Part II: Graph signal processing

Ricaud, Benjamin, et al. "Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs." Comptes Rendus. Physique 20.5 (2019): 474-488.

Tremblay, Nicolas, et al. "Design of graph filters and filterbanks." Cooperative and Graph Signal Processing. Academic Press, 2018. 299-324.

Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." Proceedings of the IEEE 106.5 (2018): 808-828.

The Laplacian

- The Graph Laplacian (or combinatorial graph Laplacian) is:

$$L = D - W$$

One of the most important objects in spectral graph theory, graph signal processing and graph machine learning.

Proposition 1 (Properties of L) *The matrix L satisfies the following properties:*

1. *For every vector $f \in \mathbb{R}^n$ we have*

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

2. *L is symmetric and positive semi-definite.*

3. *The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$.*

4. *L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Proof.
Part (1): By the definition of d_i,

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

Part (2): The symmetry of L follows directly from the symmetry of W and D. The positive semi-definiteness is a direct consequence of Part (1), which shows that $f'Lf \geq 0$ for all $f \in \mathbb{R}^n$.

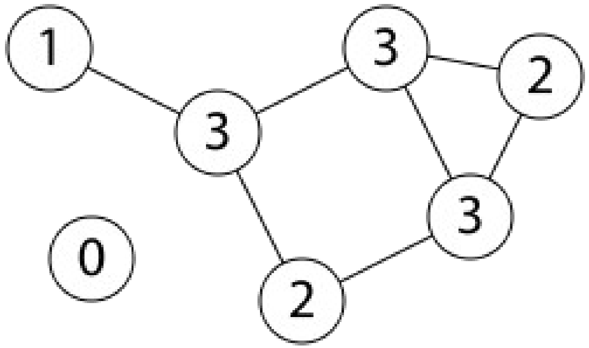
Part (3): Obvious.

Part (4) is a direct consequence of Parts (1) - (3). ☺

Multiplicity of eigenvalue 0: number of disconnected parts in the graph

Proposition 2 (Number of connected components) *Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.*

Each connected component is an independent subspace



The normalized Laplacian

- The Graph Laplacian (or combinatorial graph Laplacian) is:

$$L_N = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

Similar properties, but

- the largest eigenvalue is bounded by 2,
- the eigenvector associated to eigenvalue 0 is not constant.

$$f' L_N f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

The normalized Laplacian

$$L_N = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

- Why is it the one used in Graph neural networks?

  - The largest eigenvalue is bounded by 2,
  - Reduce the influence of hubs (highly connected nodes)

$$f' L_N f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

The Laplacian

At node i:

$$(Lf)_i = \sum_{j=1}^{n} w_{ij} (f_i - f_j)$$

The sum of differences around the node.

Why do we use this one in graph signal processing?
$\rightarrow$ we can more easily connect it to Physics and Fourier.

Graph signal processing

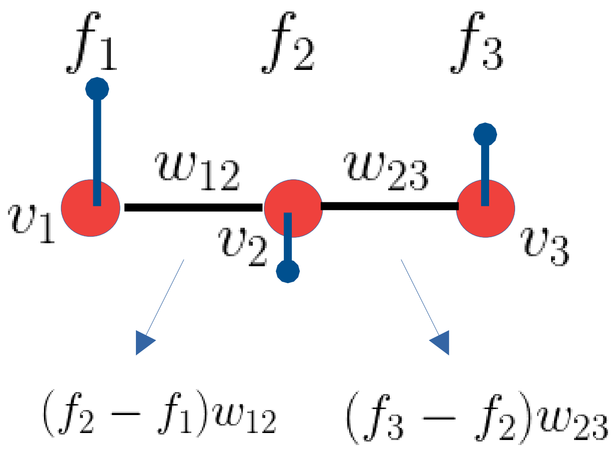
What are the basic tools for analysing functions or signals ?

Variations, derivative, gradient :

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

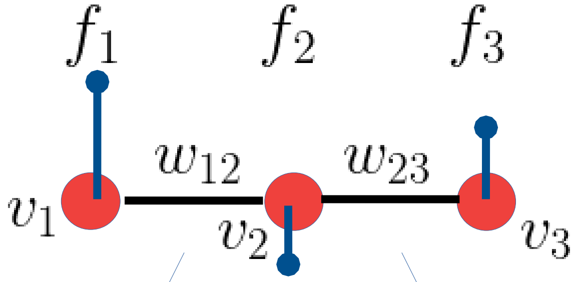
Becomes on the graph

$$\nabla f(i,j) = [f(j) - f(i)]w(i,j)$$



Values on the edges !

What are the basic tools for analysing functions or signals ?



Variations, derivative, gradient :

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

Becomes on the graph

$$\nabla f(i,j) = [f(j) - f(i)]w(i,j)$$

Values on the edges !

$(f_2 - f_1)w_{12}$ $\quad$ $(f_3 - f_2)w_{23}$

Gradient of gradient

What about the second derivative ?...

Second derivative, the Laplacian

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

Generalization of the Laplacian on a graph:

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

Sum of variations

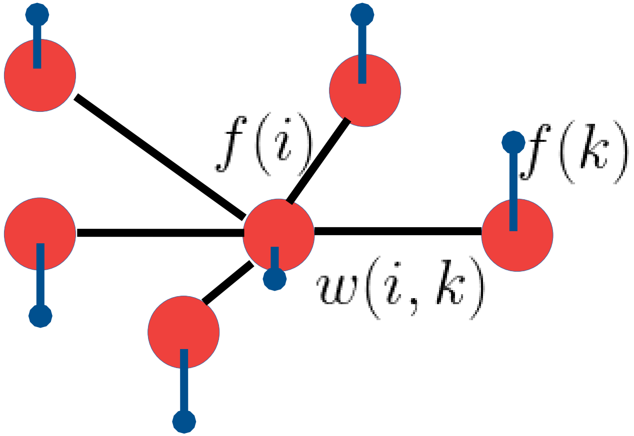
Neighbors of i

Node to node space $\rightarrow$ square matrix !

We also have: $(\nabla f, \nabla f) = (f, Lf)$

This scalar product measure the variations of a function over the graph!

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



Signal processing

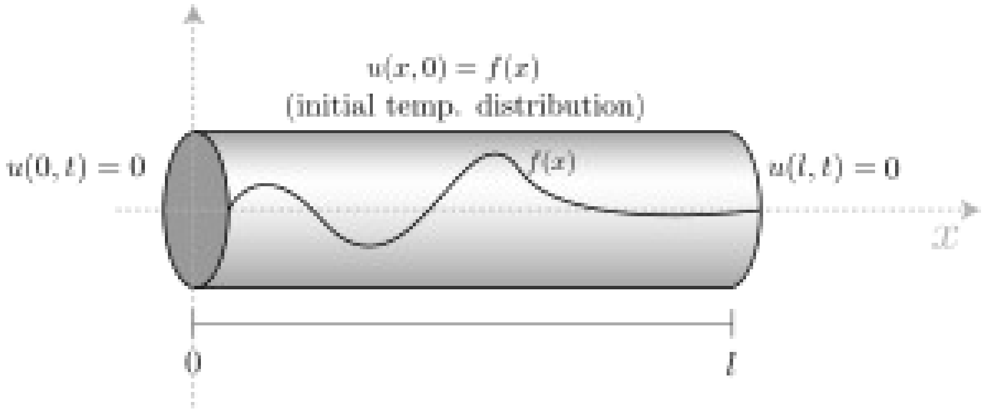
- Signal to analyze or denoise
- Filters to remove noise and keep the important information
- Linear filters
- Fourier Transform

We need a Fourier transform for graphs!

The Fourier Transformation: origin

History : Joseph Fourier (1768-1830), French mathematician.

The heat diffusion model



Wikipedia

Heat equation :

$$\frac{\partial}{\partial t} f(x,t) = -\alpha \frac{\partial^2}{\partial x^2} f(x,t) = -\alpha \Delta f(x,t)$$

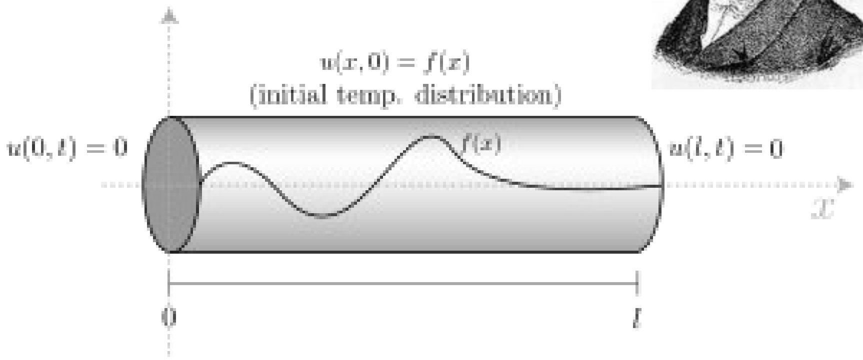
Laplacian!
(in continuous domain)

With initial conditions
$$f(x, t=0) = f_0(x)$$

Solution ?

Fourier transform

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi kx} dx$$



The Fourier transform turn the derivative into a multiplication

$$\frac{\partial}{\partial t} f(x,t) = -\alpha \frac{\partial^2}{\partial x^2} f(x,t) \qquad \frac{\partial}{\partial t} F(k,t) = -\alpha \lambda_k F(k,t)$$

$\rightarrow$ The Fourier transform diagonalize the Laplacian $\lambda_k = -4\pi^2 k^2$

Solution of the heat equation:

$$F(k,t) = e^{-\alpha \lambda_k t} F(k,0)$$

Note: different convention for the Laplacian $L = -\Delta$

$$\frac{\partial}{\partial t} f(x,t) = -\alpha \Delta f(x,t)$$

Solution of the heat equation:

$$F(k,t) = e^{-\alpha \lambda_k t} F(k,0) \,, \qquad f(x,t) = e^{-\alpha t \Delta} f(x,0)$$

Explicitly: $$f(x,t) = \int e^{-\alpha t \lambda_k} \underbrace{\int f(y,0) e^{-i2\pi k y} dy}_{F(k,0)} \; e^{i2\pi k x} dk$$

Solving in the fourier domain is much easier

$$F(k,t) = e^{-\alpha \lambda_k t} F(k,0)$$ ← High values of λ_k will be attenuated faster

Fourier Transformation & Laplacian

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi kx} dx$$

Fourier modes $\psi_k(x) = e^{-2i\pi kx}$

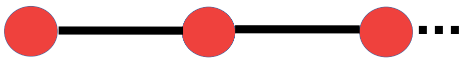
$$\Delta e^{-2i\pi kx} = -4\pi^2 k^2 e^{-2i\pi kx}$$

$$\lambda_k = -4\pi^2 k^2$$

Eigenvectors of the Laplacian !

Note : these are «generalized» eigenvectors

Discrete settings

Path graph  DCT II transform

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

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

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

Discrete Fourier transform

Ring graph

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

Eigendecomposition :

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

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

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

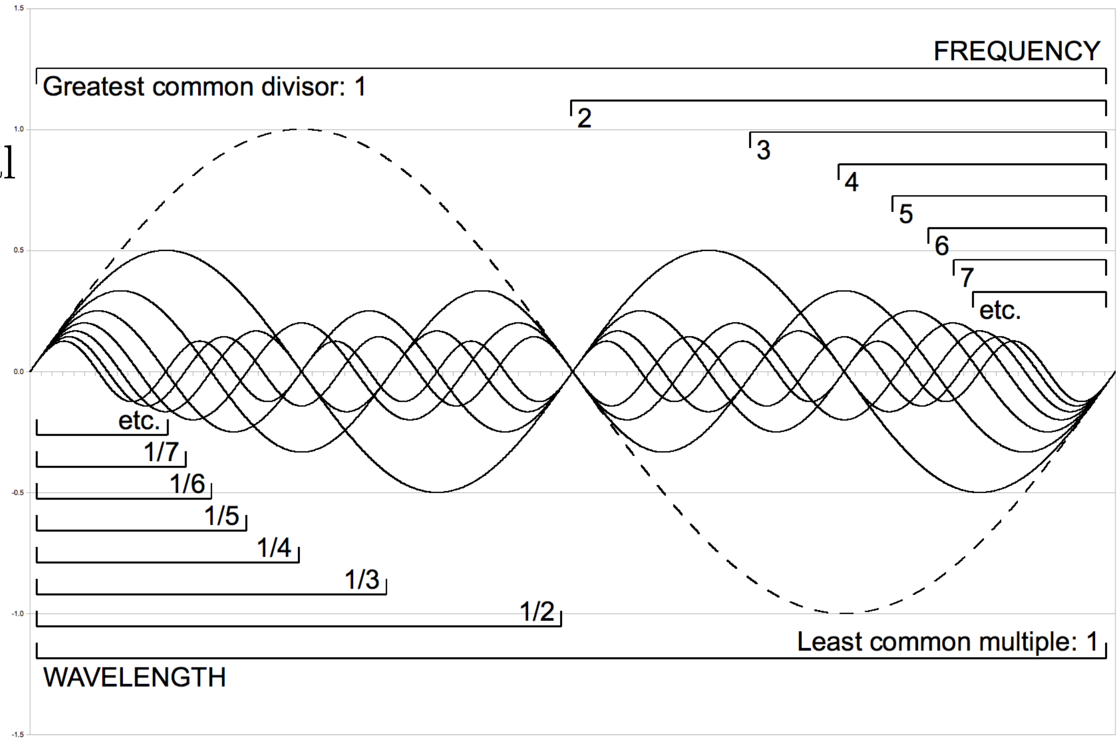
The Laplacian eigenvectors

- The Laplacian "measure" the smoothness $s = \dfrac{(f, Lf)}{\|f\|^2}$
- **Courant–Fischer–Weyl min-max principle**:

$$\lambda_k = \max_{V \subset \mathbb{R}^n, \dim V = n-k+1} \ \min_{f \in V, \|f\|=1} (f, Lf)$$

Laplacian eigenvectors :
as smooth as possible but orthogonal

Smoothest: constant $\rightarrow \lambda_1 = 0$



From Spielman, p.25:

We now prove the Spectral Theorem by generalizing this characterization to all of the eigenvalues of $\boldsymbol{M}$. The idea is to always use Theorem 2.2.1 to show that a vector is an eigenvector. To do this, we must modify the matrix for each vector.

Theorem 2.2.2. *Let $\boldsymbol{M}$ be an n-dimensional real symmetric matrix. There exist numbers $\mu_1, \ldots, \mu_n$ and orthonormal vectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ such that $\boldsymbol{M}\boldsymbol{\psi}_i = \mu_i \boldsymbol{\psi}_i$. Moreover,*

$$\boldsymbol{\psi}_1 \in \arg \max_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x},$$

and for $2 \leq i \leq n$

$$\boldsymbol{\psi}_i \in \arg \max_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \textit{for } j < i}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}. \tag{2.2}$$

Similarly,

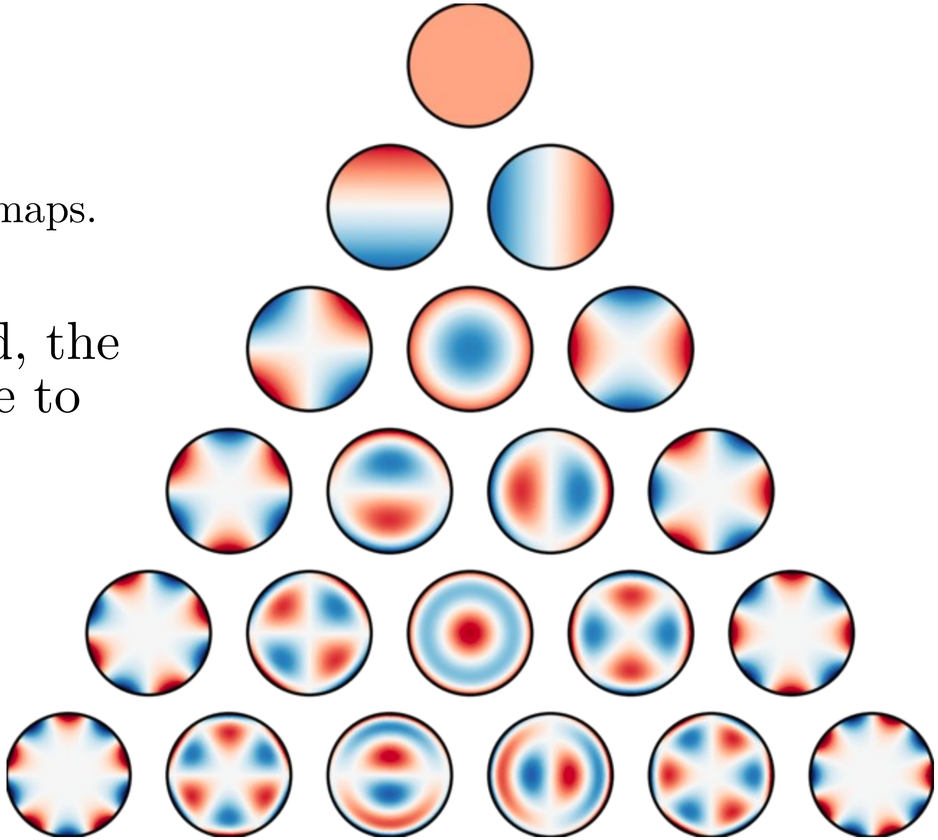
$$\boldsymbol{\psi}_i \in \arg \min_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \textit{for } j > i}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}.$$

- On the sphere: Laplacian eigenvectors are spherical harmonics. They are oscillating

$$\Delta f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial f}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial\varphi^2},$$

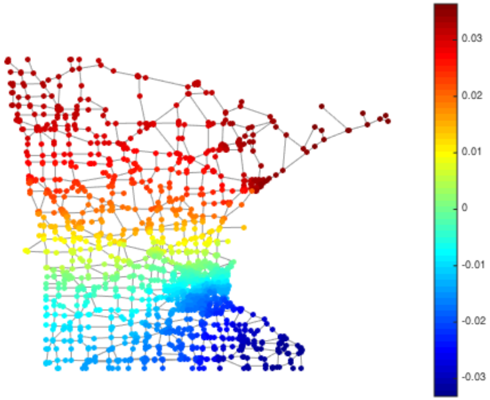
Belkin, M., & Niyogi, P. (2006). Convergence of Laplacian eigenmaps. Neurips.

For a discretized manifold, the Graph Laplacian converge to the Laplacian on the continuous manifold

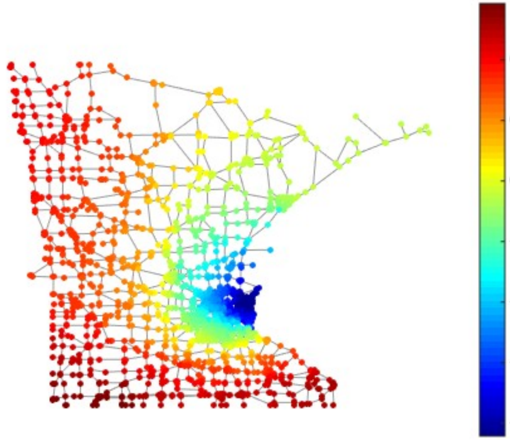


- On a graph: Laplacian eigenvectors are (more or less) oscillating

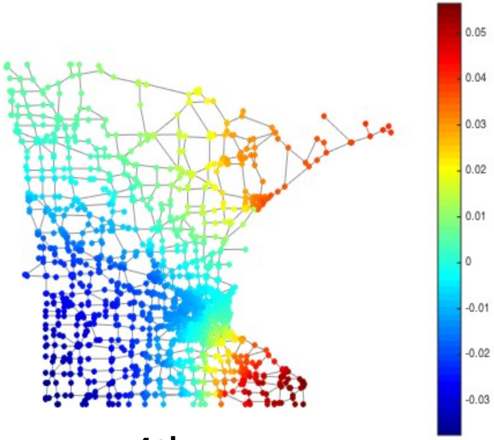
1st



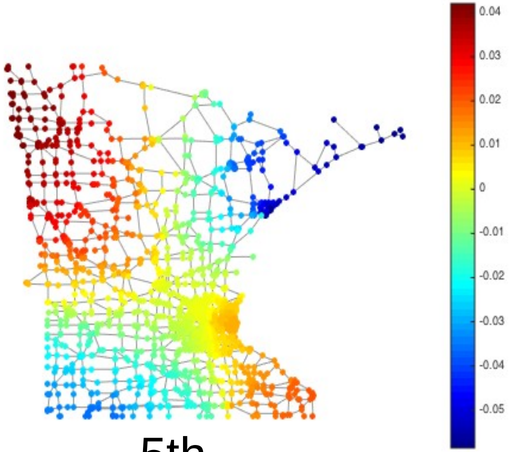
2nd



3rd

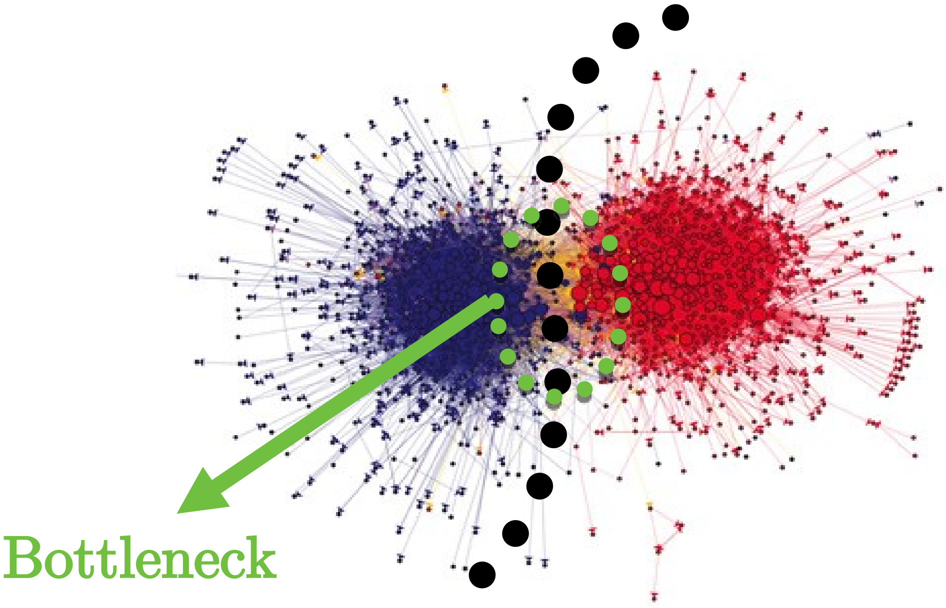


4th



5th

The Fiedler vector

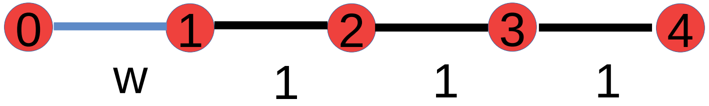


The Fiedler vector is the first eigenvector of L with non-zero eigenvalue
Interesting for cutting the graph in 2 !
It changes sign at the bottleneck (less costly in term of variations)
Remark: Eigenvectors contain global information about the graph

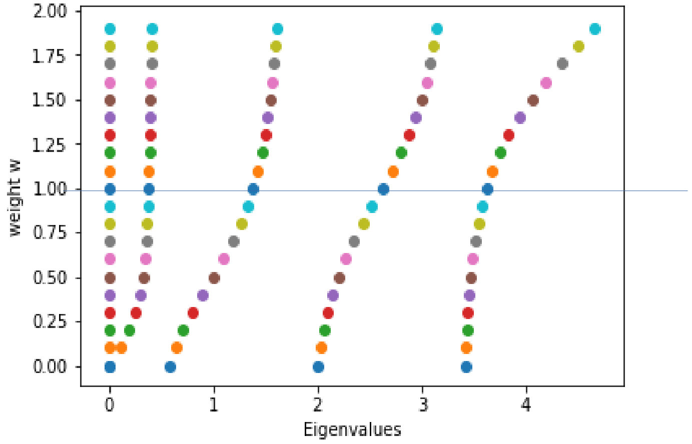
Break

- Let us compute some Graph Fourier modes.
- Make a path graph of 5 nodes.
- Diagonalize the Laplacian
- Plot the eigenvalues and visualize the value of some of the eigenvectors on the graph.
- Modify the weight between node 0 and node 1, plot the eigenvalues and eigenvectors and compare the results.
- Do the same for the comet graph with A:

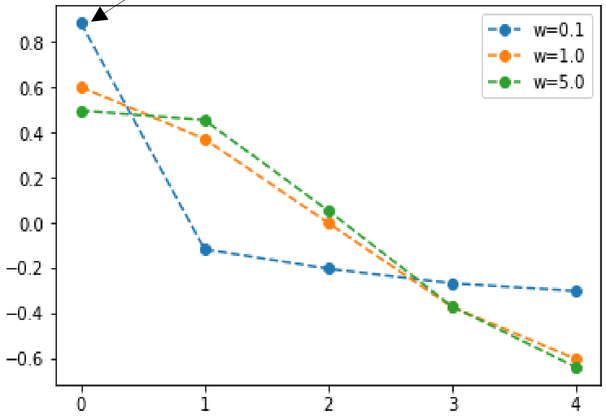
Fourier modes on a graph



Variable weight w.
Influence on the spectral properties ?



Becomes localized



Fiedler vector

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

Break

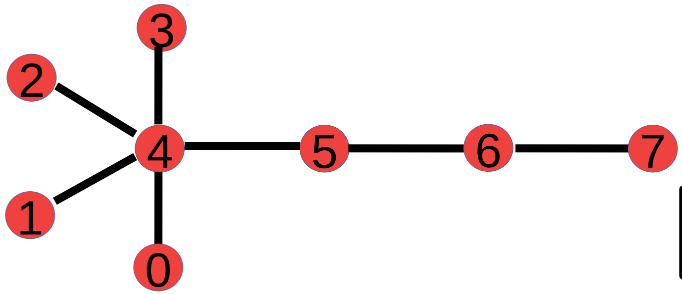
- Let us compute some Graph Fourier modes.
- Do the same for the comet graph with A:

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

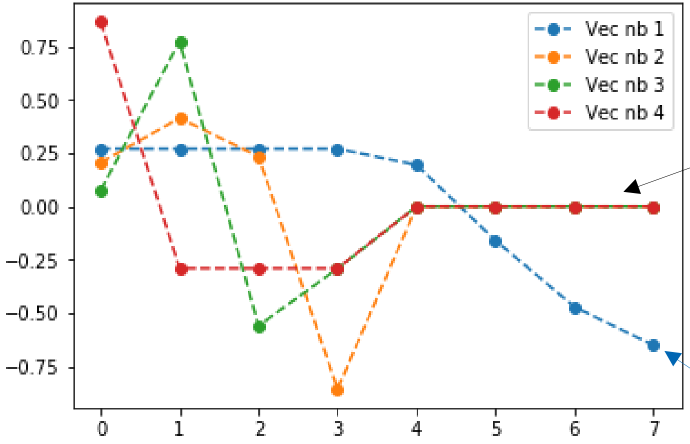
Fourier modes on a graph

Comet graph

Inhomogeneous graph,
With a high degree node

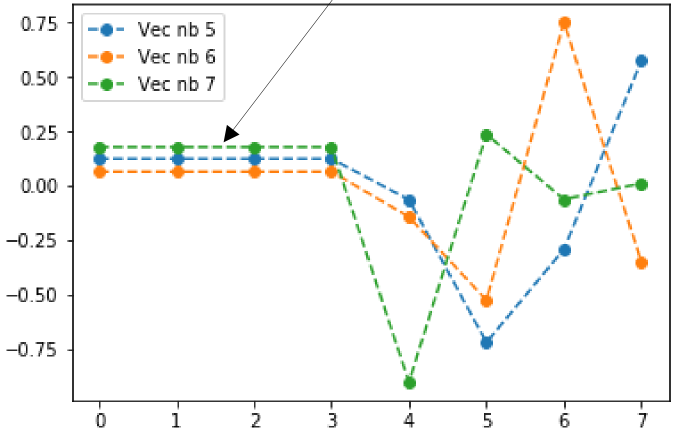


Small values on the head



Zeros on the tail

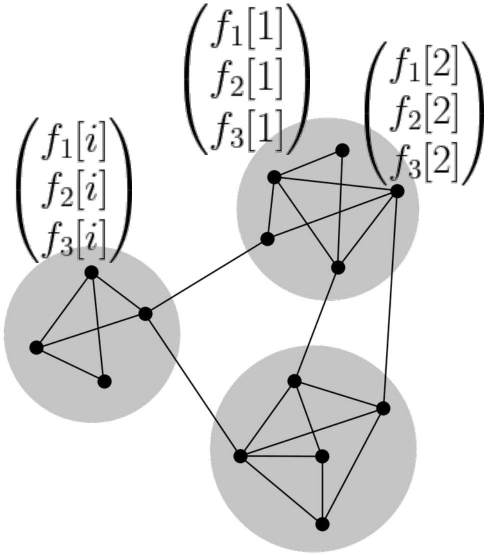
Fiedler vector



Eigenvectors localized in the different structures

Spectral clustering

Clustering a graph using the Laplacian
- For 2 clusters: cluster with the sign of the Fiedler vector
- For more than 2 clusters: cluster using several Laplacian eigenvectors, not just one



- At each node, the values of k eigenvectors of L provides a feature vector of k values. It can be clustered using standard clustering methods (e.g. k-means).

- Laplacian eigenvectors have smooth variations over the graph: neighbors have similar values, useful for clustering!

Spectral clustering

See scikit-learn “spectral clustering”:
https://scikit-learn.org/stable/modules/clustering.html#spectral-clustering

Or

U. von Luxburg, A Tutorial on Spectral Clustering