Spectral clustering : introduction

- Le **spectral clustering** est une méthode moderne de classification automatique qui prend en entrée un graphe valué et donne en sortie une partition en k clusters.
- La méthode repose sur la **décomposition spectrale** de la **matrice Laplacienne** associée à la matrice d'adjacence pondérée du graphe.
- En théorie spectrale de graphe, on montre que les vecteurs propres principaux de la matrice Laplacienne permettent de retrouver les **composantes connexes** du graphe lorsque celles-ci existent.
- Si le graphe est globalement connexe alors ces vecteurs propres représentent un **plongement des noeuds dans un espace Euclidien** au sein duquel, on peut raisonner afin de déterminer comment séparer les noeuds et former des clusters de sorte à minimiser la somme des poids entre deux clusters distincts (graph cuts).

Spectral clustering : introduction (suite)

- Intuitivement, le spectral clustering a un lien fort avec notre pb. de consensus clustering car nous cherchons typiquement des composantes connexes (cf illustration slide 8). Si le graphe est connexe alors, nous cherchons à créer des composantes connexes en supprimant des arêtes et dans cette perspective, on vise à supprimer un minimum d'arête avec un faible poids (graph min cuts).
- Dans ce qui suit, nous procèderons de la façon suivante :
  - Rappels de définitions en théorie de graphe.
  - Présentation de la mat. des degrés, la mat. Laplacienne et sa version normalisée.
  - Présentation formelle des propriétés de la mat. Laplacienne.
  - Pseudo-code du spectral clustering.
  - Lien avec les pbs de coupe dans des graphes.

Graphe de similarités

- La modélisation employée est celle de **graphe non-orienté pondéré**.

Définition. (Graphe non-orienté pondéré)

Un **graphe non-orienté** *G est défini par la donnée de deux ensembles :*

- $\mathbb{O} = \{X_1, \ldots, X_{\mathrm{n}}\}$ *dont les éléments sont appelés* **sommets** *(ou* **noeuds***).* $|\mathbb{O}| = \mathrm{n}$ *est le nb. de sommets. On dit alors que G est d'ordre* n*.*
- $\mathbb{E} = \{e_1, \ldots, e_{\mathrm{m}}\}$ *dont les éléments sont des paires non-ordonnées* $(X_i, X_{i'})$ *de sommets que l'on appelle* **arêtes**. $|\mathbb{E}| = \mathrm{m}$ *désigne le nombre d'arêtes.*

$G = (\mathbb{O}, \mathbb{E})$ est **pondéré** *s'il existe une fonction* $\mathrm{W} : \mathbb{E} \to \mathbb{R}$ *donnant une valuation à toute arête de* $\mathbb{E}$*. On écrira dans ce cas* $G = (\mathbb{O}, \mathbb{E}, \mathrm{W})$*.*

- Dans le contexte du consensus clust., nous avons $\mathrm{W}(X_i, X_{i'}) = \mathbf{C}_{ii'}$, le nb. de partitions ayant mis X_i et $X_{i'}$ dans le même cluster.

Représentation matricielle

Définition. (Matrice d'adjacence)

Soit $G = (\mathbb{O}, \mathbb{E})$ un graphe non-orienté d'ordre n*. La* **matrice d'adjacence** *de G est une matrice carrée binaire d'ordre* n *notée* **A***. Son terme général est défini comme suit :*

$$a_{ii'} = \begin{cases} 1 & \text{si } (X_i, X_{i'}) \in \mathbb{E} \\ 0 & \text{sinon} \end{cases}$$

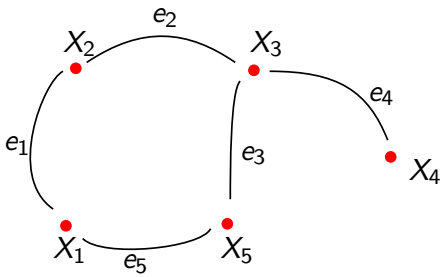
Si G est pondéré nous noterons plutôt par **W** *sa matrice d'adjacence et dans ce cas, nous avons (avec un peu d'abus de notation) :*

$$w_{ii'} = \mathrm{W}(X_i, X_{i'})$$

Dans ce cas on écrira aussi $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$ *pour* $G = (\mathbb{O}, \mathbb{E}, \mathrm{W})$*.*

- Ds le cas du consensus clustering, on prend $\mathbf{W} = \mathbf{C}$ défini slide 30.

Exemple de matrice d'adjacence



$$\mathbf{A} = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- Si le graphe est pondéré selon les valuations suivantes :

	e_1	e_2	e_3	e_4	e_5
$W(e_j)$	2	3	1	5	6

- On obtient alors la matrice d'adjacence pondérée suivante :

$$\mathbf{W} = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 0 & 6 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 5 & 1 \\ 0 & 0 & 5 & 0 & 0 \\ 6 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Degré des sommets et matrice des degrés

- Dans ce qui suit, nous supposons que G est non-orienté et pondéré.
- Nous définissons le degré du sommet X_i par :

$$d_i = \sum_{i'=1}^{\mathrm{n}} w_{ii'}$$

- Rq : comme $\mathbf{W}$ est symétrique on a aussi $d_i = \sum_{i'=1}^{\mathrm{n}} w_{i'i}$.
- La matrice des degrés est une matrice carrée diagonale d'ordre n notée $\mathbf{D}$ de terme général, $\forall i, i' = 1, \ldots, \mathrm{n}$:

$$\mathbf{D}_{ii'} = \begin{cases} d_i & \text{si } i = i' \\ 0 & \text{sinon} \end{cases}$$

Matrice Laplacienne d'un graphe

Définition. (Matrice Laplacienne)

Soit $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$ un graphe non-orienté pondéré. La matrice Laplacienne (non normalisée) de G, notée $\mathbf{L}$, est une matrice carré de même ordre que $\mathbf{W}$ définie comme suit :

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

où $\mathbf{D}$ est la matrice des degrés.

- Exemple :

$$\mathbf{W} = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 0 & 6 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 5 & 1 \\ 0 & 0 & 5 & 0 & 0 \\ 6 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array} \rightarrow \mathbf{L} = \begin{array}{c} \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \begin{array}{c} \begin{array}{ccccc} X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{pmatrix} 8 & -2 & 0 & 0 & -6 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 9 & -5 & -1 \\ 0 & 0 & -5 & 5 & 0 \\ -6 & 0 & -1 & 0 & 7 \end{pmatrix} \end{array}$$

Propriétés de la matrice Laplacienne

Propriété.

Soit $\mathbf{L}$ la matrice Laplacienne d'un graphe $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$ alors :

- *Pour tout vecteur $\mathbf{f} \in \mathbb{R}^{\mathrm{n}}$:*

$$\mathbf{f}^\top \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,i'=1}^{\mathrm{n}} w_{ii'} (f_i - f_{i'})^2$$

- *$\mathbf{L}$ est symétrique et semi-définie positive (valeurs propres ≥ 0).*
- *La plus petite valeur propre de $\mathbf{L}$ est 0 et son vecteur propre associé est $\mathbf{1}$ (vecteur rempli de 1).*
- *$\mathbf{L}$ a n valeurs propres réelles, non-négatives :*

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{\mathrm{n}}$$

Propriétés de la matrice Laplacienne (suite)

Propriété.

Soit G un graphe non-orienté pondéré d'ordre n dont les valuations sont non-négatives. Alors, l'ordre de multiplicité k de la valeur propre 0 de la matrice Laplacienne $\mathbf{L}$ est le nombre de composantes connexes du graphe que l'on notera $C_1, \ldots, C_\mathrm{k}$. De plus, le sous-espace propre associé à la valeur propre 0 est engendré par les vecteurs indicateurs $\mathrm{Ind}_{C_1}, \ldots, \mathrm{Ind}_{C_\mathrm{k}}$ de ces composantes.

- Rappel : une composante connexe est un sous-graphe tel que ses sommets sont connexes.
- Le vecteur indicateur Ind_{C_l} appartient à $\{0,1\}^\mathrm{n}$ et $[\mathrm{Ind}_{C_l}]_i = 1$ si $X_i \in C_l$ et $[\mathrm{Ind}_{C_l}]_i = 0$ sinon.
- Il est intéressant de noter les liens entre ce résultat issu de la théorie des graphes et la problématique de clustering et notamment le partitionnement en k classes à partir d'une matrice de similarités.

Matrice Laplacienne **normalisée** d'un graphe

- On démontre qu'une version normalisée de la matrice Laplacienne a de meilleures propriétés asymptotiques.

Définition. (Matrice Laplacienne normalisée)

Soit $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$ de matrice Laplacienne $\mathbf{L}$. La matrice Laplacienne normalisée de G, notée $\mathbf{L}_n$, est une matrice carrée de même ordre que $\mathbf{W}$ définie comme suit :

$$\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

où $\mathbf{D}$ est la matrice des degrés.

- Il existe un autre type de normalisation (de type random walk) : $\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}$ [Shi and Malik, 2000] mais nous ne l'étudierons pas.

Propriétés de la matrice Laplacienne **normalisée**

Propriété.

*Soit $\mathbf{L}_n$ la matrice Laplacienne **normalisée** de $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$ alors :*

- *Pour tout vecteur $\mathbf{f} \in \mathbb{R}^\mathrm{n}$:*

$$\mathbf{f}^\top \mathbf{L}_n \mathbf{f} = \frac{1}{2}\sum_{i,i'=1}^{\mathrm{n}} w_{ii'}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_{i'}}{\sqrt{d_{i'}}}\right)^2$$

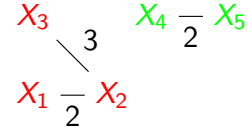
- *$\mathbf{L}_n$ est symétrique et sdp et possède n valeurs propres réelles, non-négatives :*

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_\mathrm{n}$$

- *L'ordre de multiplicité k de la valeur propre 0 est le nombre de composantes connexes de G que l'on notera $C_1, \ldots, C_\mathrm{k}$. Le sous-espace propre associé à la valeur propre 0 est engendré par les vecteurs $\mathbf{D}^{1/2}\mathrm{Ind}_{C_1}, \ldots, \mathbf{D}^{1/2}\mathrm{Ind}_{C_\mathrm{k}}$.*

Exemple de matrice $\mathbf{L}_n$ et décomposition spectrale

- $\mathbb{O} = \{X_1, X_2, X_3, X_4, X_5\}$
- $\mathbb{E} = \{\underbrace{(X_1, X_2)}_{2}, \underbrace{(X_2, X_3)}_{3}, \underbrace{(X_4, X_5)}_{2}\}$



$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \qquad \rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\rightarrow \mathbf{L} = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -2 & 2 \end{pmatrix} \qquad \rightarrow \mathbf{L}_n = \begin{pmatrix} 1 & -.63 & 0 & 0 & 0 \\ -.63 & 1 & -.77 & 0 & 0 \\ 0 & -.77 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$\rightarrow$ Spectre de $\mathbf{L}_n$: {2,2,1,0,0}

$$\rightarrow \mathbf{D}^{1/2}\mathrm{Ind}_{C_1} = \begin{pmatrix} 1.41 \\ 2.24 \\ 1.73 \\ 0 \\ 0 \end{pmatrix} \text{ et } \mathbf{D}^{1/2}\mathrm{Ind}_{C_2} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1.41 \\ 1.41 \end{pmatrix}.$$

Pseudo-code des méthodes de spectral embedding (clustering)

1 **Input : W** (mat. de sim.), normalisation ou pas, k (nb. de classes)
(2 Construire un graphe de voisinage à partir de **W** -facultatif-)
3 Construire la matrice Laplacienne non-normalisée **L** ou normalisée $\mathbf{L}_n$
4 Calculer $\mathbf{f}^1, \ldots, \mathbf{f}^{\mathrm{k}}$, les k premiers vect. propres de **L** ou $\mathbf{L}_n$
5 Construire $\mathbf{F} = \begin{pmatrix} \mathbf{f}^1 & \ldots & \mathbf{f}^{\mathrm{k}} \end{pmatrix} \in \mathbb{R}^{\mathrm{n} \times \mathrm{k}}$ la matrice des k vect. propres mis en colonne
6 Si $\mathbf{L}_n$ est utilisée, normer les vecteurs lignes de **F**
7 Utiliser les k-means pour partitionner les n lignes de **F**
8 **Ouput : F**

Lien avec le pb. de coupe de graphe

- La décomposition spectrale de la matrice Laplacienne d'un graphe est en lien avec un problème de coupe dans un graphe. Soit un graphe de similarité $G = (\mathbb{O}, \mathbb{E}, \mathbf{W})$. L'objectif est de trouver une partition des sommets $\mathbb{O}$ de sorte à ce que la somme des poids des arêtes des sommets d'une même classe/cluster soit grande et que la somme des poids des arêtes de sommets de classes/clusters distincts soit petite. Il s'agit d'un pb. de **partitionnement de graphe**.
- Soit deux classes C_m et $C_{m'}$. Par abus de notations, soit $\mathrm{W}(C_m, C_{m'})$ la somme des poids des arêtes entre toutes les paires de sommets adjacents avec l'un dans C_m et l'autre dans $C_{m'}$:

$$\mathrm{W}(C_m, C_{m'}) = \sum_{X_i \in C_m} \sum_{X_{i'} \in C_{m'}} \mathrm{W}(X_i, X_{i'})$$

- Notons également par $\overline{C}_m$, le complémentaire de C_m par rapport à $\mathbb{O}$.

Critères de coupe dans un graphe

- Pour résoudre le pb. de coupe de graphe par une partition en s classes les trois critères suivants sont proposés dans la litérature [Von Luxburg, 2007] :

$$\mathrm{Cut}(C_1, \ldots, C_{\mathrm{s}}) = \frac{1}{2} \sum_{m=1}^{\mathrm{s}} \mathrm{W}(C_m, \overline{C}_m)$$

$$\mathrm{RatioCut}(C_1, \ldots, C_{\mathrm{s}}) = \frac{1}{2} \sum_{m=1}^{\mathrm{s}} \frac{\mathrm{W}(C_m, \overline{C}_m)}{|C_m|} = \sum_{m=1}^{\mathrm{s}} \frac{\mathrm{Cut}(C_m, \overline{C}_m)}{|C_m|}$$

$$\mathrm{NCut}(C_1, \ldots, C_{\mathrm{s}}) = \frac{1}{2} \sum_{m=1}^{\mathrm{s}} \frac{\mathrm{W}(C_m, \overline{C}_m)}{\mathrm{Vol}(C_m)} = \sum_{m=1}^{\mathrm{s}} \frac{\mathrm{Cut}(C_m, \overline{C}_m)}{\mathrm{Vol}(C_m)}$$

où $\mathrm{Vol}(C_m) = \sum_{X_i \in C_m} d_i$ et, $d_i = \sum_{i'=1}^{\mathrm{n}} \underbrace{\mathrm{W}(X_i, X_{i'})}_{w_{ii'}}$ (cf slide 38).

Minimisation des critères de coupe

- Le critère Cut ne donne pas de solutions intéressantes en pratique. Pour le cas $\mathrm{s} = 2$ par ex., on obtient souvent C_1 un singleton et C_2 le reste des individus. Il y donc implicitement pour Cut un déséquilibre extrême entre la taille des deux clusters.
- Les autres critères RatioCut et Ncut (Normalized Cut) ont été proposés pour corriger cet inconvénient. En effet, le minimum des fonctions $\sum_{m=1}^{\mathrm{s}} 1/|C_m|$ et $\sum_{m=1}^{\mathrm{s}} 1/\mathrm{Vol}(C_m)$ est atteint lorsque les caridnaux et les volumes sont identiques ce qui encourage la détection de cluster homogènes en taille ou en volume.
- Toutefois, cette normalisation rend le pb. de coupe NP-dur.
- Le spectral embedding peut alors être vu comme l'étape cruciale pour résoudre approximativement le pb. de partitionnement de graphe par un type de pb. d'optimisation relaxé.

Minimisation de RatioCut (suite)

- Pour minimiser RatioCut, l'encodage d'une partition $\{C_1, \ldots, C_s\}$ se fait par une matrice $\mathbf{Y} = (y_{im})$ de taille $n \times s$ avec :

$$y_{im} = \begin{cases} 1/\sqrt{|C_m|} & \text{si } X_i \in C_m \\ 0 & \text{sinon} \end{cases}$$

- Soit $\mathbf{y}^m$ la colonne m de $\mathbf{Y} = \begin{pmatrix} \mathbf{y}^1 & \ldots & \mathbf{y}^s \end{pmatrix}$. Le terme général de $\mathbf{y}^m$ est $[\mathbf{y}^m]_i = y_{im} = 1/\sqrt{|C_m|}$ si $X_i \in C_m$ et 0 sinon. On interprète la mat. Laplacienne $\mathbf{L}$ (qui est symétrique et sdp) telle une forme quadratique. On peut alors montrer que,

$$[\mathbf{y}^m]^\top \mathbf{L} \mathbf{y}^m = \frac{\text{Cut}(C_m, \overline{C}_m)}{|C_m|}$$

et également que,

$$[\mathbf{y}^m]^\top \mathbf{L} \mathbf{y}^m = [\mathbf{Y}^\top \mathbf{L} \mathbf{Y}]_{mm}$$

Minimisation de RatioCut (suite)

- En combinant les précédents faits, on a la formulation suivante :

$$\text{RatioCut}(\mathbf{Y}) = \sum_{m=1}^{s} [\mathbf{Y}^\top \mathbf{L} \mathbf{Y}] = \text{Tr}(\mathbf{Y}^\top \mathbf{L} \mathbf{Y})$$

où Tr(.) est la trace d'une matrice carrée.

- On remarque par ailleurs qu'une autre propriété caractéristique de $\mathbf{Y}$ est que ses colonnes vérifient la propriété d'orthogonalité si bien que :

$$\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_s$$

- Le pb. d'optimisation se formalise alors comme suit :

$$\min_{\mathbf{Y}} \text{Tr}(\mathbf{Y}^\top \mathbf{L} \mathbf{Y})$$
$$\text{slc. } \begin{cases} y_{im} = \begin{cases} 1/\sqrt{|C_m|} & \text{si } X_i \in C_m \\ 0 & \text{sinon} \end{cases} \\ \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_s \end{cases}$$

Minimisation de RatioCut (suite)

- $\mathbf{Y}$ est une mat. d'affectation (pondérée) et a une structure discrète. Ainsi la min. de RatioCut par rapport à $\mathbf{Y}$ est un pb. d'optimisation combinatoire qui est NP-dur.
- En pratique, on utilise une relaxation du pb. afin d'obtenir des solutions approchées dans des temps raisonnables.
- La relaxation classique ici consiste à considérer $\mathbf{F} \in \mathbb{R}^{n \times s}$ où f_{im} est un réel quelconque (au lieu d'une fraction de type "$1/\sqrt{|C_m|}$" qui est un ensemble discret de valeurs).
- Le pb. relaxé est donc formalisé comme suit :

$$\min_{\mathbf{F} \in \mathbb{R}^{n \times s}} \text{Tr}(\mathbf{F}^\top \mathbf{L} \mathbf{F})$$
$$\text{slc. } \mathbf{F}^\top \mathbf{F} = \mathbf{I}_s$$

Formulation algébrique de RatioCut

- Supposons que $\mathbf{f} \in \mathbb{R}^n$, et étudions la forme quadratique associée à $\mathbf{L}$ donnée par $q(\mathbf{f}) = \mathbf{f}^\top \mathbf{L} \mathbf{f}$ qui est au coeur de la fct. objectif RatioCut :

$$\begin{aligned} \mathbf{f}^\top \mathbf{L} \mathbf{f} &= \mathbf{f}^\top (\mathbf{D} - \mathbf{W}) \mathbf{f} \\ &= \mathbf{f}^\top \mathbf{D} \mathbf{f} - \mathbf{f}^\top \mathbf{W} \mathbf{f} \\ &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,i'=1}^{n} f_i f_{i'} w_{ii'} \\ &= \frac{1}{2} \left(\sum_{i=1}^{n} d_i f_i^2 + \sum_{i'=1}^{n} d_{i'} f_{i'}^2 - 2 \sum_{i,i'=1}^{n} f_i f_{i'} w_{ii'} \right) \\ &= \frac{1}{2} \sum_{i,i'}^{n} w_{ii'} (f_i - f_{i'})^2 \end{aligned}$$

Cette dernière expression est celle de la propriété slide 40 !

Formulation algébrique de RatioCut (suite)

- Soit le pb. relaxé de part. de G en s clusters qui min. le RatioCut :

$$\min_{\mathbf{F}\in\mathbb{R}^{\mathrm{n}\times\mathrm{s}}} \overbrace{\sum_{m=1}^{\mathrm{s}} [\mathbf{f}^m]^\top \mathbf{L}\mathbf{f}^m}^{\mathrm{Tr}(\mathbf{F}^\top\mathbf{L}\mathbf{F})}$$
$$\text{slc. } \mathbf{F}^\top\mathbf{F} = \mathbf{I}_{\mathrm{s}}$$

- Celui ci s'interprète donc comme la recherche de s vecteurs de $\mathbb{R}^{\mathrm{n}}$ de norme unitaire qui sont mutuellement orthogonaux et qui minimisent la forme quadratique associée à $\mathbf{L}$.
- La solution de ce pb. est bien connue : il s'agit de déterminer les s vecteurs propres de $\mathbf{L}$ associée aux plus petites valeurs propres de $\mathbf{L}$!
- La décomposition spectrale de $\mathbf{L}$ correspond à la solution du pb. relaxé et ce qu'on obtient n'est donc pas en général une partition (sauf si composantes connexes comme indiqué en slide 40).

RatioCut, décomp. spec. de $\mathbf{L}$ et spectral embedding

- Les s vect. propres principaux de $\mathbf{L}$ nous donnent en fait une base d'un espace vect. au sein de laquelle, nous pouvons représenter les somments (càd. les objets). On parle alors de **plongement spectral**. Les dist. Euclidiennes dans cet espace de dimension s (avec s petit donc réduction de dimension) visent à approximer les mesures de la relation de voisinage/affinité/similarité données dans $\mathbf{W}$.
- Nous pouvons constater ceci au travers de l'expression suivante :

$$\mathbf{f}^\top\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i,i'}^{\mathrm{n}} w_{ij}(f_i - f_{i'})^2$$

- Pour que $\mathbf{f}$ minimise la forme quad. on voit en effet qu'il faut que $(f_i - f_{i'})^2$ soit petit quand $w_{ii'}$ est grand. Autrement dit, si X_i et $X_{i'}$ sont des proches voisins avec $w_{ii'}$ grand alors ils auront tendance à avoir des coordonnées proches sur l'axe engendré par $\mathbf{f}$ avec $f_i \approx f_{i'}$.

Minimisation de NCut

- Les développement précédents ont permis de mettre en lumière le rôle de la mat. Laplacienne $\mathbf{L}$ dans le pb. de partitionnement de graphe et le spectral embedding par minimisation de RatioCut.
- De façon très similaire, on peut montrer des propriétés identiques entre d'une part la mat. Laplacienne normalisée $\mathbf{L}_n$ et d'autre part, le critère Ncut donné slide 47.
- Dans le cas du NCut la partition recherchée $C_1, \ldots, C_{\mathrm{s}}$ est encodée par une mat. $\mathbf{Y} = (y_{im})$ de terme général :

$$y_{im} = \begin{cases} 1/\sqrt{\mathrm{Vol}(C_m)} & \text{si } X_i \in C_m \\ 0 & \text{sinon} \end{cases}$$

- Dans ce cas on a $\mathbf{Y}^\top\mathbf{D}\mathbf{Y} = \mathbf{I}_{\mathrm{s}}$ et le critère NCut($\mathbf{Y}$) s'écrit :

$$\mathrm{NCut}(\mathbf{Y}) = \sum_{m=1}^{\mathrm{s}} [\mathbf{Y}^\top\mathbf{L}\mathbf{Y}] = \mathrm{Tr}(\mathbf{Y}^\top\mathbf{L}\mathbf{Y})$$

Minimisation de NCut (suite)

- La min. de NCut($\mathbf{Y}$) sous les contraintes précédentes est un pb. discret NP-dur mais comme précédemment, on peut relaxer la contrainte "discrète" du pb.
- Dans cette perspective et afin de traiter plus facilement $\mathbf{Y}^\top\mathbf{D}\mathbf{Y} = \mathbf{I}_{\mathrm{s}}$ la contrainte d'orthogonalité de $\mathbf{Y}$ mais par rapport à la métrique diagonale $\mathbf{D}$, on pose $\mathbf{F} = \mathbf{D}^{1/2}\mathbf{Y}$. Avec ce chang. de var. on a :

$$\mathbf{F}^\top\mathbf{F} = \mathbf{Y}^\top\mathbf{D}\mathbf{Y}$$
$$\mathrm{NCut}(\mathbf{F}) = \mathrm{Tr}(\mathbf{F}^\top\mathbf{D}^{1/2}\mathbf{L}\mathbf{D}^{1/2}\mathbf{F}) = \mathrm{Tr}(\mathbf{F}^\top\mathbf{L}_n\mathbf{F})$$

- La relaxation consiste alors à prendre $\mathbf{F}\in\mathbb{R}^{\mathrm{n}\times\mathrm{s}}$ et on résoud :

$$\min_{\mathbf{F}\in\mathbb{R}^{\mathrm{n}\times\mathrm{s}}} \mathrm{Tr}(\mathbf{F}^\top \underbrace{\mathbf{D}^{1/2}\mathbf{L}\mathbf{D}^{1/2}}_{\mathbf{L}_n}\mathbf{F})$$
$$\text{slc. } \mathbf{F}^\top\mathbf{F} = \mathbf{I}_{\mathrm{s}}$$

- La solution est alors donnée par la décomposition spectrale de $\mathbf{L}_n$!

NCut, décomp. spec. de $\mathbf{L}_n$ et spectral embedding

- Les interprétations en terme de plongement spectral sont les mêmes que précédemment : les vecteurs principaux de $\mathbf{L}_n$ permettent de représenter dans un espace vectoriel les sommets du graphe de sorte à respecter les relations de voisinage.
- La différence entre RatioCut et NCut est liée à la pondération de chq. classe. L'expression algébrique de la forme quad. associée à $\mathbf{L}_n$ nous permet de mieux constater l'impact de cette distinction :

$$\mathbf{f}^\top \mathbf{L}_n \mathbf{f} = \frac{1}{2} \sum_{i,i'}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_{i'}}{\sqrt{d_{i'}}} \right)^2$$

- On voit qu'avec NCut et contrairement à RatioCut, la représentation des sommets dans le sous-espace propre est accompagnée d'une pondération qui dépend du degré/volume de chaque individu. En bref, f_i sera d'autant plus proche de $f_{i'}$ que $w_{ii'}$ est grand ET que $\sqrt{d_i}$ est proche de $\sqrt{d_{i'}}$ (objets ayant des degrés similaires).

Quelques références I

Ah-Pine, J. (2018).
An efficient and effective generic agglomerative hierarchical clustering approach.
The Journal of Machine Learning Research, 19(1) :1615–1658.

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000).
Reducing multiclass to binary : A unifying approach for margin classifiers.
Journal of machine learning research, 1(Dec) :113–141.

Alpaydin, E. (2010).
Introduction to Machine Learning.
MIT Press.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999).
When is Nearest Neighbor Meaningful ?
In ICDT.

Breiman, L. (1996).
Bagging predictors.
Machine learning, 24(2) :123–140.

Breiman, L. (2001).
Random forests.
Machine learning, 45(1) :5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984).
Classification and Regression Trees.
Chapman & Hall, New York, NY.

Quelques références II

Chen, T. and Guestrin, C. (2016).
Xgboost : A scalable tree boosting system.
In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794.

Cutler, A., Cutler, D. R., and Stevens, J. R. (2012).
Random forests.
In Ensemble machine learning, pages 157–175. Springer.

Drucker, H. (1997).
Improving regressors using boosting techniques.
In ICML, volume 97, pages 107–115. Citeseer.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004).
Least angle regression.
The Annals of statistics, 32(2) :407–499.

Freund, Y., Schapire, R. E., et al. (1996).
Experiments with a new boosting algorithm.
In Icml, volume 96, pages 148–156.

Friedman, J., Hastie, T., and Tibshirani, R. (2000).
Additive logistic regression : a statistical view of boosting (with discussion and a rejoinder by the authors).
The annals of statistics, 28(2) :337–407.

Friedman, J. H. (2001).
Greedy function approximation : a gradient boosting machine.
Annals of statistics, pages 1189–1232.

Quelques références III

Hastie, T., Rosset, S., Zhu, J., and Zou, H. (2009).
Multi-class adaboost.
Statistics and its Interface, 2(3) :349–360.

Hastie, T., Tibshirani, R., and Friedman, J. (2011).
The Elements of Statistical Learning.
Springer.

Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013).
Understanding variable importances in forests of randomized trees.
Advances in neural information processing systems, 26.

Marcotorchino, F. and Michaud, P. (1982).
Agregation de similarites en classification automatique.
Revue de statistique appliquée, 30(2) :21–44.

Perrone, M. P. and Cooper, L. N. (1992).
When networks disagree : Ensemble methods for hybrid neural networks.
Technical report, Brown Univ Providence Ri Inst for Brain and Neural Systems.

Quinlan, J. (1986).
Induction of decision trees.
Machine Learning, 1(1) :81–106.

Quinlan, J. R. (1993).
C4.5 : programs for machine learning.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quelques références IV

Rokach, L. (2010).
Ensemble-based classifiers.
Artificial Intelligence Review, 33(1-2) :1–39.

Schapire, R. E. (1990).
The strength of weak learnability.
Machine learning, 5(2) :197–227.

Schapire, R. E. and Singer, Y. (1999).
Improved boosting algorithms using confidence-rated predictions.
Machine learning, 37(3) :297–336.

Segal, M. and Xiao, Y. (2011).
Multivariate random forests.
Wiley interdisciplinary reviews : Data mining and knowledge discovery, 1(1) :80–87.

Shi, J. and Malik, J. (2000).
Normalized cuts and image segmentation.
IEEE Transactions on pattern analysis and machine intelligence, 22(8) :888–905.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011).
A survey of clustering ensemble algorithms.
International Journal of Pattern Recognition and Artificial Intelligence, 25(03) :337–372.

Von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and computing, 17(4) :395–416.

Quelques références V

Wolpert, D. H. (1992).
Stacked generalization.
Neural networks, 5(2) :241–259.

Zhou, Y. and Sharma, A. (2017).
Automated identification of security issues from commit messages and bug reports.
In Proceedings of the 2017 11th joint meeting on foundations of software engineering, pages 914–919.

Zhou, Z.-H. (2012).
Ensemble methods : foundations and algorithms.
CRC press.