Multiscale Processing on Networks and Community Mining

Part 1 - Communities in networks
Graph Signal Processing

Pierre Borgnat

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

Équipe SISYPHE : Signaux, Systèmes et Physique

05/2014

Overview of the lecture

- General objective: revisit the classical question of finding communities in networks using multiscale processing methods on graphs.
- The things we will discuss:
  - Recall the notion of community in networks and brief survey of some aspects of community detection
  - Introduce you to the emerging field of graph signal processing
  - Show a connexion between the two: detection of communities with graph signal processing
- Organization:
  1. A (short) lecture about communities in networks
  2. Signal processing on networks; Spectral graph wavelets
  3. Multiscale community mining with wavelets

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 also, and studied some complex systems
- Strong bias: nonstationary and/or multiscale approaches
- You will then hear about

  signal processing for network science
- Examples of topics that we study:

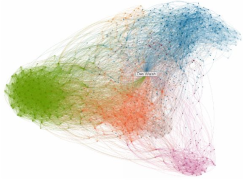
  Technological networks (Internet, mobile phones, sensor networks,...)
  Social networks; Transportation networks (Vélo'v)
  Biosignals: Human bran networks; genomic data; ECG
  ...

Introduction: on signals and graphs

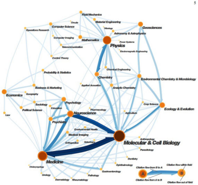
Why signal processing might be useful for network science ?

- 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**

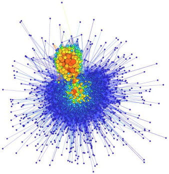
Examples of networks from our digital world



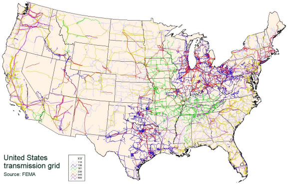
LinkedIn Network



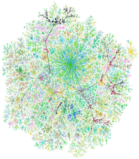
Citation Graph



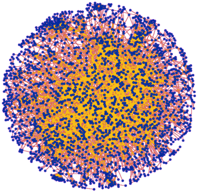
Vehicle Network



USA Power grid

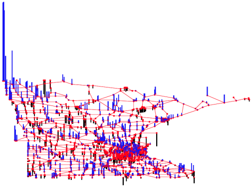


Web Graph

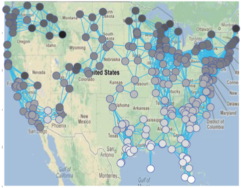


Protein Network

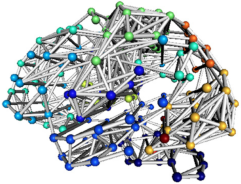
Examples of graph signals



Minnesota Roads



USA Temperature



fcMRI Brain Network

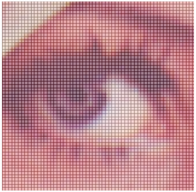


Image Grid

Color Point Cloud

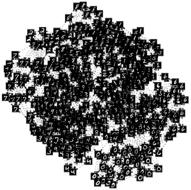


Image Database

Communities in networks

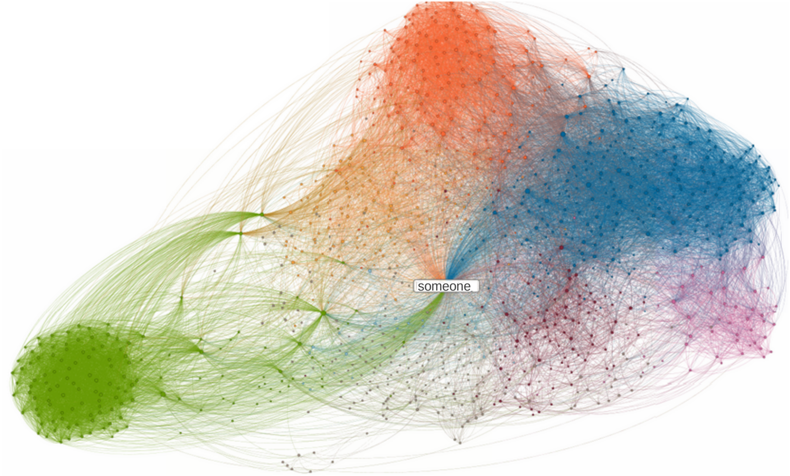
- Networks are often inhomogeneous, made of communities (or modules):
  groups of nodes having a larger proportion of links inside the group than with the outside
- This is observed in various types of networks: social, technological, biological,...
- There exist several extensive surveys:

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

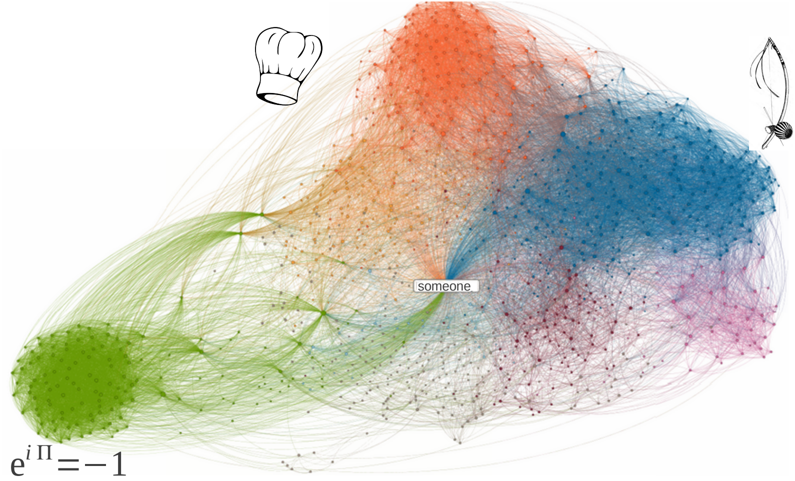
  [von Luxburg, *Statistics and Computating*, 2007]

  ...

Purpose of community detection?

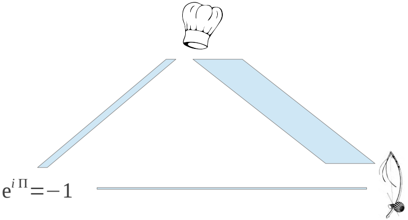


Purpose of community detection?



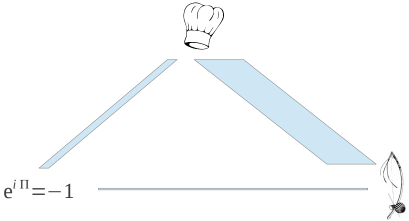
Purpose of community detection?

1) Gives us a sketch of the network:



Purpose of community detection?

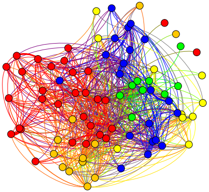
1) Gives us a sketch of the network:



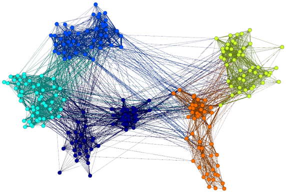
2) Gives us intuition about its components:

Some examples of networks with communities or modules

- Social face-to-face interaction networks

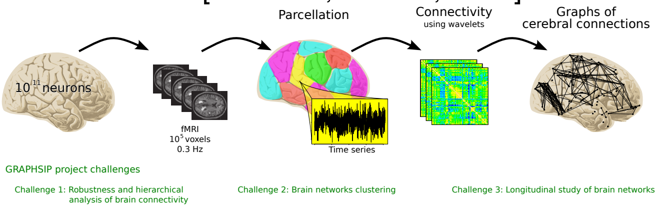


(Lab. physique, ENSL, 2013)



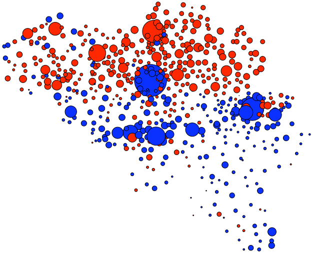
(école primaire, Sociopatterns)

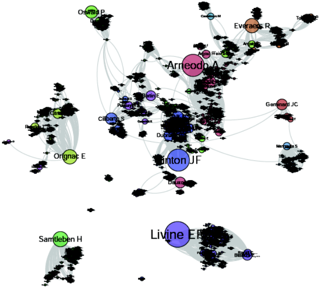
- Brain networks [Bullmore, Achard, 2006]



Some examples of networks with communities or modules

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





Methods to find communities

- I will not pretend to make a full survey... Some important steps are:
- Cut algorithms (legacy from computer science)
- Spectral clustering (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 [Rosvall, Bergstrom, 2008]

From graph bisection to spectral clustering

- Graph bisection (or cuts): find the partition in two (or more) groups of nodes that minimize the cut size (i.e., the number of links cut)
- Exhaustive search can be computationally challenging
- Also, the cut is not normalized correctly to find groups of relevant sizes
- Spectral interpretation: compute the cut as function of the adjacency matrix A

Wait... What means **spectral** for networks ?

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	
$N = \|V\|$	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	

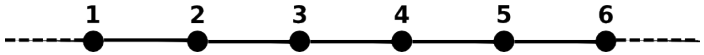
Definition of the Laplacian matrix of graphs

Laplacian matrix

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	$L\,\chi_i = \lambda_i\,\chi_i$

Note: $\chi_0 = \mathbf{1}$.

A simple example: the straight line



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

For this regular line graph, L is the 1-D classical laplacian operator (i.e. double derivative operator).

Going back to spectral clustering

- Let $R = \frac{1}{2} \sum_{i,j \text{ in} \neq \text{groups}} A_{ij}$.
  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
- $R = \frac{1}{2} \sum_{i,j} A_{ij}(1 - s_i s_j) = \frac{1}{4} \sum_{i,j} L_{ij} s_i s_j = \frac{1}{4} \mathbf{s}^\top L \mathbf{s}$
- Spectral decomposition of the Laplacian:

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

- 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...
- However, $s_i = +1$ or -1.

Spectral clustering

- Problem with relaxed constraints:

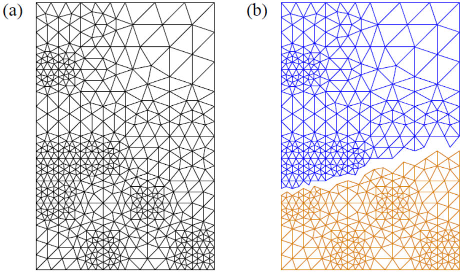
$$\min_{\mathbf{s}} \quad \mathbf{s}^\top L \mathbf{s}$$
$$\text{such that } \mathbf{s}^\top \mathbf{1} = 0, \ ||\mathbf{s}||_2 = \sqrt{\mathrm{N}}$$

- Simplest solution of this spectral bisection: $s_i = \mathrm{sign}((\chi_1)_\mathrm{i})$
- This estimates communities that are close to χ_1 (known as the the Fiedler vector)
- 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



- Not really good for natural modules / communities

Spectral clustering

- More general spectral clustering: Use all (or some) of the eigenvectors χ_i of L
- For instance: use a classical K-means on the first K non-null eigenvectors of L
  (each node a having 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	$\mathscr{L}\, \chi_i = \lambda_i \, \chi_i$

Interpretation as random walks (part 1)

- A random walk on a graph can be described by means on the adjacency operator. In particular, the occupancy probability $\mathbf{p}(t)$ at time t evolves like:

$$\mathbf{p}(t) = AD^{-1}\mathbf{p}(t-1) = W\mathbf{p}(t-1)$$

- Transition matrix W has a symmetrized version

$$S = D^{-1/2}AD^{1/2}$$

  which has same eigenvalues
- Many properties of random walks derives from the normalized Laplacian (symmetric or not)

Interpretation as random walks (part 1)

- Example 1: lazy random walk (which stays in place with prob. 1/2) converges to equilibrium π in

$$||\mathbf{p}_a(t) - \pi(a)||_2 \leq \sqrt{\frac{d(a)}{\min_u d(u)}}\,(1 - \lambda_{N-1}(W))^t$$

and $1 - \lambda_{N-1}(W) = \lambda_1(\mathscr{L})$.

- Example 2: relation to normalized cuts

$$\lambda_1(\mathscr{L}) = \min_{\mathbf{s},\, d^\top \mathbf{s}=0} \frac{\mathbf{s}^\top L\mathbf{s}}{\mathbf{s}^\top D\mathbf{s}}$$

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

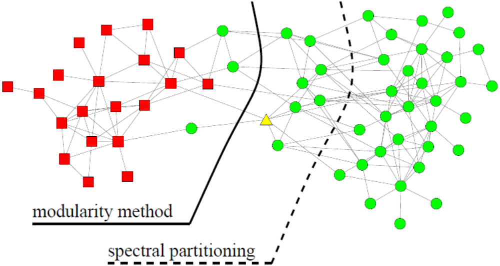
- Interpretation: $\frac{d_i d_j}{2m}$ is, for a null model as a Bernoulli random graph (with prob. $2m/N/(N-1)$ of existence of each edge), the fraction of edges expected between nodes i and j.
  (Note: in fact, it's best seen as a Chung-Lu model (2002))
- Re-written in term of groups, it gives

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

  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

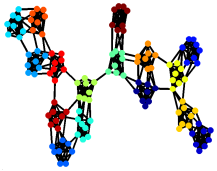
- By optimization of Q
- For instance: by simulated annealing, by spectral approaches,...
- Works well, better than spectral clustering.



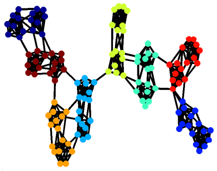
- Better algorithm: the greedy (ascending) Louvain approach (ok for millions of nodes !) [Blondel et al., 2008]

Existence of multiscale community structure in a graph

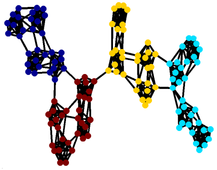
finest scale (16 com.):



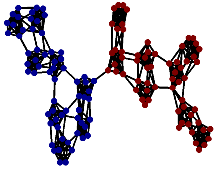
fine scale (8 com.):



coarser scale (4 com.):

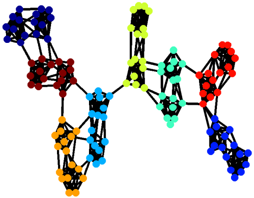


coarsest scale (2 com.):



Multiscale community structure in a graph

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

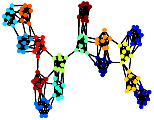


Where the modularity function reads:

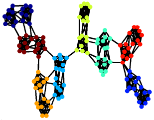
$$Q = \frac{1}{2N} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2N} \right] \delta(c_i, c_j)$$

Multiscale community structure in a graph

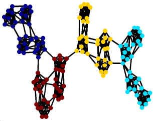
Q=0.80 :



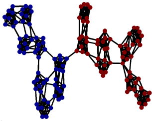
Q=0.83 :



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]

Interpretation as random walks (part 2)

- Let us recall that $\mathbf{p}(t) = AD^{-1}\mathbf{p}(t-1) = W\mathbf{p}(t-1)$
- Equilibrium distribution: $\pi_i = \dfrac{d_i}{2m}$
- One step of random walk; the probability of staying in the same community is

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

- Random walk after t steps (even if t continuous)

$$R(t) = \sum_{ij} \left[\left(e^{t(D^{-1}A-I)} \right)_{ij} \frac{d_j}{2m} - \frac{d_i d_j}{(2m)^2} \right] \frac{d_i d_j}{(2m)^2}$$

This is called **stability**.

Interpretation as random walks (part 2)

- If $t = 0$, $R(0) = 1 - \sum_{ij} \frac{d_i d_j}{(2m)^2} \frac{d_i d_j}{(2m)^2}$;
  best partition = single nodes
- If t small, $R(t) \simeq (1-t)R(0) + tQ_c$;
  trade-off between single nodes and modularity; falls down in the Reichardt and Bornholdt formulation
- If $t = 1$, classical modularity
- If t large, the optimum partition is in 2 groups, as given by spectral clustering (Fiedler vector)

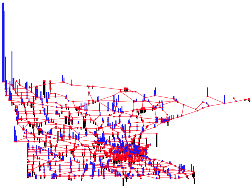
- In practice, optimization with same methods as for modularity
- It works well

Referenced works on multiscale communities

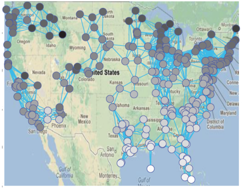
- Lambiotte, ”Multiscale modularity in complex networks“ [*WiOpt*, 2010]
- Schaub, Delvenne et al., ”Markov dynamics as a zooming lens for multiscale community detection: non clique-like communities and the field-of-view limit” [*PloS One*, 2012]
- Arenas et al., ”Analysis of the structure of complex 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]

More on that later in the next part of the lecture

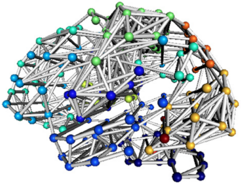
Examples of graph signals



Minnesota Roads



USA Temperature



fcMRI Brain Network

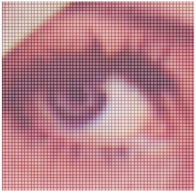


Image Grid

Color Point Cloud

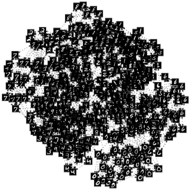


Image Database

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 ?

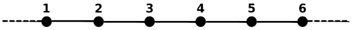
- Inversion: $x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k e^{-i2\pi kn/N}$
- Best domain to define **Filtering** (operator is diagonal)
- Definition of the **Spectal analysis** (FT of the autocorrelation)
- Alternate representation domains of signals are useful: Fourier domain, DCT, time-frequency representations,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

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



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

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

Objective and 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 :

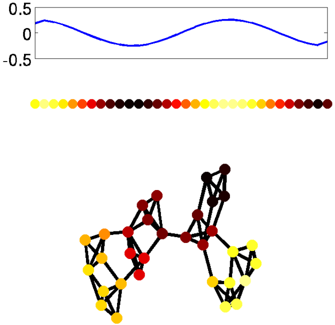
$$\hat{f} = \begin{pmatrix} <\chi_0, f> \\ <\chi_1, f> \\ <\chi_2, f> \\ ... \\ <\chi_{N-1}, f> \end{pmatrix}$$

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

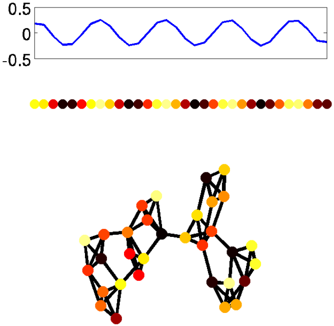
- Reciprocally, the inverse Fourier transform reads: $\boxed{f = \boldsymbol{\chi} \hat{f}}$
- The Parseval theorem is valid:
  $\forall (g, h) \quad < g, h > = < \hat{g}, \hat{h} >$

Fourier modes: examples in 1D and in graphs

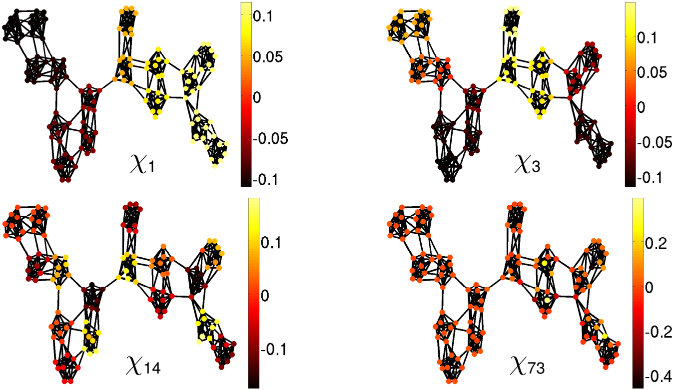
LOW FREQUENCY:



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

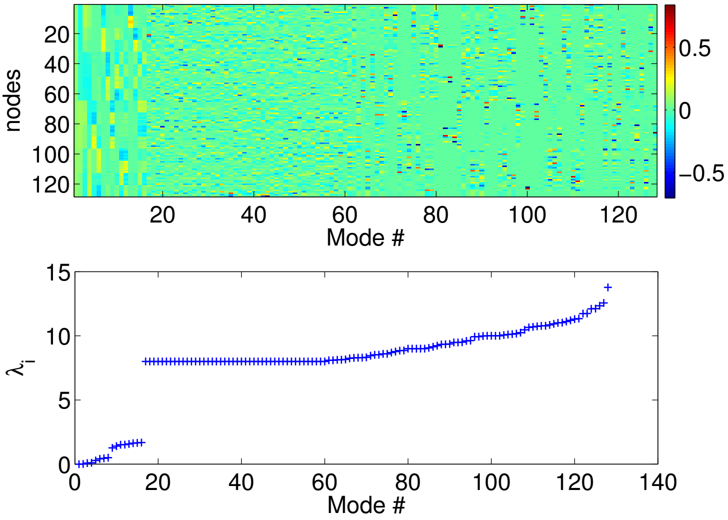
We define a filter function g in the Fourier space.
It is discrete and defined on the eigenvalues $\lambda_i \rightarrow g(\lambda_i)$.

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

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

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

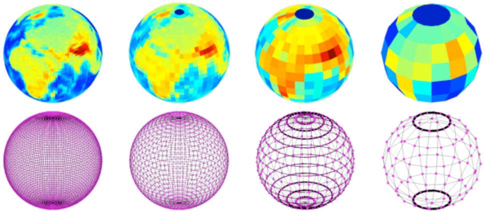
Spectral analysis: the χ_i and λ_i of a multiscale toy graph



Typical problems for graph signal processing

[P. Vandergheynst, EPFL, 2013]

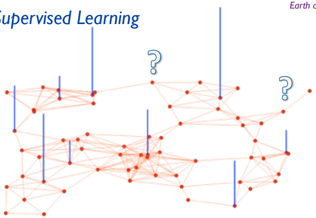
Compression / Visualization

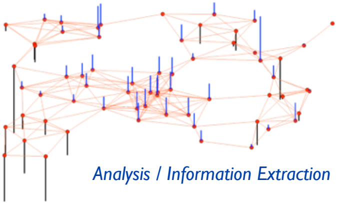


Earth data source: Frederik Simons

Denoising

Semi-Supervised Learning





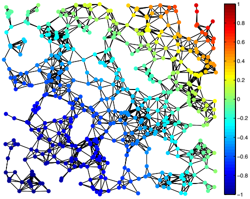
Analysis / Information Extraction

Recovery of signals on graphs

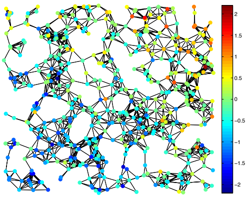
[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Tikhonov regularization

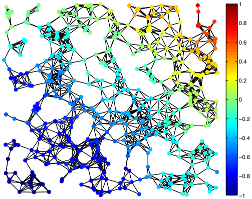
$$\arg\min_{f} ||f - y||_2^2 + \gamma f^\top L f$$



Original



Noisy



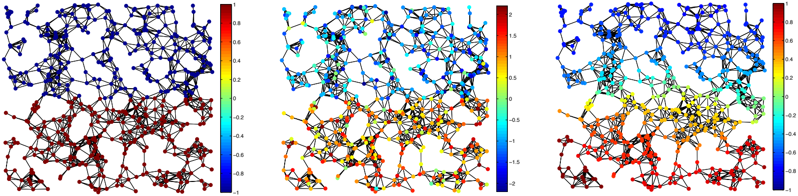
Denoised

Recovery of signals on graphs

[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Tikhonov regularization

$$\arg\min_f ||f - y||_2^2 + \gamma f^\top L f$$



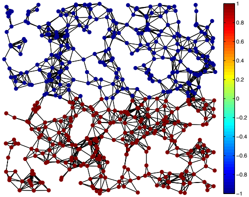
Original — Noisy — Denoised

Recovery of signals on graphs

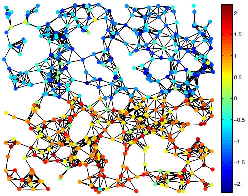
[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Wavelet regularization

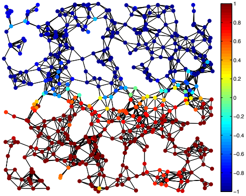
$$\arg\min_{a} ||W^{\top} a - y||_2^2 + \gamma ||a||_1$$



Original



Noisy



Denoised

- Wavelets will be described soon... Stay tuned.

Writing Tikhonov denoising as a Graph filter

[P. Vandergheynst, EPFL, 2013]

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

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

- In the graph Fourier domain

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

- Solution:

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

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

Generalized translations

[Shuman, Ricaud, Vandergheynst, 2014]

- Classical translation:

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

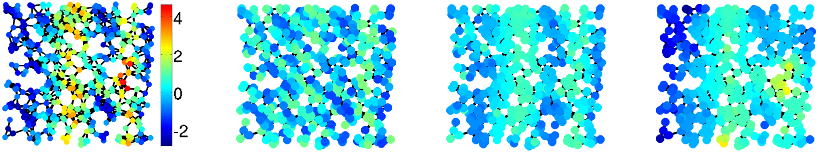
- Graph translations by fundamental analogy:

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

- Example on the Minnesota road networks

Empirical mode decomposition on graphs

- Objective: decompose a graph signal in various “elementary” modes in a data-driven approach



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

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]
- Now, we still have on our program:
  - The wavelet transform on graphs (hence a notion of scaling)
  - Make a connexion with community detection

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.