CR16: Signal Processing and Networks
Data analysis and processing for networks

Part 1 - Data, signal and image processing on graphs using spectral theory

Pierre Borgnat

CR1 CNRS – Laboratoire de Physique, ENS de Lyon, Université de Lyon

Équipe SiSyPhe : Signaux, Systèmes et Physique

09/2014

ENS DE LYON

Overview of the lecture

- General objective: revisit classical data analysis techniques (most used in signal and image processing) in the context of discrete structures such as networks and signals defined on graphs.
- The things we will discuss:
  - Introduce you to the emerging field of graph signal & image processing
  - Basic of spectral analysis of graphs, and on the graph Laplacian
  - Harmonic analysis on graphs: wavelets
  - Examples: denoising on graphs; communities;...
- Organization:
  1. This introduction with several examples
  2. Spectral analysis of the Laplacian; properties
  3. Spectral graph Fourier transform, operators and wavelets
  4. Laplacian pyramid, graph downsampling; applications

Introduction: on signals and graphs

- **My own bias**: I work in the SiSyPhe (Signal, Systems and Physics) group in statistical signal processing, located in the Physics Laboratory of ENS de Lyon
- I have worked also on Internet traffic analysis, and on studies of complex systems
- Strong bias: nonstationary and/or multiscale approaches
- Hence, I will talk about

  data analysis and processing for network
- Examples of topics that we study:

  Technological networks (Internet, mobile phones, sensor networks,...)
  Social networks; Transportation networks (Vélo'v)
  Biosignals: Human brain networks; genomic data; ECG
  ...

Introduction: on signals and graphs

Why data analysis and processing is useful for networks?

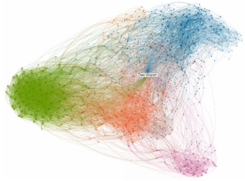
- Many examples of data having both labels or values ("signals") and relational properties (graphs)
- Non-trivial estimation issues (e.g., non repeated measures; variables with large distributions (or power-laws); ...)

  $\rightarrow$ **advanced statistical approaches**
- large networks

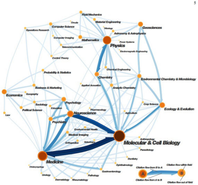
  $\rightarrow$ **multiscale approaches**
- dynamical networks

  $\rightarrow$ **nonstationary methods**

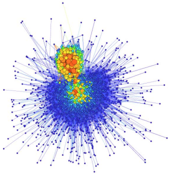
Examples of networks from our digital world



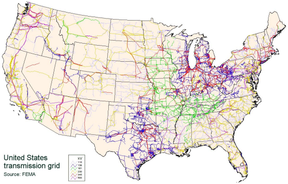
LinkedIn Network



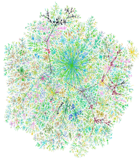
Citation Graph



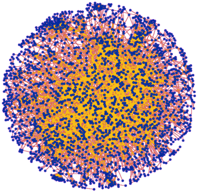
Vehicle Network



USA Power grid

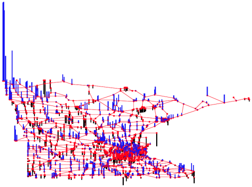


Web Graph

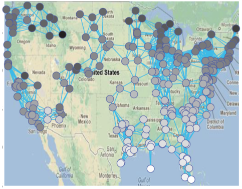


Protein Network

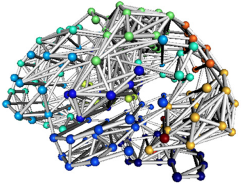
Examples of graph signals



Minnesota Roads



USA Temperature



fcMRI Brain Network

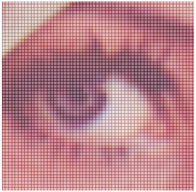


Image Grid

Color Point Cloud

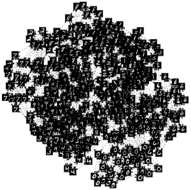
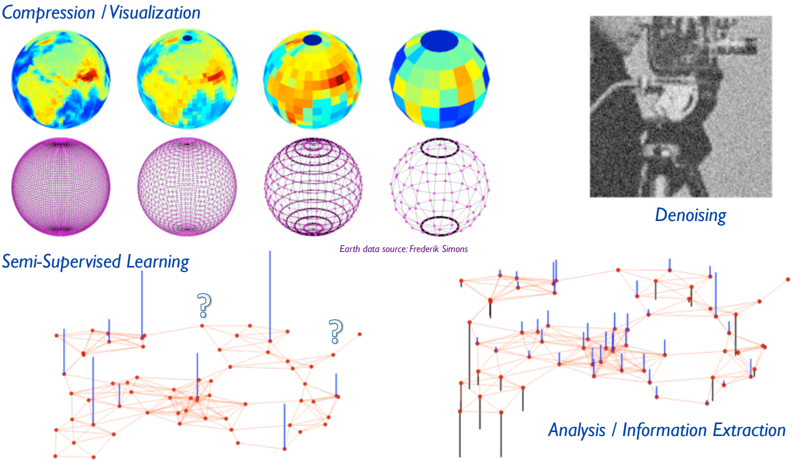


Image Database

Typical problems

[P. Vandergheynst, EPFL, 2013]

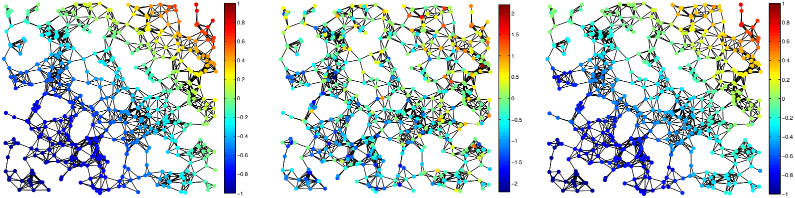


Recovery of signals on graphs

[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Tikhonov regularization

$$\arg\min_{f} ||f - y||_2^2 + \gamma f^\top L f$$



Original Noisy Denoised

Writing Tikhonov denoising as a Graph filter

[P. Vandergheynst, EPFL, 2013]

- It is easy to solve this regularization problem in the spectral domain

$$\arg\min_f \frac{\tau}{2}||f-y||_2^2 + f^\top L f \Rightarrow L f_* + \frac{\tau}{2}(f_* - y) = 0$$

- Move to the spectral domain of the Laplacian

$$\widehat{Lf_*}(i) + \frac{\tau}{2}(\hat{f}_*(i) - \hat{y}(i)) = 0, \quad \forall i \in \{0, 1, ...N-1\}$$

- Solution:

$$\hat{f}_*(i) = \frac{\tau}{\tau + 2\lambda_i}\hat{y}(i)$$

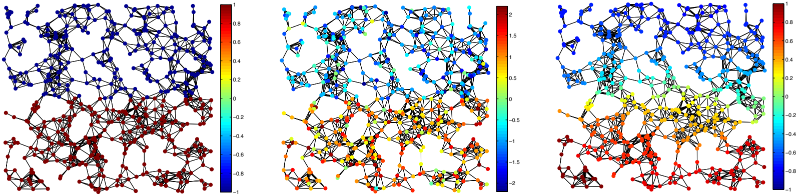
- This is a 1st-order “low pass” filtering

Recovery of signals on graphs

[P. Vandergheynst, EPFL, 2013]

- Limit of Tikhonov regularization

$$\arg\min_{f} ||f - y||_2^2 + \gamma f^\top L f$$



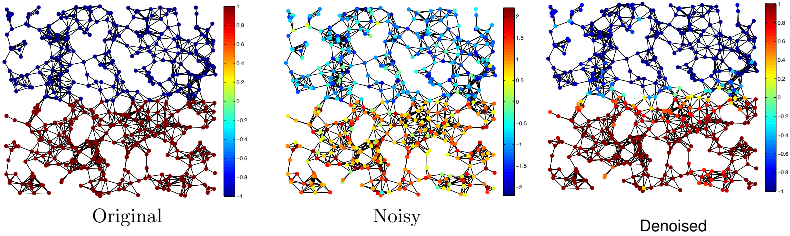
Original Noisy Denoised

Recovery of signals on graphs

[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Wavelet regularization

$$\arg\min_{a} ||W^{\top} a - y||_2^2 + \gamma ||a||_1$$



- Wavelets will be described later on in the lectures... Stay tuned.

Generalized translations

[Shuman, Ricaud, Vandergheynst, 2014]

- Classical translation:

$$(T_\tau g)(t) = g(t-\tau) = \sum_{\mathbb{R}} \hat{g}(\xi) e^{-i2\pi\tau\xi} e^{-i2\pi t\xi} d\xi$$

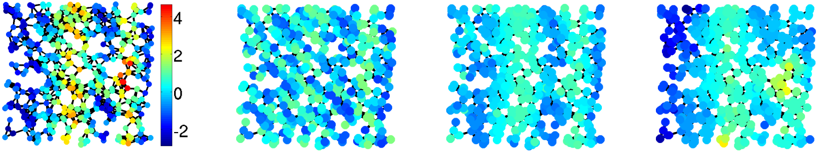
- Graph translations by fundamental analogy:

$$(T_n f)(a) = \sum_{i=0}^{N-1} \hat{f}(i) \chi_i^*(n) \chi_i(a)$$

- Example on the Minnesota road networks

Empirical mode decomposition on graphs

- Objective: decompose a graph signal in various "elementary" modes in a data-driven and non stationary approach



[N. Tremblay, P. Flandrin, P. Borgnat, 2014]

Fourier transform of signals

"Signal processing 101"

The Fourier transform is of paramount importance:
Given a times series $x_n, \; n = 1, 2, ..., N$, let its Discrete Fourier Transform (DFT) be

$$\forall k \in \mathbb{Z} \quad \hat{x}_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$$

Why ?

- Inversion: $x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k e^{-i2\pi kn/N}$
- Best domain to define **Filtering** (operator is diagonal)
- Definition of the **Spectral analysis** (FT of the autocorrelation)
- Alternate representation domains of signals are useful: Fourier domain, DCT, time-frequency representations, wavelets, chirplets,...

Spectral analysis of networks

Spectral theory for network

This is the study of graphs through the **spectral analysis** (eigenvalues, eigenvectors) of matrices **related to the graph**: the adjacency matrix, the Laplacian matrices,....

Notations

$\mathcal{G} = (V, E, w)$	a weighted graph	
$N = \lvert V \rvert$	number of nodes	
A	adjacency matrix	$A_{ij} = w_{ij}$
d	vector of strengths	$d_i = \sum_{j \in V} w_{ij}$
D	matrix of strengths	$D = diag(d)$
f	signal (vector) defined on V	

Relating the Laplacian of graphs to Signal Processing

Laplacian matrix

L or $\mathscr{L}$	laplacian matrix	$L = D - A$
(λ_i)	L's eigenvalues	$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N-1}$
(χ_i)	L's eigenvectors	$L\,\chi_i = \lambda_i\,\chi_i$

A simple example: the straight line

$$\longleftrightarrow \quad L = \begin{pmatrix} \ddots & & & & & & \\ \ldots & -1 & 0 & 0 & 0 & 0 & \\ \ldots & 2 & -1 & 0 & 0 & 0 & \\ & -1 & 2 & -1 & 0 & 0 & \\ & 0 & -1 & 2 & -1 & 0 & \\ & 0 & 0 & -1 & 2 & -1 & \\ & 0 & 0 & 0 & -1 & 2 & \ldots \\ & 0 & 0 & 0 & 0 & -1 & \ldots \\ & & & & & & \ddots \end{pmatrix}$$

For this regular line graph, L is the 1-D classical laplacian operator (i.e. double derivative operator):
its eigenvectors are the Fourier vectors, and its eigenvalues the associated (squared) frequencies

A fundamental analogy

[Shuman et al., *IEEE SP Mag*, 2013]

Objective: Definition of a Fourier Transform adapted to graph signals

f : signal defined on V $\longleftrightarrow$ $\hat{f}$: Fourier transform of f

Fundamental analogy

On *any* graph, the eigenvectors χ_i of the Laplacian matrix L will be considered as the Fourier vectors, and its eigenvalues λ_i the associated (squared) frequencies.

- Works exactly for all regular graphs (+ Beltrami-Laplace)
- Conduct to natural generalizations of signal processing

The graph Fourier transform

- $\hat{f}$ is obtained from f's decomposition on the eigenvectors χ_i :

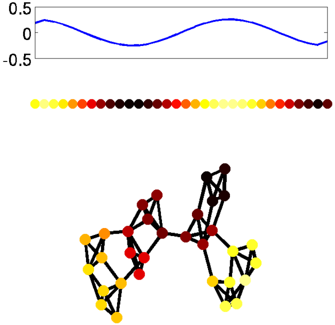
$$\hat{f} = \begin{pmatrix} < \chi_0, f > \\ < \chi_1, f > \\ < \chi_2, f > \\ \dots \\ < \chi_{N-1}, f > \end{pmatrix}$$

Define $\boldsymbol{\chi} = (\chi_0|\chi_1|\dots|\chi_{N-1})$: $\boxed{\hat{f} = \boldsymbol{\chi}^\top f}$

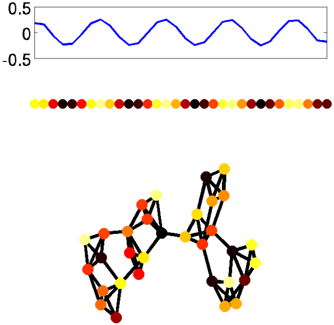
- Reciprocally, the inverse Fourier transform reads: $\boxed{f = \boldsymbol{\chi}\,\hat{f}}$
- The Parseval theorem is valid:
  $\forall (g, h) \quad < g, h > = < \hat{g}, \hat{h} >$

Fourier modes: examples in 1D and in graphs

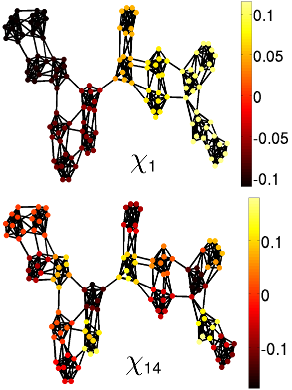
LOW FREQUENCY:



HIGH FREQUENCY:



More Fourier modes



Alternative fundamental spectral correspondance

- With the Normalized Laplacian matrix

$$\mathscr{L} = I - D^{-1/2} A D^{-1/2}$$

  - Related to Ng. et al. normalized spectral clustering
  - Good for degree heterogeneities
  - Related to random walks
  - For community detection

- With the random-walk Laplacian matrix (non symmetrized)

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

  - Better related to random walks
  - Used by Shi-Malik spectral clustering (and graph cuts)

- Using the Adjacency matrix
  - Wigner semi-circular law
  - Discrete Signal Processing in Graphs (good for undirected graphs) [Sandryhaila, Moura, *IEEE TSP*, 2013]

Filtering

Definition of graph filtering

We define a filter function g in the Fourier space.
It is discrete and defined on the eigenvalues $\lambda_i \to g(\lambda_i)$.

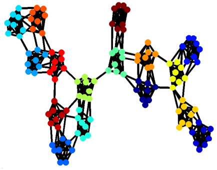
$$\hat{f}^g = \begin{pmatrix} \hat{f}(0)\, g(\lambda_0) \\ \hat{f}(1)\, g(\lambda_1) \\ \hat{f}(2)\, g(\lambda_2) \\ \cdots \\ \hat{f}(N-1)\, g(\lambda_{N-1}) \end{pmatrix} = \hat{\boldsymbol{G}}\, \hat{f} \text{ with } \hat{\boldsymbol{G}} = \begin{pmatrix} g(\lambda_0) & 0 & 0 & \dots & 0 \\ 0 & g(\lambda_1) & 0 & \dots & 0 \\ 0 & 0 & g(\lambda_2) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & g(\lambda_{N-1}) \end{pmatrix}$$

In the node-space, the filtered signal f^g can be written:

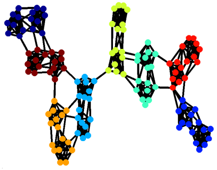
$$f^g = \chi\, \hat{\boldsymbol{G}}\, \chi^\top f$$

For a graph with multiscale structure

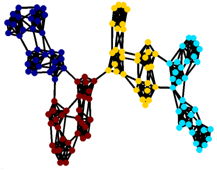
finest scale (16 com.):



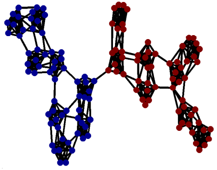
fine scale (8 com.):



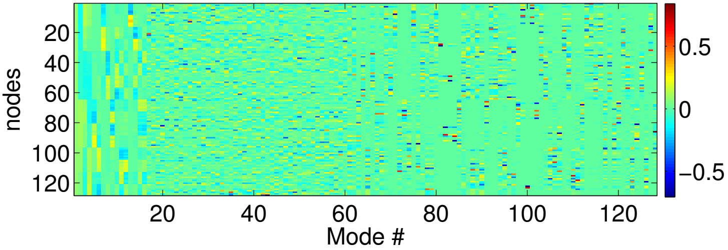
coarser scale (4 com.):

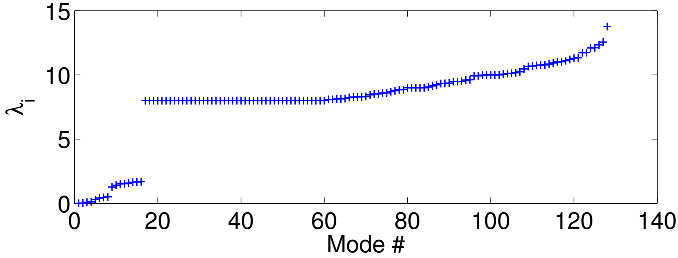


coarsest scale (2 com.):



Spectral analysis: the χ_i and λ_i of a multiscale toy graph





Cuts, clustering and communities

The good, the bad and the ugly

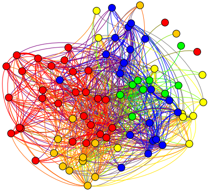
- Networks are often inhomogeneous, with important links, hubs, clusters, or communities (modules)
- These are observed in various types of data on networks: social, technological, biological,...
- Importance of cuts: the min-cut max-flow theorem.
  These are two primal-dual linear programs.
  The max value of a flow = the min capacity over all cuts.
- For clusters and communities, see the extensive surveys:

  [S. Fortunato, *Physic Reports*, 2010]

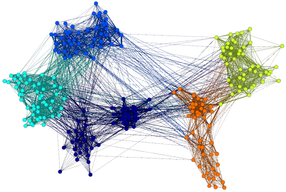
  [von Luxburg, *Statistics and Computating*, 2007]

Some examples of networks with clusters or communities

- Social face-to-face interaction networks

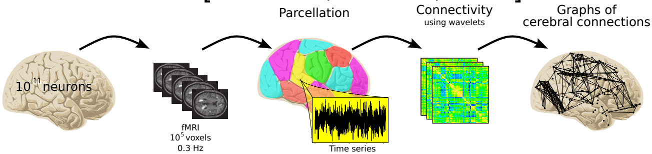


(Lab. physique, ENSL, 2013)



(école primaire, Sociopatterns)

- Brain networks [Bullmore, Achard, 2006]



GRAPHSIP project challenges

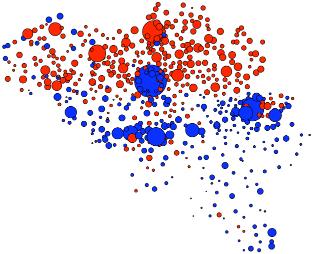
Challenge 1: Robustness and hierarchical analysis of brain connectivity

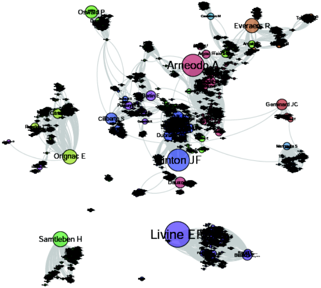
Challenge 2: Brain networks clustering

Challenge 3: Longitudinal study of brain networks

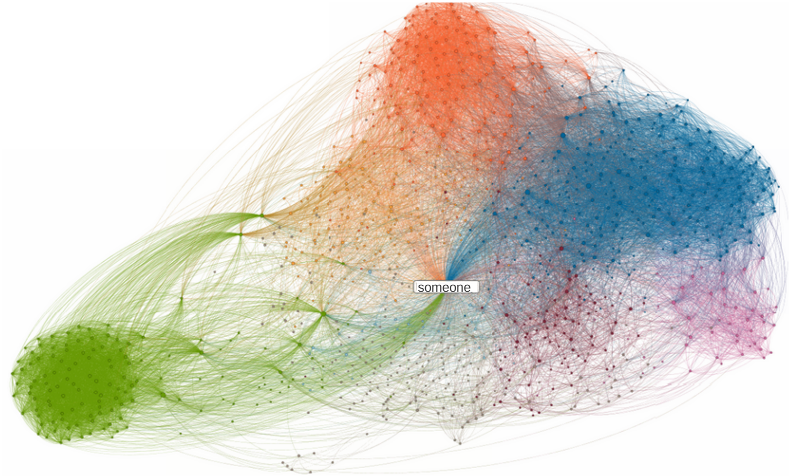
Some examples of networks with clusters or communities

- Mobile phones (The Belgium case, [Blondel et al., 2008])
- Scientometric (co)-citation (or publication) networks [Jensen et al., 2011]

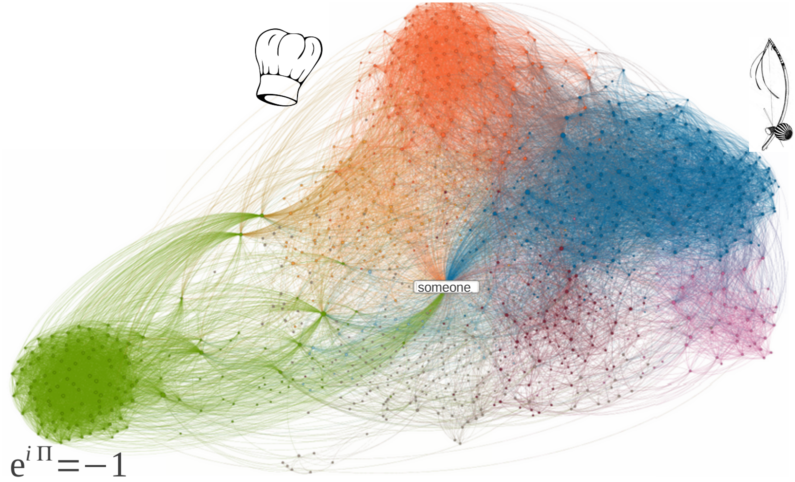




Purpose of community detection?

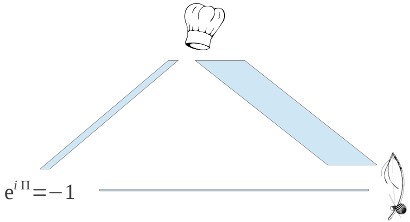


Purpose of community detection?



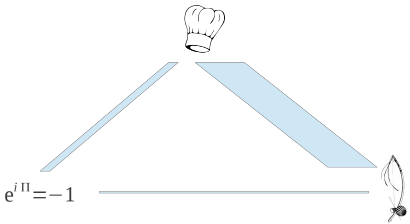
Purpose of community detection?

1) Gives us a sketch of the network:



Purpose of community detection?

1) Gives us a sketch of the network:



2) Gives us intuition about its components:

Methods to find clusters or communities

- I will not pretend to make a full survey... Some important steps are:
- Cut algorithms (classical problem in computer science)
- Spectral clustering (seen as relaxed cut problem)
- Modularity optimization (there arrive the physicists) [Newman, Girvan , 2004]
- Greedy modulatity optimization a la Louvain (computer science strikes back) [Blondel et al., 2008]
- Ideas from information compression (and random walks) [Rosvall, Bergstrom, 2008]

From graph cuts to spectral clustering

- Graph cuts in 2 (or bisection): find the partition in two groups of nodes that minimize the cut size (i.e., the number of links cut)
- Exhaustive search can be computationally challenging
- Also, the cut has to be normalized correctly to find groups of relevant sizes
- One usual metric:
  the Ratio-Cut between sets I and J of nodes

$$R(I,J) = \sum_{i \in I, j \in J} A_{ij}$$

and

$$\mathrm{RatioC}(A,\bar{A}) = \frac{1}{2}\frac{R(A,\bar{A})}{|A|}.$$

Spectral clustering for min-cut

- Spectral interpretation: compute the cut as function of the adjacency matrix A
- We have $R = \frac{1}{2} \sum_{i,j \text{ in} \neq \text{groups}} A_{ij}$.
  This is equal to the cut size between the two groups
- Let us note $s_i = \pm 1$ the assignment of node i to group labelled $+1$ or -1
- $R = \frac{1}{2} \sum_{i,j} A_{ij}(1 - s_i s_j) = \frac{1}{4} \sum_{i,j} L_{ij} s_i s_j = \frac{1}{4} \mathbf{s}^\top L \mathbf{s}$
- Hence, the problem reads as:

$$\min_{\mathbf{s}} \quad \mathbf{s}^\top L \mathbf{s}$$

Spectral clustering for min-cut

- Let us assume the spectral decomposition of the Laplacian (to be seen later on):

$$L_{ij} = \sum_{k=1}^{N-1} \lambda_k (\chi_k)_i (\chi_k)_j$$

- The optimal assignment vector (that minimizes R) would be $s_i = (\chi_1)_i \ldots$ if there were no constraints on the s_i's... Note: χ_1 is known as the the Fiedler vector.
- However, $s_i = +1$ or -1...
- Approximated solution: $s_i = \mathrm{sign}((\chi_1)_\mathrm{i})$.
- The estimated groups are still close to χ_1.

Spectral clustering for min-RatioCut

- Normalization by the size of the sets: $\min_{A \subset V} \text{RatioC}(A, \bar{A})$
- The same problem written in a relaxed form introducing:

$$\mathbf{f}(i) = +\sqrt{\frac{|\bar{A}|}{|A|}} \text{ if } i \in A \text{ and } \mathbf{f}(i) = -\sqrt{\frac{|A|}{|\bar{A}|}} \text{ if } i \in \bar{A}$$

  Then, $||\mathbf{f}|| = \sqrt{|V|}$ and $\mathbf{f}^\top \mathbf{1} = 0$.
- Finally, one has

$$\mathbf{f}^\top L\, \mathbf{f} = |V| \cdot \text{RatioC}(A, \bar{A}).$$

- Hence, problem with relaxed constraints:

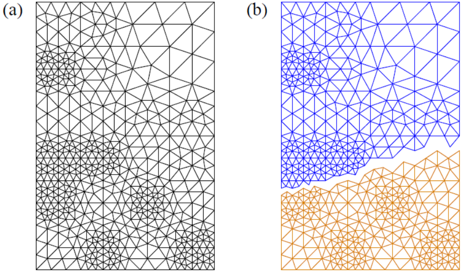
$$\begin{aligned} &\min_{\mathbf{f}} \ \ \mathbf{f}^\top L\, \mathbf{f} \\ &\text{such that } \mathbf{f}^\top \mathbf{1} = 0, \ ||\mathbf{f}||_2 = \sqrt{|\mathrm{V}|} \end{aligned}$$

- This allows also for *Spectral clustering of data* represented by networks

cf. [von Luxburg, *Statistics and Computating*, 2007]

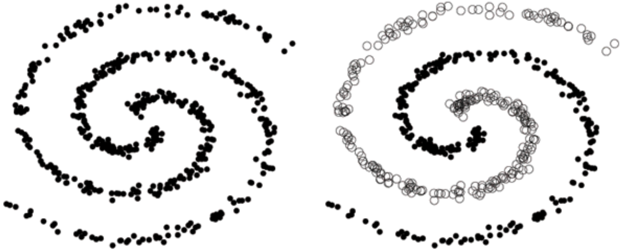
Spectral clustering

- Example of spectral bisection on an irregular mesh



Spectral clustering

- Example of spectral bisection on data irregularly spread in a space



- It's good, very good in fact for clustering
- However, not really good for natural modules / communities

Quality of a partition: the Modularity

- Problems with spectral clustering:
  1) No assessment of the quality of the partitions
  2) No reference to comparison to some null hypothesis (or "mean field") situation
- Improvement: the modularity [Newman, 2003]

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(c_i, c_j)$$

  where $2m = \sum_i d_i$.
- Q is between -1 and $+1$ (in fact, lower than $1 - 1/n_c$ if n_c groups)

Quality of a partition: the Modularity

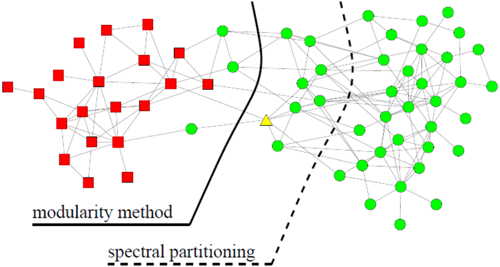
- Interpretation: $\frac{d_i d_j}{2m}$ is, for a null model as a Bernoulli random graph (with prob. $2m/N/(N-1)$ of existence of each edge), the fraction of edges expected between nodes i and j.
  (Note: in fact, it's best seen as a Chung-Lu model (2002))
- Re-written in term of groups, it gives

$$Q = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right]$$

  where l_c is the number of edges in group c and d_c is the sum of degrees of nodes in group c.
- Consequence: the larger Q is, the most pronounced the communities are
- Algebraic form: modularity matrix $B = \frac{A}{2m} - \frac{dd^\top}{(2m)^2}$ and $Q = Tr(\mathbf{c}^\top B \mathbf{c})$ for a partition matrix $\mathbf{c}$ (size $n_c \times N$) of the nodes.

Community detection with modularity

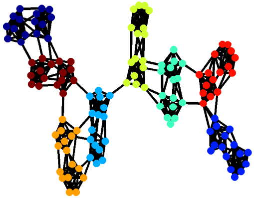
- By optimization of Q
- For instance: by simulated annealing, by spectral approaches,...
- Works well, better than spectral clustering.



- Better algorithm: the greedy (ascending) Louvain approach (ok for millions of nodes !) [Blondel et al., 2008]

Multiscale community structure in a graph

Classical community detection algorithms do not have this "scale-vision" of a graph. Modularity optimisation finds:

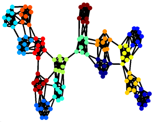


Where the modularity function reads:

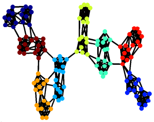
$$Q = \frac{1}{2N}\sum_{ij}\left[A_{ij} - \frac{d_i d_j}{2N}\right]\delta(c_i, c_j)$$

Multiscale community structure in a graph

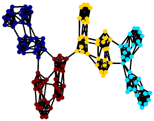
Q=0.80 :



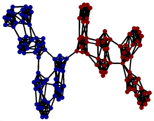
Q=0.83 :



Q=0.74 :



Q=0.50 :



All representations are correct but
modularity optimisation favours one.

- Added to that: limit in resolution for modularity [Fortunato, Barthelemy, 2007]

Integrate a scale into modularity

- [Arenas et al., 2008] Remplace A by $A + rI$ in Q. Self-loops.
- [Reichardt and Bornholdt, 2006]

$$Q_\gamma = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \gamma \frac{d_i d_j}{2m} \right] \delta(c_i, c_j)$$

- Equivalent for regular graph if $\gamma = 1 + \frac{r}{\bar{d}}$.
- "Corrected Arenas modularity": use $A_{ij} + r\frac{d_i}{\bar{d}}\delta_{ij}$; equivalent to Reichardt and Bornholdt [Lambiotte, 2010]

Some works on multiscale communities

- Lambiotte, ”Multiscale modularity in complex networks“ [*WiOpt*, 2010]
- Schaub, Delvenne et al., ”Markov dynamics as a zooming lens for multiscale community detection: non clique-like communities and the field-of-view limit” [*PloS One*, 2012]
- Arenas et al., ”Analysis of the structure of complex networks at different resolution levels” [*New Journal of Physics*, 2008]
- Reichardt and Bornholdt, ”Statistical Mechanics of Community Detection” [*Physical Review E*, 2006]
- Mucha et al., ”Community Structure in Time-Dependent, Multiscale, and Multiplex Networks” [*Science*, 2010]
- Tremblay, Borgnat, “Graph Wavelets for Multiscale Community Mining” [*IEEE TSP*, 2014]

More on that later in the next part of the lecture

Spectral clustering

- More general spectral clustering: Use all (or some) of the eigenvectors χ_i of L
- Then, use a classical K-means on the first K non-null eigenvectors of L (each node a has the $(\chi_k)_a$ avec features)
- If large heterogeneity of degrees: the normalized Laplacian gives better results

Normalized Laplacian matrix

$\mathscr{L}$	Laplacian matrix	$\mathscr{L} = I - D^{-1/2} A D^{-1/2}$
(λ_i)	$\mathscr{L}$'s eigenvalues	$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{N-1}$
(χ_i)	$\mathscr{L}$'s eigenvectors	$\mathscr{L}\,\chi_i = \lambda_i\,\chi_i$

- Choice of K by eigengaps $|\lambda_{k+1} - \lambda_k|$

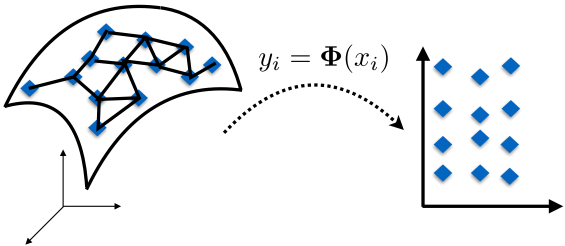
Graph embedding, Laplacian maps

- Spectral clustering := embedding + K-means

$$\forall a \in V : a \to (\chi_1(a), \chi_2(a), ..., \chi_k(a)) \in \mathbb{R}^k$$

- Objective of embedding: embed vertices in low dimensional space, so as to discover geometry

$$x_i \in \mathbb{R}^d \to y_i \in \mathbb{R}^k \quad \text{with } k < d$$



Graph embedding, Laplacian maps

- A good embedding preserves locality in the embedding space, so that nearby points are mapped nearby.
  It preserves smoothness.
- For that, minimize the variations of the embedding:

$$\sum_{i,j} A_{ij}(y_i - y_j)^2$$

- Laplacian eigenmaps:

$$\begin{aligned} \operatorname{argmin}\ & \mathbf{y}^\top L\,\mathbf{y} \\ \text{such that } & \mathbf{y}^\top A\,\mathbf{y} = 1 \\ \text{and }\ & \mathbf{y}^\top L\,\mathbf{1} = 0 \end{aligned}$$

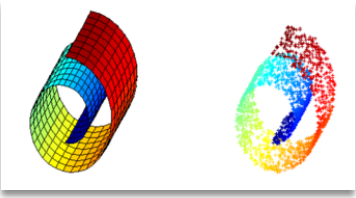
  Alternative formulation:

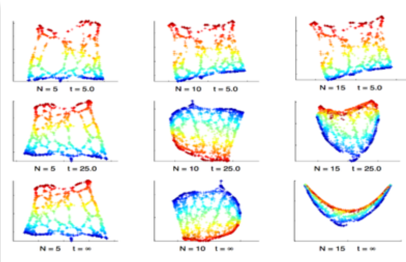
$$L\mathbf{y} = \lambda A\mathbf{y}$$

  (generalized eigenproblem)

Graph embedding, Laplacian maps

- Some examples





[Belkin, Niyogi, 2003]

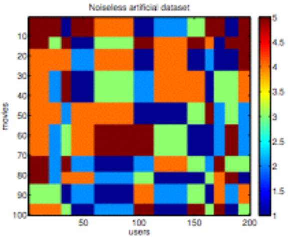
A recommander system

[Vandergheynst et al., EPFL, 2014]

- Assume data in the form M[movie, user] $=$ movie rating

$$\mathcal{L}_c$$

$$\mathcal{L}_r$$



- One observes only a subset of M. How to complete it?
- Hypotheses:
  - Users structured as communities,
    and users in community rate similarly
  - Movies are clustered in genres,
    and similar movies are rated similarly by users

A recommander system

[Vandergheynst et al., EPFL, 2014]

- Let us write $A_\Omega(M)$ the observed part of the matrix M
- Matrix completion problem:

$$\min_X \operatorname{rank}(X) \;\; s.t. \;\; A_\Omega(X) = A_\Omega(M).$$

- Problem relaxed with the nuclear norm:
  $||X||_* = Tr((XX^\top)^{1/2}) = \sum_k \sigma_k$
  (where the σ_k are the singular values of $X = U\Sigma V^\top$)
- Tolerance to noise: change $A_\Omega(X) = A_\Omega(M)$ into a penalty term $||A_\Omega \circ (A - M)||$
- Completion by smoothness on the two graphs (users and movies), as quantified by a term

$$\gamma XLX^\top = \gamma \sum_{j,j'} Aij||x_j - x_{j'}||^2.$$

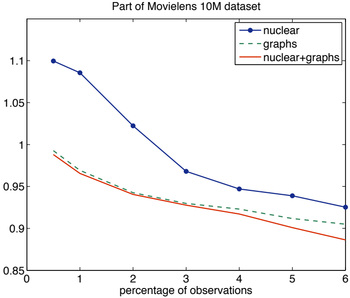
A recommander system

[Vandergheynst et al., EPFL, 2014]

- Hence an optimization problem to solve

$$\min_X \gamma_n ||X||_* + ||A_\Omega \circ (A - M)|| + \gamma_r X L_r X^\top + \gamma_c X L_c X^\top$$

- Solution by advanced optimization tools for convex non-smooth functions.
- Here, the ADMM approach (Alternating Direction Method of Multipliers) works well (see other parts of the lectures)



A small pause

- This was an invitation to "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains"
  See [Shuman, Narang, Frossard, Ortega, Vandergheynst, *IEEE SP Mag*, 2013]
- Next on our program:
  - Spectral analysis of the Laplacian; some properties
  - Spectral graph Fourier transform, operators and wavelets (hence a notion of scales)
  - Laplacian pyramid, graph downsampling - Applications

http://perso.ens-lyon.fr/pierre.borgnat

Acknowledgements: thanks to Renaud Lambiotte, Pierre Vandergheynst and Nicolas Tremblay for borrowing some of their figures or slides.