Apprentissage non-supervisé

Master parcours SSD - UE Apprentissage Statistique I

Pierre Mahé - bioMérieux & Université de Grenoble-Alpes

Apprentissage non-supervisé

Données d'entrée : échantillon $\{x_i\}_{i=1,\ldots,n}$.
$\Rightarrow$ pas de variable de sortie

Objectif : identifier des "structures" dans les données
$\Rightarrow$ "comprendre" les données.

Par exemple :

- sous-groupes dans les observations
- relations entre les variables (corrélation, redondance)
- données aberrantes
- représentations et visualisations informatives

$\Rightarrow$ souvent mené dans un cadre exploratoire
$\Rightarrow$ pas de critère objectif : fortement empirique

Apprentissage non-supervisé

Identifier des structures ou régularités :

1. estimation de densité
2. réduction de dimension
3. clustering
4. détection d'anomalie

Estimation de densité

Estimation de densité

Estimation de densité : savoir où les données "vivent"

Objectifs :

- exploratoire, descriptif
- visualisation
- détection d'anomalies / de données aberrantes

Différentes approches :

- estimation paramétrique "classique"
- estimation non-paramétrique
- modèles de mélanges

Estimation paramétrique

Estimation paramétrique

1. On choisit un modèle probabiliste pour expliquer nos données :
$$P(x) = f(x; \Theta), \;\; \forall x \in \mathcal{X}$$
où :
    - $\mathcal{X}$ est l'espace de définition des données
    - f définit une densité de probabilité :
        - i.e., $f(x;\Theta) \geq 0 \; \forall x, \Theta$ et $\int_{\mathcal{X}} f(x)dx = 1$
    - Θ est un vecteur de paramètres définissant cette loi
        - e.g., $\Theta = (\mu, \sigma)$ pour une loi normale

Estimation paramétrique

1. On choisit un modèle probabiliste pour expliquer nos données :
$$P(x) = f(x; \Theta), \;\; \forall x \in \mathcal{X}$$
où :
   - $\mathcal{X}$ est l'espace de définition des données
   - f définit une densité de probabilité :
     - i.e., $f(x;\Theta) \geq 0 \; \forall x, \Theta$ et $\int_{\mathcal{X}} f(x)dx = 1$
   - Θ est un vecteur de paramètres définissant cette loi
     - e.g., $\Theta = (\mu, \sigma)$ pour une loi normale
2. On choisit un estimateur $\hat{\Theta}$ de Θ.
   - e.g., l'estimateur de maximum de vraisemblance

Estimation paramétrique

1. On choisit un modèle probabiliste pour expliquer nos données :

$$P(x) = f(x; \Theta), \;\; \forall x \in \mathcal{X}$$

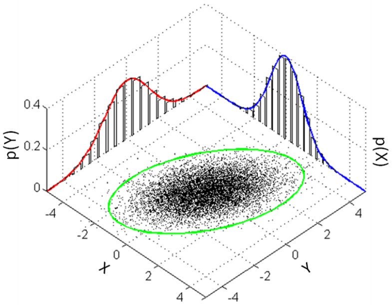
   où :

   - $\mathcal{X}$ est l'espace de définition des données
   - f définit une densité de probabilité :
     - i.e., $f(x;\Theta) \geq 0 \; \forall x, \Theta$ et $\int_{\mathcal{X}} f(x)dx = 1$
   - Θ est un vecteur de paramètres définissant cette loi
     - e.g., $\Theta = (\mu, \sigma)$ pour une loi normale

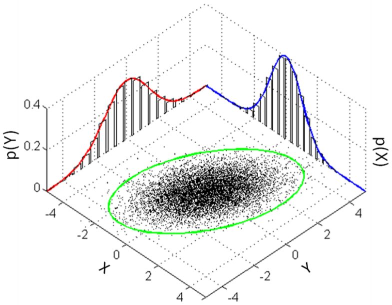
2. On choisit un estimateur $\hat{\Theta}$ de Θ.
   - e.g., l'estimateur de maximum de vraisemblance

3. On travaille à partir de $\hat{\Theta}$
   - e.g., on recherche des "anomalies" : des points où $f(x; \hat{\Theta})$ est faible

Loi Normale multivariée



Loi Normale multivariée



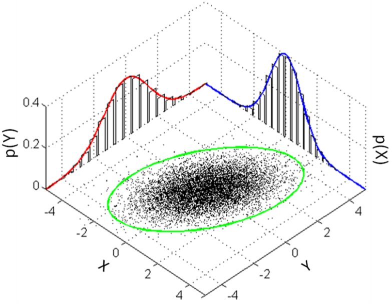
Les observations $\mathbf{x} \in \mathbb{R}^p$ suivent la loi $\mathcal{MN}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$, où :

- $\boldsymbol{\mu} \in \mathbb{R}^p$ est le vecteur moyen
- Σ est la matrice de variance/covariance :

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}, \text{ pour } p = 2$$

(en général $\Sigma = E\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right]$ – de taille $p \times p$)

Loi Normale multivariée - densité



Fonction densité :

$$\mathcal{MN}(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^p|\Sigma|}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big)$$

(NB : loi Normale univariée $\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$)

Loi Normale multivariée - matrice de covariance

Nature de la matrice de covariance Σ :

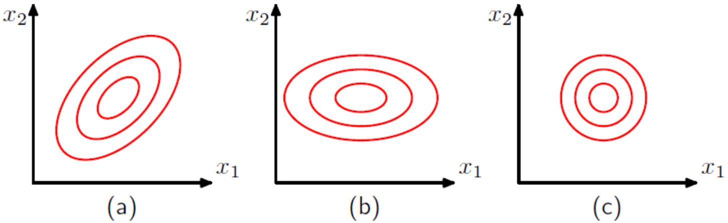


Figure: Image tirée de Bishop (2006)

Loi Normale multivariée - matrice de covariance

Nature de la matrice de covariance Σ :

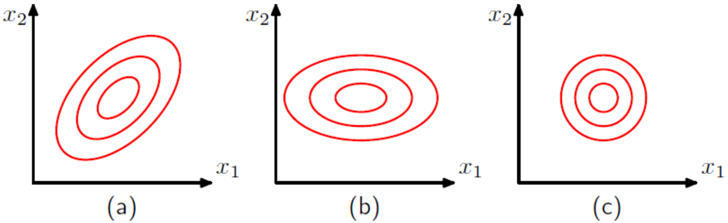


Figure: Image tirée de Bishop (2006)

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

Loi Normale multivariée - matrice de covariance

Nature de la matrice de covariance Σ :

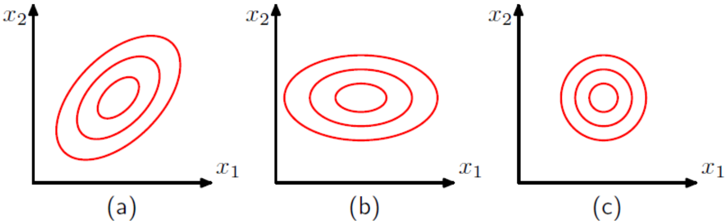


Figure: Image tirée de Bishop (2006)

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

Loi Normale multivariée - matrice de covariance

Nature de la matrice de covariance Σ :

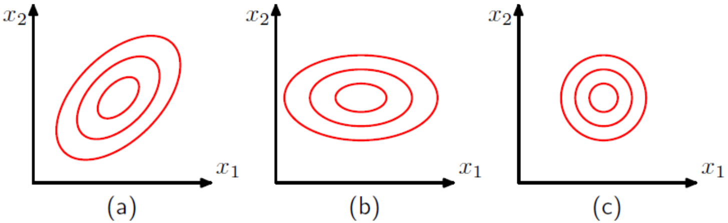


Figure: Image tirée de Bishop (2006)

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

Loi Normale multivariée - distance de Mahalanobis

Fonction densité :

$$\mathcal{MN}(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^p|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\Rightarrow$ distance de Mahalanobis (du point $\mathbf{x}$ à la moyenne $\boldsymbol{\mu}$) :

$$d_{\mathcal{M}}(\mathbf{x},\boldsymbol{\mu}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Loi Normale multivariée - distance de Mahalanobis

Fonction densité :

$$\mathcal{MN}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\Big)$$

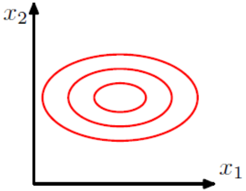
$\Rightarrow$ distance de Mahalanobis (du point $\mathbf{x}$ à la moyenne $\boldsymbol{\mu}$) :

$$d_{\mathcal{M}}(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

- généralise la distance Euclidienne
  - $||\mathbf{x} - \boldsymbol{\mu}|| = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T (\mathbf{x} - \boldsymbol{\mu})}$ $\Big(= \sqrt{(x-\mu)^2} \text{ si } x \in \mathbb{R}\Big)$
- prend en compte les variances/covariances des variables
- $d_{\mathcal{M}}(\mathbf{x}, \mathbf{y})$ est valable pour tout $\mathbf{x}, \mathbf{y} \to \mathcal{MN}(.; \boldsymbol{\mu}, \Sigma)$

Distance de Mahalanobis - illustration

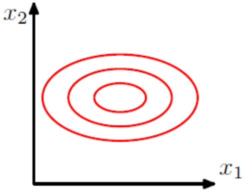
On considère l'exemple suivant
où $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$:



Distance de Mahalanobis - illustration

On considère l'exemple suivant

où $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$:



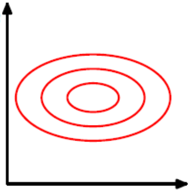
On a :

$$\begin{aligned} d^2_{\mathcal{M}}(\mathbf{x}, \boldsymbol{\mu}) &= (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= \sum_{i=1}^{2} \sum_{j=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i) \times \Sigma^{-1}_{ij} \times (\mathbf{x}_j - \boldsymbol{\mu}_j) \\ &= \sum_{i=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i)^2 \times \Sigma^{-1}_{ii} = \sum_{i=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i)^2 \times \frac{1}{\sigma_i^2} \end{aligned}$$

Distance de Mahalanobis - illustration

On considère l'exemple suivant

où $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$:



On a :

$$\begin{aligned} d^2_{\mathcal{M}}(\mathbf{x}, \boldsymbol{\mu}) &= (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= \sum_{i=1}^{2} \sum_{j=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i) \times \Sigma_{ij}^{-1} \times (\mathbf{x}_j - \boldsymbol{\mu}_j) \\ &= \sum_{i=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i)^2 \times \Sigma_{ii}^{-1} = \sum_{i=1}^{2} (\mathbf{x}_i - \boldsymbol{\mu}_i)^2 \times \frac{1}{\sigma_i^2} \end{aligned}$$

$\Rightarrow$ la distance mesurée selon chaque axe est inversement pondérée par la variance correspondante.
$\Rightarrow$ l'axe 1 "compte moins" que l'axe 2.

Estimation non paramétrique

Estimation non-paramétrique :

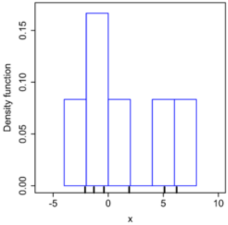
- pas d'hypothèse sur la distribution des données
- nature de la densité dictée par les données

Estimation non paramétrique

Estimation non-paramétrique :

- pas d'hypothèse sur la distribution des données
- nature de la densité dictée par les données

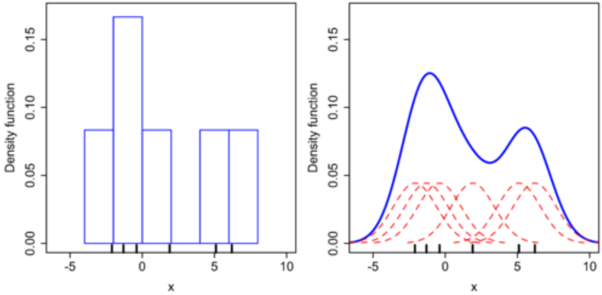
Point de départ : l'histogramme



⇒ comment estimer la densité en tout point du support ?
⇒ comment avoir une propriété de continuité ?

Estimation par noyau - principe

Principe :



- on positionne un "noyau" sur chaque observation
- on les moyenne pour estimer la densité

$\Rightarrow$ méthode de Parzen : Kernel Density Estimation

Estimation par noyau - définition

Formellement, à partir de l'échantillon $(x_1, ..., x_n)$:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i), \quad \forall x \in \mathcal{X}$$

où $K(.)$ est un noyau = une fonction :

- non-négative
- dont l'intégrale vaut 1
- qui est centrée sur zéro

Estimation par noyau - définition

Formellement, à partir de l'échantillon $(x_1, ..., x_n)$:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i), \quad \forall x \in \mathcal{X}$$

où $K(.)$ est un noyau = une fonction :

- non-négative
- dont l'intégrale vaut 1
- qui est centrée sur zéro

⇒ Intuitivement : une moyenne locale, avec une notion de proximité définie par K.

Estimation par noyau - définition

Formellement, à partir de l'échantillon $(x_1, ..., x_n)$:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i), \quad \forall x \in \mathcal{X}$$

où $K(.)$ est un noyau = une fonction :

- non-négative
- dont l'intégrale vaut 1
- qui est centrée sur zéro

⇒ Intuitivement : une moyenne locale, avec une notion de proximité définie par K.

Noyau typique = Gaussien : $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$.

Estimation par noyau - fonction noyau

Noyaux classiques :

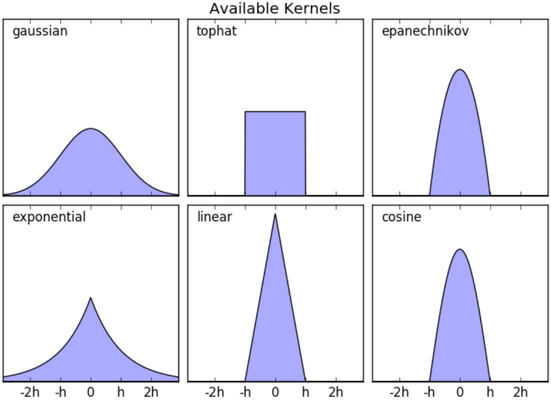


Figure: Noyaux disponibles dans `Scikit-Learn` (et `R`).

Estimation par noyau - fonction noyau

Une question clé : le choix de la largeur de bande

$$\hat{f}(x) = \frac{1}{n}\sum_i K(x - x_i) \Rightarrow \hat{f}_h(x) = \frac{1}{nh}\sum_i K(\frac{x - x_i}{h})$$

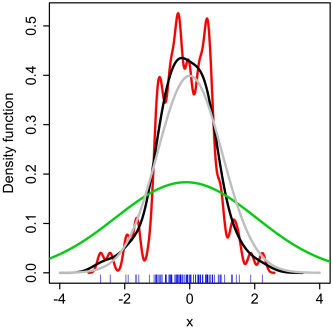


Figure: réalité, h=2, h=0.05, h=0.337

Estimation par noyau - remarques

- Avantage : pas d'hypothèses sur les données = flexibilité

Estimation par noyau - remarques

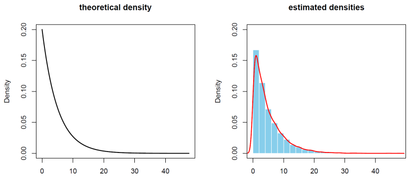
- Avantage : pas d'hypothèses sur les données = flexibilité
- Inconvénients :
  - sensible au choix du noyau...et à sa largeur de bande
  - noyau à évaluer pour toutes les observations pour évaluer la densité en un point

Estimation par noyau - remarques

- Avantage : pas d'hypothèses sur les données = flexibilité
- Inconvénients :
  - sensible au choix du noyau...et à sa largeur de bande
  - noyau à évaluer pour toutes les observations pour évaluer la densité en un point
- En `R` : procédure implémentée par la fonction `density`
  - par défaut : noyau Gaussien, heuristique pour choisir h

Estimation par noyau - remarques

- Avantage : pas d'hypothèses sur les données = flexibilité
- Inconvénients :
  - sensible au choix du noyau...et à sa largeur de bande
  - noyau à évaluer pour toutes les observations pour évaluer la densité en un point
- En `R` : procédure implémentée par la fonction `density`
  - par défaut : noyau Gaussien, heuristique pour choisir h
- Toujours bon de comparer avec l'histogramme
  - exemple de la loi exponentielle :



Modèles de mélanges

Modèle de mélange = modèle probabiliste :

- prenant en compte des sous-populations ...
- ...sans qu'elles soient spécifiées à l'avance

Modèles de mélanges

Modèle de mélange = modèle probabiliste :

- prenant en compte des sous-populations ...
- ...sans qu'elles soient spécifiées à l'avance

$\Rightarrow$ Chaque composante = 1 distribution paramétrique

Modèles de mélanges

Modèle de mélange = modèle probabiliste :

- prenant en compte des sous-populations ...
- ...sans qu'elles soient spécifiées à l'avance

$\Rightarrow$ Chaque composante = 1 distribution paramétrique

$\Rightarrow$ Plusieurs composantes $\rightarrow$ flexibilité.

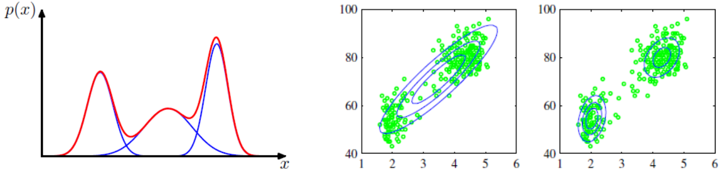
Modèle de mélange = modèle probabiliste :

- prenant en compte des sous-populations ...
- ...sans qu'elles soient spécifiées à l'avance

$\Rightarrow$ Chaque composante = 1 distribution paramétrique

$\Rightarrow$ Plusieurs composantes $\rightarrow$ flexibilité.

Illustration : mélanges de Gaussiennnes (1D et 2D) :



Modèles de mélanges - définition

Mélange de Gaussiennes à K composantes :

$$f(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k), \qquad \text{si } x \in \mathbb{R}$$

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{MN}(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k), \qquad \text{si } \mathbf{x} \in \mathbb{R}^p,\ p > 1.$$

Les paramètres $\{\pi_k\}$ = proportions de mélange : $\sum_{k=1}^{K} \pi_k = 1$.

Modèles de mélanges - définition

Mélange de Gaussiennes à K composantes :

$$f(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k), \qquad \text{si } x \in \mathbb{R}$$

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{MN}(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k), \qquad \text{si } \mathbf{x} \in \mathbb{R}^p,\ p > 1.$$

Les paramètres $\{\pi_k\}$ = proportions de mélange : $\sum_{k=1}^{K} \pi_k = 1.$

Questions :

1. estimer les paramètres
2. choisir le nombre de composantes K

$\Rightarrow$ voir prochain cours (lien avec méthode K-means)

Réduction de dimension

Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

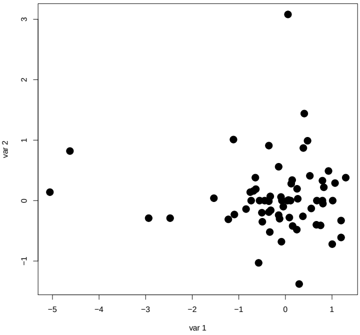
- n observations $\times$ p variables

Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 2 :

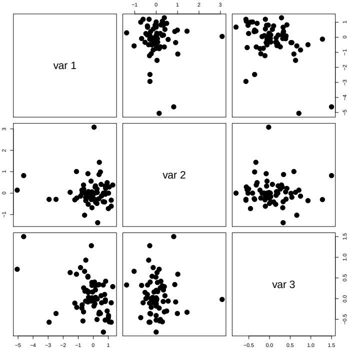


Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 3 :

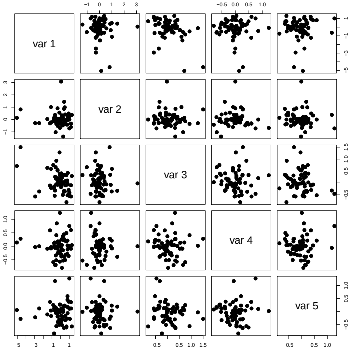


Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 5 :

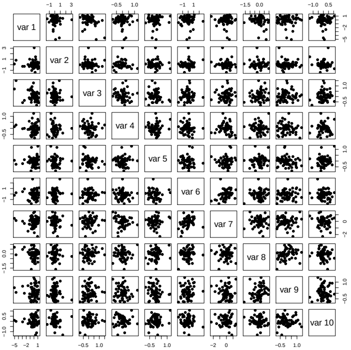


Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 10 :

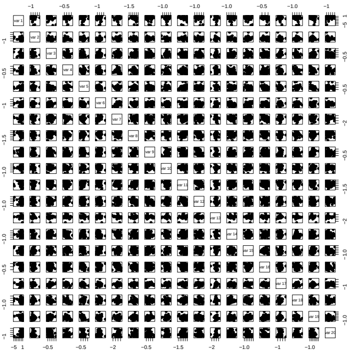


Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 20 :

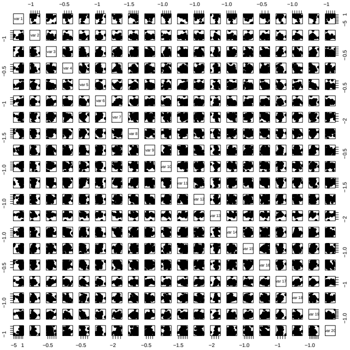


Réduction de dimension - motivation

Visualiser un jeu de données $X \in \mathbb{R}^{n \times p}$

- n observations $\times$ p variables

$\Rightarrow$ p = 20 :



....et si $p = 100$? ou $p = 1.000$? ? ou $p = 100.000$? ? ?

Réduction de dimension - motivation

Objectif : fournir une représentation compacte des données

Applications :

- Exploratoire - visualisation
- Compression des données
- Réduction de variables pour analyses ultérieures
  - e.g., clustering ou apprentissage supervisé

Challenge :

- conserver le plus d'information possible

Méthode clé : l'Analyse en Composantes Principales.

Analyse en Composantes Principales

PCs = combinaisons linéaires des variables d'entrée

Analyse en Composantes Principales

PCs = combinaisons linéaires des variables d'entrée

$\Rightarrow$ si on note :

- $X \in \mathbb{R}^{n \times p}$ la matrice de données
- $X_1, .., X_p$ les vecteurs colonnes (= les p variables)

alors $\mathrm{PC}_i = a_{i1} X_1 + a_{i2} X_2 + ... + a_{ip} X_p$.

Analyse en Composantes Principales

PCs = combinaisons linéaires des variables d'entrée

$\Rightarrow$ si on note :

- $X \in \mathbb{R}^{n \times p}$ la matrice de données
- $X_1, .., X_p$ les vecteurs colonnes (= les p variables)

alors $\text{PC}_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

Question : comment estimer les coefficients a_{ij} ?

Analyse en Composantes Principales

PCs = combinaisons linéaires des variables d'entrée

$\Rightarrow$ si on note :

- $X \in \mathbb{R}^{n \times p}$ la matrice de données
- $X_1, .., X_p$ les vecteurs colonnes (= les p variables)

alors $\mathrm{PC}_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

Question : comment estimer les coefficients a_{ij} ?

Critère de l'ACP pour maintenir le maximum d'information :

1. PC_1 doit avoir la plus grande variance possible
2. PC_i doit avoir la plus grande variance possible, dans une direction orthogonale à $\{PC_1, ..., PC_{i-1}\}$

ACP - formalisation

Rappel - PCs : $PC_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

ACP - formalisation

Rappel - PCs : $\text{PC}_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

Si les X_i sont centrées, alors les PCs le sont également et :

$$\text{Var}(PC_1) = \frac{1}{n}\sum_{i=1}^{n} \text{PC}_{1i}^2 = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{p} a_{1j}X_{ij}\Big)^2$$

ACP - formalisation

Rappel - PCs : $\mathrm{PC}_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

Si les X_i sont centrées, alors les PCs le sont également et :

$$\mathrm{Var}(PC_1) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{PC}_{1i}^2 = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{p} a_{1j}X_{ij}\Big)^2$$

$\Rightarrow$ on obtient PC_1 en maximisant cette expression par rapport à $(a_{11}, .., a_{1p})$....sous la contrainte $\sum_{j=1}^{p} a_{1j}^2 = 1$.

ACP - formalisation

Rappel - PCs : $\mathrm{PC}_i = a_{i1}X_1 + a_{i2}X_2 + ... + a_{ip}X_p$.

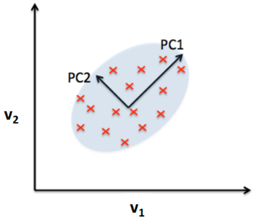
Si les X_i sont centrées, alors les PCs le sont également et :

$$\mathrm{Var}(PC_1) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{PC}_{1i}^2 = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{p} a_{1j}X_{ij}\Big)^2$$

$\Rightarrow$ on obtient PC_1 en maximisant cette expression par rapport à $(a_{11}, .., a_{1p})$....sous la contrainte $\sum_{j=1}^{p} a_{1j}^2 = 1$.

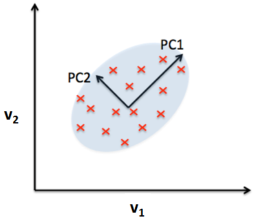
On obtient PC_2 en appliquant la même procédure après avoir orthogonalisé la matrice X par rapport à PC_1.
$\Rightarrow$ ACP = problème de décomposition en valeurs singulières.

ACP - illustration



- $PC_i = a_{i1}v_1 + a_{i2}v_2$
- PC_1 = la plus forte variance
- PC_2 = la plus forte variance résiduelle

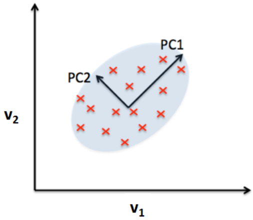
ACP - illustration



- $\mathrm{PC}_i = a_{i1}v_1 + a_{i2}v_2$
- PC_1 = la plus forte variance
- PC_2 = la plus forte variance résiduelle

A retenir :

ACP - illustration

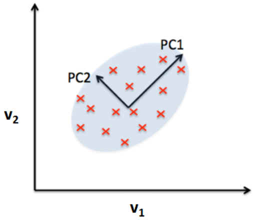


- $PC_i = a_{i1} v_1 + a_{i2} v_2$
- PC_1 = la plus forte variance
- PC_2 = la plus forte variance résiduelle

A retenir :

- quand p est grand : la première étape d'analyse

ACP - illustration

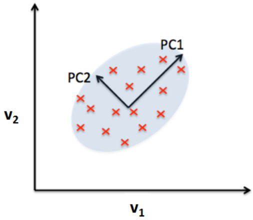


- $\mathrm{PC}_i = a_{i1}v_1 + a_{i2}v_2$
- $\mathrm{PC}_1 =$ la plus forte variance
- $\mathrm{PC}_2 =$ la plus forte variance résiduelle

A retenir :

- quand p est grand : la première étape d'analyse
- si $X \in \mathbb{R}^{n \times p} \Rightarrow \min(n, p)$ composantes principales

ACP - illustration

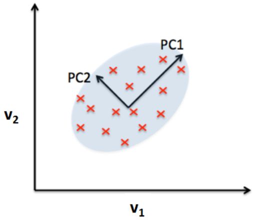


- $PC_i = a_{i1}v_1 + a_{i2}v_2$
- PC_1 = la plus forte variance
- PC_2 = la plus forte variance résiduelle

A retenir :

- quand p est grand : la première étape d'analyse
- si $X \in \mathbb{R}^{n \times p} \Rightarrow \min(n, p)$ composantes principales
- les PCs sont ordonnées de la plus à la moins informative

ACP - illustration



- $\mathrm{PC}_i = a_{i1} v_1 + a_{i2} v_2$
- PC_1 = la plus forte variance
- PC_2 = la plus forte variance résiduelle

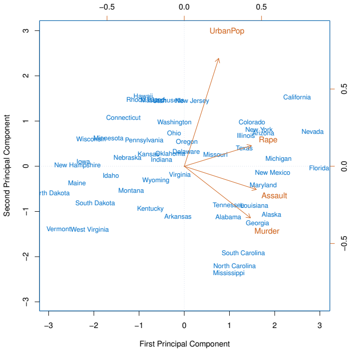
A retenir :

- quand p est grand : la première étape d'analyse
- si $X \in \mathbb{R}^{n \times p} \Rightarrow \min(n, p)$ composantes principales
- les PCs sont ordonnées de la plus à la moins informative
- pouvoir informatif = proportion de variance expliquée :

$$\boxed{\mathrm{var}(\mathrm{PC}_i) / \sum_j \mathrm{var}(\mathrm{PC}_j)}$$

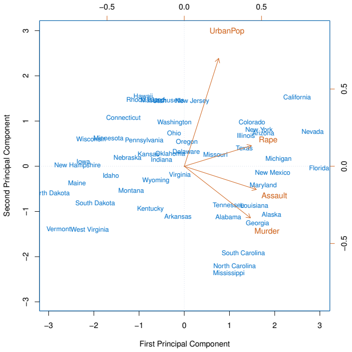
ACP en pratique

Exploration du jeu de données :



ACP en pratique

Exploration du jeu de données :

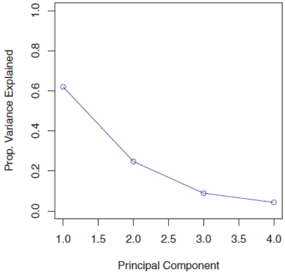


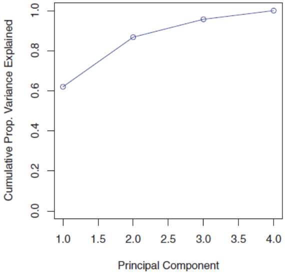
1. au niveau des observations : relations de proximités
   - valeurs prises par PC_i
2. au niveau des variables : contributions aux axes
   - coefficients ("loadings") a_{ij}

ACP en pratique

Combien de PCs considérer ?

⇒ "scree plot" : % de variance expliquée par composante

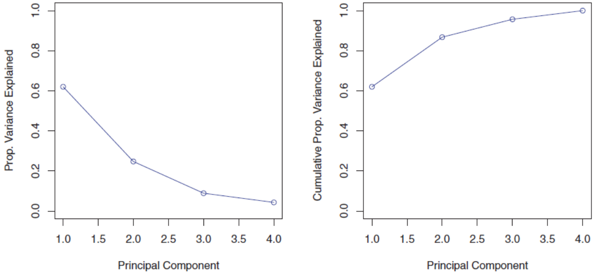




ACP en pratique

Combien de PCs considérer ?

$\Rightarrow$ "scree plot" : % de variance expliquée par composante



Critère objectif pour :

- choisir le nombre de composantes à garder
- mesurer le ratio "compression/information"

ACP en pratique

ACP en préalable au clustering ou approches supervisées ?

- modèles de mélange : - de paramètres à estimer
- régression linéaire quand $p > n$
  - solution des moindres carrés mal définie :

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

  - ACP = alternative à selection "foward/backward"
  - (NB : cadre prédictif, pas inférence)
- ...

Au delà de l'ACP...

ACP :

- + : simple à mettre en oeuvre
- + : efficace
- + / - : +/- facile à interpréter
- - : limité au cadre linéaire et Euclidien

Au delà de l'ACP...

ACP :

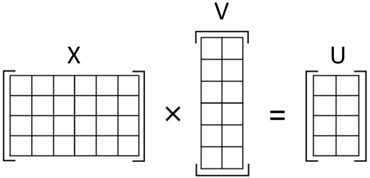
- + : simple à mettre en oeuvre
- + : efficace
- + / - : +/- facile à interpréter
- - : limité au cadre linéaire et Euclidien

Quelques méthodes alternatives :

- Factorisation de matrice non-negative
- Multi-Dimensional Scaling (MDS / PCoA)
- tSNE
- ...

Factorisation de matrice non négative

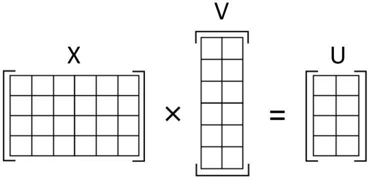
Projection ACP et réduction de dimension :



- $X \in \mathbb{R}^{n \times p}$: données d'entrée, n observations, p variables
- $U \in \mathbb{R}^{n \times k}$: k composantes principales
- $V \in \mathbb{R}^{p \times k}$: "loadings" des k premières PCs, $V^T V = I$

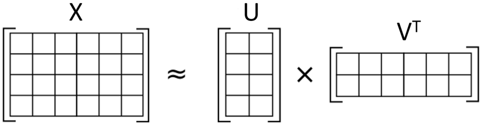
Factorisation de matrice non négative

Projection ACP et réduction de dimension :



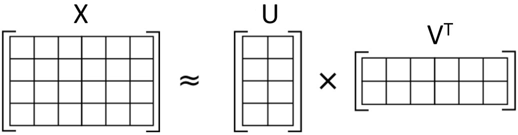
- $X \in \mathbb{R}^{n \times p}$: données d'entrée, n observations, p variables
- $U \in \mathbb{R}^{n \times k}$: k composantes principales
- $V \in \mathbb{R}^{p \times k}$: "loadings" des k premières PCs, $V^T V = I$

$\Rightarrow$ interprétation en factorisation de matrice :



Factorisation de matrice non négative

ACP et factorisation de matrice :



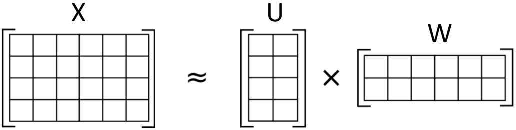
$\Rightarrow$ solution : $\min_{U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{p\times k}} ||X - UV^T||_F^2$, avec $V^T V = I$.

- $||M||_F^2 = \sum_i \sum_j M_{ij}^2 =$ norme de Frobenius
- en pratique : résolu par SVD

$\Rightarrow$ permet de nombreuses généralisations...

Factorisation de matrice non négative

Factorisation de matrice non-négative :

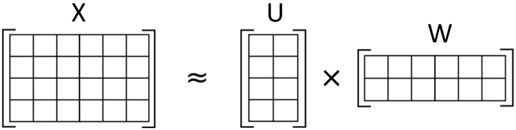


- $X \in \mathbb{R}^{n \times p}$, $U \in \mathbb{R}^{n \times k}$, $W \in \mathbb{R}^{k \times p}$
- U et W non-négatives : $U_{ij} \geq 0, W_{ij} \geq 0, \forall i, j$

$\Rightarrow$ positivité $=$ interprétation "additive"

Factorisation de matrice non négative

Factorisation de matrice non-négative :



- $X \in \mathbb{R}^{n\times p}$, $U \in \mathbb{R}^{n\times k}$, $W \in \mathbb{R}^{k\times p}$
- U et W non-négatives : $U_{ij} \geq 0, W_{ij} \geq 0, \forall i, j$

$\Rightarrow$ positivité = interprétation "additive"

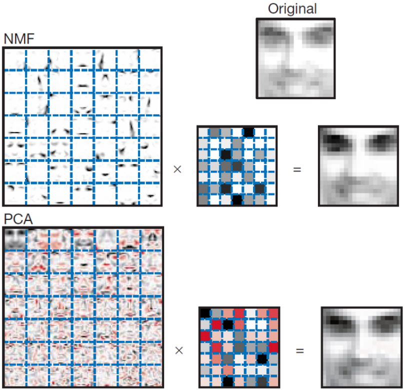
Solution :

$$\min_{U\in\mathbb{R}^{n\times k}, W\in\mathbb{R}^{k\times p}} ||X - UW||_F^2 \,, \text{ avec } U \geq 0, W \geq 0.$$

- problème d'optimisation $\neq$ SVD
- nombreuses extensions (parcimonie, critère d'erreur)

Factorisation de matrice non négative

Illustration (Lee and Seung) :



- NMF : reconstruction additive de "parties" de visages
- ACP : difficile d'interpréter les "loadings"
- beaucoup d'applications en traitement du signal

Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) :

- en entrée : une matrice de distance D de taille $n \times n$
- en sortie : projections $(z_1, z_2, ..., z_n) \in \mathbb{R}^k$
- critère : préserver les distances dans la projection

Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) :

- en entrée : une matrice de distance D de taille $n \times n$
- en sortie : projections $(z_1, z_2, ..., z_n) \in \mathbb{R}^k$
- critère : préserver les distances dans la projection

Différentes formulations...typiquement :

$$\boxed{\min \sum_{i \neq j} \left(D_{i,j} - ||z_i - z_j||\right)^2.}$$

Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) :

- en entrée : une matrice de distance D de taille $n \times n$
- en sortie : projections $(z_1, z_2, ..., z_n) \in \mathbb{R}^k$
- critère : préserver les distances dans la projection

Différentes formulations...typiquement :

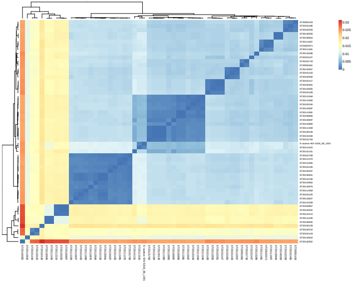
$$\boxed{\min \sum_{i \neq j} \left(D_{i,j} - ||z_i - z_j||\right)^2.}$$

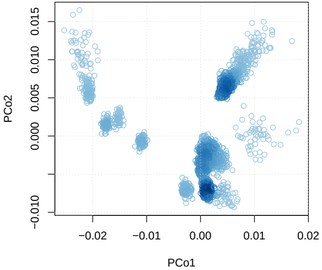
"classical" MDS $\sim$ ACP à partir d'une matrice de distance

- solution par décomposition en valeurs singulières
- si D est la distance Euclidienne : équivalent à l'ACP
- également appelé PCoA : Principal Coordinate Analysis

Multi-Dimensional Scaling

Illustration : comparaison de génomes bactériens





- gauche : matrice de distance entre génomes bactériens
  - (distance `mash`, matrice sous-échantillonnée)
- droite : projection en 2 dimensions par cMDS / PCoA

⇒ en `R` : fonction `cmdscale`.

tSNE - Student Stochastic Neighbour Embedding

(t)SNE : même philosophie que MDS

- passer de points $\{x_i\} \in \mathbb{R}^p, i = 1, ..., n$,
- à des projections $\{z_i\} \in \mathbb{R}^k, i = 1, ..., n$, avec $k << p$,
- en préservant les distances $||x_i - x_j||$.

tSNE - Student Stochastic Neighbour Embedding

(t)SNE : même philosophie que MDS

- passer de points $\{x_i\} \in \mathbb{R}^p, i = 1, ..., n$,
- à des projections $\{z_i\} \in \mathbb{R}^k, i = 1, ..., n$, avec $k << p$,
- en préservant les distances $||x_i - x_j||$.

Cadre probabiliste (Stochastic Neighbour Embedding) :

$$\boxed{p_{j|i} = \frac{\exp\left(-||x_i - x_j||^2/2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-||x_i - x_k||^2/2\sigma_i^2\right)}}$$

$\Rightarrow \sim$ probabilité que x_i tire x_j comme voisin selon $\mathcal{N}(x_i, \sigma_i)$.

$\Rightarrow$ σ_i ajusté pour que chaque x_i ait le même nombre de voisins "effectifs" (lié au paramètre de **perplexité**).

- distributions $p_{j|i}$ ont toutes la même entropie.

tSNE

SNE "classique : considère le même modèle dans $\mathbb{R}^k$

$$q_{j|i} = \frac{\exp\big(-||z_i - z_j||^2\big)}{\sum_{k\neq i}\exp\big(-||z_i - z_k||^2\big)}$$

- avec la même variance pour tout le monde

tSNE

SNE "classique : considère le même modèle dans $\mathbb{R}^k$

$$q_{j|i} = \frac{\exp\big(-||z_i - z_j||^2\big)}{\sum_{k\neq i}\exp\big(-||z_i - z_k||^2\big)}$$

- avec la même variance pour tout le monde

Principe : minimiser la divergence entre les distributions $p_{j|i}$ et $q_{j|i}$ (pour $i = 1, ..., n$) :

$$\boxed{C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}}$$

$\Rightarrow$ un problème d'optimisation (descente de gradient)

tSNE

Critère à minimiser :

$$\boxed{C = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}}$$

$\Rightarrow$ point clé : préserve les distances faibles

- coût fort si $||x_i - x_j||$ faible et $||z_i - z_j||$ élevé
  - $p_{j|i}$ élevé et $q_{j|i}$ faible
- coût faible si $||x_i - x_j||$ élevé et $||z_i - z_j||$ faible
  - $p_{j|i}$ faible et $q_{j|i}$ élévé

tSNE

Critère à minimiser :

$$\boxed{C = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}}$$

$\Rightarrow$ point clé : préserve les distances faibles

- coût fort si $||x_i - x_j||$ faible et $||z_i - z_j||$ élevé
  - $p_{j|i}$ élevé et $q_{j|i}$ faible
- coût faible si $||x_i - x_j||$ élevé et $||z_i - z_j||$ faible
  - $p_{j|i}$ faible et $q_{j|i}$ élévé

$\Rightarrow$ comportement inverse du MDS :

$$\boxed{C = \sum_{i \neq j} \left(D_{i,j} - ||z_i - z_j|| \right)^2}$$

- coût dominé par les distances $D_{i,j}$ élevées

tSNE

tSNE[1] : deux modifications au SNE "classique"

- utiliser une loi de Student pour $q_{j|i}$
- utiliser une fonction de perte un peu différente

$\Rightarrow$ mieux adapté aux données en haute dimension

1. van der Maaten and Hinton (2008)

tSNE

tSNE[1] : deux modifications au SNE "classique"

- utiliser une loi de Student pour $q_{j|i}$
- utiliser une fonction de perte un peu différente

$\Rightarrow$ mieux adapté aux données en haute dimension

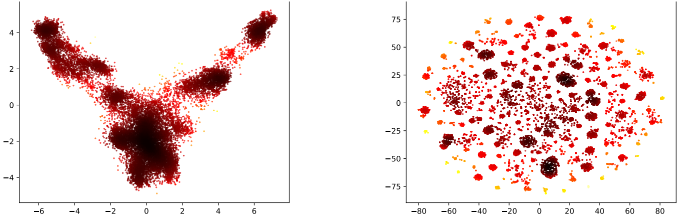
En pratique :

- très adapté aux données en haute dimension...mais optimisation parfois un peu lourde
  - commencer par réduire la dimension par ACP !
- permet surtout d'interpréter les groupes (clusters)
  - accent moindre sur distances lointaines
- outil de visualisation
- fort impact des paramètres (perplexité et # itérations)

1. van der Maaten and Hinton (2008)

tSNE

Illustration : analyse de séquences génomiques bactériennes



- gauche : deux premières composantes principales
  - (représentation en profil de k-mers)
- droite : projection des 50 premières PCs en 2 dimensions

⇒ en R : plusieurs packages (e.g., `Rtsne` et `tsne`).

Clustering

Clustering

Clustering = classification non-supervisée

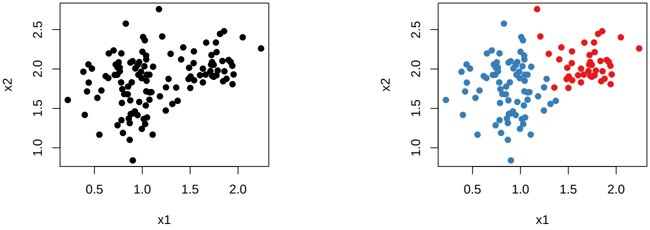
- catégoriser les observations (sous-populations)
- ...ou catégoriser les variables (corrélation / redondance)

Objectifs : exploratoire

- présence de sous-groupes dans les données
- adéquation critères de similarité / données

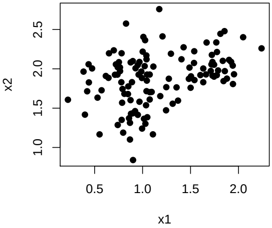
Clustering

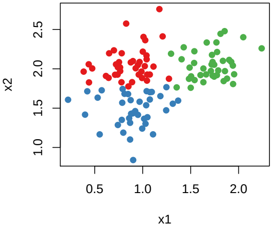
Catégoriser les observations ?



Clustering

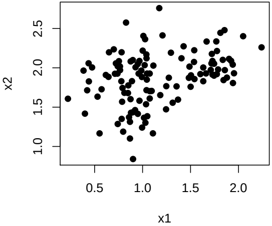
Catégoriser les observations ? Oui mais...

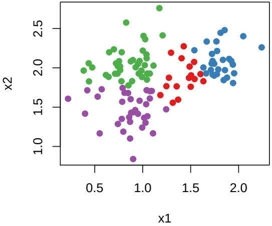




Clustering

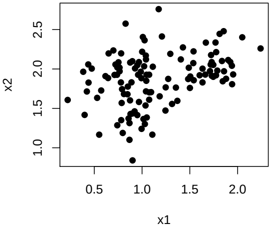
Catégoriser les observations ? Oui mais...

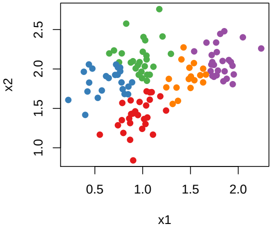




Clustering

Catégoriser les observations ? Oui mais...





Clustering - qualité

But du clustering :

- déterminer des ensembles de points proches ...
- ... qui soient distants les uns des autres

Clustering - qualité

But du clustering :

- déterminer des ensembles de points proches ...
- ... qui soient distants les uns des autres

Fonction objective (à minimiser) = dispersion "intra" cluster

$$W(C) = \sum_{k=1}^{K} \sum_{i:C(i)=k} \sum_{j:C(j)=k} d(x_i, x_j)$$

- K = nombre de clusters
- C = clustering : $C(i) = k \Leftrightarrow x_i \in$ cluster k
- $d(x, y)$ = distance/disimilarité entre x et y

Clustering - qualité

But du clustering :

- déterminer des ensembles de points proches ...
- ... qui soient distants les uns des autres

Fonction objective (à minimiser) = dispersion "intra" cluster

$$W(C) = \sum_{k=1}^{K} \sum_{i:C(i)=k} \sum_{j:C(j)=k} d(x_i, x_j)$$

- K = nombre de clusters
- C = clustering : $C(i) = k \Leftrightarrow x_i \in$ cluster k
- $d(x, y)$ = distance/disimilarité entre x et y

$\Rightarrow$ problème combinatoire, présence de minima locaux.

Clustering - challenges

Questions centrales :

- choisir la fonction de distance entre observations
  - dicté par nature du problème et des données
- choisir le nombre de clusters
  - pas de réponse absolue, tester plusieurs valeurs
- évaluer la stabilité du clustering

Méthodes clé :

- clustering hiérarchique
- K-means
- modèles de mélanges

$\Rightarrow$ le programme des 2 prochains cours.

Détection d'anomalie

Outlier & Novelty detection

Objectif = identifier des "anomalies" dans les données

Deux configurations :

1. au sein d'un jeu de données
2. vis à vis d'un jeu de données existant

⇒ outlier detection vs novelty detection

Nombreuses méthodes ... ici :

- modélisation paramétrique de distribution
- "One-Class SVM" & estimation non paramétrique
- "Local Outlier Factor" sur la base des voisins
- "Isolation Forest" par algorithme de classification

Détection d'anomalie & modélisation paramétrique

L'approche la plus simple :

1. on choisit un modèle probabiliste pour expliquer nos données :
$$P(x) = f(x; \Theta), \ \ \forall x \in \mathcal{X}$$
où :
    - f définit une densité de probabilité :
        - i.e., $f(x;\Theta) \geq 0 \ \forall x, \Theta$ et $\int_{\mathcal{X}} f(x) dx = 1$
    - Θ est un vecteur de paramètres définissant cette loi
2. on choisit un estimateur $\hat{\Theta}$ de Θ.
3. outlier / nouveauté : point x où $f(x; \hat{\Theta})$ est faible

Détection d'anomalie & modélisation paramétrique

L'approche la plus simple :

1. on choisit un modèle probabiliste pour expliquer nos données :
$$P(x) = f(x; \Theta), \ \forall x \in \mathcal{X}$$
où :
   - f définit une densité de probabilité :
     - i.e., $f(x; \Theta) \geq 0 \ \forall x, \Theta$ et $\int_{\mathcal{X}} f(x)dx = 1$
   - Θ est un vecteur de paramètres définissant cette loi
2. on choisit un estimateur $\hat{\Theta}$ de Θ.
3. outlier / nouveauté : point x où $f(x; \hat{\Theta})$ est faible

$\Rightarrow$ Typiquement : en utilisant une gaussienne multivariée

- après une ACP si beaucoup de variables

Détection d'anomalie & modélisation paramétrique

Illustration :

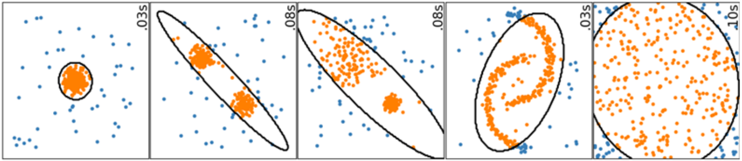


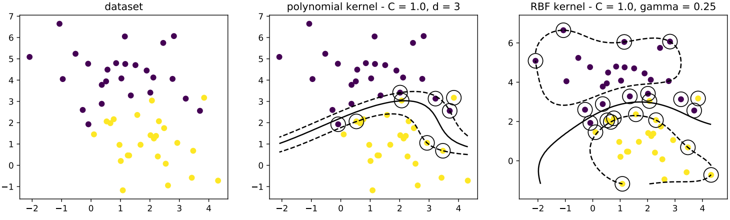
Figure: images tirées de la documentation de `scikit-learn`

Questions / limites :

- modèle gaussien pas toujours adapté
  - modèle de mélange ? combien de composantes ?
- "x où $f(x;\hat{\Theta})$ est faible" : seuil à définir
- beaucoup de paramètres à estimer $(\sim p^2/2)$
  - d'où intérêt ACP - même en dimension modeste

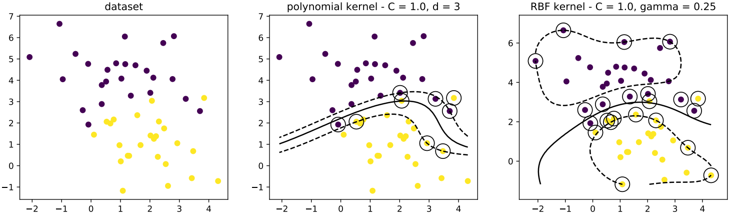
One-Class SVM & estimation non paramétrique

Support Vector Machines : algorithme de classification binaire



One-Class SVM & estimation non paramétrique

Support Vector Machines : algorithme de classification binaire



$\Rightarrow$ se formalise comme un problème d'optimisation :

$$\boxed{\begin{aligned}(w^*, b^*) = \underset{w \in \mathbb{R}^p, b \in \mathbb{R}}{\operatorname{argmin}} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y_i(\langle w, x_i \rangle + b) \geq 1, \; i = 1, \ldots, n.\end{aligned}}$$

(à suivre en M2 dans le cours "fouille de données"...)

One-Class SVM & estimation non paramétrique

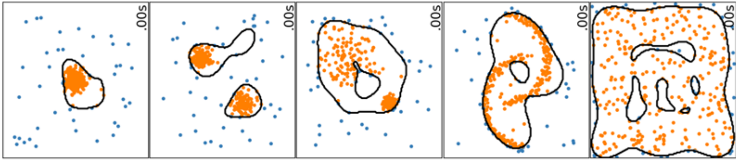
One-Class SVM : extension à un problème à 1 classe

- classe +1 : la (grande) majorité des points
- classe -1 : une (faible) proportion qu'on rejette

⇒ le modèle identifie ces deux ensembles de points

⇒ estimation non paramétrique du support de la distribution

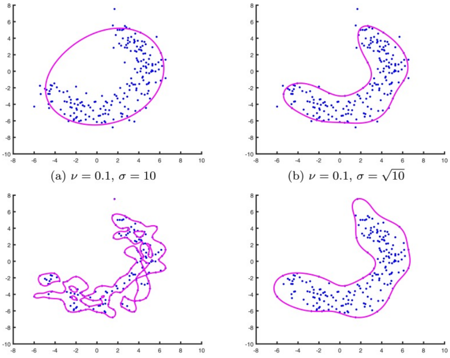
Illustration :



One-Class SVM & estimation non paramétrique

Questions / limites :

- réglage des paramètres (en général 2 : rejet + noyau)



(a) $\nu = 0.1,\ \sigma = 10$ (b) $\nu = 0.1,\ \sigma = \sqrt{10}$

(c) $\nu = 0.1,\ \sigma = 1$ (d) $\nu = 0.01,\ \sigma = \sqrt{10}$

- passage à l'échelle pour grands jeux de données

Local Outlier Factor

Principe général :

- approche "par observation"
  - vs approches précédentes - "par population"
- s'appuie sur des mesures de distances locales

$\Rightarrow$ anomalie liée à la proximité d'un point à son voisinage.

Local Outlier Factor

Principe général :

- approche "par observation"
  - vs approches précédentes - "par population"
- s'appuie sur des mesures de distances locales

$\Rightarrow$ anomalie liée à la proximité d'un point à son voisinage.

Approche naïve :

- extraire les k plus proches voisins de chaque observation
- calculer $D_k(x)$: la distance de x à son k-ième voisin

$\Rightarrow$ point atypique quand $D_k(x)$ est élevée

Local Outlier Factor

Principe général :

- approche "par observation"
  - vs approches précédentes - "par population"
- s'appuie sur des mesures de distances locales

$\Rightarrow$ anomalie liée à la proximité d'un point à son voisinage.

Approche naïve :

- extraire les k plus proches voisins de chaque observation
- calculer $D_k(x)$: la distance de x à son k-ième voisin

$\Rightarrow$ point atypique quand $D_k(x)$ est élevée

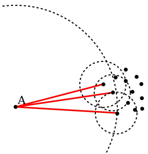
Exemple jouet :

- $k \geq 2$: deux outliers
- $k = 1$: pas d'outlier

Local Outlier Factor[2]

Local Outlier Factor :

- une extension de ce principe
- compare $D_k(x)$ à $\left\{D_k(y),\ y \in N_k(x)\right\}$

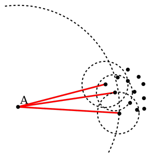


2. images tirées de Wikipedia

Local Outlier Factor [2]

Local Outlier Factor :

- une extension de ce principe
- compare $D_k(x)$ à $\left\{D_k(y),\ y \in N_k(x)\right\}$



$\Rightarrow$ s'adapte à la distribution locale des points

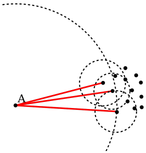
- outlier = distance à ses voisins plus grande que la distance de ses voisins à leurs voisins

2. images tirées de Wikipedia

Local Outlier Factor [2]

Local Outlier Factor :

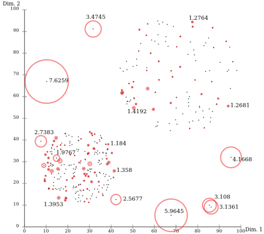
- une extension de ce principe
- compare $D_k(x)$ à $\left\{D_k(y),\ y \in N_k(x)\right\}$



$\Rightarrow$ s'adapte à la distribution locale des points

- outlier = distance à ses voisins plus grande que la distance de ses voisins à leurs voisins

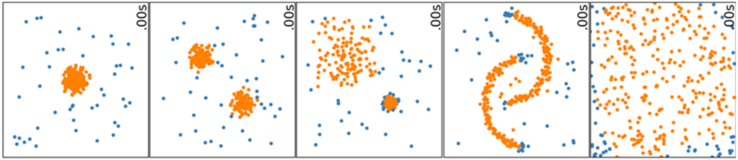
Illustration :



2. images tirées de Wikipedia

Local Outlier Factor

Illustration :



Questions / limites :

- tendance à detecter des outliers à la frontière des cluster
  - voir exemples 3 et 4
- influence du nombre de voisins
- critère numérique dur (impossible) à interpréter

Plusieurs extensions pour améliorer l'approche

- influence k et interprétabilité

Isolation Forest

Principe :

- approche "par observation"
- s'appuie sur les arbres de classification

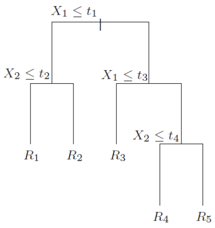
Isolation Forest

Principe :

- approche "par observation"
- s'appuie sur les arbres de classification

Arbre de classification :

- découpage récursif du jeu de données :
  - racine : tout le jeu de données
  - noeud interne :
    - un seuil sur la valeur d'une variable
    - sépare l'ensemble en deux
- critère d'arrêt :
  - feuille = population homogène
  - prédiction par classe majoritaire



(à suivre en M2 dans le cours "fouille de données"...)

Isolation Forest

Isolation tree :

- une procédure de construction aléatoire d'abre
  - variables et seuils choisis aléatoirement
- poussée jusqu'à avoir une instance par feuille
  - chaque instance est **"isolée"**

Isolation Forest

Isolation tree :

- une procédure de construction aléatoire d'abre
  - variables et seuils choisis aléatoirement
- poussée jusqu'à avoir une instance par feuille
  - chaque instance est **"isolée"**

$\Rightarrow$ outlier : probabilité plus forte d'être haut dans l'arbre

- beaucoup de découpages pour isoler des points proches

Isolation Forest

Isolation tree :

- une procédure de construction aléatoire d'abre
  - variables et seuils choisis aléatoirement
- poussée jusqu'à avoir une instance par feuille
  - chaque instance est **"isolée"**

$\Rightarrow$ outlier : probabilité plus forte d'être haut dans l'arbre

- beaucoup de découpages pour isoler des points proches

$\Rightarrow$ mesure d'anomalie : longueur du chemin dans l'arbre

- plus faible pour les points atypiques

Isolation Forest

Isolation tree :

- une procédure de construction aléatoire d'abre
  - variables et seuils choisis aléatoirement
- poussée jusqu'à avoir une instance par feuille
  - chaque instance est **"isolée"**

⇒ outlier : probabilité plus forte d'être haut dans l'arbre

- beaucoup de découpages pour isoler des points proches

⇒ mesure d'anomalie : longueur du chemin dans l'arbre

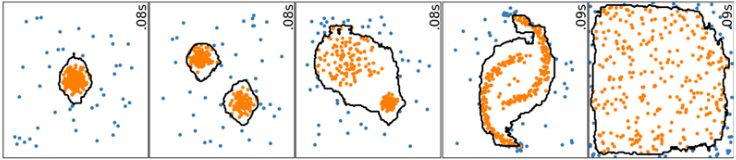
- plus faible pour les points atypiques

⇒ en pratique : moyennée sur un ensemble d'abres

- une **"isolation forest"** : plus grande robustesse

Isolation Forest

Illustration :



Remarques :

- méthode relativement récente (2008)
- efficace sur le plan calculatoire (temps et mémoire)
- applicable en haute dimension

Remarques et conclusion

Conclusion

Non-supervisé = identifier des structures/régularités :

1. estimation de densité
2. réduction de dimension
3. clustering
4. détection d'anomalie

Conclusion

Non-supervisé = identifier des structures/régularités :

1. estimation de densité
2. réduction de dimension
3. clustering
4. détection d'anomalie

Ce cours = une introduction !

- domaine vaste
- un aperçu des concepts & méthodes clés

Conclusion

Non-supervisé = identifier des structures/régularités :

1. estimation de densité
2. réduction de dimension
3. clustering
4. détection d'anomalie

Ce cours = une introduction !

- domaine vaste
- un aperçu des concepts & méthodes clés

TP : ACP, gaussiennes multivariées.

Conclusion

Non-supervisé = identifier des structures/régularités :

1. estimation de densité
2. réduction de dimension
3. clustering
4. détection d'anomalie

Ce cours = une introduction !

- domaine vaste
- un aperçu des concepts & méthodes clés

TP : ACP, gaussiennes multivariées.

La suite : méthodes de clustering.

Mise en oeuvre R

Mise en oeuvre R

Gaussiennes multivariées :

- génération : fonction `mvrnorm` du package `MASS`
- estimation : fonction `mvn` du package `mclust`

Kernel Density Estimation :

- fonction `density`

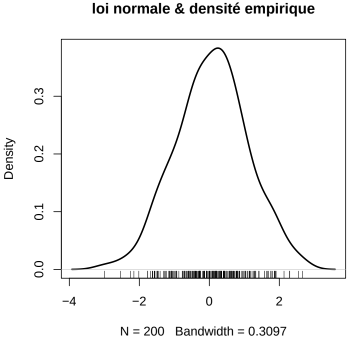
ACP :

- fonction `prcomp`
- (et sans doute d'autres)

Mise en oeuvre R

Kernel Density Estimation :

```
> n = 1000      # nombre d'échantillons
> x = rnorm(n)  # tirage selon la loi N(0,1)
> plot(density(x), main = "")
> title("loi normale & densité empirique"
> rug(x)
```



Références

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401 : 788–791, 1999.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 :2579–2605, 2008. URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.