Apprentissage non supervisé : algorithme EM

Cours 9

Nicolas Baskiotis
nicolas.baskiotis@lip6.fr

Master 1 DAC

équipe MLIA, Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (UPMC)

S2 (2014-2015)

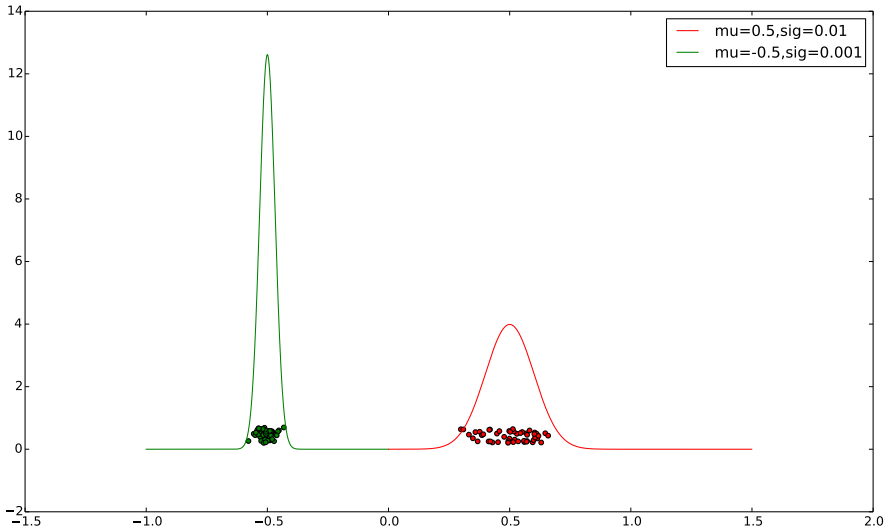
Plan

Rappel : k-means

- $x_1, \cdots, x_n \in X$
- Optimisation de : $F(\mu, C) = \sum_j \|\mu_{C(x_j)} - x_j\|^2$
- μ : centroïdes des clusters ($\mu_i = \sum_{j:C(x_j)=i} \|\mu - x_j\|^2$)
- C : affectation des exemples ($C(x_j) = argmin_i \|\mu_i - x_j\|^2$)
- Algorithme :
  - Première étape : on fixe μ, on optimise $C \Rightarrow$ espérance
  - Seconde étape : on fixe C, on optimise $\mu \Rightarrow$ maximum de vraissemblance

Rappel : distribution gaussienne

- En 1d : $p(x) = \mathcal{N}(x|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{(-\frac{1}{2\sigma^2}(x-\mu)^2)}$



  Remarque : à quoi sert la constante : $\frac{1}{(2\pi\sigma^2)^{1/2}}$?

- Multivariée en d dimensions:

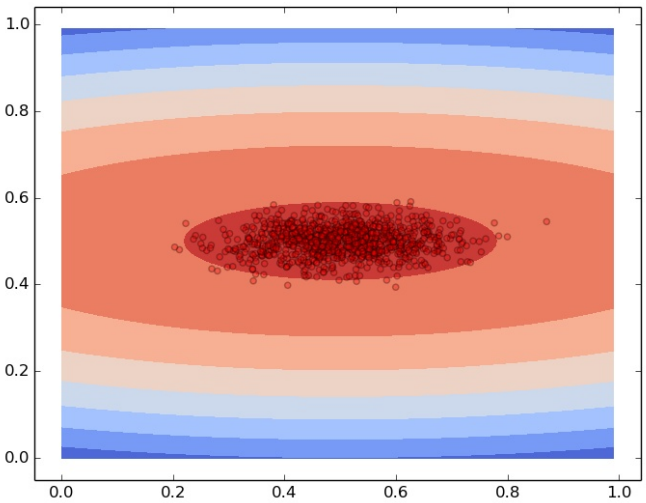
$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu))$$

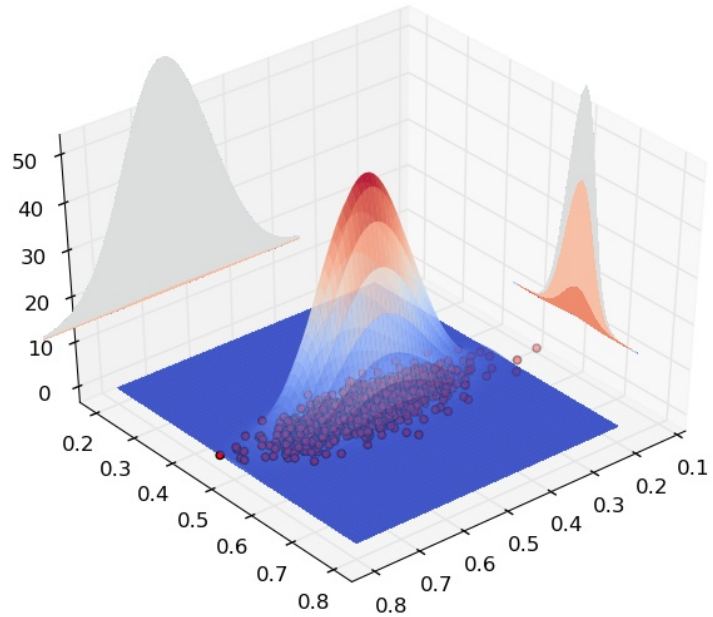
- $\mu = (\mu_1, \mu_2, \cdots, \mu_d)$, mais Σ ?

Gaussienne 2D : cas simple

- En 2d : on suppose que $x_1 \sim \mathcal{N}(\mu_1, \sigma_1)$ et $x_2 \sim \mathcal{N}(\mu_2, \sigma_2)$
- hypothèse Naive Bayes, x_1 indépendant de x_2
- $p(x) = p(x|\mathcal{N}_1)p(x|\mathcal{N}_2) = \frac{1}{(2\pi\sigma_1^2)^{1/2}} e^{-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2} \frac{1}{(2\pi\sigma_2^2)^{1/2}} e^{-\frac{1}{2\sigma_2^2}(x_2-\mu_2)^2}$
- $p(x) = \frac{1}{(2\pi)^{2/2}(\sigma_1^2\sigma_2^2)^{1/2}} e^{-\frac{1}{2}(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2})} = \frac{1}{2\pi\Sigma^{1/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}$

  avec $\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$





Gaussienne 2D : cas générique

Transformation affine

- Supposons $x_1, x_2 \sim \mathcal{N}(0,1)$ et $X = (x_1, x_2)$;
- Soit T une transformation affine inversible $T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}$
- Soit $Y = TX + \mu$, $y_1 = t_{11}x_1 + t_{12}x_2 + \mu_1$, $y_2 = t_{21}x_1 + t_{22}x_2 + \mu_2$
- alors $\mathbb{E}(Y) = \begin{pmatrix} \mathbb{E}(t_{11}x_1 + t_{12}x_2 + \mu_1) \\ \mathbb{E}(t_{21}x_1 + t_{22}x_2 + \mu_2) \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$
- Variance d'un vecteur aléatoire ?

Covariance

- Covariance de deux variables aléatoires :
  $Cov(x,y) = \mathbb{E}((x - \mathbb{E}(x))(y - \mathbb{E}(y))) = \mathbb{E}(xy) - \mathbb{E}(x)\mathbb{E}(y)$
- Matrice de covariance d'un vecteur aléatoire X, $Cov(X)$:

$$\begin{pmatrix} Cov(x_1,x_1) & \cdots & Cov(x_1,x_n) \\ \vdots & \ddots & \vdots \\ Cov(x_n,x_1) & \cdots & Cov(x_n,x_n) \end{pmatrix} = \mathbb{E}((X-\mu)(X-\mu)') = \mathbb{E}(XX') - \mu\mu'$$

Gaussienne 2D : cas générique

Covariance de $Y = TX + \mu$: $Cov(Y) = TT'$

- $Cov(Y) = \begin{pmatrix} \mathbb{E}((t_{11}x_1 + t_{12}x_2)^2) & \mathbb{E}((t_{11}x_1 + t_{12}x_2)(t_{21}x_1 + t_{22}x_2)) \\ \mathbb{E}((t_{11}x_1 + t_{12}x_2)(t_{21}x_1 + t_{22}x_2)) & \mathbb{E}((t_{21}x_1 + t_{22}x_2)^2) \end{pmatrix}$

$= \begin{pmatrix} t_{11}^2 + t_{12}^2 & t_{11}t_{21} + t_{12}t_{22} \\ t_{11}t_{21} + t_{12}t_{22} & t_{21}^2 + t_{22}^2 \end{pmatrix}$

On note $\Sigma = Cov(Y)$

Changement de variable

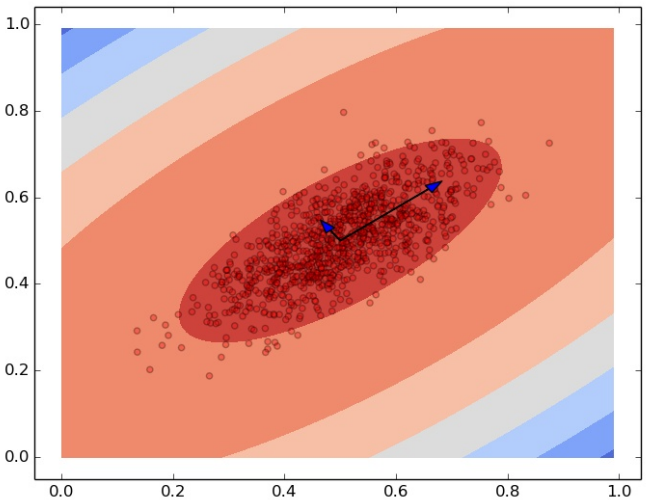
- $p(x) = \frac{1}{2\pi\Sigma_{\mathcal{N}(0,1)}^{1/2}} e^{-\frac{1}{2}(x-\mu_{\mathcal{N}(0,1)})'\Sigma_{\mathcal{N}(0,1)}^{-1}(x-\mu_{\mathcal{N}(0,1)})}$, avec $\mu_{\mathcal{N}(0,1)} = 0, \Sigma_{\mathcal{N}(0,1)} = I$
- Si $Y = TX + \mu$, alors $p(Y) = \frac{1}{|det(T)|} p(T^{-1}(Y - \mu))$
- $p(Y) = \frac{1}{2\pi|\Sigma|^{-1/2}} e^{-\frac{1}{2}((T^{-1}(Y-\mu))'IT^{-1}(Y-\mu))} = \frac{1}{|\Sigma|^{-1/2}2\pi} e^{-\frac{1}{2}(Y-\mu)'T'^{-1}T^{-1}(Y-\mu)}$
- $p(Y) = \frac{1}{2\pi|\Sigma|^{-1/2}} e^{\frac{1}{2}(Y-\mu)'\Sigma^{-1}(Y-\mu)}$

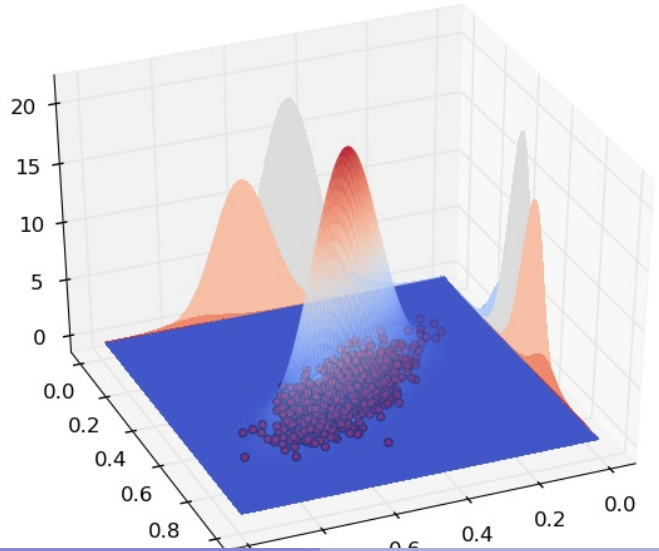
Gaussienne 2D : interprétation géométrique

Transformation affine inversible

- T peut être décomposé en $T = UD$, D diagonale (valeurs propres) et U orthogonale (vecteurs propres, matrice de rotation et reflexion)
- $\Sigma = UD(UD)' = UDD'U' = UD^2U'$
- $Det(\Sigma) = Det(UD^2U') = Det(D^2) = \sum_i \sigma_i^2$, σ_i valeurs propres de T

$\Rightarrow$ Loi normale multivariée : D représente la variance sur chaque composante normale indépendante des autres,
U représente la rotation/reflexion par rapport aux axes.



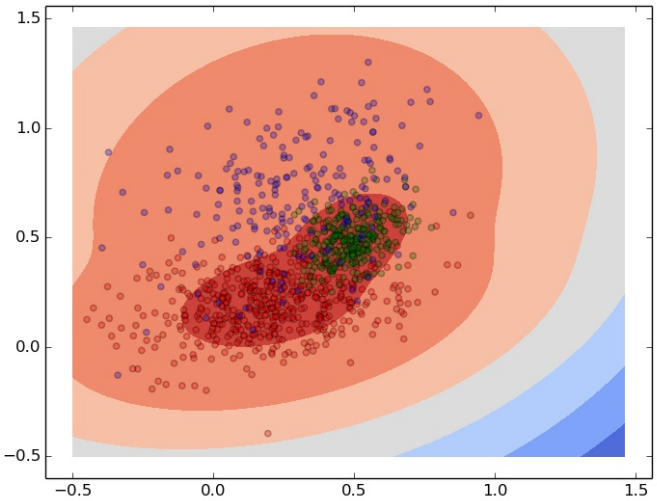


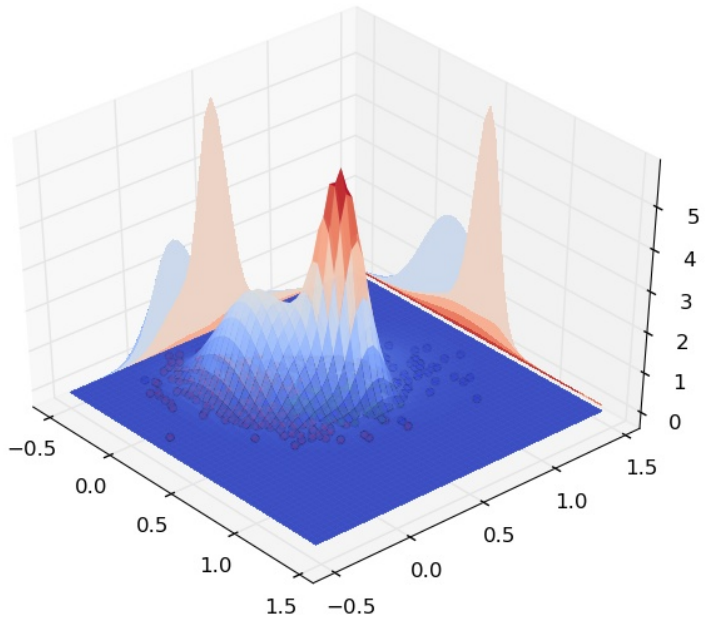
Plan

Mixture de gaussienne

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \quad \sum_k \pi_k = 1$$

Les π_k : coefficients de mixture





Algorithme EM

Principe

- Généralisation de l'algorithme k-means
- Hypothèse :
  - les exemples sont issus d'une mixture $\mathcal{N}(x|\mu,\Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k)$
  - chaque exemple est issu d'une des distribution $\mathcal{N}_i$

  $\Rightarrow$ un cluster : une des distribution $\mathcal{N}_i$
- On ne connait pas :
  - les paramètres (μ_i,Σ_i) de chaque $\mathcal{N}_i$
  - le nombre de composantes K de la mixture et leur coefficient de mélange π_k
  - de quel élément de la mixture chaque exemple est tiré (le cluster)

$\Rightarrow$ Objectifs : estimer l'ensemble de ces paramètres

- Interprétation : π_j probabilité de chaque composante

Formalisation

Variable latente

- Un exemple n'est issu que d'une seule distribution i

$\Rightarrow$ Introduction d'une variable aléatoire $\mathbf{z} = (z_1, \cdots, z_K),\ z_j \in \{0, 1\},\ \sum_j z_j = 1$

- Cette variable *latente* indique à quelle distribution l'exemple appartient : $z_i = 1,\ z_{j \neq i} = 0$
- $p(z_j = 1) = \pi_j$, comme π_j probabilité de chaque composante
- $p(\mathbf{z}) = \prod_{j=1}^{K} \pi_j^{z_j}$

Probabilité conditionnelle

- $p(x|z_j = 1) = \mathcal{N}(x|\mu, \Sigma) = \mathcal{N}(x|\mu_j, \Sigma_j)$
- $p(x|\mathbf{z}) = \prod_{j=1}^{K} \mathcal{N}(x|\mu_j, \Sigma_j)^{z_j}$

Formalisation

Distribution marginale

- $p(x) = \sum_{\mathbf{z}} p(x, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(x|\mathbf{z}) = \sum_{\mathbf{z}} \prod_{j=1}^{K} \pi_k^{z_j} \mathcal{N}(x|\mu_j, \Sigma_j)^{z_j}$

$= \sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)$

- Avec $\mathbf{z}$ K-binaire, la distribution de x est bien une mixture de gaussienne.
- Pour chaque observation x^i, nous avons besoin d'une variable latente z^i.

contribution de chaque composante

- Soit $\gamma(z_i) = p(z_i = 1|x)$
- $\gamma(z_i) = \frac{p(z_i=1)p(x|z_i=1)}{\sum_{j=1}^{K} p(z_j=1)p(x|z_j=1)} = \frac{\pi_i \mathcal{N}(x|\mu_i,\Sigma_i)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j,\Sigma_j)}$
- $\pi_j \Rightarrow$ probabilité *a priori* de $z_j = 1$
- γ_j probabilité *a posteriori*, contribution de la composante à l'explication de x.

Estimation par maximum de vraissemblance

Dans le cas supervisé

Pour chaque cluster j, soit les x^i de ce cluster ($z_j^i = 1$) :

- $\mu_j = \frac{1}{|x^i / z_j^i = 1|} \sum_{x^i / z_j^i = 1} x^i$
- $\Sigma_j = \frac{1}{|x^i / z_j^i = 1|} \sum_{x^i / z_j^i = 1} (x^i - \mu_j)(x^i - \mu_j)'$

Dans le cas non supervisé

- On cherche $\theta = argmax_\theta \prod_i p(x^i; \theta)$
- avec θ codant pour tous les paramètres : (π_j, μ_j et Σ_j) et sans connaître les z^i ...
- On peut marginaliser : $\theta = argmax_\theta \prod_i \sum_{j=1}^{K} p(x_i | z^i = j; \theta) p(z^i = j; \theta) = argmax_\theta \sum_i log \sum_{j=1}^{k} p(x^i | z^i = j; \theta) p(z^i = j; \theta)$
- Impossible à optimiser directement

EM pour une mixture de gaussiennes

Expectation step

Pour tout i, j, on fixe :

$w_j^i = p(z^i = j | x^i, \mu, \Sigma, \pi) = \frac{p(x^i | z^i = j; \mu, \Sigma) p(z^i = j; \pi)}{\sum_k p(x^i | z^i = k, \mu, \Sigma) p(z^i = k; \pi)}$, avec $p(z^i = k; \pi) = \pi_k$.

Maximization step

On met à jour les paramètres :

- $\pi_j = \frac{1}{n} \sum_{i=1}^m w_j^i$
- $\mu_j = \frac{\sum_{i=1}^m w_j^i x^i}{\sum_{i=1}^m w_j^i}$
- $\Sigma_j = \frac{\sum_{i=1}^m w_j^i (x^i - \mu_j)(x^i - \mu_j)'}{\sum_{i=1}^m w_j^i}$

Contrairement à k-means, ici on a un soft-clustering.

EM généralisé

Contexte

- Donnée $\{x^1, \cdots, x^n\}$
- Paramétrisation θ d'un modèle : $\ell(\theta) = \sum_{i=1}^{m} logp(x;\theta)$
- Introduction de variables latentes z, généralement non observées : $\ell(\theta) = \sum_{i=1}^{m} log \sum_z p(x,z;\theta)$
- Si z^i connues, alors maximum de vraissemblance "facile" à calculer.
- Sinon, $\Rightarrow$ EM.

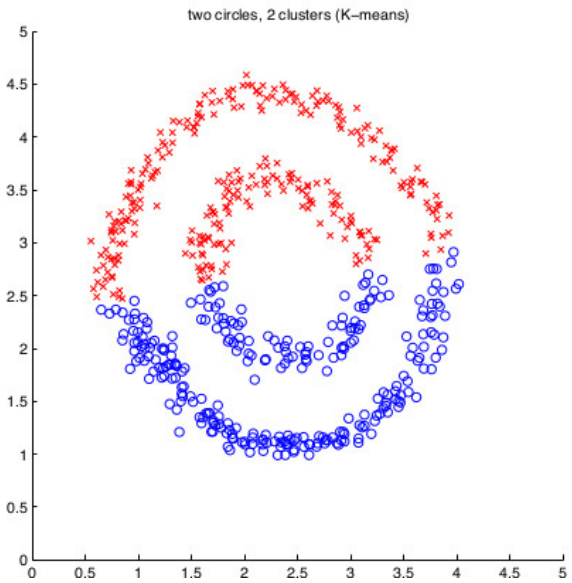
Algorithme

- Soit Q_i une distribution sur z : $\sum_z Q_i(z) = 1$.
- Expectation step : $Q_i(z^i) = p(z^i|x^i;\theta)$
- Maximization step : $\theta = argmax_\theta \sum_i \sum_z^i Q_i(z^i) \frac{logp(x^i,z^i;\theta)}{Q_i(z^i)}$

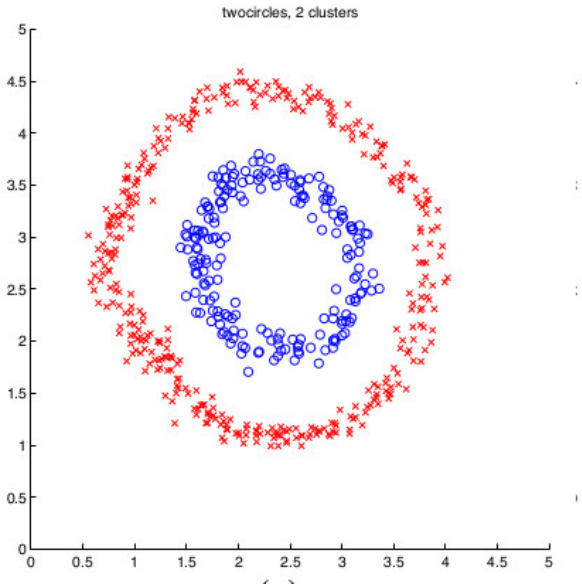
Plan

Problèmatique

K-means



Spectral clustering



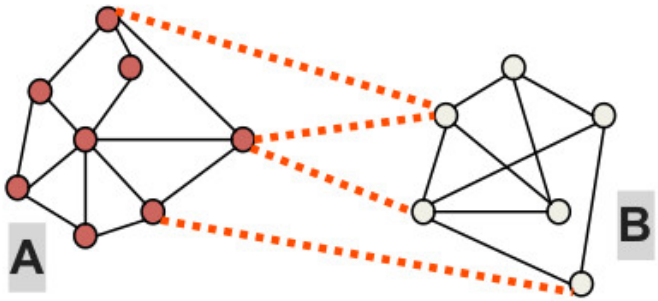
Limites des approches vues

- 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

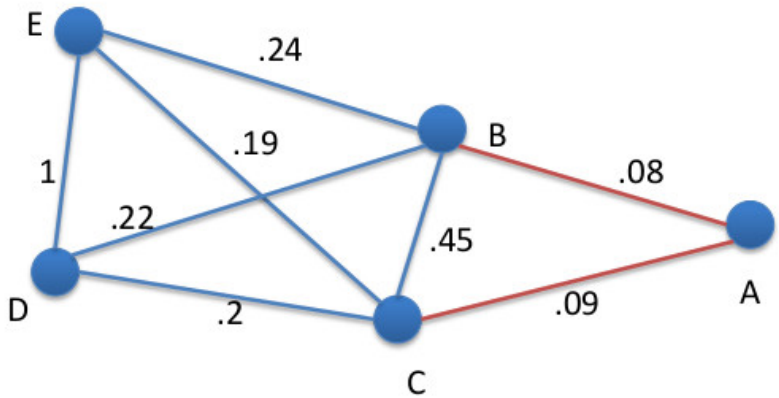


Création du graphe

- 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 cluter très similaires
  - données de différent cluster dissimilaire
- 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. . .

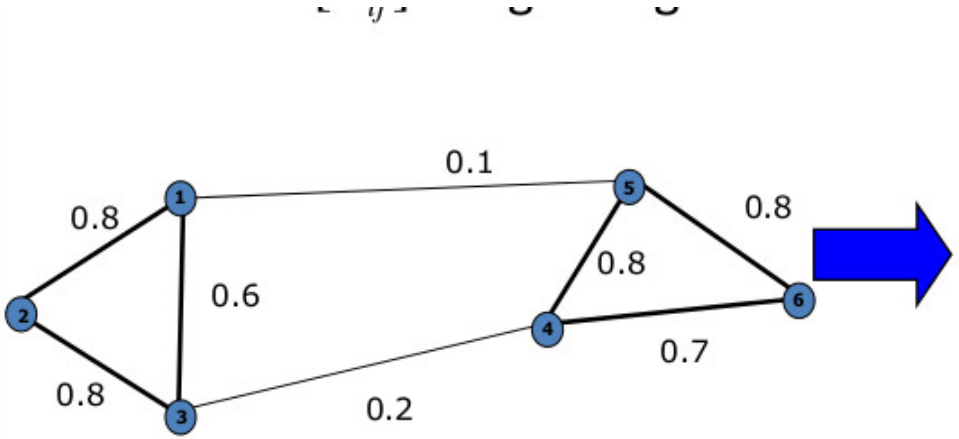


$Cut(BCDE, A) = 0.17$

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

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 considére : matrice laplacienne : $L = D - W$
- Propriétés :
  - Valeurs propres positives
  - Vecteurs propres orthogonaux
  - Ce sont des indicateurs de la connectivité du graphe
- Interprétation : u vecteur binaire de taille n,
  - Lu : poids des connections entrantes des noeuds exprimés par f.
  - $u'L$: poids des conenctions entrantes des noeuds exprimés par f
- 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 $i \in C_2$.
  - $f'Lf = \sum_{i,j} w_{i,j}(f_i - f_j)^2$:
    $f'Lf = f'(D - W)f = f'Df - f'Wf = \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij}$
    $= \frac{1}{2}(\sum_i(\sum_j w_{ij})f_i^2 - 2\sum_{i,j} f_i f_j w_{ij} + \sum_j(\sum_i w_{ij})f_j^2) = \frac{1}{2}\sum_{i,j} w_{i,j}(f_i - f_j)^2$
- Objectif : trouver f qui minimise $\frac{f'Lf}{f'Df}$ tel que $f'D\mathbf{1} = 0$
- Relaxation : $min_f f'Lf$ tel que $f'Df = 1 \Rightarrow Lf = \lambda Df$: deuxième vecteur propre.