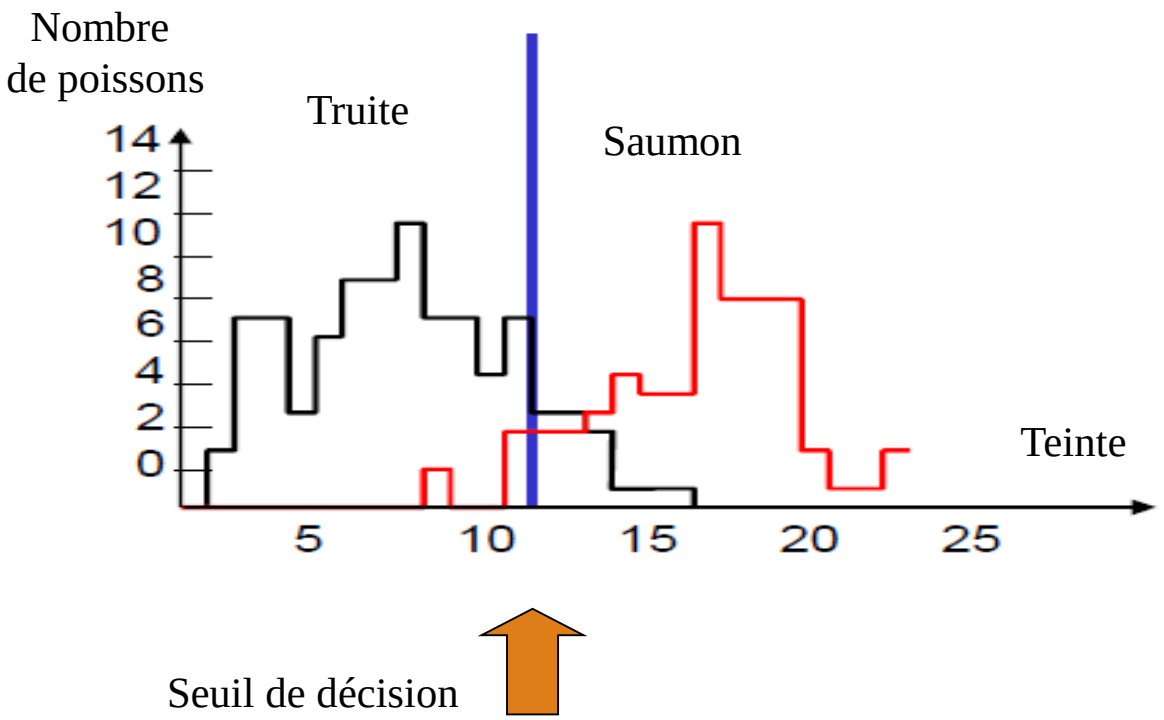
Classification avec la taille des poissons



Trop de chevauchement, **décision pas robuste**. Que faire ?

Importance du codage
Exemple sur un cas simple

Classification avec la teinte des poissons

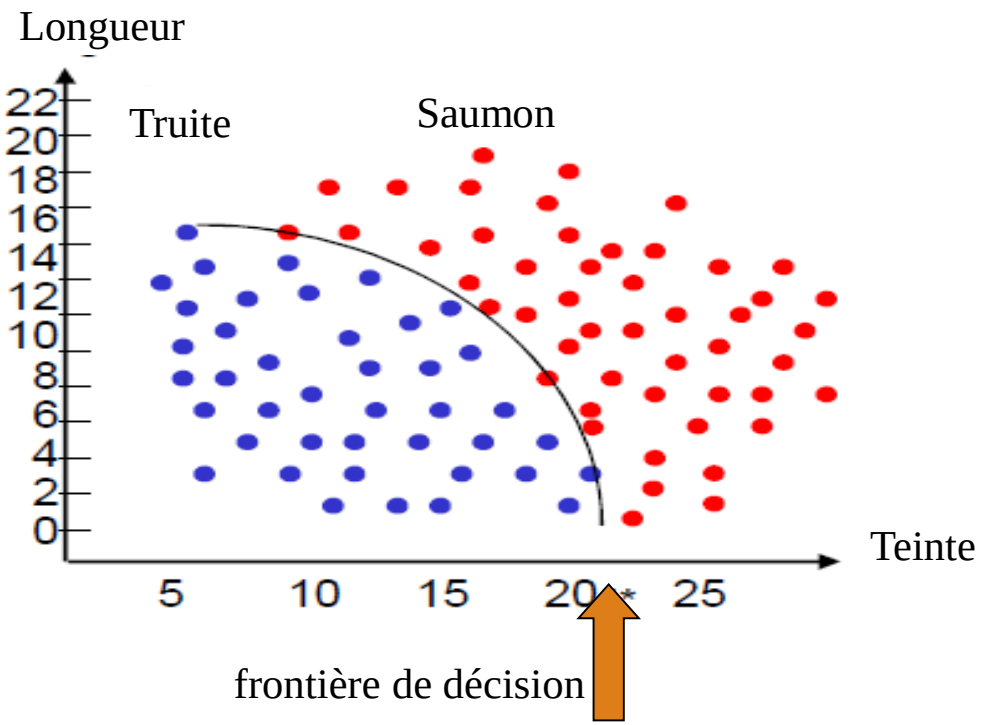


Trop de chevauchement, **décision pas robuste**. Que faire ?

Importance du codage
Exemple sur un cas simple

Classification avec la teinte et la longueur des poissons

➔ Vecteur de caractéristiques à 2 dimensions



On peut ajouter d'autres caractéristiques pour améliorer la classification

Mais les données ne sont pas toujours idéales

Exemple de la détection

Comment détecter un visage dans une image par apprentissage ?

Exemple de la détection

On change le problème de détection en un problème de classification

On présente une imagette (une zone de l'image) en entrée d'un système de reconnaissance. Celui-ci nous dit si cette imagette est un visage ou non (sortie binaire)

Problème

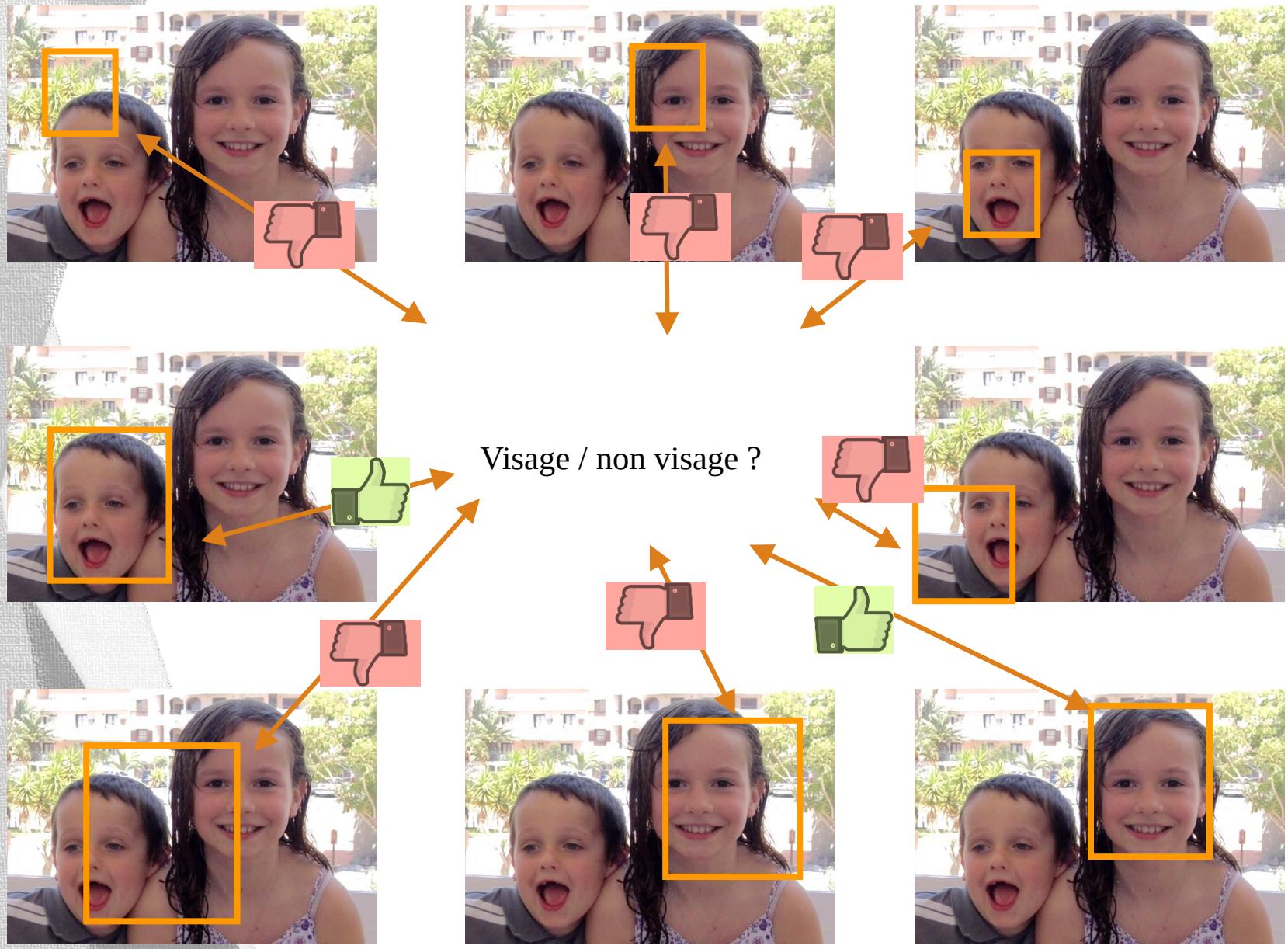
- Comment savoir où rechercher dans l'image ?
- Comment détecter à plusieurs échelles ?

Solution

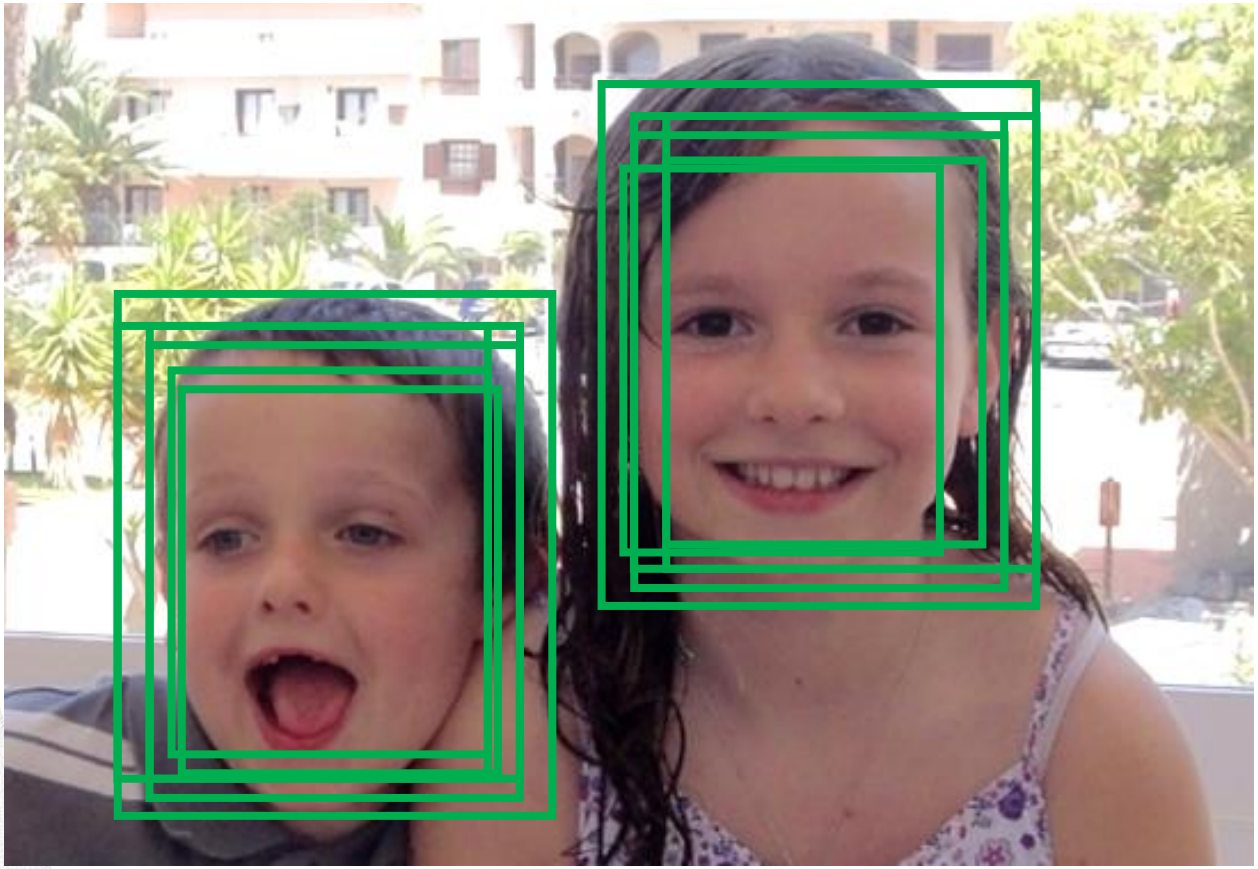
En testant toutes les combinaisons possibles

En chaque position, problème de reconnaissance. Est-ce un visage ou non ?

Exemple de la détection



Exemple de la détection



Matrice de covariance

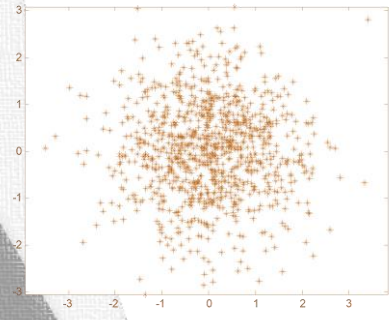
Rappel sur les matrices de covariance

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

$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \mu)(\boldsymbol{x}_i - \mu)^T$ où μ est le vecteur moyen des données $\boldsymbol{x}_i$

La matrice de covariance est de dimension *n* x *n* et est symétrique.

Exemple en dimension 2
On peut représenter chaque point Xi dans le plan.



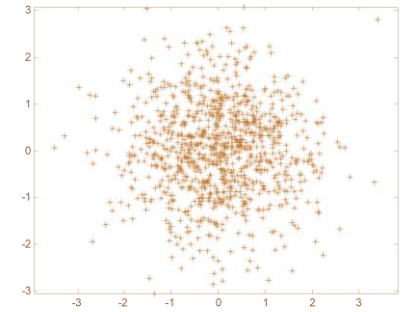
La matrice de covariance est de dimension 2x2

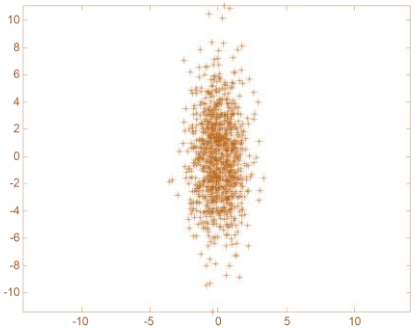
$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix}$$

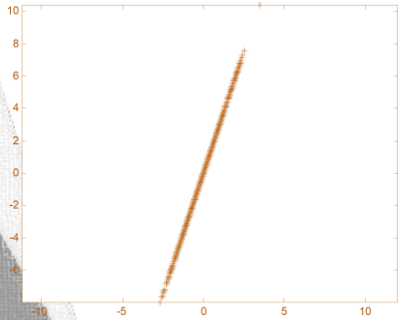
La matrice de covariance est symétrique. Ses valeurs codent la **forme du nuage de point**.

Matrice de covariance

Rappel sur les matrices de covariance







Exercice
Attribuer chaque ensemble de points à sa matrice de covariance

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

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

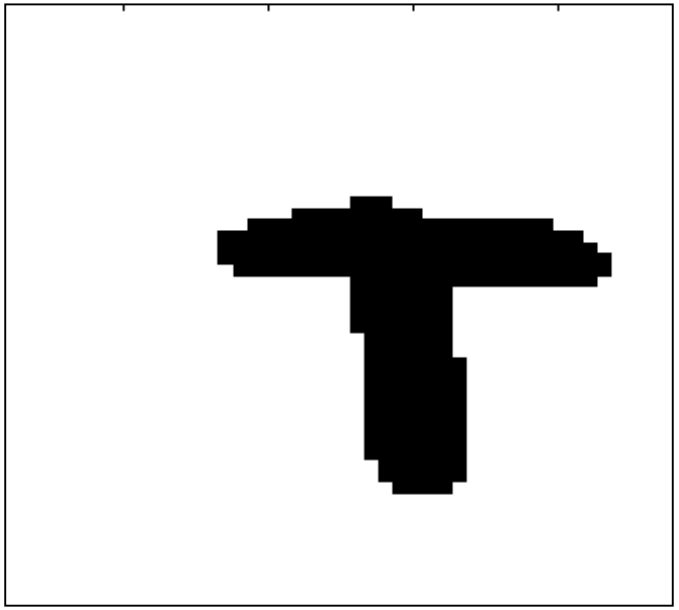
- Codage le plus insensible possible au bruit
- En fonction des applications, invariance en translation, rotation, changement d'échelle

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)

Exemple de codage : le codage rétinien

On garde toute l'information directement dans une rétine :



Problème

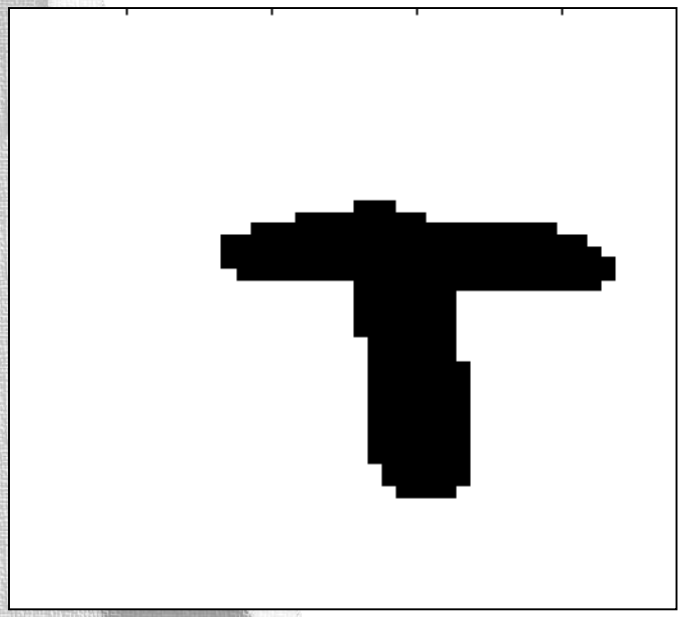
-La lettre n'est pas toujours à la même position

-La résolution de l'image n'est pas toujours la même

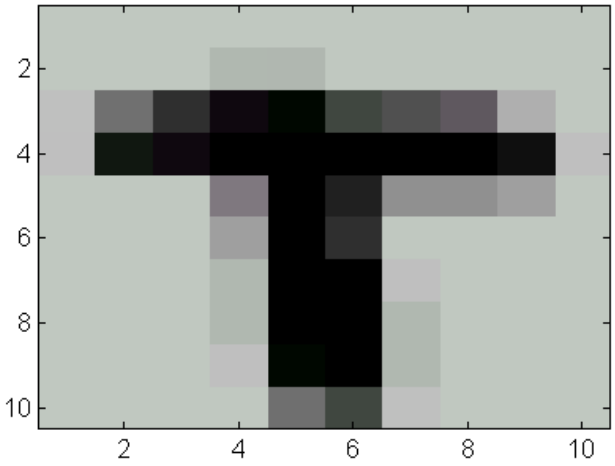
Exemple de codage : le codage rétinien

Codage

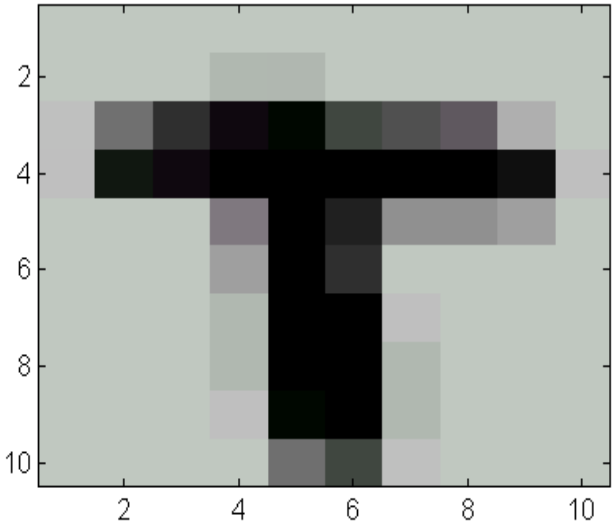
- Calcul du centre de gravité
- Sélection du plus petit carré centré en G et englobant tous les pixels
- Réduction de la dimension (attention, pas binaire !!)



Rétine 10x10
après centrage et
réduction
➔
Vecteur de caractéristiques de dimension 100



Exemple de codage : le codage rétinien



Codage quasiment neutre

- Pas de perte d'information
- Laisse le classifieur travailler
- Efficace si base d'exemples importante

Ne tolère ni les transformations ni les déformations

Comment trouver un bon codage ?

Méthodes empiriques

- Choix des caractéristiques pertinentes

Méthodes exploratoires

- Algorithmes génétiques

Méthodes par apprentissage

- Boosting, forêt d'arbre aléatoire,…

Méthodes statistiques pour réduire la dimension ou la taille des données

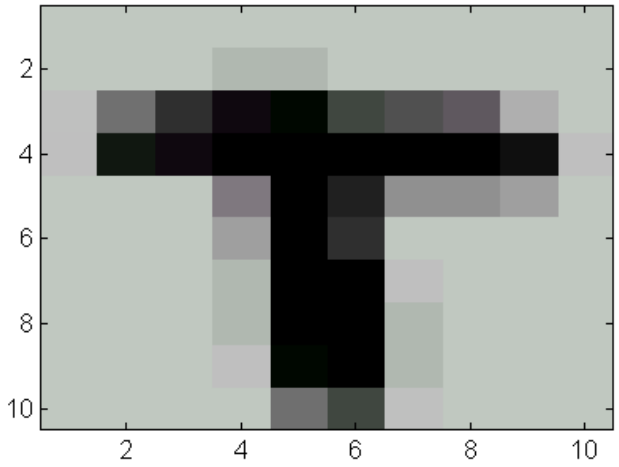
- Analyse en composantes principales
- Analyse discriminante linéaire
- Sélection/extraction de caractéristiques

Un bon codage aura une faible dimension

- Fléau des grandes dimensions

Fléau des grandes dimensions

La dimension des données *n* est la dimension du code les représentant



Rétine 10 X 10:

(après centrage et réduction)

$n = 100$

Premier problème

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

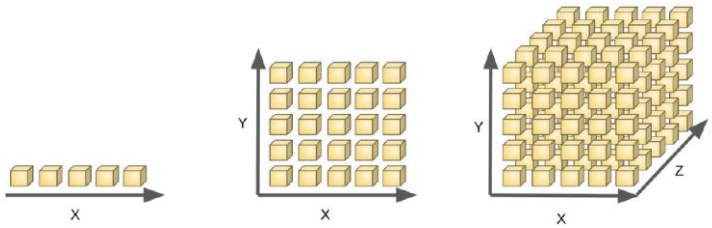
Second problème

- 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

Fléau des grandes dimensions

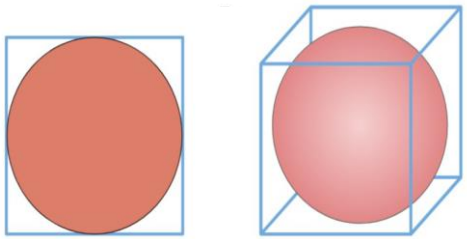
Troisième problème : la folie géométrique

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

Dimension 2

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

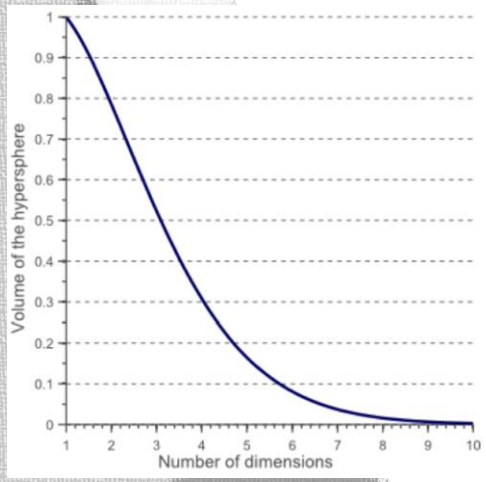
Le rapport de surface est $\frac{\pi 0.5^2}{1} = 0.78$ ➔ **22% des points sont dans les coins**



Dimension 3

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

Et en continuant



Donc,

- la plupart des données vont être dans les coins, à égale distance du centre
- L'espace est très peu occupé

➔ Les mesures de distances deviennent de moins en mois représentatives

➔ Il faut essayer de diminuer la dimension de l'espace de représentation

Analyse en composantes principales (ACP)
Principal component analysis (PCA)

- Projection des données depuis l'espace à *n* dimensions vers un espace à *d* dimensions ($d < n$)
- Il faut qu'un maximum d'information soit conservée
- L'information est mesurée par la variance du nuage de points

➔ Normalisation préalable des données

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

	Mesure 1	Mesure 2	Mesure 3	Mesure 4	Mesure 5
Donnée 1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
Donnée 2	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}
Donnée 3	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}
Donnée 4	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}

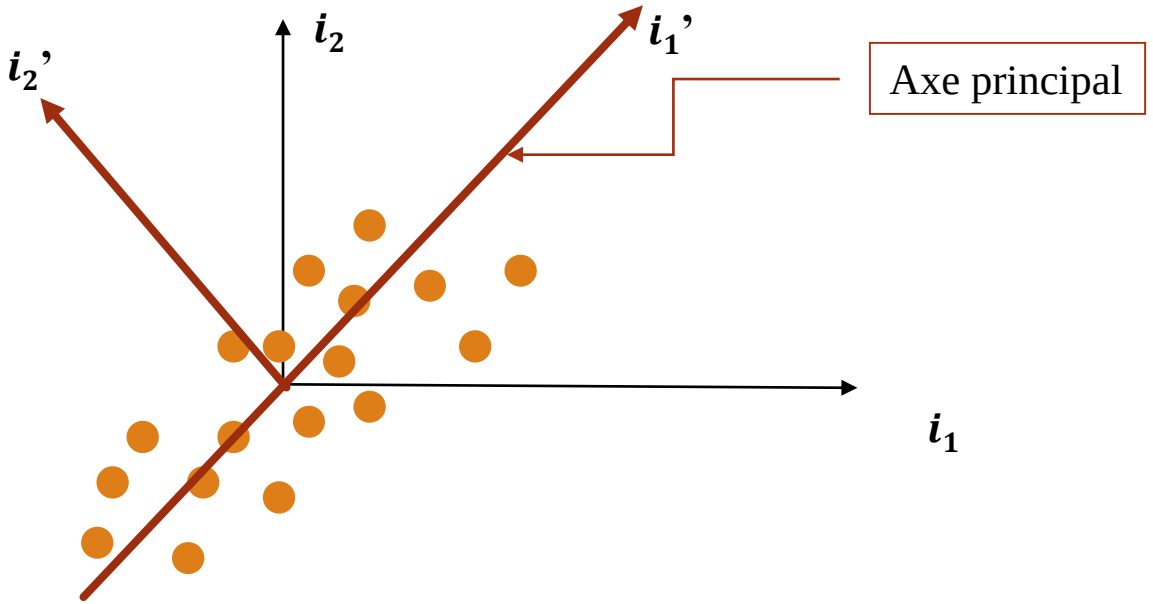
Le tableau contient des informations de natures différentes. Il faut

- **centrer les données** : on soustrait la moyenne de la colonne à chaque valeur.
- **normaliser les données** : on divise chaque valeur par l'écart type de sa colonne.

Analyse en composantes principales

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 deux (x_{i1},x_{i2}) 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.



Analyse en composantes principales

De manière plus formelle, chaque point $\boldsymbol{x_i}$ s'exprime dans le repère initial par:

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

La projection des données sur l'axe $\boldsymbol{i_1}'$ donne la nouvelle coordonnée des points:

$$x_{i1}' = \boldsymbol{x_i}^T \boldsymbol{i_1}' \quad \text{et} \quad \boldsymbol{i_1}'^T \boldsymbol{i_1} = 1$$

Et la variance des données sur cet axe sera:

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

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

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

Analyse en composantes principales

Problème de maximisation sous contrainte que l'on résout à partir du 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

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

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

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

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

Exercice : déterminer $\boldsymbol{i_1}'$

Analyse en composantes principales

Problème de maximisation sous contrainte que l'on résout à partir du 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

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

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

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

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

Exercice : déterminer $\boldsymbol{i_1}$

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

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

Equation des valeurs et vecteurs propres de la matrice de covariance $\boldsymbol{\Sigma}$.

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

Analyse en composantes principales

Problème de maximisation sous contrainte que l'on résout à partir du 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

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

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

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

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

Exercice : déterminer $\boldsymbol{i_1}$

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

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

➔ $\boldsymbol{i_1}'$ est une 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 λ

Analyse en composantes principales

➔ **Le premier axe de projection est le premier vecteur propre de la matrice de covariance Σ**

Le second axe correspondra au second vecteur propre de la matrice,…

Rmq:
La matrice de covariance est par construction symétrique et au moins positive semi-définie. Ceci implique que les valeurs propres et les vecteurs propres seront réels, que les valeurs propres seront positives ou nulles et que les vecteurs propres seront orthogonaux entre eux.

Analyse en composantes principales

Chaque donnée $\boldsymbol{x}_i$ peut s'exprimer dans la base des vecteurs propres:

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

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

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

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

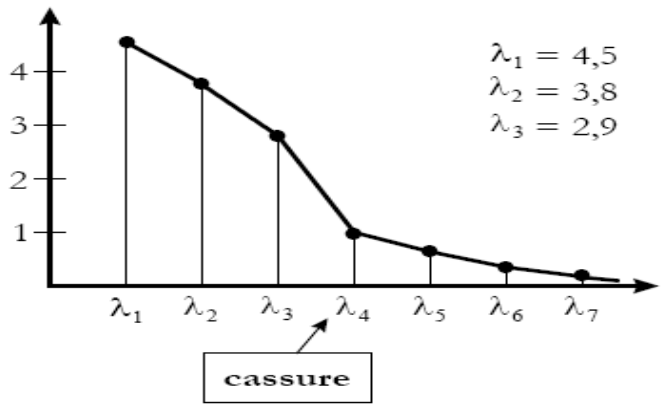
On définit **l'inertie** portée par les d premiers axes par

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

Analyse en composantes principales

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. Par validation sur une base de validation

Analyse en composantes principales

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 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 composantes principales

Exemples d'ACP sur des images de visages

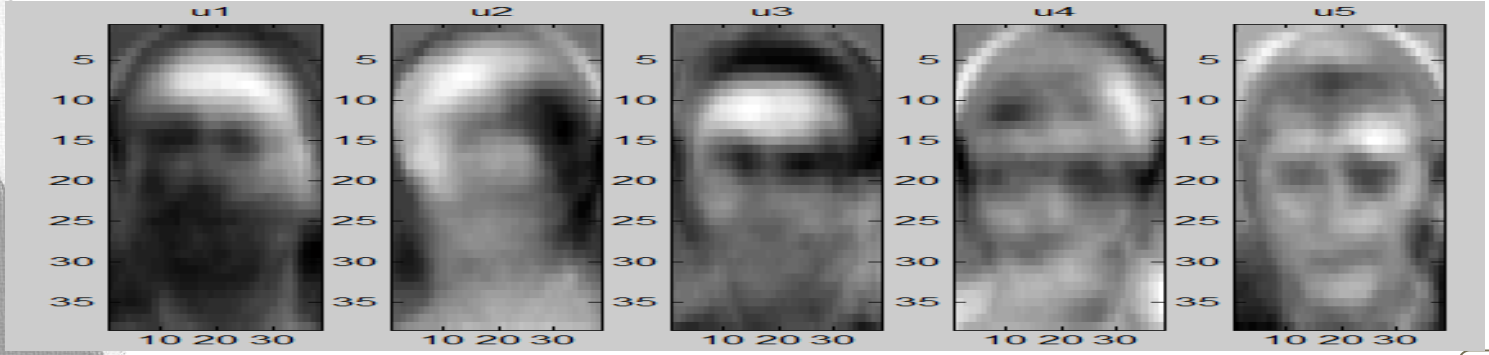
On dispose d'une base de références de 270 visages,
Chaque visage a pour dimension 38x38=1444 pixels ➔ n=1444

On range tous ces visages dans une matrice de dimension 270x1444.

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) :



Analyse en composantes principales

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

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

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 composantes principales

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

$$\boldsymbol{x_i} = x_{i1}' \boldsymbol{i_1}' + x_{i2}' \boldsymbol{i_2}' + \ldots + xi_5' \boldsymbol{i_5}'$$

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 : $\boldsymbol{x_i} = (x_{i1} \; x_{i2} \; \ldots \; xi_5)^T$

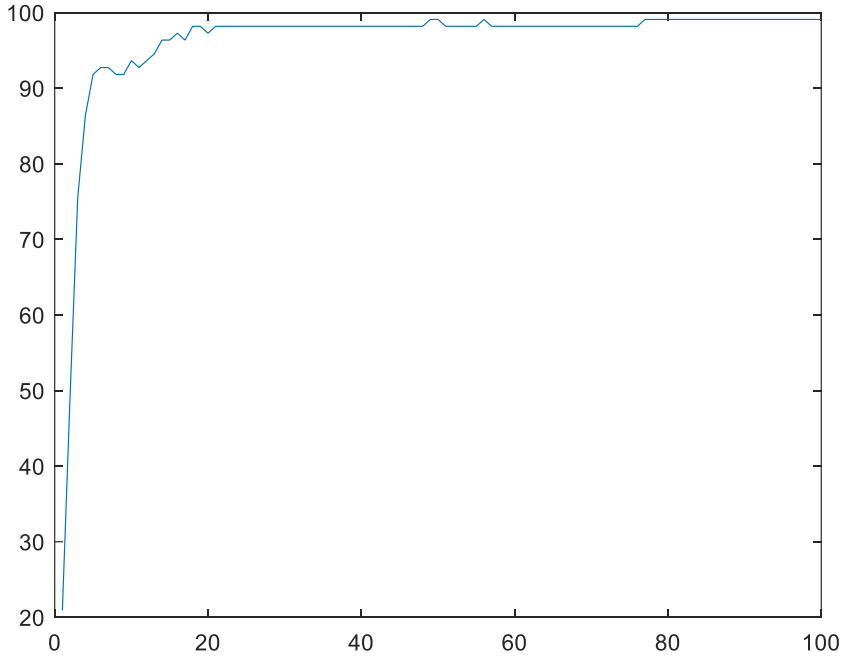
- Pour le reconstruire, à la décompression, on utilise les eigen image $\boldsymbol{x_i} = x_{i1}' \boldsymbol{i_1}' + x_{i2}' \boldsymbol{i_2}' + \ldots + xi_5' \boldsymbol{i_5}'$
- Pour le reconnaitre, on utilise le vecteur $\boldsymbol{x_i} = (x_{i1} \; x_{i2} \; \ldots \; xi_5)^T$ comme codage du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

Analyse en composantes principales

Exemple sur une base de visages

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

Taux de reconnaissance



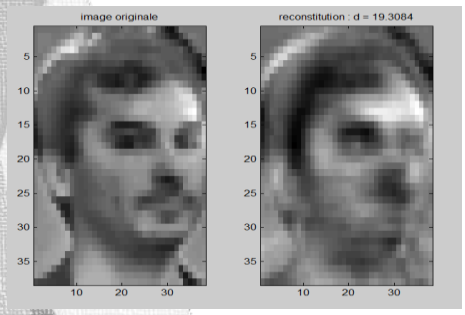
Nombre de dimension gardé

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

Analyse en composantes principales

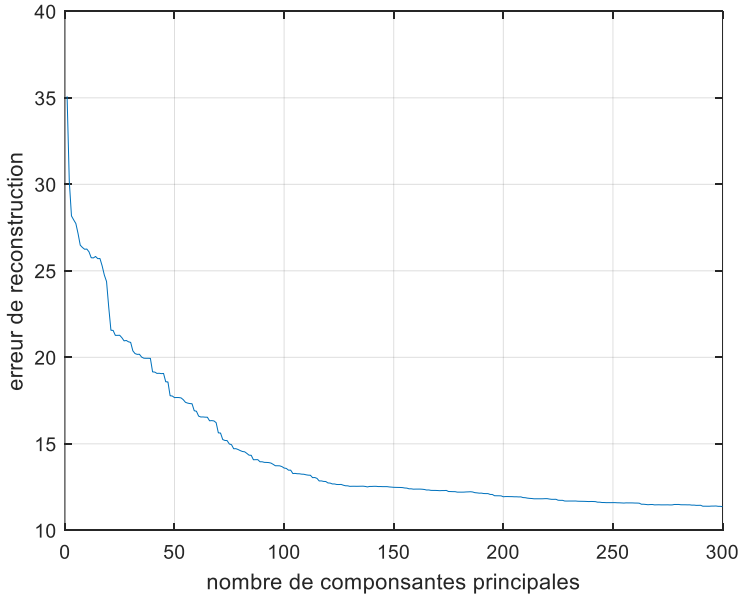
Sur la même base, en reconstruction

Reconstruction avec
5 « eigen images »
Compression = 288%



Reconstruction avec
49 « eigen images »
Compression = 32%

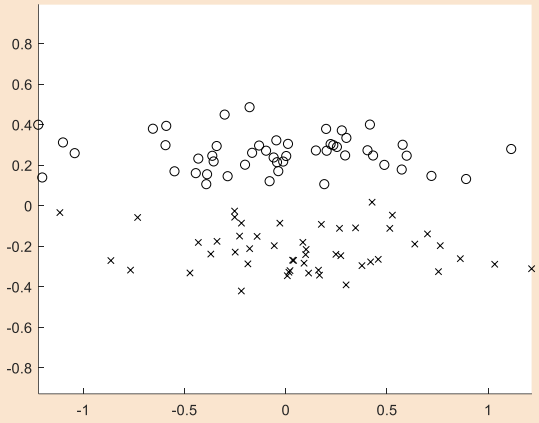
Reconstruction avec
144 « eigen images »
Compression = 10%

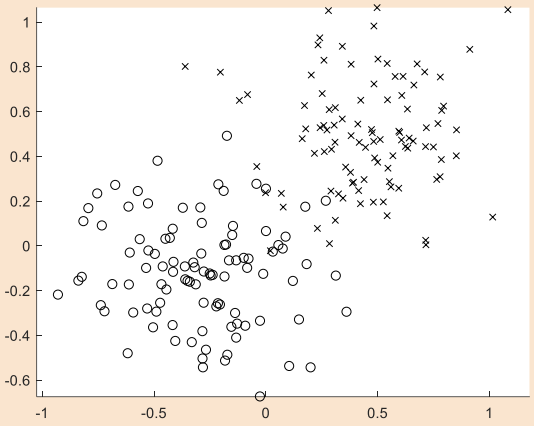


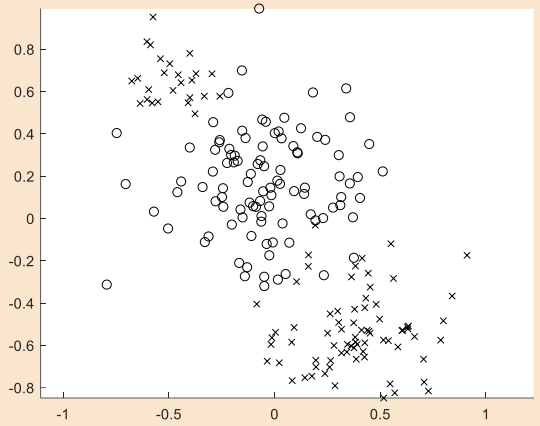
Analyse en composantes principales

Exercice

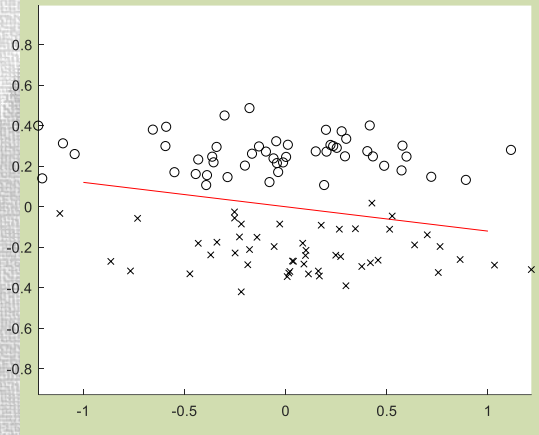
Pour ces trois ensembles de données, tracer l'axe de projection obtenu par ACP.
Conclusion



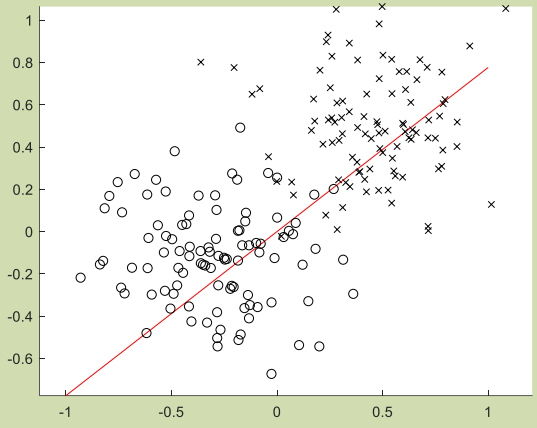




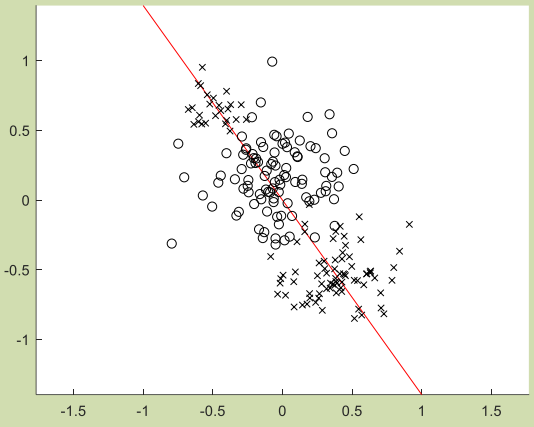
Analyse en composantes principales



Très mauvais



Bien

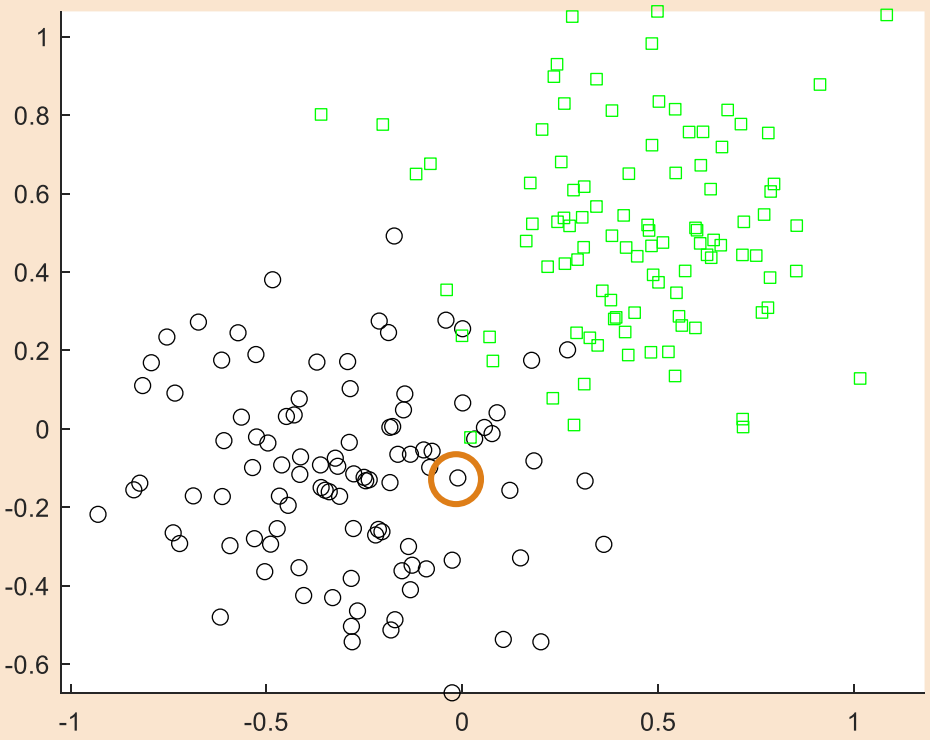


Bien

Analyse en composantes principales

Exercice

On considère le nuage de point 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 :

0.0659 et 0.2969

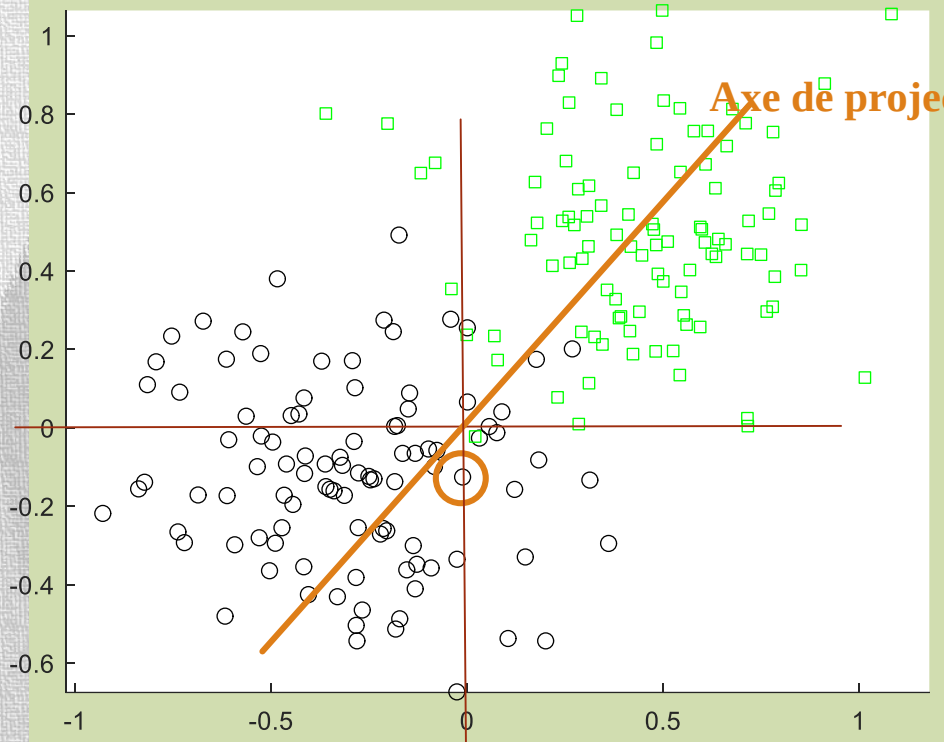
Vecteurs propres :

0.6132 -0.7900
-0.7900 -0.6132

Quel sera le vecteur de projection ?
Tracer le
Quelle sera la nouvelle coordonnée du point

$$x = \begin{pmatrix} -0.15 \\ 0 \end{pmatrix}$$

Analyse en composantes principales



Matrice de covariance de tout le nuage :

0.2100 0.1119

0.1119 0.1528

Valeurs propres :

0.0659 et 0.2969

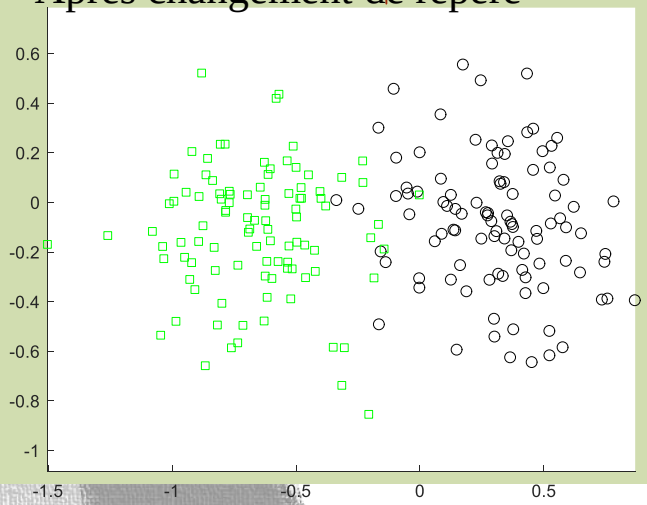
Vecteurs propres :

0.6132 -0.7900

-0.7900 -0.6132

Plus grande valeur propre

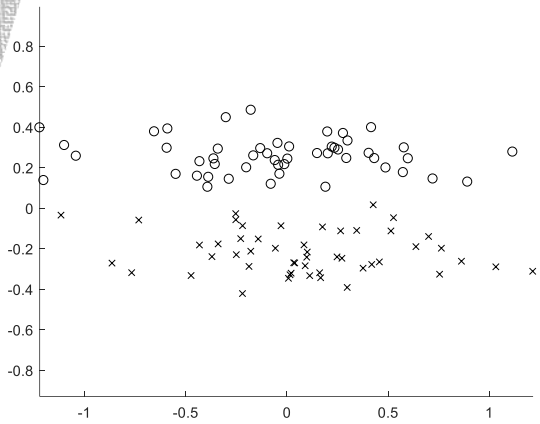
Après changement de repère

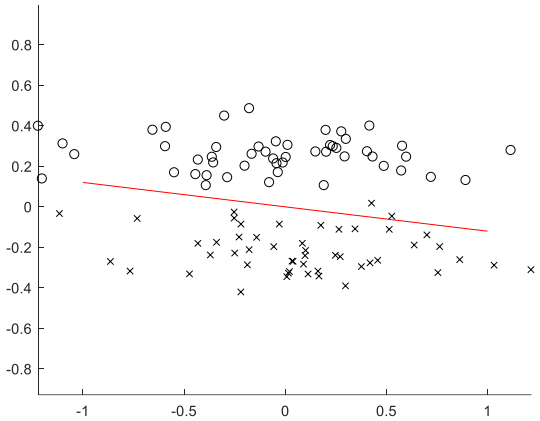


$$x = \begin{pmatrix} -0.15 \\ 0 \end{pmatrix}$$

$$x_{i1}' = (-\mathbf{0.15} \quad \mathbf{0}) \begin{pmatrix} -0.7900 \\ -0.6132 \end{pmatrix} = 0{,}12$$

Analyse en composantes principales





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

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

Analyse discriminante linéaire
Linear Discriminant Analysis (LDA)

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.**

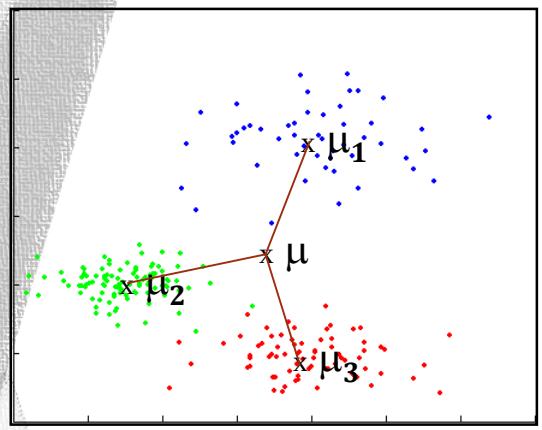
Problème

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*:

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

Analyse discriminante linéaire



On définit la moyenne de chaque classe et la moyenne globale

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

Où $p_1, p_2, \dots p_K$ sont les probabilités de chaque classe

On définit également la matrice de covariance de chaque classe

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

Dispersion intra classe

$$\Sigma_{intra} = p_1 * \Sigma_1 + p_2 * \Sigma_{2 + \cdots +} + p_K * \Sigma_K$$

Dispersion inter classe

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

Analyse discriminante linéaire

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

La projection des points $\boldsymbol{x_i}$ selon le vecteur $\boldsymbol{i_1}'$ s'exprime par :

$$x'_{i1} = \boldsymbol{x_i}^T \boldsymbol{i_1}' \quad \text{et} \quad \boldsymbol{i_1}'^T \boldsymbol{i_1}' = 1$$

Si la matrice de covariance des données de départ est

$$\Sigma = E\{\boldsymbol{x_i}\boldsymbol{x_i}^T\}$$

dans le nouveau repère, on a

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

$$\sigma = \frac{1}{N}\boldsymbol{i_1}'^T \Sigma \boldsymbol{i_1}'$$

$\boldsymbol{i_1}'$ doit maximiser le rapport variance inter /variance intra

$$\frac{\sigma_{inter}}{\sigma_{intra}} = \frac{\boldsymbol{i_1}'^T \boldsymbol{\Sigma_{inter}} \boldsymbol{i_1}'}{\boldsymbol{i_1}'^T \boldsymbol{\Sigma_{intra}} \boldsymbol{i_1}'}$$

Ce qui revient à maximiser $\boldsymbol{i_1}'^T \boldsymbol{\Sigma_{inter}} \boldsymbol{i_1}'$ sous la contrainte $\boldsymbol{i_1}'^T \boldsymbol{\Sigma_{intra}} \boldsymbol{i_1}' = 1$ (car peu importe la norme de $\boldsymbol{i_1}'$)

Analyse discriminante linéaire

Lagrangien dont on annule la dérivée:

$$L = \boldsymbol{i_1}'^T \boldsymbol{\Sigma_{inter}} \boldsymbol{i_1}' - \lambda(\boldsymbol{i_1}'^T \boldsymbol{\Sigma_{intra}} \boldsymbol{i_1}' - 1)$$

$\frac{\partial L}{\partial \boldsymbol{i_1}'} = 0$ ➔ $\boldsymbol{\Sigma_{inter}} \boldsymbol{i_1}' = \lambda \boldsymbol{\Sigma_{intra}} \boldsymbol{i_1}'$ ➔ $\boldsymbol{\Sigma_{intra}^{-1}} \boldsymbol{\Sigma_{inter}} \boldsymbol{i_1}' = \lambda \boldsymbol{i_1}'$

➔ Problème aux valeurs/vecteurs propres

➔ $\boldsymbol{i_1}'$ est le premier vecteur propre de $\boldsymbol{\Sigma_{intra}^{-1}} \boldsymbol{\Sigma_{inter}}$

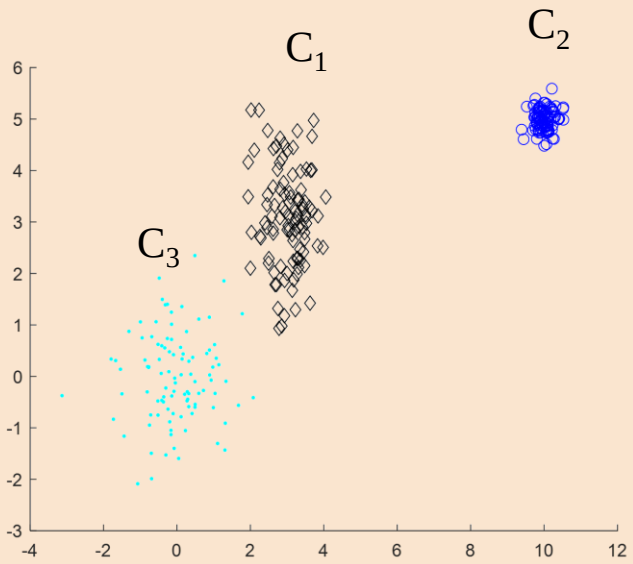
Les autres axes de projection correspondent aux autres vecteurs propres de $\boldsymbol{\Sigma_{intra}^{-1}} \boldsymbol{\Sigma_{inter}}$

De la même manière que l'ACP, on garde les axes correspondant aux *d* premiers vecteurs propres pour réduire la dimension

Analyse discriminante linéaire

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

Les deux matrices $\boldsymbol{\Sigma_{intra}}$ et $\boldsymbol{\Sigma_{inter}}$ sont données par :

$$\begin{pmatrix} 23.04 & 25.76 \\ 12.36 & 9.64 \end{pmatrix} \begin{pmatrix} 0.33 & 0.00 \\ 0.00 & 0.56 \end{pmatrix}$$

Qui est qui ?

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

Eigen vector =

0.9351	-0.7160
0.3543	0.6981

eigenvalue =

83.7880	0
0	-4.8162

Analyse discriminante linéaire

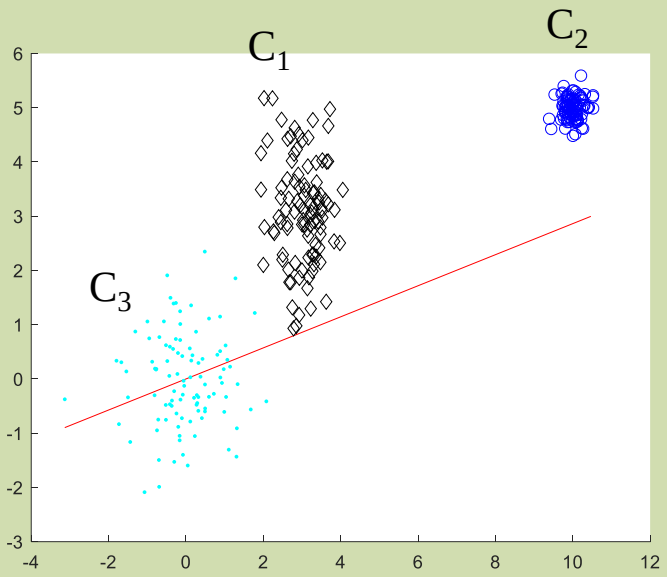
Exercice

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



Les deux matrices $\boldsymbol{\Sigma_{intra}}$ et $\boldsymbol{\Sigma_{inter}}$ sont données par :

$\boldsymbol{\Sigma_{inter}} \begin{pmatrix} 23.04 & 25.76 \\ 12.36 & 9.64 \end{pmatrix}$ $\boldsymbol{\Sigma_{intra}} \begin{pmatrix} 0.33 & 0.00 \\ 0.00 & 0.56 \end{pmatrix}$

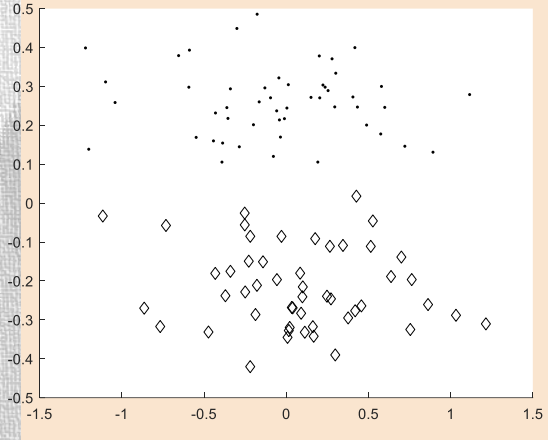
Représenter sur la figure le premier axe obtenu par LDA sachant que les 2 vecteurs/valeurs propre de $\boldsymbol{\Sigma_{intra}^{-1}\Sigma_{inter}}$ sont:

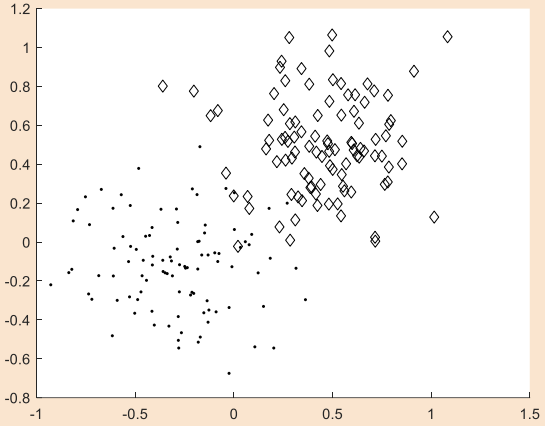
Eigen vector =
0.9351 -0.7160
0.3543 0.6981

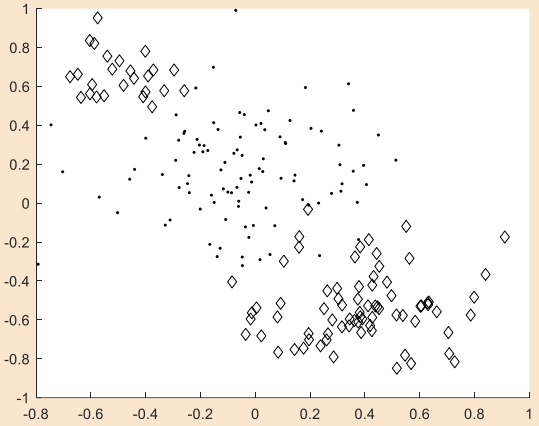
eigenvalue =
83.7880 0
0 -4.8162

Analyse discriminante linéaire

Reprenons les 3 exemples précédents

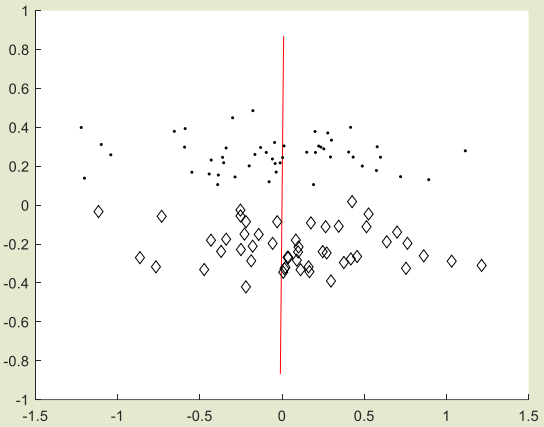




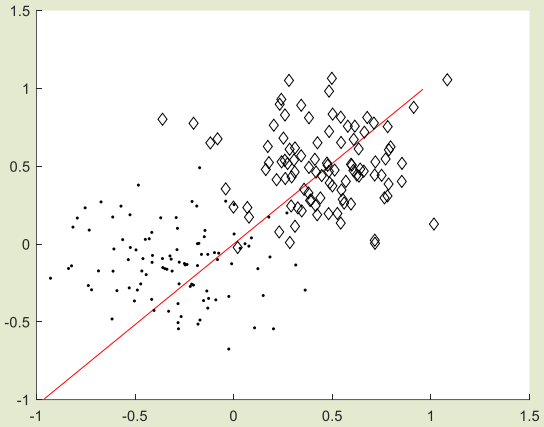


Donner le premier axe principal obtenu par LDA

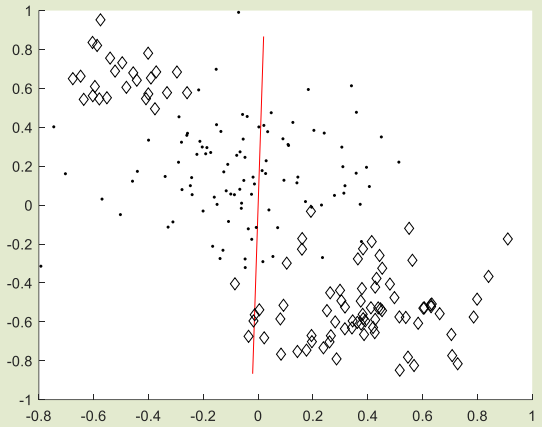
Analyse discriminante linéaire



Bien



Bien



Moyen

Sélection/extraction 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

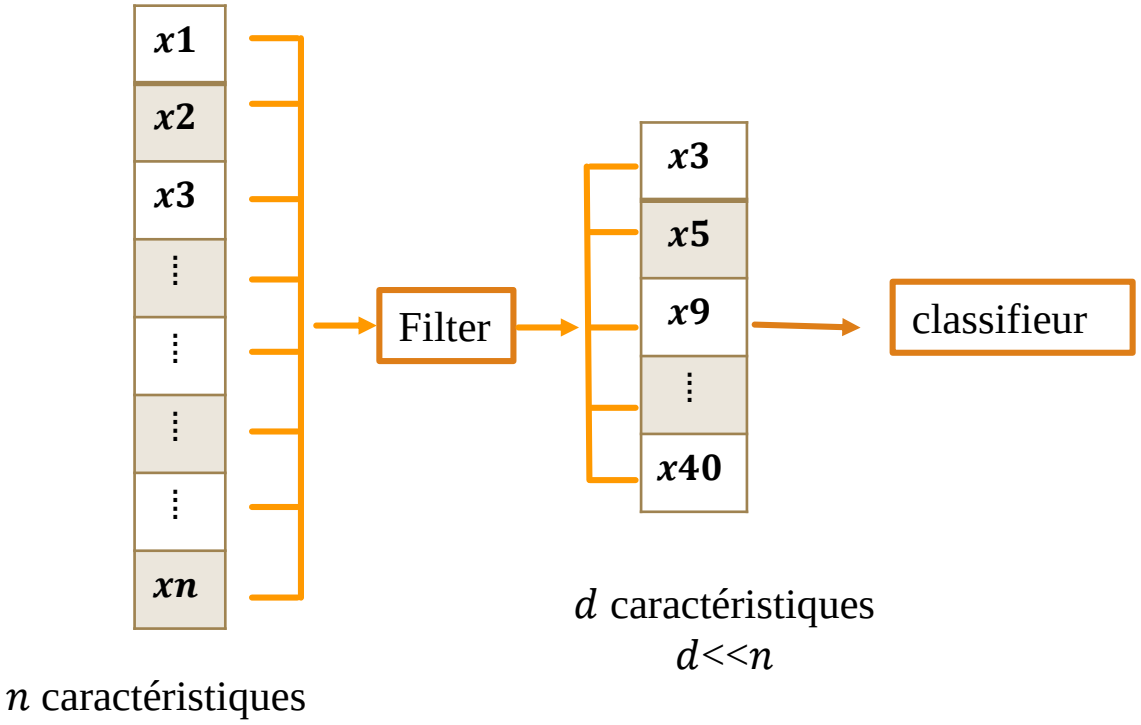
Autre idée : la sélection de caractéristiques
On choisit certaines des dimensions de l'espace initial.
Comment ?
Trois catégories principales :

- Filter
- Wrapper
- Embedded

Sélection/extraction 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/extraction de caractéristiques

Avantage des méthodes de type filter ?

Inconvénient des méthodes de type filter ?

Sélection/extraction de caractéristiques

Avantage des méthodes de type filter

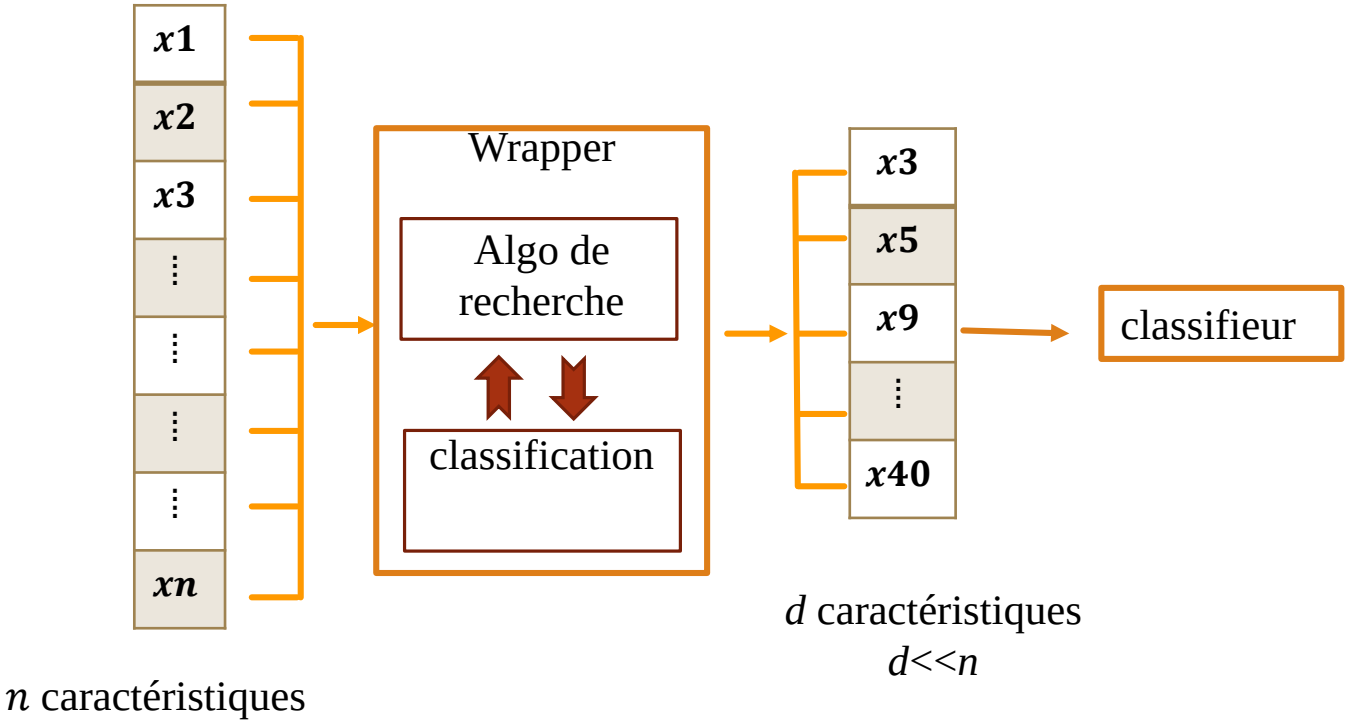
- efficacité calculatoire

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/extraction de caractéristiques

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



Sélection/extraction 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 ?

Sélection/extraction 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
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

Sélection/extraction de caractéristiques

Les **méthodes Embedded ou intégrées** incorporent la sélection de variables lors du processus d'apprentissage (boosting, arbre de décision),

Ces méthodes seront vues dans la suite du cours