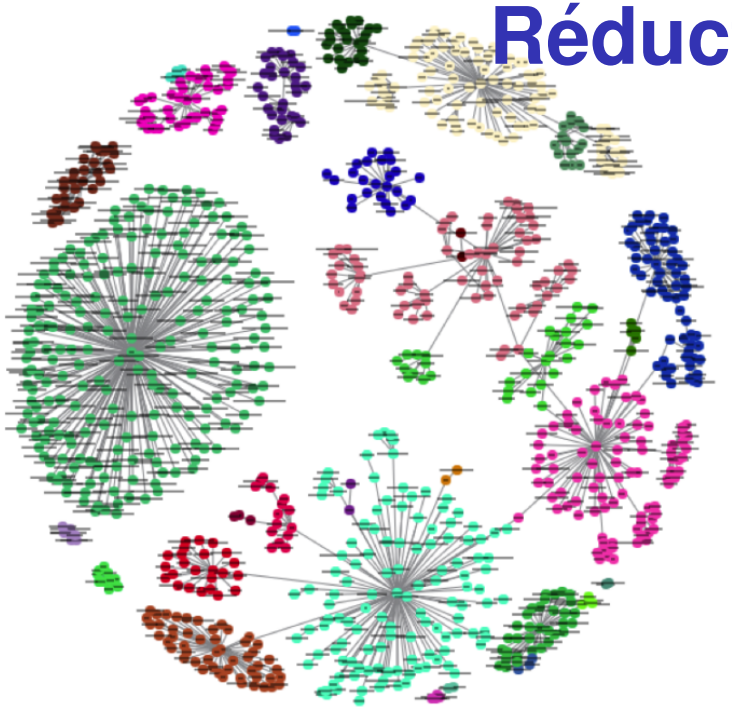
Apprentissage non supervisé
Réduction de dimension



Cours 7
ML
Master DAC

Nicolas Baskiotis
nicolas.baskiotis@sorbonne-universite.fr

équipe MLIA, Institut des Systèmes Intelligents et de Robotique (ISIR)
Sorbonne Université

S2 (2023-2024)

Plan

Que faire sans label de disponible . . .

Pourquoi et quand ?

- pas le temps ni l'argent
- pas de spécialiste pour étiquetter
- impossible à étiquetter
- évolution dynamique des structures
- trop de catégories sans beaucoup de sens
- l'important est la structuration des données, les motifs
- on ne sait pas ce qu'on cherche
- . . .

Principe

- Regrouper tout ce qui se ressemble,
- Eloigner tout ce qui est franchement différent.
- Un *cluster* : un regroupement de donnée.

Simple, mais . . .

Données

Clustering

Principe

- Regrouper tout ce qui se ressemble,
- Eloigner tout ce qui est franchement différent.
- Un *cluster* : un regroupement de donnée.

Simple, mais . . .

Données

Clustering

Principe

- Regrouper tout ce qui se ressemble,
- Eloigner tout ce qui est franchement différent.
- Un *cluster* : un regroupement de donnée.

Simple, mais . . .

Données

Clustering

Principe

- Regrouper tout ce qui se ressemble,
- Eloigner tout ce qui est franchement différent.
- Un *cluster* : un regroupement de donnée.

Simple, mais . . .

Données

Clustering

Principe

- Regrouper tout ce qui se ressemble,
- Eloigner tout ce qui est franchement différent.
- Un *cluster* : un regroupement de donnée.

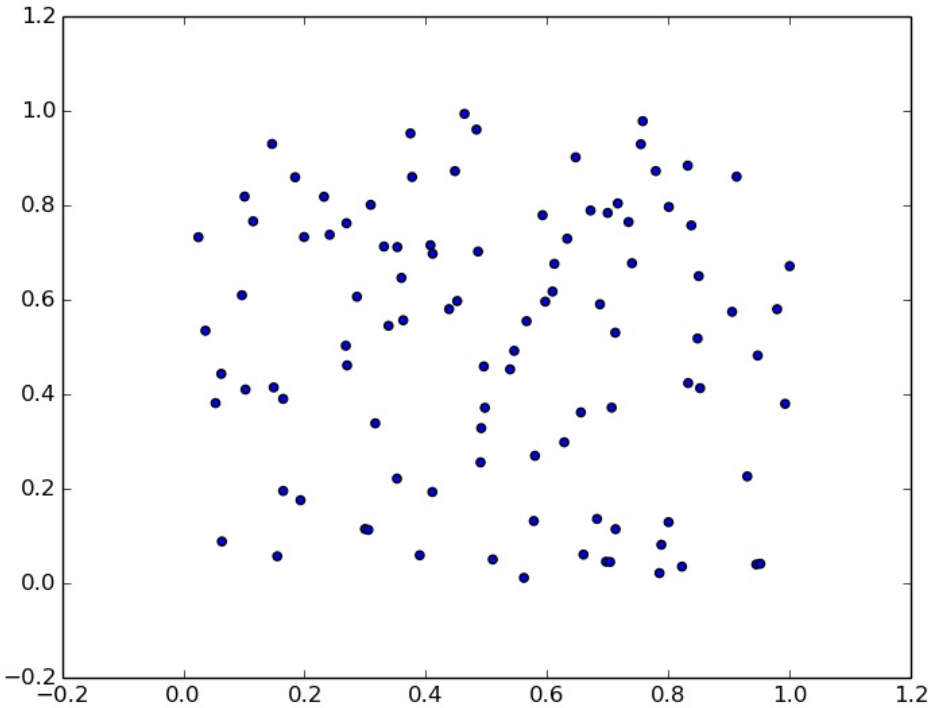
Simple, mais . . .

Données

Clustering

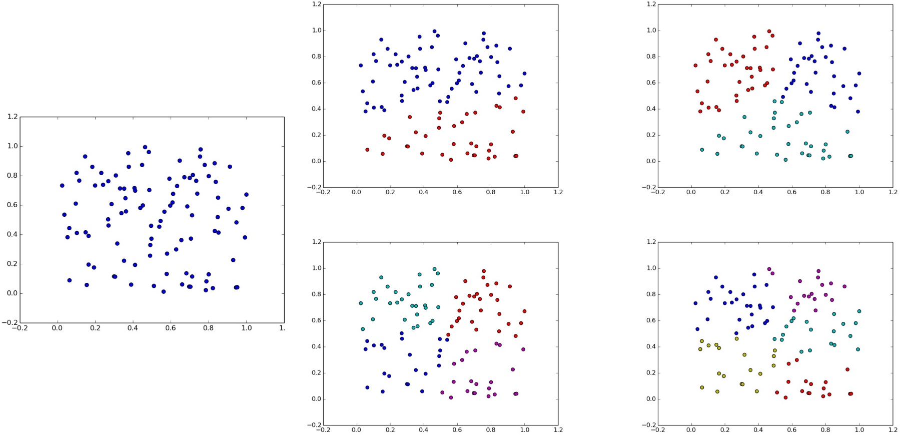
- L'apprentissage non supervisé : très subjectif !
- Pas de but global bien défini, l'objectif est induit par la formulation du problème.

Exemple



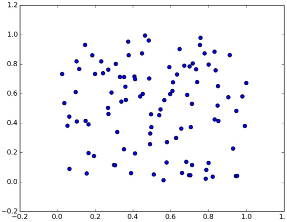
Exemple

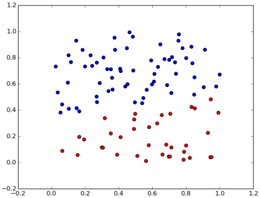
Quel est le bon clustering ?

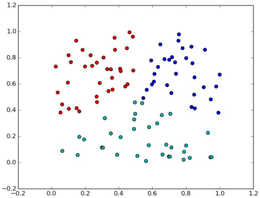


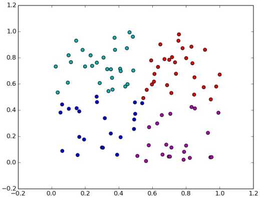
Exemple

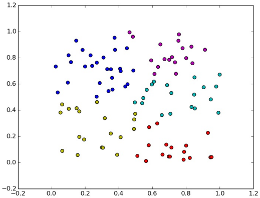
Quel est le bon clustering ?











aucun ! (distribution uniforme)

Apprentissage non supervisé

- Ensemble très varié de techniques qui visent à trouver des sous-ensembles cohérents des données
- Tout ce qui ressemble s'assemble $\Rightarrow$ définir une *similarité* entre exemples
- Deux principales approches :
  - par partitionnement
  - par modélisation
- Un clustering peut être:
  - *hard* (un exemple n'appartient qu'à un groupe)
  - *soft* (probabilité d'appartenance)

$\Rightarrow$ Domaine-spécifique, pas de règle générale, tout dépend de la tâche !

Formalisation et évaluation

Objectif

- Soit $D = x^1, \ldots, x^N \in \mathcal{X}$ un jeu de données
- Construire une projection $f : \mathbb{R}^d \rightarrow \{1, \ldots, K\}$ telle que à chaque x^i un cluster j est associé
- Le choix de K est difficile !

Comment évaluer ?

- Similarité: mesure quantitative de la similarité dans un cluster (distance intra-cluster) comparée aux autres clusters (distance extra-cluster)
- Variance: stabilité des résultats (sous-échantillonage, bruit artificiel . . .)
- Connaissances expertes: expertise qualitative sur la signification des clusters

Plan

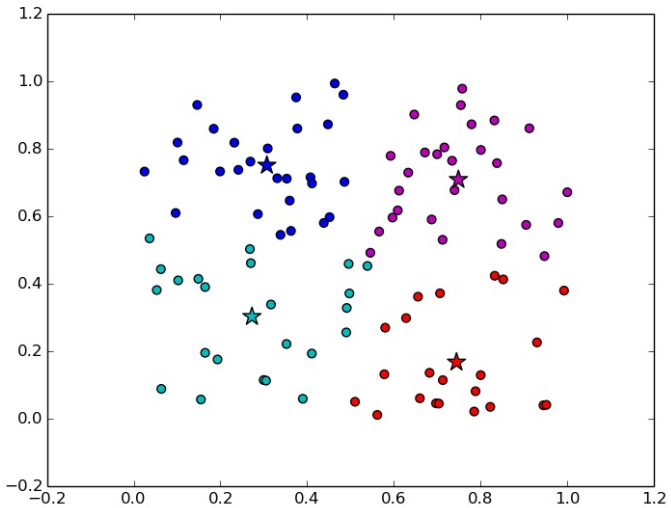
Clustering par partitionnement : k-means

Objectif

- Clustering des données pour minimiser la distance intra-cluster
- Etant donnée $d : X \times X \to \mathbb{R}$ une distance
- Minimiser :

$$argmin_{\pi=(D_1,\cdots,D_K)} \sum_{i=1}^{K} \sum_{x_j \in D_i} d(x_j, \mu_i)$$

- avec μ_i le centroïde du cluster i, i.e. $\mu_i = \frac{1}{|D_i|} \sum_{x_j \in D_i} x_j$



Clustering par partitionnement : k-means

Objectif

- Clustering des données pour minimiser la distance intra-cluster
- Etant donnée $d : X \times X \rightarrow \mathbb{R}$ une distance
- Minimiser :

$$argmin_{\pi=(D_1,\cdots,D_K)} \sum_{i=1}^{K} \sum_{x_j \in D_i} d(x_j, \mu_i)$$

- avec μ_i le centroïde du cluster i, i.e. $\mu_i = \frac{1}{|D_i|} \sum_{x_j \in D_i} x_j$

Remarques

- NP-difficile (très difficile)
- Chaque centroide est la quantization d'un cluster : *prototype*
- Très similaire à la notion de compression

Algorithme des K-means

Algorithme en deux étapes

Initialiser avec un clustering aléatoire

1. Mise à jour des centroïdes: $\mu_i = \frac{1}{|D_i|} \sum_{x_j \in D_i} x_j$
2. Assigner à chaque exemple x_j le cluster le plus proche $(argmin_{i \in \{1,\ldots,K\}} d(x_j, \mu_i))$

Répéter le deux étapes jusqu'à stabilité

Initialisation

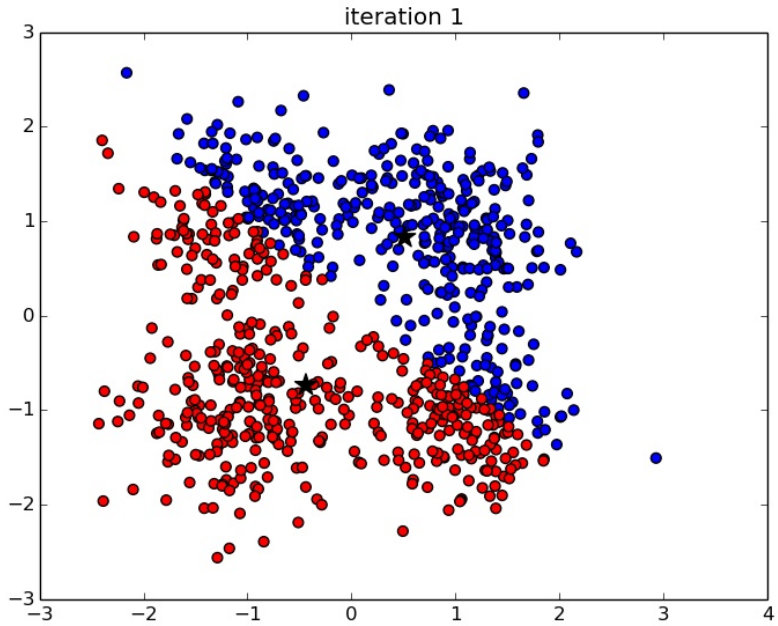
- Résultats très sensibles à l'initialisation
- Converge souvent vers un minimum local $\Rightarrow$ multiples tentatives et prendre la meilleure

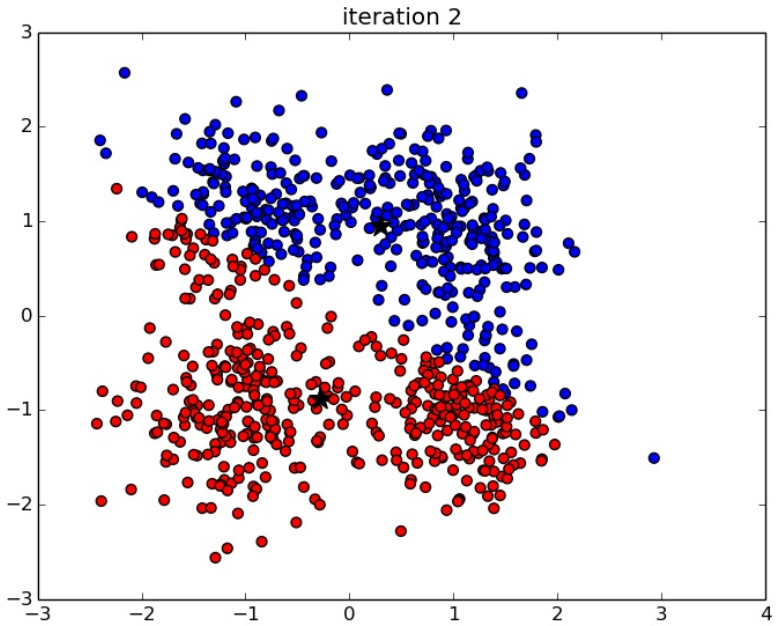
Détails

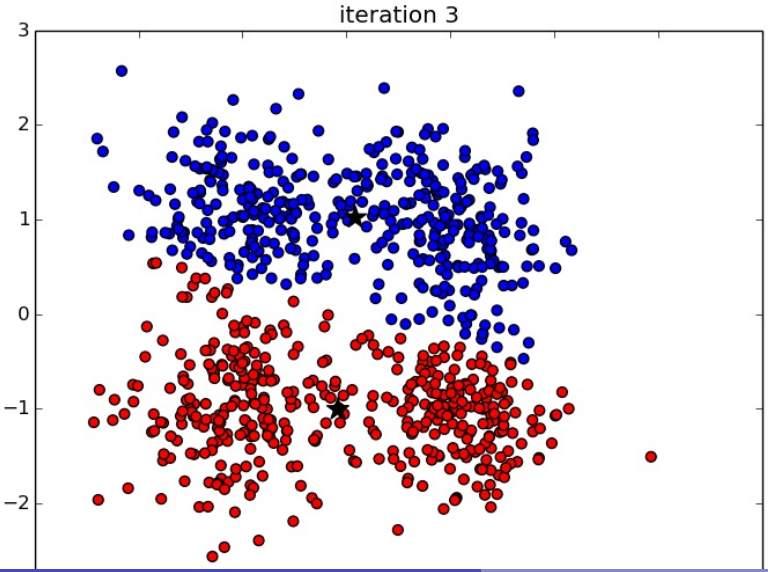
- Initialisation aléatoire des k centres :$\mu^0 = (\mu_1^0, \cdots, \mu_K^0)$
- Affectation des points x_j : classe $C^t(x_j) = argmin_i \|\mu_i^t - x_j\|^2$
- Estimation des centres : $\mu_i^{t+1} = \sum_{j:C^t(x_j)=i} \|\mu - x_j\|^2$
- On optimise $C^t = (C^t(x_j))$ et $\mu = (\mu_i)$
- Fonction de coût : $F(\mu, C) = \sum_j \|\mu_{C(x_j)} - x_j\|^2 = \sum_{i=1}^K \sum_{j:C(x_j)=i} \|\mu_i - x_j\|^2$
- Première étape : on fixe μ, on optimise $C \Rightarrow$ Calcul de l'espérance
- Seconde étape : on fixe C, on optimise $\mu \Rightarrow$ Calcul du maximum de vraissemblance

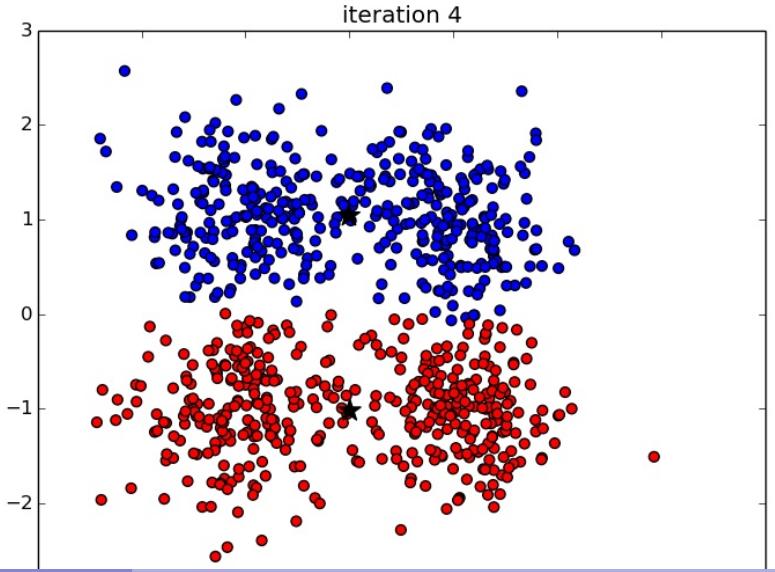
Algorithme dit *Expectation/Maximization, (EM)*

Exemples

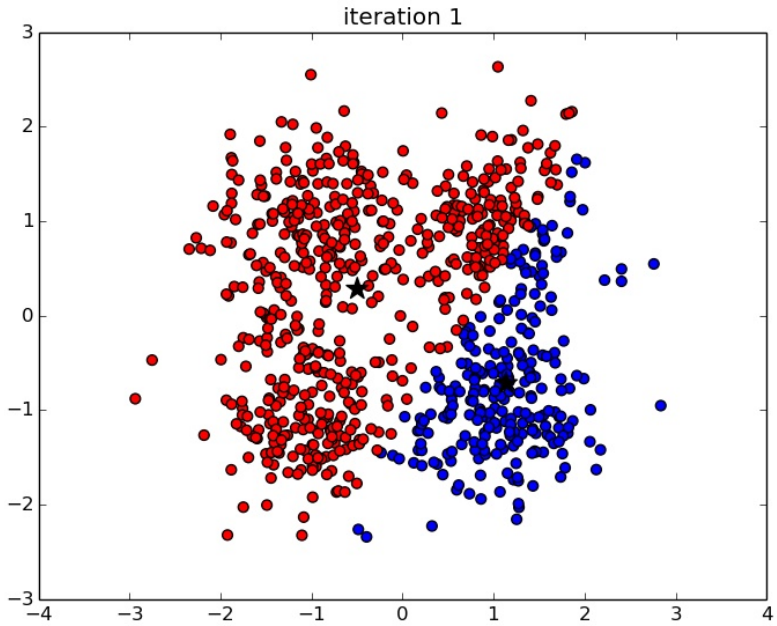


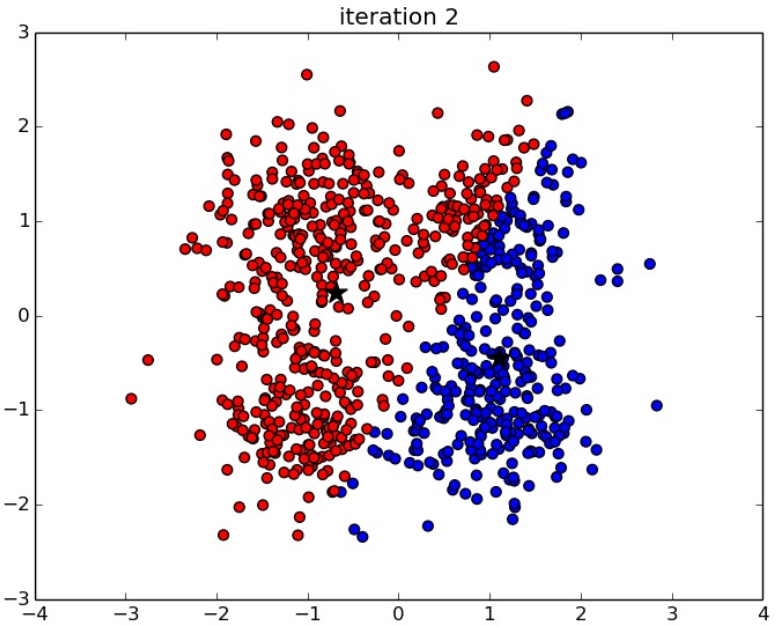


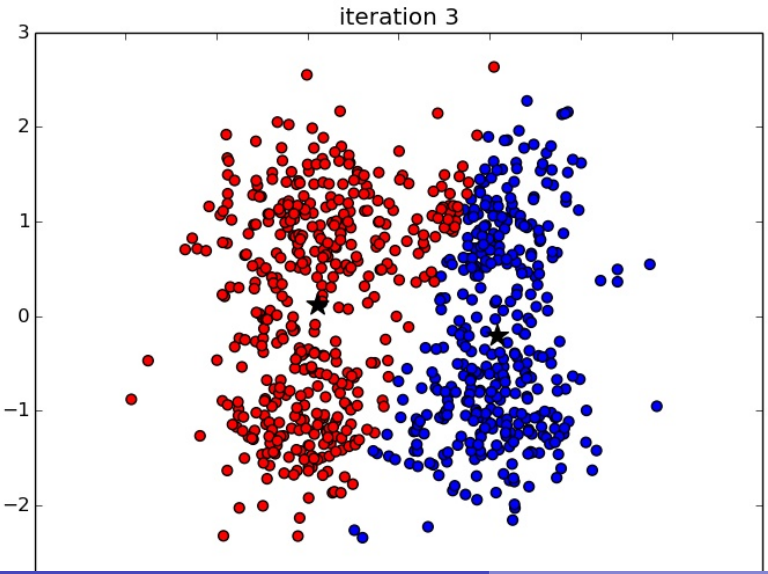


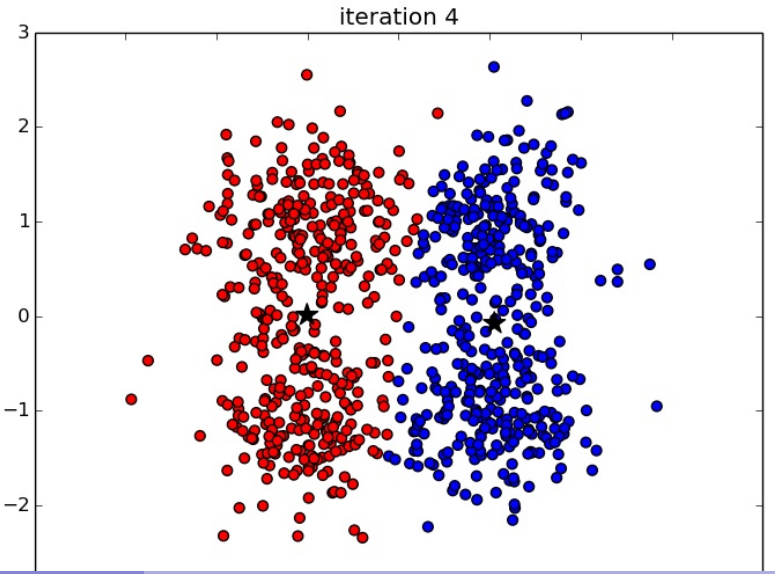


Exemples





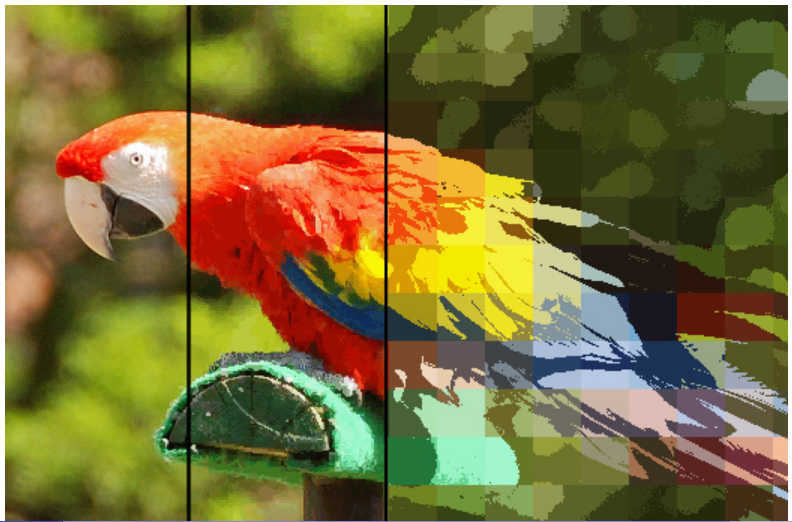




Utilisation pour la segmentation et la compression

Par moyennage des couleurs : nombre de couleurs = nombre de clusters

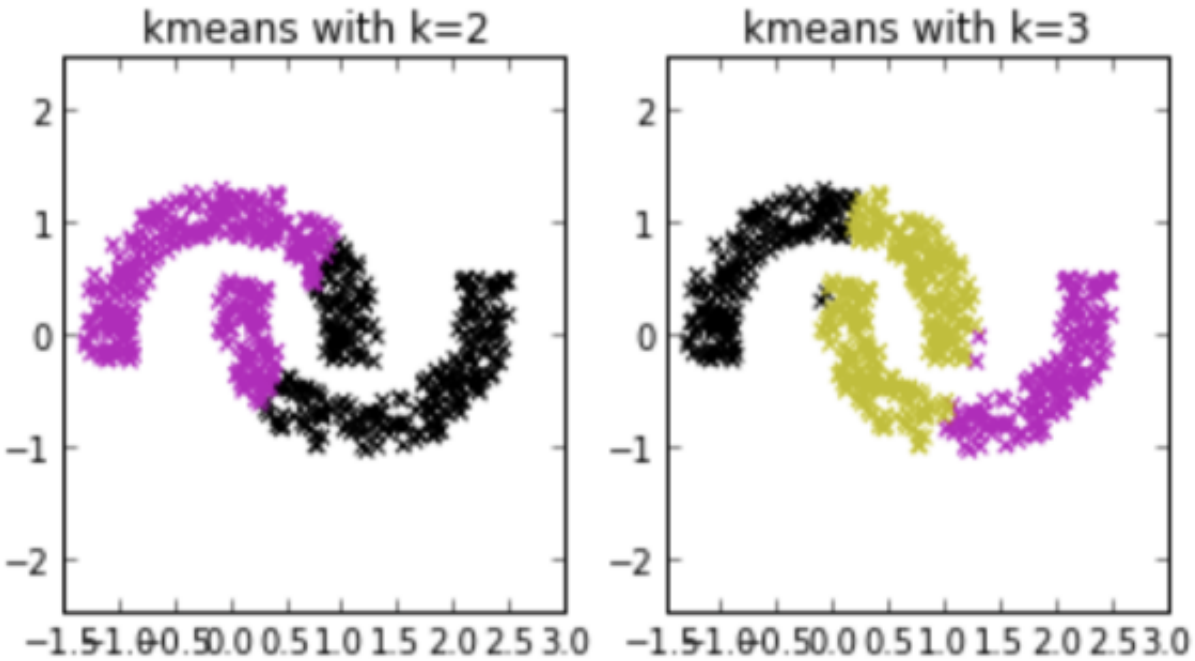
Autre approche, 2 couleurs par cellule :



Inconvénients des K-moyennes

Clusters convexes

- En fait, partitionnement Voronoi
- Clusters définis uniquement par leur centroïde
- Pas d'hierarchie: on ne peut fusionner/diviser des clusters



Clustering hiérarchique

Inconvénients des K-moyennes

Pas d'hiérarchie dans les clusters (i.e. décroître le K ne fusionne pas les clusters)

Comment prendre en compte une hiérarchie ?

- Considérer des méthodes très simples en prenant en compte des mesures de similarité entre clusters
- Soit avec des approches agglomératives en fusionnant les clusters de manière bottom-up
- ou au contraire en divisant de manière top-down les clusters

Approches agglomératives

Algorithme glouton

- Commencer avec N clusters chacun contenant un exemple
- Fusionner les clusters les plus similaires deux à deux
- Jusqu'à n'avoir plus qu'un cluster

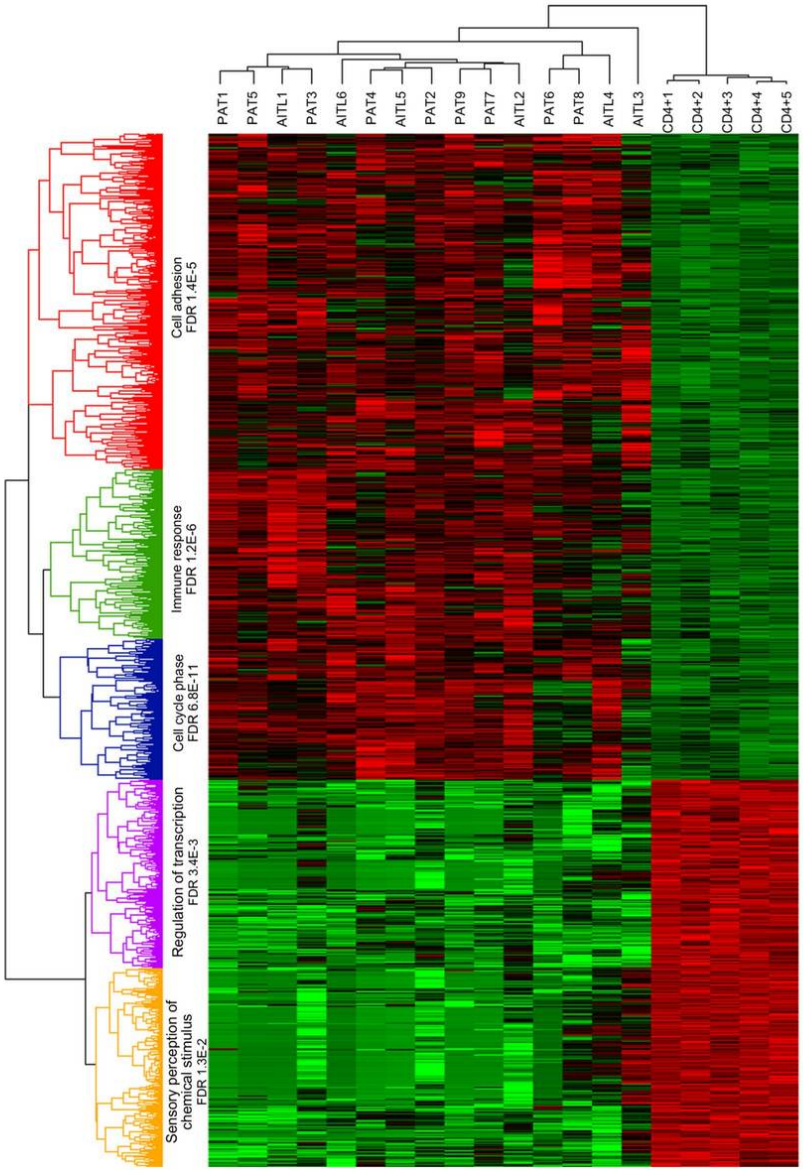
$\Rightarrow$ Construction d'un dendogramme: un arbre de partitionnement

Distance entre clusters ?

Plusieurs choix, mais souvent pas une vraie distance (un *linkage*) :

- $d(c_1, c_2) = min\ d(x, x'),\ x \in c_1, x' \in c_2$ (*single linkage*)
- $d(c_1, c_2) = max\ d(x, x'),\ x \in c_1, x' \in c_2$ (*complete linkage*)
- $d(c_1, c_2) = \mathbb{E}(\ d(x, x')),\ x \in c_1, x' \in c_2$

Exemples


PAT1
PAT5
AITL1
PAT3
AITL6
PAT4
AITL5
PAT2
PAT9
PAT7
AITL2
PAT6
PAT8
AITL4
AITL3
CD4+1
CD4+2
CD4+3
CD4+4
CD4+5
Cell adhesion
FDR 1.4E-5
Immune response
FDR 1.2E-6
Cell cycle phase
FDR 6.8E-11
Regulation of transcription
FDR 3.4E-3
Sensory perception of
chemical stimulus
FDR 1.3E-2


Quinolones
Aminoglycosides
Tetracyclines
Resistomycins
Macrolides
β-lactams
Amphenicols
Actinomycin
Anthracyclines
Rifamycins
Prefractions

Clustering par densité

Hypothèse: régions denses sont séparées par des régions de faible densité

Algorithme générique:

- Estimer la densité d'une région
- Trouver les points appartenant à cette région dense
- Rassembler ces points en un cluster

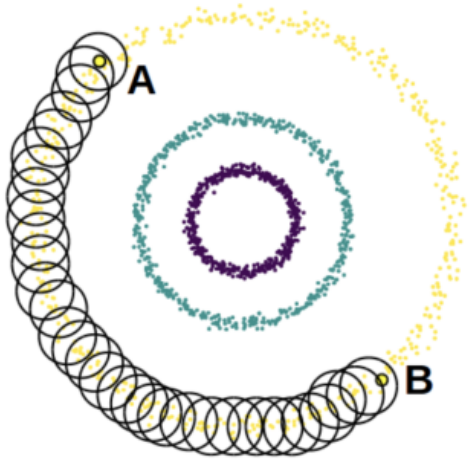
- Estimation de densité: estimation par noyaux
- Regroupement: lier les points de haute densité et les considérer comme un cluster.
- DBSCAN est l'algorithme le plus utilisé: efficace et capable de déterminer automatiquement le nombre de clusters

DBSCAN Density-based Spatial Clustering of Applications with Noise

Pour chaque point, considérer l'ensemble des points atteignables par densité autour de ce point

- Calculer le voisinage à ϵ de ce point
- Si le voisinage contient plus de m points, calculer le voisinage de chacun de ces points
- jusqu'à ce que le cluster se stabilise.

Si le point n'a pas assez de voisins, il est considérer comme du bruit.



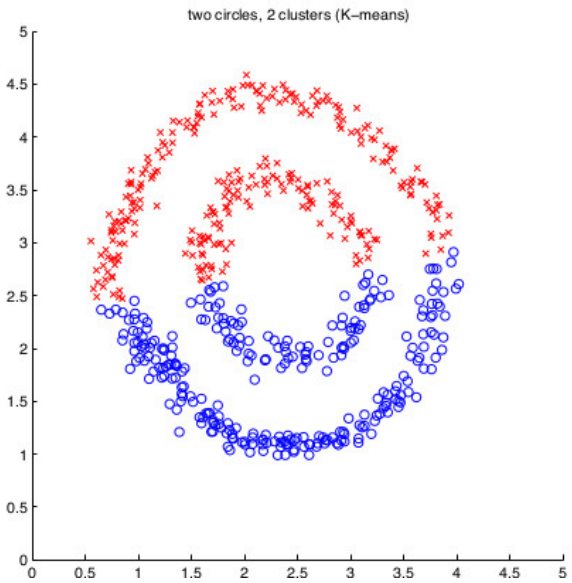
For m=4,
the orange point is not a core point,
but belongs to the neighborhood of
the red one,
so not considered as noise.

Pour/contre de DBSCAN

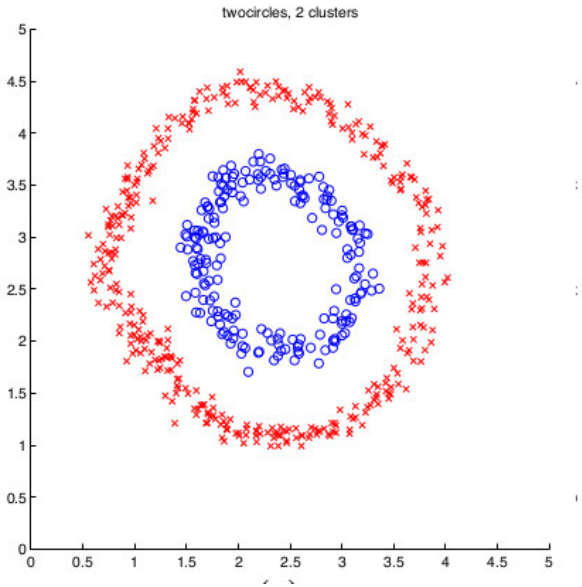
- Pas besoin de pré-définir un nombre de clusters
- Peut trouver des clusters de forme arbitraire
- Evite l'effet single-link (clusters connectés par une connection fine)
- Introduit de manière naturelle la notion de outliers/bruits
  Mais
- Difficulté du choix des hyper-paramètres (m et ϵ)
- Peut pas trouver des clusters avec des densités différentes
- Curse of dimensionality . . .

Spectral clustering : Problématique

K-means



Spectral clustering



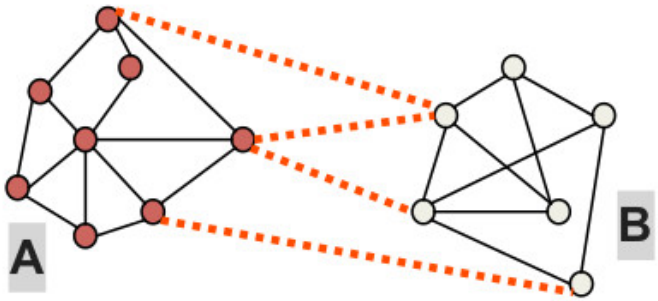
Limites des approches vues

- K-means (et en général clustering de métrique) ne trouvent que des clusters sphériques
- Comment encoder une structuration des données ? des relations de voisinages ?
- Une solution parmi d'autres : spectral clustering $\Rightarrow$ projeter les données sur un graphe de relation

Graphe de données

Notations graphe

- Données : les nœuds $V = \{x_i\}$ du graphe
- Les liens/arêtes pondérés : $E = \{w_{ij} = s(x_i, x_j)\}$ similarité entre données
- Restriction : graphe connexe

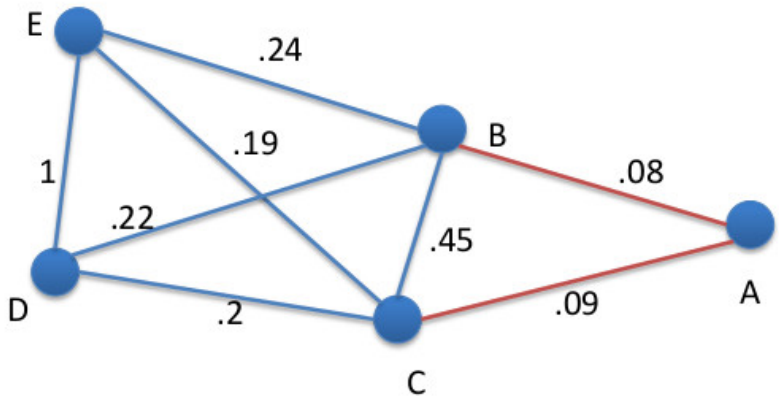


Création du graphe

- Difficile de travailler sur un graphe entièrement connecté :
  - seuil sur la mesure de similarité
  - k-nn avec k fixé
- ou utilisation de noyaux pour pondérer les arêtes : $w_{ij} = e^{-\|x_i - x_j\|^2/\sigma^2}$

Objectif

- Toujours le même :
  - données d'un même cluster très similaires
  - données de différent cluster dissimilaires
- En termes de graphe :
  - Notion de coupe : $cut(C_1, C_2) = \sum_{i \in C_1, j \in C_2} w_{ij}$, $C_1 \bigcap C_2 = \emptyset$
  - Coupe normalisé : $NormCut(C_1, C_2) = \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)}$, $Vol(C) = \sum_{i,j \in C} w_{ij}$
- Problème NP-difficile. . .

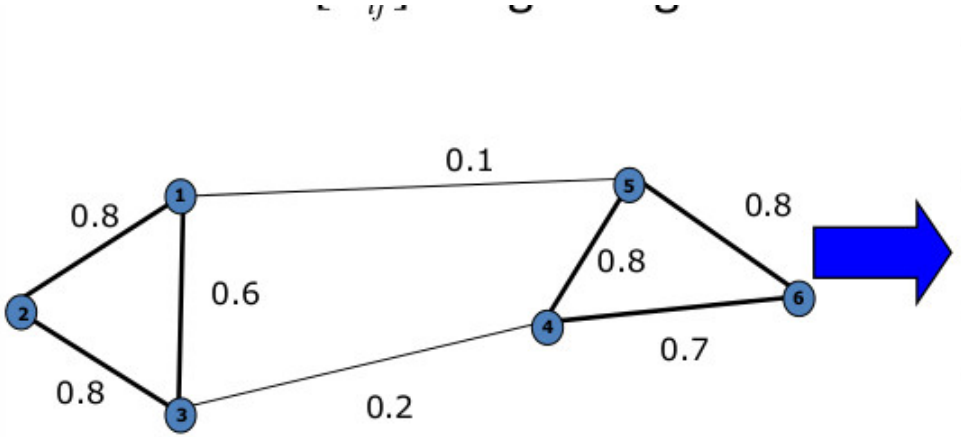


$Cut(BCDE, A) = 0.17$

$NormCut(BCDE, A) = 1.067,\ NormCut(ABC, DE) = 1.038$

Representation matricielle

- Matrice de similarité/d'adjacence: $N \times N$,
- $W : \{w_{i,j}\}$
- Matrice symétrique
- D matrice des degrés : $d_{i,i} = \sum_j w_{ij}$ pour normaliser la matrice d'adjacence



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

Représentation matricielle

- Matrice considéré : matrice laplacienne : $L = D - W$
- Propriétés :
  - Valeurs propres positives
  - Vecteurs propres orthogonaux
  - Ce sont des indicateurs de la connectivité du graphe
- Pour deux partitions C_1, C_2 :
  - soit f un vecteur dans $\{-1, 1\}$ de taille n, tel que $f_i = 1$ si $i \in C_1$, -1 si $i \in C_2$.
  - On montre que $f'Lf = \frac{1}{2}\sum_{i,j} w_{i,j}(f_i - f_j)^2$, c'est-à-dire le coût de la coupe selon cette partition :

$$f'Lf = f'(D-W)f = f'Df - f'Wf = \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij}$$

$$= \frac{1}{2}(\sum_i(\sum_j w_{ij})f_i^2 - 2\sum_{i,j} f_i f_j w_{ij} + \sum_j(\sum_i w_{ij})f_j^2) = \frac{1}{2}\sum_{i,j} w_{i,j}(f_i - f_j)^2$$

Optimisation de la coupe normalisée

Soit A un sous-ensemble de nœud du graphe

- Soit $f_i = \sqrt{\frac{|\bar{A}|}{|A|}}$ si le nœud i est dans A, $-\sqrt{\frac{|A|}{|\bar{A}|}}$ sinon.
- $f'Lf = \sum_{i,j} w_{i,j}(f_i - f_j)^2 =$

  $\sum_{i\in A, j\in\bar{A}} w_{i,j}\left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 + \sum_{i\in\bar{A}, j\in A} w_{i,j}\left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2$
- Or $cut(A,\bar{A}) = \sum_{i\in A, j\in\bar{A}} w_{i,j}$, donc $f'Lf = cut(A,\bar{A})(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2)$

$\Rightarrow f'Lf = 2|V|Ratiocut(A,\bar{A})$

- De plus, $f'.1 = \sum_i f_i = \sum_{i\in A}\sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i\in\bar{A}}\sqrt{\frac{|A|}{|\bar{A}|}} = 0$
- et $\|f\|^2 = |V|$

Le problème d'optimisation relaxé est donc :

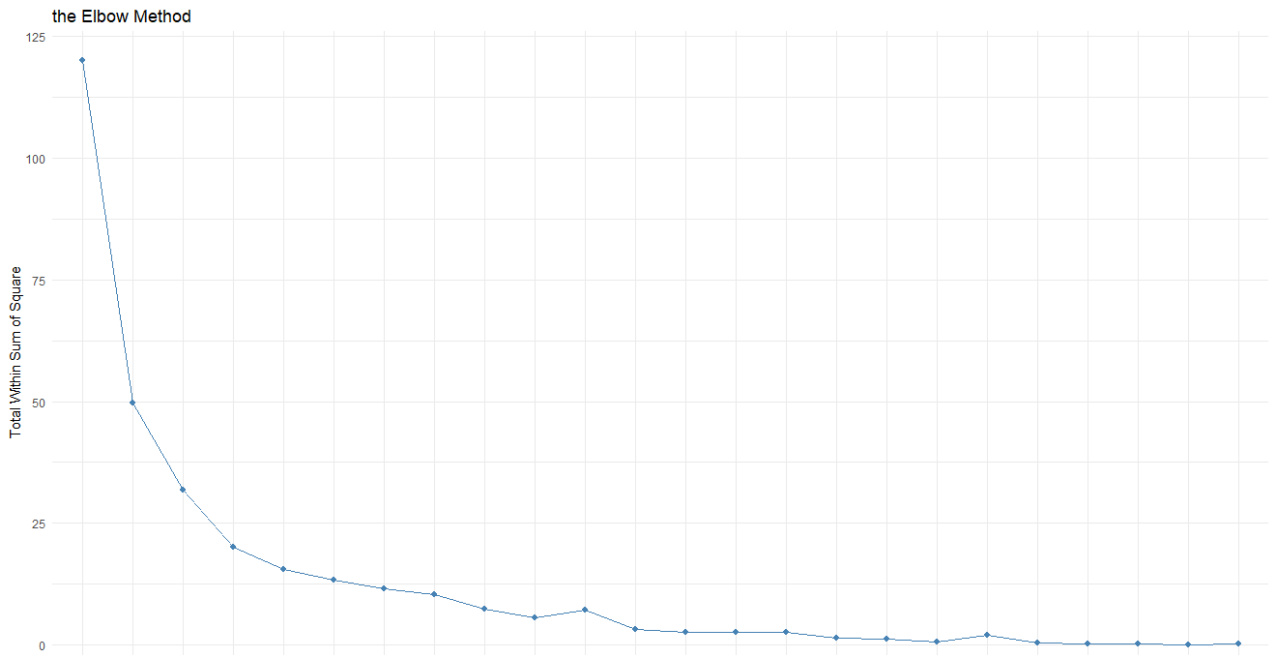
$$min_{\ f}\ f'Lf,\ \text{tel que } f'.1 = 0 \text{ et } \|f\|^2 = |V|$$

Comment choisir le nombre de clusters ?

Méthode Elbow

- Faire varier le nombre de clusters
- Etudier le ratio entre la moyenne des distances intra-cluster et la variance totale du jeu de données
- Equivalent à étudier le pourcentage de la variance expliquée par le clustering comme une fonction du nombre de clusters

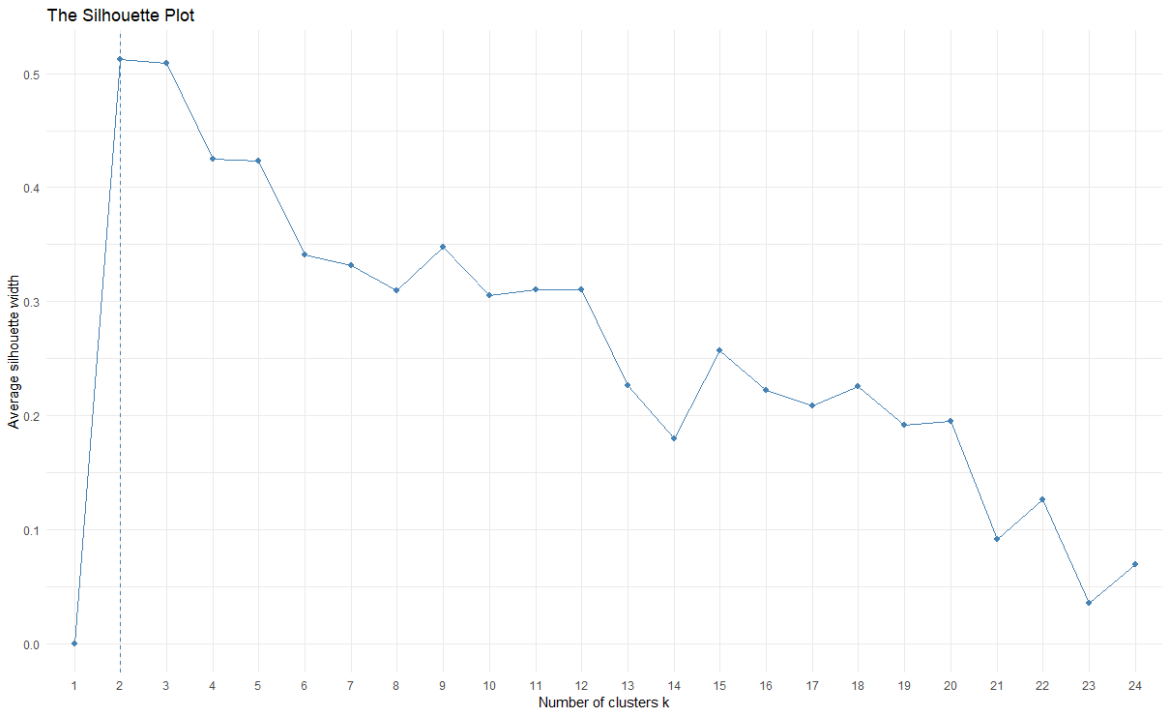
La courbe a généralement deux régimes: une avec une chute rapide du ratio et l'autre plus stable.



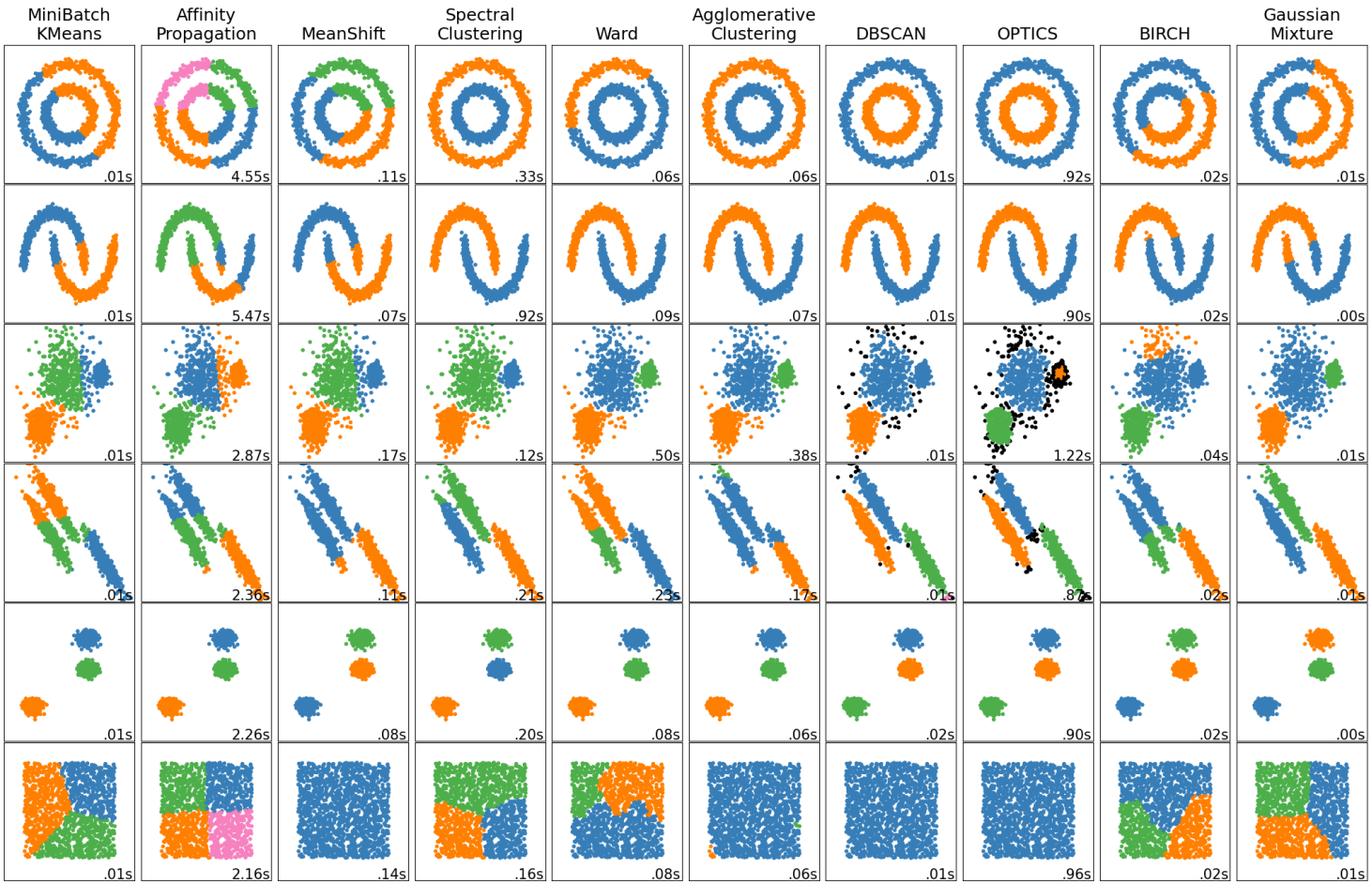
Comment choisir le nombre de clusters ?

Coefficient silhouette: dénote une sorte d'homogénéité des clusters

- Silhouette d'un point i is $\frac{b(i)-a(i)}{max(a(i),b(i))}$
- avec $a(i)$ la moyenne de la distance intra-cluster d'un point
- et $b(i)$ le minimum des distances d'un point aux autres clusters.
- Une grande valeur de silhouette dans un cluster dénote une haute homogénéité.



Conclusion : Différentes approches pour des problèmes variés ...



sklearn - Overview of clustering methods

Conclusions générales

Questions à se poser

- Qu'est-ce qu'un cluster ?
- Comment définir la similarité ?
- Quels features, quelle normalisation ?
- Combien de clusters ?
- Quelle méthode de clustering ?
- Les résultats ont-ils un sens ? clusters valides ?

Plan

Réduction de dimension: Pourquoi ?

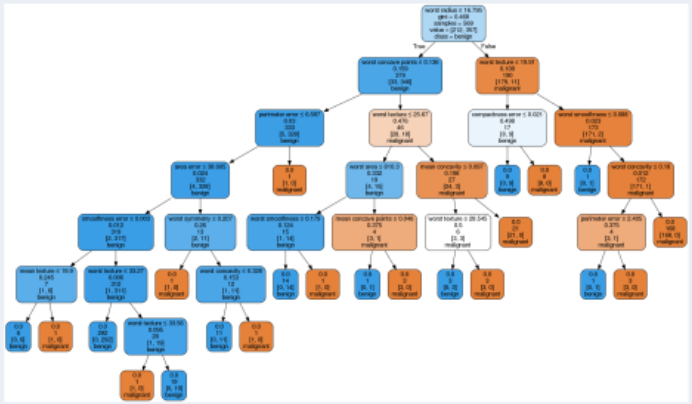
Analyse de données/Visualisation

- Comment sont distribuées les données ?
- Des dimensions sont-elles importantes ?
- Les classes sont-elles bien séparées ?

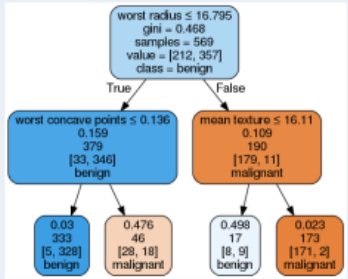
Réduction de dimension: Pourquoi ?

Interprétabilité

- Moins de dimension $\Rightarrow$ meilleure explicabilité
- Des dimensions redondantes et non pertinentes dégradent les performances des algorithmes



Real risk $= 0.92 \pm 0.05$

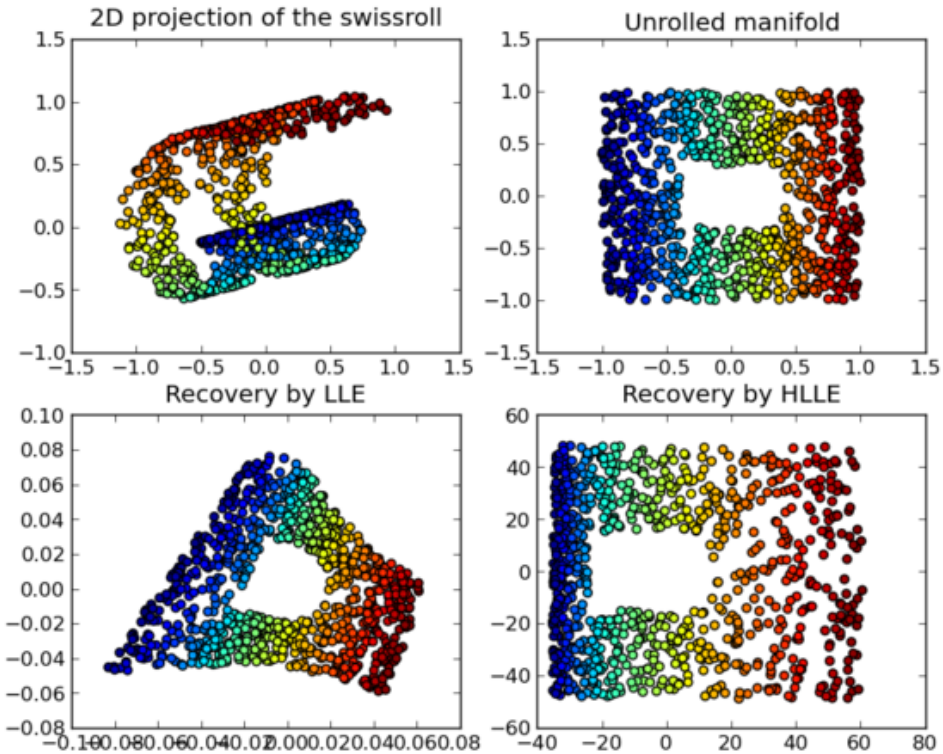


Real risk $= 0.92 \pm 0.06$

Réduction de dimension: Pourquoi ?

Souvent les données résident dans un espace de petite dimension

- Pas de perte d'information en réduisant le nombre de dimensions
- Curse of Dimensionality :
  $d^{-1/2}(max\|X_i - X_j\| - min\|X_i - X_j\|) \to 0$, $\frac{max\|X_i - X_j\|}{min\|X_i - X_j\|} \to 1$ quand $d \to \infty$
  tous les points sont quasiment équidistants . . .



Réduction de dimension: Pourquoi ?

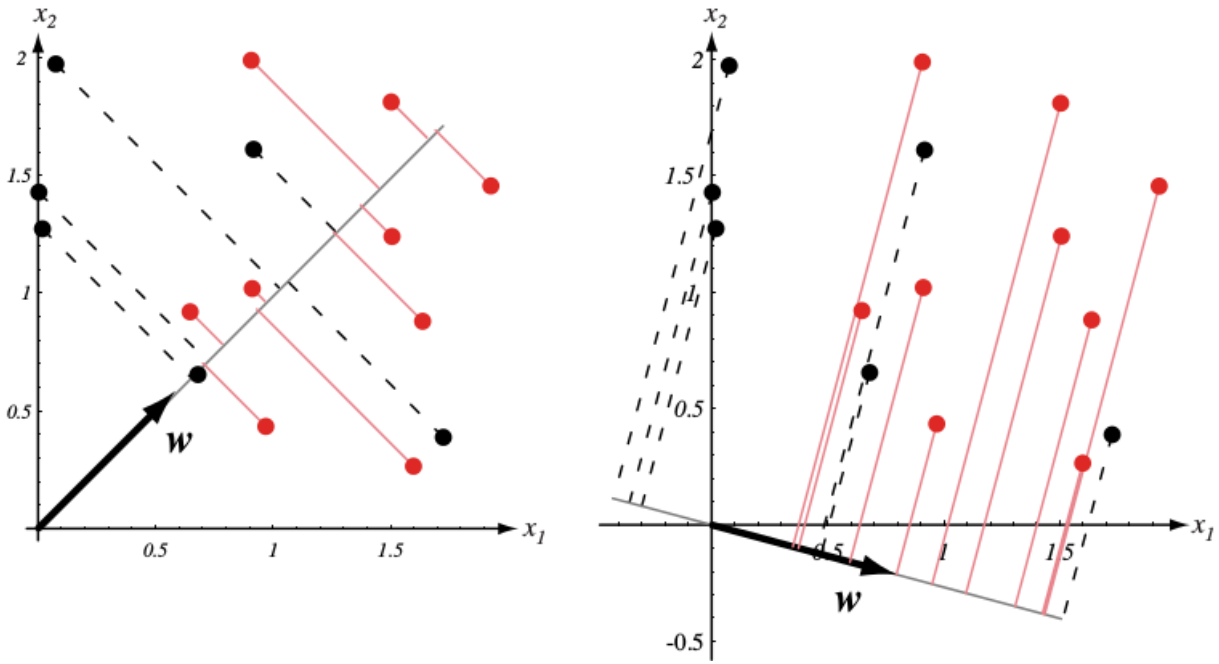
Deux approches principales

- Sélection de dimensions : trouver un sous-ensemble des dimensions d'origine
  - L'algorithme lui-même se restreint à un certain nombre de dimensions (pénalisation, expressivité)
  - Filtrage : des mesures d'évaluations pour juger l'intérêt de chaque dimension
  - Posthoc : éliminer certaines dimensions et observer la variance des performances
- Construire de nouvelles dimensions !
  - en considérant une erreur de reconstruction
  - ou en conservant les distances entre exemples

Principal Component Analysis

Principe:

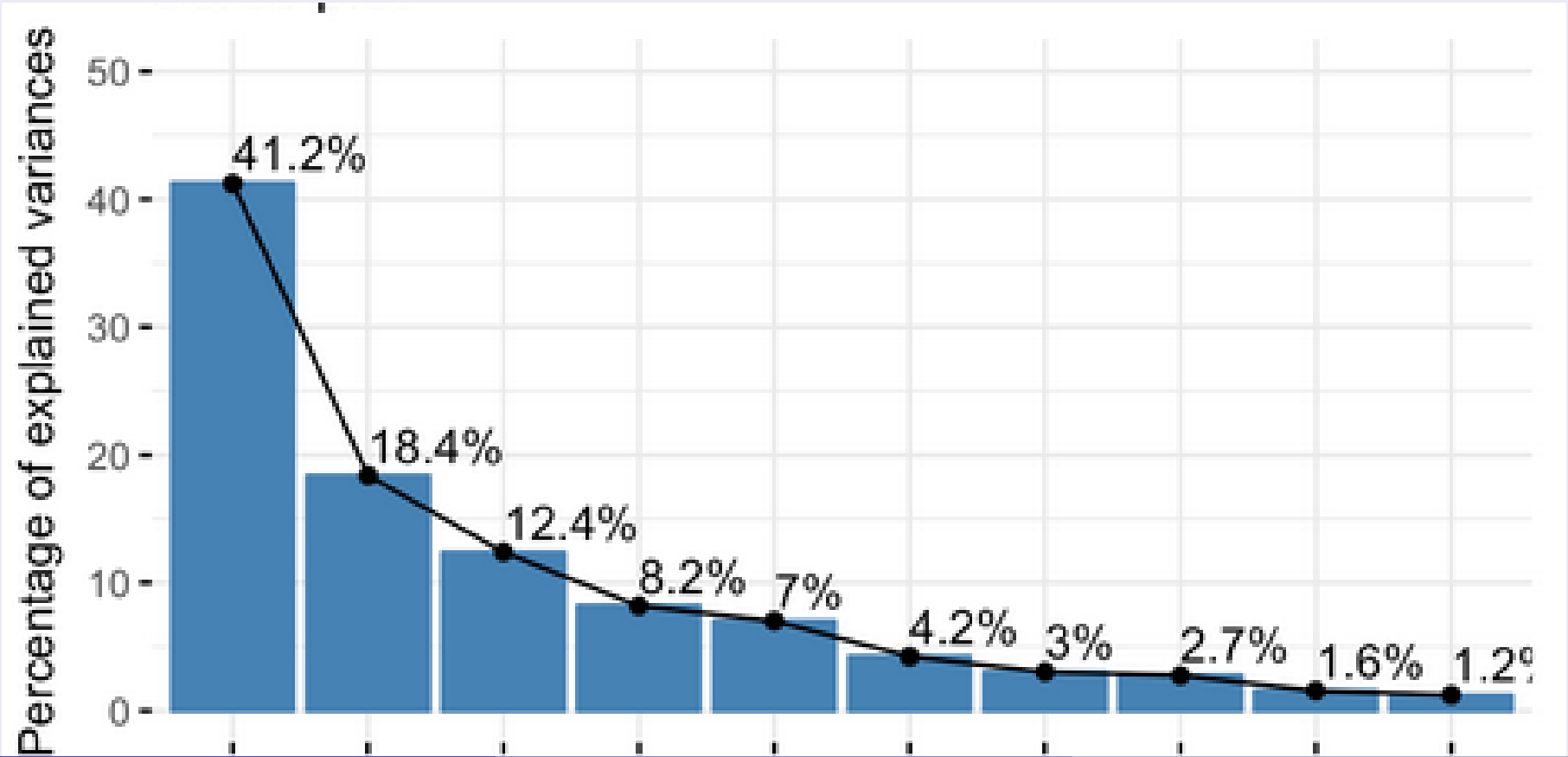
- Trouver une nouvelle base avec très peu de dimensions
- Chaque dimension est une combinaison linéaire des dimensions d'origine
- Chaque nouvelle dimension doit décrire le plus d'information possible $\Rightarrow$ Maximisation de la variance



Principal Component Analysis

Pour plusieurs dimensions

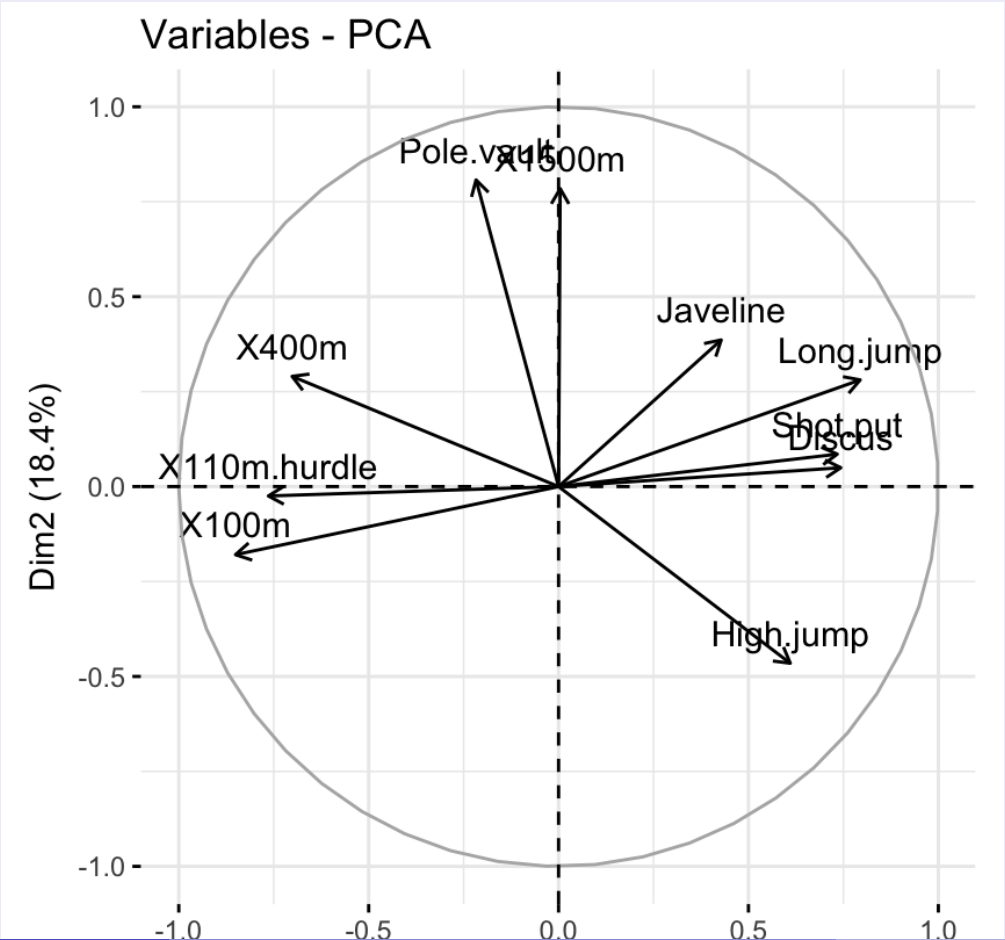
- Choix optimal donné par la valeur propre maximale de la matrice de covariance $(X - \mu)(X - \mu')$
- Inertie projettée donnée par la somme des valeurs propres.
- Souvent chute rapide des valeurs propres.



Principal Component Analysis : Interprétation

Cercle de corrélation

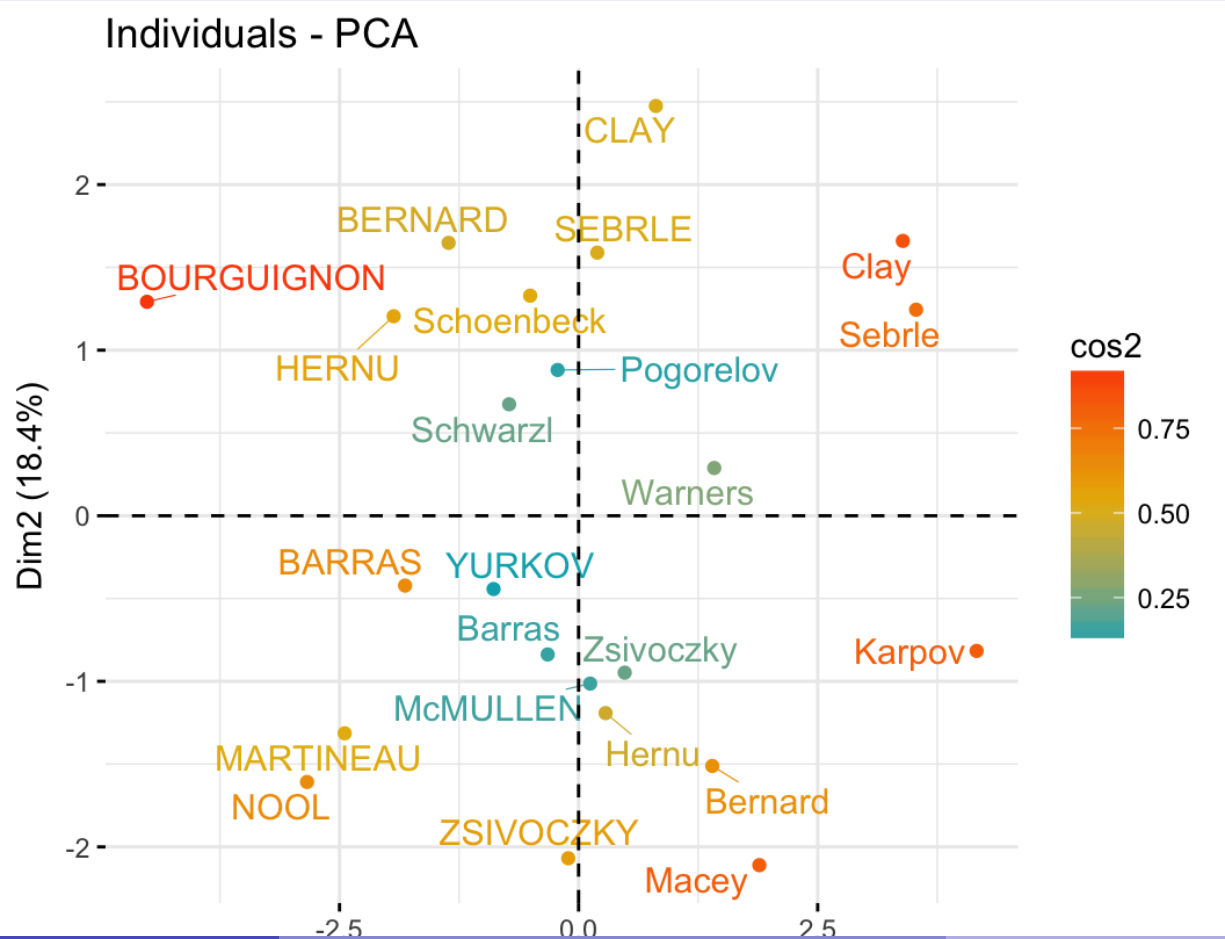
- les variables corrélées positivement sont groupées.
- les variables corrélées négativement sont positionnées à l'opposé
- les variables loin de l'origine sont bien représentées



Principal Component Analysis : Interprétation

Graphique des individus

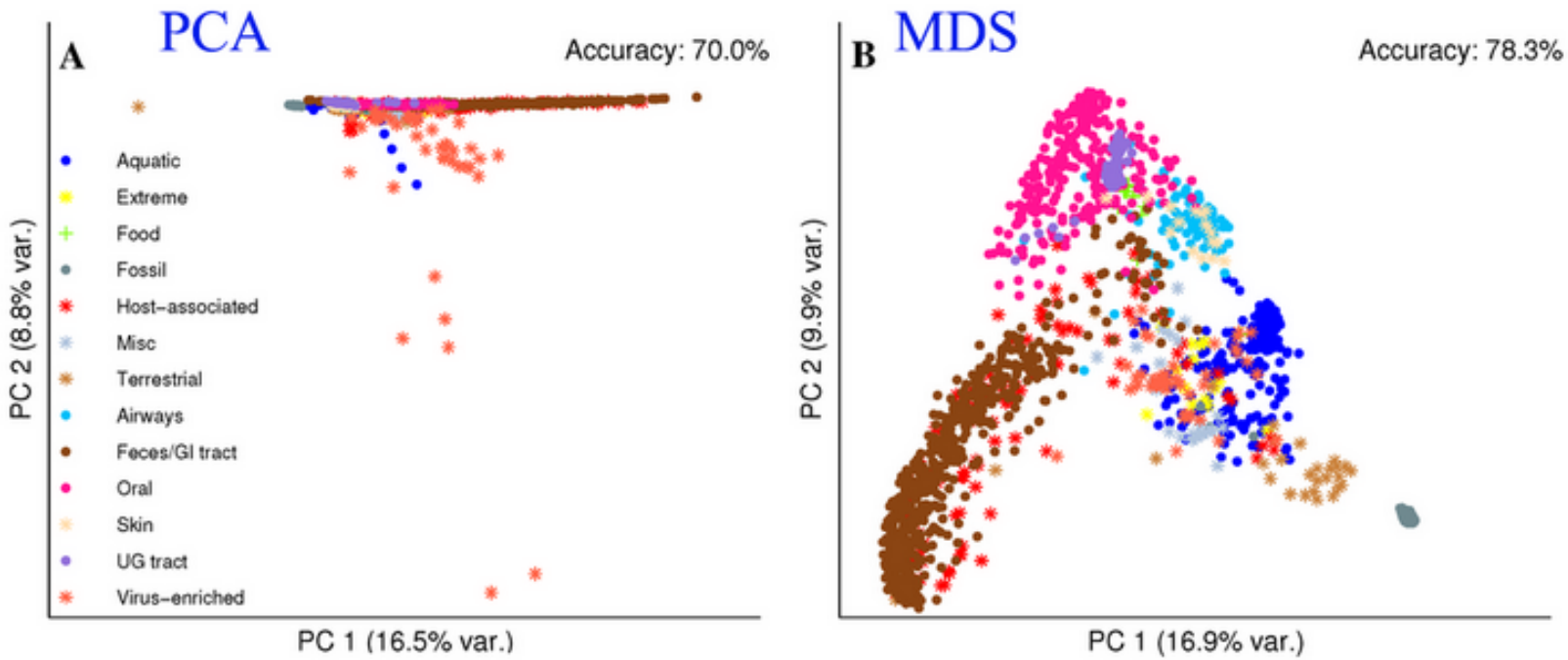
- Individus proches ont tendance à être similaire
- Mais il faut regarder les autres dimensions !
- Ceux prochent de l'origine ne sont pas bien représentés par les axes



Multi-dimensional Scaling (MDS)

Objectifs:

- Trouver une projection des données de manièr à préserver les distances deux à deux des points
- Ne calcule pas explicitement des nouvelles dimensions
- Utilisé essentiellement pour de la visualisation
- Ne nécessite pas explicitement la description des exemples, uniquement les distances deux à deux.



Multi-dimensional Scaling (MDS)

Principe:

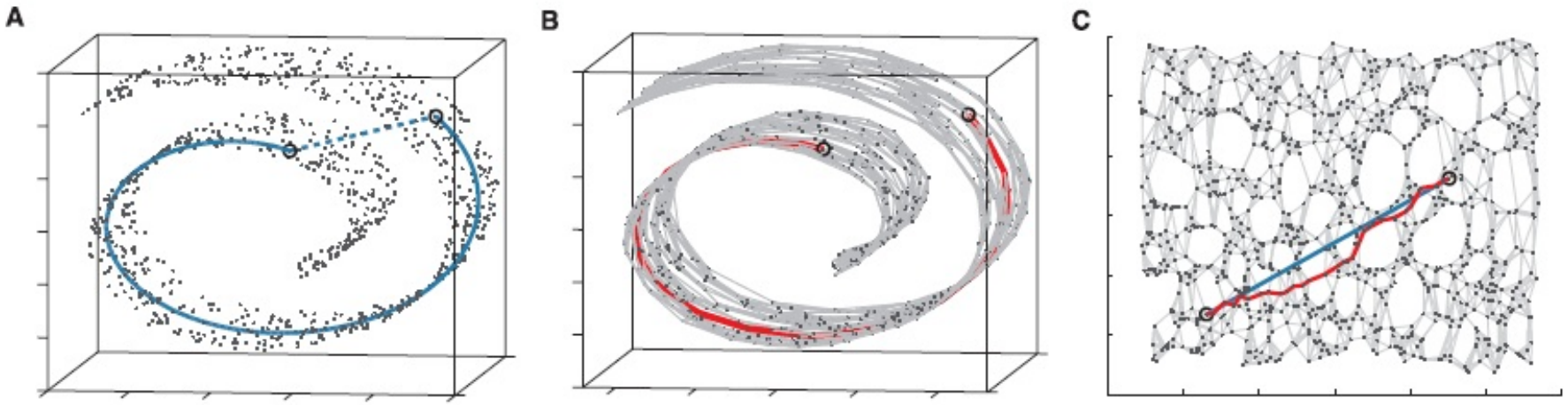
- Matrice de Gram : $G = (X - m)(X - m)^T$ avec $m = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$
- Nouvelles coordonées : $x' = U^T(x - m)$ avec U orthonormale
- Objectif : Minimiser le produit scalaire entre les vecteurs $\sum_{i=1}^{N} \sum_{j=1}^{N} (\langle x_i - m, x_j - m \rangle - \langle x'_i, x'_j \rangle)^2$
- Equivalent à minimiser : $\sum_{i=1}^{N} \sum_{j=1}^{N} (\langle x_i - m, x_j - m \rangle - (x_i - m)^T U U^T (x'_j - m))^2$ wrt $U^T(x_i - m)$
- Solution donnée par les vecteurs propres de G
- Très similaire à la PCA mais sur XX^T plutôt que $X^T X$

Inconvénients:

- Prend en compte toutes les dimensions (bruit, corrélation, . . .)
- Applicable que si $N << d$ et si d est petit . . .

Isomap

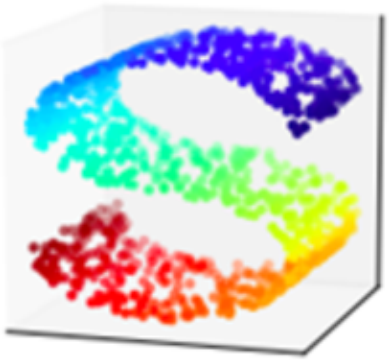
- La distance euclidienne n'est pas toujours la *bonne* distance
- Essaye de préserver la distance géodésique des données
- En suivant un chemin à travers les données



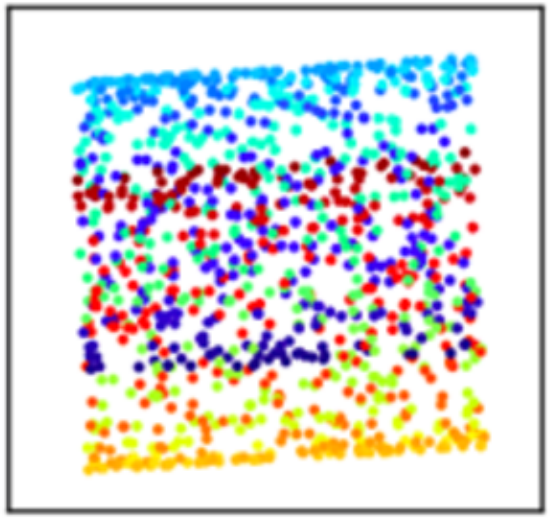
Isomap

Principe:

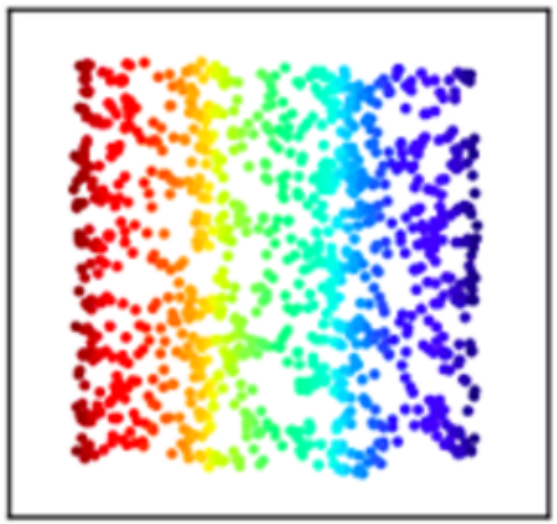
- Calcule le graphe de voisinage (avec un K-NN par exemple)
- Calcule la distance entre les points deux à deux dans ce graphe (chemin pondéré le plus court entre les points)
- Un MDS est utilisé sur le résultat



PCA projection



IsoMap projection



t-SNE

Objectif

- Trouver une projection qui conserve le voisinage de chaque point
- Pas de transformation explicite, seulement les nouvelles coordonées
- Pas d'hypothèse sur l'espace d'origine, seul la distance deux à deux est utilisée

Principe

- Soit $P_{j|i} = \frac{e^{-\|x_i - x_j\|^2/2\sigma_i^2}}{\sum_k e^{-\|x_i - x_j\|^2/2\sigma^2}}$, la probabilité que x_i choisisse x_j comme voisin
- Trouver les nouvelles coordonnées x'_i telles que $Q_{j|i}$ la distribution de probabilité dans le nouvel espace est proche de $P_{j|i}$.
- Utilise une distance de Kullback-Leibler afin de faire correspondre les distributions en utilisant une descente de gradient

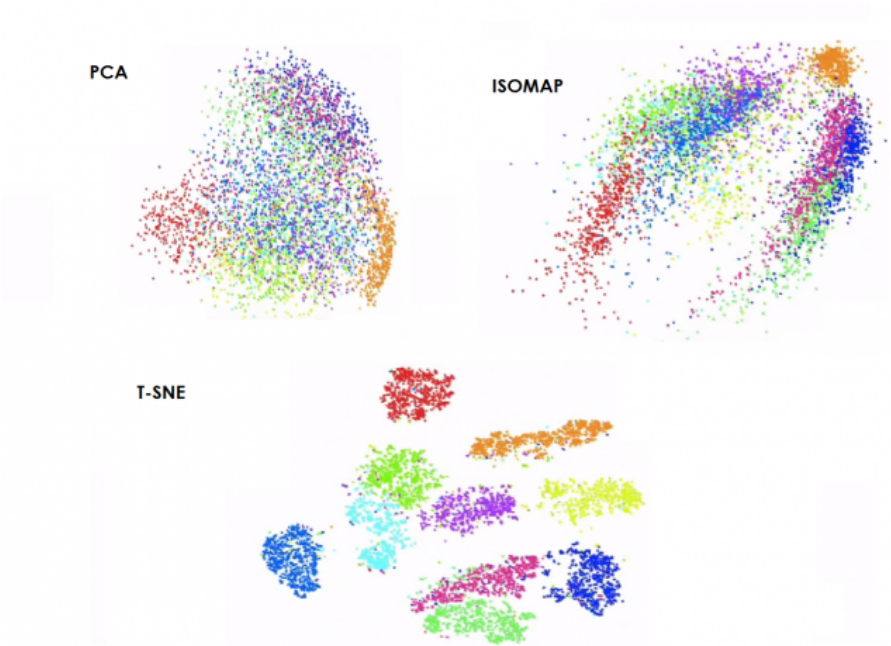
t-SNE

Interpretation

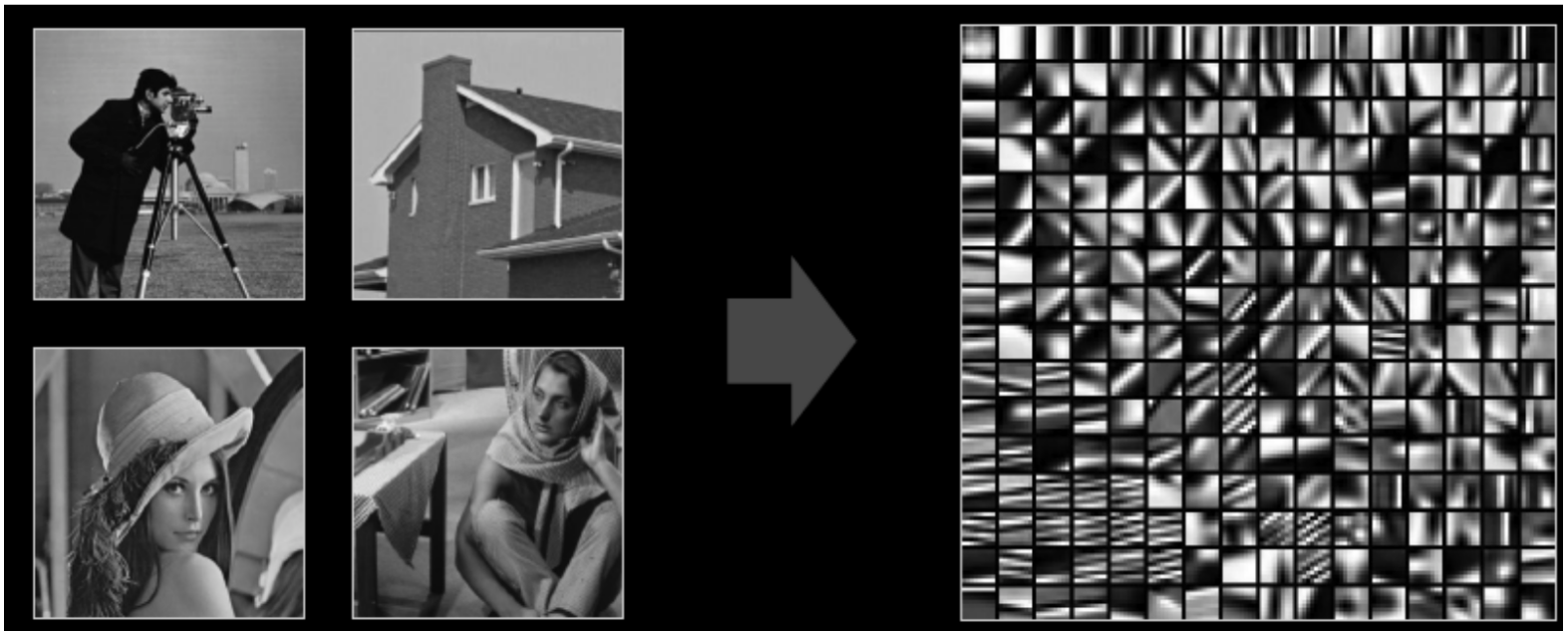
- le voisinage local est conservé : les points proches sont également proches dans l'espace d'origine
- Mais les distances ne sont pas préservées !

$\Rightarrow$ en dehors d'une petite région autour d'un point, les autres distances n'ont pas de sens !

- Attention : en fonction de l'hyperparamètre, il est très facile de construire des petits clusters isolés, mais souvent sans aucune signification !



Apprentissage de dictionnaire



- Une image : constituée d'un petit ensemble de primitives.
- Problème de la PCA : base orthogonale ! pas de redondance
- Peut-on apprendre un dictionnaire de primitives pour représenter un jeu de données ?

Compress sensing

- Objectif : trouver D tel que $x \approx Dx'$
- Contrainte de sparsité : $\|x'\|_0$ très faible, peu d'*atomes* sont nécessaires à reconstruire x
  - simplicité : quelques atomes suffisent à expliquer x
  - signification : la représentation explique x
  - parcimonie : x est décrit que parce qui le représente
- Problème d'optimisation : $argmin_D \|Dx' - x\| + \lambda \|x'\|_0$ (différentes normes, différentes variantes)
- Approche dérivée de la physique (analyse de wavelet, fourier, . . .)

Applications

Débruitage

[Mairal et al 2009]

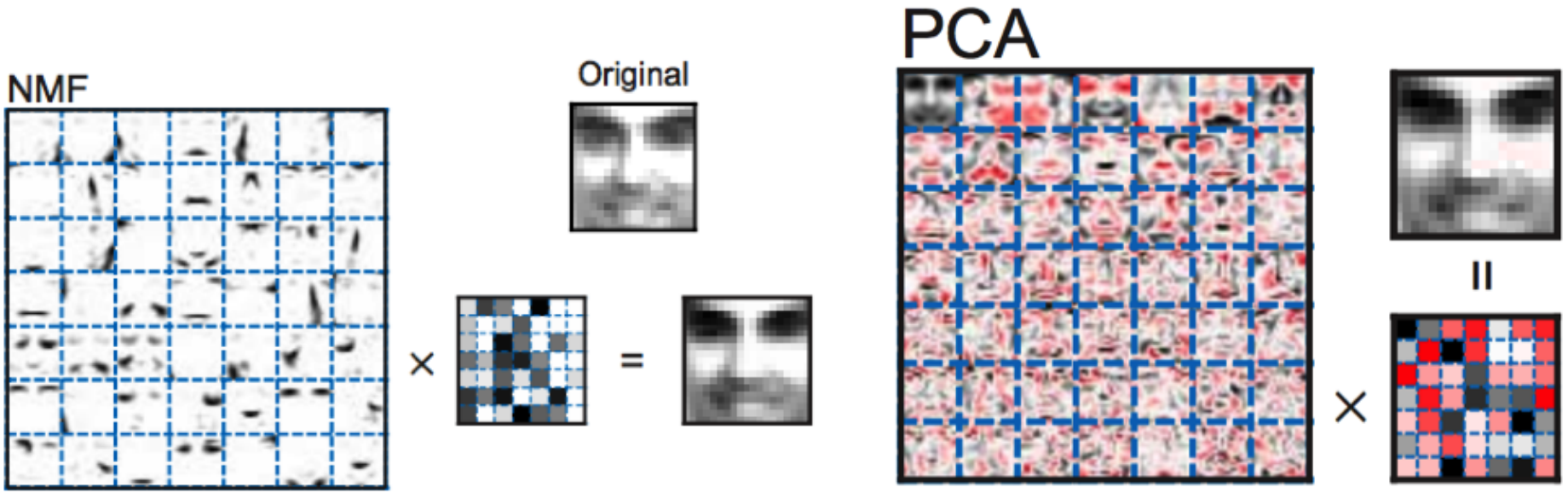
Applications

Inpainting

[Mairal, Elad, Sapiro 2008]

Factorisation matricielle non négative

- Décomposition sur un dictionnaire additif uniquement
- $x \approx Dx'$, avec $x' > 0$
- Intérêt : plus interprétable, plus réaliste sur un ensemble de problèmes

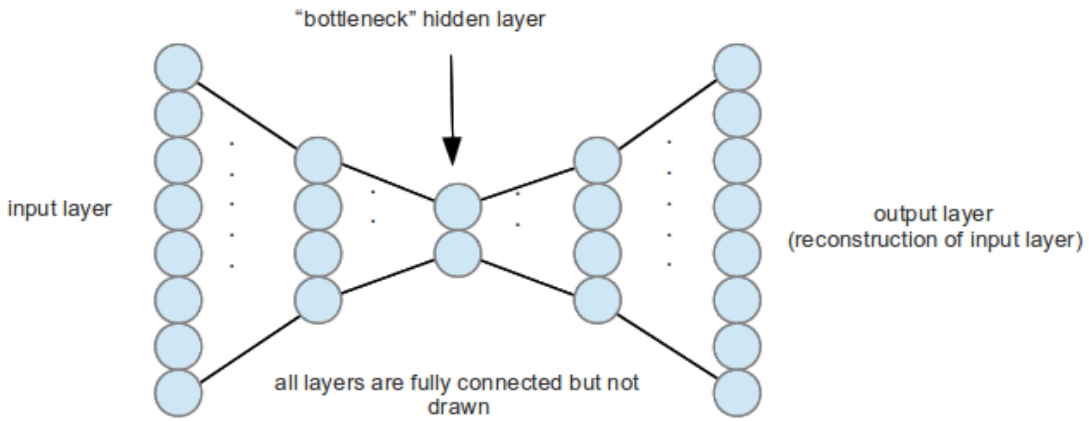


Multiples applications : séparation de sources, en topic discovery, . . .

Auto-encoders

Principe

- Apprendre un réseau de neurones qui reconstruit au mieux l'entrée
- Encodage sur une couche cachée $\Rightarrow$ réduction de dimension



Applications

- Visualisation
- Débruitage
- Réduction de dimension, espace latent de représentation