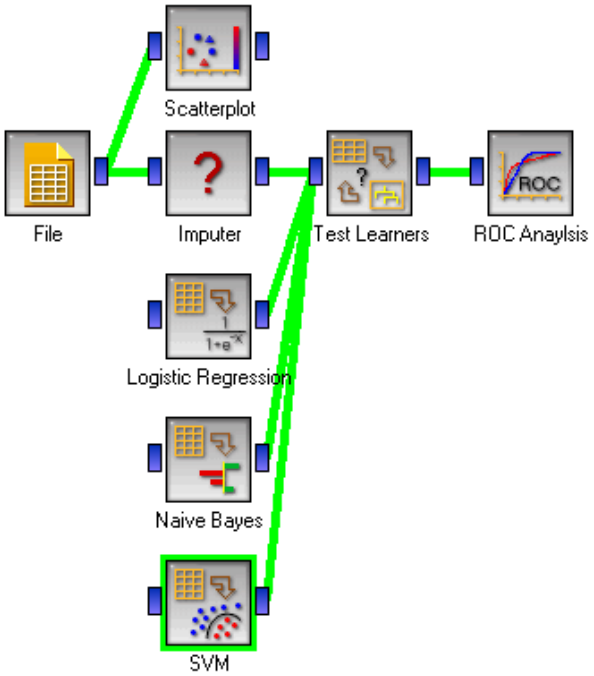
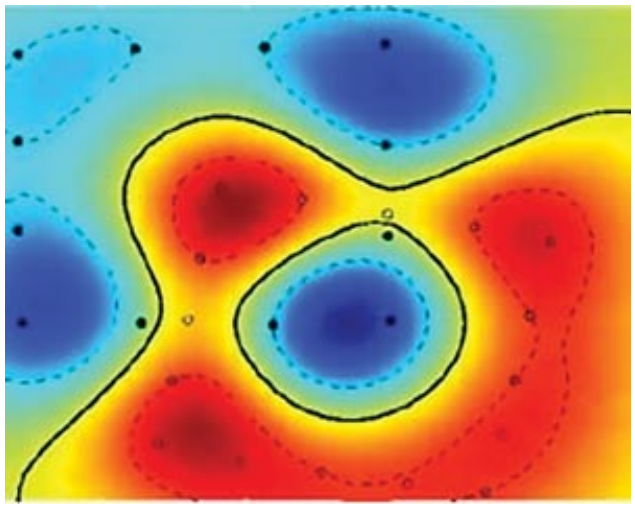
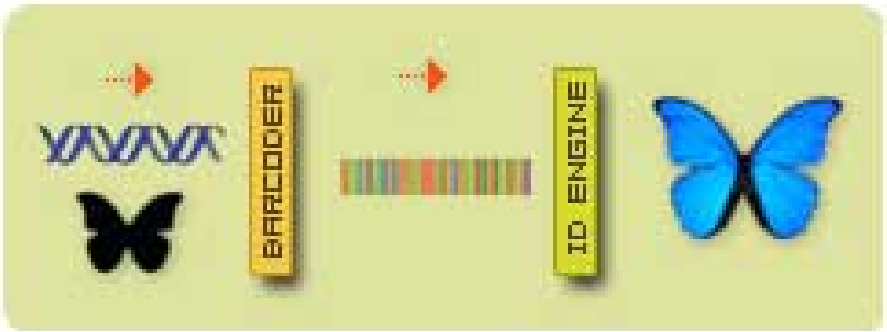
Unsupervised Learning







"Learning without a teacher"

- Observation X, but no label information Y
- Statistical model:
  $\{f(x,\theta) \ : \ \theta \in \Theta\}$, Θ (possibly) nonparametric space
- Recover (properties of) the density function based on data D_n
- Loss function:

$$\ell(x,\theta) = -\log p(x,\theta)$$

Unsupervised learning - Issues

- Modes vs. anomaly/novelty detection: find $\Gamma \subset \mathbb{R}^d$ s.t.

$$\mathbb{P}\{X \in \Gamma\} \geq \alpha \text{ and } \lambda(\Gamma) \text{ minimum}$$

- Density level set estimation $\{x \in \mathbb{R}^d : \quad f(x) = \alpha\}$
- Visualization of massive data, highlight structure
- Clustering: $K - means$, hierarchical methods, *etc.*
- Mixture estimation: $f(x) = \sum_{i=1}^{K} \omega_i f_{\theta_i}(x)$
  ex: Expectation-Maximization algorithm
- Latent variable analysis:

$$X = AZ, \quad X \in \mathbb{R}^D \text{ and } Z \in \mathbb{R}^d \text{ with } d << D$$

A mixing matrix, $Z = (Z_1, \ldots, Z_d)$ latent variables

Unsupervised learning - Applications

- Monitoring of industrial processes, complex systems
  ex: monitoring of aero engines by means of censors measuring temperature, pressure, flows, speed, vibration, *etc.*
- Marketing: segmentation of CRM databases
- Audio source separation - the "Cocktail Party" problem
- Finance: identifying the market driving factors
  ex: Capital Asset Pricing Model, Arbirtage Pricing Theory, Barra
- Visualisation - low dimensional representation of complex data - Feature extraction

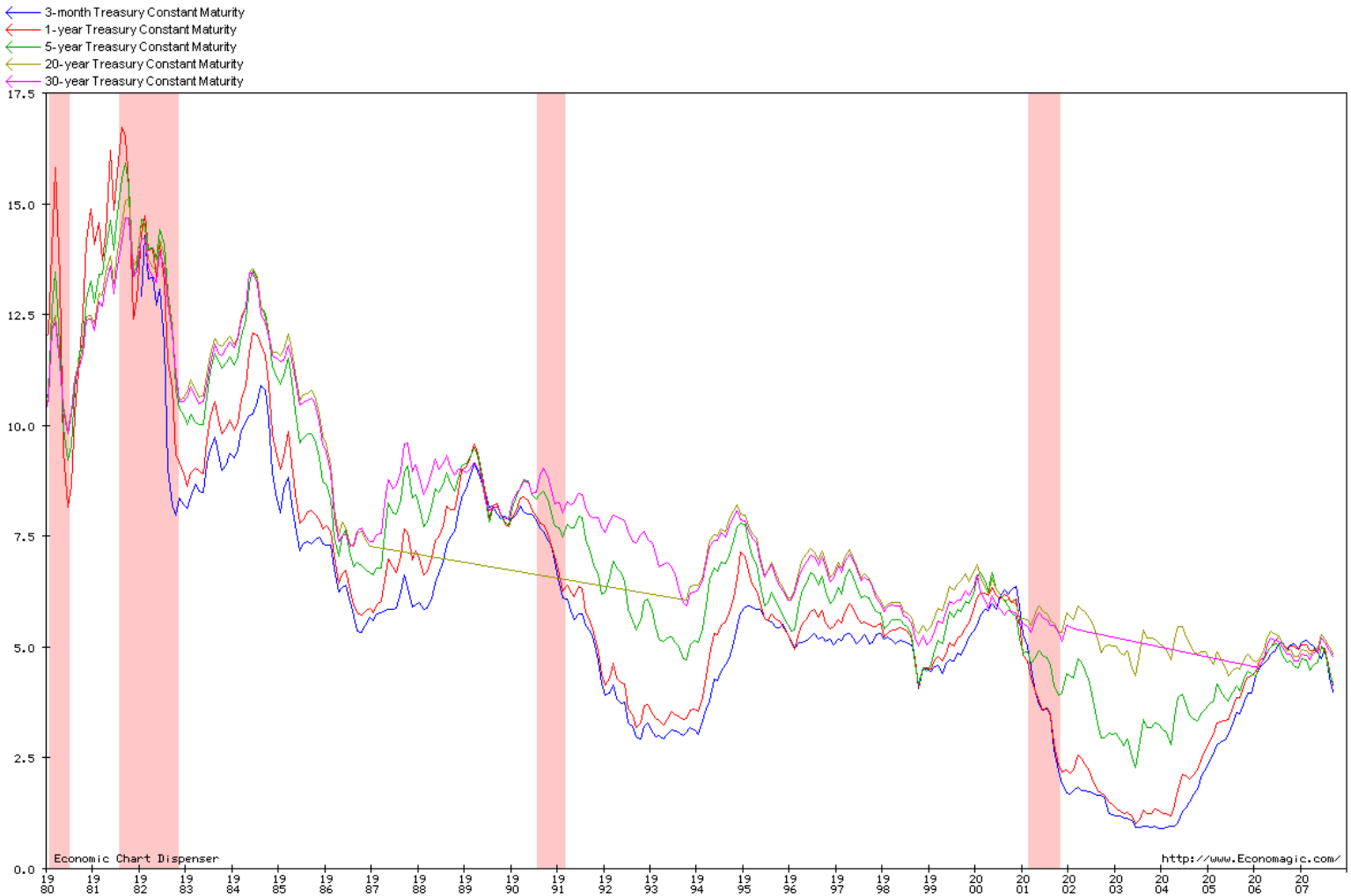
Data analysis

- Standard tools revisited
- Nonlinear PCA, kernel PCA
- Sparse PCA
- Independent Component Analysis

Réduction de la dimension

Exemple 1 - Finance

Analyse des taux d'intérêt



Exemple 1 - Finance (2)

- Variables = 18 maturités
  = 1M, 3M, 6M, 9M, 1Y, 2Y, ..., 30Y

- Observations = Historique mensuel sur 8 ans
  = 96 valeurs

Exemple 2 - Web 2.0

Last-FM - webradio de type collaboratif

Exemple 2 - Web 2.0 (2)

- 28302 artistes et leurs "tags"
- Variables = 735 tags
  = trance, techno, ambient, alternative,
  rap metal, rock, ...
- Observations = 2840 utilisateurs

Exemple 3 - Reconnaissance de visages

Exemple 3 - Reconnaissance de visages (2)

- Variables = 256 x 256 pixels
- Observations = 64 images

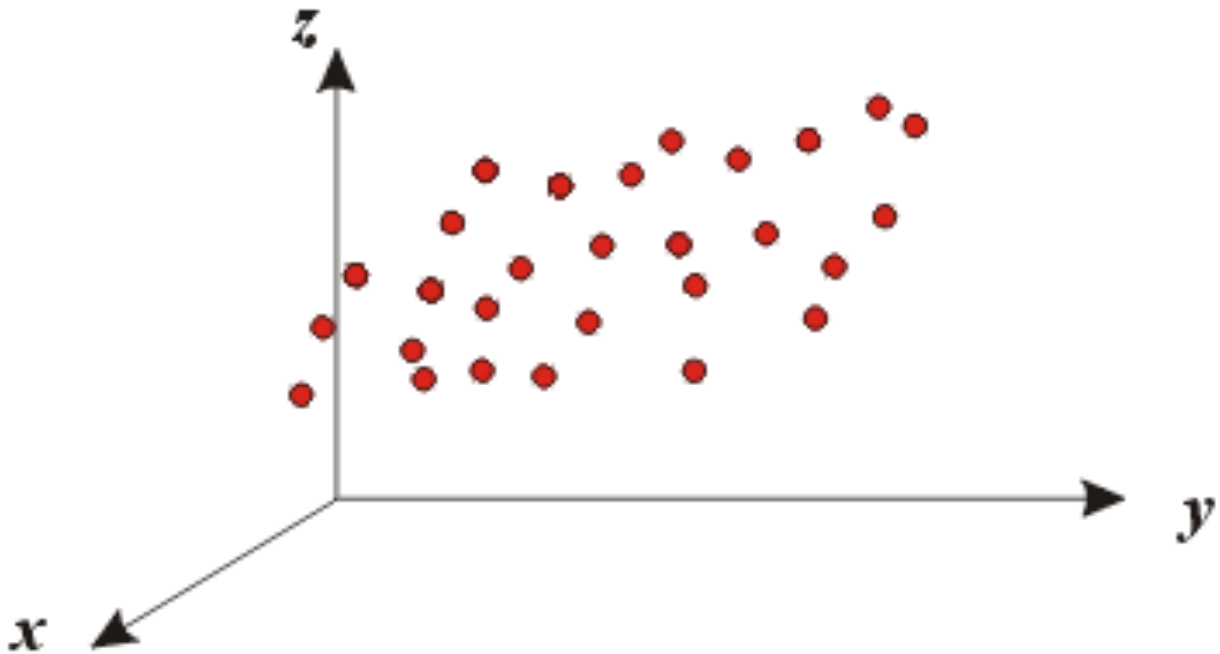
Traits communs

- Données multivariées
- Besoin d'interprétation
- Variabilité expliquée par des combinaisons de variables

Données

- Dimension = nombre de variables = p
- Taille de l'échantillon = nombre d'observations = n
- Tableau $n \times p$ de variables quantitatives

Représentation graphique



$\Rightarrow$ Nuage de n points dans $\mathbb{R}^p$

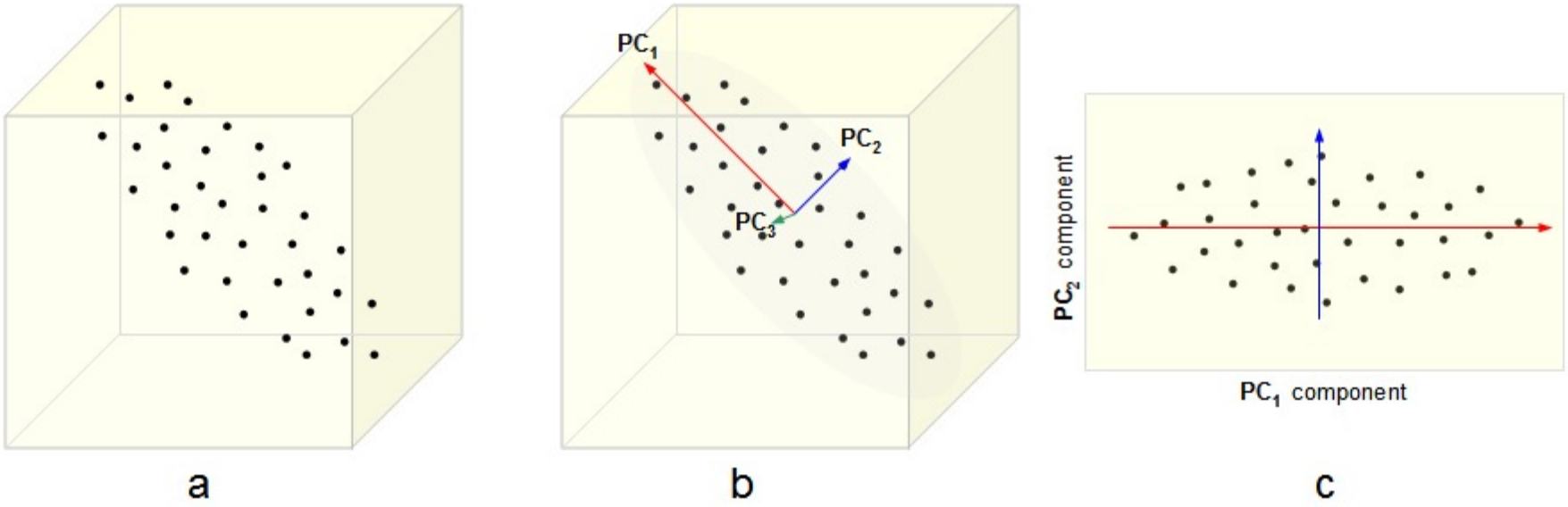
Objectifs

- Réduction de la dimension
- Visualisation du nuage en 2D ou 3D
- Explication de la variabilité

Analyse en Composantes Principales (ACP)

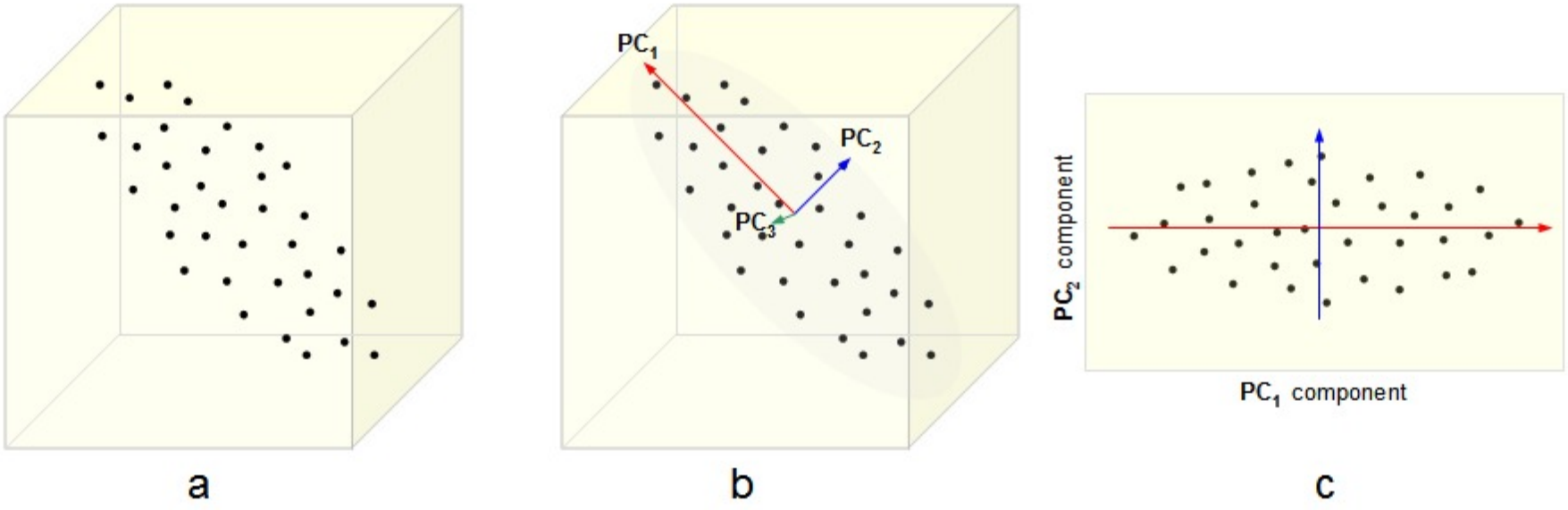
Philosophie de l'ACP

$\longrightarrow$ Projeter le nuage selon la "bonne" direction



Philosophie de l'ACP

$\longrightarrow$ Projeter le nuage selon la "bonne" direction



Idée : maximiser la dispersion

Cadre statistique : Tableau de données

- Observations : $X_i \in \mathbb{R}^p \quad , \quad 1 \leq i \leq n$
- Variable j : $X_{1j}, \ldots, X_{nj}$
- Matrice $n \times p$ de données $X = (X_1, \ldots, X_n)^T$

$$X = (X_{ij})_{i,j} = \begin{pmatrix} X_{11} & \ldots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{np} \end{pmatrix}$$

Matrice de covariance empirique

- Barycentre

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \in \mathbb{R}^p$$

- Matrice de covariance empirique $(p \times p)$

$$S = (s_{kj})_{k,j} = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \bar{X} \bar{X}^T$$

Meilleure direction

- Direction de projection $a \in \mathbb{R}^p$
- Echantillon (1D) $= (a^T X_1, \ldots, a^T X_n)$
- Maximiser la variance empirique en a :

$$s_a^2 = a^T S a$$

- Solution :

  vecteur propre $g_{(1)}$ de la plus grande valeur propre l_1

Diagonalisation de S symétrique réelle

- Valeurs propres : $l_1 \geq \ldots \geq l_p$
- Vecteurs propres orthonormés $g_{(1)}, \ldots, g_{(p)}$
- Réduction de la matrice $S = GLG^T$ où
  - $L = diag(l_1, \ldots, l_p)$ matrice diagonale $p \times p$
  - G matrice orthogonale $p \times p$

$$G = (g_{(1)}, \ldots, g_{(p)}) = (g_{kj})_{k,j}$$

Composantes principales (CP)

- Composantes principales : pour tout vecteur $z \in \mathbb{R}^p$

$$y_j(z) = g_{(j)}^T(z - \bar{X}) \ , \quad 1 \leq j \leq p$$

- La matrice $n \times p$

$$Y = (y_j(X_i))_{1 \leq i \leq n,\ 1 \leq j \leq p}$$

remplace la matrice X des données initiales.

Corrélation empirique "Variable vs. CP"

- Corrélations empiriques entre la variable k et la CP y_j :

$$\tilde{r}_{kj} = g_{kj}\sqrt{\frac{l_j}{s_{kk}}} \qquad \text{(définition)}$$

- Propriété :

$$\sum_{j=1}^{p} \tilde{r}_{kj}^2 = 1$$

Variance empirique de la k-ème variable

- Part de la variance empirique de la k-ème variable expliquée par les 2 premières CP (y_1, y_2) :

$$\tilde{r}_{k1}^2 + \tilde{r}_{k2}^2$$

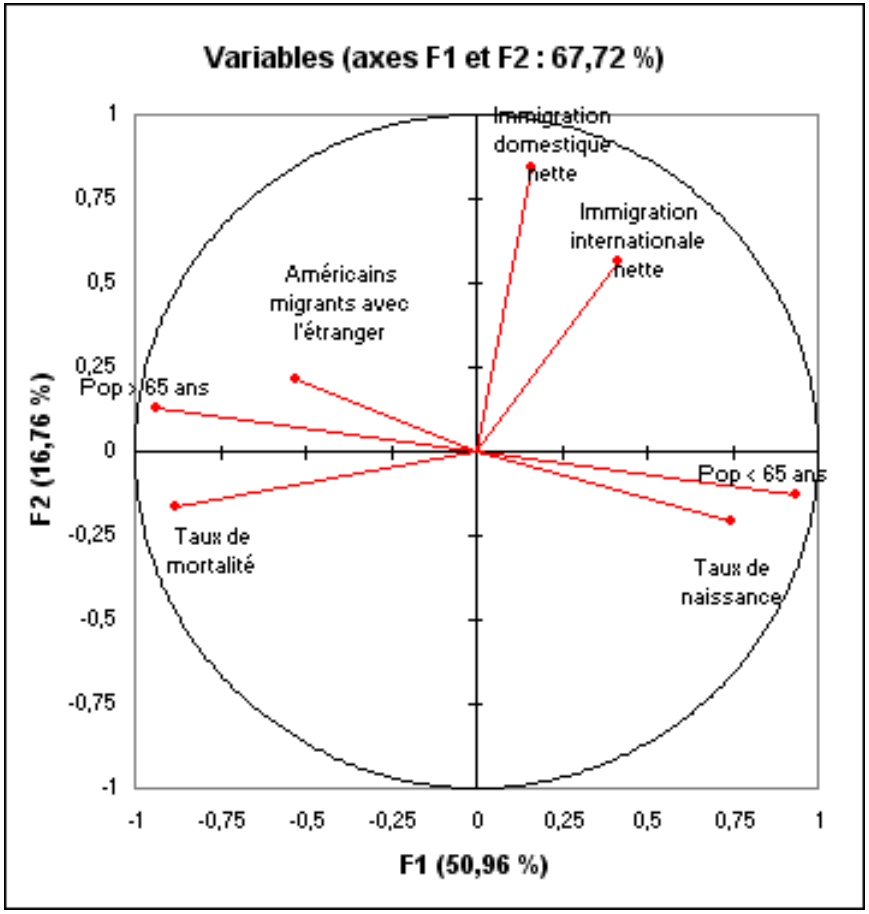
- On a :

$$I_1 + I_2 = \sum_{k=1}^{p} s_{kk}(\tilde{r}_{k1}^2 + \tilde{r}_{k2}^2)$$

- Visualisation 2D : Disque des corrélations

Disque des corrélations

- Point $(\tilde{r}_{k1}, \tilde{r}_{k2})$ correspond la variable k



Variance empirique des données

- Part de la variance empirique du nuage de points expliquée par la CP y_j:

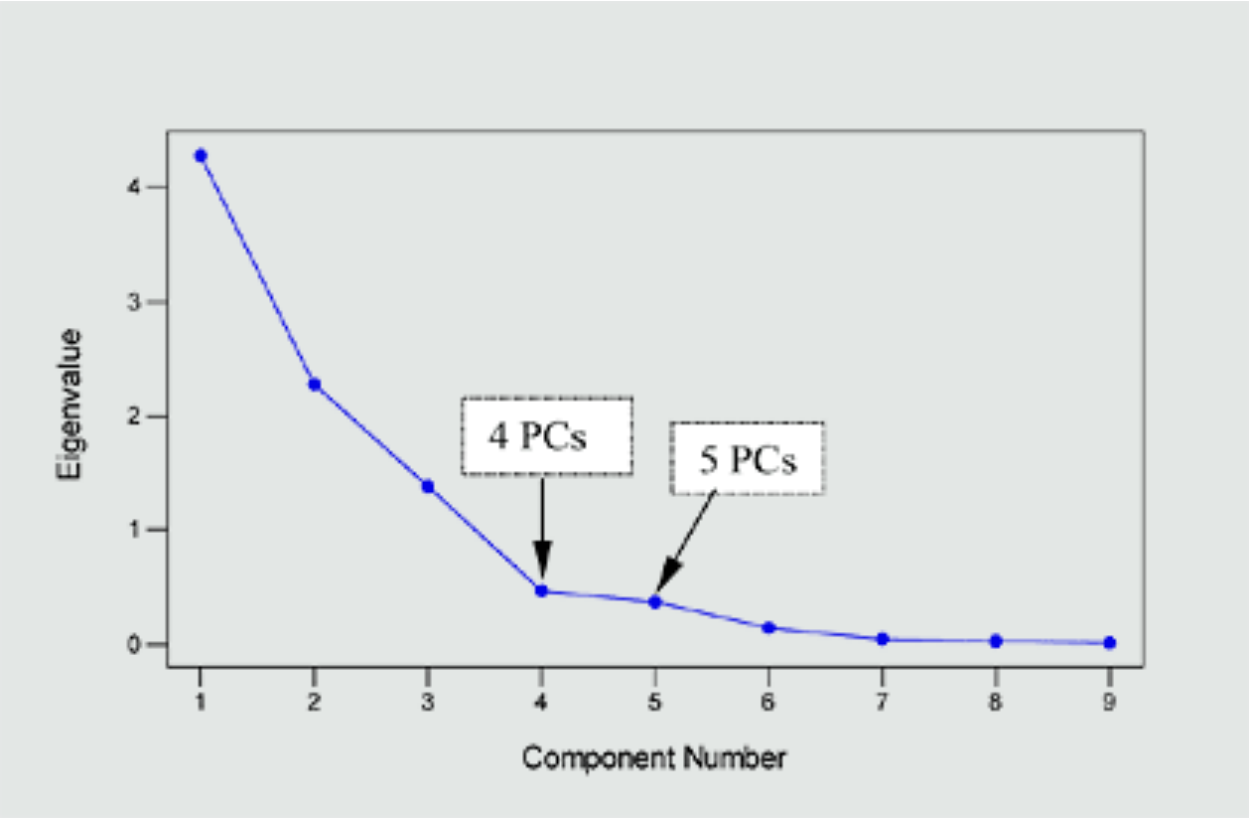
$$v_j = \frac{l_j}{Tr(S)}$$

où $Tr(S) = \sum_{j=1}^{p} l_j$.

- Visualisation : scree-graph

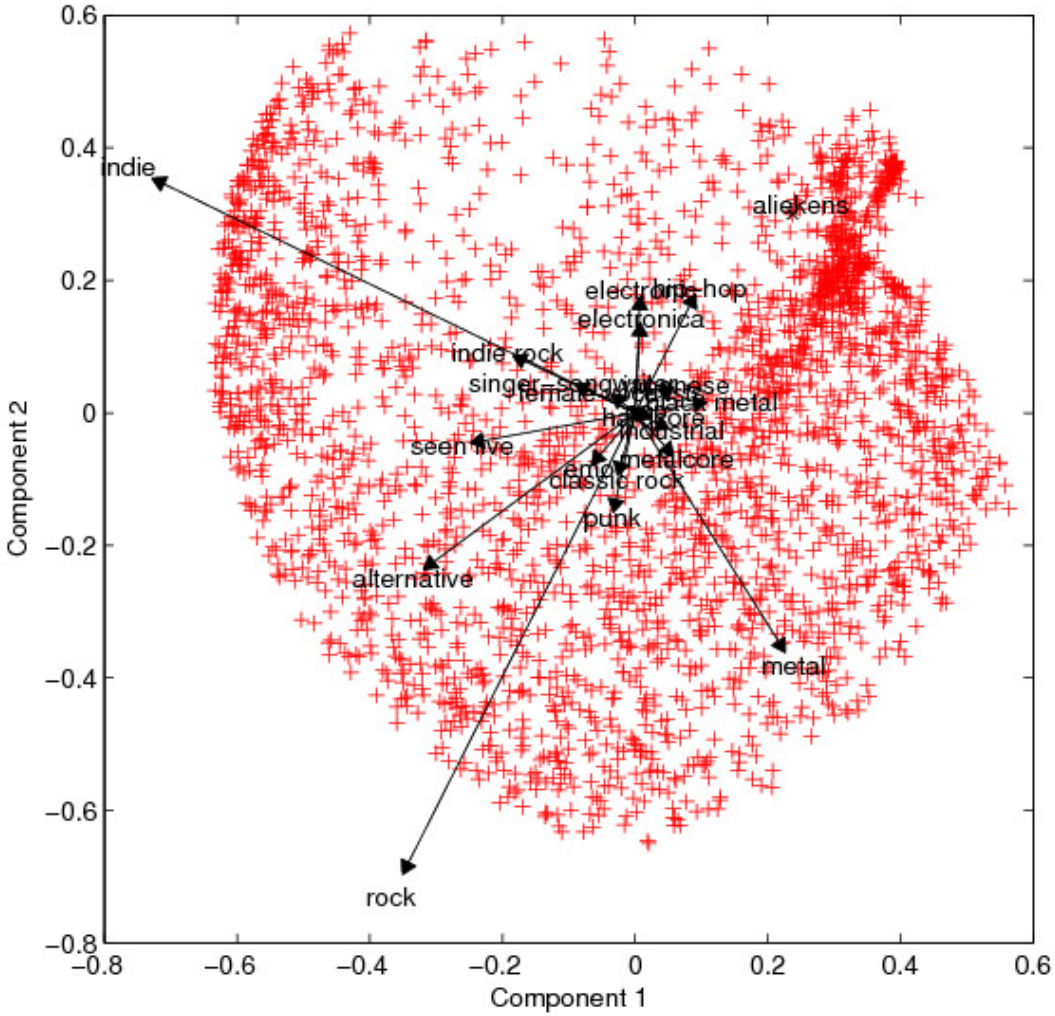
Scree-graph

- Axes = indice j de la CP et part de variance v_j



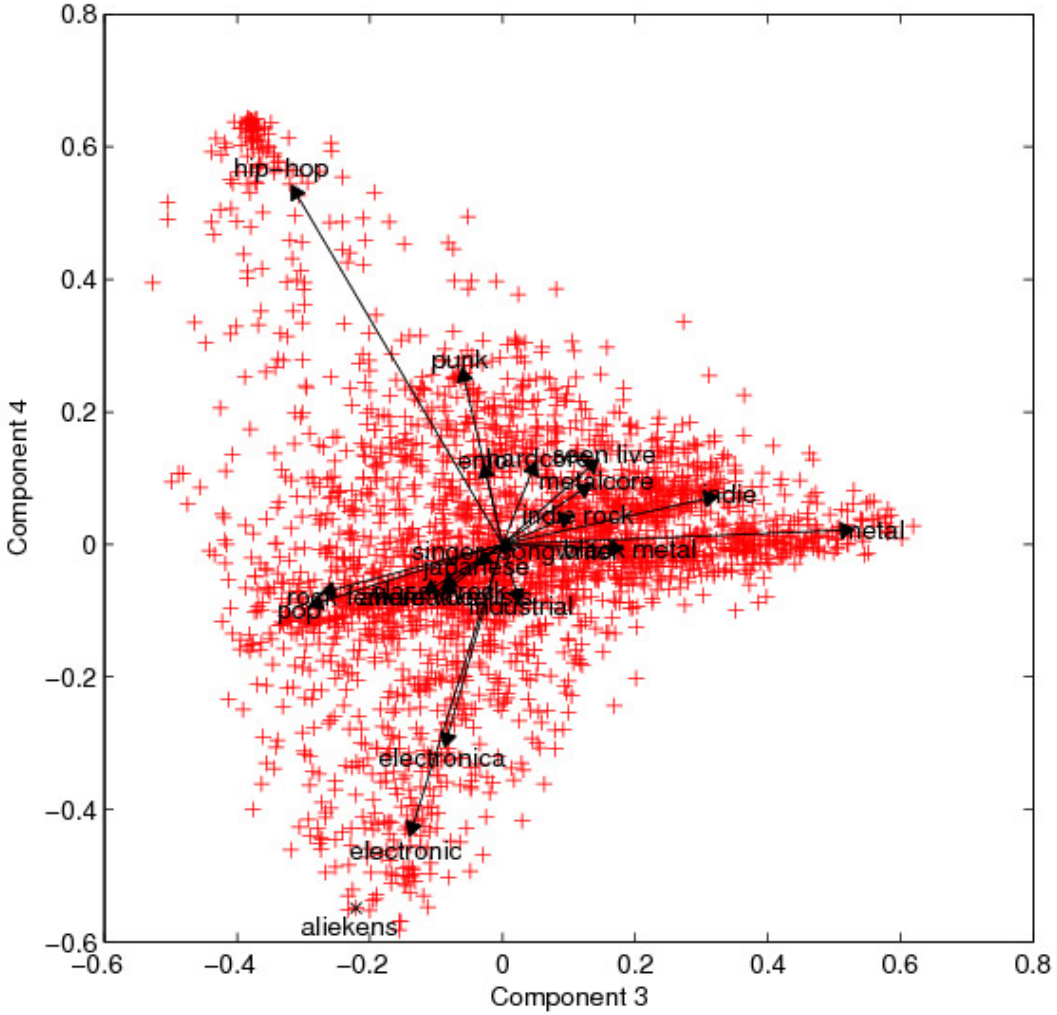
Résultats de l'ACP - Last-FM (1)

Projection du nuage de points sur (CP1, CP2)



Résultats de l'ACP - Last-FM (2)

Projection du nuage de points sur (CP3, CP4)

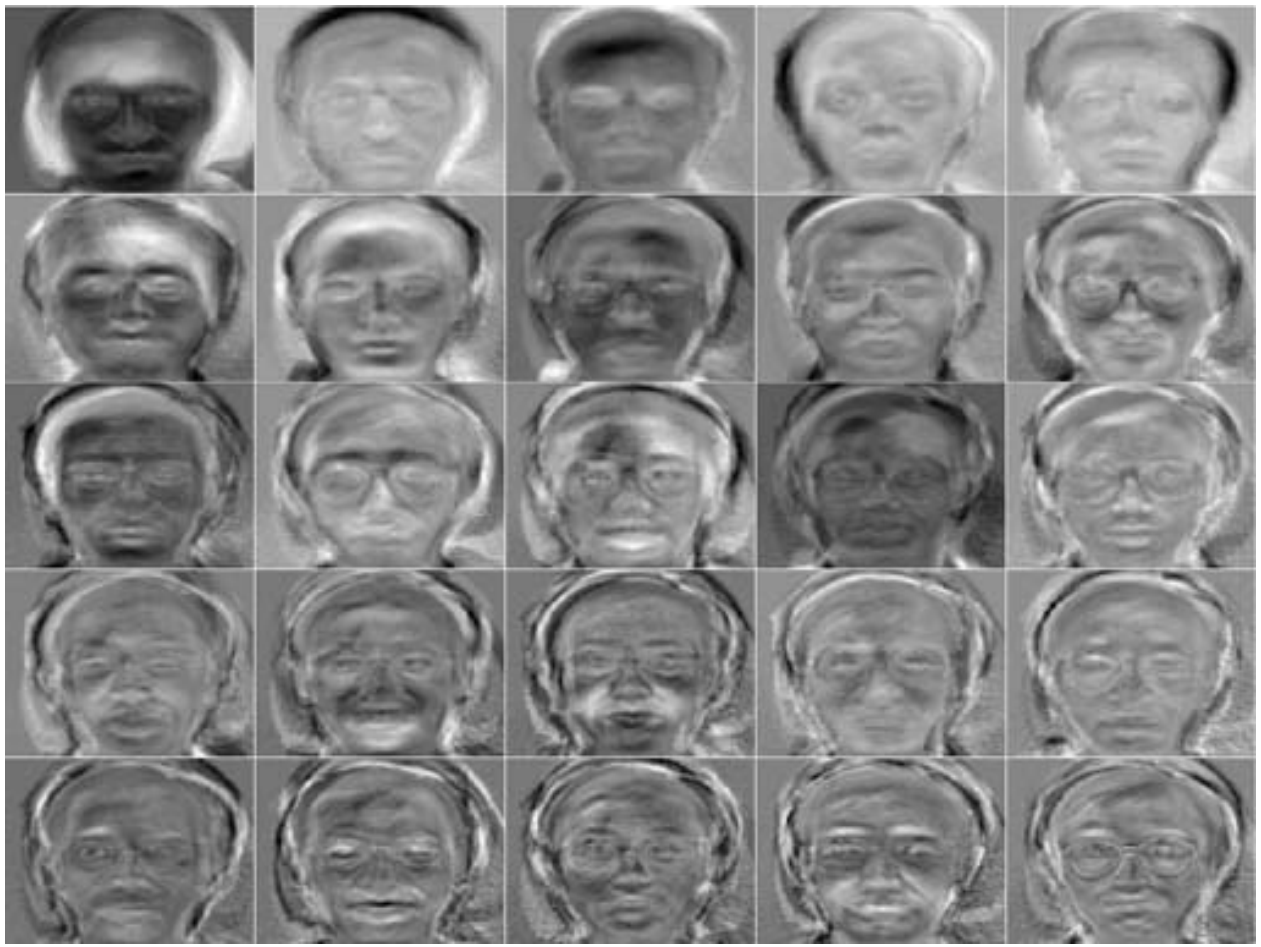


Résultats de l'ACP - Visages (1)

Données

Résultats de l'ACP - Visages (2)

"Visages propres"



Résultats de l'ACP - Visages (3)

Reconstruction partielle (sous-colonne de la matrice Y)

Résultats de l'ACP - Visages (4)

Projection d'autres images

Quelques remarques

- ACP = outil linéaire
- Orthogonalité des composantes principales
- **En pratique :**

  Réduction de la matrice $R = (r_{kj})_{k,j}$ des corrélations

$$r_{kj} = \frac{s_{kj}}{\sqrt{s_{kk} s_{jj}}}$$

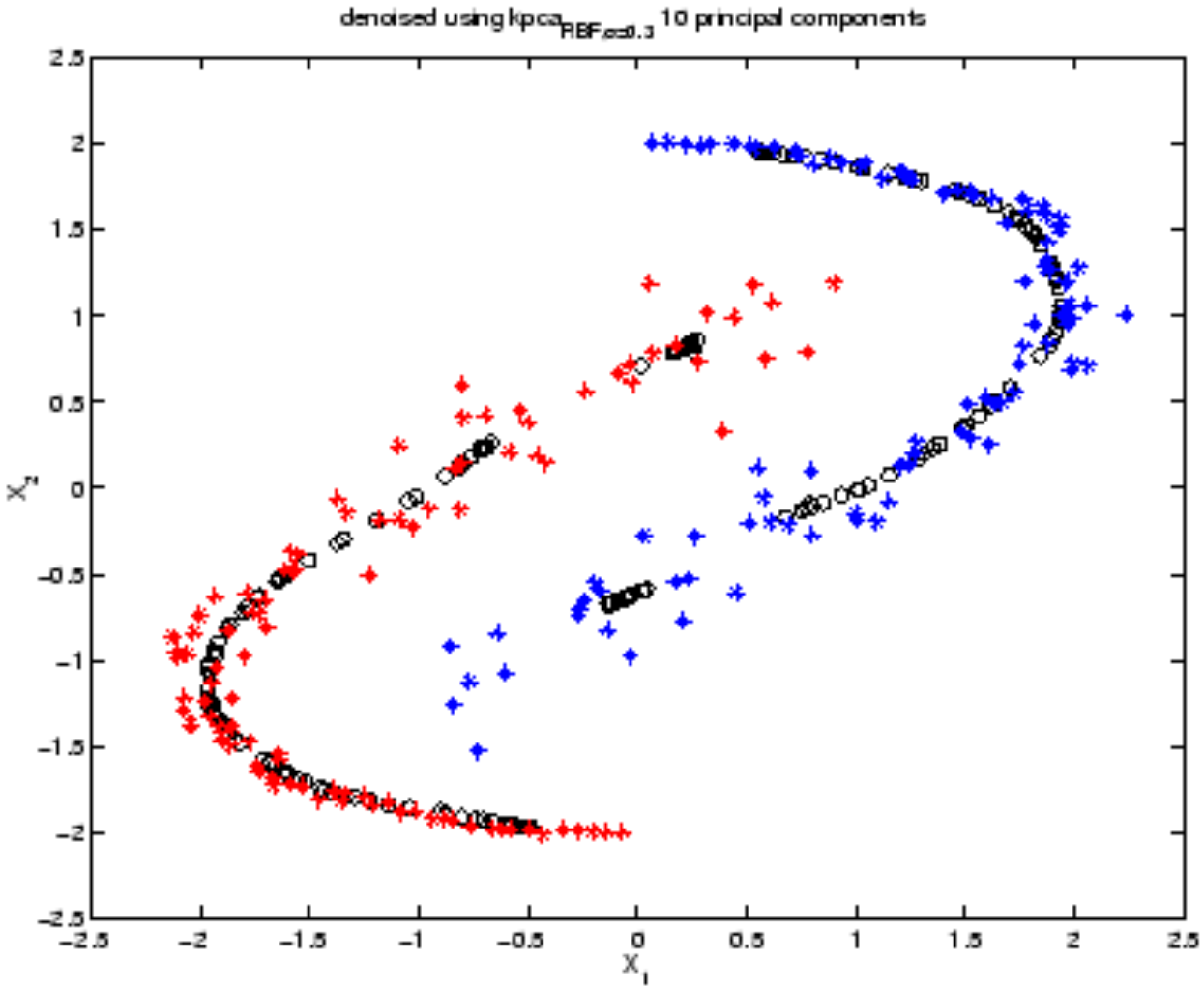
- **Obstacle numérique :**

  Réduction de S en très grande dimension

Quand est-ce que ça marche ?

- Nuages de points ellipsoïdaux
- Modèle implicite = modèle gaussien
- Information portée par les statistiques d'ordre 2
- Absence de valeurs aberrantes

Echec de l'ACP



$\Rightarrow$ **Extension :** ACP non-linéaire (à noyau)

Noyaux positifs

Définition

Soit $\mathcal{X}$ l'espace où vivent les observations.

Noyau positif

Une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est un noyau positif si et seulement si

1. k est symétrique: $k(x, x') = k(x', x)$, $\forall x, x' \in \mathcal{X}$
2. k est positive:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \quad \forall c_i \in \mathbb{R}, \quad \forall x_i \in \mathcal{X}, \quad \forall n \geq 1$$

Un théorème d'analyse

Théorème de Mercer

Pour tout noyau positif k sur $\mathcal{X}$ il existe un espace de Hilbert $\mathcal{H}$ et une application Φ tels que:

$$k(x, x') = < \Phi(x), \Phi(x') >, \quad \forall x, x' \in \mathcal{X}$$

où $< , >$ représente le produit scalaire sur $\mathcal{H}$.

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire
- $\mathcal{H}$ est un espace de représentation des données, connu sous le nom de "feature space" ou espace de caractéristiques

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire
- $\mathcal{H}$ est un espace de représentation des données, connu sous le nom de "feature space" ou espace de caractéristiques
- L'astuce du noyau consiste à faire l'impasse sur $\mathcal{H}$ et Φ si on sait qu'ils existent!

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$
- Noyau: $k(x, x') =< \Phi(x), \Phi(x') >$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$
- Noyau: $k(x, x') =< \Phi(x), \Phi(x') >$
- Distance image:

$$d_\Phi(x, x') = \|\Phi(x) - \Phi(x')\| = \sqrt{k(x, x) + k(x', x') - 2k(x, x')}$$

$\Rightarrow$ la distance induite par Φ ne fait intervenir que le noyau

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Exemple

Soit $f(x, y) = ax^2 + bx + c - y = 0$ une surface de décision polynomiale (parabole dans $\mathbb{R}^2$).

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Exemple

Soit $f(x, y) = ax^2 + bx + c - y = 0$ une surface de décision polynomiale (parabole dans $\mathbb{R}^2$).

Rôle clé de la transformation:

$$\begin{array}{rccc} \Phi & : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^4 \\ & & x & \mapsto & \left(x^2, x, 1, y\right)^T \end{array}$$

Du non-linéaire au linéaire

Exemple (suite)

On peut écrire:

$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

L'équation $g(u, v, w, y) = 0$ définit une surface de décision linéaire dans $\mathbb{R}^4$.

Du non-linéaire au linéaire

Exemple (suite)

On peut écrire:

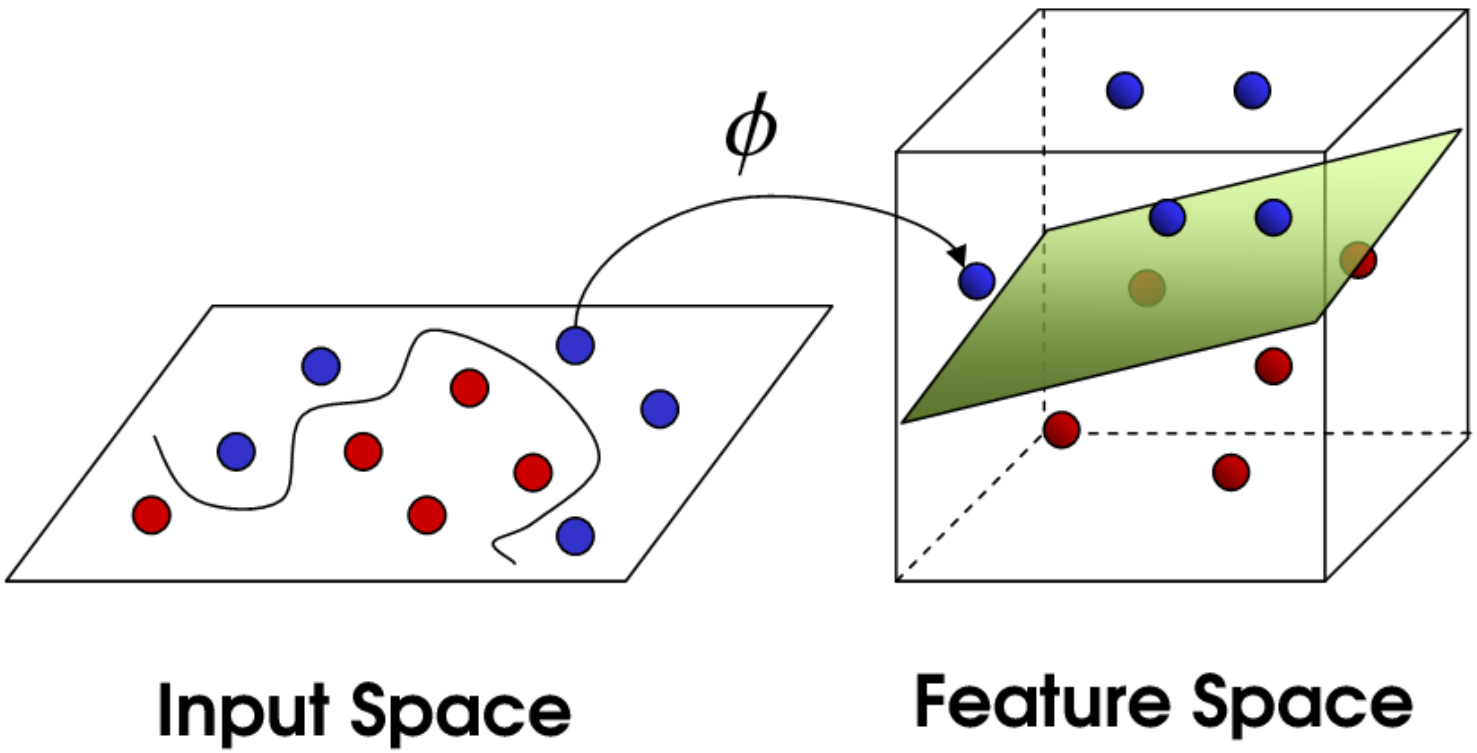
$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

L'équation $g(u, v, w, y) = 0$ définit une surface de décision linéaire dans $\mathbb{R}^4$.

Un problème non-linéaire dans un certain espace peut parfois se formuler comme un problème linéaire dans un espace plus grand.

Du non-linéaire au linéaire



ACP à noyau

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale
2. puis, la meilleure direction de projection orthogonale à la première

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

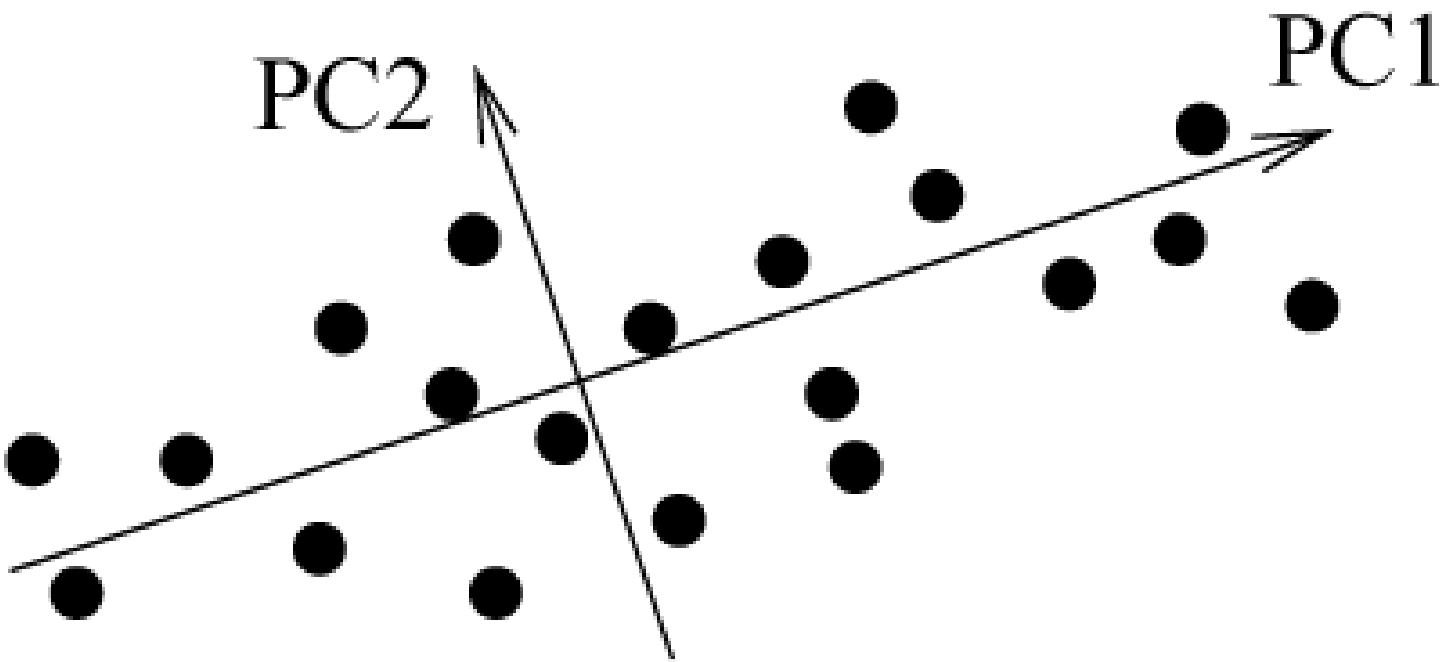
Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale
2. puis, la meilleure direction de projection orthogonale à la première
3. et, ainsi de suite, jusqu'à la n-ième

ACP



ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Matrice de covariance empirique $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Matrice de covariance empirique $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$
- On a donc :

$$\mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

Les composantes principales sont les vecteurs propres de la Σ rangés selon la décroissance des valeurs propres correspondantes.

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

Les composantes principales sont les vecteurs propres de la Σ rangés selon la décroissance des valeurs propres correspondantes.

Remarque : la matrice Σ est symétrique réelle donc diagonalisable dans une base orthonormée.

ACP (suite)

On cherche un vecteur v et un réel λ tels que:

$$\Sigma v = \lambda v$$

Or, on a :

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

D'où:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

ACP (suite)

On cherche un vecteur v et un réel λ tels que:

$$\Sigma v = \lambda v$$

Or, on a :

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

D'où:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

On utilise

$$x_j^T \Sigma v = \lambda < x_j, v >, \quad \forall j$$

et on y substitue les expressions de Σ et v:

$$\frac{1}{n} \sum_{i=1}^{n} \alpha_i \left\langle x_j, \sum_{k=1}^{n} < x_k, x_i > \ x_k \right\rangle = \lambda \sum_{i=1}^{n} \alpha_i < x_j, x_i >$$

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram
- On peut écrire alors le système:

$$K^2\alpha = n\lambda K\alpha$$

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram
- On peut écrire alors le système:

$$K^2\alpha = n\lambda K\alpha$$

- Pour résoudre en α, on résout donc le problème aux éléments propres

$$K\alpha = n\lambda\alpha$$

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires
  - les nuages de points ne sont pas tous ellipsoidaux!!

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires
  - les nuages de points ne sont pas tous ellipsoidaux!!
  - alternative : Kernel PCA

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire
- La matrice de covariance de $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ est alors

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^T$$

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire
- La matrice de covariance de $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ est alors

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T$$

- **Astuce du noyau :** $K = \left(k(x_i, x_j)\right)_{i,j} = \left(\Phi(x_i)^T \Phi(x_j)\right)_{i,j}$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- le vecteur $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ est solution du problème d'optimisation:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

sous les contraintes: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- le vecteur $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ est solution du problème d'optimisation:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

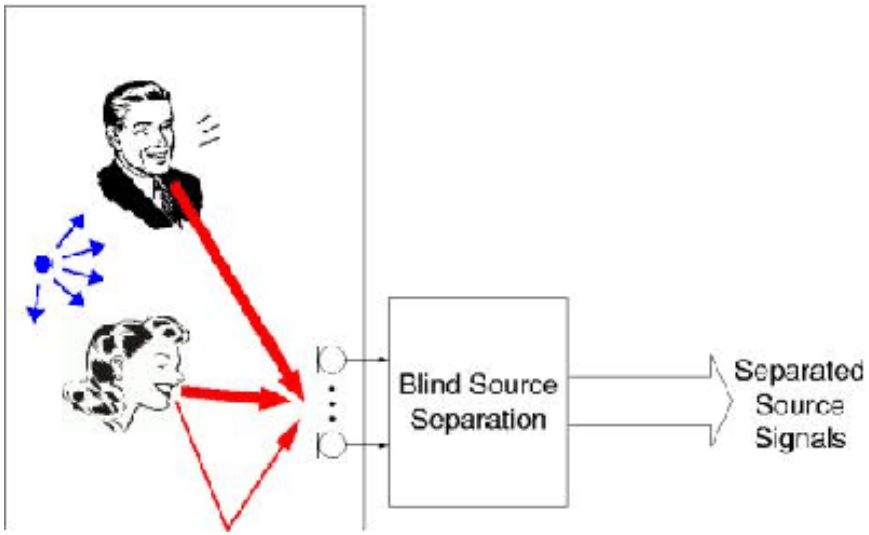
  sous les contraintes: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

- on résout le problème aux éléments propres:

$$K\alpha = n\lambda\alpha$$

Analyse en Composantes Indépendantes (ACI)

Problème du "cocktail-party"



- ACP = fondée sur la notion de corrélation
- Bonne notion = notion d'indépendance

Corrélation vs. Indépendance

- Or : X et Y indépendants $\Rightarrow \mathrm{cov}(X, Y) = 0$
- Réciproque fausse en général, sauf cas gaussien...
- De l'ACP vers l'ACI... (beaucoup plus difficile !)

Formulation du problème

- $S = (S_1, \ldots, S_d)^T$ sources indépendantes et non-gaussiennes inconnues
- $\mathbf{A}$ matrice de mélange $d \times d$ inconnue
- $X = (X_1, \ldots, X_d)^T$ observations (capteurs), on suppose $\mathrm{Cov}(X) = \mathbf{I}$
- On a le système : $X = \mathbf{A}S$
- On cherche $\mathbf{A}$ orthogonale telle que :

$$S = \mathbf{A}^T X \qquad \text{ait des composantes indépendantes}$$

Théorie de l'information

- Entropie d'une v.a. $Z \sim p(z)$:

$$H(Z) = -\mathbb{E}(\log(p(Z)))$$

- Considérons les v.a. T de variance v, alors

$$Z \sim \mathcal{N}(0,1) \quad \rightarrow \quad \max_{T} H(T)$$

- Information mutuelle pour $S = (S_1, \ldots, S_d)^T$:

$$I(S) = \sum_{i=1}^{d} H(S_i) - H(S)$$

ACI par méthode entropique

- Propriété de l'entropie : si $S = \mathbf{A}^{\mathbf{T}} X$

$$H(S) = H(X) + \log(|\det(\mathbf{A})|)$$

- On a donc le problème d'optimisation suivant :

$$\rightarrow \quad \min_{\mathbf{A}:\mathbf{A}^T\mathbf{A}=\mathbf{I}} I(\mathbf{A}^T X) = \sum_{i=1}^{d} H(S_i) - H(X)$$

- Interprétation : écart du comportement gaussien (minimisation de l'entropie des composantes)

Clustering

Goal of clustering

- Overall objective: form a partition $C_1, \ldots, C_K$ of a data sample $\{X_1, \ldots, X_n\}$ so that *"data belonging to the same group are more similar to each other than to data lying in different groups"*
- Issues:
  - *"Similar"* in which sense? Metric? Groups should correspond to *modes*, reflect *distribution structure*? How to deal with qualitative data?
  - **Combinatorial** problem: there are

$$\sum_{m=1}^{K} (-1)^{K=m} \left(\begin{array}{c} k \\ m \end{array} \right) m^n$$

    partitions with $K \leq n$ non empty groups
    How to cluster the data in practice?
  - How to choose K?

Techniques for clustering

- Very diverse methods, implemented as a preprocessing stage
- Three groups:
  - hierarchical techniques: agglomerative vs. divisive
  - (nonparametric) Bayesian methods
  - partitional: centroïds, model-based, graph theoretic, spectral clustering
- Most popular procedures:
  - $K - means$
  - Agglomerative hierarchical clustering
  - The EM-algorithm

K-means

- Input: data points in $\mathcal{X}$, distance d on $\mathcal{X}$, number K of clusters
- The clusters are defined by means of **centroïds** $c_1, \ldots, c_K$ in $\mathcal{X}$

$$x \in C_k \Leftrightarrow k = \underset{1 \leq l \leq K}{\arg\min}\, d(x, c_l)$$

- General principle:
  - start with an initial clustering,
    1. define centroïds by means of a given method (ex: cluster means, cluster medians)
    2. reassign the data to new clusters defined by proximity to centroïds
- How many iterations?

K-means

- Usually, centroïds are the current **cluster means**:

$$c_k = \frac{1}{\#\{i : X_i \in C_k\}} \sum_{i:\, X_i \in C_k} X_i$$

- When the metric is the **square Euclidean distance**, the goal is to minimize over $c_1, \ldots, c_K$

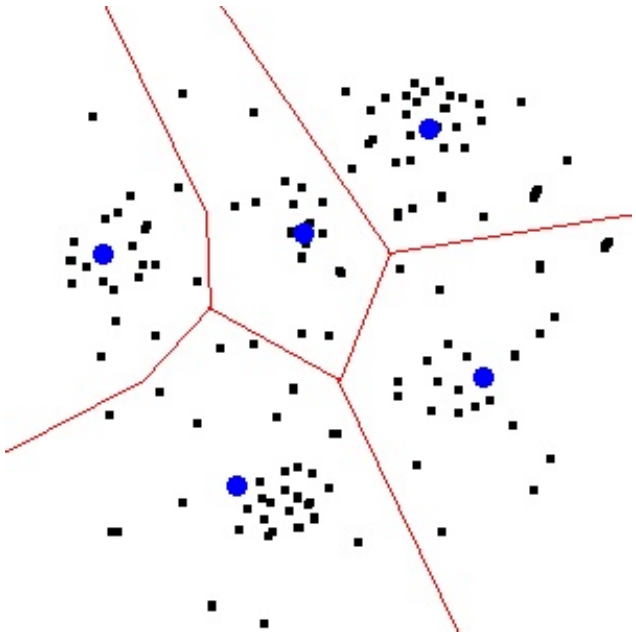
$$\sum_{k=1}^{K} \sum_{i:\, X_i \in C_k} ||X_i - c_k||^2$$

- Minimizing intra-cell variability is equivalent to maximizing inter-cell variability

$$\sum_{(i,j)} ||X_i - X_j||^2 = \sum_{k \neq l} \sum_{(i,j):\, (X_i, X_j) \in C_k \times C_l} ||X_i - X_j||^2 + \sum_{k} \sum_{(i,j):\, (X_i, X_j) \in C_k^2} ||X_i - X_j||^2$$

K-means

- Monotonicity: the within-cluster variability **decreases**
- Convergence to a **possibly local** minimum
- Use the R function $\textsc{kmeans}(.)$

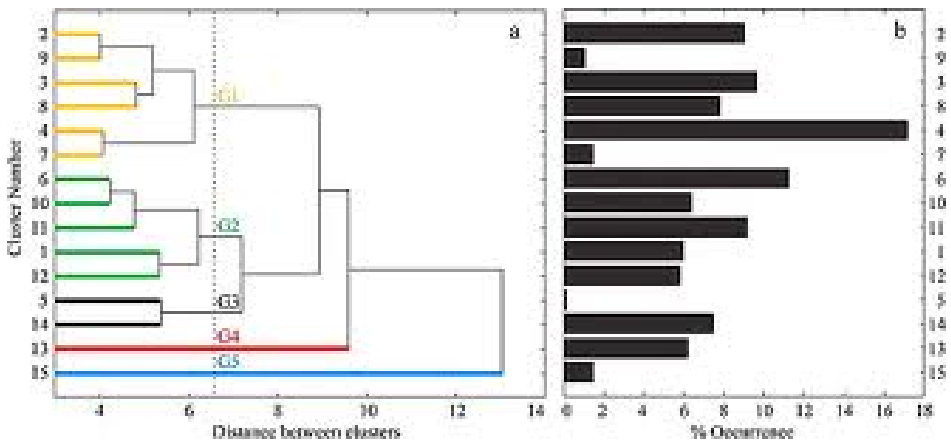


Hierarchical clustering

- Produce a **nested sequence** of clusterings
- The sequence can be represented by a **tree schematic** (dendrogram)
- Either **agglomerative** or else **divisive**
- No need to specify the number K of clusters in advance

Agglomerative hierarchical clustering

- Initially, start with n clusters: the singletons $\{X_i\}$
- Merge the pair of singletons $\{X_i\}$ and $\{X_j\}$ with minimum dissimilarity, yielding $n - 1$ clusters
- Iterate: merge the two clusters with minimum dissimilarity
- ...
- Stop when all points have been agglomerated into a single cluster of cardinality n



Agglomerative hierarchical clustering - How to measure dissimilarity between clusters

- "Single linkage"

$$D(C, C') = \min_{(x,x') \in C \times C'} d(x, x')$$

- "Complete linkage"

$$D(C, C') = \max_{(x,x') \in C \times C'} d(x, x')$$

- "Centroïd linkage"

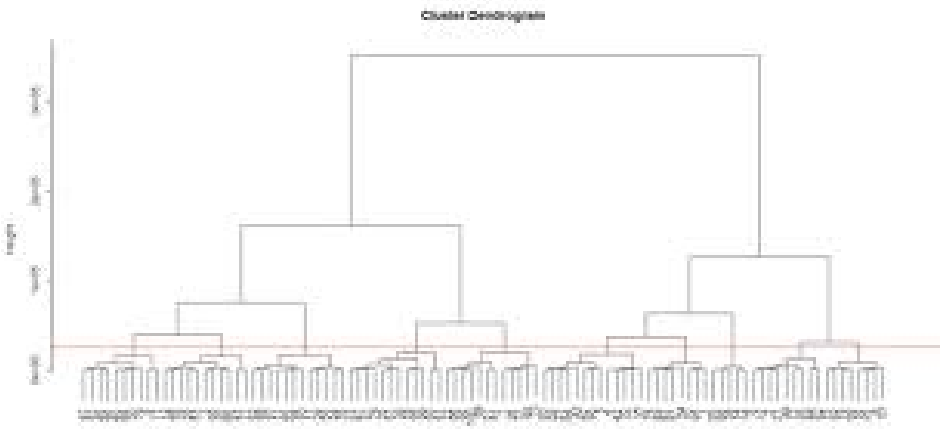
$$D(C, C') = d(\bar{x}_C, \bar{x}_{C'})$$

- "Average linkage"

$$D(C, C') = \frac{1}{\#C \#C'} \sum_{(x,x') \in C \times C'} d(x, x')$$

Divisive hierarchical clustering

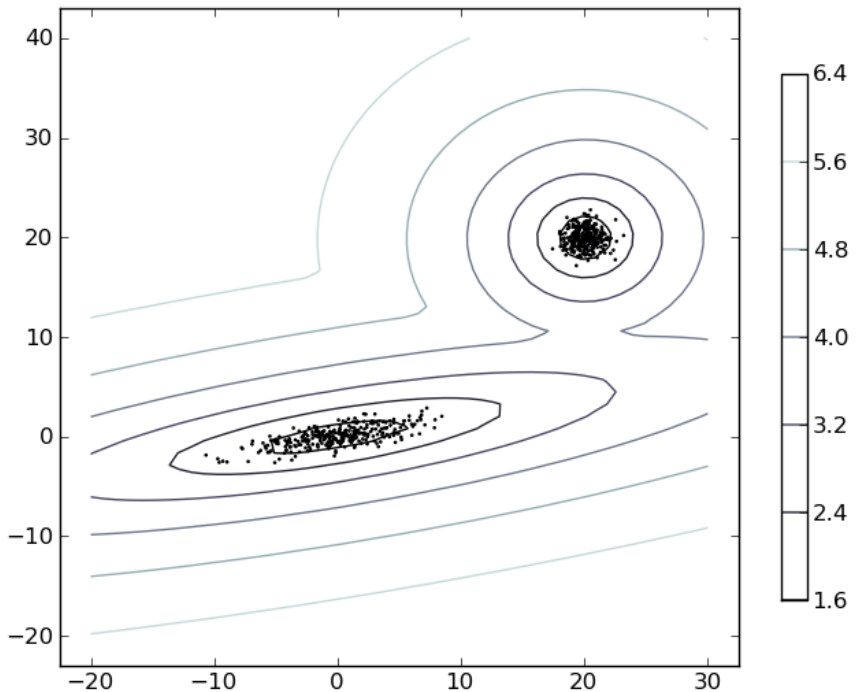
- Start with a single cluster of cardinality n and fix a threshold λ
- Determine the pair (X_i, X_j) with maximum dissimilarity $d_{\max}$
- Compare $d_{\max}$ and t.
  If $d_{\max} < \lambda$, then stops.
  If $d_{\max} > \lambda$, form two clusters: assign X_m to X_i's cluster if $d(X_i, X_m) < d(X_j, X_m)$, to X_j's cluster otherwise
- ...



Model-based clustering

- **Mixture** density model: $f_\theta(x) = \sum_{k=1}^{K} \omega_k f_{\theta_k}(x)$
  $\omega_k \geq 0$, $\sum_{k=1}^{K} \omega_k = 1$, $\theta = ((\theta_1, \omega_1), \ldots, (\theta_K, \omega_K))$
- Consider Y, **"hidden" class label**:

$$X \mid Y \sim f_{\theta_Y}(x)dx \text{ and } \omega_k = \mathbb{P}\{Y = k\}$$



Model-based clustering - The EM algorithm

- Goal: find a (local) maximum for the log-likelihood

$$L(\theta, \mathbf{X}^{(n)}) = \mathbb{E}_\theta \left[\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \mathbb{I}\{Y_i = k\} f_{\theta_k}(X_i) \right) \mid \mathbf{X}^{(n)} \right]$$

- Initialization: start with a guess $\widehat{\theta}^{(0)}$
- Iterations:
  1. E-step: compute $Q(\theta', \widehat{\theta}^{(j)}) = \mathbb{E}_{\widehat{\theta}^{(j)}}[L(\theta', \mathbf{X}^{(n)}) \mid \mathbf{X}^{(n)}]$ for any θ'
  2. M-step: find $\widehat{\theta}^{(j+1)} = \arg\max_{\theta'} Q(\theta', \widehat{\theta}^{(j)})$
- stops when $\widehat{\theta}^{(j+1)} - \widehat{\theta}^{(j)}$ becomes negligible
- The EM-algorithm increases the log-likelihood

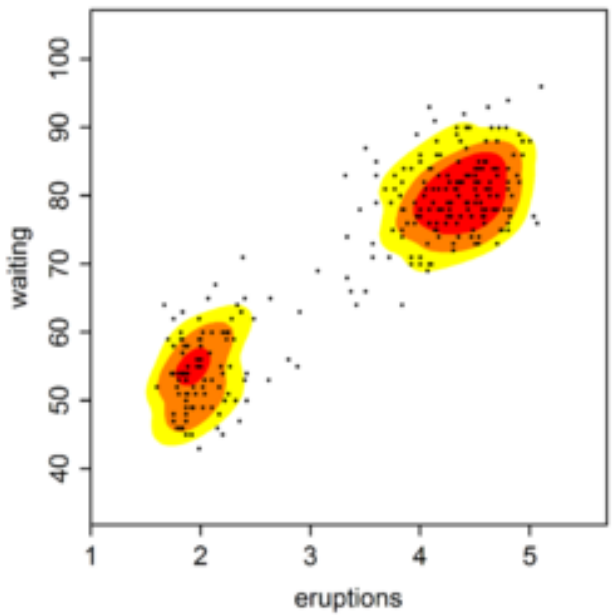
Novelty detection - the MV set approach

- Minimize the volume $\mu(G)$ over $G \in \mathcal{G}$ s.t. $\mathbb{P}\{X \in G\} \geq 1 - \alpha$
- ERM approach: replace $R(G) = \mathbb{P}\{X \in G\}$ by its empirical version

$$\widehat{R}_n(G) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{X_i \in G\}$$

and add a tolerance level Φ of the order of

$$\sup_{G \in \mathcal{G}} |\widehat{R}_n(G) - R(G)|$$



One-class SVM

- SVM rule in order to detect *outliers* in a dataset
- create a spherical decision boundary around a set of data points by means of SV
- Formulation:

$$\text{minimize } \frac{1}{2}||w||^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$

subject to

$$< \phi(X_i), w > + b \geq \rho - \xi_i, \quad \xi_i \geq 0, \text{ for } 1 \leq i \leq n$$

- The parameter ν controls the volume of the sphere (*i.e.* the number of oultiers)