Apprentissage non supervisé
Classification spectrale - spectral clustering

Master "Machine Learning for Data Science", Paris V

Allou Samé
allou.same@ifsttar.fr

2017/2018

Plan

Introduction

Objectifs

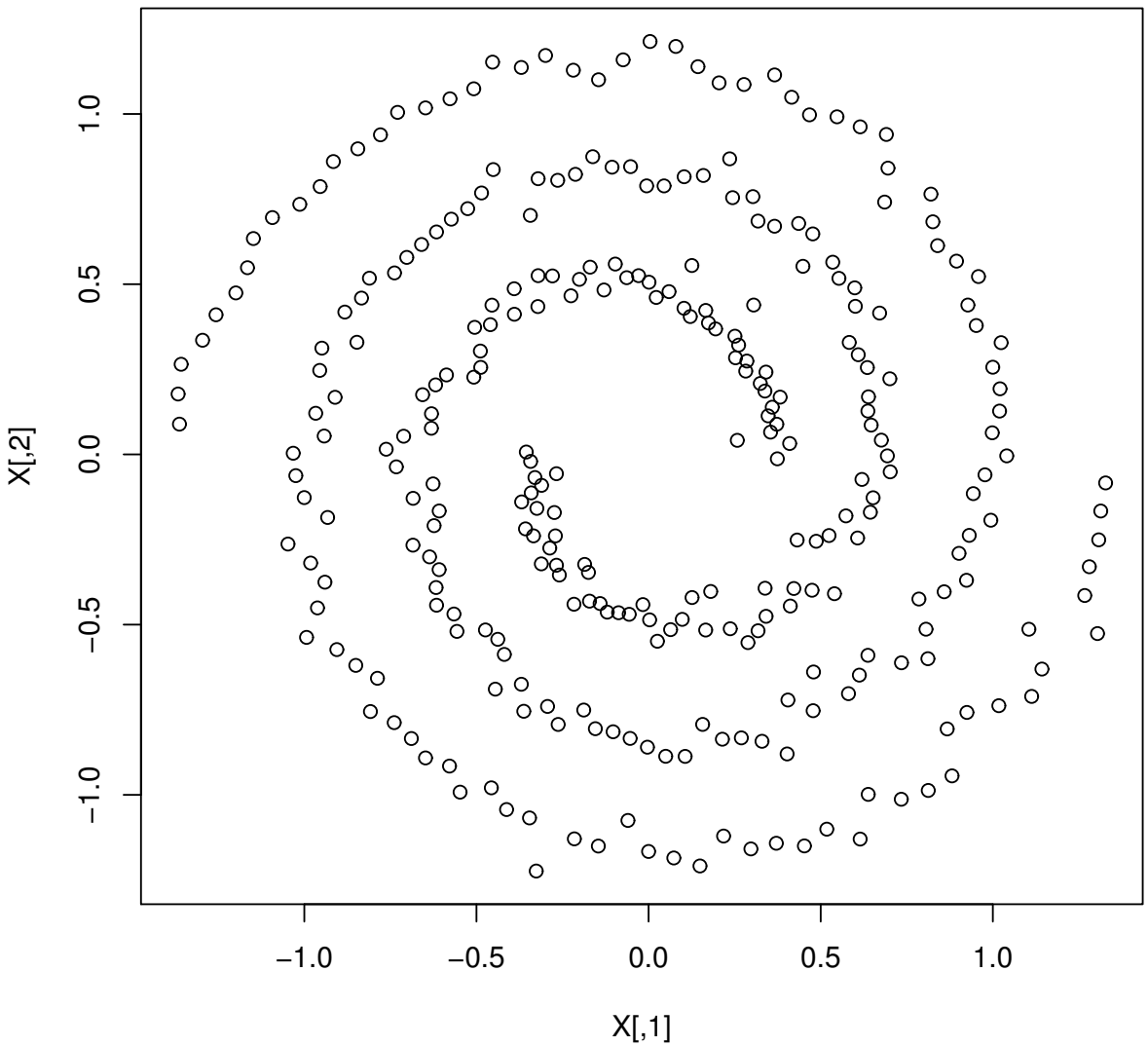
Tout comme les autres méthodes de clustering par partitionnement, l'objectif de la classification spectrale est de trouver une partition d'un ensemble de données $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ en K classes mais en se basant sur l'analyse spectrale d'un graphe de similarité

Avantages

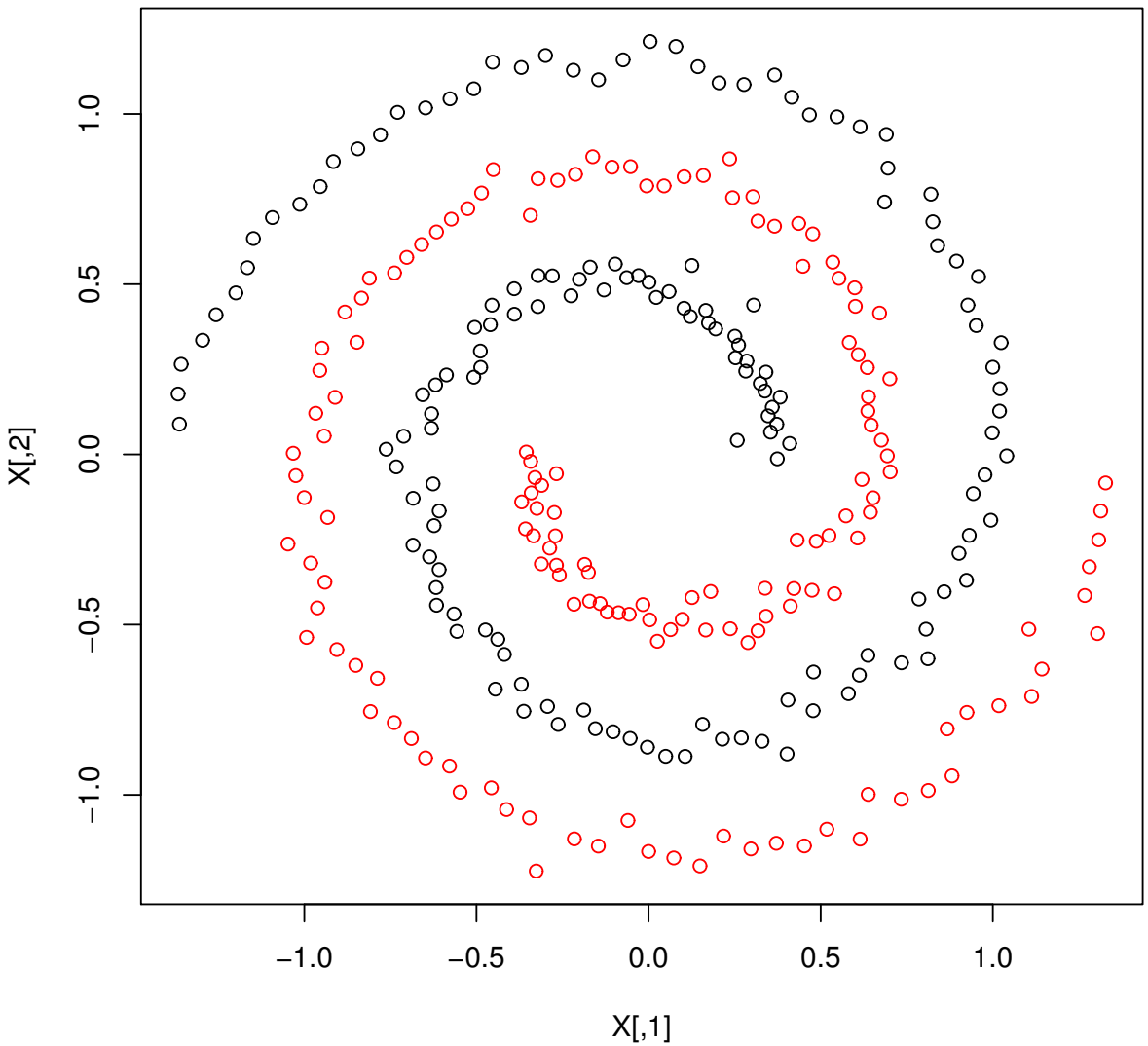
Le clustering spectral peut présenter quelques avantages par rapport aux méthodes étudiées dans les chapitres précédents :

- sa simplicité d'implémentation : essentiellement basée sur de l'algèbre linéaire (calcul de valeurs propres et vecteurs propres)
- sa capacité à extraire des classes non linéairement séparables (exemple de classes en forme de spirale) ; ce qui peut présenter un intérêt dans certaines applications comme la vision par ordinateur

Exemple de classes de séparation complexe



Exemple de classes de séparation complexe



Graphe de similarité

- On suppose disposer d'un ensemble de données $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ et d'une matrice de similarité $W = (w_{ij})_{1 \leq i,j \leq n}$ de taille $n \times n$
- On associe la matrice de similarité à un graphe non orienté
  - W est aussi appelée matrice d'adjacence du graphe
  - Les sommets du graphe sont les n observations à classer
  - Il existe une arête entre deux nœuds ssi $w_{ij} > 0$
  - Les w_{ij} sont les poids ou valuations des arêtes
  - Le graphe étant non orienté, on a $w_{ij} = w_{ji}$
  - Matrice des degrés $D \in \mathbb{R}^{n \times n}$: matrice diagonale dont les éléments sont les degrés des sommets : $d_i = \sum_{j=1}^{n} w_{ij}$
  - Soit $A \subset \Omega$. On définit : $|A| = cardinal(A)$ et $vol(A) = \sum_{\mathbf{x}_i \in A} d_i$
  - D'après la théorie des graphes, les composantes connexes du graphe définissent une partition de Ω

Exemple de graphes de similarités utilisés

Similarité basée sur un seuil ($\varepsilon-$neighborhood graph)

Deux points sont connectés si la distance les séparant est inférieure à un seuil ε.

Similarités basées sur les plus proches voisins ($k-$nearest neighborhood graph)

- $\mathbf{x}$ et $\mathbf{y}$ sont connectés si $\mathbf{x}$ fait partie des m plus proches voisins de $\mathbf{y}$ ou si $\mathbf{y}$ fait partie des m plus proches voisins de $\mathbf{x}$.
- $\mathbf{x}$ et $\mathbf{y}$ sont connectés si $\mathbf{x}$ fait parties des m plus proches voisins de $\mathbf{y}$ et si $\mathbf{y}$ fait partie des m plus proches voisins de $\mathbf{x}$ (plus proches voisins mutuels).

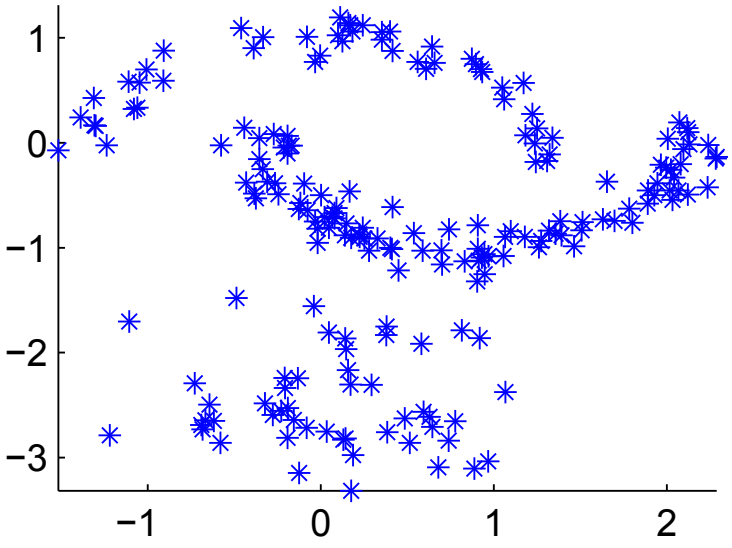
Similarité gaussienne (graphe complet)

Graphe complet dont les poids sont donnés par $w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$.

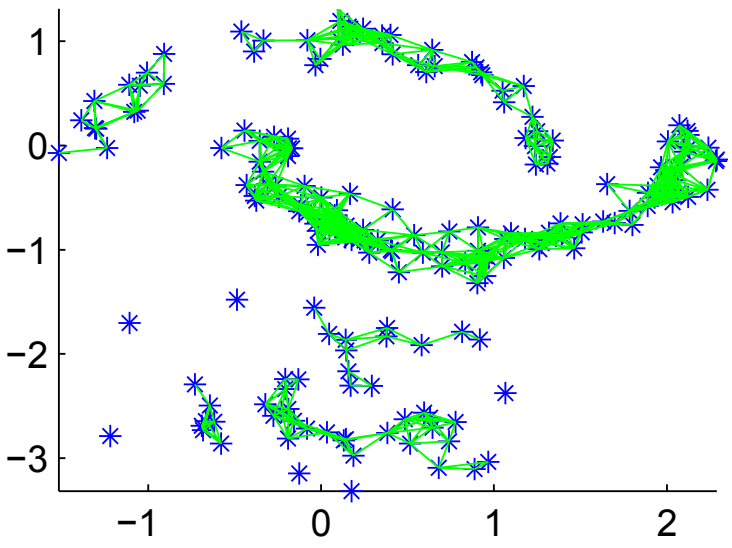
Le paramètre σ contrôle la "largeur" des voisinages, tout comme le paramètre ε ci-dessus

Illustrations des graphes de similarité

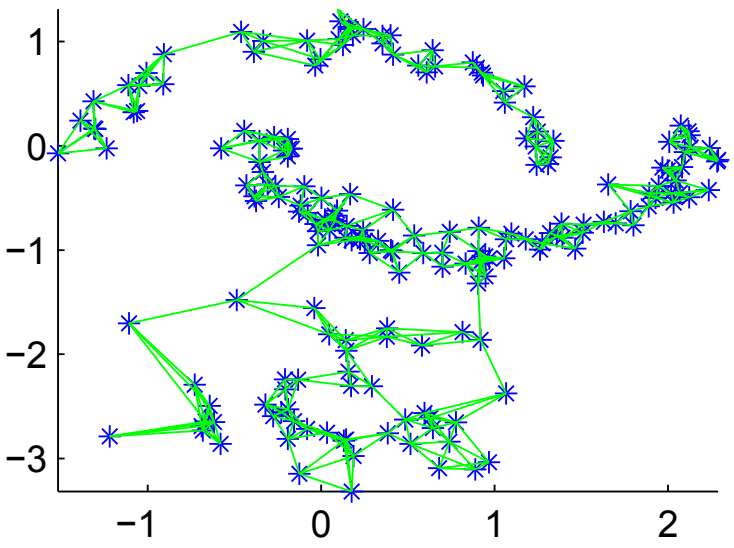
Données (3 classes)



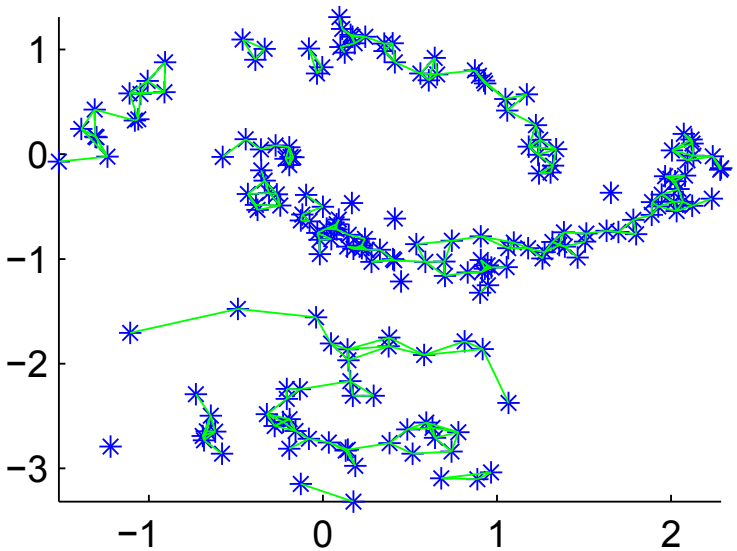
$\varepsilon-$graph ($\varepsilon = 0.3$)



kppv ($k = 5$)



kppv mutuels $k = 5$



Matrice Laplacienne

Matrice Laplacienne non normalisée

$$L = D - W$$

Propriétés

(1) $f' L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \qquad \forall f \in \mathbb{R}^n.$

(2) L est symétrique semi-définie positive.

(2) L possède n valeurs propres réelles $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

(3) La plus petite valeur propre de L est 0. Elle est associée au vecteur propre $\mathbb{1}_n = (1, \ldots, 1)^T$.

(4) La multiplicité de la valeur propre $\lambda_1 = 0$ est le nombre de composantes connexes $P_1, \ldots, P_K$ du graphe

Matrices Laplaciennes normalisées

$$\begin{aligned} L_{sym} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \\ L_{rw} &= D^{-1} L = I - D^{-1} W \end{aligned}$$

Propriétés

(1) pour tout vecteur $f \in \mathbb{R}^n$,

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}})^2$$

(2) λ est valeur propre de L_{rw} associée au vecteur propre u ssi
λ est valeur propre de L_{sym} associée au vecteur propre $D^{1/2}u$

(3) λ est valeur propre de L_{rw} associée au vecteur propre u ssi
$L\,u = \lambda\, D\, u$ (λ et u : valeur et vecteur propre généralisés)

Matrices Laplaciennes normalisées

Propriétés

(4) 0 est valeur propre de L_{rw} (resp. L_{sym}) associée au vecteur propre $\mathbb{1}$ (resp. $D^{1/2}\mathbb{1}$)

(6) L_{sym} et L_{rw} sont symétriques semi-définies positives et possèdent n valeurs propres positives ou nulles

(5) La multiplicité de la valeur propre 0 de L_{rw} (resp. L_{sym}) est le nombre de composantes connexes $P_1, \ldots, P_K$ du graphe

Algorithme de classification spectrale non normalisé

1. Construire le graphe de similarité $W \in \mathbb{R}^{n \times n}$.
2. Calculer la matrice Laplacienne L.
3. Calculer les valeurs propres et les vecteurs propres orthonormés de L.
4. Former la matrice $U \in \mathbb{R}^{n \times K}$ dont les colonnes sont les K vecteurs propres associés aux K plus petites valeurs propres de L.
5. Pour $i = 1, \ldots, n$, construire le vecteur $\mathbf{y}_i \in \mathbb{R}^K$ obtenu par transposition de la i^{e} ligne de U.
6. Partitionner l'ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ en K classes $C_1, \ldots, C_K$ par l'algorithme des k-means.
7. En déduire la partition $(P_1, \ldots, P_k)$ de Ω telle que $P_k = \{\mathbf{x}_i \mid \mathbf{y}_i \in C_k\}$.

Algorithme de classification spectrale normalisé (Shi et Malik, 2000)

1. Construire le graphe de similarité $W \in \mathbb{R}^{n \times n}$.
2. Calculer la matrice Laplacienne normalisée L_{rw}.
3. Calculer les valeurs propres et les vecteurs propres orthonormés de L_{rw}.
4. Former la matrice $U \in \mathbb{R}^{n \times K}$ dont les colonnes sont les K vecteurs propres associés aux K plus petites valeurs propres de L_{rw}.
5. Pour $i = 1, \ldots, n$ construire le vecteur $\mathbf{y}_i \in \mathbb{R}^K$ obtenu par transposition de la i^{e} ligne de U.
6. Partitionner l'ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ en K classes $C_1, \ldots, C_K$ par l'algorithme des k-means.
7. En déduire la partition $(P_1, \ldots, P_k)$ de Ω telle que $P_k = \{\mathbf{x}_i \mid \mathbf{y}_i \in C_k\}$.

Algorithme de classification spectrale normalisé (Ng, Jordan et Weiss, 2002)

1. Construire le graphe de similarité $W \in \mathbb{R}^{n \times n}$.
2. Calculer la matrice Laplacienne normalisée L_{sym}.
3. Calculer les valeurs propres et les vecteurs propres orthonormés de L_{sym}.
4. Former la matrice $V \in \mathbb{R}^{n \times K}$ dont les colonnes sont les K vecteurs propres associés aux K plus petites valeurs propres de L_{sym}.
5. Calculer la matrice U en normalisant chaque ligne de la matrice V, c'est-à-dire en posant $u_{ij} = \frac{v_{ij}}{\sqrt{\sum_{\ell=1}^{K} v_{i\ell}^2}}$
6. Pour $i = 1, \ldots, n$ construire le vecteur $\mathbf{y}_i \in \mathbb{R}^K$ obtenu par transposition de la i^{e} ligne de U.
7. Partitionner l'ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ en K classes $C_1, \ldots, C_K$ par l'algorithme des k-means.
8. En déduire la partition $(P_1, \ldots, P_k)$ de Ω telle que $P_k = \{\mathbf{x}_i \mid \mathbf{y}_i \in C_k\}$.

Remarque sur les algorithmes de classification spectrale

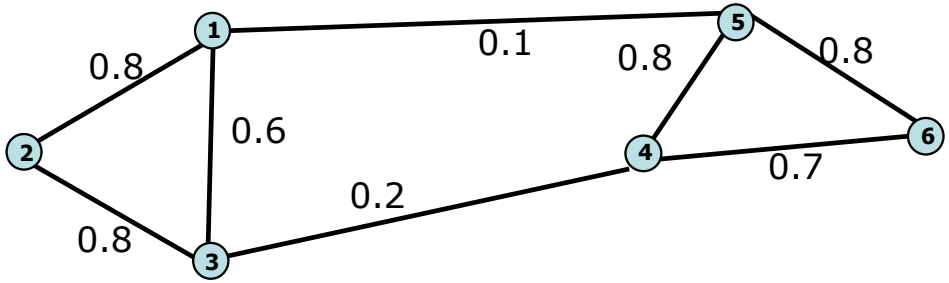
- La méthode utilisée par chaque algorithme consiste à représenter les données initiales $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ dans l'espace spectral (celui des vecteurs propres) qui met davantage en évidence les classes de sorte qu'une simple utilisation de l'algorithme des k-means suffise pour classer les observations dans l'espace propre.
- Les trois algorithmes diffèrent par les graphes Laplaciens utilisés.
- L'algorithme de Ng, Weiss et Jordan nécessite une normalisation supplémentaire des lignes de la matrice U.
- Les algorithmes de classification spectrale permettent de minimiser des critères de coupure de graphe.

Nombre de classes

Approche suggérée

Rechercher le saut le plus significatif dans les valeurs propres ; en supposant que les valeurs propres de la matrice Laplacienne sont ordonnées par ordre croissant, cela revient à rechercher K tel que $\lambda_1, \ldots, \lambda_K$ sont relativement petites par rapport à λ_{K+1}.

Exemple illustratif



Matrice de similarité

$$W = \begin{pmatrix} 1 & 0.8 & 0.6 & 0 & 0.1 & 0 \\ 0.8 & 1 & 0.8 & 0 & 0 & 0 \\ 0.6 & 0.8 & 1 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 1 & 0.8 & 0.7 \\ 0.1 & 0 & 0 & 0.8 & 1 & 0.8 \\ 0 & 0 & 0 & 0.7 & 0.8 & 1 \end{pmatrix}$$

Exemple illustratif

Matrice de similarités : W

$$\begin{pmatrix} 1 & 0.8 & 0.6 & 0 & 0.1 & 0 \\ 0.8 & 1 & 0.8 & 0 & 0 & 0 \\ 0.6 & 0.8 & 1 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 1 & 0.8 & 0.7 \\ 0.1 & 0 & 0 & 0.8 & 1 & 0.8 \\ 0 & 0 & 0 & 0.7 & 0.8 & 1 \end{pmatrix}$$

Matrice des degrés : D

$$\begin{pmatrix} 2.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.5 \end{pmatrix}$$

Matrice Laplacienne : $L = D - W$

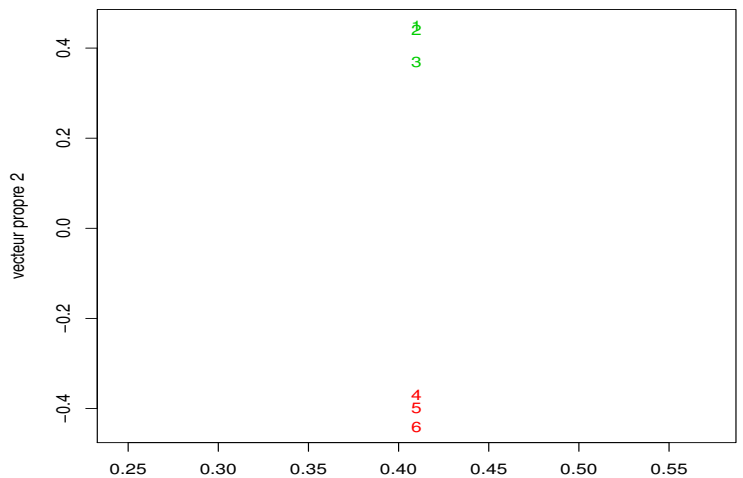
$$\begin{pmatrix} 1.5 & -0.8 & -0.6 & 0 & -0.1 & 0 \\ -0.8 & 1.6 & -0.8 & 0 & 0 & 0 \\ -0.6 & -0.8 & 1.6 & -0.2 & 0 & 0 \\ 0 & 0 & -0.2 & 1.7 & -0.8 & -0.7 \\ -0.1 & 0 & 0 & -0.8 & 1.7 & -0.8 \\ 0 & 0 & 0 & -0.7 & -0.8 & 1.5 \end{pmatrix}$$

Exemple illustratif

Valeurs propres					
0.0	0.2	2.1	2.3	2.5	2.6

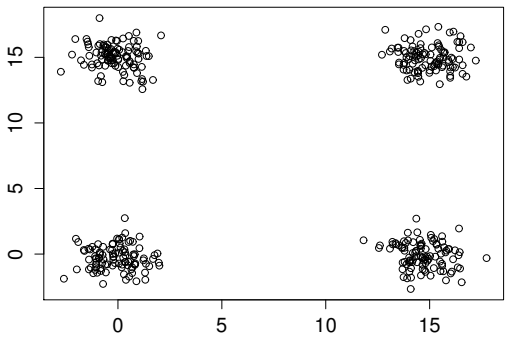
Vecteurs propres					
0.41	0.45	0.64	-0.30	0.37	-0.10
0.41	0.44	-0.01	0.30	-0.70	-0.21
0.41	0.37	-0.63	0.04	0.38	0.36
0.41	-0.37	-0.33	-0.45	0.00	-0.61
0.41	-0.40	0.16	-0.30	-0.35	0.65
0.41	-0.44	0.17	0.71	0.28	-0.08

kmeans dans l'espace propre

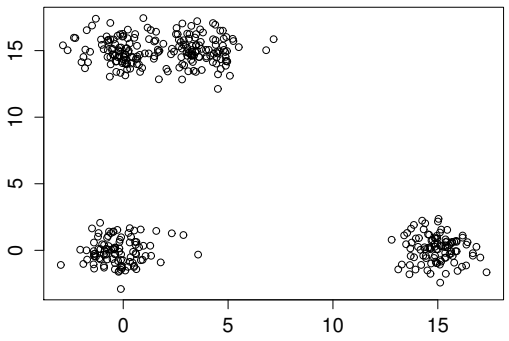


Autres illustrations - Choix du nombre de classes

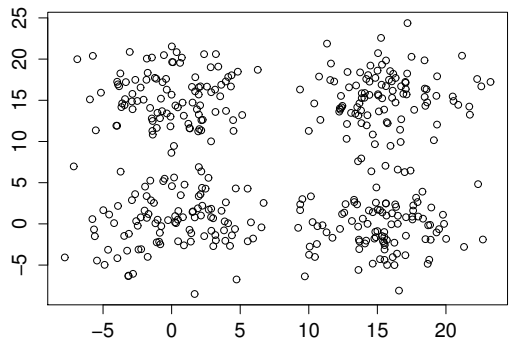
Données



Données

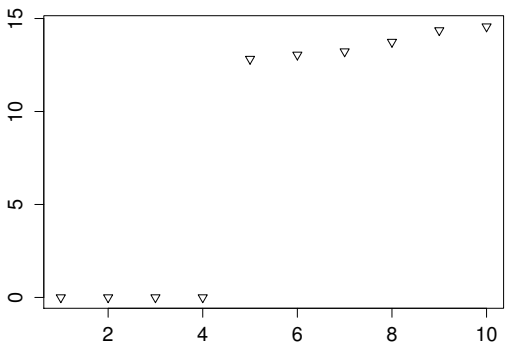


Données

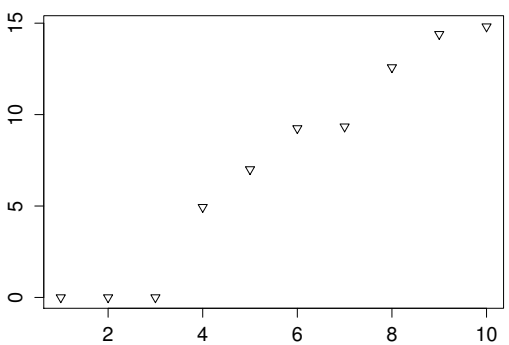


Utilisation de la similarité gausienne avec $\sigma = 2$

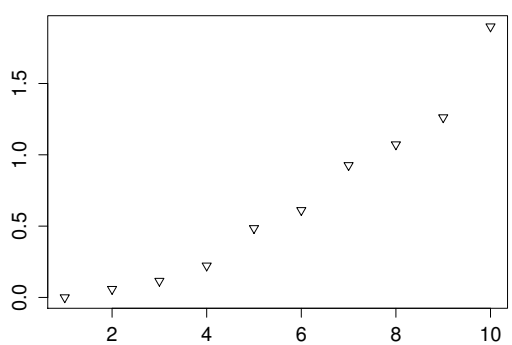
Valeurs propres de L



Valeurs propres de L



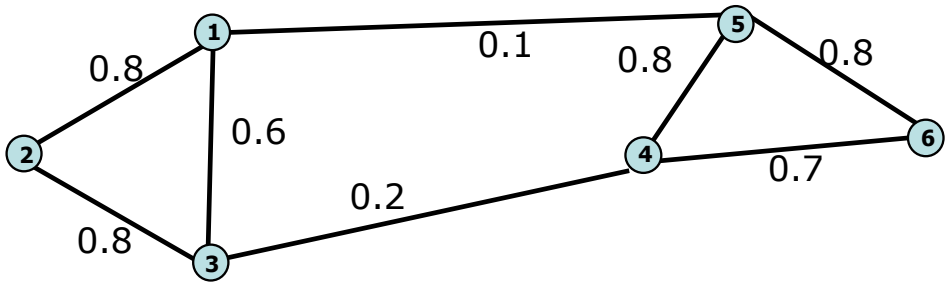
Valeurs propres de L



Critères de coupure de graphe

Idée générale

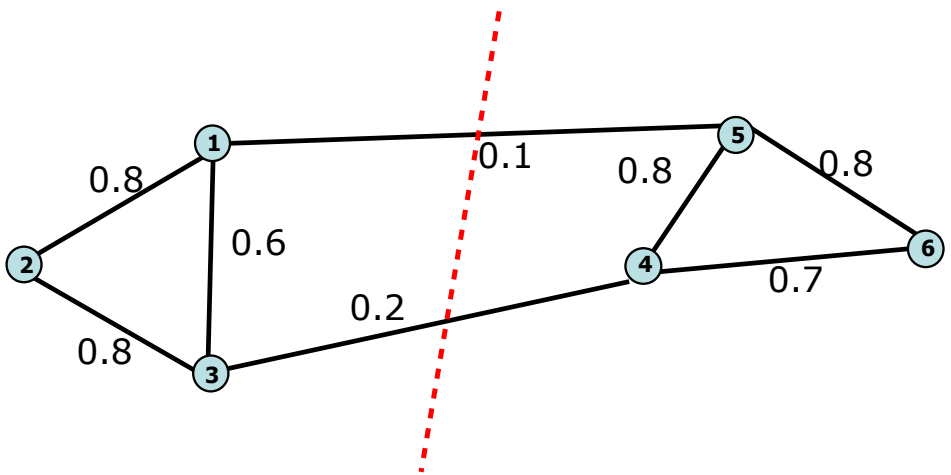
Rechercher une partition telle que les arêtes entre différentes classes aient les poids les plus faibles et que les arêtes au sein d'une même classe aient les poids les plus élevés.



Critères de coupure de graphe

Idée générale

Rechercher une partition telle que les arêtes entre différentes classes aient les poids les plus faibles et que les arêtes au sein d'une même classe aient les poids les plus élevés.



Critères de coupure d'un graphe

Critère de coupure en deux ensembles disjoints

$$\text{cut}(A, B) = \sum_{\mathbf{x}_i \in A} \sum_{\mathbf{x}_j \in B} w_{ij}$$

Critère de coupure associé à une partition en K classes

Recherche de la partition P qui minimise

$$\text{cut}(P) = \sum_{\ell=1}^{K} \text{cut}(P_\ell, \overline{P_\ell}) \quad \text{avec} \quad \overline{P_\ell} = \Omega \backslash P_\ell$$

Remarque

- Ce critère de coupure peut conduire à des partitions déséquilibrées (quelques classes formées de singletons)
- Comment imposer aux classes des tailles minimales ?

Critères de coupure normalisés

Ratiocut

$$\mathrm{RatioCut}(P) = \sum_{\ell=1}^{K} \frac{\mathrm{cut}(P_\ell, \overline{P_\ell})}{|P_\ell|}$$

avec $|P_\ell|$ = nombre d'éléments de la classe P_ℓ

Ncut

$$\mathrm{Ncut}(P) = \sum_{\ell=1}^{K} \frac{\mathrm{cut}(P_\ell, \overline{P_\ell})}{\mathrm{Vol}(P_\ell)}$$

avec $\mathrm{Vol}(P_\ell) = \sum_{\mathbf{x}_i \in P_\ell} d_i$

La minimisation de ces deux critères permet d'obtenir des partitions équilibrées mais elle conduit à des problèmes NP difficiles

Minimisation de RatioCut(P)

Notons $H = (h_{ik})$ la matrice $n \times K$ définie par :

$$h_{ik} = \begin{cases} \frac{1}{\sqrt{|P_k|}} & \text{si } \mathbf{x}_i \in P_k \\ 0 & \text{sinon} \end{cases}$$

On remarque que :

$$\begin{aligned} H'H &= I \\ \text{RatioCut}(P) &= \text{Trace}(H'\,L\,H) \end{aligned}$$

Finalement :

$$\min_P \text{RatioCut}(P) \iff \min_P \text{Trace}(H'\,L\,H) \text{ sous } H'H = I$$

Minimisation de RatioCut(P)

- En approximant l'hypothèse d'appartenance (discrète) aux classes par une appartenance "continue" aux classes, on peut montrer que le problème

$$\min_{P} \mathrm{Trace}(H'\,L\,H) \text{ sous } H'H = I$$

est équivalent au problème (relaxé)

$$\min_{H \in \mathbb{R}^{n \times K}} trace(H'\,L\,H) \text{ sous la contrainte } H'\,H = I$$

dont la matrice-solution est constituée (en colones) des vecteurs propres associés aux K plus petites valeurs propres de L.

- Le partitionnement final des données est obtenu en lançant l'algorithme des k-means sur les lignes de la matrice H.
- L'algorithme de classification spectrale (non normalisé) revient à minimiser $\mathrm{RatioCut}(P)$.

Minimisation de NCut(P)

Notons $H = (h_{ik})$ la matrice $n \times K$ définie par :

$$h_{ik} = \begin{cases} \frac{1}{\sqrt{\mathrm{Vol}(P_k)}} & \text{si } \mathbf{x}_i \in P_k \\ 0 & \text{sinon} \end{cases}$$

On remarque que :

$$\begin{aligned} H'DH &= I \\ \mathrm{Ncut}(P) &= \mathrm{Trace}(H'\,L\,H) \end{aligned}$$

Finalement :

$$\min_P \mathrm{Ncut}(P) \iff \min_P \mathrm{Trace}(H'\,L\,H) \text{ sous } H'DH = I$$

Minimisation de NCut(P)

- En posant $V = D^{1/2}\,H$, le problème (relaxé) s'écrit

$$\min_{V \in \mathbb{R}^{n \times K}} \mathrm{Trace}(V'\,D^{-1/2}\,L\,D^{-1/2}\,V) \text{ sous } V'\,V = I$$

  dont la matrice-solution V est constituée (en colones) des vecteurs propres associés aux K plus petites valeurs propres de L_{sym}.
- La matrice H est obtenue par resubstitution : $H = D^{-1/2}V$.
- D'après les propriétés des matrices Laplaciennes normalisées, la matrice H est formée des vecteurs propres de L_{rw}.
- Le partitionnement final des données est obtenu en lançant l'algorithme des k-means sur les lignes de la matrice H
- L'algorithme de classification spectrale de Shi et Malik (L_{rw}) conduit donc à la minimisation de $\mathrm{Ncut}(P)$.

Classification spectrale dans R

```
library(kernlab)
data(spirals)
sc <- specc(spirals,kernel = "rbfdot",centers=2)
plot(spirals, col=sc)
```