Introduction to signal processing on graphs

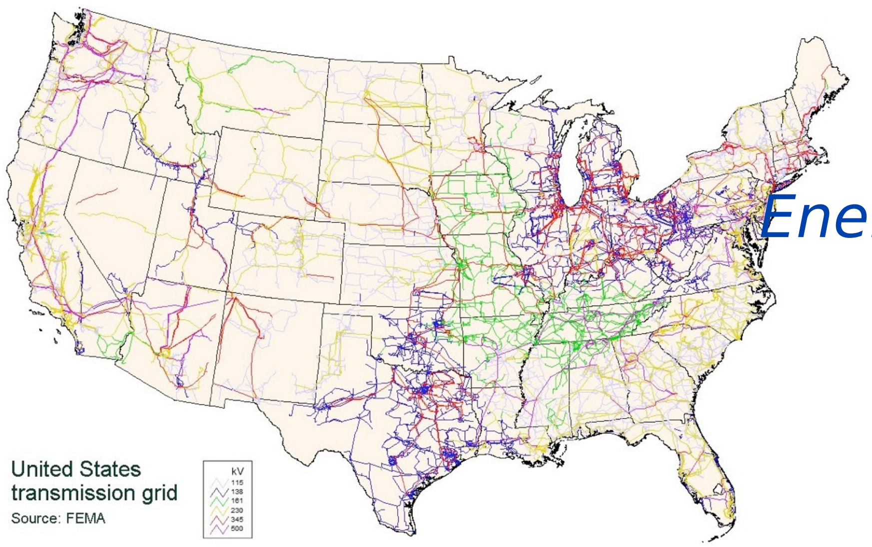
References:

First few chapters of

"Spectral graph theory", Fan Chung

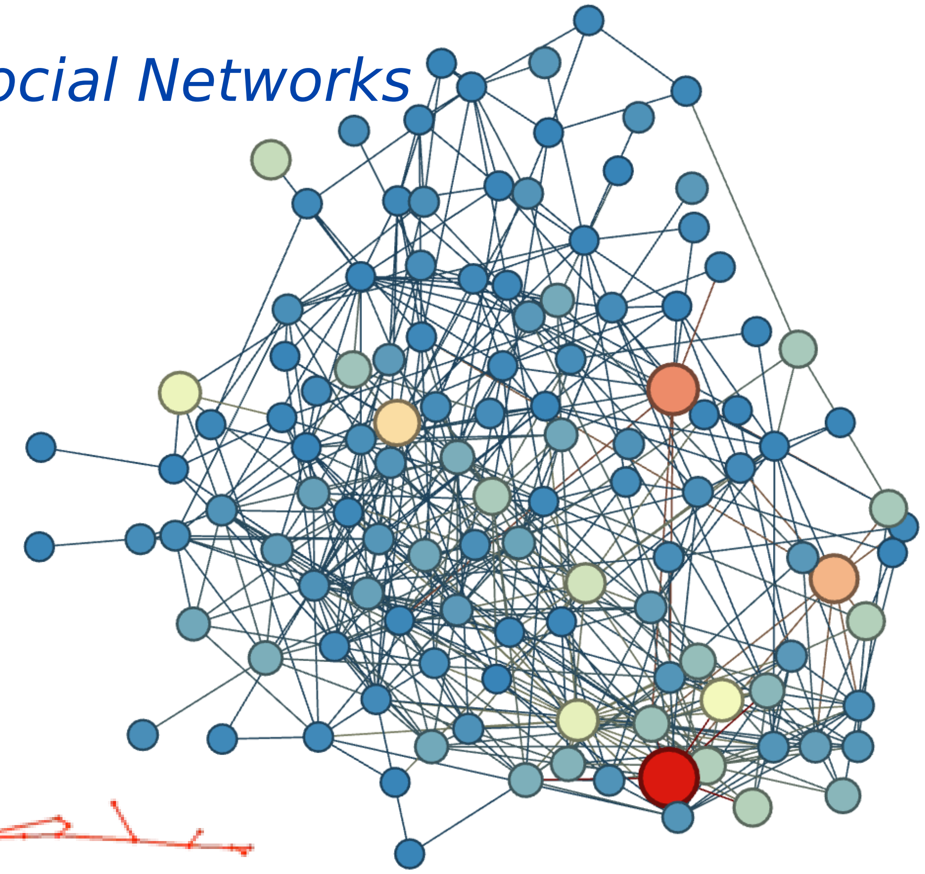
LTS EPFL
LTS2
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

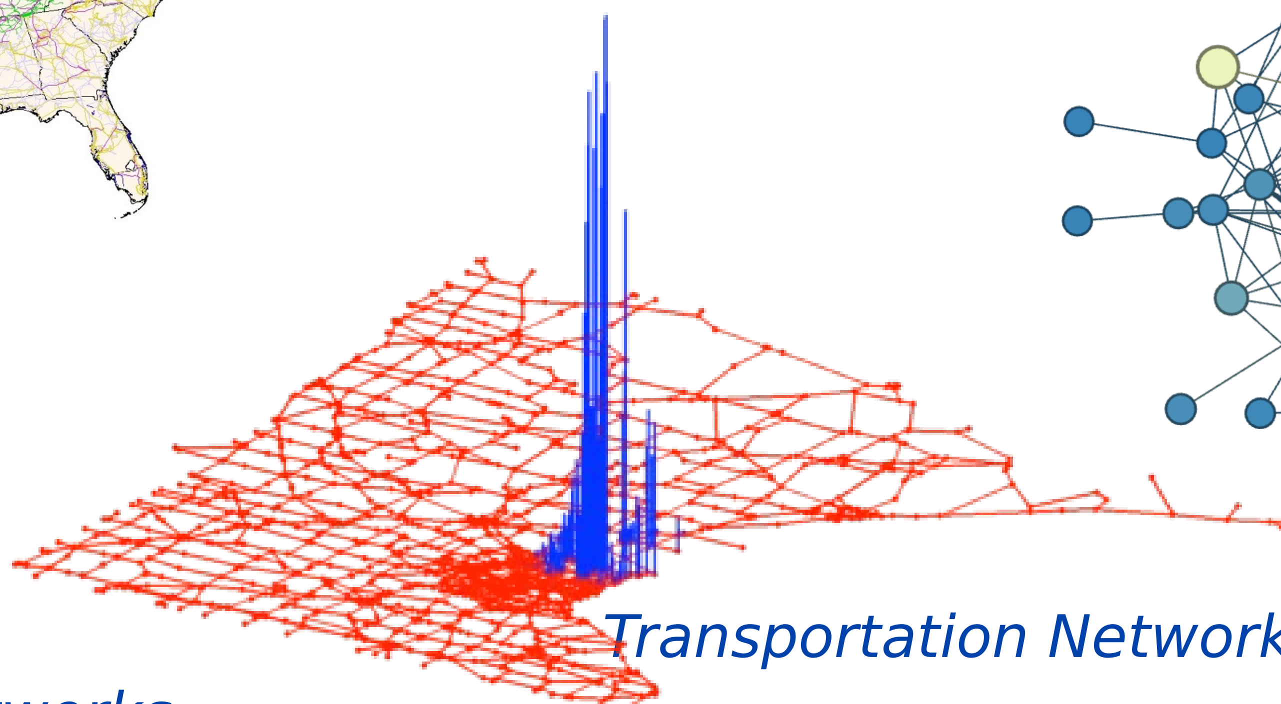
Processing Data on/with Graphs



Energy Networks

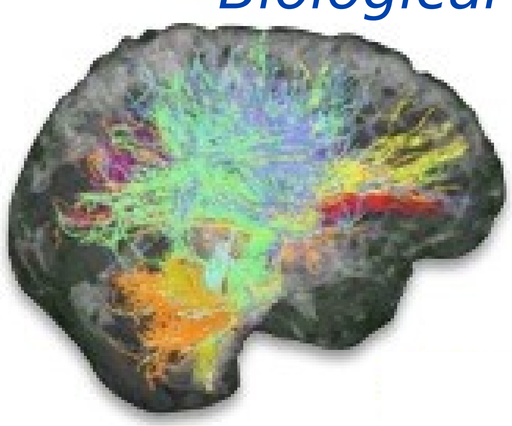
Social Networks

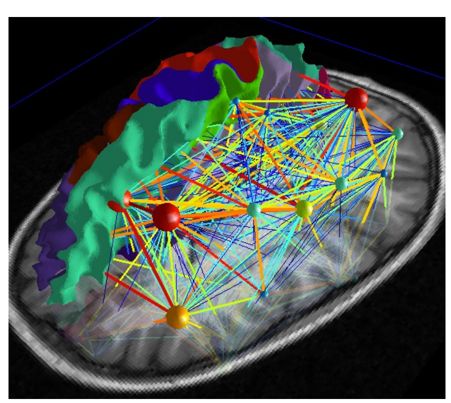


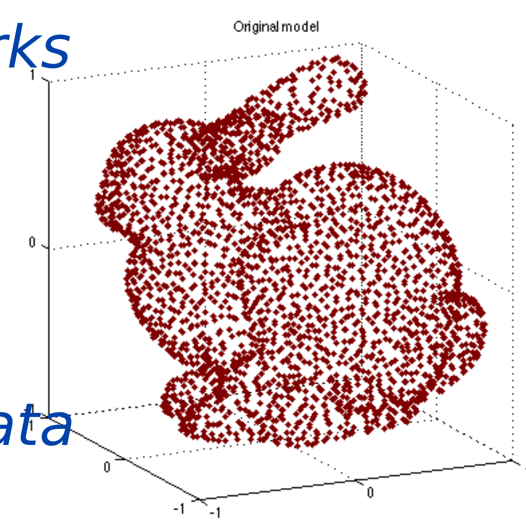


Transportation Networks

Biological Networks





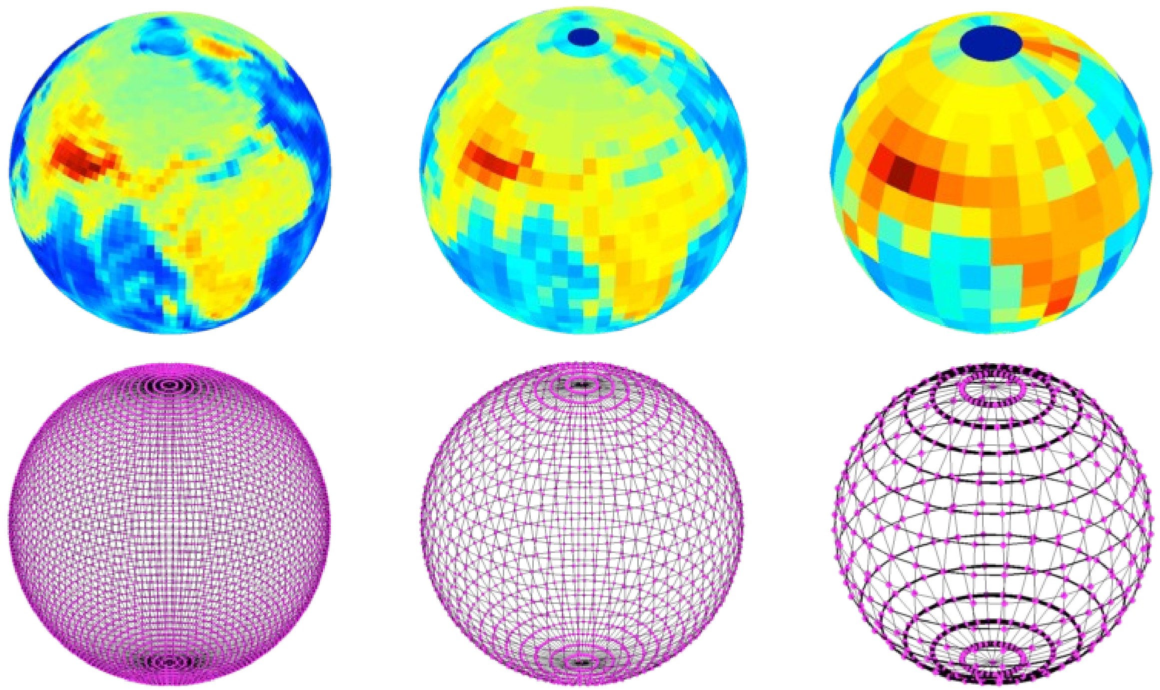


Irregular Data Domains

Encode neighbors relationship, locality, affinity

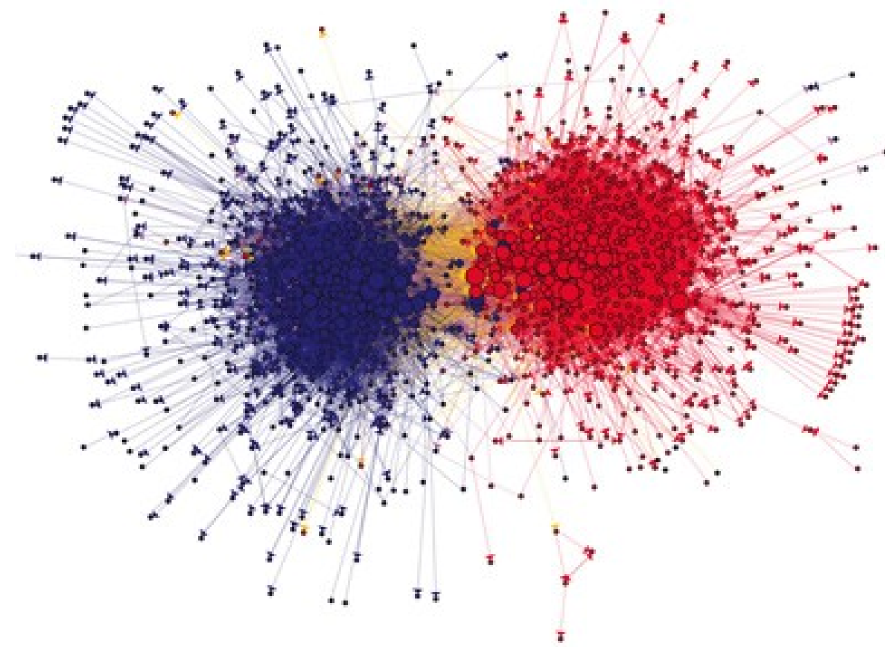
Some Typical Processing Problems

Compression / Visualization / multi-scale



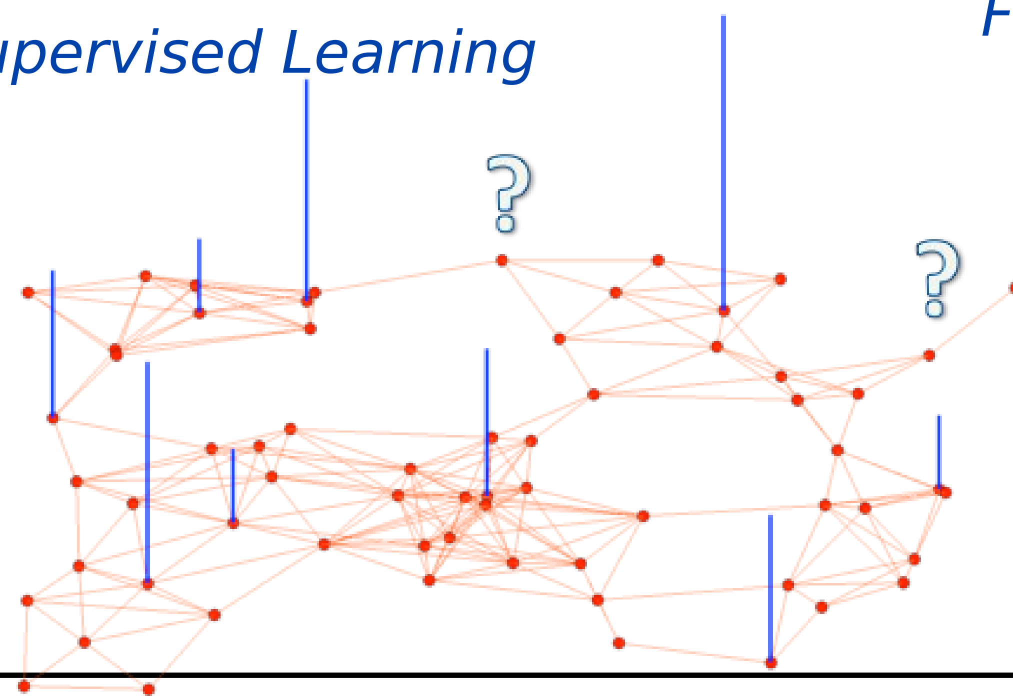
Earth data source: Frederik Simons

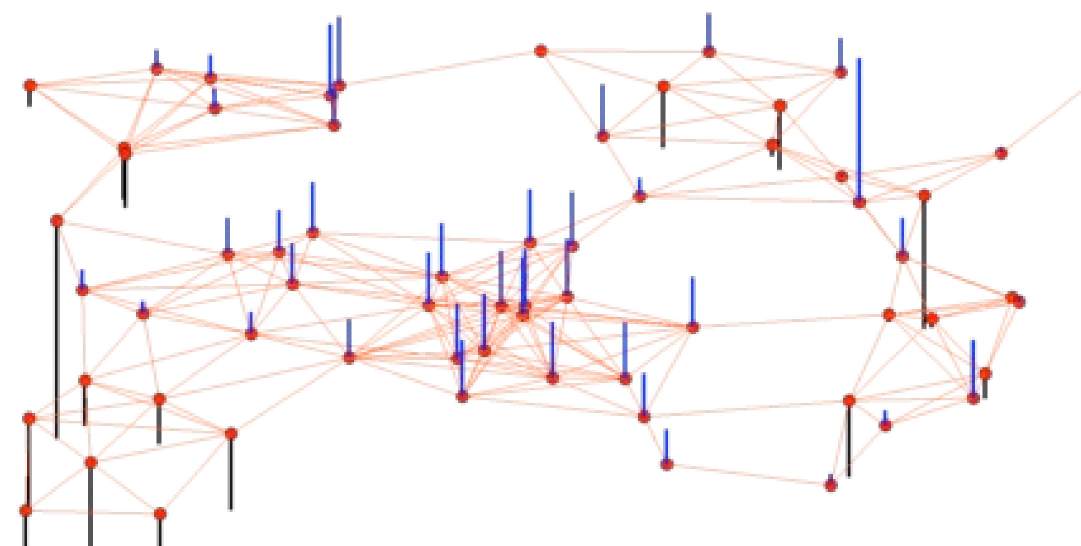
Classification



Semi-Supervised Learning

Filtering





Analysis / Information Extraction

Graph Signal Processing framework Outline

Standard signal processing $\xrightarrow{?}$ Graph signal processing

- Graph and signals
  - Definition, types of graphs, regulars / irregulars, functions on the nodes
- The Laplacian operator
  - smoothness, spectral properties, Fiedler vector, Fourier transform,
- Good and bad graphs, limits of Graph SP

  - Irregularities, small worlds, large graphs

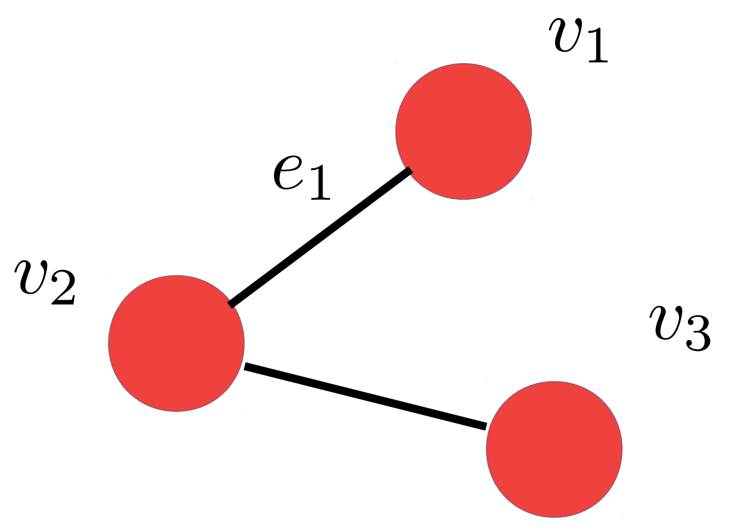
Definitions

Graph and signal

Mathematical definition

$$V = \{v_1, \ldots, v_N\} \qquad E = \{e_1, \ldots, e_M\}$$

$$e_1 = (v_1, v_2)$$



Degree, sum of connections :

$$d(v) = \left|\{u \in V \text{s.t. } (u,v) \in E \text{ or } (v,u) \in E\}\right|$$

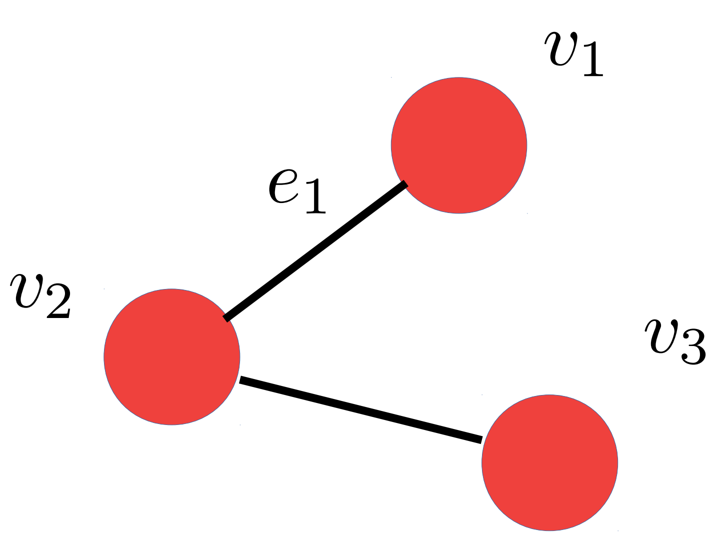
$$\mathbf{D}(G) = \text{diag}(d_1, \ldots d_N)$$

Mathematical definition

$$V = \{v_1, \ldots, v_N\} \qquad E = \{e_1, \ldots, e_M\}$$

Adjacency matrix

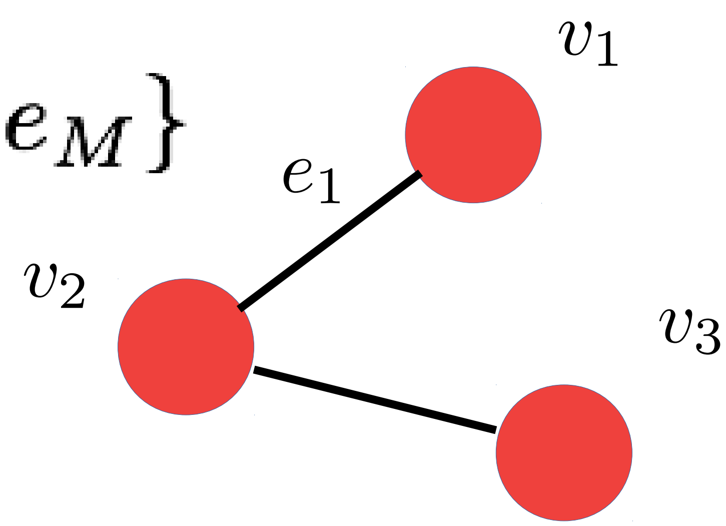
$$\mathbf{A}(i,j) = \begin{cases} +1 & \text{if there is an edge } (v_i, v_j) \text{ or } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases}$$



$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Extensions to weighted graphs

$$V = \{v_1, \ldots, v_N\} \qquad E = \{e_1, \ldots, e_M\}$$



Weight Matrix:

*A symmetric N-by-N matrix **W***

$$\mathbf{W}(i,j) \geqslant 0 \qquad \mathbf{W}(i,i) = 0$$

$$W = \begin{pmatrix} 0 & w_{12} & 0 \\ w_{21} & 0 & w_{23} \\ 0 & w_{32} & 0 \end{pmatrix}$$

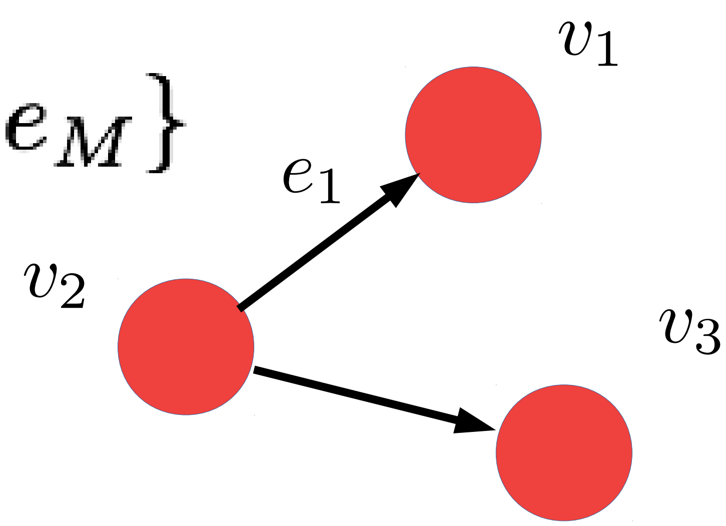
W(*i*,*j*) is the weight (“strength”) of the edge between *i*,*j* (if any)

Degrees:

$$d(v_i) = \sum_{j \sim i} \mathbf{W}(i,j)$$

Extensions to directed graphs

$$V = \{v_1, \ldots, v_N\} \qquad E = \{e_1, \ldots, e_M\}$$



Weight Matrix:

*A non-symmetric N-by-N matrix **W***

$$\mathbf{W}(i,j) \geqslant 0 \qquad W(i,j) \neq W(j,i)$$

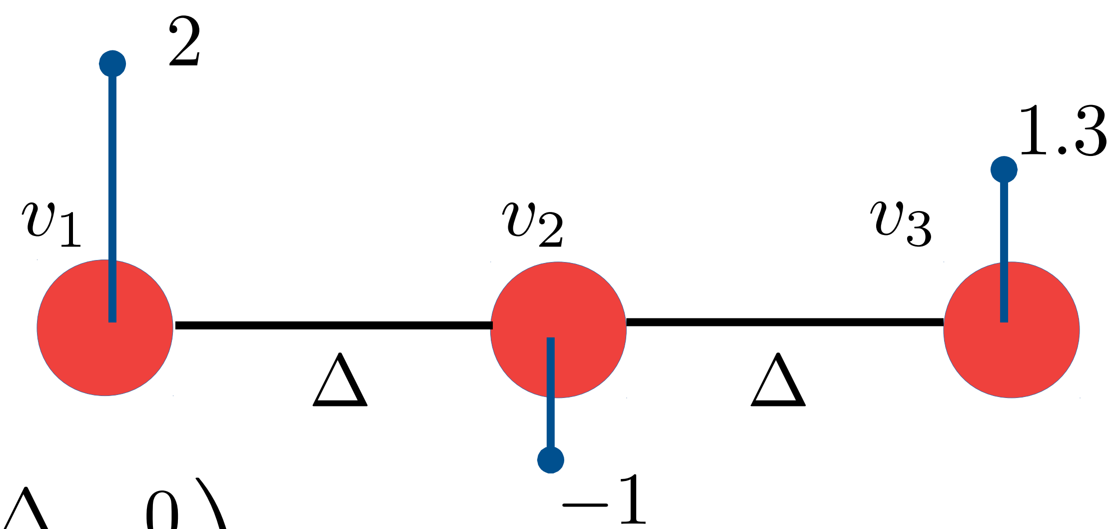
$$W = \begin{pmatrix} 0 & w_{12} & 0 \\ w_{21} & 0 & w_{23} \\ 0 & w_{32} & 0 \end{pmatrix}$$

Degrees:

$$d(v_i) = \sum_{j \sim i} \mathbf{W}(i,j)$$

Basic examples

Line graph:

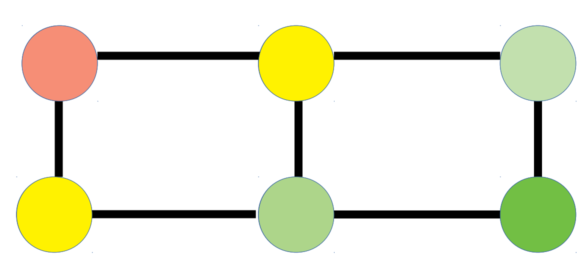


(time series)

$$W = \begin{pmatrix} 0 & \Delta & 0 \\ \Delta & 0 & \Delta \\ 0 & \Delta & 0 \end{pmatrix}$$

Values on the nodes

Lattice:

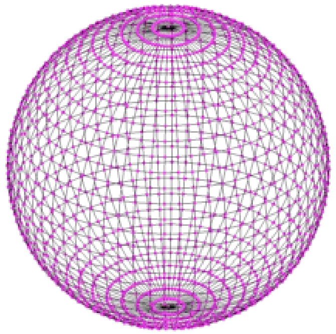


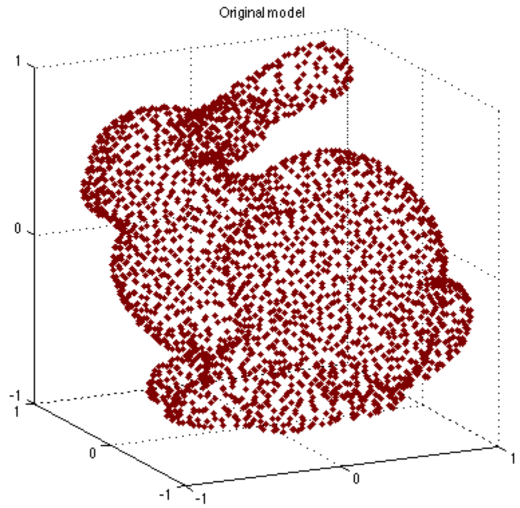
(images)

Time series, images : particular cases of Graph SP

Different kinds of graphs

Manifolds:

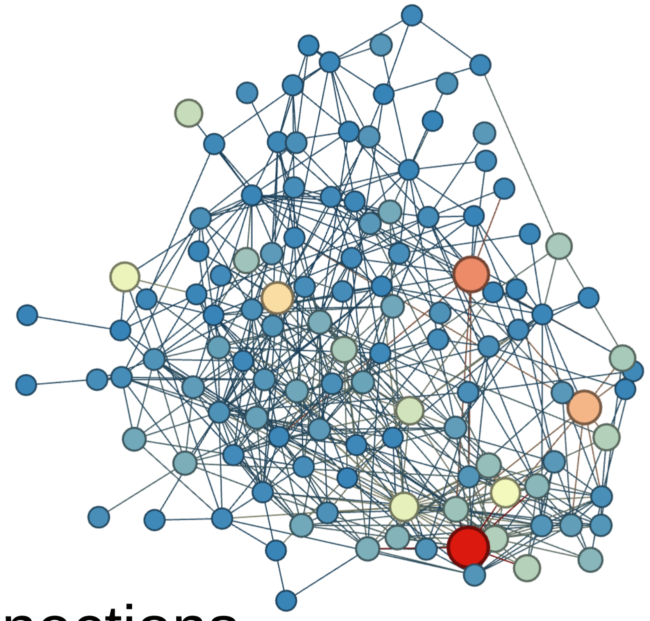


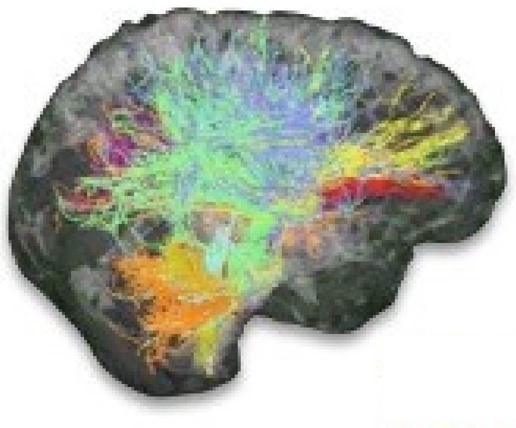


Easy representation

Smooth and regular, homogeneous degree distribution

Irregular graphs:





Small world, hubs, weak connections

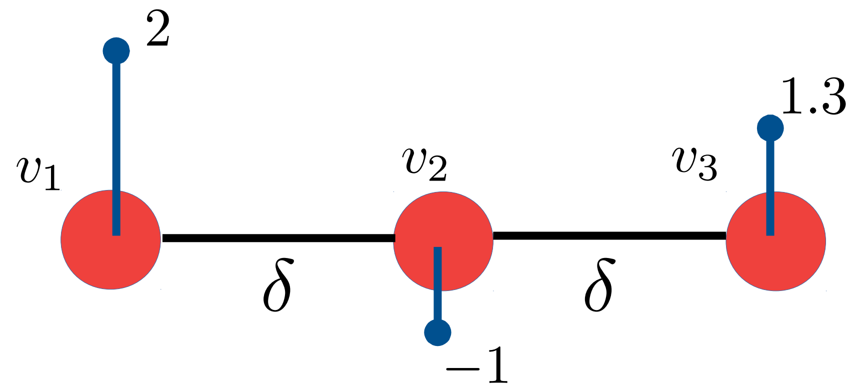
Locality, patch size depend on the node !

Definitions

Graph Laplacian

Functions defined on a graph

Basic function properties:



Variations, derivative, gradient.

$$\nabla f(i) = [f(i+1) - f(i)]/\delta$$

Becomes

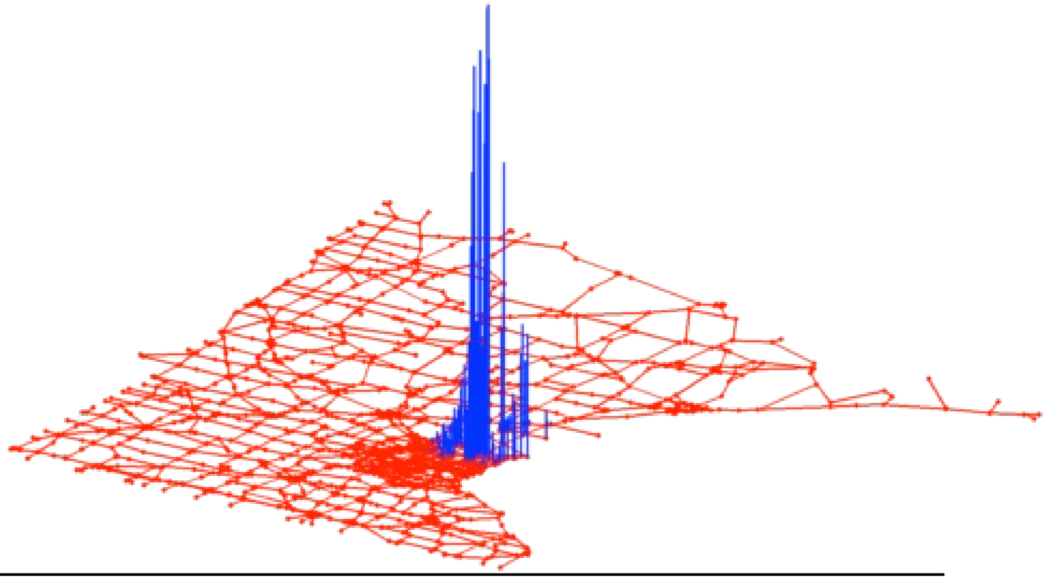
$$\nabla f(i,j) = [f(j) - f(i)]w(i,j)$$ Values on the edges !

What about the second derivative ?...

The Laplacian

$$Lf(i) = \sum_{j \in \Omega_i} [f(j) - f(i)]w(i,j)$$

Node to node space → square matrix



Graph Laplacian

With these definitions we have:

$$Lf(i) = \sum_{j \in \Omega_i} [f(j) - f(i)] w(i,j) = ((D - W)\, f)\,(i)$$

L is called ***unnormalized*** *or* ***combinatorial Laplacian*** of G

L is a symmetric, positive semi-definite matrix

- 1) There exist a normalized version of **L**

$$L_N f(i) = \sum_{j \in \Omega_i} [f(j) - f(i)] \frac{w(i,j)}{\sqrt{d(i)d(j)}} = \left(\left(I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right) f\right)(i)$$

- 2) There exist a version for directed-graph **L** = **D**-**W** , but not symmetric.

Graph Laplacian

Proposition: **L** is positive semi-definite

For any *N*-by-*N* weight matrix **W**, if **L = D-W** where **D** is the degree matrix of **W**, then

$$x^T \mathbf{L} x = \frac{1}{2} \sum_{i,j} \mathbf{W}(i,j)(x[i] - x[j])^2 \geqslant 0 \quad \forall x \in \mathbb{R}^N$$

Rem: to ease notations we will sometimes use $w_{ij} = \mathbf{W}(i,j)$

Since **L** is real, symmetric and PSD:

- It has an eigendecomposition into real eigenvalues and eigenvectors λ_i, u_i
- The eigenvalues are non-negative

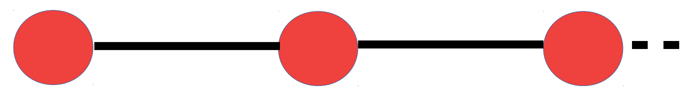
$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_N$$

$\updownarrow$

$$\mathbf{L}\mathbf{1} = 0$$

What can be learned from eigenvectors and eigenvalues ?

Some examples

Path graph  DCT II transform

$$\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 2 & & & \\ & & 2 & & \\ & & & \ddots & \\ & & & & 2 & \\ & & & & & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{bmatrix}$$

$$\lambda_k = 2 - 2\cos\frac{\pi k}{N} = 4\sin^2\frac{\pi k}{2N},\ k = 0, \ldots, N-1$$

$$u_k[\ell] = \cos\left(\pi k(\ell + \frac{1}{2})/N\right),\ \ell = 0, \ldots, N-1$$

Some examples

Ring graph

$$\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ -1 & & & -1 & 2 \end{pmatrix}$$

Discrete Fourier transform

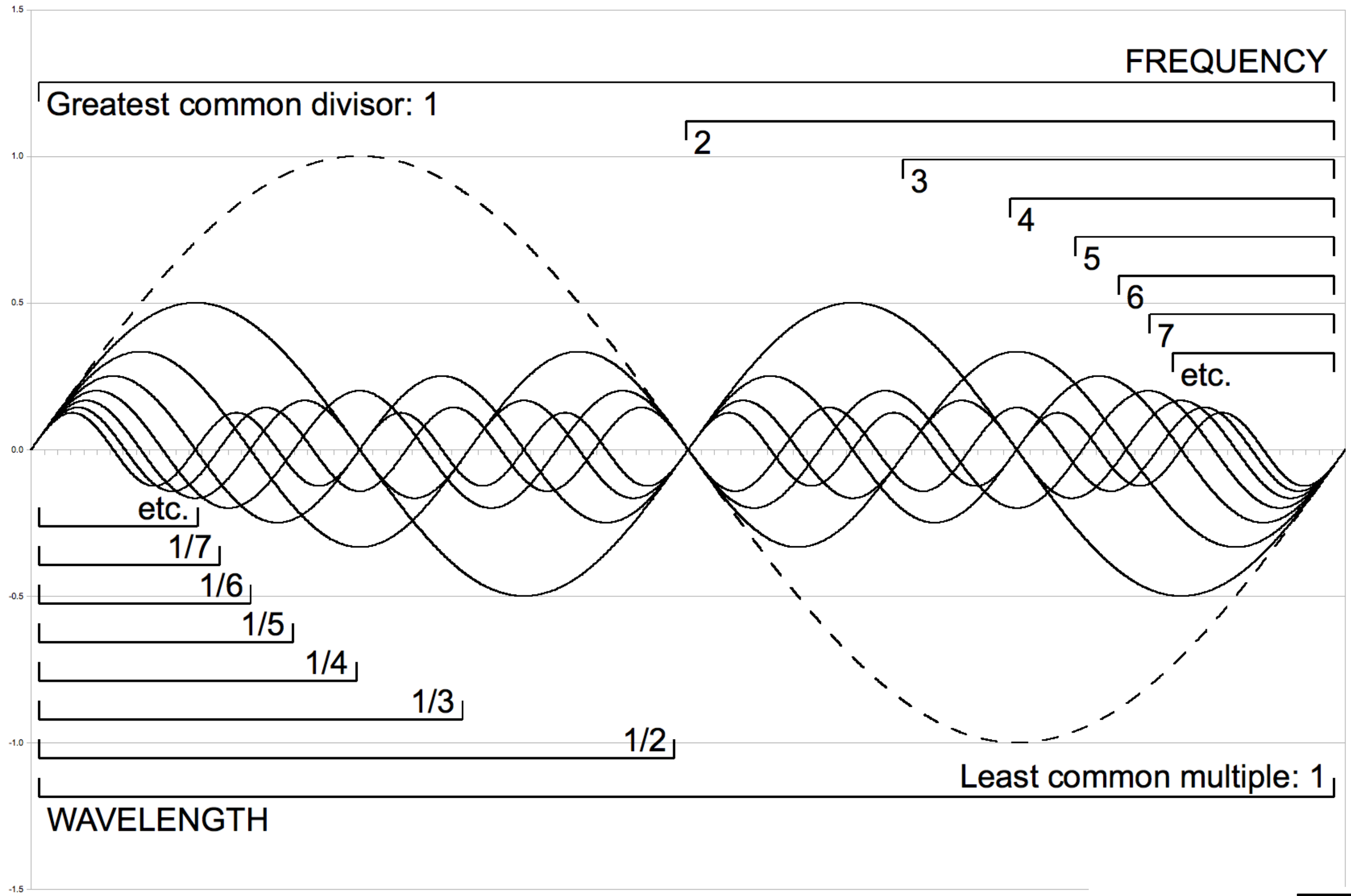
$$\lambda_k = 2 - 2\cos\frac{\pi k}{N} = 4\sin^2\frac{\pi k}{2N},\ k = 0, \ldots, N-1$$

$$u_k^c[\ell] = \cos\left(2\pi k\ell/N\right),\ \ell = 0, \ldots, N-1$$

$$u_k^s[\ell] = \sin\left(2\pi k\ell/N\right),\ \ell = 0, \ldots, N-1$$

Graph Fourier transform

Generalization of the Fourier transform to graphs :
Eigenvectors of the Graph Laplacian = Graph Fourier modes



As smooth as possible + orthogonality

$$x^T y = \sum_i x[i]y[i]$$

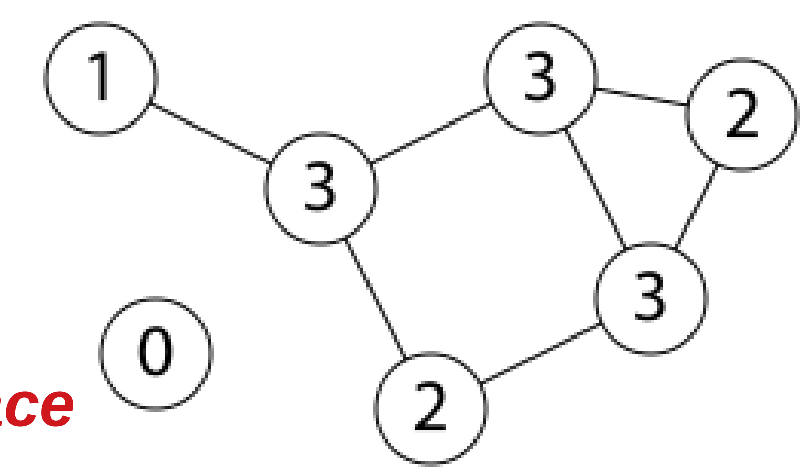
Proposition: eigendecomposition of **L** and structure of *G*

The number of connected components *c* of *G* is the dimension of the nullspace of **L**. Furthermore the null space of **L** has a basis of indicator vectors of the connected components of *G*

$$x^T L x = \frac{1}{2}\sum_{i,j} w_{i,j}^2 (x[i] - x[j])^2$$

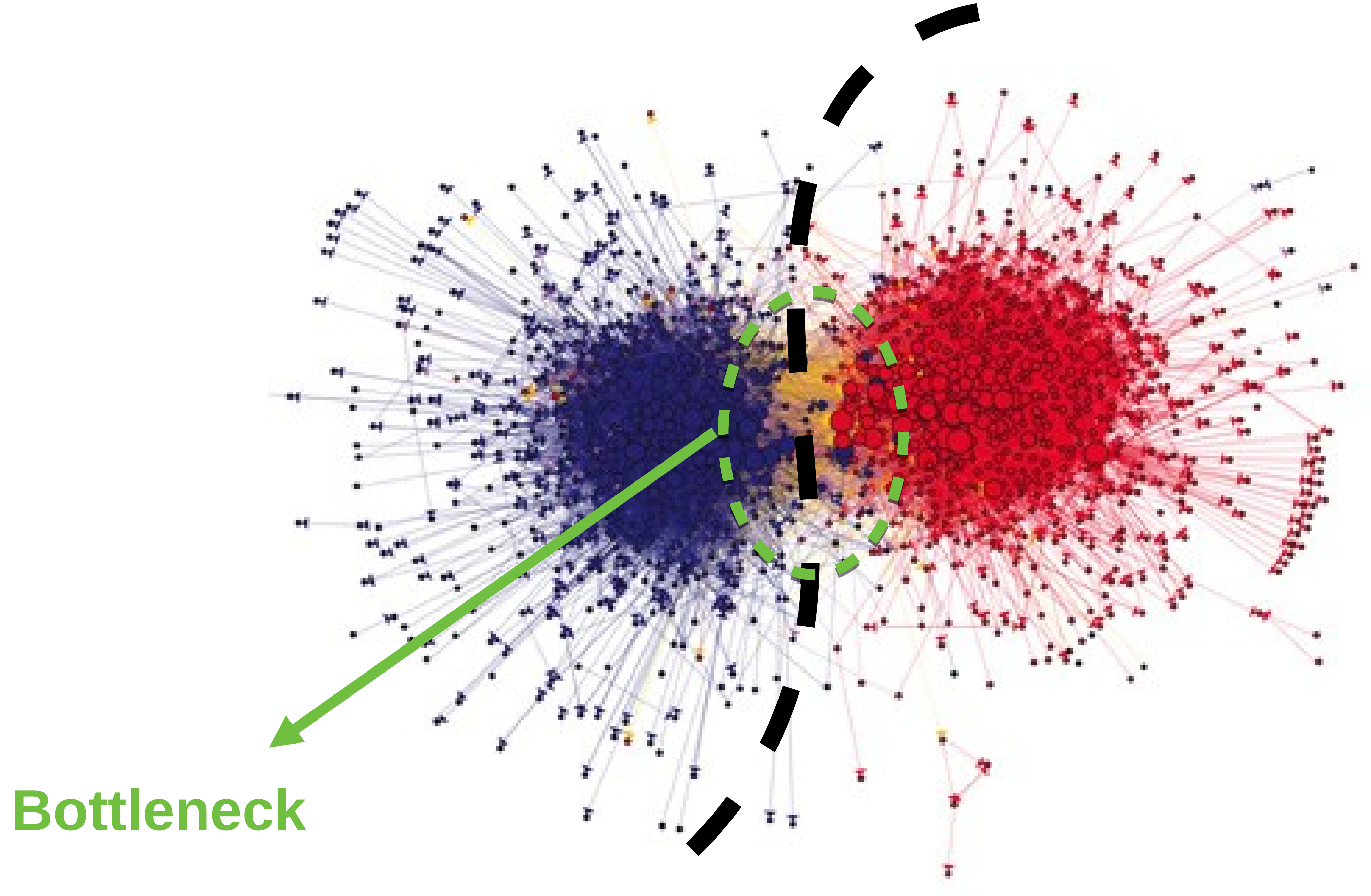
Indicator of a subset *H* of *V* is

$$x \in \mathbf{R}^N \text{ s.t. } \begin{cases} x[i] = 1 & \text{if } \in H \\ x[i] = 0 & \text{otherwise} \end{cases}$$



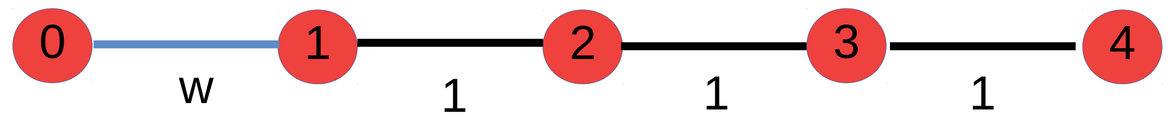
Each connected component is an independent space

The Fiedler vector

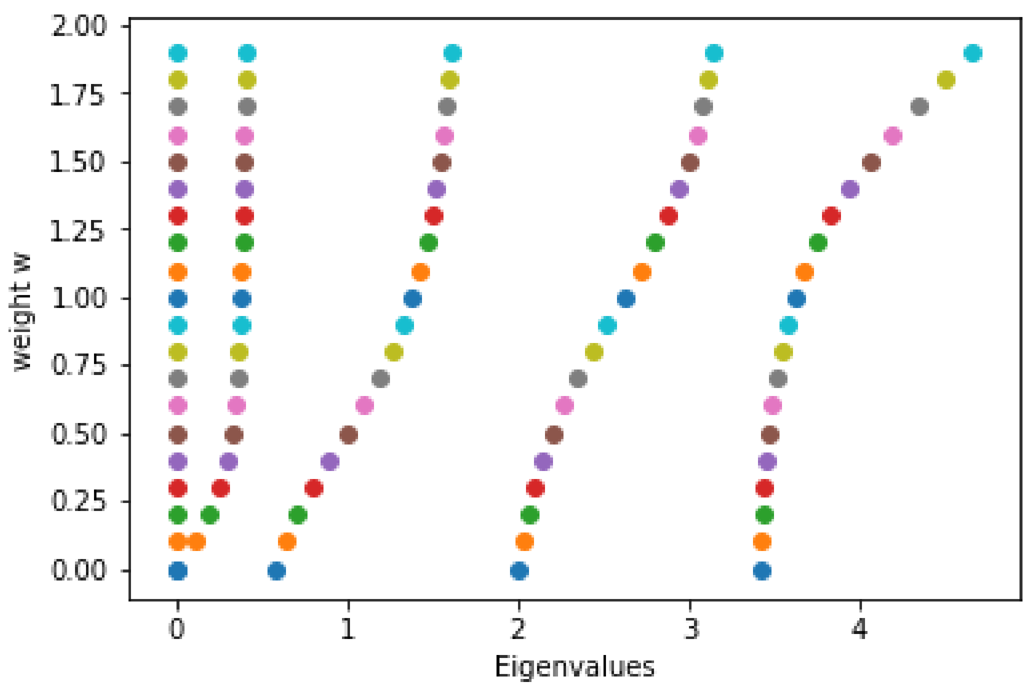


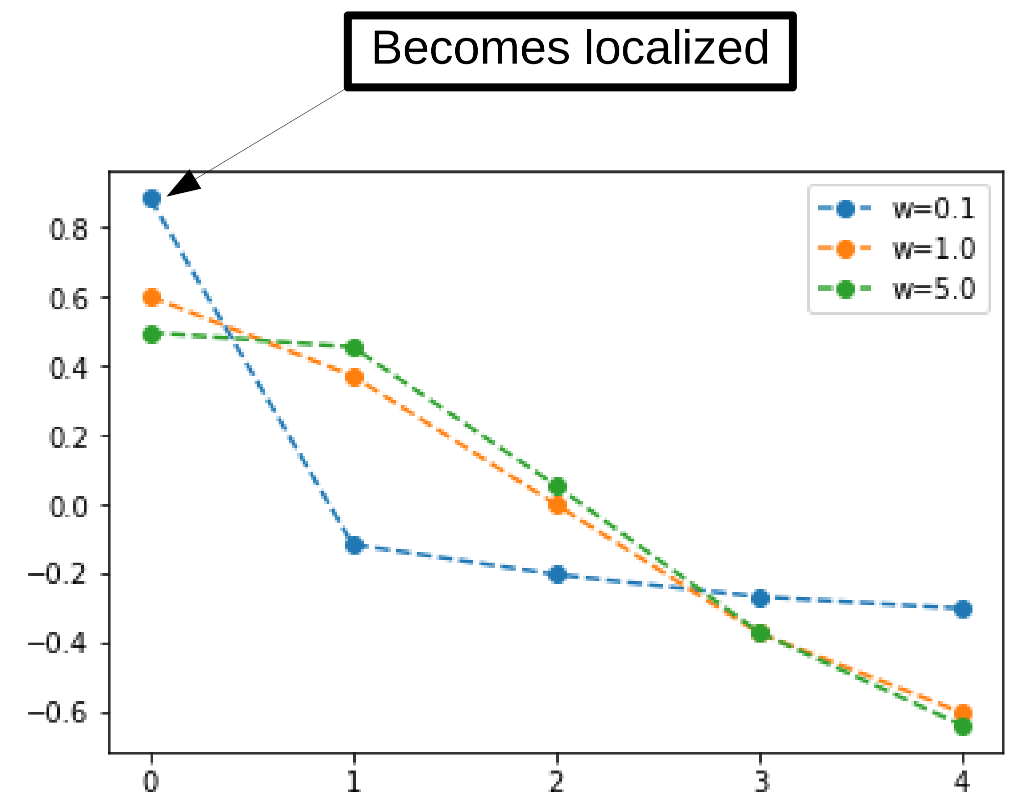
Rem. : Eigenvectors contain global information about the graph

Laplacian eigenvectors



Variable weight w.
Influence on the spectral properties ?





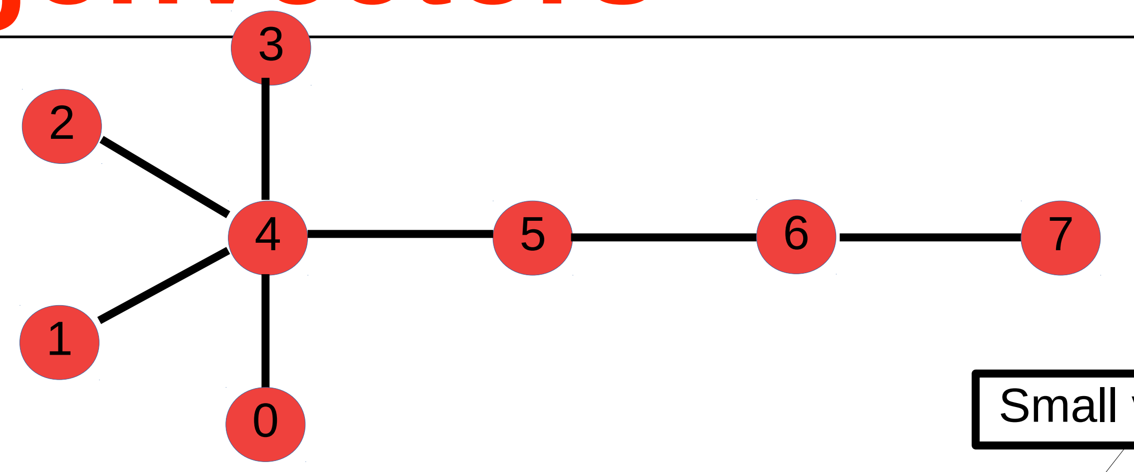
Fiedler vector

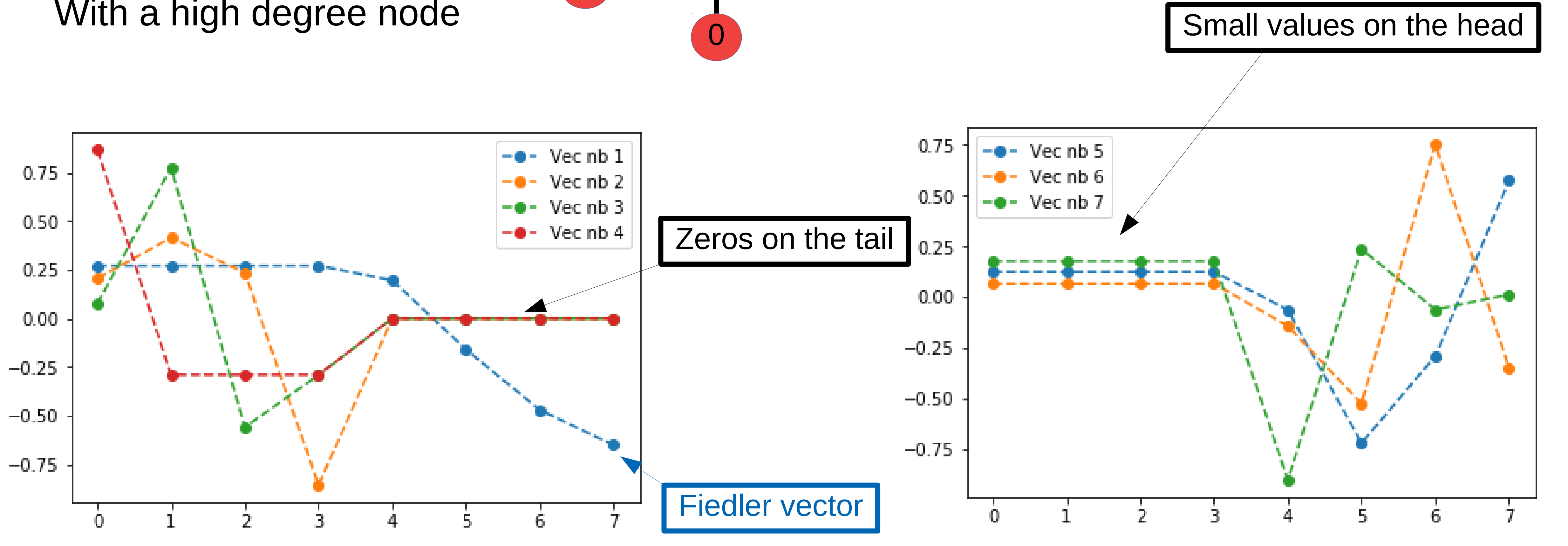
W=0 : 2 disconnected components
W=5 : strong connection between node 0 and 1
Localized variation impacts all the spectrum

Laplacian eigenvectors

Comet graph

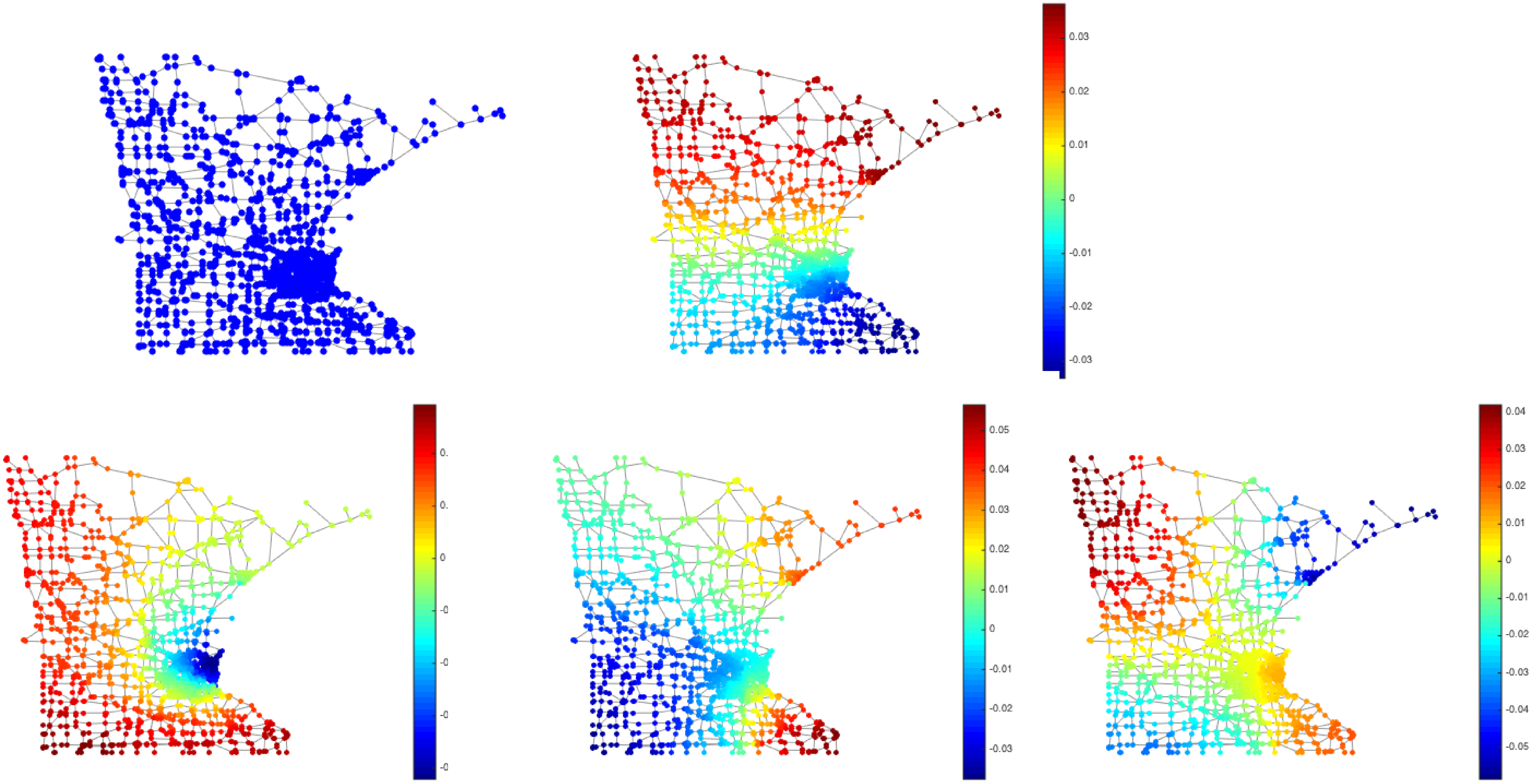
Inhomogeneous graph,
With a high degree node





Eigenvectors localized in the different structures

A Few Laplacian Eigenvectors

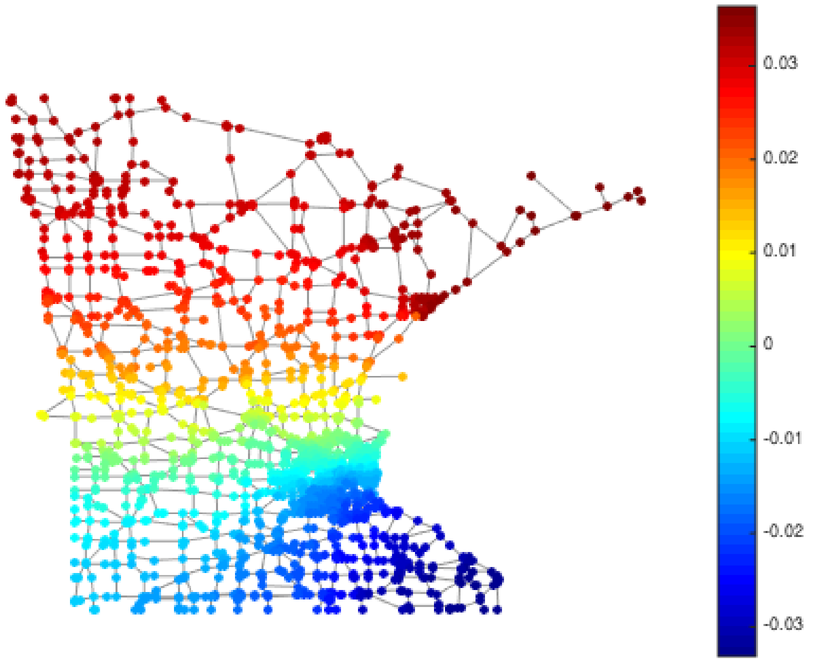


Localization versus spreading

Standard Fourier :

$$\delta \rightarrow c$$

$$c \rightarrow \delta$$



Well spread on the graph
GFT : delta on the Laplacian spectrum

Sparse on the graph $\rightarrow$ well spread in Graph Fourier
NOT ALWAYS : depends on the graph

Localized eigenvectors may exist
- near weakly connected nodes, hubs

Normalized Graph Laplacian

Note: we will sometimes need to consider the generalised problem

$$\mathbf{L}u = \lambda \mathbf{D}u$$

In this case it makes sense to introduce the normalised Laplacian

$$\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$$

Eigenvectors are closely related

$$\mathbf{L}_{\text{norm}} f = \lambda f \rightarrow u = D^{-1/2} f$$

Normalized Graph Laplacian

Eigenvalues of the normalised Laplacian

$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_N \leqslant 2$$

Algebraic connectivity

IFF bipartite graph!

Summary

- Signal variations : no gradient but laplacian
- Laplacian spectrum : a way to generalize the Fourier Transform
- Multiplicity of $\lambda=0$: number of connected components
- Fiedler vector separates the graph in 2
- Laplacian eigenvectors : contain global information
- Eigenvectors not always spread over the graph
- Concentration where inhomogeneous

Locality and diffusion

Neighborhood, locality and diffusion

Example: Diffusion on Graphs

Consider the following « heat » diffusion model

$$\frac{\partial f}{\partial t} = -\mathcal{L} f \qquad \frac{\partial}{\partial t}\hat{f}(\ell, t) = -\lambda_\ell \hat{f}(\ell, t) \qquad \hat{f}(\ell, 0) := \hat{f}_0(\ell)$$

$$\hat{f}(\ell, t) = e^{-t\lambda_\ell}\hat{f}_0(\ell) \qquad f = e^{-t\mathcal{L}} f_0$$ by functional calculus

Explicitly: $f(i,t) = \sum_\ell \sum_{j \in V} e^{-t\lambda_\ell} u_\ell(i) u_\ell(j) f_0(j)$

Discrete version :

$$\frac{f(t+\delta t) - f(t)}{\delta t} = -Lf(t)$$

$$f(t+\delta t) = f(t) - \delta t\ Lf(t)$$

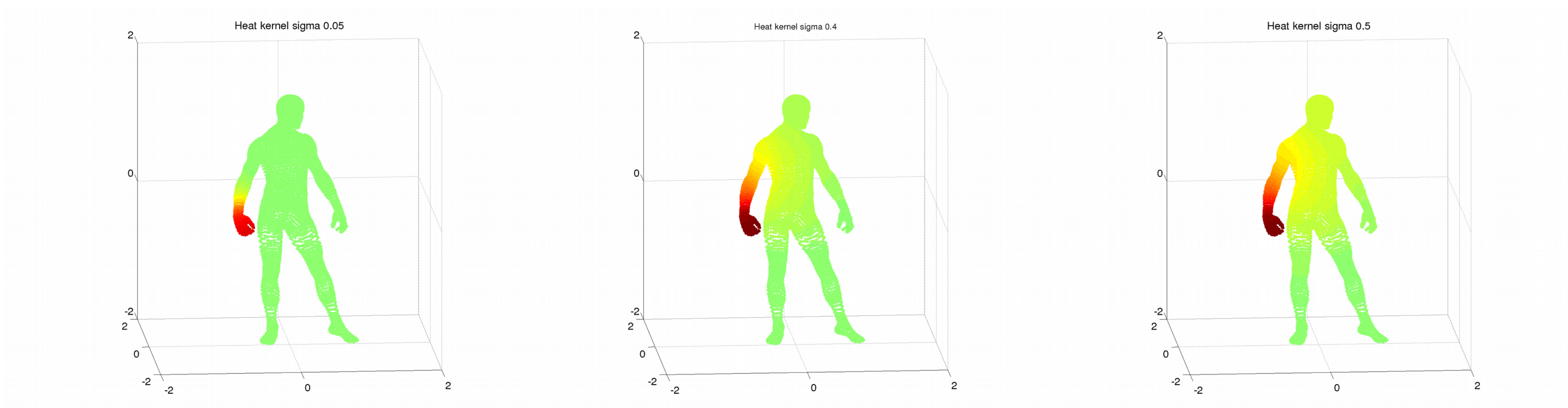
Iterative process

$$f(n) = (1 - \delta t\ L)^n f_0$$

L *: one-hop neighbors* L^n *: n-hop neighbors*

Example: Diffusion on Graphs

Examples of heat kernel on graph



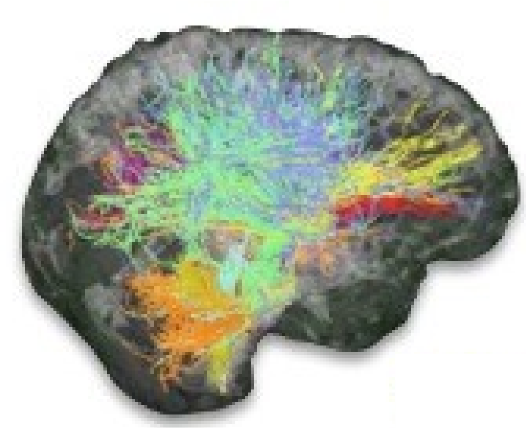
$$f_0(j) = \delta_k(j)$$

$$f(i) = \sum_{\ell} e^{-t\lambda_\ell} \hat{f}_0(\ell) u_\ell(i)$$

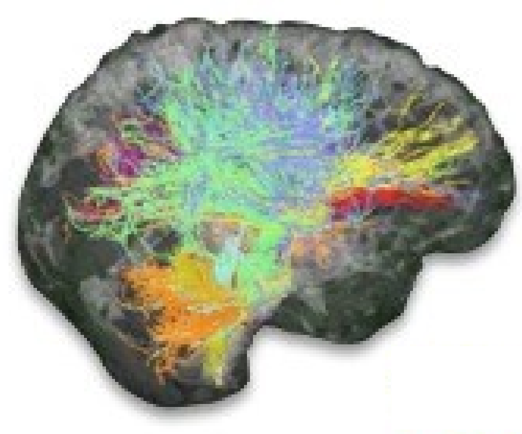
$$= \sum_{\ell} e^{-t\lambda_\ell} u_\ell(k) u_\ell(i)$$

What about a well connected graph ?

→ *Extremely fast diffusion*



Good graph, bad graph

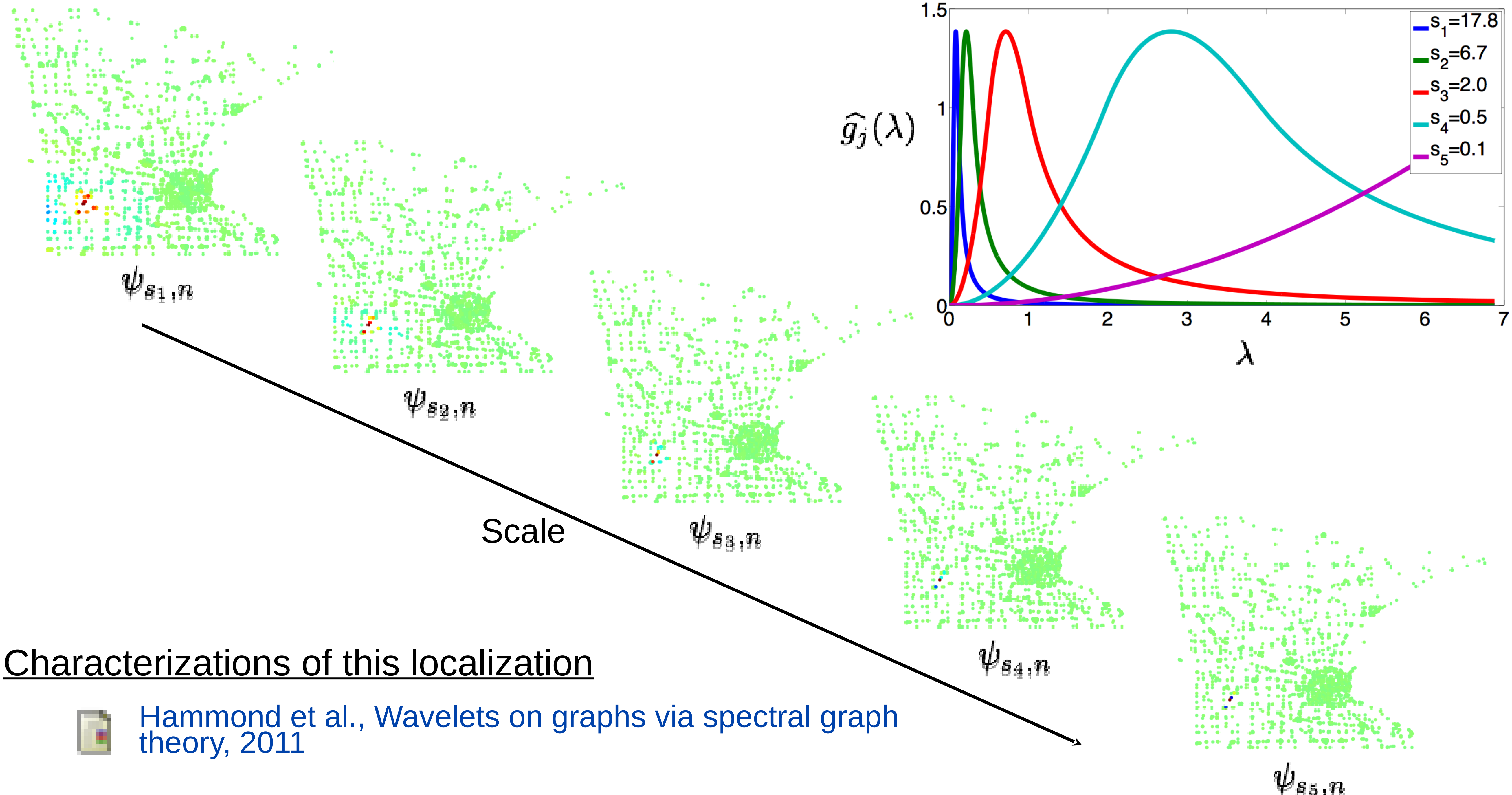


Well connected graph :
good for diffusion, transmission of information

But good for signal processing ?
If all nodes are connected together :
-how to denoise ?
-how to infer labels ?
-what is the size of a patch, neighborhood ?

Detecting patterns at different scales

Spectral Graph Wavelet Localization

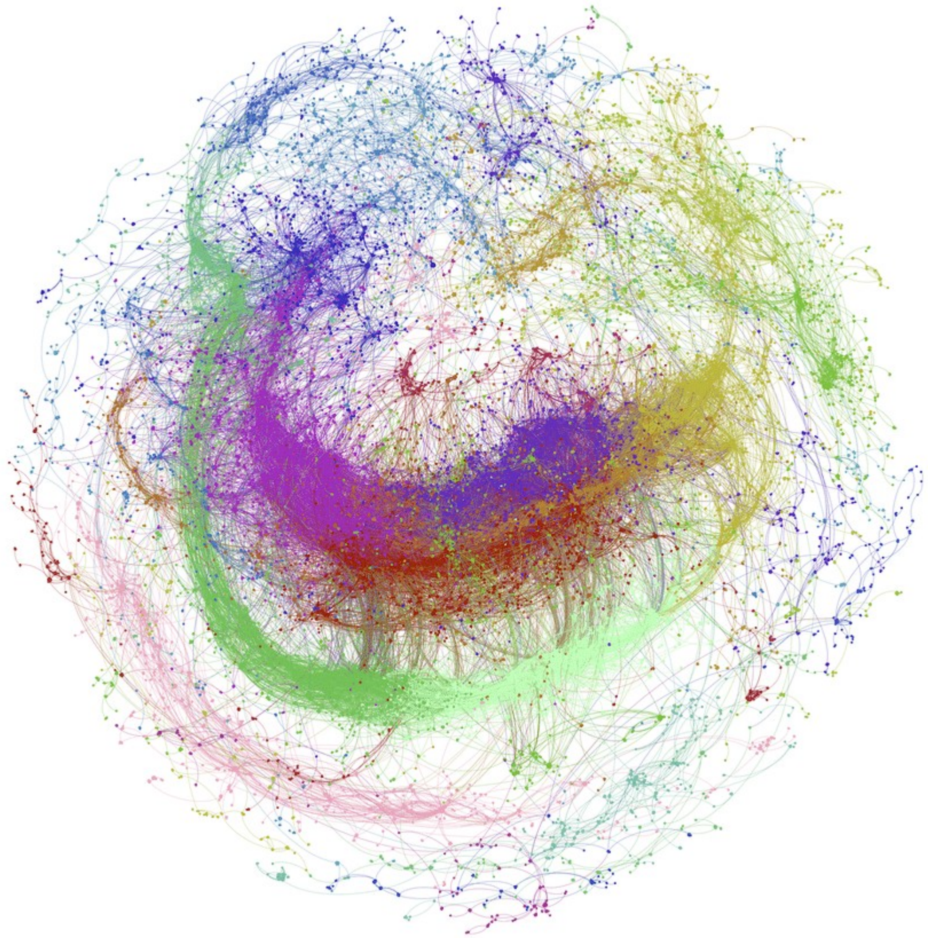


Characterizations of this localization

- Hammond et al., Wavelets on graphs via spectral graph theory, 2011
- Shuman et al., Vertex-frequency analysis on graphs, 2013

Large graphs

Large graphs :
Diagonalization of the Laplacian prohibitive
Sparse but N^3



Some solutions

Iterative application of the Laplacian faster (sparse matrix)

$(a_0 L^0 + a_1 L^1 + a_2 L^2 + \dots) f$

Graph coarsening Lukas A., Graph reduction by local variation, 2018

Approximate methods, random sampling

Puy G. et al. Random sampling of bandlimited signals on graphs 2016