Partionnement de données I
Introduction

Objectif : Regrouper les individus qui se ressemblent au sein d'un même groupe et repartir les individus différents dans des groupes séparés.
Classification non supervisé : Les individus ne sont pas étiquetés, ils consistent seulement en un ensemble d'attributs.
Usages :

- Quantification Vectorielle. Compresser des données en apprenant un dictionnaire (ex : GIF, GSM). Recouvrir un espace géographique (ex : antennes téléphoniques). Réduire les dimensions.
- Segmentation. Dans une image, extraire des zones pour décrire les objets qui la composent et le fond. Dans un dialogue, regrouper les passages où c'est la même personne qui parle.

Partionnement de données II
Introduction

- Extraction de communautés (ou de catégories). Pour le business et le marketing, extraire les produits ou les clients semblables (ex : recommandations de films, de news, etc). Dans un corpus de documents, retrouver les sujets.

Différent types de partitionnement :

- Dur. Les individus appartiennent chacun à un seul groupe.
- Flou. Les individus appartiennent à plusieurs groupes. Un poids, parfois interprété comme une probabilité, donne le degré d'appartenance d'un individu à un groupe.
- Hiérarchique. Découpage en groupes et sous-groupes.

Trois types de méthodes :

- Géométrique. Distance de similarités entre individus. Minimisation de l'erreur empirique de quantification.
- Spectral. Graphe d'affinité entre individus. Recherche de

Partionnement de données III
Introduction

- Probabiliste. Chaque groupe est représenté par une distribution. Les individus sont des échantillons d'un mélange de distributions. On cherche les paramètres des distributions, du mélange et les degrés d'appartenance qui maximisent la vraisemblance.

Dans ce qui suit,

- $\mathbf{x}_i$ est le vecteur colonne décrivant l'individu i.
- $S_N = \{\mathbf{x}_i | i = 1..N\}$ est l'ensemble d'apprentissage.
- Le nombre de partition est supposé connu égal à K.
- Partitionnement dur, C_k est la k-ième partie (ensemble).
- Partitionnement flou, π_{ik} est le coefficient d'appartenance de l'individu i à la k-ième partie.
- Les centres des parties sont notés $\mathbf{c}_k$.

1. Introduction

2. Méthodes géométriques
   - K-moyennes
   - K-moyennes flou

3. Méthodes spectrales
   - Graphes : définitions
   - Partitionner le graphe
   - Des données au graphe
   - Algorithme

Algorithme des K-moyennes

Critère : erreur de quantification

Objectif : Trouver une partition $C = \{C_k | k = 1..K\}$ (tel que $C_i \cap C_j = \emptyset$ pour tout i, j et $\cup_{k=1}^{K} C_k = S$) et les centres des parties $\{\mathbf{c}_k\}_{k=1..K}$ qui minimisent l'erreur de quantification.

Définition (Erreur de quantification)

$$EQ(C, \{\mathbf{c}_k\}_{k=1..K}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \mathbf{c}_k\|_2^2$$

Algorithme :

1. Initialise les centres $\mathbf{c}_k$ $(k = 1..K)$ aléatoirement dans S
2. Assigne chaque individu $\mathbf{x}_i$ à la partie k_i tel que,

$$k_i = \arg\min_k \|\mathbf{x}_i - \mathbf{c}_k\|_2$$

3. On recalcule les centres des parties, tel que

$$\mathbf{c}_k = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

4. On itère les étapes 2 et 3 jusqu'à convergence (invariance des C_k)

Dans l'étape 3, la moyenne $\mathbf{c}_k$ correspond au barycentre des points de C_k. Ce n'est vrai que pour la distance euclidienne.

Algorithme des K-moyennes
Propriétés

- Trouver C qui minimise EQ est en général NP-dur.
- A chaque itération des k-moyennes, $EQ(C)$ décroit.
- Comme $EQ(C)$ est borné par en dessous, l'algorithme converge vers un minimum local.
- On peut aussi partir de partitions aléatoires.
- A cause de la distance euclidienne, les parties ne peuvent être retrouvées que si les elles sont linéairement séparables.

Algorithme des K-moyennes
Variantes

Il existe de nombreuses variantes :

- Pour la norme L_1, on a l'algorithme des K-médianes
  - $\Rightarrow$ On remplace la norme L_2 par L_1 et la moyenne par la médiane à l'étape 3.
- On peut remplacer la norme L_2 par n'importe quelle distance $d(\cdot, \cdot)$ pour obtenir l'algorithme des K-médoïdes
  - $\Rightarrow$ On remplace $\| \cdot - \cdot \|_2$ par $d(\cdot, \cdot)$. A l'étape 3 cherche parmi S les meilleurs centres (dans chaque partie, celui qui minimise la somme des distances au centre).
- En remplaçant la distance L_2 par les cosinus des angles entre vecteurs, on obtient l'algorithme des K-moyennes sphériques
  - $\Rightarrow$ Les centres sont normalisés. Le résultat dépend que de la direction des vecteurs plutôt que de leur taille (ex : les étoiles dans le ciel)

Algorithme flou des K-moyennes
Objectif

Objectif : Trouver $\Pi = (\pi_{ik})_{ik}$ (tel que $\pi_{ik} \geq 0$ et $\sum_{k=1}^{K} \pi_{ik} = 1$) et les centres des parties $\{\mathbf{c}_k\}_{k=1..K}$ qui minimisent l'erreur de quantification flou. Les π_{ik} peuvent être interprétées comme des probabilités.

Définition (Erreur de quantification flou)

$$EQF(\Pi, \{\mathbf{c}_k\}_{k=1..K}) = \sum_{i=1}^{N} \sum_{k=1}^{K} (\pi_{ik})^m \|\mathbf{x}_i - \mathbf{c}_k\|_2^2,$$

où m est un paramètre qui représente le degré de mélange autorisé entre les parties.

De même que pour les K-moyennes, l'algorithme est une minimisation alternée du coût mais cette fois sous contraintes. On résout en introduisant le Lagrangien.

Algorithme flou des K-moyennes
Algorithme

Algorithme :

1. Initialise les centres c_k $(k = 1..K)$ aléatoirement dans S
2. Pour chaque $\mathbf{x}_i$ et chaque partie $\mathbf{c}_k$ on calcule le coefficient d'appartenance π_{ik},

$$\pi_{ik} = \frac{1}{\sum_{j=1}^{K} \left(\frac{\|\mathbf{x}_i - \mathbf{c}_k\|_2}{\|\mathbf{x}_i - \mathbf{c}_j\|_2} \right)^{\frac{2}{m-1}}}$$

3. On recalcule les centres des parties,

$$\mathbf{c}_k = \frac{\sum_{i=1}^{N} (\pi_{ik})^m \mathbf{x}_i}{\sum_{\mathbf{x} \in S} (\pi_{ik})^m}$$

4. On itère les étapes 2 et 3 jusqu'à convergence (changement faible dans les paramètres Π ou $\{\mathbf{c}_k\}_{k=1..K}$)

1. Introduction

2. Méthodes géométriques
   - K-moyennes
   - K-moyennes flou

3. Méthodes spectrales
   - Graphes : définitions
   - Partitionner le graphe
   - Des données au graphe
   - Algorithme

Méthodes spectrales
Introduction

- On ne cherche plus des parties compactes mais plutôt des parties connectées.
- Les connexions sont représentés par un graphe dont les sommets sont les individus.
- Certains données ont déjà une structure de graphe :
  - les pages internet, connectées par des liens ;
  - les structures des protéines ;
  - les graphes de citations.
- Pour les autres, on va voir comment construire un graphe.
- Au final, on va reconstruire les parties en utilisant les vecteurs propres d'une matrice dérivée des données.

Définition

Un graphe $G = (V, E)$ est défini par un ensemble V de sommets et d'un ensemble E d'arêtes. Si les arêtes peuvent être orientées ou non. On parle alors de graphe orienté ou non.

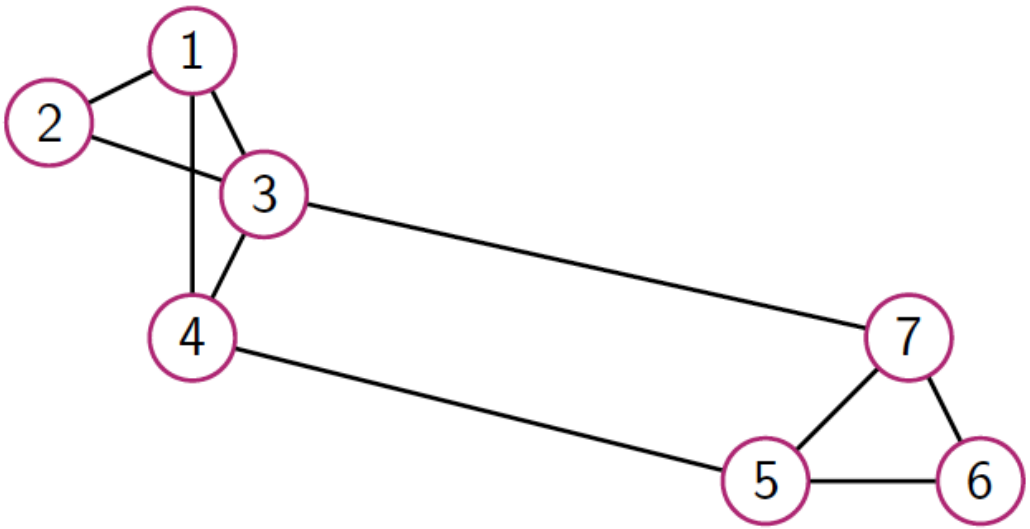
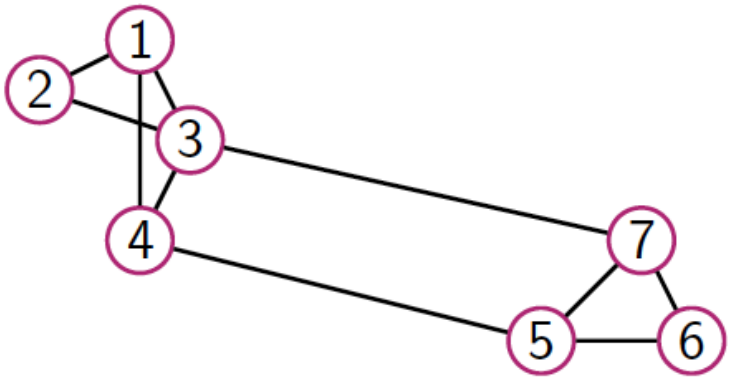


FIGURE: Un graphe non orienté

Graphes
Matrice d'adjacence

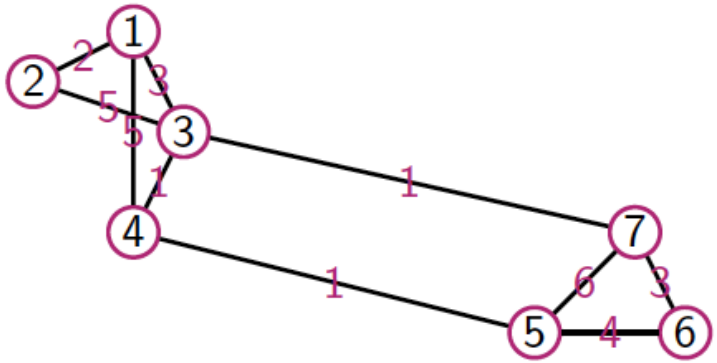
La matrice d'adjacence corresponds aux connexions entre les sommets du graphe,



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

Si i et j sont connectés alors le coefficient i, j vaut 1, 0 sinon. Si le graphe est non orienté la matrice d'adjacence est symétrique.

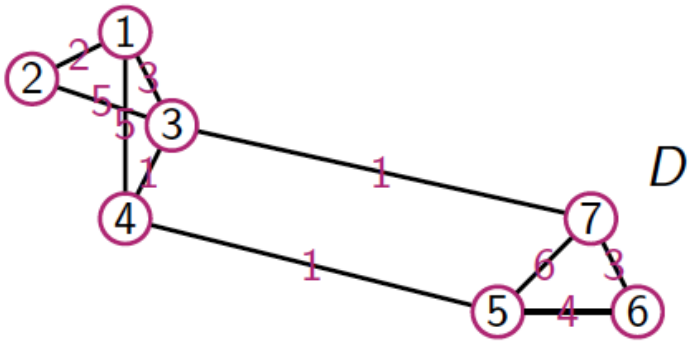
Maintenant, donnons des poids aux arêtes.



$$A = \begin{pmatrix} 0 & 2 & 3 & 5 & 0 & 0 & 0 \\ 2 & 0 & 5 & 0 & 0 & 0 & 0 \\ 3 & 5 & 0 & 1 & 0 & 0 & 1 \\ 5 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 4 & 6 \\ 0 & 0 & 0 & 0 & 4 & 0 & 3 \\ 0 & 0 & 1 & 0 & 6 & 3 & 0 \end{pmatrix}$$

La matrice d'affinité est similaire à la matrice d'adjacence mais contient le poids des arêtes.

La matrice de degré quantifie l'importance d'un sommet dans le graphe. Un sommet est important si il est fortement connecté aux autres.



$$D = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix}$$

La matrice de degré est diagonal. Ses coefficients diagonaux valent la somme des poids des arêtes connectées aux sommets.

Il existe de nombreuses définitions du Laplacien, en voici deux :

- Laplacien (non-normalisé) : $L = D - A$
- Laplacien normalisé : $L_{norm} = D^{-1}L = I - D^{-1}A$

Laplacien
Propriétés spectrales

Le Laplacien a d'intéressantes propriétés spectrales.

- Bien que L dépend de l'ordre des sommets, son spectre en est indépendant.
- Pour tout $\mathbf{z}$, on a $\mathbf{z}^\top L\mathbf{z} = \frac{1}{2}\sum_{i,j=1}^{N} a_{ij}(z_i - z_j)^2$.
- L est symétrique (par définition) et semi-définie positive (propriété précédente)
  - $\Rightarrow$ Ses valeurs propres sont réelles et positives.
- Sa plus petite valeur propre vaut 0 et est associée à $\mathbf{1}$ ($L\mathbf{1} = D\mathbf{1} - A\mathbf{1} = diag(D) - diag(D) = \mathbf{0}$)
- La multiplicité de la valeur propre 0 est égale au nombre de composantes connectées. Le sous espace propre associé est engendré par les vecteurs $\mathbf{1}_{C_i}$ $(i = 1..K)$.

Partitionner le graphe
Première tentative

Un bon partitionnement groupe les individus similaires ensembles et sépare les autres. Sur un graphe, on veut partitionner les sommets tel que

- les arêtes allant d'un sommet à un autre de la même partie aient des poids forts ;
- les arêtes dont les sommets appartiennent à des parties différentes aient des poids faibles.

Pour cela, on définit l'affinité entre deux parties

$$A(C_k, C_l) = \sum_{\mathbf{x}_i \in C_k, \mathbf{x}_j \in C_l} a_{ij}$$

On cherche alors le partitionnement qui minimise,

$$coupe(C_1, ..., C_K) = \frac{1}{2} \sum_{k=1}^{K} A(C_i, \overline{C}_i),$$

où $\overline{C}_i$ est le complémentaire de C_i

Partitionner le graphe
Première tentative

Un bon partitionnement groupe les individus similaires ensembles et sépare les autres. Sur un graphe, on veut partitionner les sommets tel que

- les arêtes allant d'un sommet à un autre de la même partie aient des poids forts ;
- les arêtes dont les sommets appartiennent à des parties différentes aient des poids faibles.

Pour cela, on définit l'affinité entre deux parties

$$A(C_k, C_l) = \sum_{\mathbf{x}_i \in C_k, \mathbf{x}_j \in C_l} a_{ij}$$

On cherche alors le partitionnement qui minimise,

$$coupe(C_1, ..., C_K) = \frac{1}{2} \sum_{k=1}^{K} A(C_i, \overline{C}_i),$$

où $\overline{C}_i$ est le complémentaire de C_i

Partitionner le graphe
Illustration

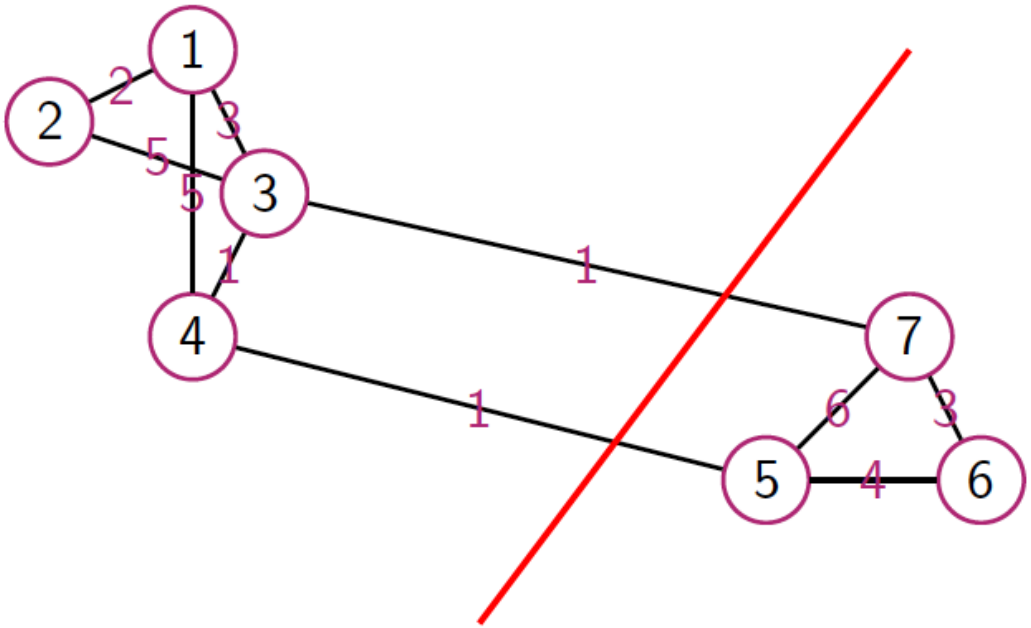


FIGURE: Coupe minimale

Dans cette exemple, la coupe vaut $\frac{1}{2}(2+2) = 2$.

Partitionner le graphe
Problème

La coupe minimum ne partitionne pas de façon satisfaisante.
Elle tend à séparer les sommets isolés du reste du graphe.

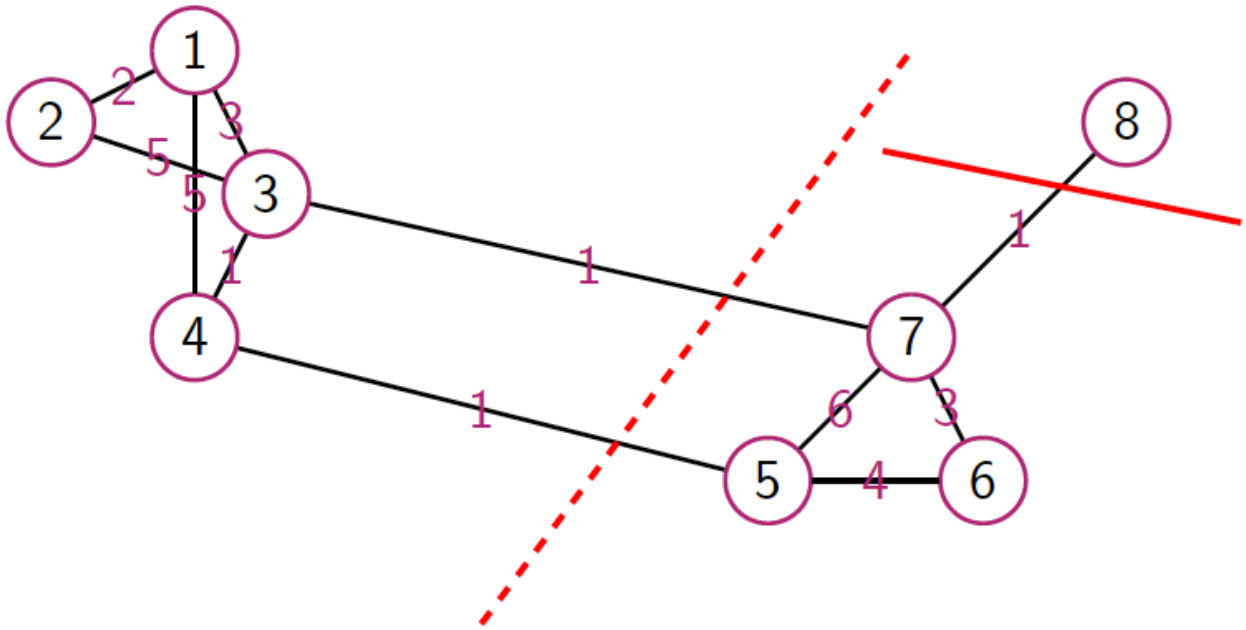


FIGURE: Coupe minimale

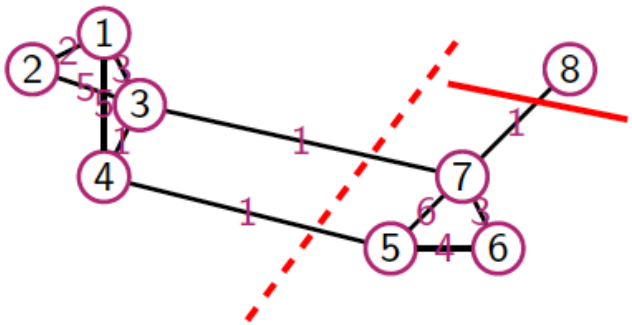
Dans cette exemple, la coupe vaut $\frac{1}{2}(1+1) = 1$. Ce qui est meilleur que la coupe tracée en pointillés.

Partitionner le graphe
Deuxième tentative

Le critère de coupe, ne prend pas en compte la taille des parties.
On propose de prendre le critère suivant,

$$RatioCut(C_1, ..., C_K) = \sum_{i=1}^{K} \frac{coupe(C_i, \overline{C}_i)}{|C_i|}$$

où $|C_i|$ est le nombre d'individus dans C_i.
Dans, l'exemple précédent :



- La coupe en pointillés donne

$$\frac{1}{2}\left(\frac{2}{4}+\frac{2}{4}\right)=\frac{1}{2}$$

- La coupe pleine donne

$$\frac{1}{2}\left(\frac{1}{1}+\frac{1}{7}\right)=\frac{4}{7}>\frac{1}{2}$$

Partitionner le graphe
Relaxation du critère

- Introduire le poids des partitions dans le critère rend la minimisation NP-dur.
- On va le résoudre de façon approchée.
- Démonstration pour $K = 2$, on veut trouver

$$\min_{C} RatioCut(C, \overline{C})$$

- Soit $\mathbf{z}$ tel que,

$$z_i = \begin{cases} \sqrt{|\overline{C}|/|C|}, & \text{if } \mathbf{x}_i \in C \\ -\sqrt{|C|/|\overline{C}|}, & \text{sinon} \end{cases}$$

on a,

$$\mathbf{z}^\top L\mathbf{z} = N \cdot RatioCut(C, \overline{C}) \quad \text{et} \quad \mathbf{z}^\top \mathbf{1} = \sum_{i=1}^{N} z_i = 0$$

Partitionner le graphe
Relaxation du critère (suite)

- Notre problème ce réécrit donc,

$$\min_{C} \mathbf{z}^\top L\mathbf{z} \text{ , tel que } \mathbf{z}^\top \mathbf{1} = 0 \text{ , } \mathbf{z}^\top \mathbf{z} = \sqrt{N} \text{ ,}$$

  avec $\mathbf{z}$ qui s'écrit comme précédemment.

- Si on relâche la contrainte sur les valeurs que peut prendre $\mathbf{z}$, alors la solution est le vecteur propre associée à la seconde plus petite valeur propre de L. (Par le Lagrangien, voir cour précédent)

Partitionner le graphe
Relaxation du critère (suite)

- Pour $K \geq 2$, la démonstration est similaire avec les vecteurs $\mathbf{z}_k$ $(k = 1..K)$

$$z_{ik} = \begin{cases} 1/\sqrt{|C_k|}, & \text{if } \mathbf{x}_i \in C \\ 0, & \text{sinon} \end{cases}$$

- On montre que

$$\mathbf{z}_k^\top L \mathbf{z}_k = \frac{coupe(C_k, \overline{C}_k)}{|C_k|}$$

- Ainsi

$$RatioCut(C_1, ..., C_K) = \sum_{k=1}^{K} \mathbf{z}_k^\top L \mathbf{z}_k = Trace(Z^\top L Z)$$

- En relâchant la contrainte sur les valeurs possibles des $\mathbf{z}_k$, on montre que ce sont les k vecteurs propres associés aux plus petites valeurs propres.

Chaque sommet du graphe représente un individu.
Il existe plusieurs façons pour définir les arêtes.

- Pleinement connecté : tout les sommets sont reliés entre eux.
- Connecté au R-voisinage : tout les sommets distants d'au plus R sont connectés entre eux.
- Connecté au k-plus proches voisins : connecte chaque sommet à ses k plus proches voisins. Attention, le graphe devient orienté. Pour avoir un graphe non orienté, soit
  - on connecte si il y a une arête dans chaque sens (ET).
  - on connecte si il y a au moins une arête (OU).

Des données aux graphes
Affinité

Maintenant, on peut ajouter du poids aux arêtes.

- Jusqu'à présent, on a utilisé des distances. Deux individus similaires sont séparés par une faible distance.
- Pour construire notre graphe, on a besoin de définir une affinité. L'affinité entre deux individus est forte si les individus sont similaires.
- On peut construire une affinité à partir d'une distance en utilisant un noyau. Par exemple (noyau gaussien),

$$a_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right)$$

- Le choix du noyau est très important en pratique.

Partionnement spectrale
Algorithme

On peut enfin donner l'algorithme,

1. A partir des données, définir un graphe de connexion entre individus et une matrice d'affinité A.
2. Calculer le Laplacien $L = D - A$.
3. Trouver les K plus petits vecteurs propres de L (les $\mathbf{z}_k$). Soit U la matrice contenant les $\mathbf{z}_k$ en colonne.
4. Soit $\mathbf{u}_i^\top$ $(i = 1..N)$ les lignes de U, partitionner les $\mathbf{u}_i$ en utilisant les K-moyennes.
5. Associer $\mathbf{x}_i$ à la partition de $\mathbf{u}_i$.

Partionnement spectrale
Variante : la coupe normalisée

- Une meilleure alternative au critère *RatioCut* est celui donné par la coupe normalisée :

$$NCut(C_1, ...C_K) = \sum_{i=1}^{K} \frac{coupe(C_i, \overline{C}_i)}{vol(C_i)},$$

où le volume de $vol(C_i)$ vaut la somme des poids des arêtes allant de C_i à C_i.

- Comparé à *RatioCut*, *NCut* prend en compte les similarités inter parties. Il semble donc plus approprié au partitionnement.
- L'algorithme est globalement le même, sauf que cette fois les $\mathbf{u}_i$ sont solutions de $L\mathbf{z} = \lambda D\mathbf{z}$
- Ce sont donc les vecteurs propres de $L_{norm} = D^{-1}L$.