Traitement de l'image et du signal
Partie TI

Emanuel Aldea <emanuel.aldea@u-psud.fr>
http://hebergement.u-psud.fr/emi/453

Master Electronique, énergie électrique, automatique 1ère année

Plan du cours

- Définition
- Visualisation de données
- Réduction de dimensionnalité (PCA)
- Analyse linéaire discriminante (LDA)
- La classification non supervisée
  - l'algorithme K-means
- La classification supervisée
  - l'algorithme kNN
  - l'algorithme SVM

La classification

Objectifs

- Obtenir une représentation simplifiée mais pertinente des données originales
- Mettre en évidence les similarités entre les "objets"

Définitions préalables

- Espace d'entrée $\mathbb{X}$ et des instances à traiter $x_i \in \mathbb{X}$
  - Une image, un pixel, une composante connexe etc.
- Espace de caractéristiques $\mathbb{R}^d$ et $y : \mathbb{X} \Rightarrow \mathbb{R}^d : \forall x_i \in \mathbb{X}, y_i = y(x_i) \in \mathbb{R}^d$
  - Un histogramme, un descripteur de forme etc.
  - Besoin d'une métrique
- Espace de décision $\Omega = \{\omega_i, i \in [0c]\}$ et $\forall x \in \mathbb{X}, \omega(x) \in \Omega$
  - $\{0, 1\}$ pour une décision binaire, { "fond", "route", "panneau"} pour une application de navigation autonome etc.
- Critère de performance
  - Comment est-ce qu'on évalue la performance de la classification ?

Exemple en visualisation

La base IRIS

- Des attributs (en cm) portant sur les caractéristiques de trois espèces de plantes : longueur et largeur du sépale, et longueur et largeur du pétale
- dimensionnalité de l'espace de caractéristiques : $d = 4$
- dimensionnalité de l'espace de décision : $\|\Omega\| = 3$
- nombre d'instances $n = 150$ (approx. 50/classe)

Iris setosa

Iris versicolor

Iris virginica

Exemple en visualisation

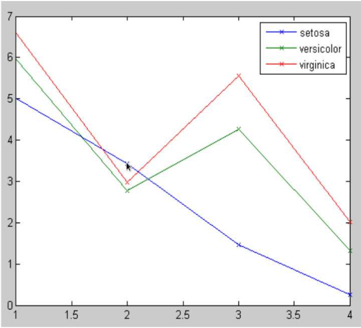
Affichage des moyennes des classes

Moyenne :

	'S Length'	'S Width'	'P Length'	'P 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

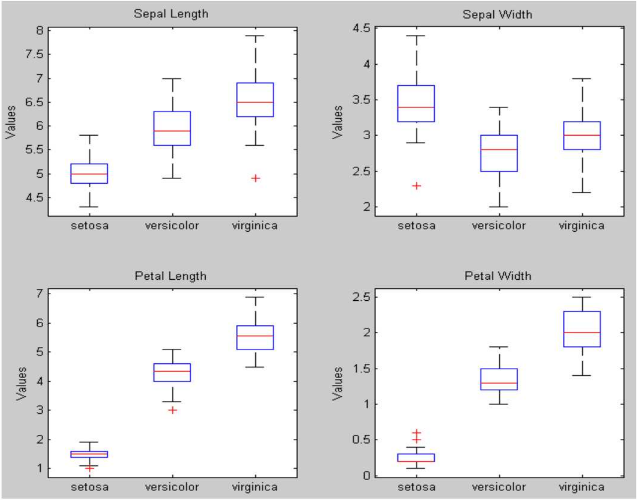
Ecart type :

	'S Length'	'S Width'	'P Length'	'P Width'
'setosa'	0,352	0,379	0,174	0,105
'versicolor'	0,516	0,314	0,470	0,198
'virginica'	0,636	0,322	0,552	0,275

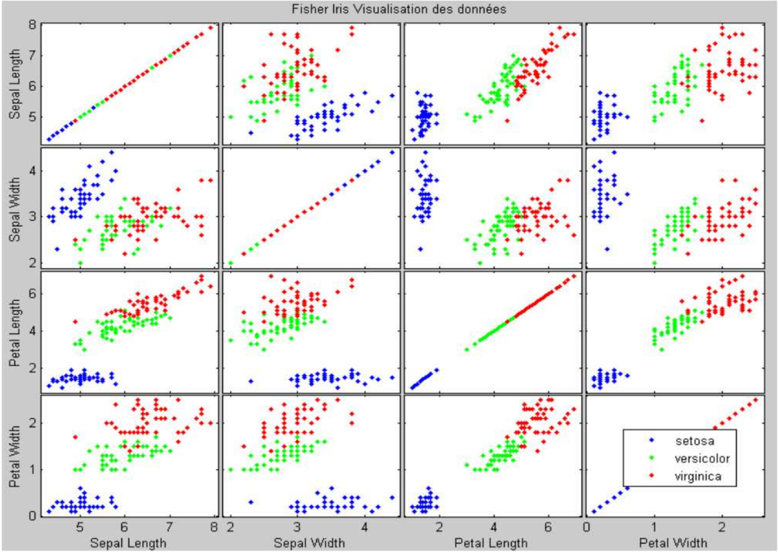


Exemple en visualisation

Plus informatif : les boxplots



Exemple en visualisation - analyse croisée



Exemple en visualisation

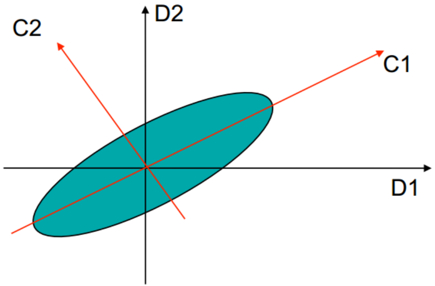
Statistiques descriptives simples

- moyenne, boxplots, analyse croisée
- utiles en dimensions réduites

Analyse en composantes principales (PCA)

- indispensable quand le nombre de variables est très grand
- analyse de la variabilité / dispersion des données
- objectif : décrire à partir de $q < d$ dimensions cette variabilité
- réduction des données a q nouveaux descripteurs
- visualisation si $q = 2$ ou $q = 3$
- interprétation des données : liaisons inter-variables

Analyse en composantes principales

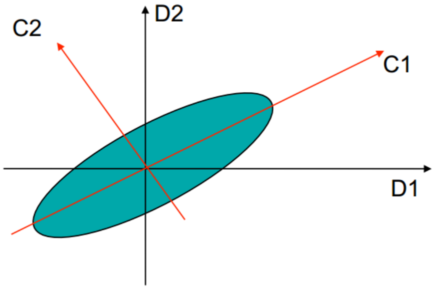


- composantes $C_1, \ldots, C_q$, avec C_k combinaison lineaire de $D_1, \ldots, D_d$

$$C_k = \sum_{i=1}^{d} a_{ik} x_i$$

- les composantes sont deux à deux non corrélées
- les composantes sont de variance maximale
- les composantes sont d'importance décroissante

Analyse en composantes principales

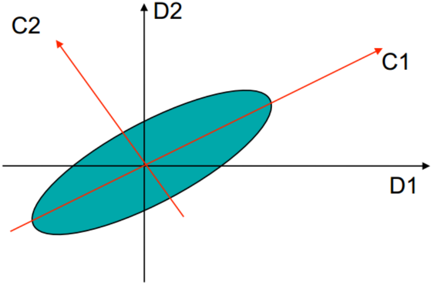


- on peut montrer que la variance de la projection par rapport à une direction $\mathbf{v}$ est :

$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \Sigma \mathbf{v}$$

- Σ étant la matrice de covariance
- on cherche $\max \sigma_{\mathbf{v}}^2$ avec $\mathbf{v}$ unitaire : $\mathbf{v}^T\mathbf{v} = 1$
- on obtient $\Sigma\mathbf{v} = \lambda\mathbf{v}$ et $\sigma_{\mathbf{v}}^2 = \lambda$
- solution PCA : projection sur le vecteur propre ayant la valeur propre λ la plus élevée

Analyse en composantes principales

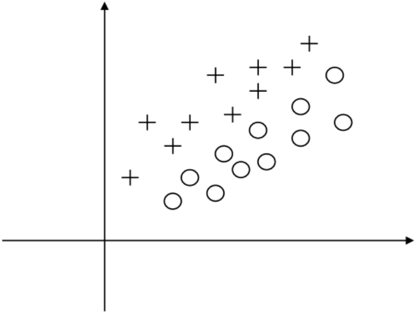


Résumé de l'algorithme

1. recentrage des données $\mathbf{X} = (\mathbf{x} - \mu)^T$
2. calcul de la matrice de covariance Σ
3. diagonalisation de Σ et classement par valeurs propres croissantes
4. sélection des q premiers vecteurs propres C_k
5. calcul des valeurs réduites $\mathbf{a}_i$ qui remplacent $\mathbf{x}_i$ par $a_{ik} = < \mathbf{x}_i, C_k >$

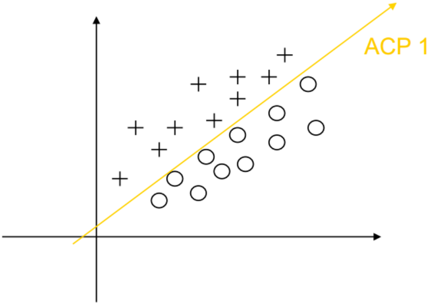
Analyse linéaire discriminante (LDA)

Limitation PCA : ne prend pas en compte la notion de classe



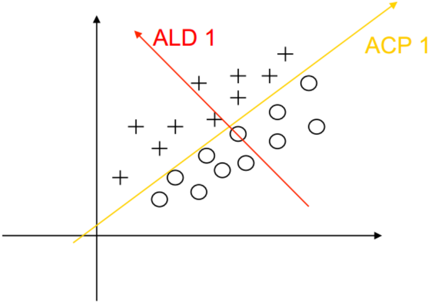
Analyse linéaire discriminante (LDA)

Limitation PCA : ne prend pas en compte la notion de classe



Analyse linéaire discriminante (LDA)

Limitation PCA : ne prend pas en compte la notion de classe



- idée : mettre en évidence des différences entre les classes
- méthode proche de PCA
- maximiser la variance inter-classes
- variance intra-classe minimale

Analyse linéaire discriminante (LDA)

Décomposition de la variance totale :

$$\sigma^2 = \sigma^2_{(w)} + \sigma^2_{(b)}$$

et par rapport à une direction de projection **v** :

$$\sigma^2_{\mathbf{v}} = \sigma^2_{(w)\mathbf{v}} + \sigma^2_{(b)\mathbf{v}} \Leftrightarrow 1 = \frac{\sigma^2_{(w)\mathbf{v}}}{\sigma^2_{\mathbf{v}}} + \frac{\sigma^2_{(b)\mathbf{v}}}{\sigma^2_{\mathbf{v}}}$$

Optimisation de : $\max_{\mathbf{v}} \frac{\mathbf{v}^T \mathbf{B} \mathbf{v}}{\mathbf{v}^T \Sigma \mathbf{v}}$
Condition nécessaire : $\partial_{\mathbf{v}} \left(\frac{\mathbf{v}^T \mathbf{B} \mathbf{v}}{\mathbf{v}^T \Sigma \mathbf{v}} \right) = 0$
D'où : $\Sigma^{-1} \mathbf{B} \mathbf{v} = \lambda \mathbf{v}$
Solution LDA : projection des données sur le vecteur propre de $\Sigma^{-1}\mathbf{B}$ ayant la valeur propre λ la plus élevée.

Analyse linéaire discriminante (LDA)

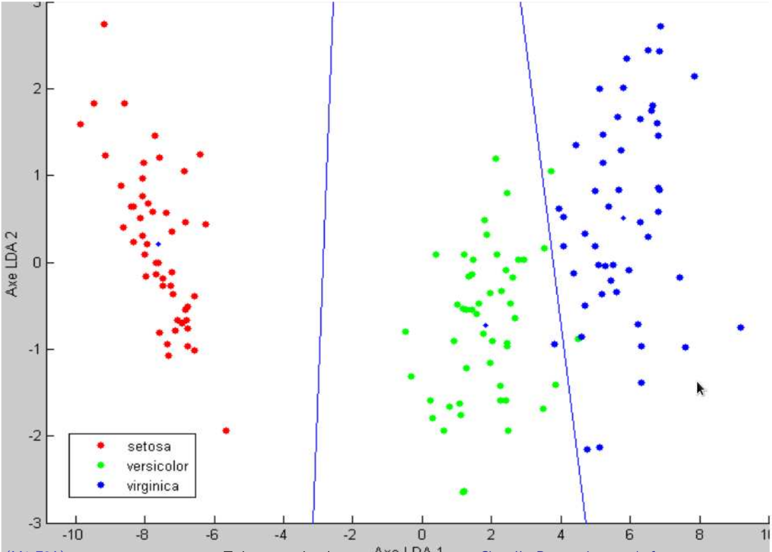
Résumé de l'algorithme

1. recentrage des données $\mathbf{X} = (\mathbf{x} - \mu)^T$
2. calcul de la matrice de covariance Σ
3. calcul de la matrice de covariance inter-classe $\mathbf{B}$:

$$\mathbf{B} = \sum_{k=1}^{c} n(k) \frac{(\mu(k) - \mu)(\mu(k) - \mu)^T}{n}$$

4. diagonalisation de $\Sigma^{-1}\mathbf{B}$ et classement par valeurs propres croissantes
5. sélection des q premiers vecteurs propres C_k
6. calcul des valeurs réduites $\mathbf{a}_i$ qui remplacent $\mathbf{x}_i$ par $a_{ik} = < \mathbf{x}_i, C_k >$
7. classification d'une nouvelle observation par la distance au centroïde le plus proche
8. classification linéaire : médiane entre les centroïdes

Analyse linéaire discriminante (LDA)



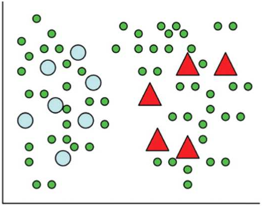
Intégration de la connaissance

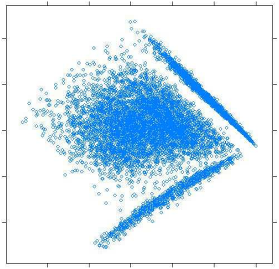
Cas supervisé

- Connaissance a priori des caractéristiques des classes
- Apprentissage à partir d'objets déjà étiquetés (exemples, ou training data)

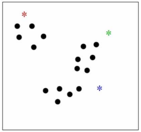
Cas non supervisé

- Définition d'un critère
  - minimisation de la dispersion intra-classe / maximisation de la dispersion inter-classes
  - minimisation de la probabilité d'erreur
  - proposition d'un algorithme d'optimisation
    - convergence
    - caractéristiques de la solution





Classification non supervisée - K-means



- Algorithme itératif qui identifie k clusters dans les observations

$$\min_{\{\mu_1,\ldots,\mu_k\}} \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \mu_i)^2$$

Fonctionnement

```
choose μ_i, i ∈ [1 k] // initialisation
while (any μ_i changes){
  assign all x_j to closest μ
  update μ_i, i ∈ [1 k]
}
```

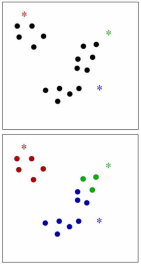
Avantages :

- convergence : à chaque itération la fonction objectif diminue
- rapide
- parallélisable

Classification non supervisée - K-means

- Algorithme itératif qui identifie k clusters dans les observations

$$\min_{\{\mu_1,\dots,\mu_k\}} \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \mu_i)^2$$



Fonctionnement

```
choose μ_i, i ∈ [1 k] // initialisation
while (any μ_i changes){
  assign all x_j to closest μ
  update μ_i, i ∈ [1 k]
}
```

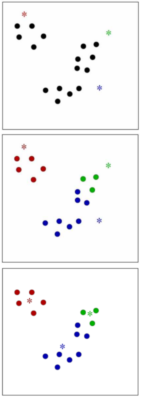
Avantages :

- convergence : à chaque itération la fonction objectif diminue
- rapide
- parallélisable

Classification non supervisée - K-means

- Algorithme itératif qui identifie k clusters dans les observations

$$\min_{\{\mu_1,\dots,\mu_k\}} \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \mu_i)^2$$



Fonctionnement

```
choose μ_i, i ∈ [1 k] // initialisation
while (any μ_i changes){
  assign all x_j to closest μ
  update μ_i, i ∈ [1 k]
}
```

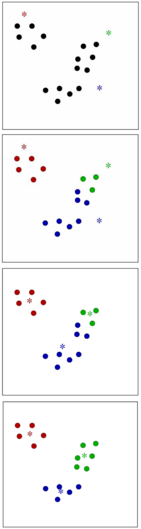
Avantages :

- convergence : à chaque itération la fonction objectif diminue
- rapide
- parallélisable

Classification non supervisée - K-means

- Algorithme itératif qui identifie k clusters dans les observations

$$\min_{\{\mu_1,\dots,\mu_k\}} \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \mu_i)^2$$



Fonctionnement

choose $\mu_i, i \in [1\ k]$ // initialisation
while (any μ_i changes){
  assign all x_j to closest μ
  update $\mu_i, i \in [1\ k]$
}

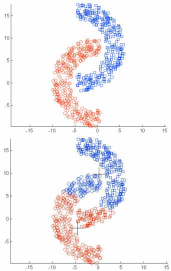
Avantages :

- convergence : à chaque itération la fonction objectif diminue
- rapide
- parallélisable

Classification non supervisée - K-means

Limitations :

- densités ou tailles variables, formes complexes
- initialisation des centres $\mu_i \Leftrightarrow$ minima locaux
- choix du $k \Leftarrow$ techniques d'initialisation
- outliers, clusters vides

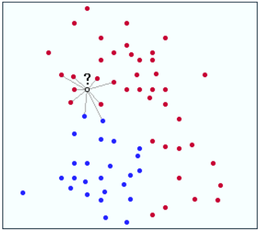


Quelques références :

- Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall
- Banerjee A, Merugu S, Dhillon I, Ghosh J (2005) Clustering with Bregman divergences. J Mach Learn Res
- Nock, R. and Nielsen, F. (2006) On Weighting Clustering, IEEE Trans. on Pattern Analysis and Machine Intelligence

Classification supervisée - kNN

- un ensemble d'apprentissage $\{(y_i, \omega_i)\}$
- requête y ; calcul de ses k plus proches voisins
- décision ω par maximum de votes



Limitations :

- distribution inégale des exemples d'apprentissage
- choix du k
- attributs pas relevants

Améliorations :

- pondération de la distance
- pondération des attributs

Classification supervisée - SVM

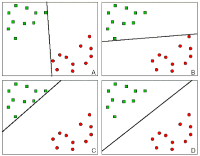
- classification linéaire ambiguë (voir perceptron)
- maximisation de la marge inter-classes

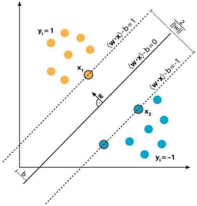
Points forts :

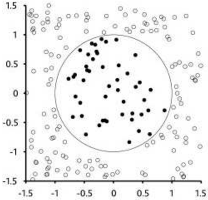
- optimisation globale
- sélection des vecteurs support
- classification non linéaire possible par projection des features dans des espaces de plus grande dimension
- possibilité de tolérer des outliers

Hyperplan de séparation : $\langle \mathbf{w}, \mathbf{x} \rangle - b = 0$
Première classe : $\langle \mathbf{w}, \mathbf{x} \rangle - b \geq 1$
Deuxième classe : $\langle \mathbf{w}, \mathbf{x} \rangle - b \leq -1$

$$\min_{\mathbf{w},b} \|\mathbf{w}\| \quad t.q. \;\; \forall i, \omega_i(\langle \mathbf{w}, \mathbf{x} \rangle - b) \geq 1$$

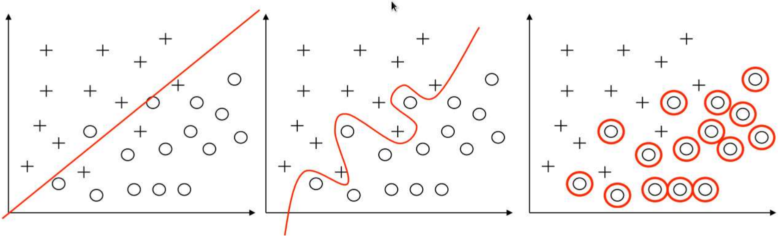




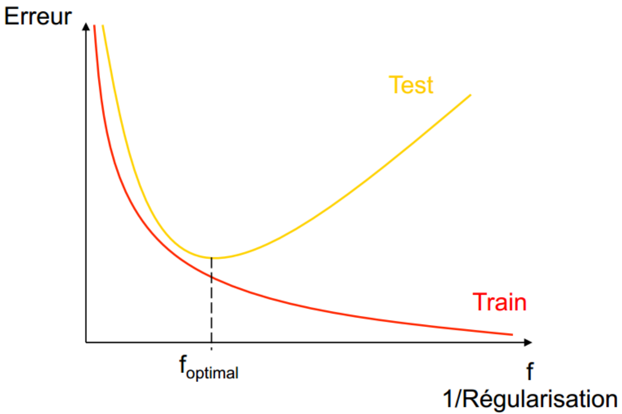


Sur-apprentissage

Comment choisir le bon modèle ?

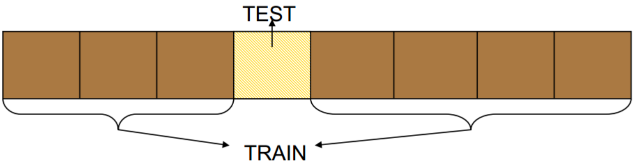


Sur-apprentissage



Sur-apprentissage

Validation croisée



- séparation en N sous-ensembles
- N fois train sur N-1
- erreur totale : moyenne des N taux d'erreur