# Short Introduction to Spectral Clustering

MLSS 2007
Practical Session on Graph Based Algorithms for Machine Learning

Matthias Hein and Ulrike von Luxburg

August 2007

# What is clustering, intuitively?

Given:

- Data set of “objects”
- Some relations between those objects (similarities, distances, neighborhoods, connections, ... )

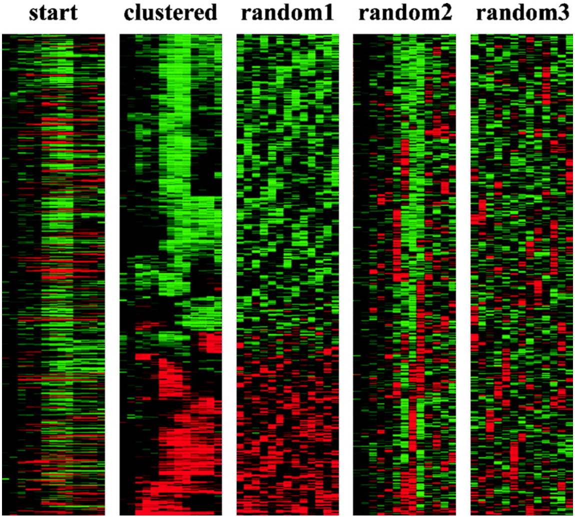
Intuitive goal: Find meaningful groups of objects such that

- objects in the same group are “similar”
- objects in different groups are “dissimilar”

Reason to do this:

- exploratory data analysis
- reducing the complexity of the data
- many more

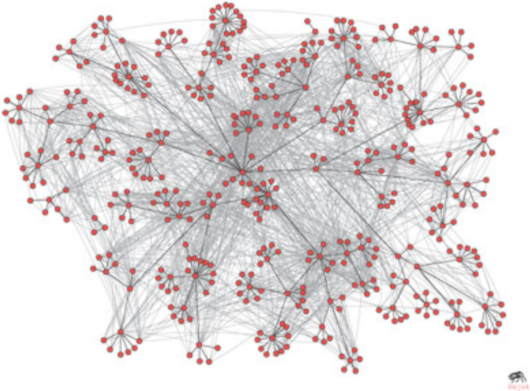
# Example: Clustering gene expression data



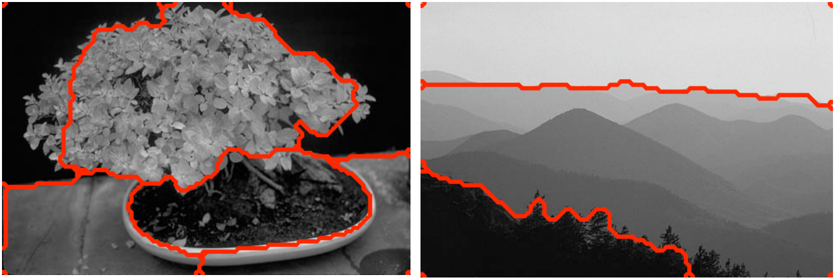

M. Eisen et al., PNAS, 1998

# Example: Social networks

Corporate email communication (Adamic and Adar, 2005)



# Example: Image segmentation

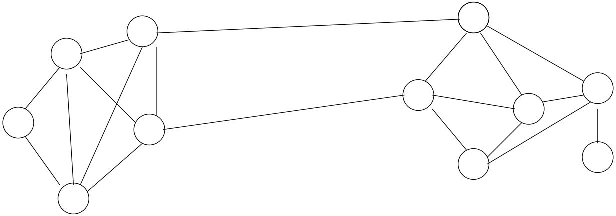


(from Zelnik-Manor/Perona, 2005)

# Spectral clustering on one slide

- Given: data points $X_1, \ldots, X_n$, pairwise similarities $w_{ij} = s(X_i, X_j)$

- Build similarity graph: vertices = data points, edges = similarities

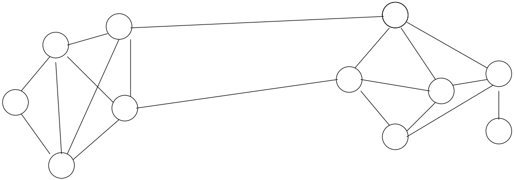


- clustering = find a cut through the graph
  - define a cut objective function
  - solve it

$\rightsquigarrow$ **spectral clustering**

# Graph notation

- $W = (w_{ij})$ adjacency matrix of the graph
- $d_i = \sum_j w_{ij}$ degree of a vertex
- $D = diag(d_1, \ldots, d_n)$ degree matrix
- $|A|$ = number of vertices in $A$
- $\text{vol}(A) = \sum_{i \in A} d_i$



In the following: vector $f = (f_1, ..., f_n)$ interpreted as function on the graph with $f(X_i) = f_i$.

# Clustering using graph cuts

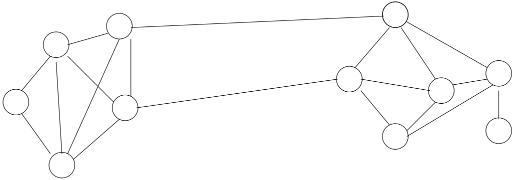
Clustering: within-similarity high, between similarity low

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

Balanced cuts:

$$\text{RatioCut}(A, B) := \text{cut}(A, B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$$

$$\text{Ncut}(A, B) := \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$



Mincut can be solved efficiently, but RatioCut or Ncut is NP hard.

Spectral clustering: relaxation of RatioCut or Ncut, respectively.

# Unnormalized graph Laplacian

Defined as

$$L = D - W$$

Key property: for all $f \in \mathbb{R}^n$

$$\begin{aligned} f'Lf &= f'Df - f'Sf \\ &= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\ &= \frac{1}{2}\left(\sum_i (\sum_j w_{ij}) f_i^2 - 2\sum_{ij} f_i f_j w_{ij} + \sum_j (\sum_i w_{ij}) f_j^2\right) \\ &= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2 \end{aligned}$$

# Unnormalized graph Laplacian (2)

*Spectral properties:*

- L is symmetric (by assumption) and positive semi-definite (by key property)
- Smallest eigenvalue of $L$ is 0, corresponding eigenvector is $\mathbb{1}$
- Thus eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

*First relation between spectrum and clusters:*

- Multiplicity of eigenvalue $0 =$ number $k$ of connected components $A_1, ..., A_k$ of the graph.
- eigenspace is spanned by the characteristic functions $\mathbb{1}_{A_1}, ..., \mathbb{1}_{A_k}$ of those components (so all eigenvectors are piecewise constant).

*Proof:* Exercise

# Normalized graph Laplacians

Row sum (random walk) normalization:

$$L_{\mathrm{rw}} = D^{-1}L \; = \; I - D^{-1}S$$

Symmetric normalization:

$$L_{\mathrm{sym}} = D^{-1/2}LD^{-1/2} \; = \; I - D^{-1/2}SD^{-1/2}$$

Spectral properties similar to the ones of $L$

# Solving Balanced Cut Problems

Relaxation for simple balanced cuts:

$$\min_{A,B} \mathrm{cut}(A,B) \text{ s.t. } |A| = |B|$$

Choose $f = (f_1, ..., f_n)'$ with $f_i = \begin{cases} 1 & \text{if } X_i \in A \\ -1 & \text{if } X_i \in B \end{cases}$

- $\mathrm{cut}(A,B) = \sum_{i \in A, j \in B} w_{ij} = \frac{1}{4} \sum_{i,j} w_{ij}(f_i - f_j)^2 = \frac{1}{4} f' L f$
- $|A| = |B| \implies \sum_i f_i = 0 \implies f^t \mathbb{1} = 0 \implies f \perp \mathbb{1}$
- $\|f\| = \sqrt{n} \sim const.$

$$\min_f f' L f \text{ s.t. } f \perp \mathbb{1},\ f_i = \pm 1,\ \|f\| = \sqrt{n}$$

Relaxation: allow $f_i \in \mathbb{R}$
By Rayleigh: solution $f$ is the second eigenvector of $L$
Reconstructing solution: $X_i \in A \iff f_i >= 0$, $X_i \in B$ otherwise

# Solving Balanced Cut Problems (2)

Similar relaxations work for the other balanced cuts:

- Relaxing RatioCut $\rightsquigarrow$ eigenvectors of $L$ $\rightsquigarrow$ unnormalized spectral clustering
- Relaxing Ncut $\rightsquigarrow$ eigenvectors of $L_{\mathrm{rw}}$ $\rightsquigarrow$ normalized spectral clustering
- Case of $k > 2$ works similar, results in a trace min problem $\min_V \operatorname{Tr} H'LH$ where $V$ is a $n \times k$ orthonormal matrix. Then again Rayleigh-Ritz.

# Spectral clustering - main algorithms

Input: Similarity matrix $S$, number $k$ of clusters to construct

- Build similarity graph
- Compute the first $k$ eigenvectors $v_1, \ldots, v_k$ of the matrix

$$\begin{cases} L & \text{for unnormalized spectral clustering} \\ L_{\text{rw}} & \text{for normalized spectral clustering} \end{cases}$$

- Build the matrix $V \in \mathbb{R}^{n \times k}$ with the eigenvectors as columns
- Interpret the rows of $V$ as new data points $Z_i \in \mathbb{R}^k$

| | $v_1$ | $v_2$ | $v_3$ |
|---|---|---|---|
| $Z_1$ | $v_{11}$ | $v_{12}$ | $v_{13}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $Z_n$ | $v_{n1}$ | $v_{n2}$ | $v_{n3}$ |

- Cluster the points $Z_i$ with the $k$-means algorithm in $\mathbb{R}^k$.

# DemoSpectralClustering

## Exploring Spectral Clustering

- Lowest number of noise dimensions
- Symmetric kNN graph with $k = 10$
- Number of clusters $= 2$

- Play around with data sets Two moons balanced and Three Gaussians (first pick reasonable $\sigma$!)
- Try to understand the plots concerning the eigenvectors and the embedding in $\mathbb{R}^d$
- Increase the number of clusters. Can you predict which clusters spectral clustering is going to choose, just by looking at the eigenvector plots?

# DemoSpectralClustering (2)

## Exploring Spectral Clustering

- Lowest number of noise dimensions
- Symmetric kNN graph with $k = 10$
- Number of clusters $= 2$

- Play around with data sets Two moons balanced and Three Gaussians (first pick reasonable $\sigma$!)
- Try to understand the plots concerning the eigenvectors and the embedding in $\mathbb{R}^d$
- Increase the number of clusters. Can you predict which clusters spectral clustering is going to choose, just by looking at the eigenvector plots?

Result: Spectral clustering works pretty well ☺

# DemoSpectralClustering (3)

## Low parameters $K$ in the context of spectral clustering

- Data set two gaussians different variance
- Lowest number of noise dimensions
- Mutual kNN graph
- Number of clusters $= 2$

Vary the number of neighbors between 3 and 15. What can you observe? Can you explain the result?

# DemoSpectralClustering (4)

## Low parameters $K$ in the context of spectral clustering

- Data set two gaussians different variance
- Lowest number of noise dimensions
- Mutual kNN graph
- Number of clusters $= 2$

Vary the number of neighbors between 3 and 15. What can you observe? Can you explain the result?

- Many connected components lead to trivial or undesirable results!
- Always choose the connectivity parameter of the graph so that the graph only has one connected component!

# DemoSpectralClustering (5)

## High parameters $K$ in the context of spectral clustering

- Data set two gaussians different variance
- Lowest number of noise dimensions
- Symmetric kNN graph
- Number of clusters $= 2$

Vary the number of neighbors. For which $k$ do the clusters in the embedding look "well separated"? In those cases, does spectral clustering always discover the correct clusters?

# DemoSpectralClustering (6)

High parameters $K$ in the context of spectral clustering

- Data set two gaussians different variance
- Lowest number of noise dimensions
- Symmetric kNN graph
- Number of clusters $= 2$

Vary the number of neighbors from low to high. For which $k$ do the clusters in the embedding look "well separated"? In those cases, does spectral clustering always discover the correct clusters?

High values of $k$ usually don't add useful information (can even be misleading) but increase the complexity. Try to choose rather low values of $K$.

# DemoSpectralClustering (7)

## High number of noise dimensions

- Data set two gaussians balanced
- Noise dimensions 50
- $\sigma = 0.5$
- Mutual kNN graph, $k = 200$

What happens?

# DemoSpectralClustering (8)

## High number of noise dimensions

- Data set two gaussians balanced
- Noise dimensions 50
- $\sigma = 0.5$
- Mutual kNN graph, $k = 200$

What happens?

Even though have one connected component, result is unreliable. Reason: similarity function is not informative, $\sigma$ is too small!

If we pick a better $\sigma$, then spectral clustering works quite well, even in the presence of noise!

# Some selected literature on spectral clustering

Of course I recommend the following ☺

- U.von Luxburg. A tutorial on spectral clustering. Statistics and Computing, to appear. On my homepage.

The three articles which are most cited:

- Meila, M. and Shi, J. (2001). A random walks view of spectral segmentation. AISTATS.
- Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. NIPS 14.
- Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation.IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888 - 905.

Nice historical overview on spectral clustering; and how relaxation can go wrong:

- Spielman, D. and Teng, S. (1996). Spectral partitioning works: planar graphs and finite element meshes. In FOCS, 1996