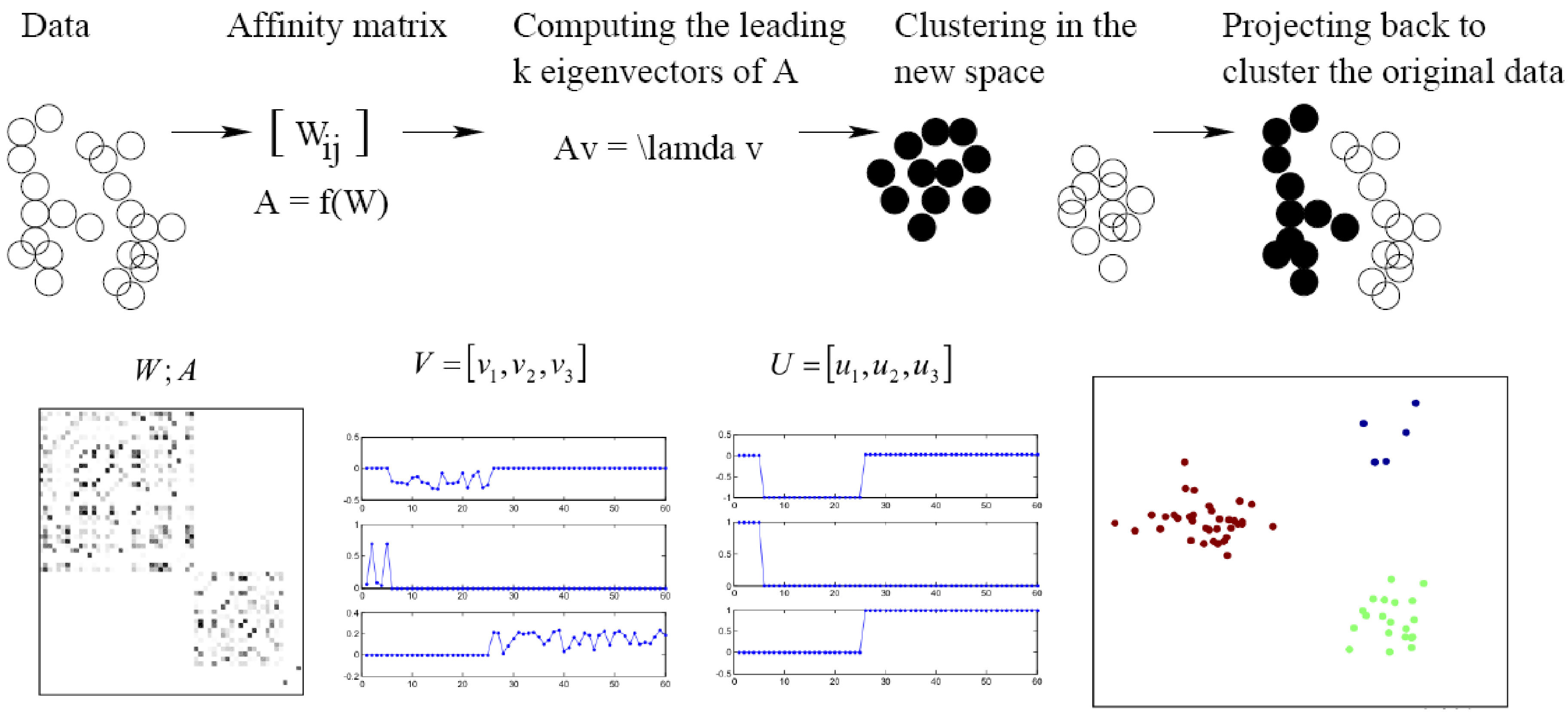
Thème 5: Méthodes avancées – partie 3

Clustering Spectral

Les diapos de cette partie viennent de Dr. Jiawei Han
et Dr. Ralf Möller Dr. Özgür L. Özçep

- Le diapo de Jiawei Han suivant illustre sommairement le fonctionnement de « spectral clustering »

Clustering spectral : illustrations et commentaires



- Clustering spectral : efficace dans des tâches telles que le traitement d'images
- Défi d'extensibilité : le calcul de vecteurs propres sur une grande matrice est coûteux
- Peut être combiné avec d'autres méthodes de clustering, telles que le bi-clustering

Une autre présentation de clustering spectral

- Prof. Dr. Ralf Möller Dr. Özgür L. Özçep
Universität zu Lübeck
Institut für Informationssysteme

Spectral Clustering

Acknowledgements for subsequent slides to

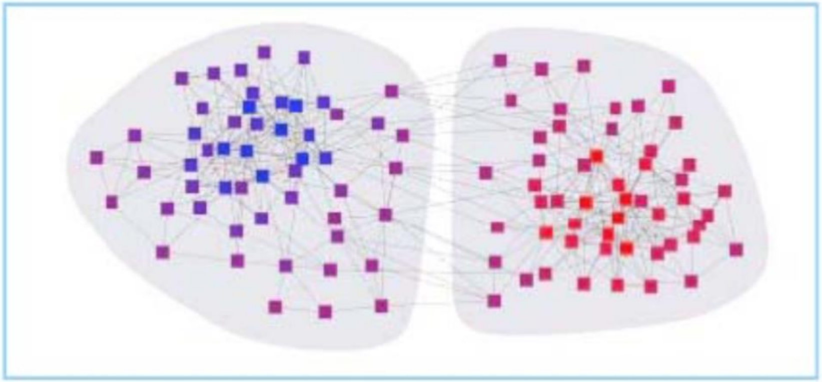
Xiaoli Fern

CS 534: Machine Learning 2011

http://web.engr.oregonstate.edu/~xfern/classes/cs534/

Clustering spectral

- Objets représentés comme sommets *V* d'un graphe *G*
- Sommets connectés par des arêtes *E*
- Les poids associés aux arêtes sont représentés par *W*
  - Grand *W*(*i*,*j*) signifie que les objets *i* et *j* sont très similiaires; petit *W*(*i*,*j*) signifie dissimilaires.

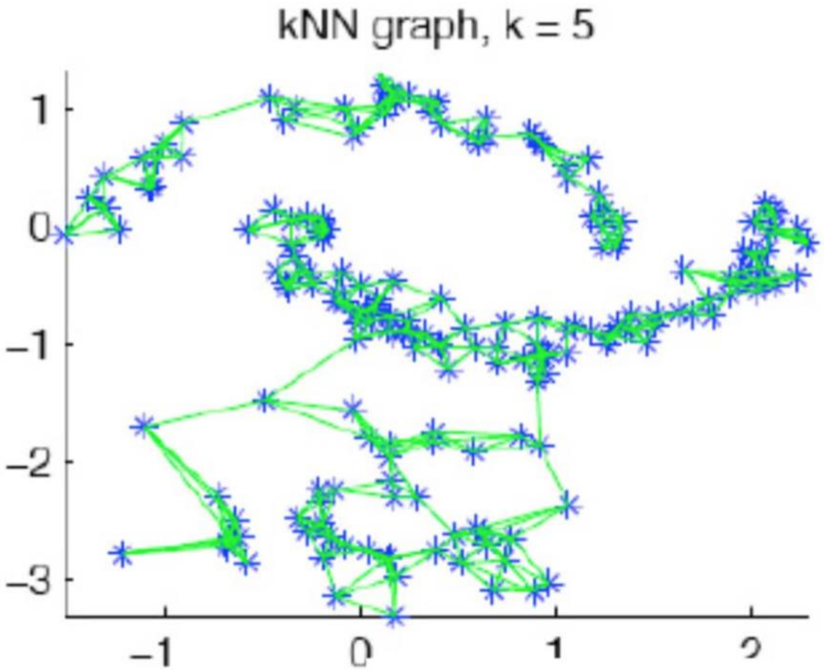


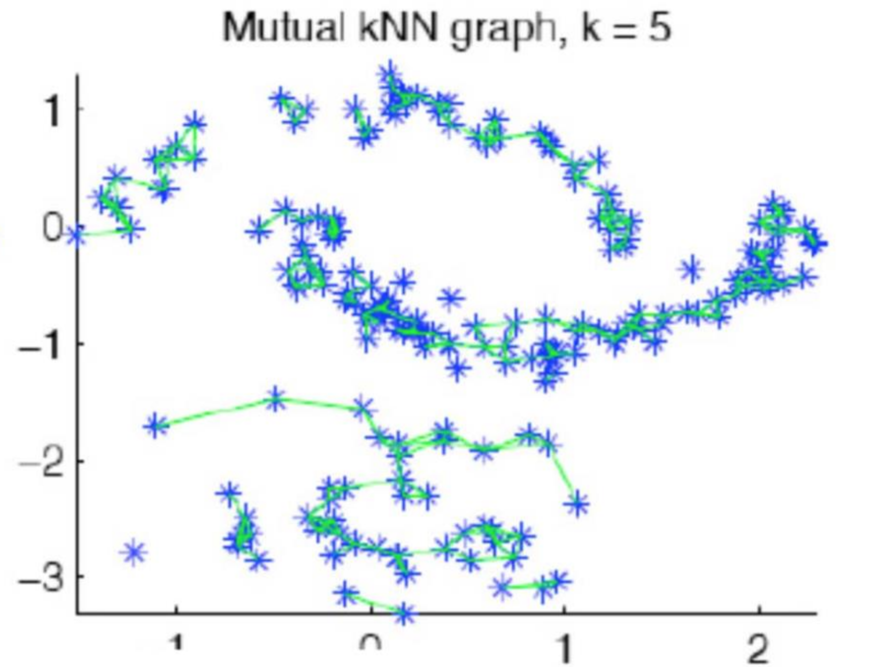
- Les méthodes de clustering qui emploient le "spectre" de la matrice W sont des méthodes de clustering spectral.

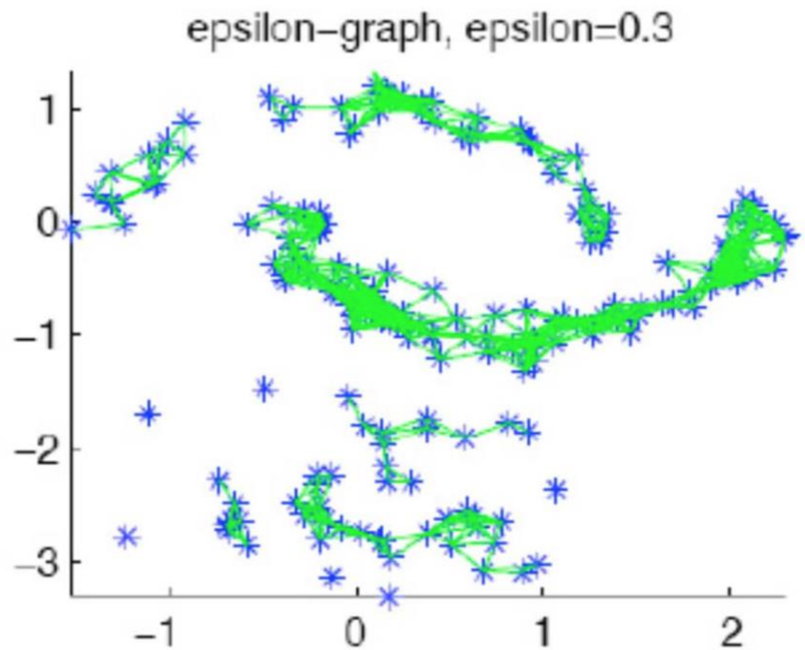
Comment créer le graphe (de similitude)?

- On peut créer
  - Un graph à connexions complètes
  - Un graphe de K plus proches voisins (K-NN)
  - Un graphe de voisinage-ε (ε-neighborhood graph)
- En pratique, on utilise souvent le noyau gaussien pour calculer la similitude entre les objets :

$$W(i,j) = \exp\frac{-\left|x_i - x_j\right|^2}{\sigma^2}$$


kNN graph, k = 5

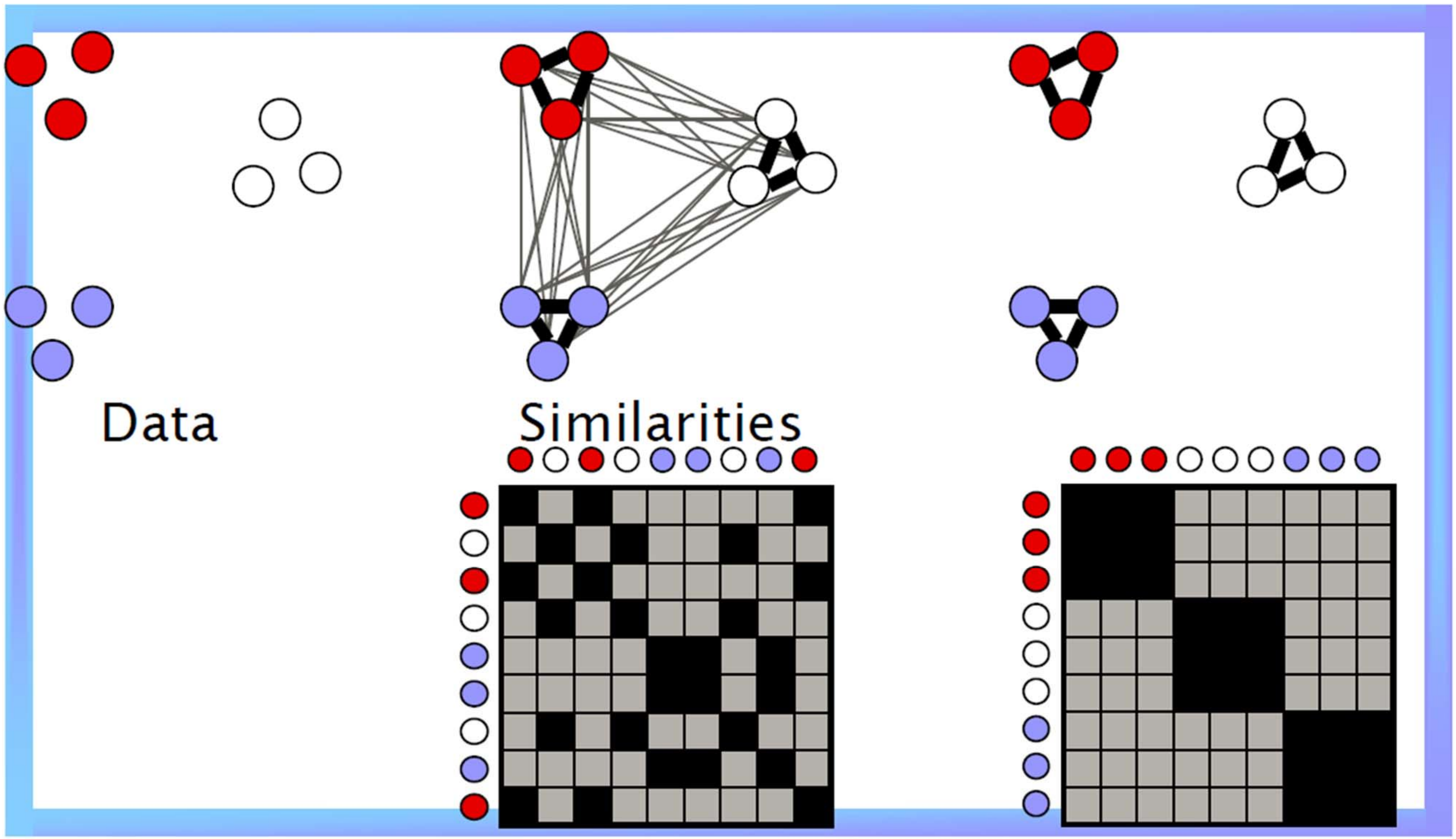

Mutual kNN graph, k = 5


epsilon−graph, epsilon=0.3

Motivations / Objectives

- Il y a de différentes façons d'interpréter le clustering spectral
  - On peut considérer le clustering spectral comme la recherche de partitions du graphe qui minimisent la « coupe normalisée » (Nomalized Cut).
  - Alternativement, nous pouvons également considérer cela comme une marche aléatoire sur le graphe.

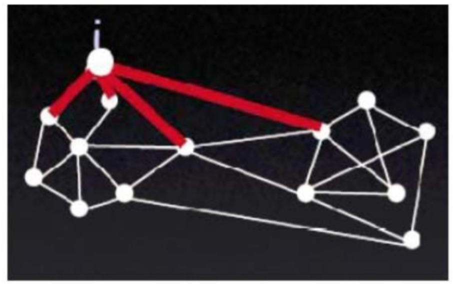
Partitionnement de graphe

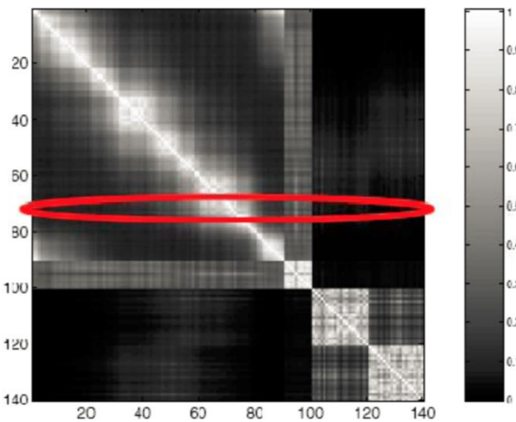


Terminologies des graphes

- Degree of nodes

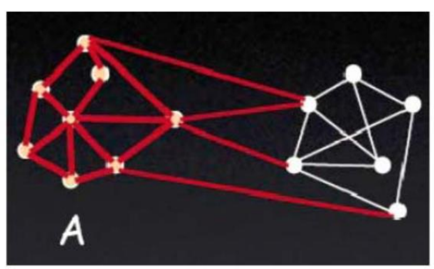
$$d_i = \sum_j w_{i,j}$$

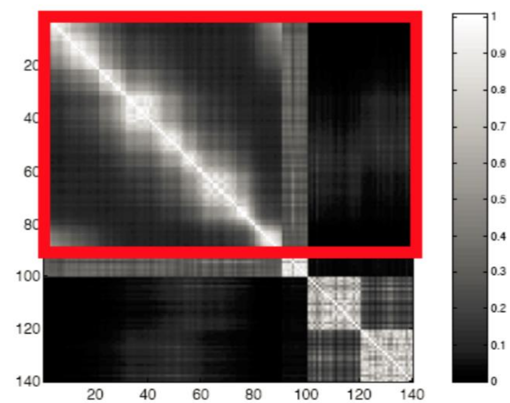




- Volume of a set

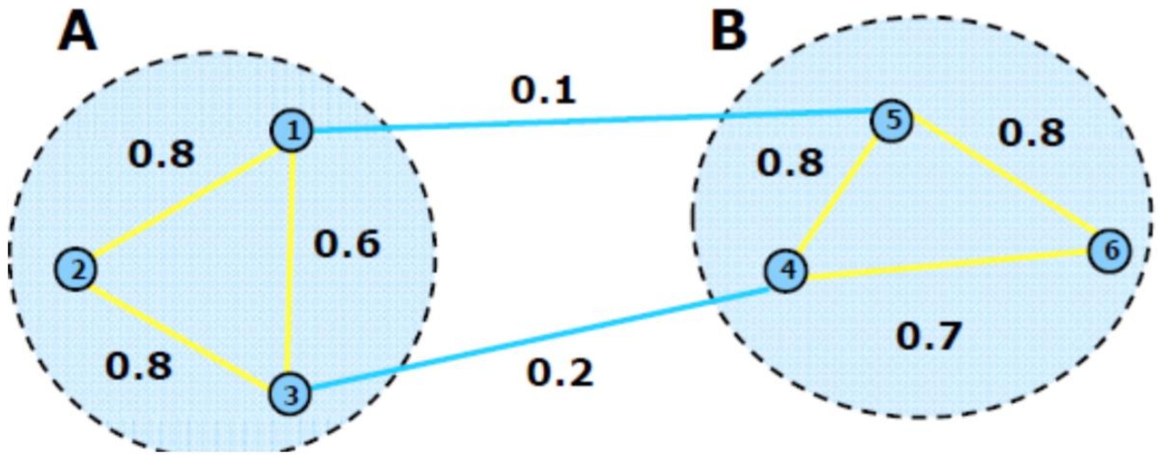
$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$





Coupe de graphe (Graph Cut)

- Soit une partititon d'un graphe en deux parties *A* et *B*



- *Cut*(*A*, *B*) : somme des poids de l'ensemble des arêtes reliant les deux groupes *A* et *B*

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij} = 0.3$$

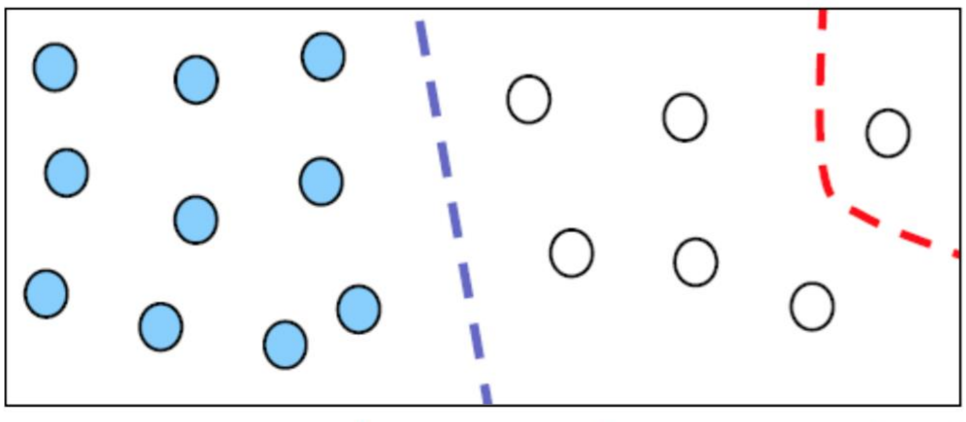
- Objectif : trouver une partition *A* et *B* qui minimise *Cut*(*A*, *B*)

Coupe minimale (fonction objectif)

- Mincut : vise à minimiser le poids des connexions entre groupes

$$\min_{A\cap B=\emptyset, A\cup B=V} Cut(A,B)$$

- Problème :
  - tend vers une solution dégénérative. Ex.

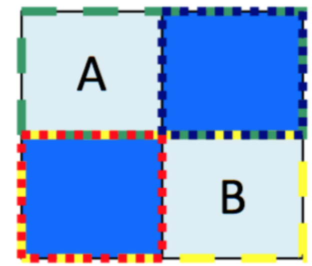


  - Nécessite donc une approche favorisant une solution plus équilibrée

Coupe normalisée (Normalized Cut)

- Considérez la connectivité entre les groupes par rapport au volume de chaque groupe

$$Ncut(A,B) = \frac{cut(A,B)}{Vol(A)} + \frac{cut(A,B)}{Vol(B)}$$



$$Ncut(A,B) = cut(A,B)\frac{Vol(A)+Vol(B)}{Vol(A)Vol(B)}$$

- Le rapport des volumes ci-dessus est minimisé seulement lors que Vol(A) et Vol(B) sont équaux.

Objectif pour optimiser NCut

- Comment minimiser *NCut* ?

$$\text{Let } W \text{ be the similarity matrix, } W(i,j) = W_{i,j};$$
$$\text{Let D be the diag. matrix, } D(i,i) = \sum_j W(i,j);$$
$$\text{Let } x \text{ be a vector in } \{1,-1\}^N, x(i) = 1 \Leftrightarrow i \in A.$$

- Après implifications, on obtient

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^T D y}$$

Rayleigh quotient

Subject to: $y^T D1 = 0$ ($\mathbf{y}$ *takes discrete values*)

NP-Hard!

Résoudre Ncut

- Relâcher le problème d'optimisation en résolvant un système de valeurs propres généralisé dans le continu

$$\min_y y^T(D-W)y \text{ subject to } y^T D y = 1$$

- Lagrangienne :

$$L(y,\lambda) = y^T(D-W)y - \lambda(y^T D y - 1)$$

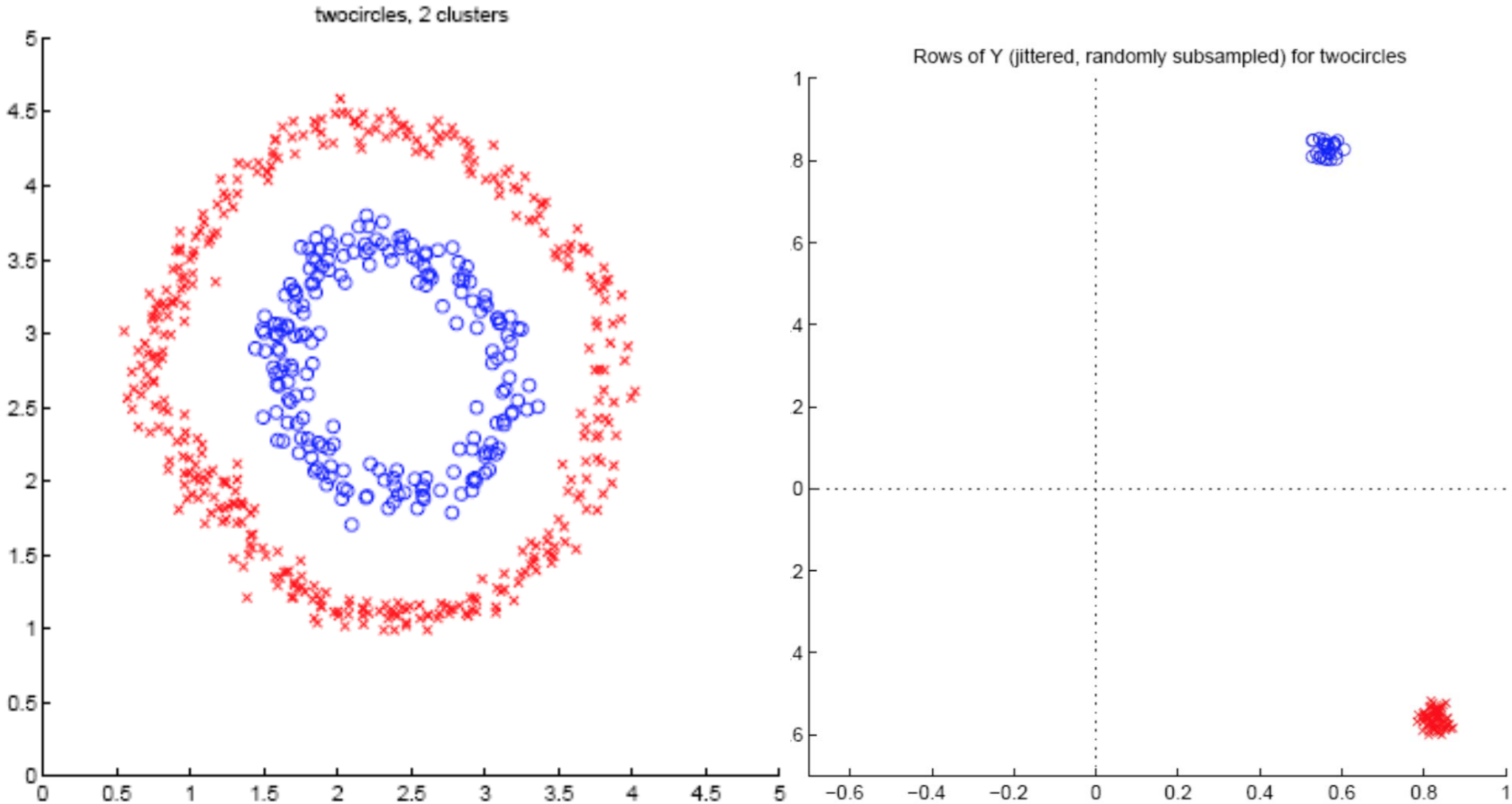
- En imposant les dérivés partiels par rapport à y à zéro

$$(D-W)y = \lambda D y$$

- Noter que $(D-W)1 = 0$, donc the premier vecteur propre est $y_0 = 1$, avec valeur propre 0.
- Le deuxième plus petit vecteur propre est la solution à valeur réelle de ce problème

2-way Normalized Cuts

1. Compute the affinity matrix W, compute the degree matrix (D), D is diagonal and $D(i,i) = \sum_{j \in V} W(i,j)$
2. Solve $(D - W)y = \lambda D y$, where $D - W$ is called the Laplacian matrix
3. Use the eigenvector with the second smallest eigen-value to bipartition the graph into two parts.



Spectral embedding of the data