Chapitre 6. Apprentissage non supervisé

Claude Petit, Inria Rennes, France - claude.petit@cpmath.fr

Jan. 2025

Overview

Formalisation et principales méthodes

Classification ascendante hiérarchique

Méthode des k-moyennes

Méthodes à base de réseaux de neurones

Spectral clustering

Apprentissage semi-supervisé : un exemple

Formalisation et principales méthodes

Pas d'étiquette car...

- Pas de temps ou d'argent.
- Pas de spécialiste pour étiquetter.
- Trop de catégories.
- Impossible à étiquetter.

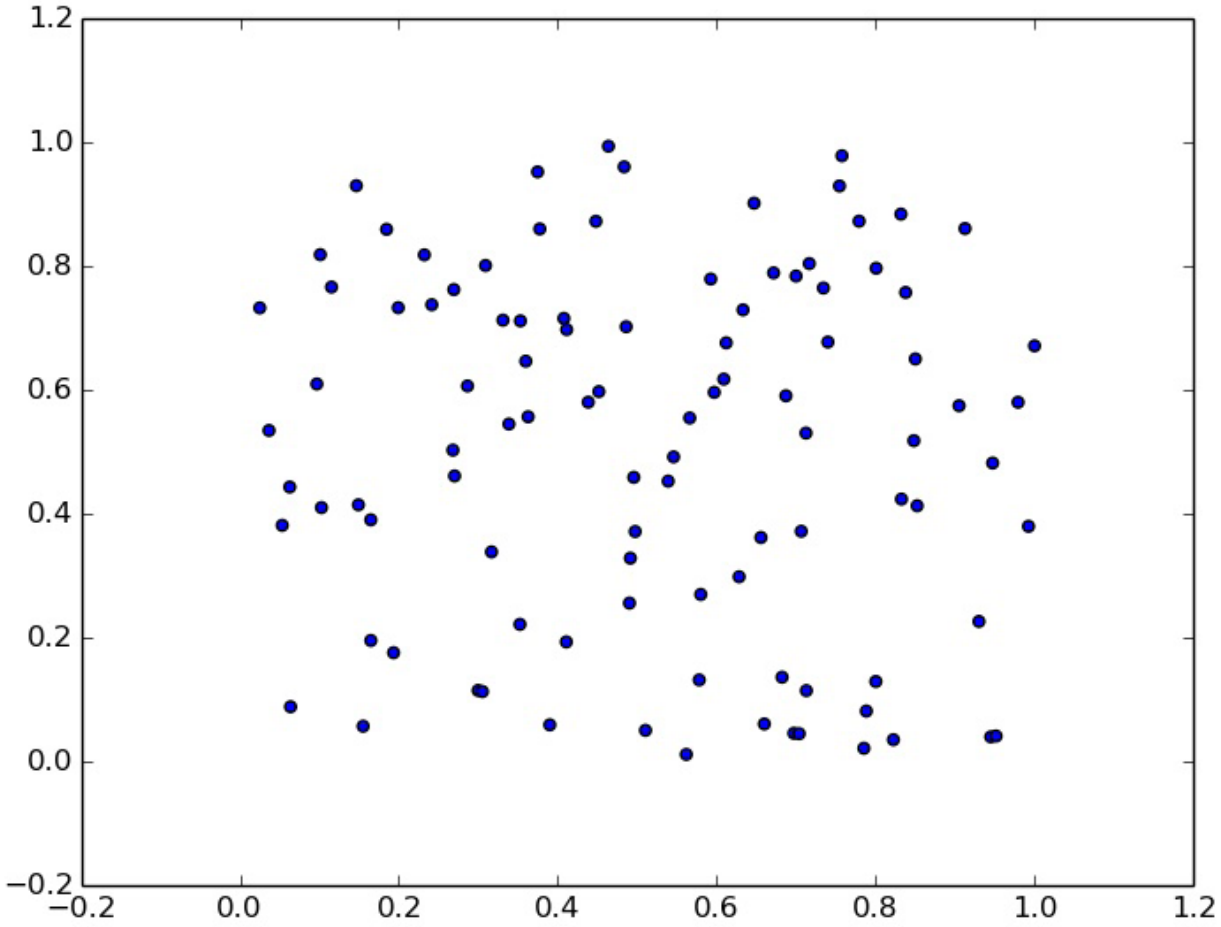
$\Rightarrow$

- Structurer les données.
- Regrouper ce qui se ressemble ("ce qui se ressemble s'assemble").
- Eloigner ce qui est vraiment différent.
- Cluster (dans une partition) : groupe de "données similaires".

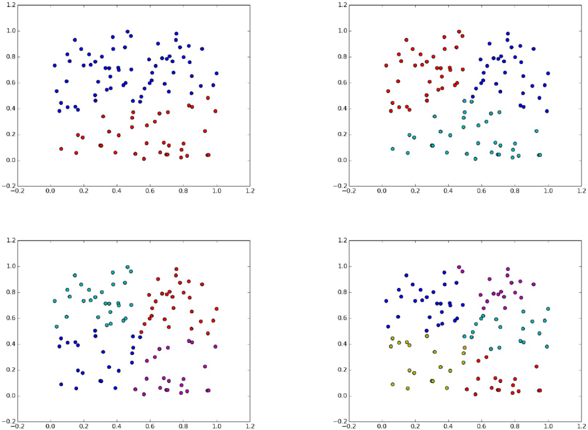
$\Rightarrow$ Importance de définir une bonne notion de similarité.

Exemple (Nicolas Baskiotis) -1-

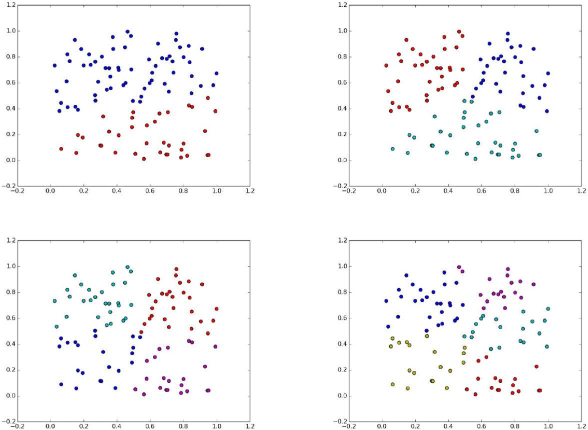
Quel est le bon partitionnement ?



Exemple (Nicolas Baskiotis) -2-



Exemple (Nicolas Baskiotis) -2-



Aucun ! Échantillon de loi uniforme.

Apprentissage non supervisé : un problème de similarité

Différentes approches :

- Géométrique : connectivité, centroïde (k-moyennes, CAH).
- Graphes (spectral clustering).
- Distribution de probabilités latentes (estimation de densités).
- Modèles bayésiens.
- Apprentissage génératif (réseaux de neurones).

Partitionnement :

- k-moyennes, DBSCAN, Mean Shift.
- Hard : une donnée appartient à un unique groupe.
- Soft : probabilité d'appartenance à un groupe.
- Nombre de classes k inconnu *a priori*.
- Similarité intra-groupe et dissimilarité inter-groupe.
- La malédiction de la dimension n'est jamais loin.

Formalisation mathématique

- Échantillon $\mathcal{D} = \{X_1, ..., X_n\}$ avec $X_i \in \mathbb{R}^d$.
- Partition π_k sur $\mathcal{D}$: $\mathcal{D}_1$,..., $\mathcal{D}_k$.
- Critère de similarité d (distance) sur $\mathbb{R}^d$ ou $\mathbb{X}$.
- Critère de similarité D sur les sous-ensembles de $\mathcal{D}$.
- Clustering : à k fixé, trouver $\pi_k^\star = \arg\min_\pi \phi(\pi)$
- ϕ est une fonction des distances d et D.

Distances sur $\mathbb{R}^d$ et sur $\mathcal{P}(\mathcal{D})$

$$d_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

- $p = 2$ distance euclidienne.
- $p = 1$ distance de Manhattan.
- $p \to 0$ distance de Hamming.
- p quelconque distance de Minkowski.

$$D(A,B) =$$

- PPV (simple linkage) : $\min\{d(x,y), x \in A, y \in B\}$
- Diamètre max (complete linkage) : $\max\{d(x,y), x \in A, y \in B\}$
- Moyenne (average linkage) : $\frac{1}{|A|.|B|} \sum_{x,y} d(x,y)$
- Ward : $\frac{|A|.|B|}{|A|+|B|} ||m_A - m_B||^2$
- Barycentres : $d(A,B) = d(m_A, m_B)$

$A, B \in \{\mathcal{D}_1, ..., \mathcal{D}_k\}$ et $m_A = \sum_{x \in A} 1/|A|$ barycentre (centroïde) de A.

Classification ascendante hiérarchique

Classification ascendante hiérarchique : principe

Algorithme glouton :

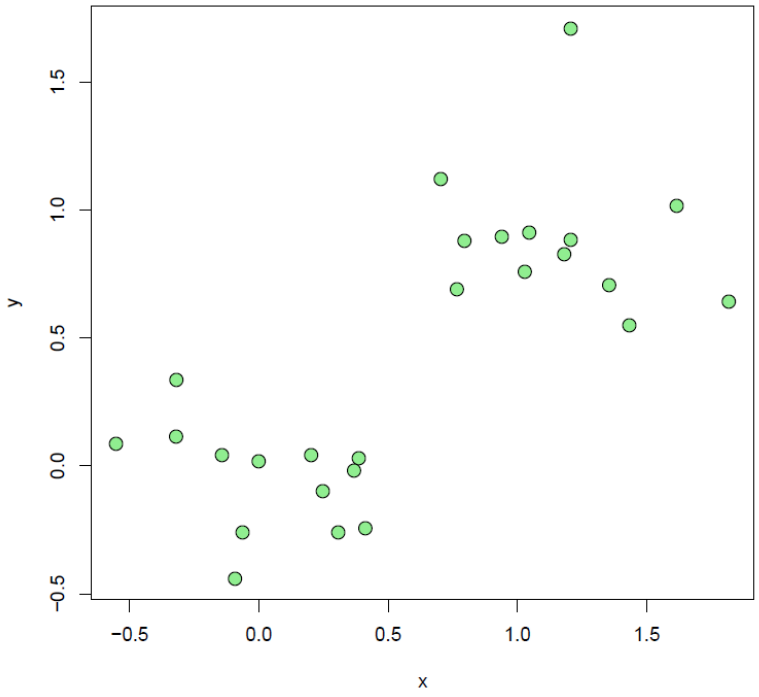
- Fusionner les partitions les plus semblables selon D.
- Construire des clusters de plus en plus larges.
- S'arrêter quand il reste un unique cluster.
- $\Rightarrow$ Arbre de partitionnement binaire : dendrogramme.

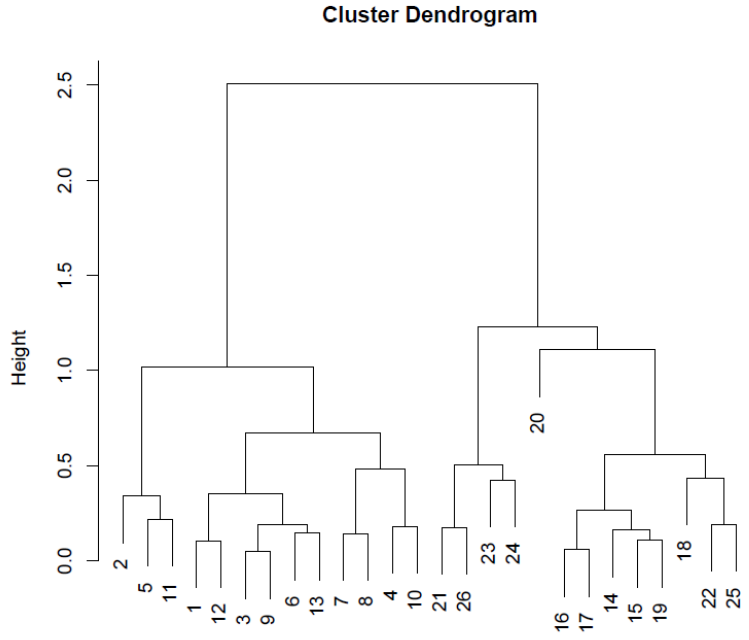
CAH : ce n'est pas une méthode de classification, mais de parititionnement (non supervisé) !

Selon le choix de D, le dendrogramme est plus ou moins équilibré.

Le choix de k est également important... et subjectif.

CAH : exemple 1 (J. Salmon, N. Verzelen) -1-

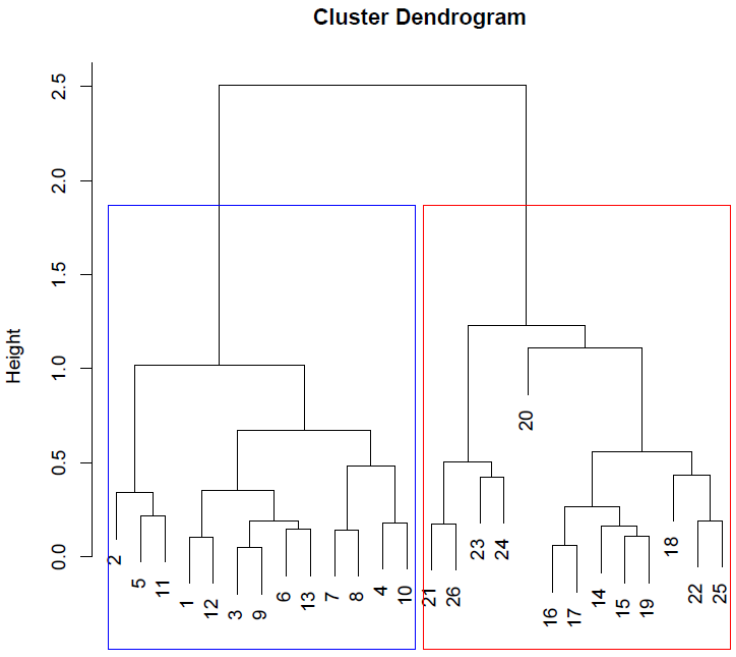


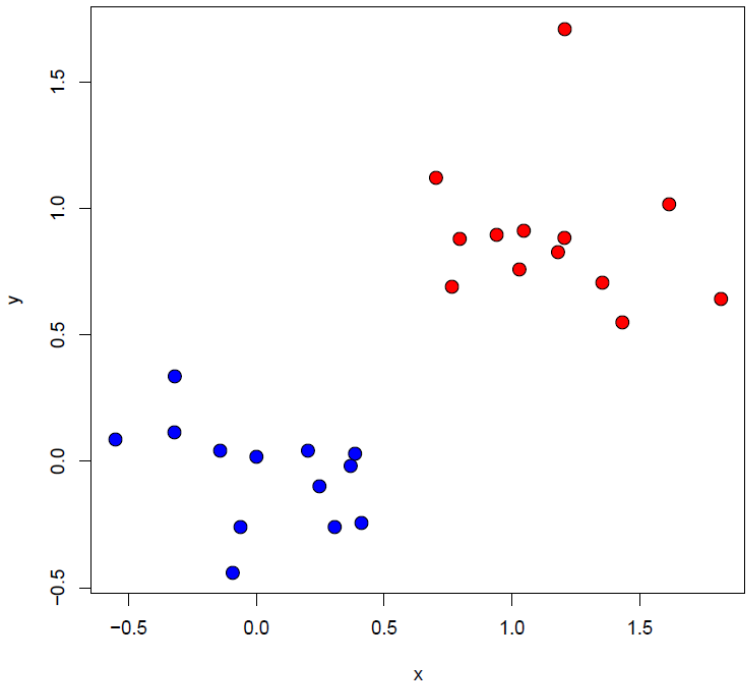


(Figure : J. Salmon, N. Verzelen)

Height sur l'axe (Oy) : distance entre les clusters.

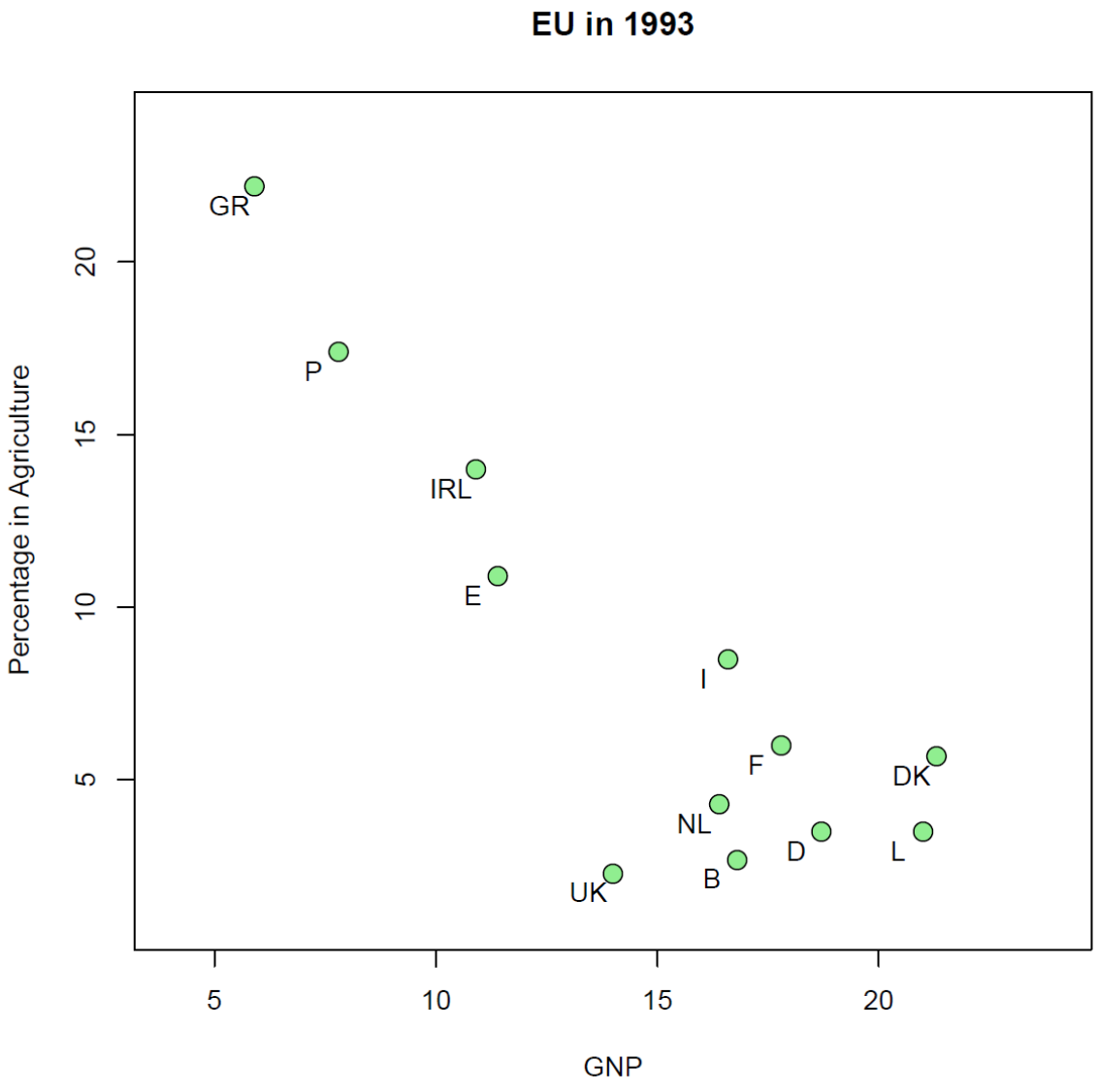
CAH : exemple 1 -2-





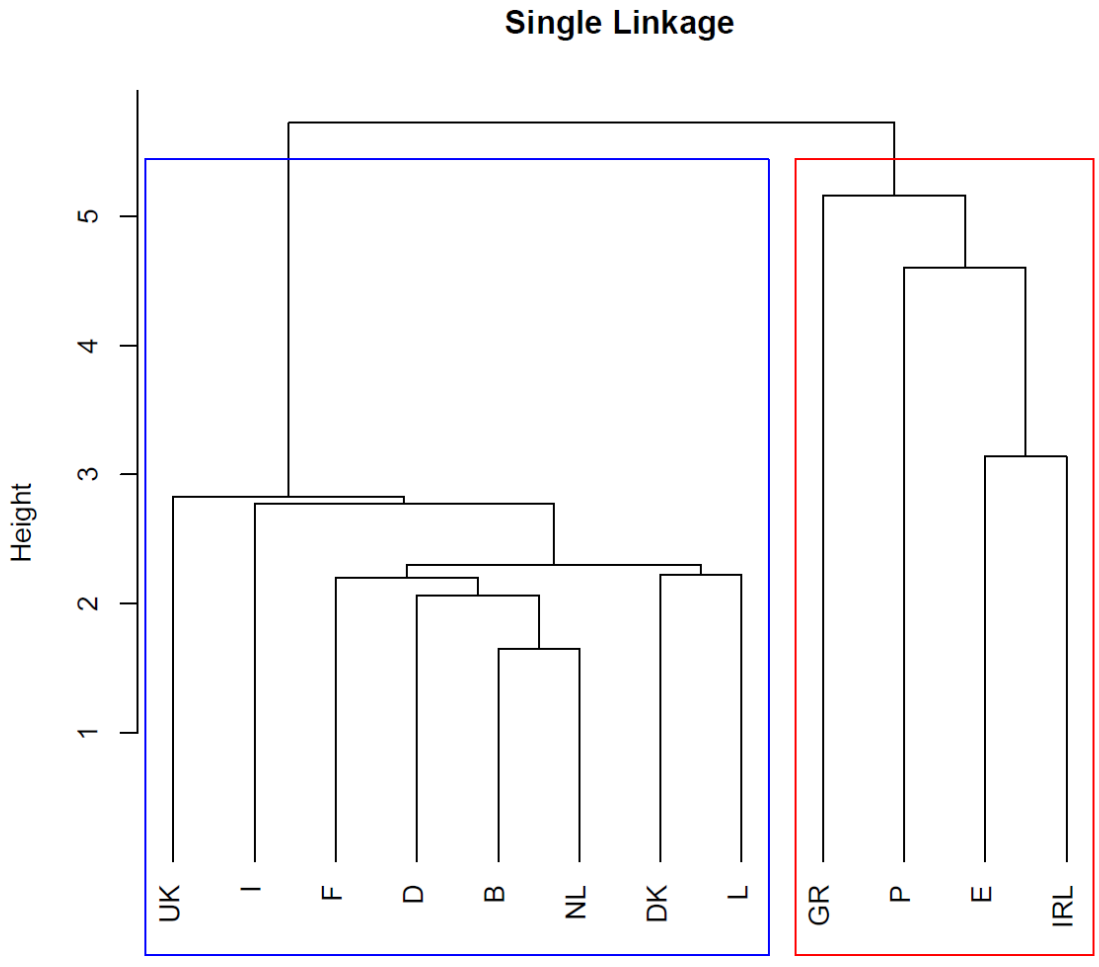
(Figure : J. Salmon, N. Verzelen)

CAH : exemple 2 (J. Salmon, N. Verzelen) -1-



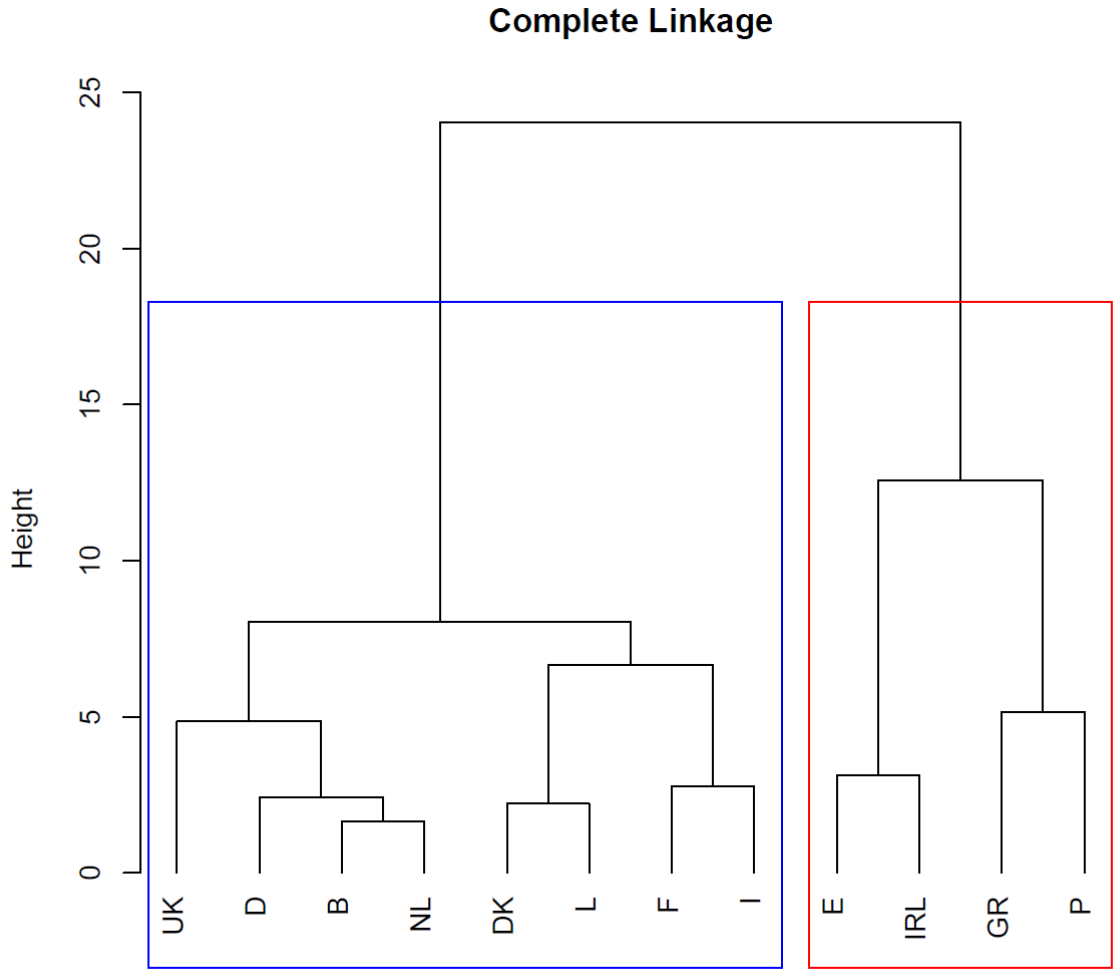
Jeu de données "Agriculture" dans l'union européenne en 1993.

CAH : exemple 2 -2-



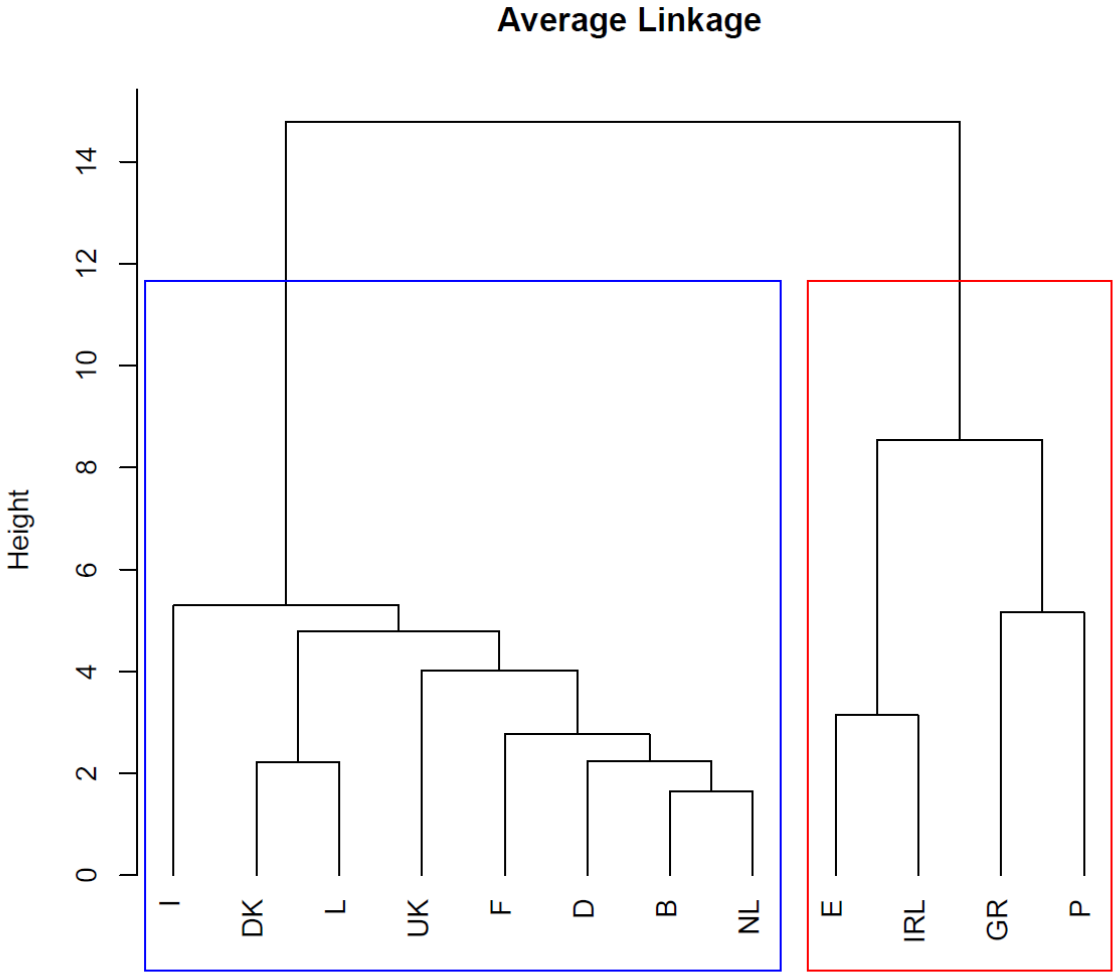
(Figure : J. Salmon, N. Verzelen)

CAH : exemple 2 -3-



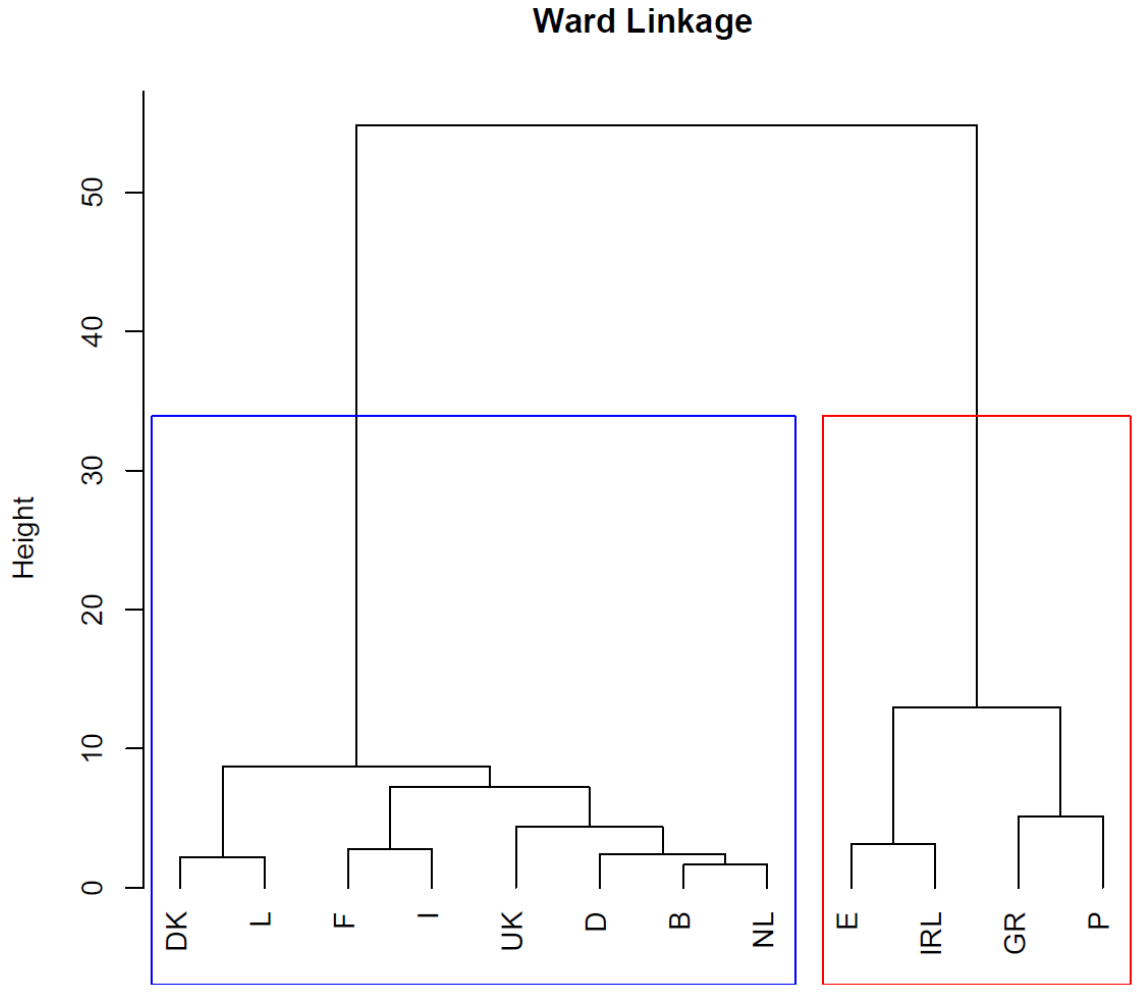
(Figure : J. Salmon, N. Verzelen)

CAH : exemple 2 -4-



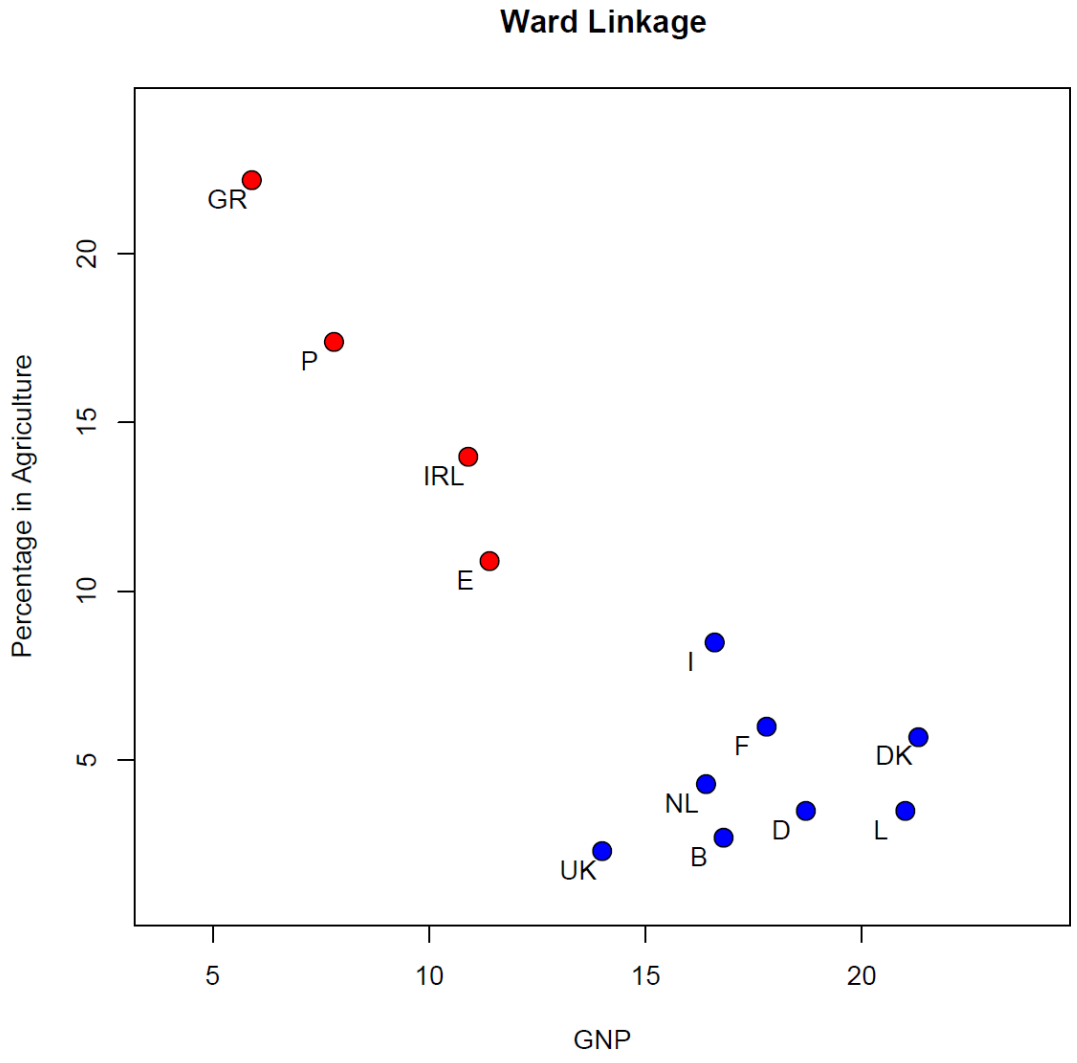
(Figure : J. Salmon, N. Verzelen)

CAH : exemple 2 -5-



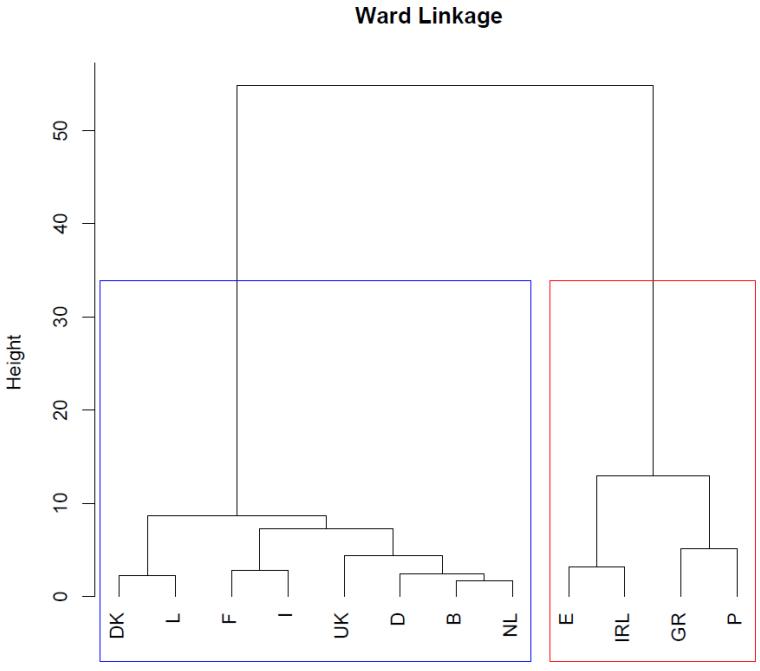
(Figure : J. Salmon, N. Verzelen)

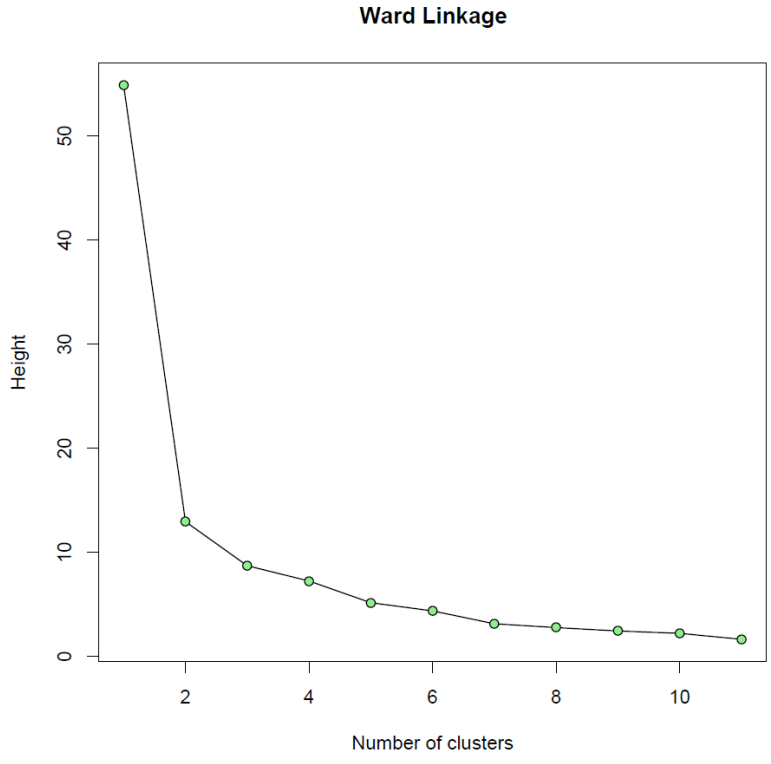
CAH : exemple 2 -5-



(Figure : J. Salmon, N. Verzelen)

CAH : exemple 2 -5-





(Figure : J. Salmon, N. Verzelen)

Choix de k : méthode du "coude".

CAH : complexité et remarques finales

Complexité :

- $\mathcal{O}(n^3)$ en implémentation naïve (n itérations sur matrice $n \times n$.).
- Meilleurs algorithmes en $\mathcal{O}(n^2 \ln n)$ voire $\mathcal{O}(n^2)$.

Remarques finales :

- Le choix de k est important : méthode du "coude".
- Introduite par J.P. Benzécri, Rennes 1982.

Méthode des k-moyennes

Algorithme des k-moyennes : principe

Construit la partition qui minimise la distance intra-cluster (inertie) :

$$\varepsilon(\pi_k) = \sum_{i=1}^{k} \sum_{x_j \in \mathcal{D}_i} ||x_j - m_i||^2, \tag{1}$$

avec m_i barycentre (ou centroïde) du cluster (ou groupe) i :

$$m_i = \frac{1}{|\mathcal{D}_i|} \sum_{x_j \in \mathcal{D}_i} x_j \tag{2}$$

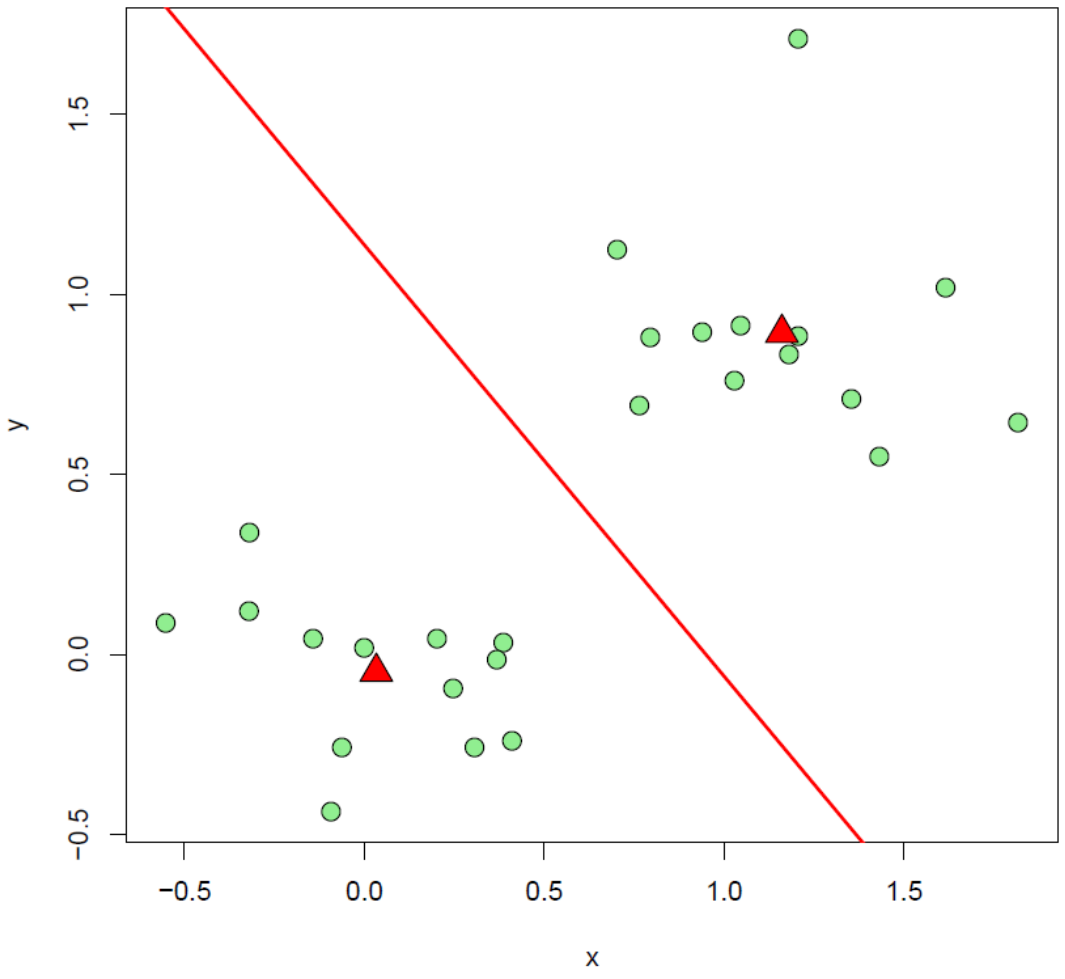
L'algorithme construit :

$$\widehat{\varepsilon}_k \in \underset{\pi_k = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}}{\arg\min} \varepsilon(\pi_k).$$

Problème NP-difficile $\Rightarrow$ obligation d'une méthode de résolution approchée.

k-means $\neq$ k-nn !!!!!

Algorithme des k-moyennes : exemple



(Figure : J. Salmon, N. Verzelen)

Résolution approchée du problème NP-difficile

Formation cluster : chaque donnée affectée au centroïde le + proche.

Algorithme de Lloyd (1957)

- Affecter chaque point au cluster de plus proche centre m_i.
- Ré-estimer les centres selon la nouvelle répartition.
- Itérer jusqu'à convergence

Complexité : $\mathcal{O}(n(k+1))$.

Converge vers un minimum local seulement.

$\Rightarrow$ En pratique, on lance plusieurs fois l'algo avec $\neq$ initialisations.

Heursitique pour choisir k : méthode du coude (Elbow). Quand la décroissance devient moins franche.

Algorithme des k-moyennes : géométrie des classes

Les centres induisent une partition de Voronoi de $\mathbb{R}^d$.

$$V_i = \{x \in \mathbb{R}^d : ||x - m_i|| \leq \min_{k \neq i} ||x - m_k||\}$$

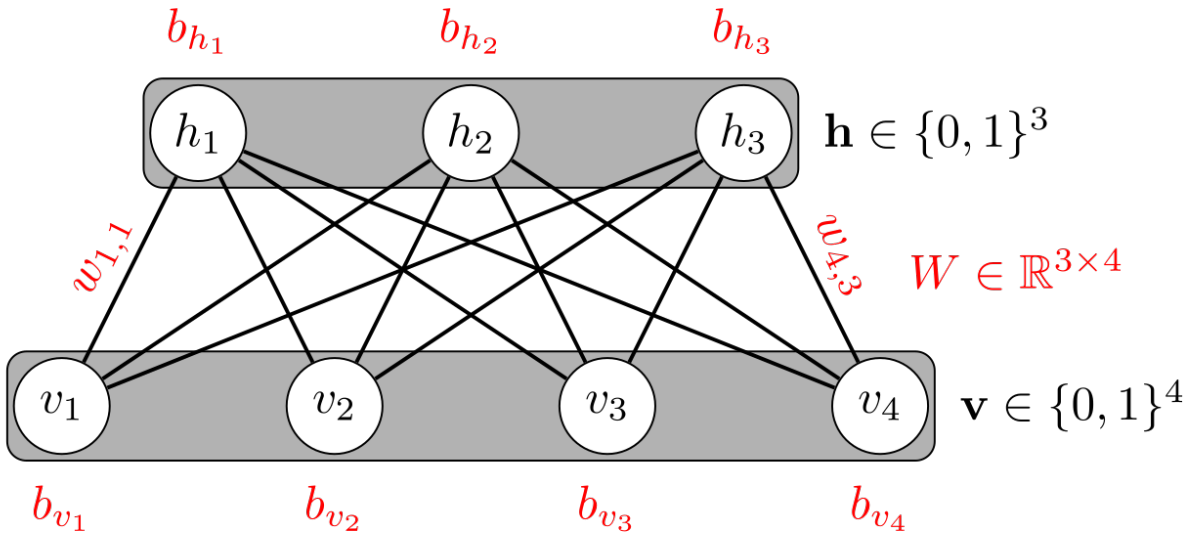
V_i est une cellule de Voronoi (convexe).

Intéressant à lire : https://freakonometrics.hypotheses.org/19156.

Méthodes à base de réseaux de neurones

Machines de Boltzmann restreintes (RBM) -1-

- Smolensky 1986, Hinton 2005.
- Réseaux de neurones binaires à deux couches (graphe biparti).
- Estimation d'une distribution de probabilités empirique.
- Apprentissage par l'algorithme de Constrastive Divergence (CD).



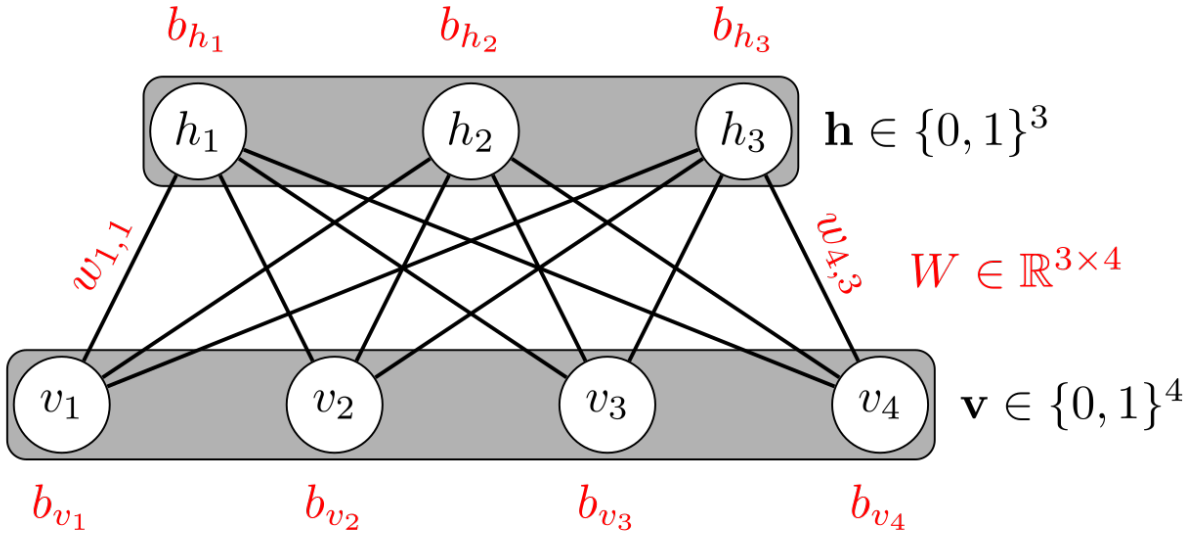
Machines de Boltzmann restreintes (RBM) -2-

Énergie et distribution de Gibbs :

$$E(v,h) = -\sum_{i,j} w_{ij} v_i h_j - b^T v - b'^T h = -S^T W S - b^T S$$

$$\mathbb{P}(v,h) \propto e^{-E(v,h)}$$

avec $s = (v,h)$ vecteur binaire regroupant neurones visibles et cachés.



Modèles génératifs

- Modèle probabiliste : stochastic parrot (Shannon, Mathematical Theory of Communication, 1948).
- Approche deep-learning : réseau antagoniste génératif (GAN), de Ian Goodfellow 2014.
- 2 réseaux de neurones antagonistes en compétition via un problème de théorie des jeux (jeu à somme nulle).
- Chat-GPT (openAI) : 2022.
- Chat-GPT : "écris-moi un cours d'apprentissage statistique pour les mastères spécialisés Data-Science".
- Meow generator.

Coût énergétique des LLM...

Spectral clustering

Partitionnement spectral

- Les méthodes géométriques (dont k-means) ne trouvent que des clusters "en boule".
- Ne tiennent pas compte d'une éventuelle structure.
- Même problème pour les estimations de densité.

$\Rightarrow$ Spectral clustering :

- On projète les données sur les nœuds d'un graphe pondéré.
- Les arêtes modélisent la similarité entre les données.
- Le poids de chaque arête est proportionnel à la distance (dissimilarité) entre données.

Données avec structure latente -1-

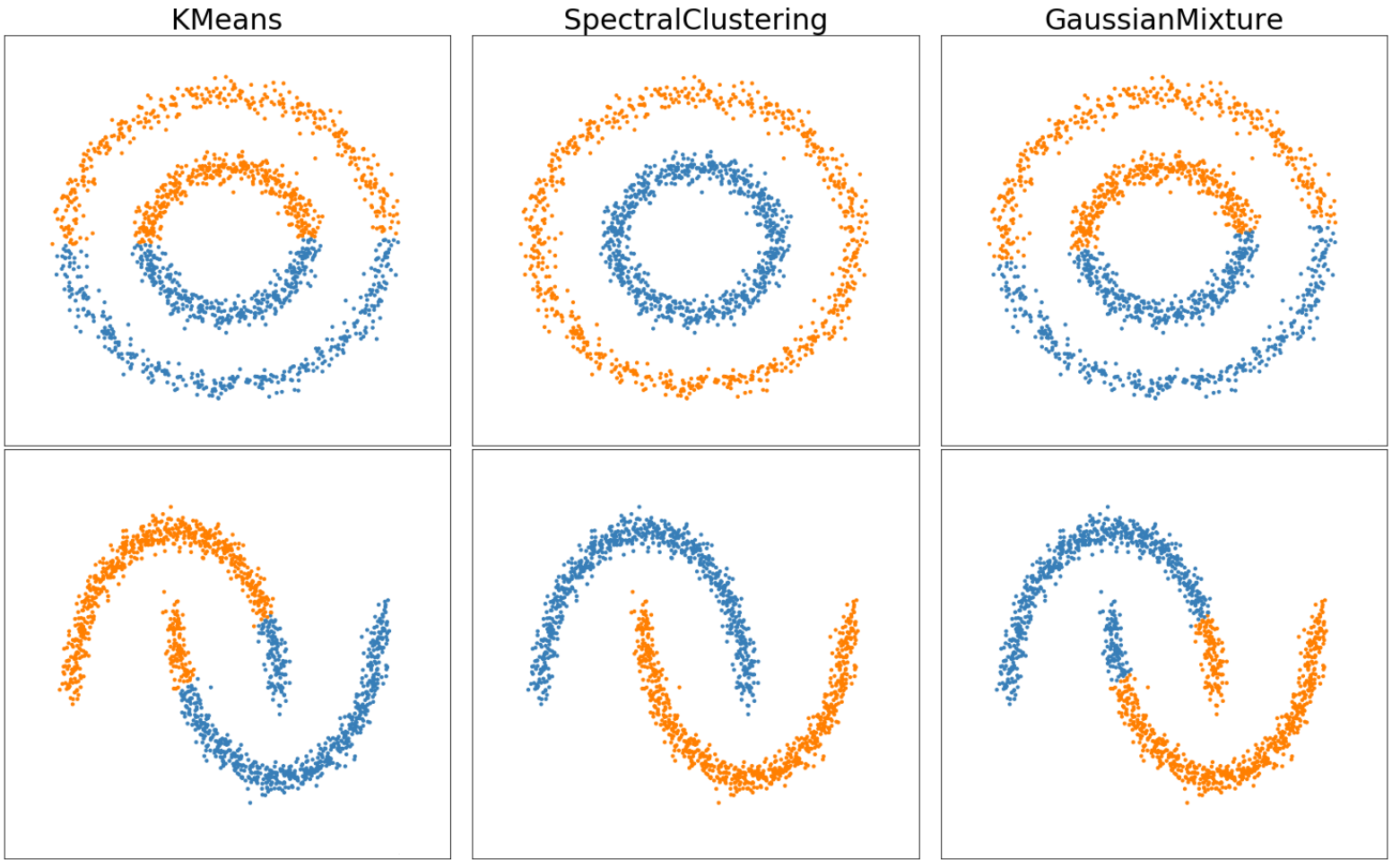


Figure : Scikit-Learn documentation.

Données avec structure latente -2-

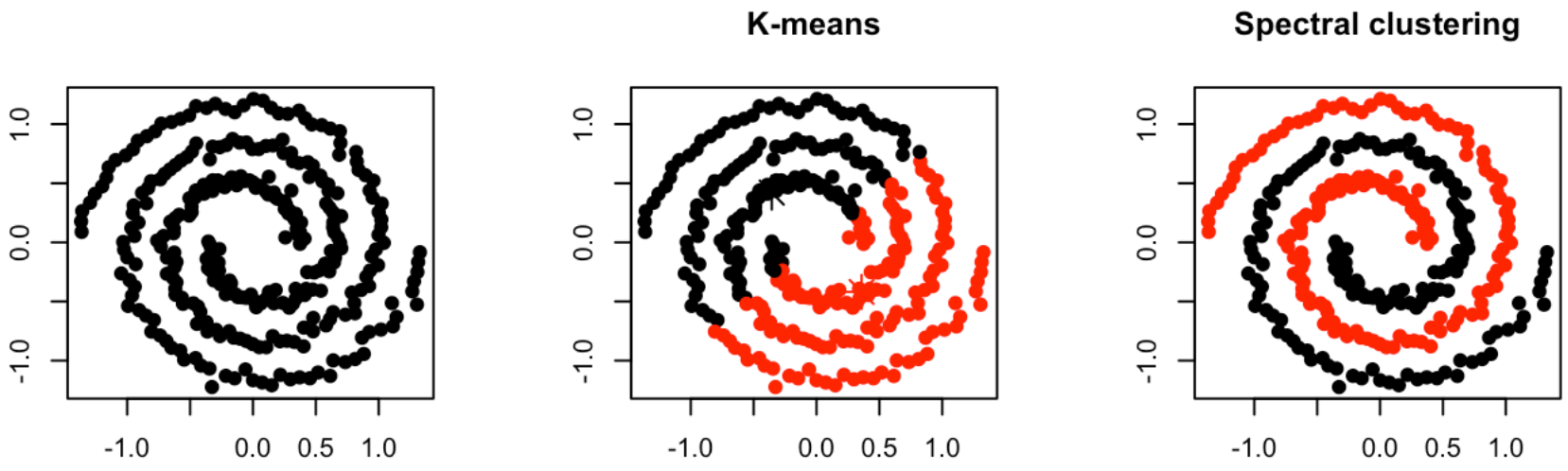


Figure : Neerja Doshi.

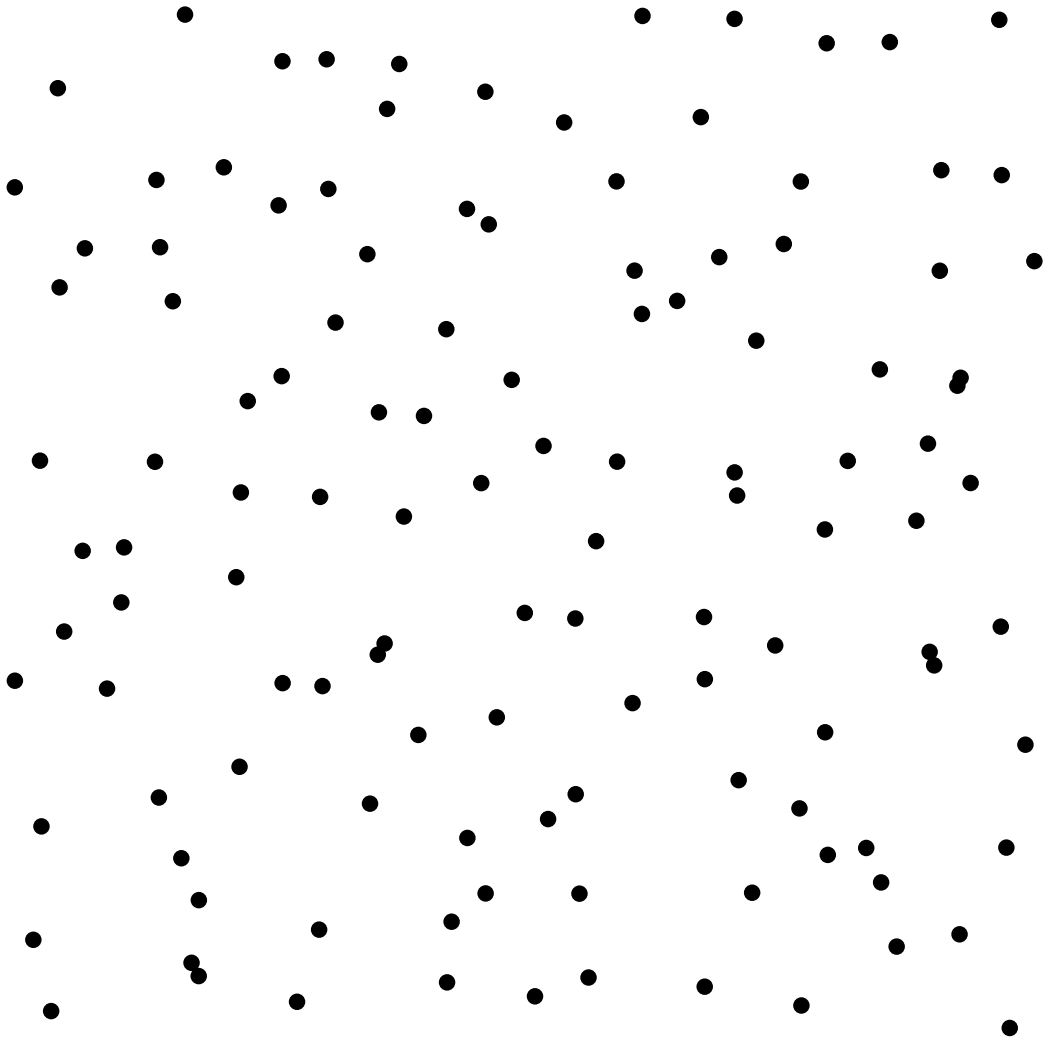
Partitionnement spectral

- Les méthodes géométriques (dont k-means) ne trouvent que des clusters "en boule".
- Ne tiennent pas compte d'une éventuelle structure.
- Même problème pour les estimations de densité.

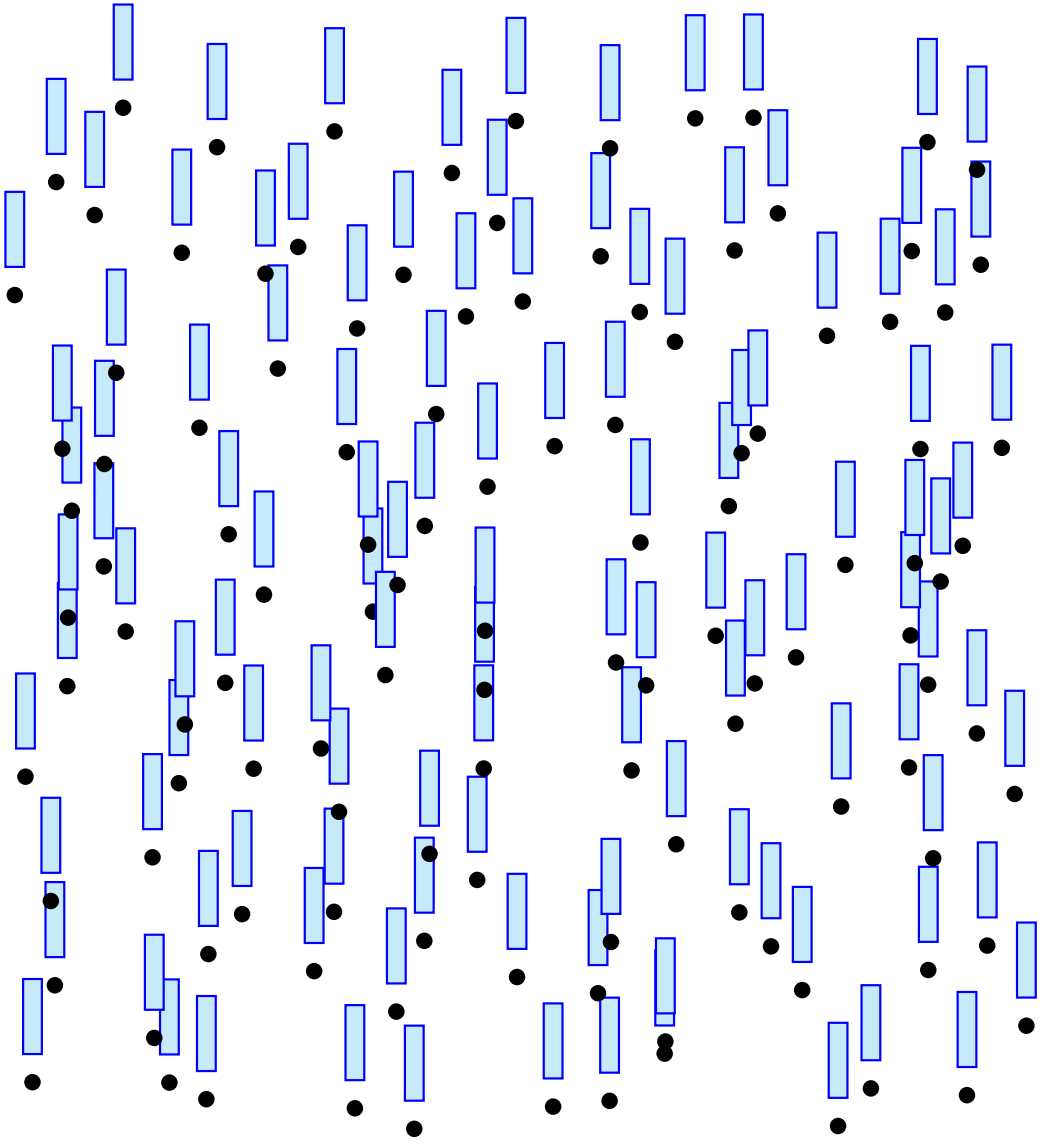
$\Rightarrow$ Spectral clustering :

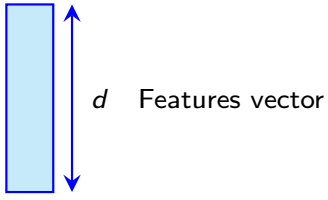
- On projète les données sur les nœuds d'un graphe pondéré.
- Les arêtes modélisent la similarité entre les données.
- Le poids de chaque arête est proportionnel à la distance (dissimilarité) entre données.

Données définies sur un graphe

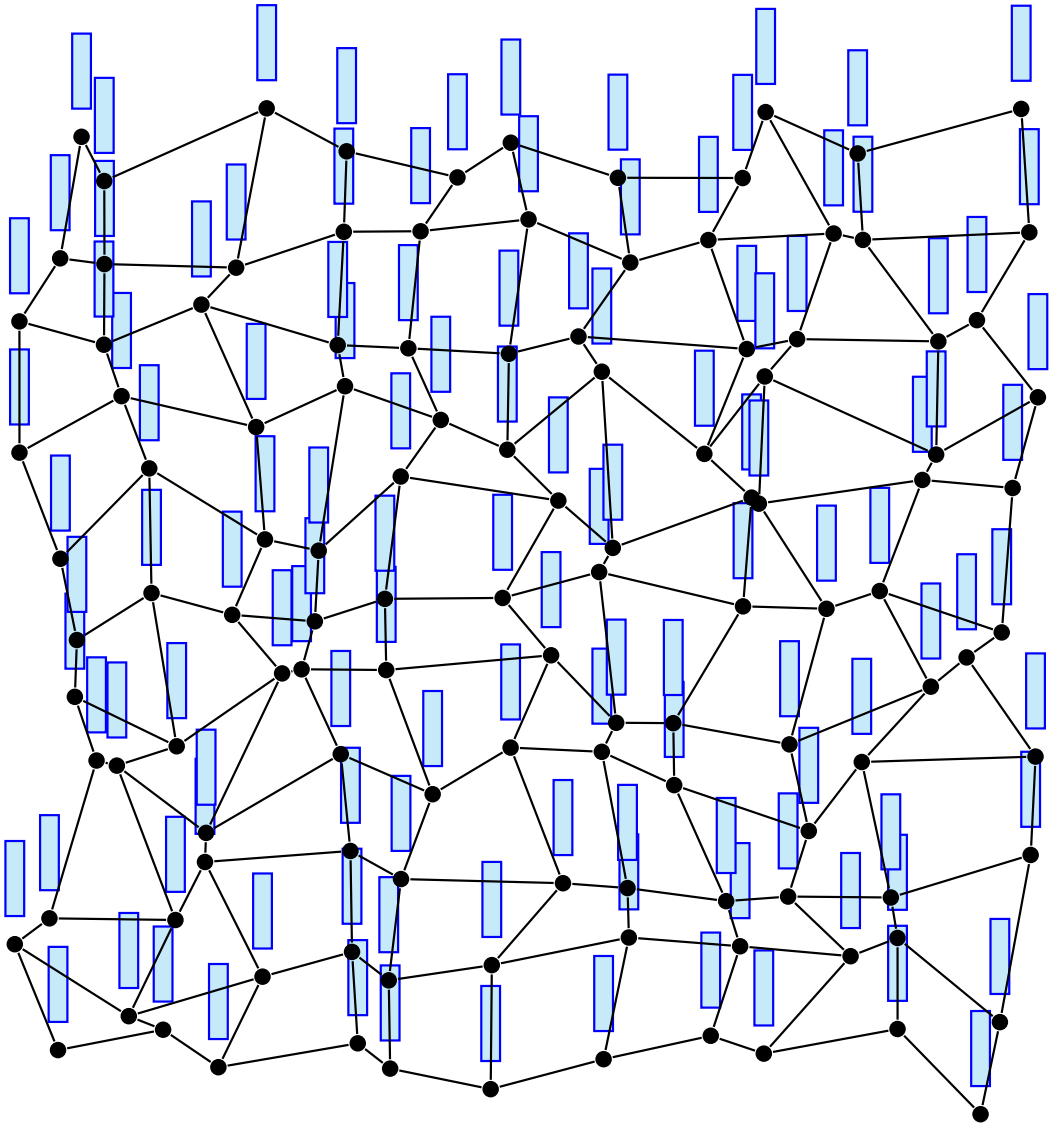


Données définies sur un graphe

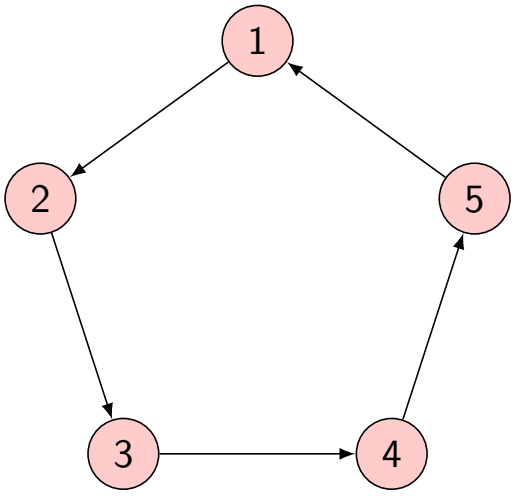


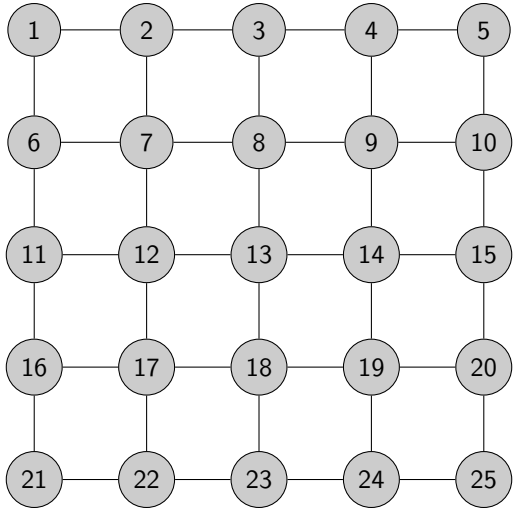


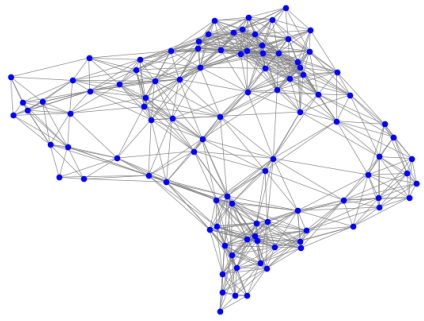
Données définies sur un graphe

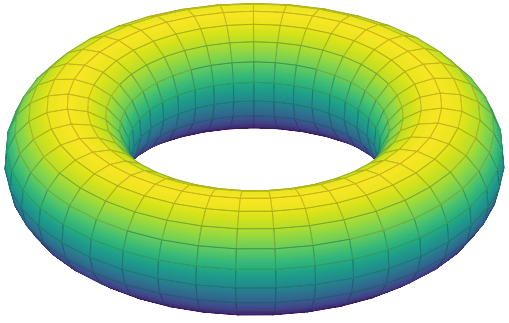


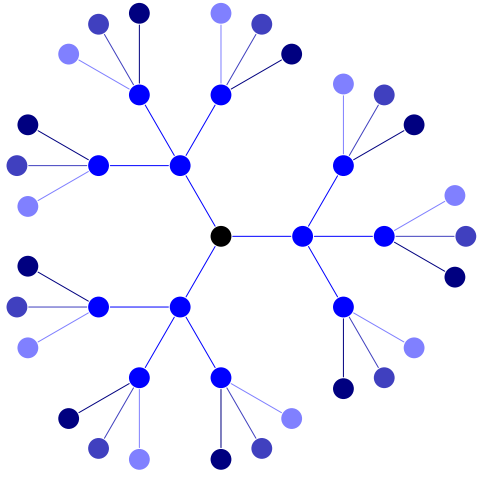
Rappels théorie des graphes : quelques exemples

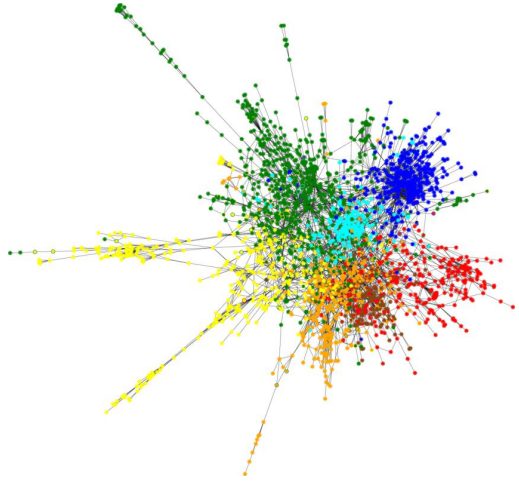






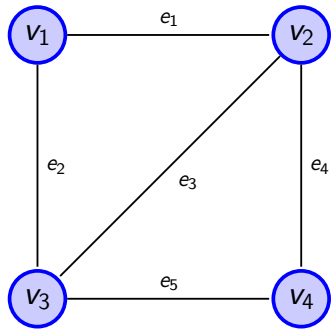






Rappels théorie des graphes : notations

- $G = (V, E)$ graphe.
- $V = (v_1, .., v_n)$ nœuds.
- $E = (e_1, .., e_m)$ arêtes $e = (v_i, v_j) = (i, j)$.
- $B \in \mathbb{R}^{n \times m}$ matrice d'incidence :
  $b_{ij} = \pm 1 \iff v_i \sim e_j$.
- $A \in \mathbb{R}^{n \times n}$ matrice d'adjacence $a_{ij} = 1 \iff v_i \sim v_j$.
- $W \in \mathbb{R}_+^{n \times n}$ si pondéré : w_{ij} poids de l'arête.
- D matrice des degrés: $d_{ii} = d(i) = \sum_{j=1}^{n} w_{ij}$.
- $L = D - W$ laplacien du graphe.
- $x_i \in \mathbb{R}^d$ donnée portée par le nœud v_i.
- Graphe de similarité : $w_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$.



$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}$$

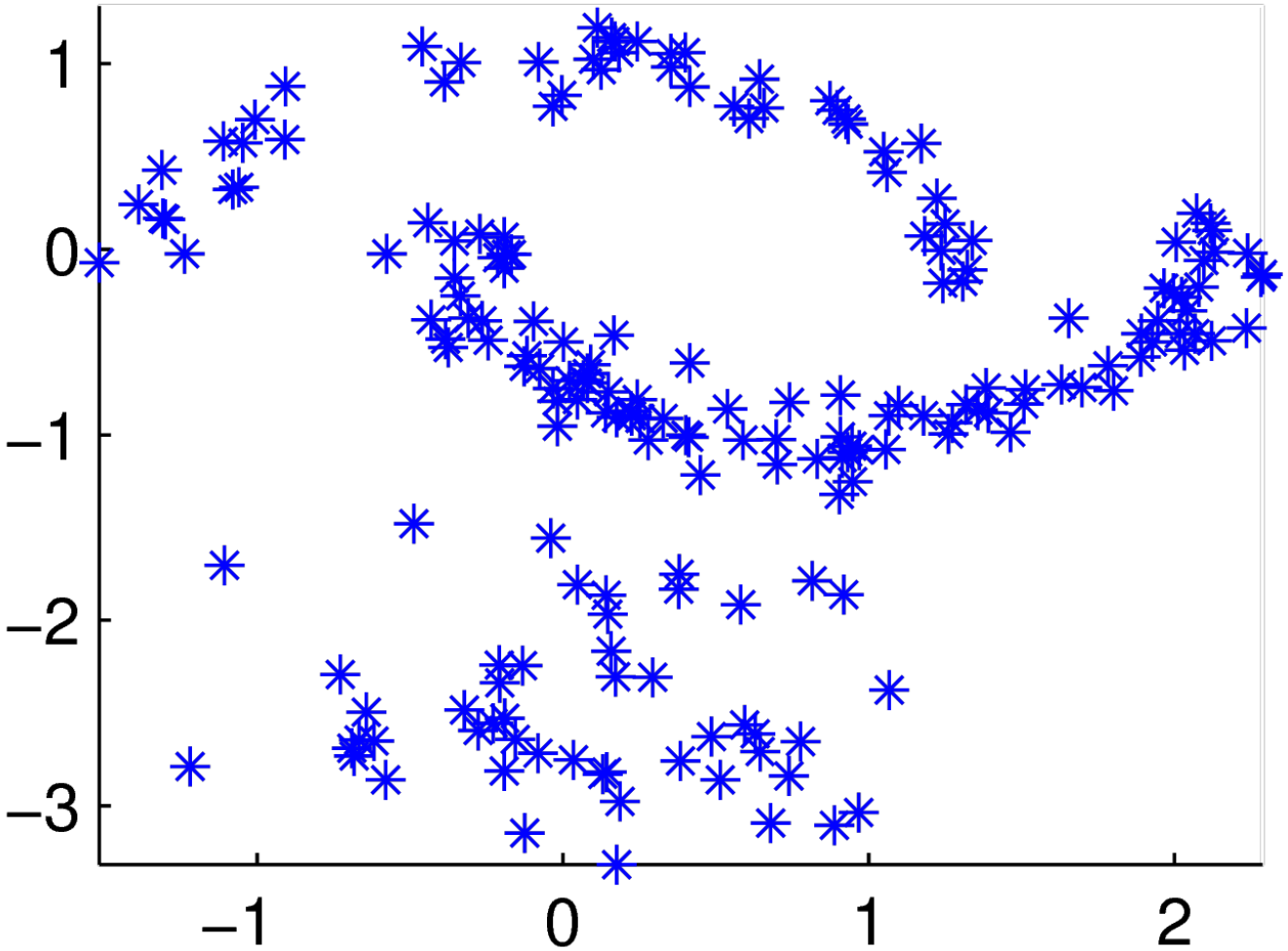
Graphes de similarité -1-

- Partant des n données on connecte chaque nœud à tous les autres.
- $\Rightarrow$ Graphe complet, nombre exponentiel d'arêtes.
- $\Rightarrow$ Il faut supprimer des arêtes.

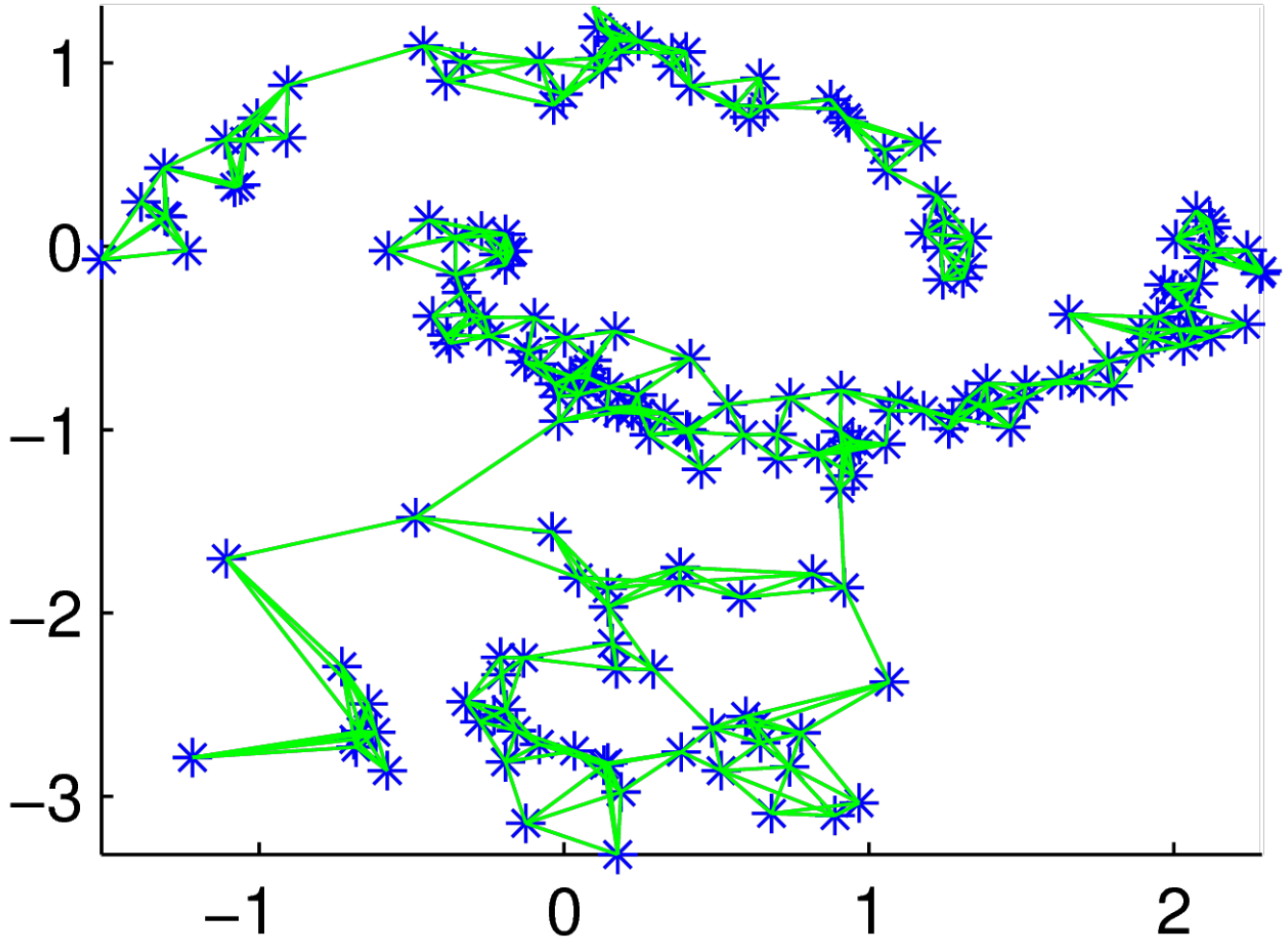
Plusieurs possibilités :

- Graphe de voisinage : on garde l'arête si distance $< \epsilon$ fixé.
- Graphe des ppv : on conserve les arêtes des k-ppv uniquement.
- Graphe des ppv symétriques : k-ppv sans tenir compte de l'orientation.

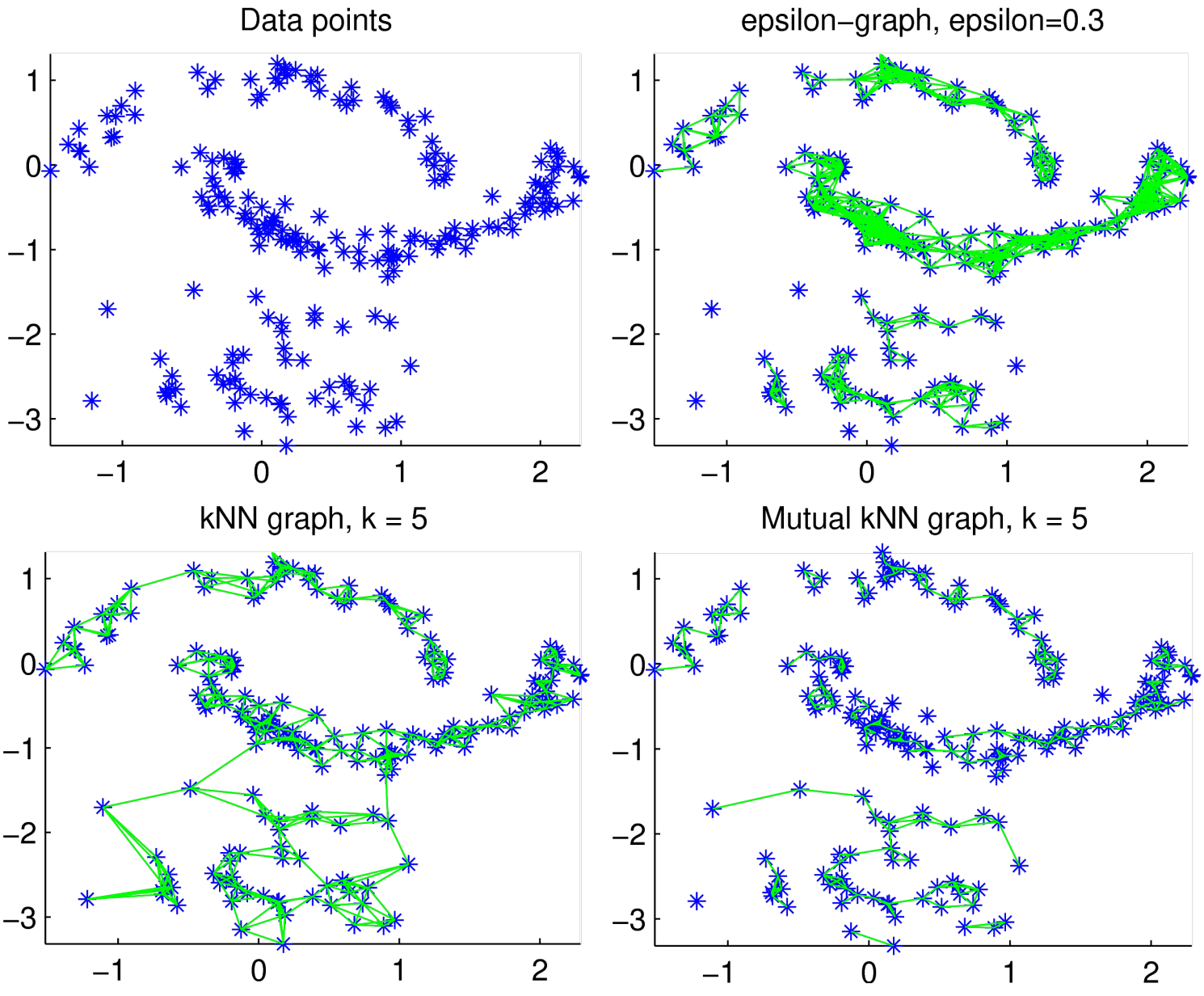
Graphes de similarité -2- (Ex. Von Luxburg 2007)



Graphes de similarité -3- (Ex. Von Luxburg 2007)



Graphes de similarité -4- (Ex. Von Luxburg 2007)



Rappels théorie des graphes : propriétés du laplacien

Si $\mathcal{G}$ non orienté, L symétrique,semi-défini, positif et diagonalisable.
Soit $\sigma(L) = \{\lambda_1 \leq ... \leq \lambda_s\}$ son spectre (vp).

- Coef (i, j) de A^l : nombre de chemins de longueur l allant de i à j.
- $\lambda_1 = 0 \in \sigma(L)$ de multiplicité k ssi $\mathcal{G}$ a k composantes connexes.
- Base de E_1 formée des vecteurs indicateurs $\mathbb{1}_{A_i}$ des composantes connexes.
- λ_1 connectivité algébrique. λ_1 grand $\Rightarrow$ graphe très connecté.
- Vecteur de Fiedler u_2 : vecteur propre (VP) associé à λ_2.
- dim $E_2 = 1$ et le signe des coordonnées de u_2 partitionne $\mathcal{G}$.
- "Mysteries around the graph Laplacian eigenvalue 4..."

$\Rightarrow A, W, L$ caractérisent $\mathcal{G}$ et contiennent toutes ses propriétés topologiques et algébriques. Le spectre de L est l'outil essentiel pour partitionner.

Partition du graphe et coupures

A, B deux sous-ensembles de V partitionnant G.
La frontière de A et B est une coupure du graphe.

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- Objectif : partitionner le graphe en deux clusters avec coupure de poids minimum.
- $\Rightarrow$ Facile à faire, mais pas concluant (nœud isolé).
- On peut normaliser la coupure pour obliger à une taille minimum.
- $\Rightarrow$ Problème NP-difficile.
- $\Rightarrow$ Relaxation continue du problème = spectral clustering.

$$\text{Ncut}(A, B) = \text{cut}(A, B) \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)$$

$$\text{vol}(A) = \sum_{i \in A} d_i$$

Coupure normalisée et laplacien -1-

Soit $x = (x_1, ..., x_n) \in \mathbb{R}^n$ (on suppose $d = 1$), avec :

$$x_i = \begin{cases} 1/\text{vol}(A) & \text{si } i \in A \\ 1/\text{vol}(B) & \text{si } i \in B \end{cases}$$

Alors

$$x^T L x = \sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2 = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)^2$$

$$x^T D x = \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)$$

$$\Rightarrow \text{Ncut}(A, B) = \frac{x^T L x}{x^T D x}$$

Coupure normalisée et laplacien -2-

Pour trouver la coupure minimale, on doit résoudre :

$$\underset{x:\ x^T D\mathbb{1}=0}{\arg\min} \frac{x^T L x}{x^T D x}$$

- Caractérisation variationnelle des vp (quotient de Rayleigh).
- La solution est le vecteur de Fiedler u_2 : $Lu_2 = \lambda_2 D u_2$.
- Analogie avec l'ACP : VP associés aux plus grandes vp.
- Se généralise (assez) facilement à k clusters.

Coupure normalisée et laplacien -3-

Que représente le laplacien ?

- x variable indicatrice de l'appartenance à un cluster.
- x doit être orthogonal au vecteur $\mathbb{1}$ (noyau de L).
- $x^T Lx$ représente l'énergie du signal x.
- L est un opérateur de moyennage, de lissage, de courbure...

Relaxation d'un problème NP-difficile :

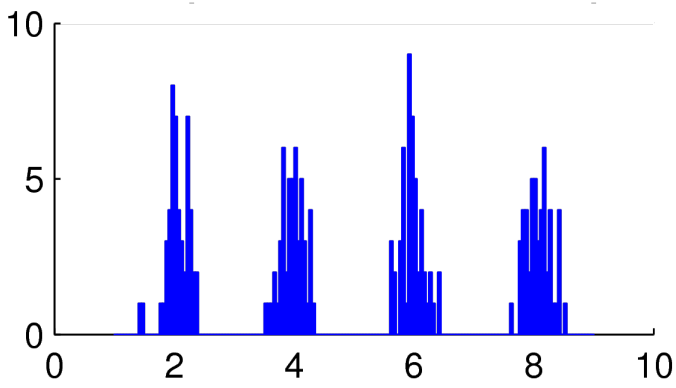
- x : coordonnées 1 ou 0.
- Problème NP-difficile.
- On relâche le problème : coordonnées de x dans $\mathbb{R}$.
- x_i est alors le degré d'appartenance au cluster i.
- Les coordonnées des VP de L mesurent l'appartenance aux clusters.

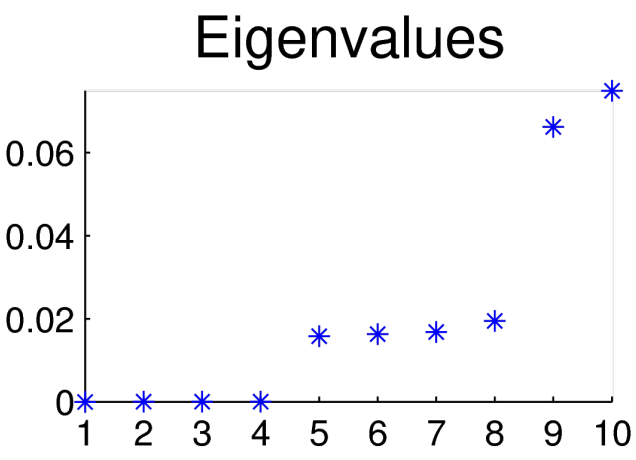
Algorithme de partitionnement spectral

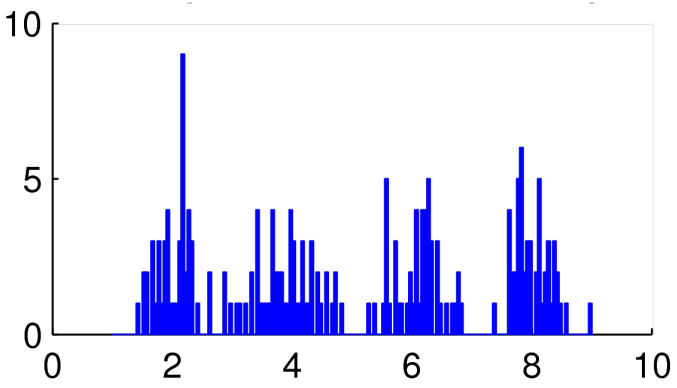
Pour le laplacien non normalisé.

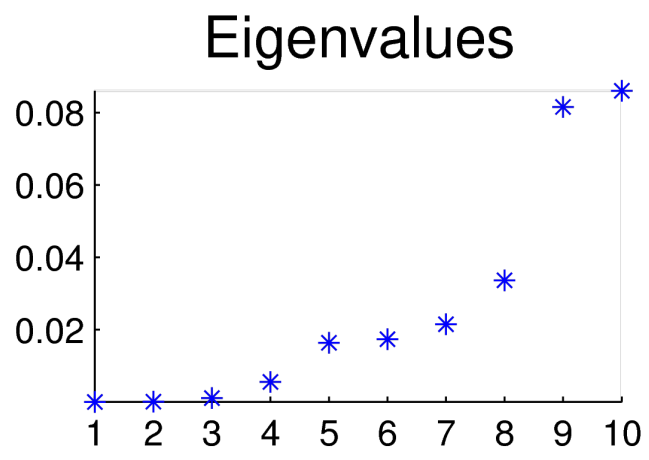
- À l'initialisation : nombre de clusters k donné (comment ?).
- Calculer les distances de similarité entre les données.
- En déduire les pondérations w_{ij} et le graphe de similarité G.
- Calculer la matrice laplacienne L.
- Calculer les k VP $u_1, ..., u_k$ associés aux k plus petites vp.
- Construire la matrice colonne $U = (u_1, ..., u_k) \in \mathbb{R}^{n \times k}$.
- Regrouper les lignes en k groupes avec l'algorithme des k-moyennes.

Exemple -1- (Von Luxburg 2007)

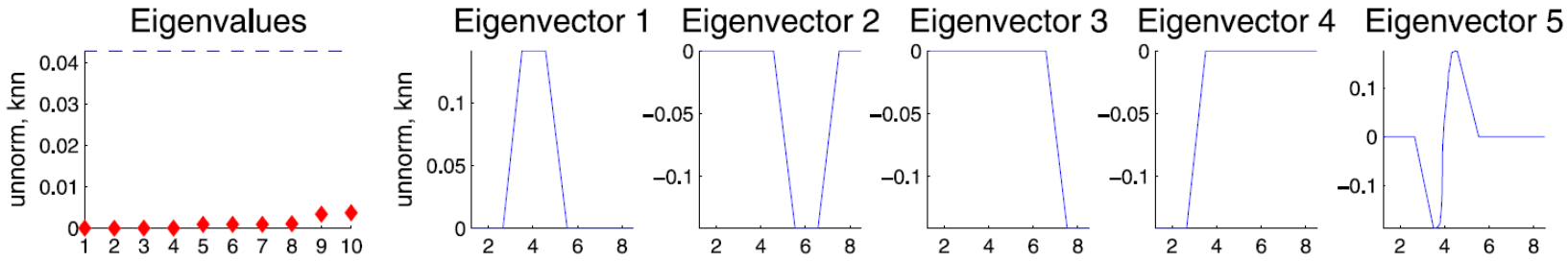








Exemple -2- (Von Luxburg 2007)



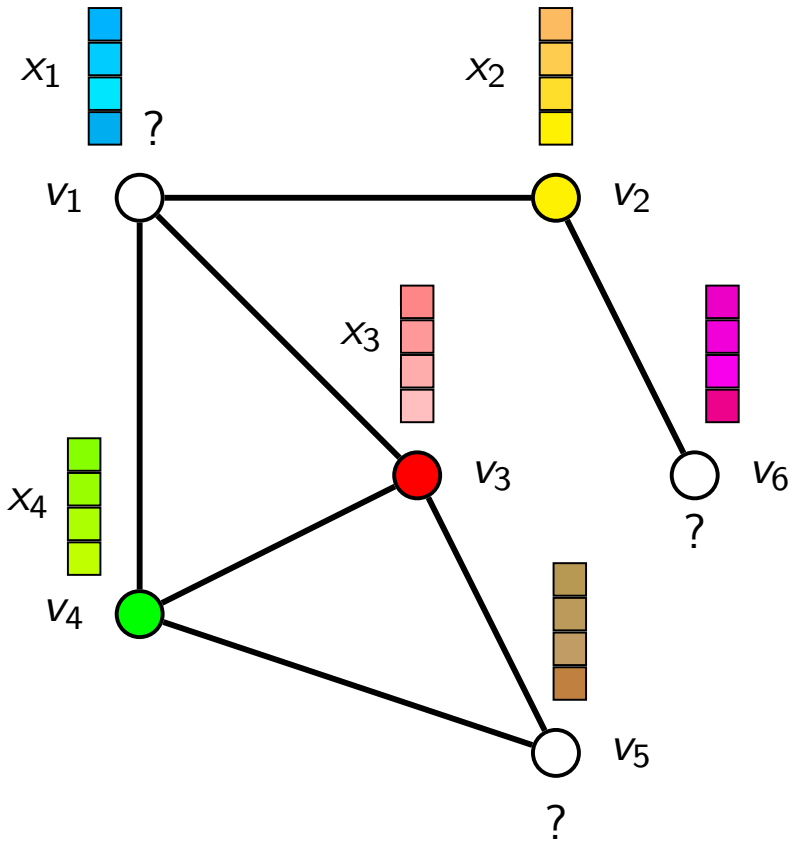
Exemple : $n = 200$ données réelles x_i générées par un mélange de 4 gaussiennes ($\sigma = 1$).

À lire : A tutorial on spectral clustering, Ulrike Von Luxburg, 2007, Stat. Comput.

Apprentissage semi-supervisé : un exemple

Réseau neuronal sur graphe (Gnn) -1-

- G: graphe avec données étiquettées portées par les nœuds.
- Labels $y_i = f\left(x_i, (x_j, y_j)_{j \in \partial v_i}\right)$.
- Certains labels inconnus : ?.
- Tâche : retrouver tous les labels en exploitant le voisinage.



Réseau neuronal sur graphe (Gnn) -2-

GNN = graphe + algorithme de "Message Passing".

Formulation mathématique :

$G = (V, E)$, $X \in \mathbb{R}^{n \times d}$ données, $x_v \in \mathbb{R}^d$ ligne de X correspondant au nœud $v \in V$.

$$\begin{cases} h_v^0 = x_v, \\ h_v^{l+1} = \phi_l \left(h_v^l, \sum_{u \sim v} \hat{A}_{uv} \psi_l \left(h_u^l \right) \right). \end{cases}$$

$$H_{l+1} = \sigma \left(D^{-1/2} \hat{A} D^{-1/2} H_l W_l \right),$$

$H_l = \left(h_v^l \right)_v \in \mathbb{R}^{n \times d}$, $D = \mathrm{diag}(d_{ii})$, $d_{ii} = \sum_j \hat{A}_{ij}$, W_l poids, $\sigma = \phi_l$ fonction d'activation (ReLU, sigmoïde), $\psi_l = \mathrm{Id}$.