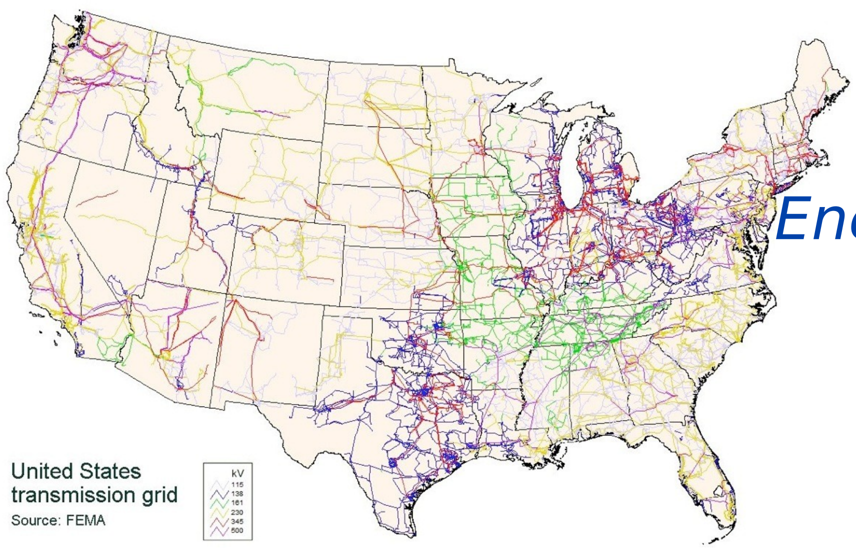
Introduction to graphs and Laplacian eigenmaps

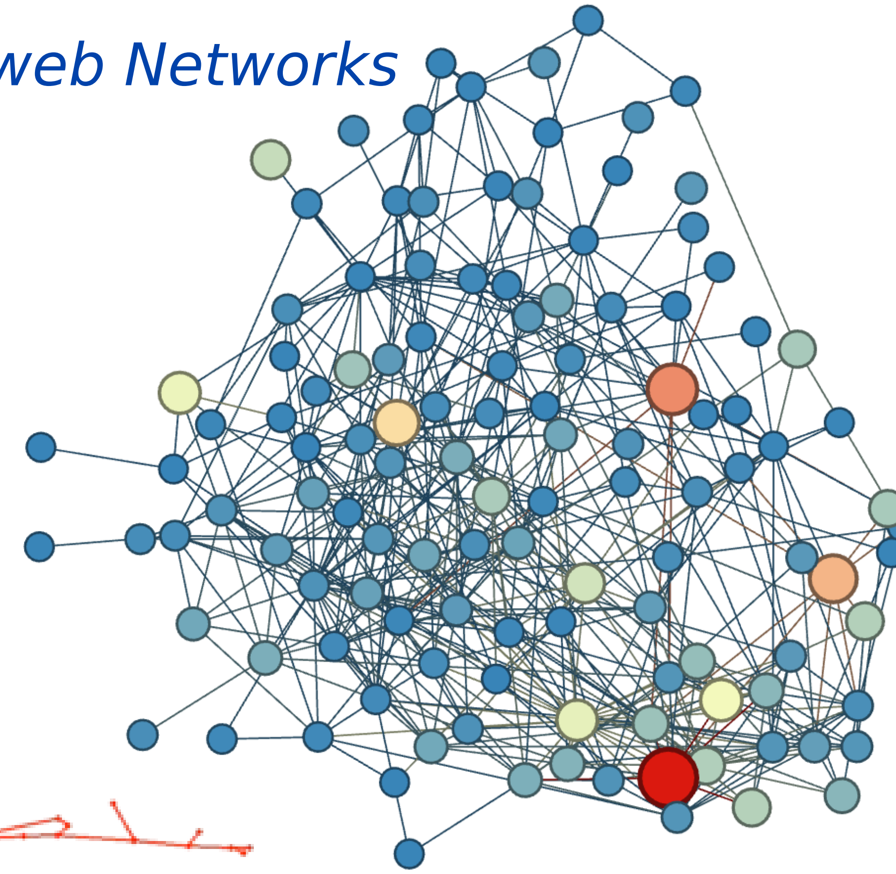
Benjamin Ricaud
Arctic University of Norway

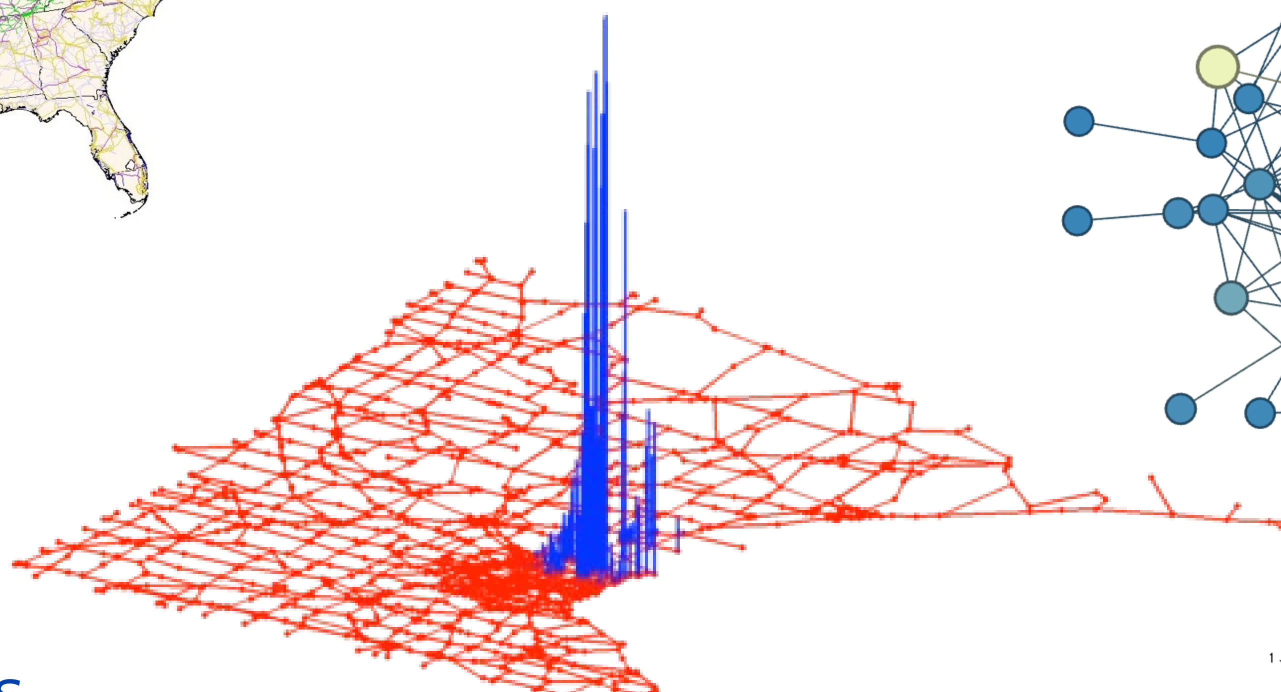
Processing Data on/with Graphs



Energy Networks

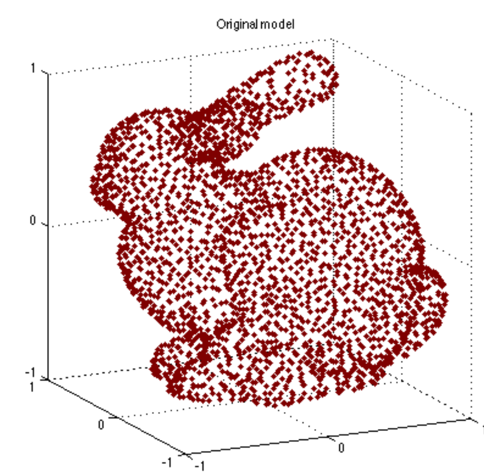
Social/web Networks

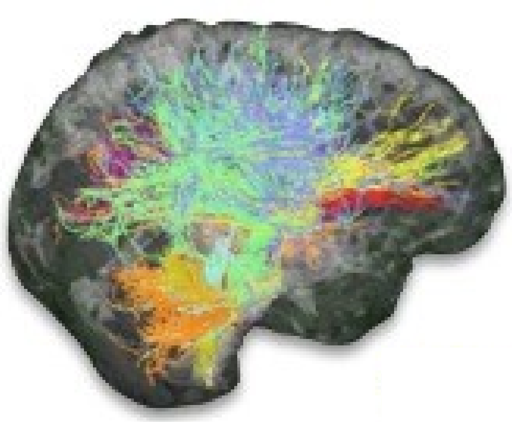




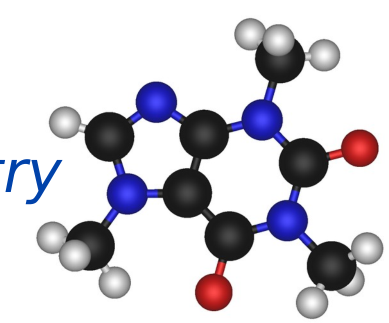
Biological Networks

Transportation Networks





Chemistry

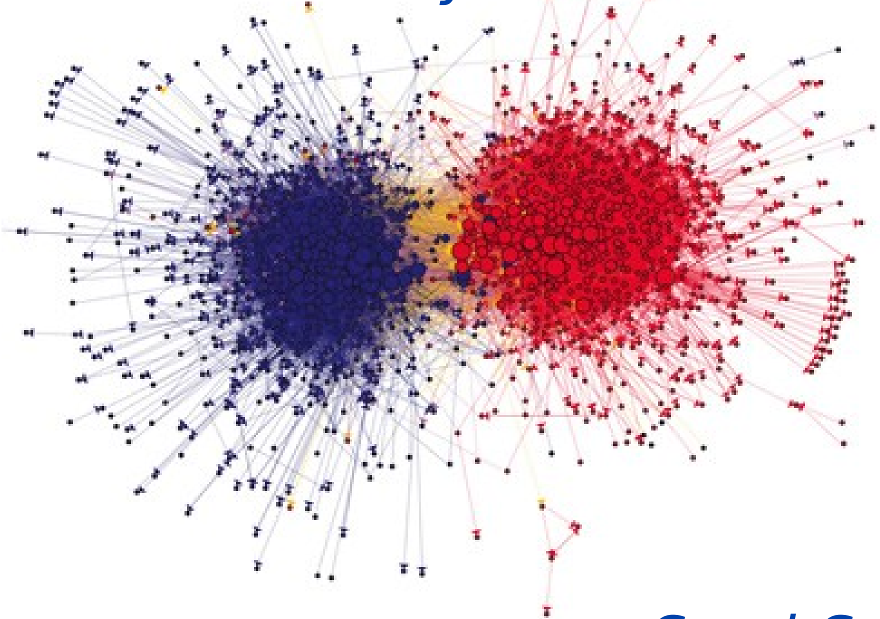


Meshes, grids

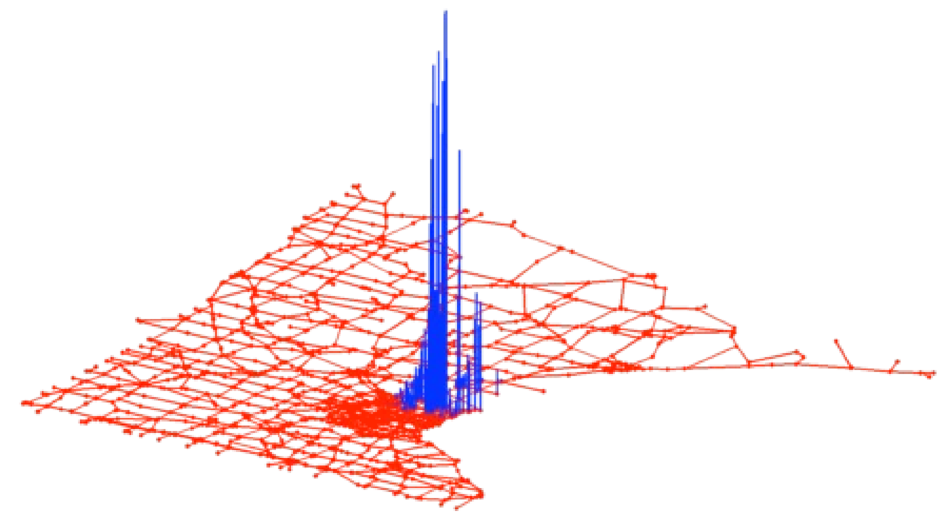
Encode neighbors relationship, locality, affinity

Some Typical Processing Problems

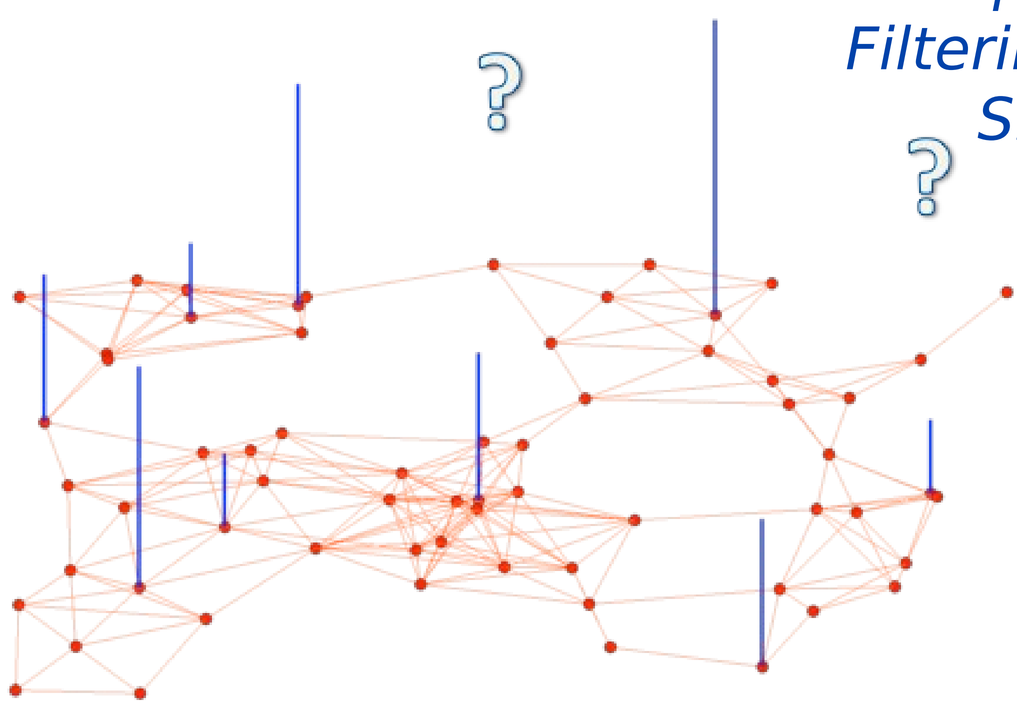
Classification
Clustering
Community detection

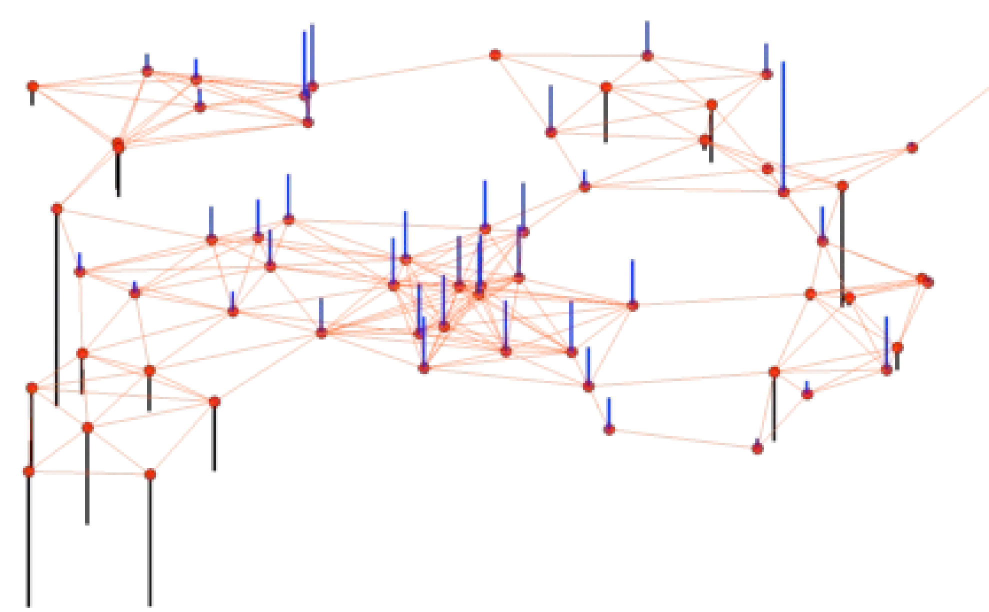


Pattern detection (GNNs)



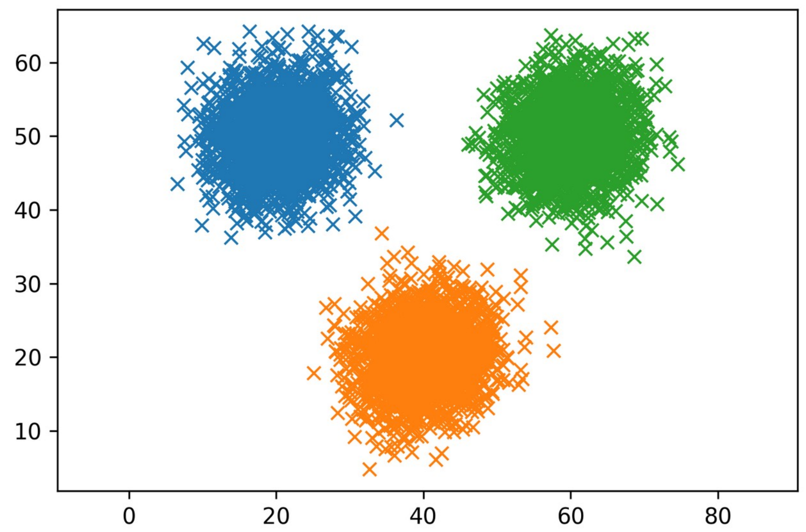
Semi-Supervised Learning
Filtering / denoising
Smoothing

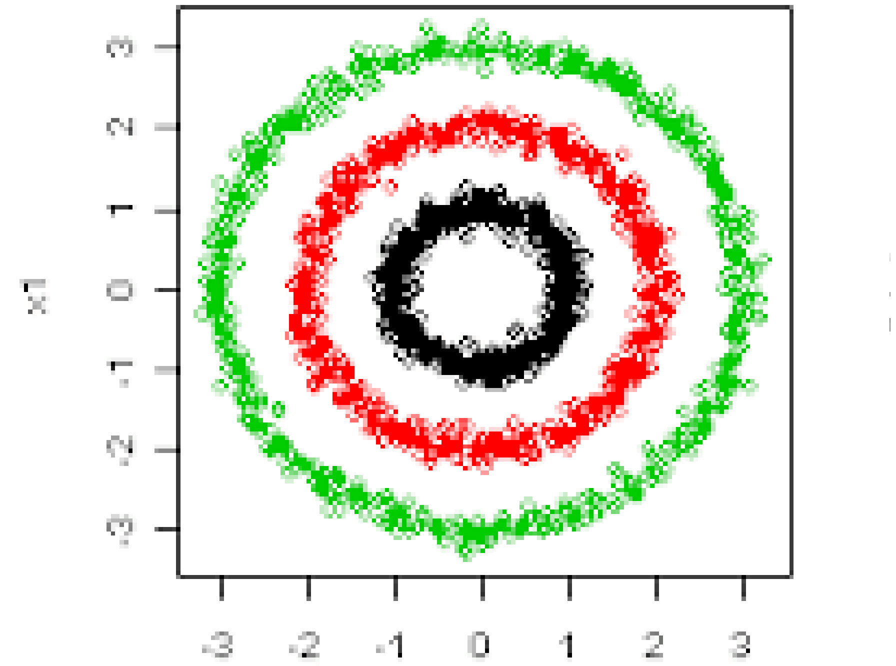


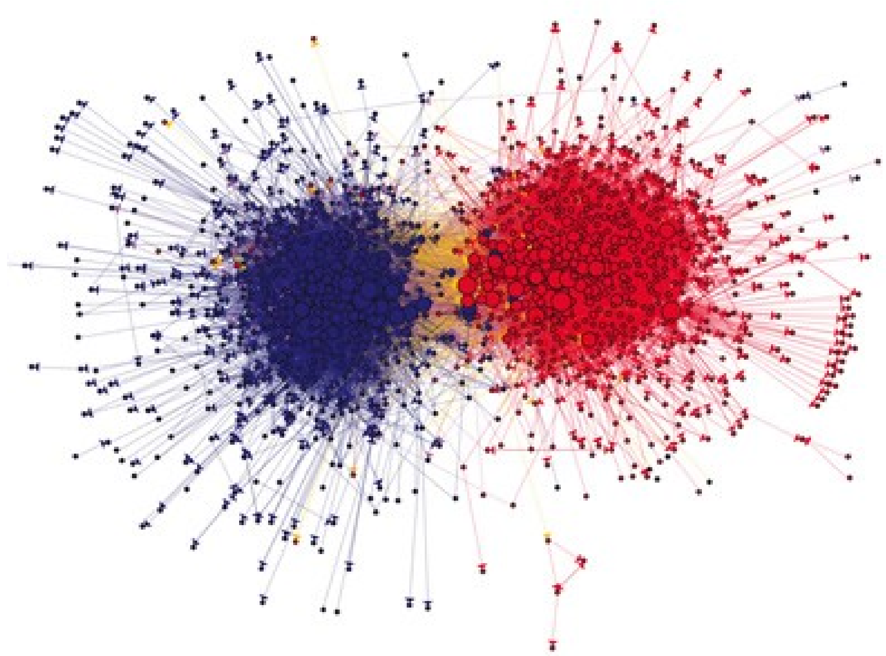


Our case : group neighbors together

Connecting the sample points in the feature space
Graph of nearest neighbors





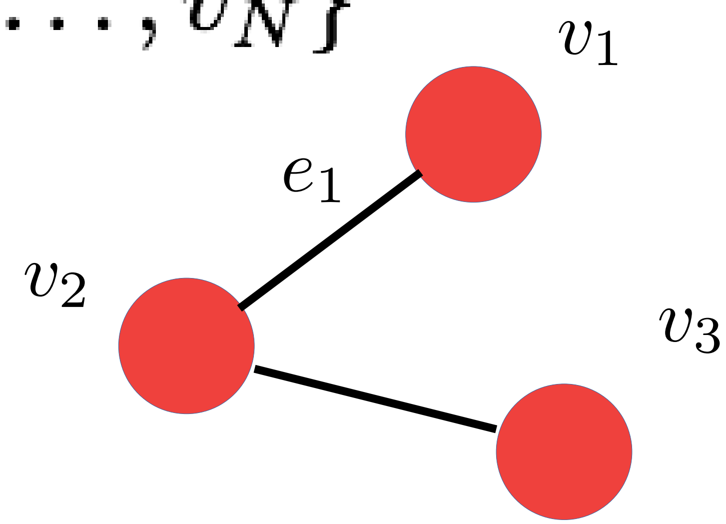


Point clouds → graph

1) Define a distance in the feature space
2) Compute the distance
3) Connect the neighbors

Mathematical definition

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

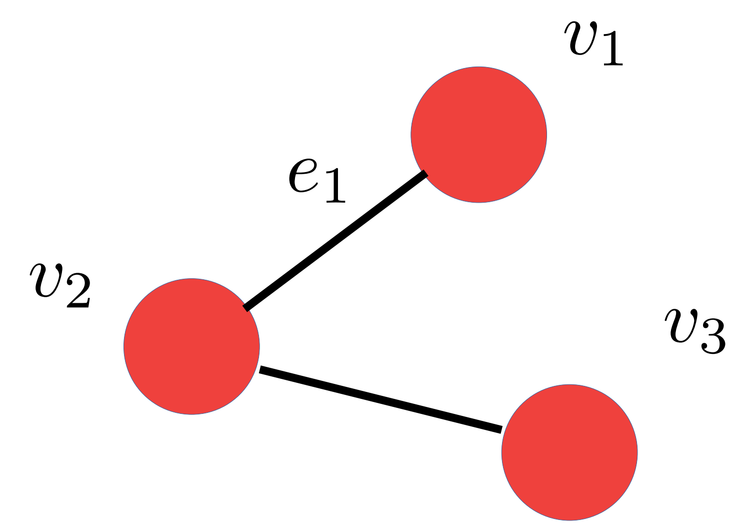


Adjacency matrix

$$A = \begin{array}{cc} & \begin{array}{ccc} \color{red}{v_1} & \color{red}{v_2} & \color{red}{v_3} \end{array} \\ & \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{array}{c} \color{red}{v_1} \\ \color{red}{v_2} \\ \color{red}{v_3} \end{array} \end{array}$$

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

Extensions to weighted graphs



Weight Matrix:

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

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

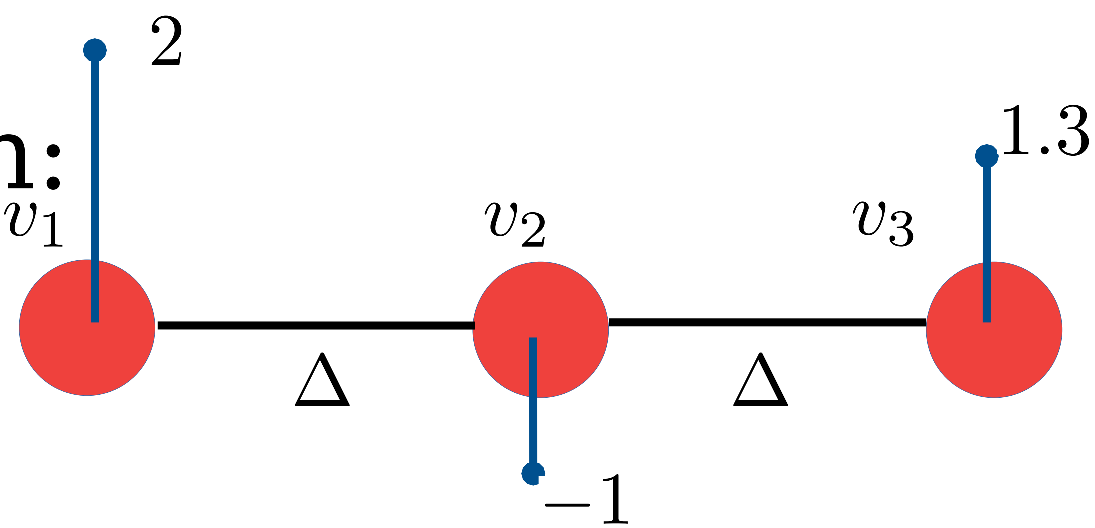
W *symmetric, positive entries*

Degrees:

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

Values on a graph : signal

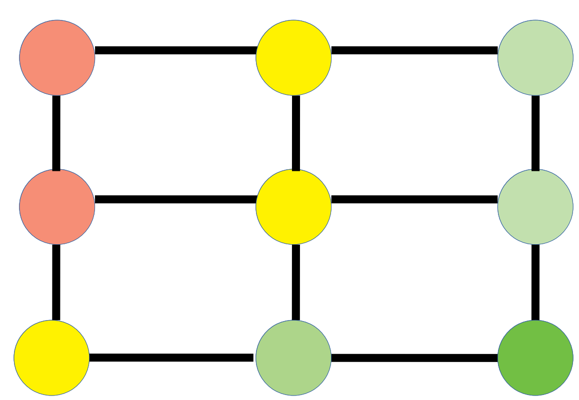
Line graph:



(time series)

$$W = \begin{pmatrix} 0 & \Delta & 0 \\ \Delta & 0 & \Delta \\ 0 & \Delta & 0 \end{pmatrix}$$ *+ Values on the nodes* $$\begin{pmatrix} 2 \\ -1 \\ 1.3 \end{pmatrix}$$

Lattice:

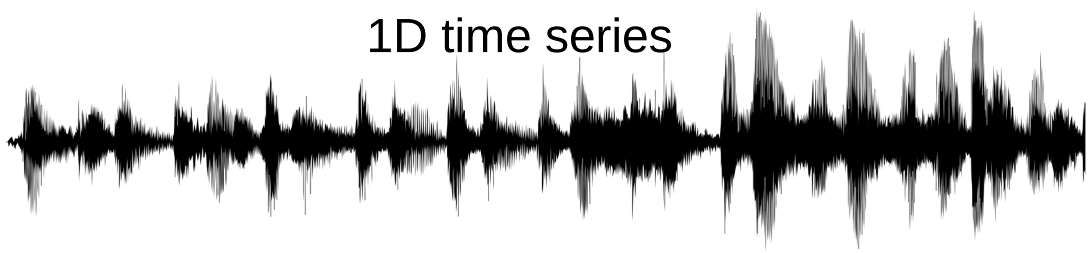


(images)

Time series, images : particular cases of Graph SP

Graph signal processing

1D time series

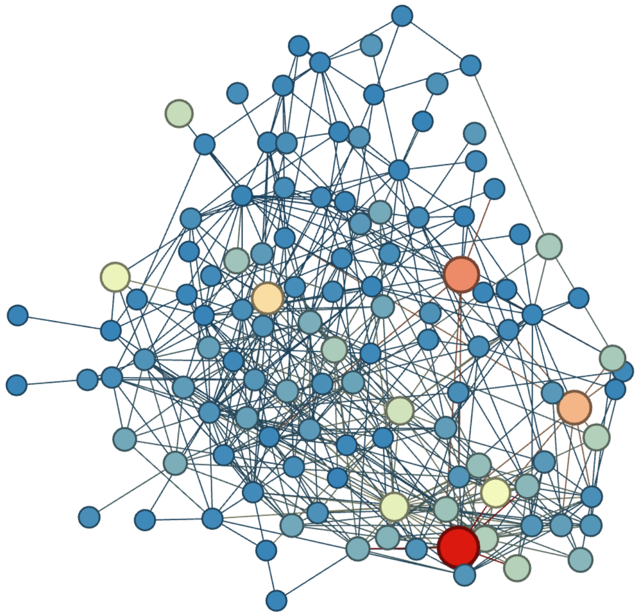


2D images

- Combining graph and signal

Irregular Manifold

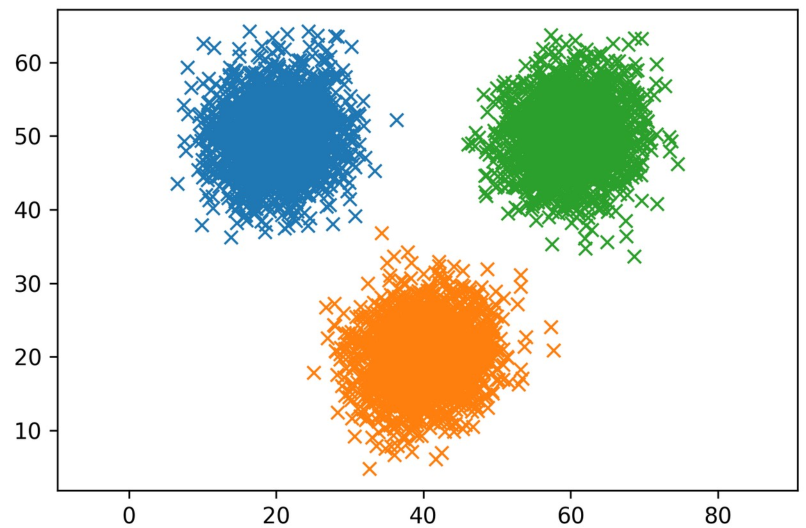
Social network ?

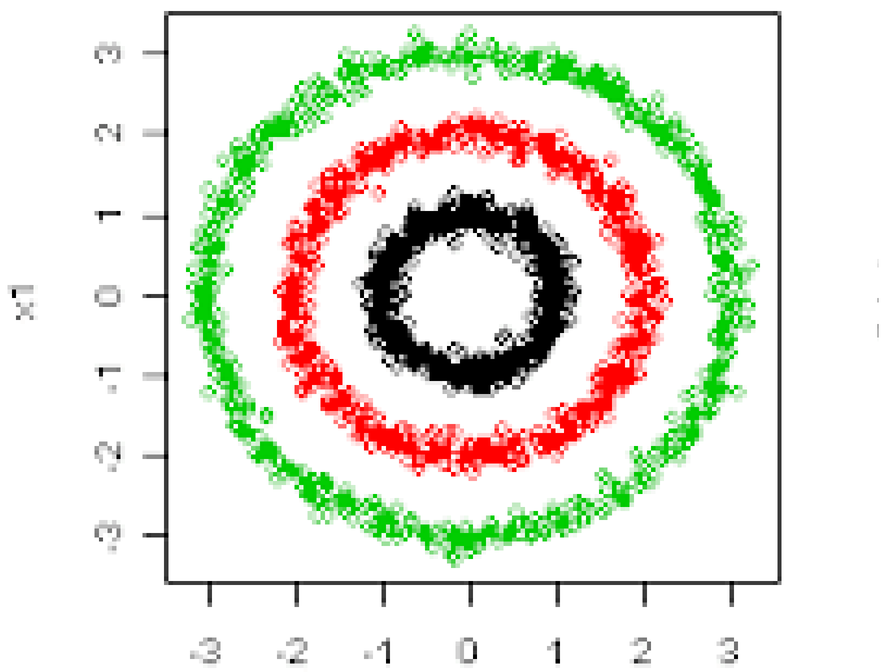


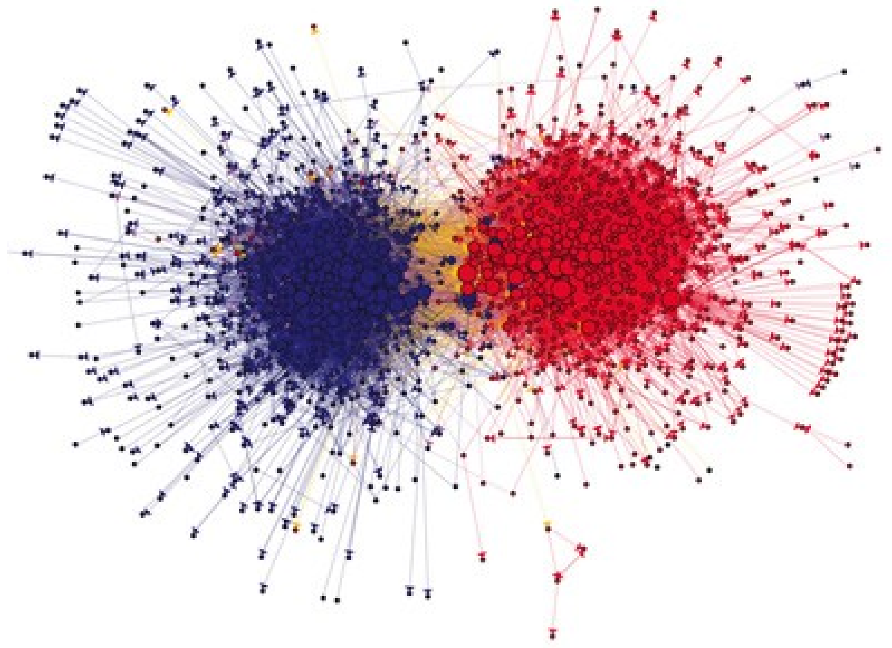
- Fourier transform, wavelets, filtering on a graph
  - ML : inductive bias, a-priori information

Our case : find communities or encode the similarity

Give a feature vector for each node



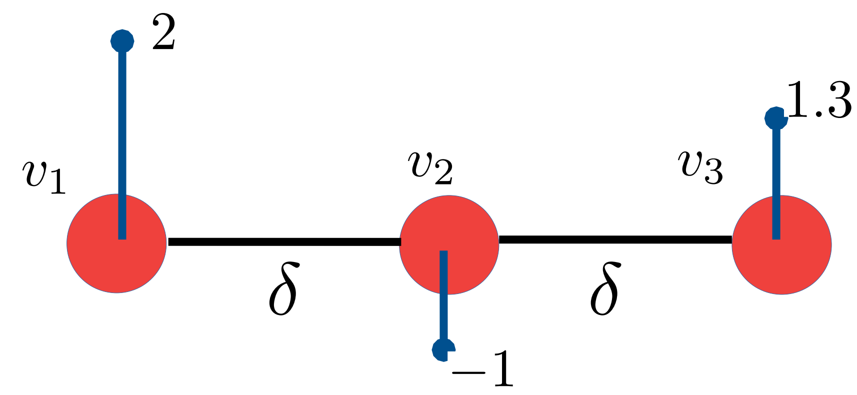




Neighbors should have close values

Analysis on a graph

Basic function properties:



Variations, derivative, gradient.

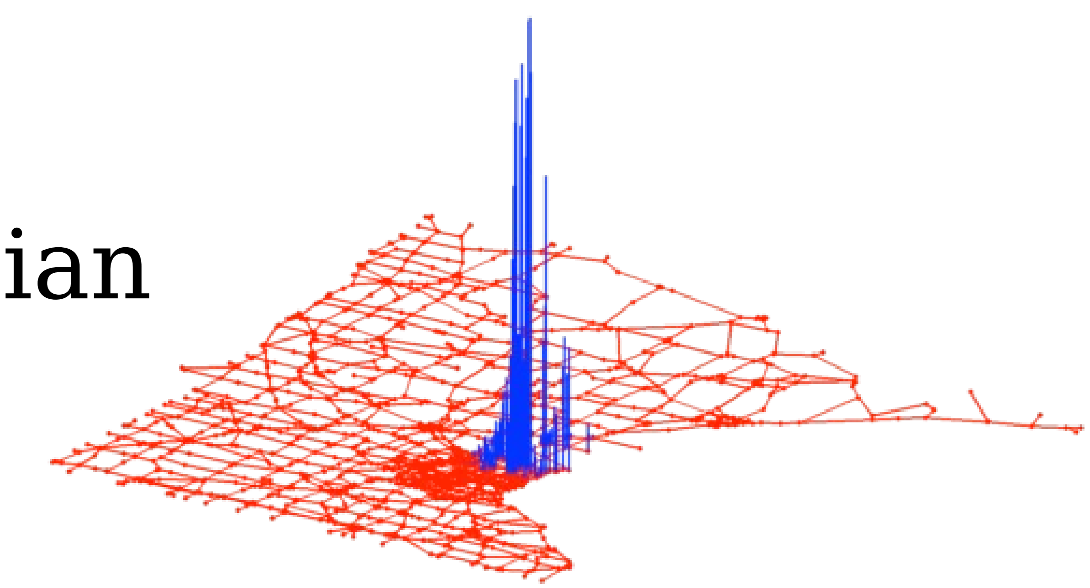
$$\nabla f(i) = [f(i+1) - f(i)]/\delta$$

Becomes

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

What about the second derivative ?...

Second derivative, the Laplacian



$$\begin{aligned} Lf(2) &= (f_3 - f_2)w_{23} - (f_2 - f_1)w_{12} \\ &= (f_3 - f_2)w_{23} + (f_1 - f_2)w_{12} \end{aligned}$$

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

Node to node space $\rightarrow$ square matrix

Graph Laplacian(s)

L : ***unnormalized*** *or* ***combinatorial Laplacian*** of G

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

1)Other definitions :
2)1) **Random walk Laplacian** (weight=probability), not symmetric

$$\mathrm{L} = \text{1-D}^{-1} W$$

1)2) **Normalized Laplacian**

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

2)3) **Directed-graph L = D-W** , but not symmetric.

Properties

$\mathbf{L}$ is real, symmetric and positive semi-definite:

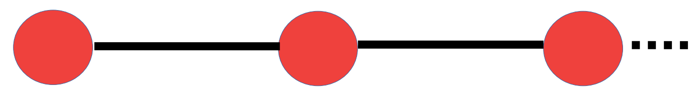
- It has an eigendecomposition into real eigenvalues and eigenvectors λ_i, u_i
- The eigenvalues are non-negative

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

$$\mathbf{L1} = 0$$

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

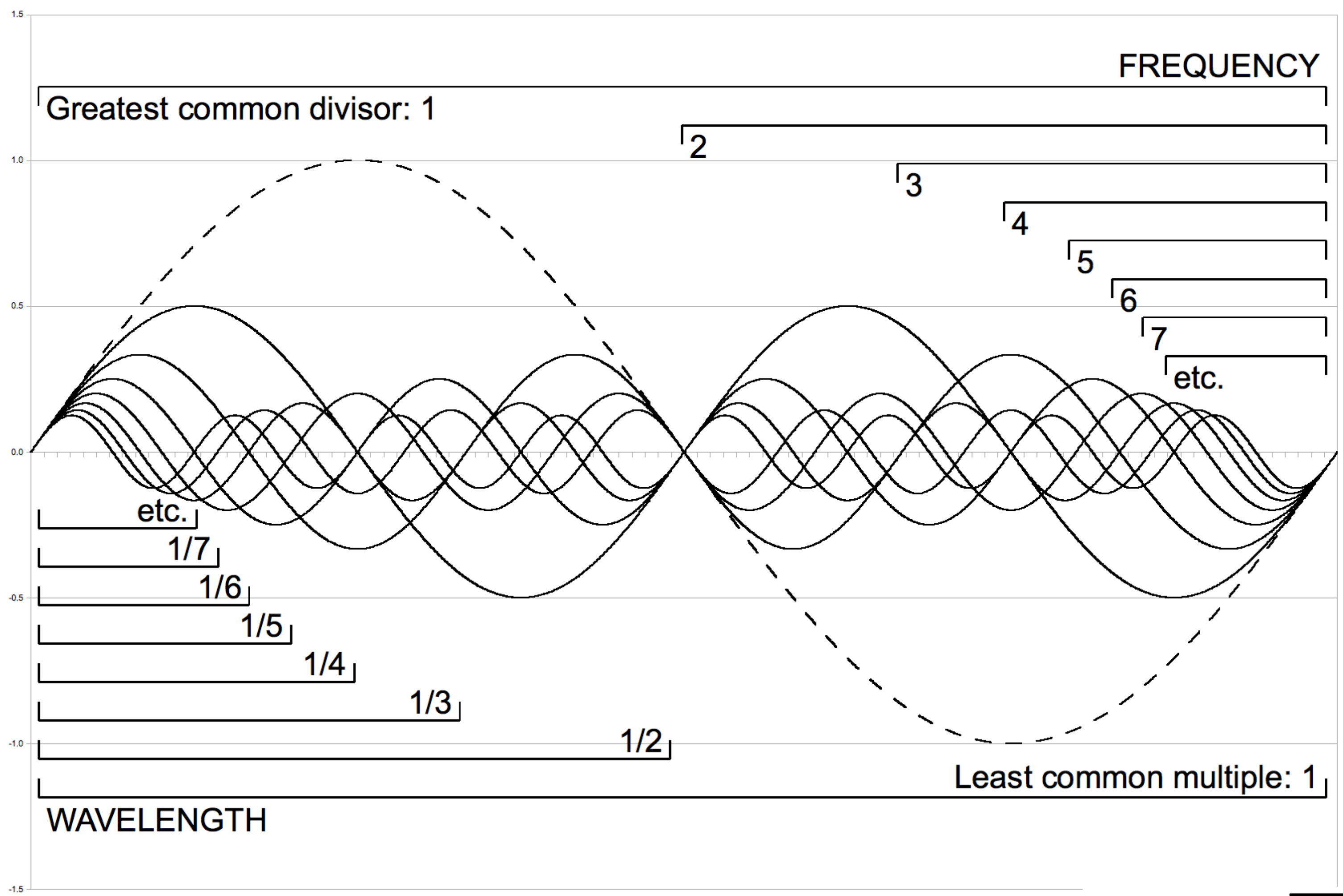
$$\lambda_k = 2 - 2\cos\frac{\pi k}{N} = 4\sin^2\frac{\pi k}{2N},\, k = 0, ..., N-1$$

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

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

Graph Fourier transform

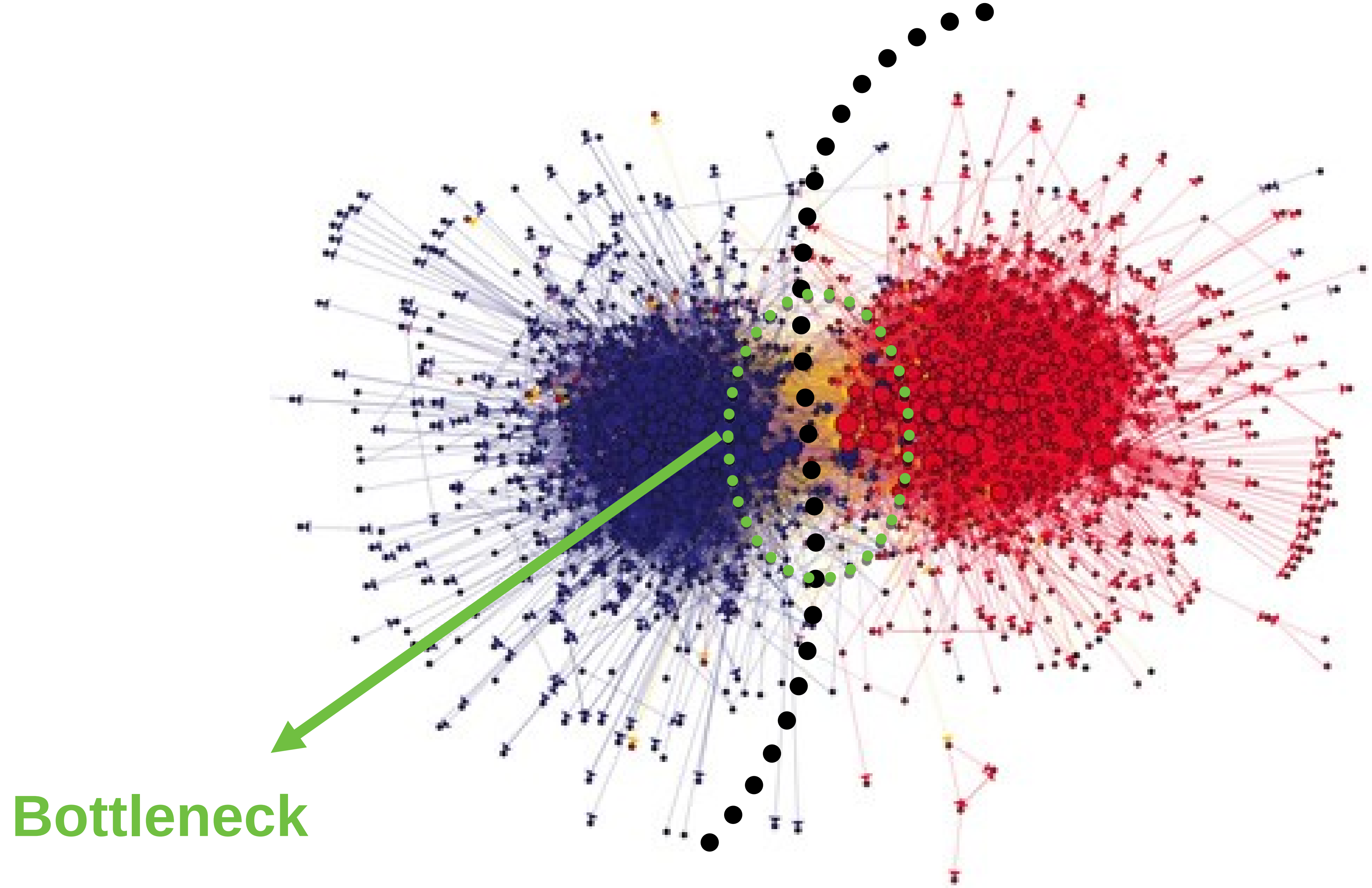
Generalization of the Fourier transform to graphs :
Eigenvectors of the Graph Laplacian = Graph Fourier modes



As smooth as possible + orthogonality

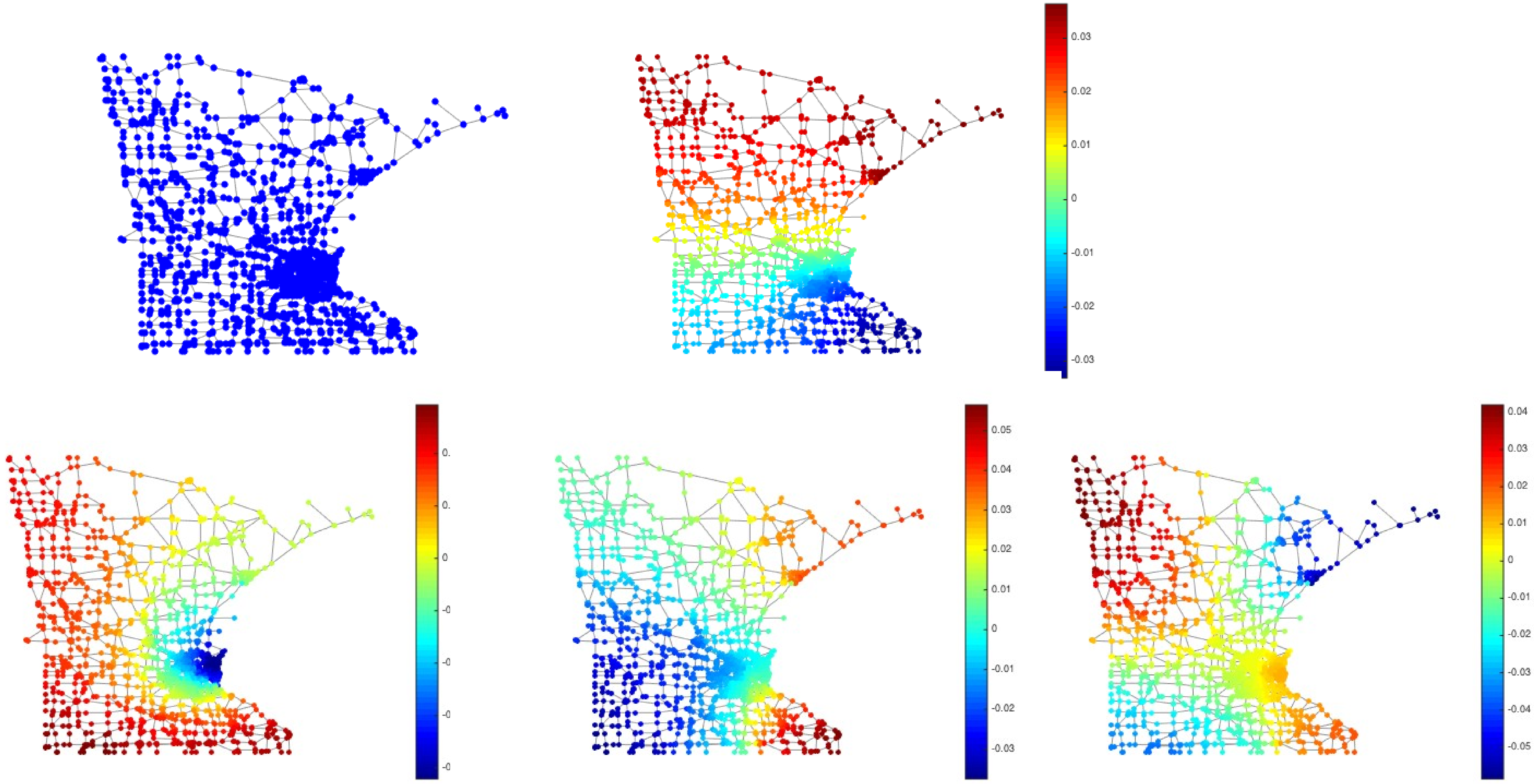
$$x^T y = \sum_i x[i]y[i]$$

The Fiedler vector

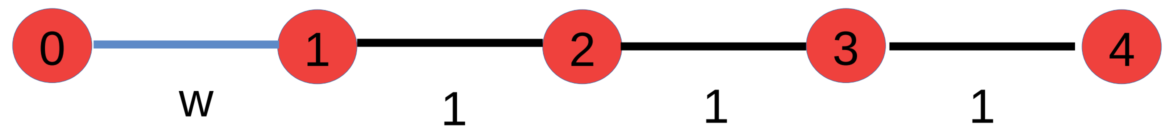


Rem. : Eigenvectors contain global information about the graph

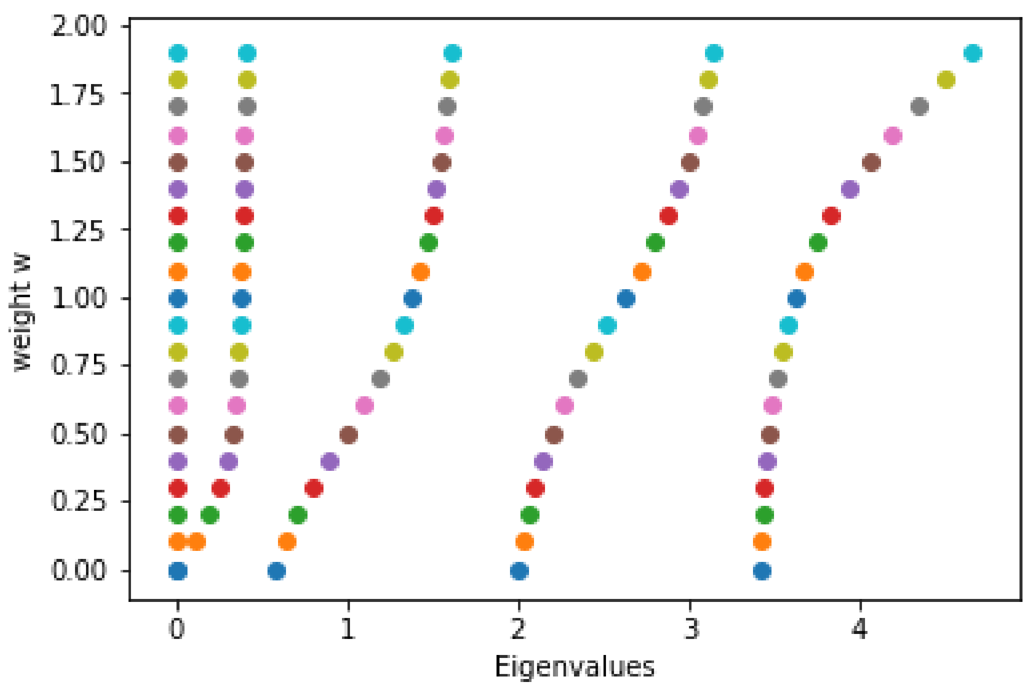
A Few Laplacian Eigenvectors

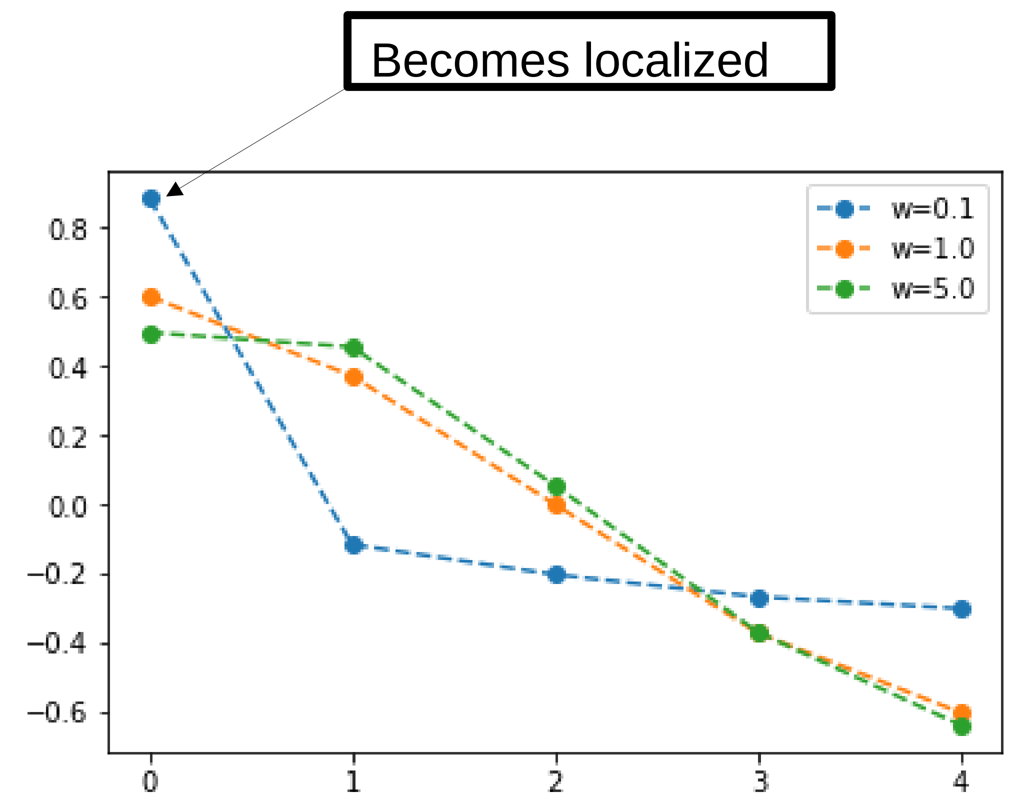


Laplacian eigenvectors



Variable weight w.
Influence on the spectral properties ?





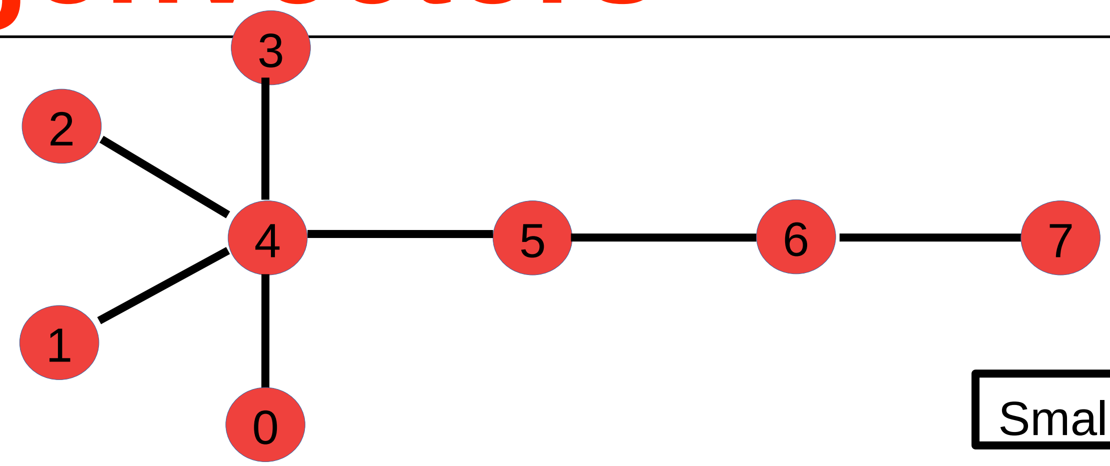
Fiedler
vector

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

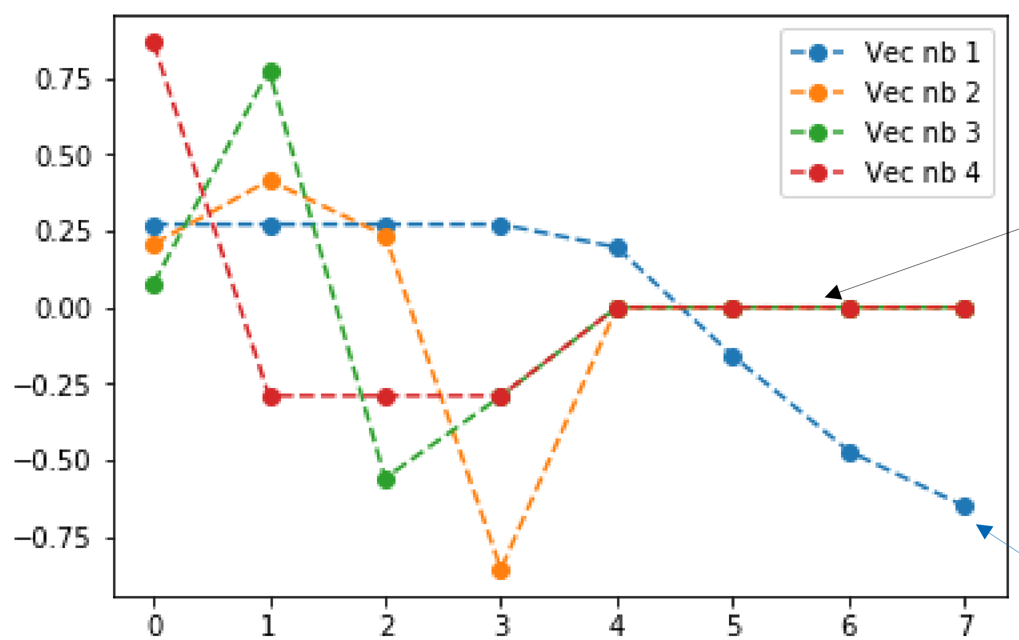
Laplacian eigenvectors

Comet graph

Inhomogeneous graph,
With a high degree node

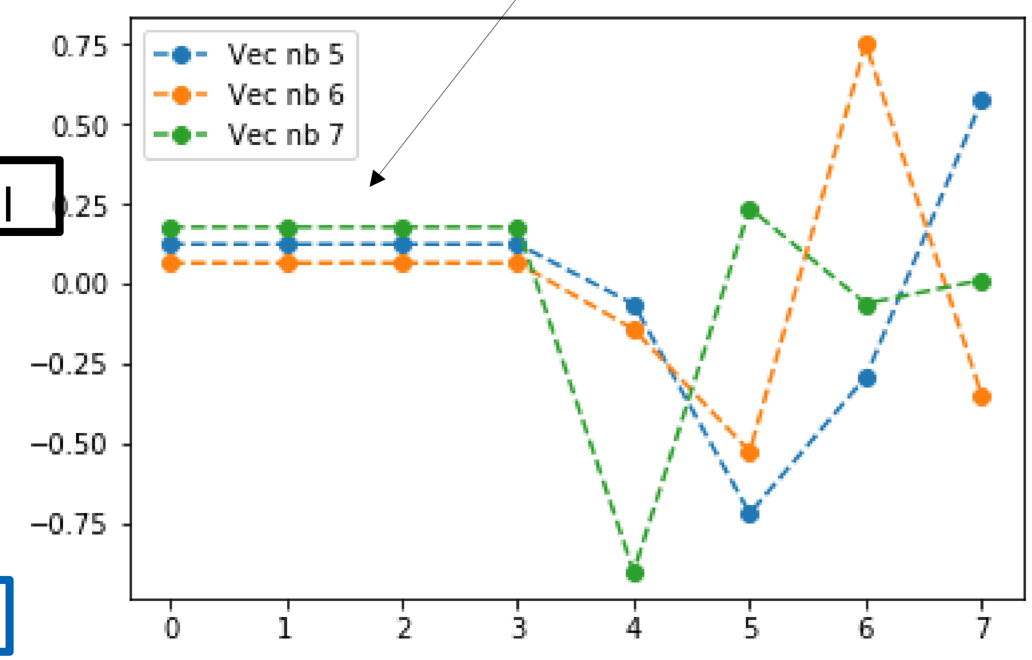


Small values on the head



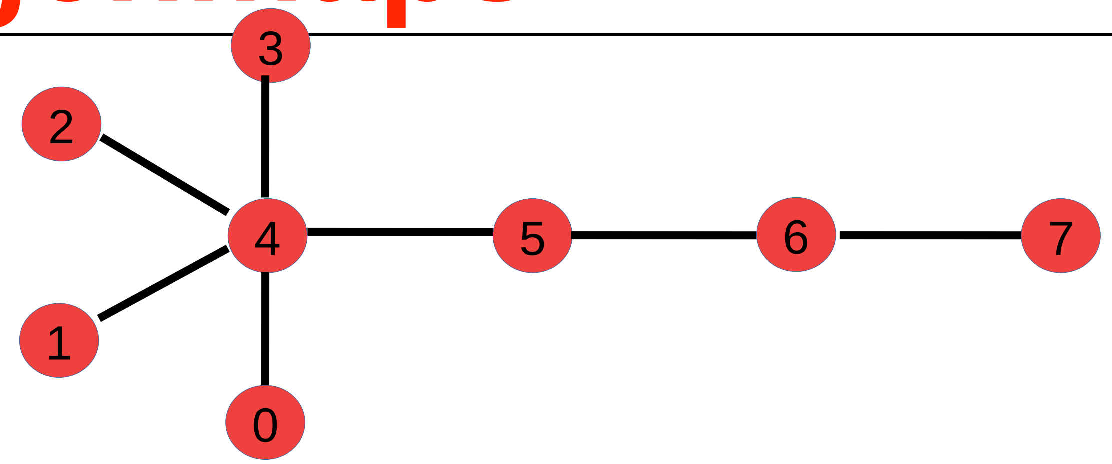
Zeros on the tail

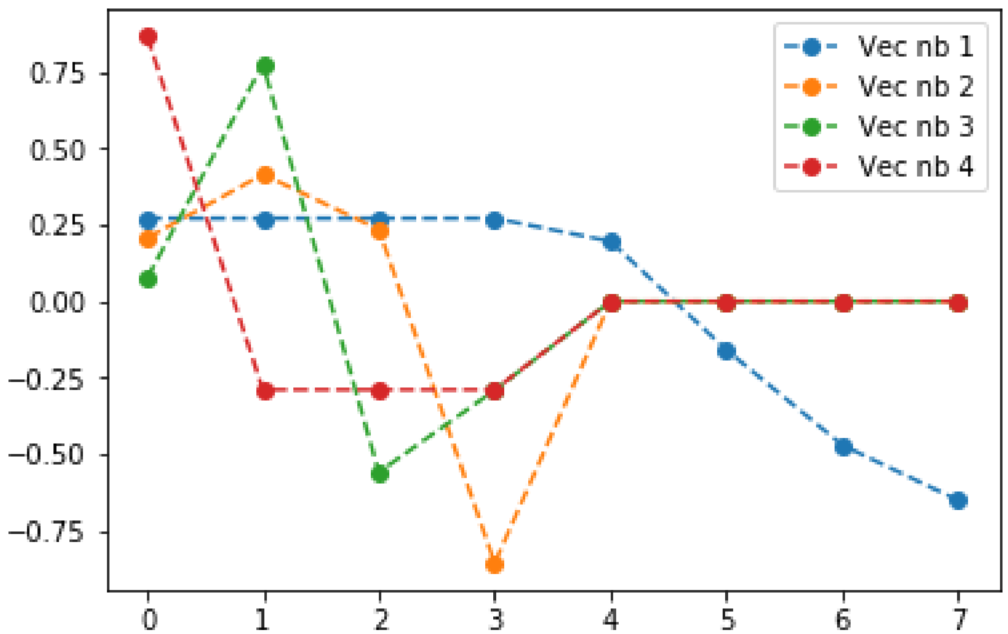
Fiedler vector

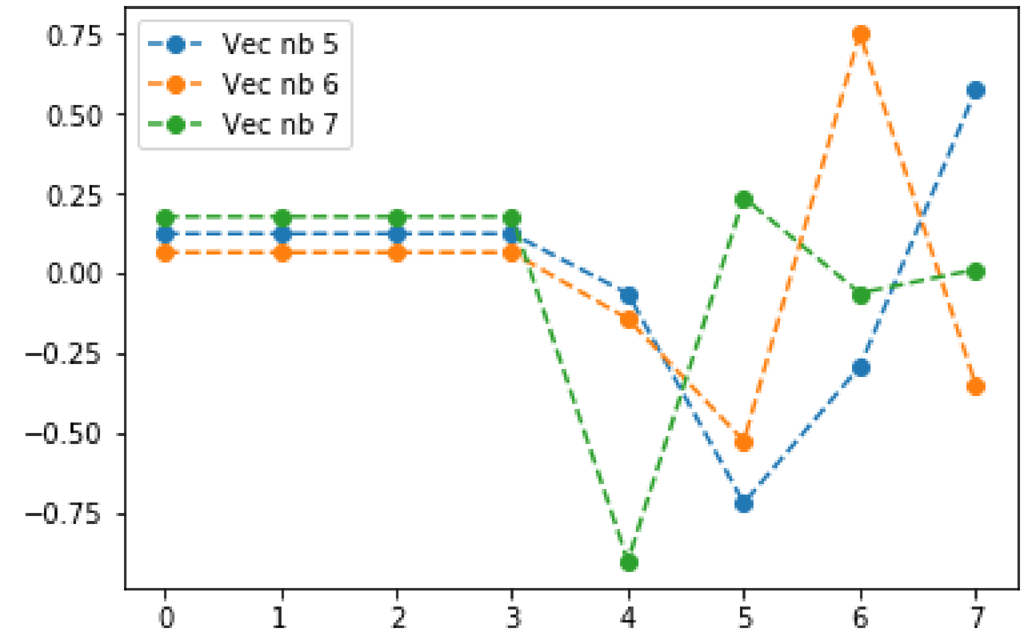


Eigenvectors localized in the different structures

Laplacian eigenmaps







Eigenvectors values as features on the nodes

- smallest eigenvalues are the most meaningful (except zero)

Applications

- Visualization with 2 dimensions “eigenmaps”
- classification