Graph signal processing

Concepts, tools and applications in neuroscience

Xiaowen Dong
Oxford-Man Institute
Department of Engineering Science

NISOx, Big Data Institute, February 2019

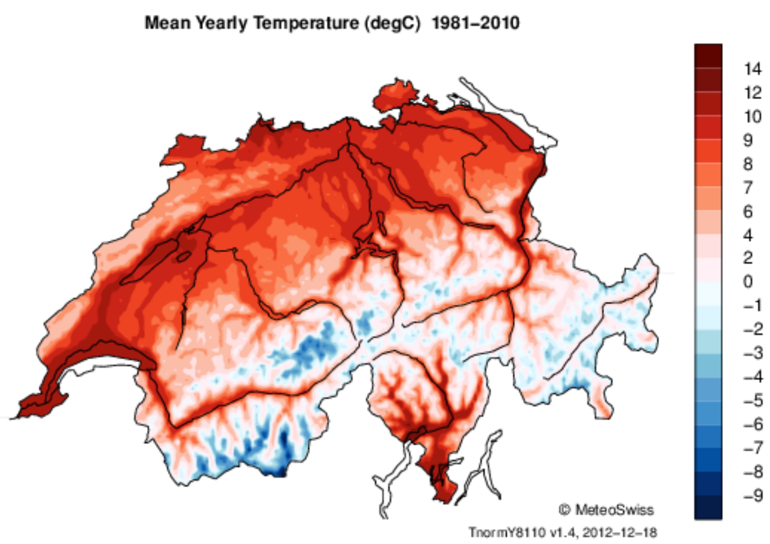
Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Outline

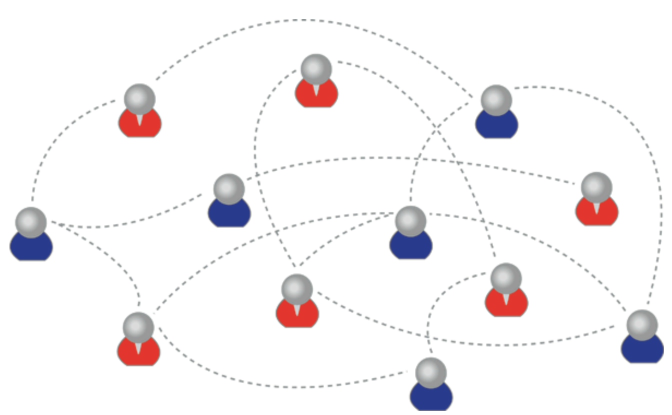
- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Data are often structured



Temperature data

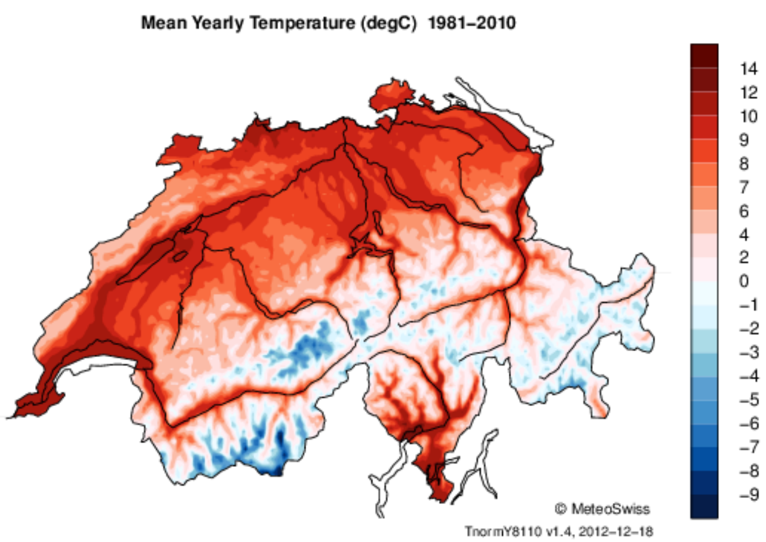
Traffic data



Social network data

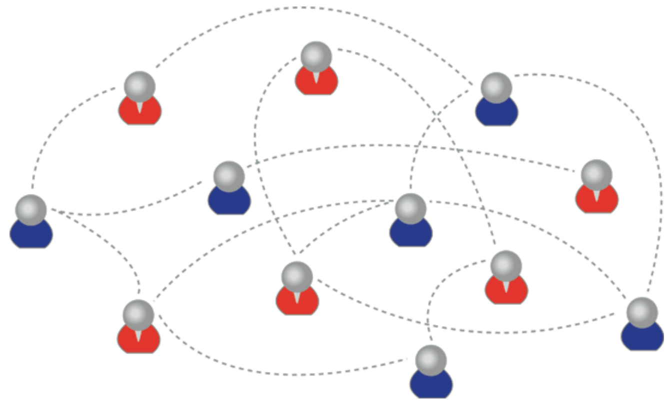
Neuroimaging data

Data are often structured



Temperature data

Traffic data



Social network data

Neuroimaging data

We need to take into account the structure behind the data

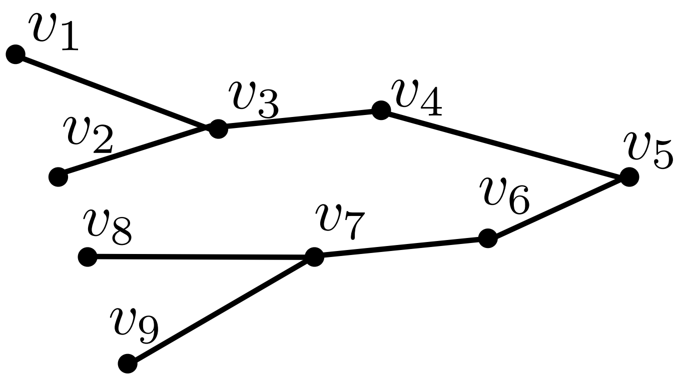
Graphs are appealing tools

- Efficient representations for pairwise relations between entities

The Königsberg Bridge Problem
[Leonhard Euler, 1736]

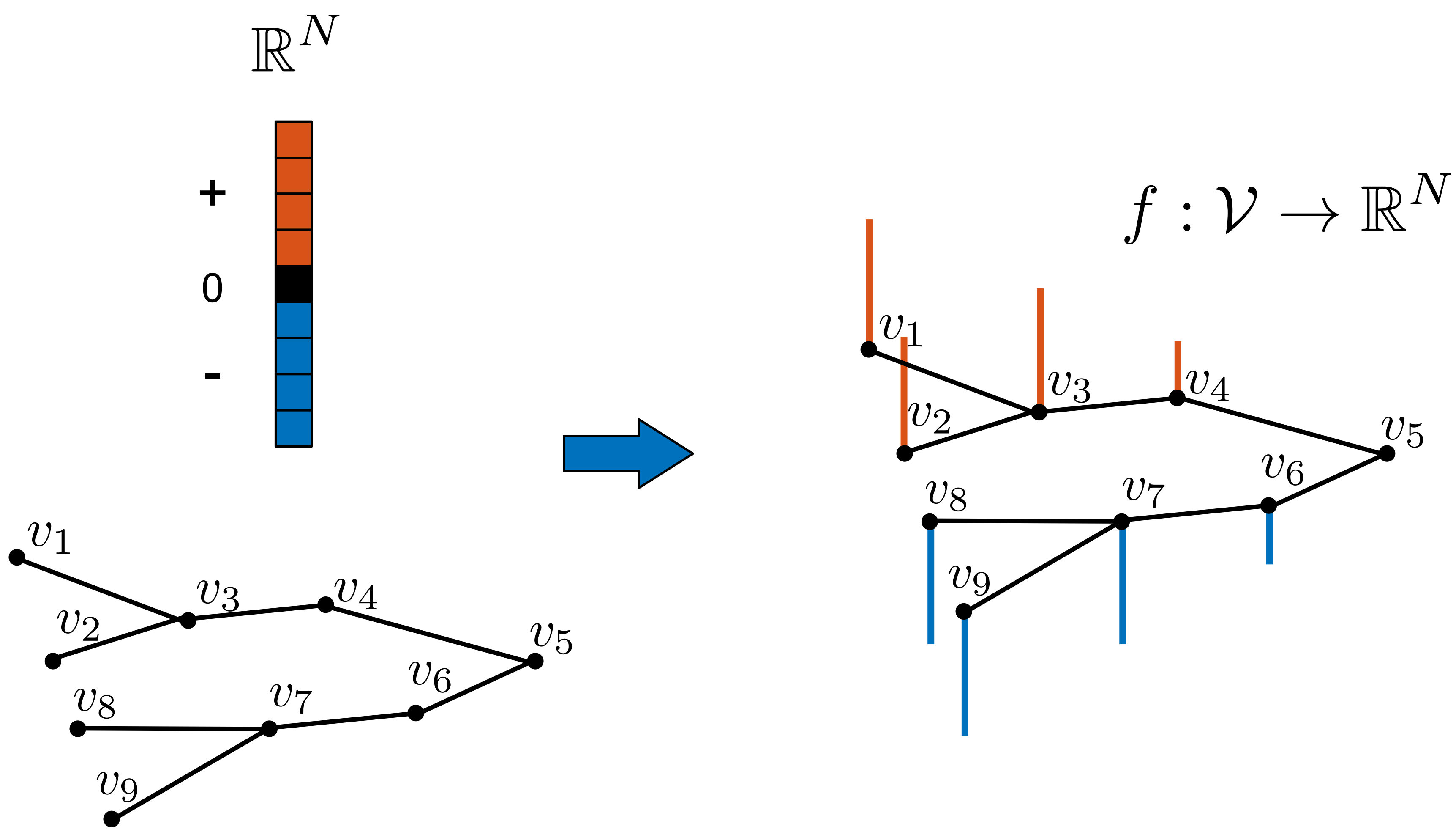
Graphs are appealing tools

- Efficient representations for pairwise relations between entities



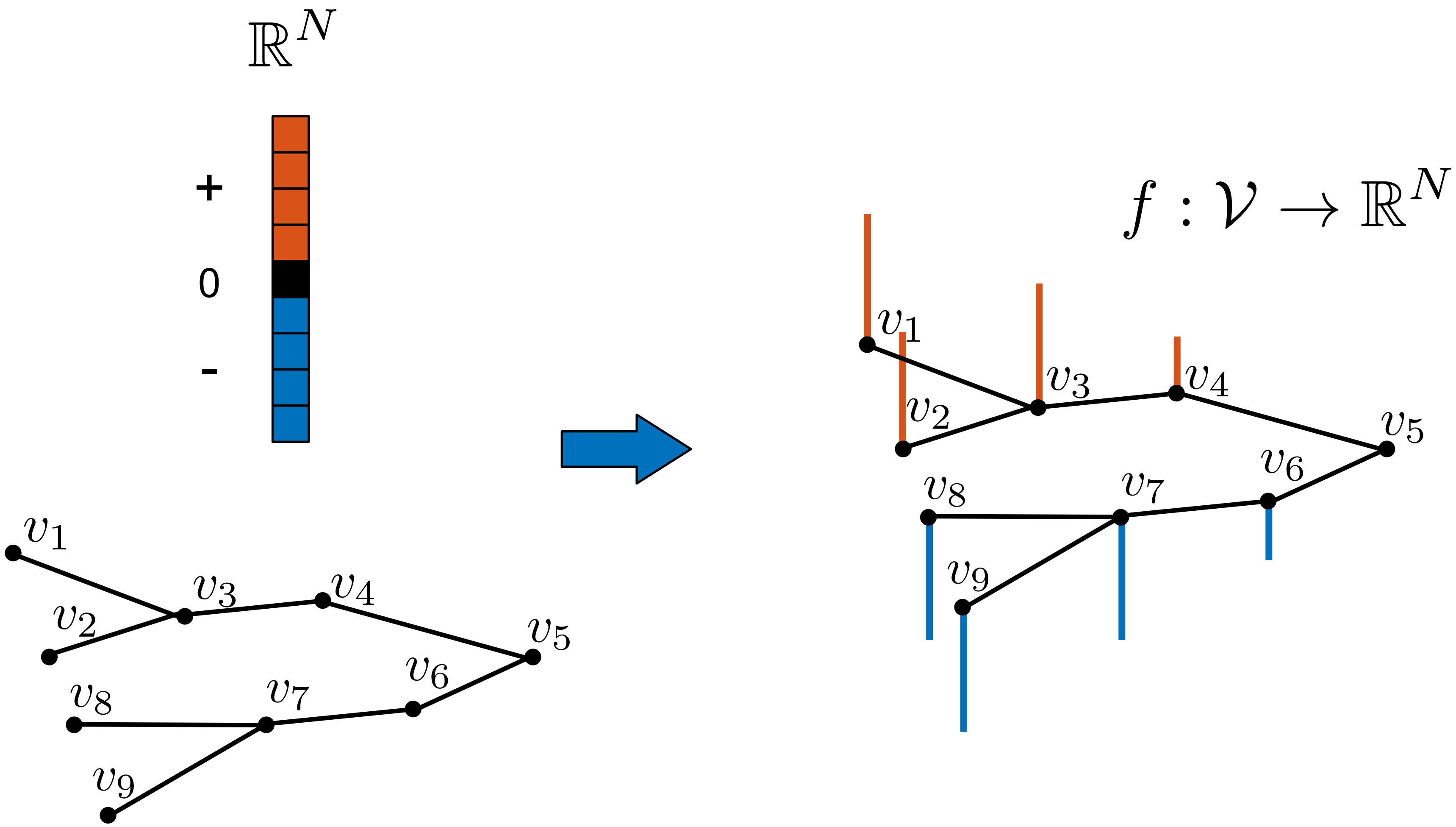
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by graph signals



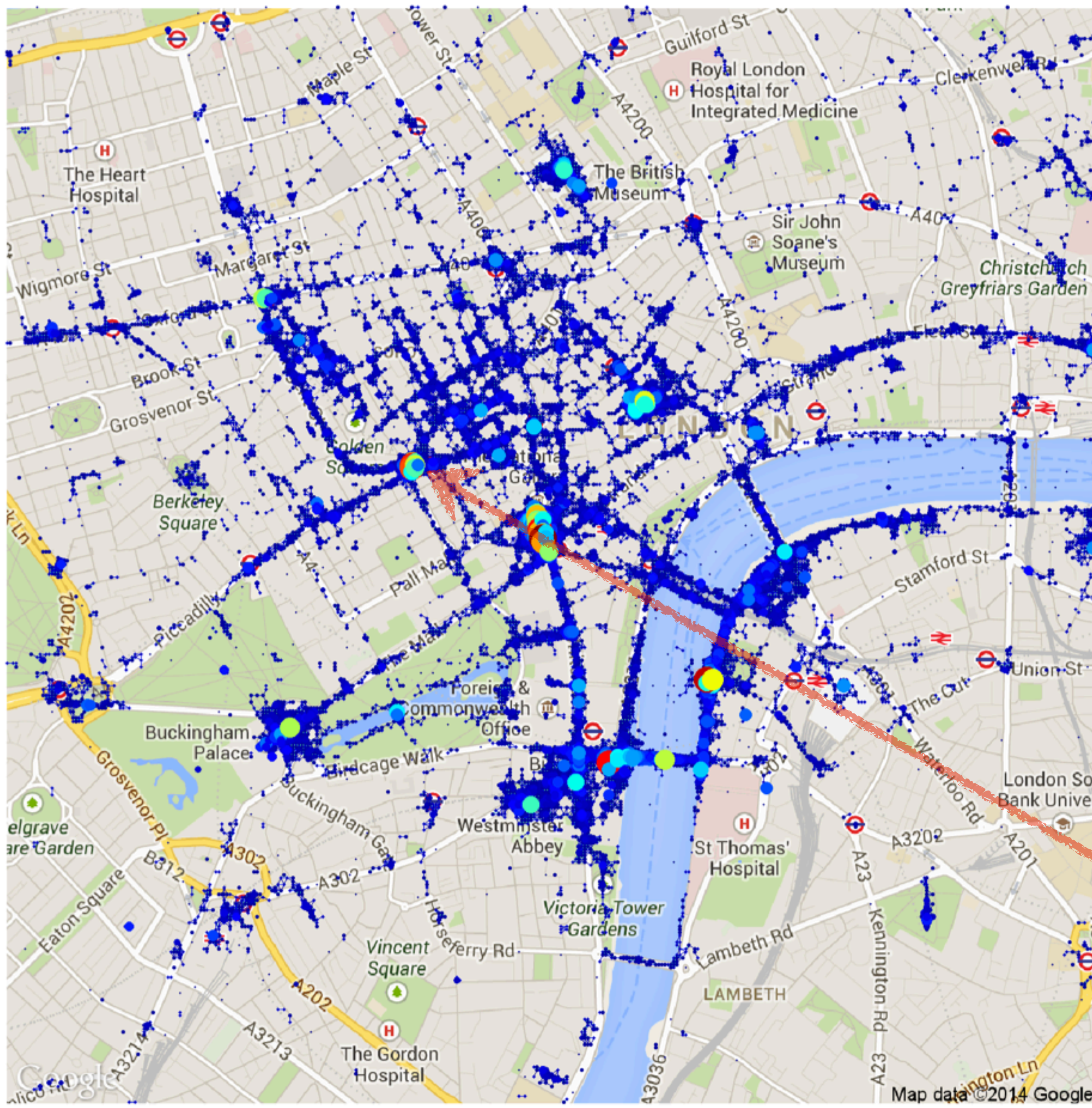
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by graph signals

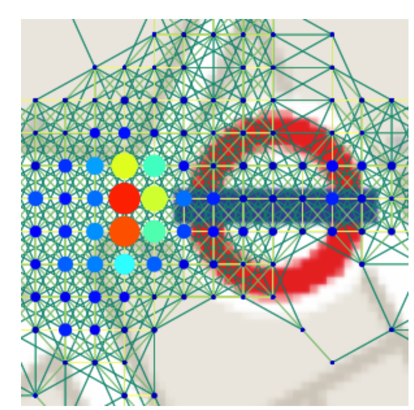


Takes into account both structure (edges) and data (values at vertices)

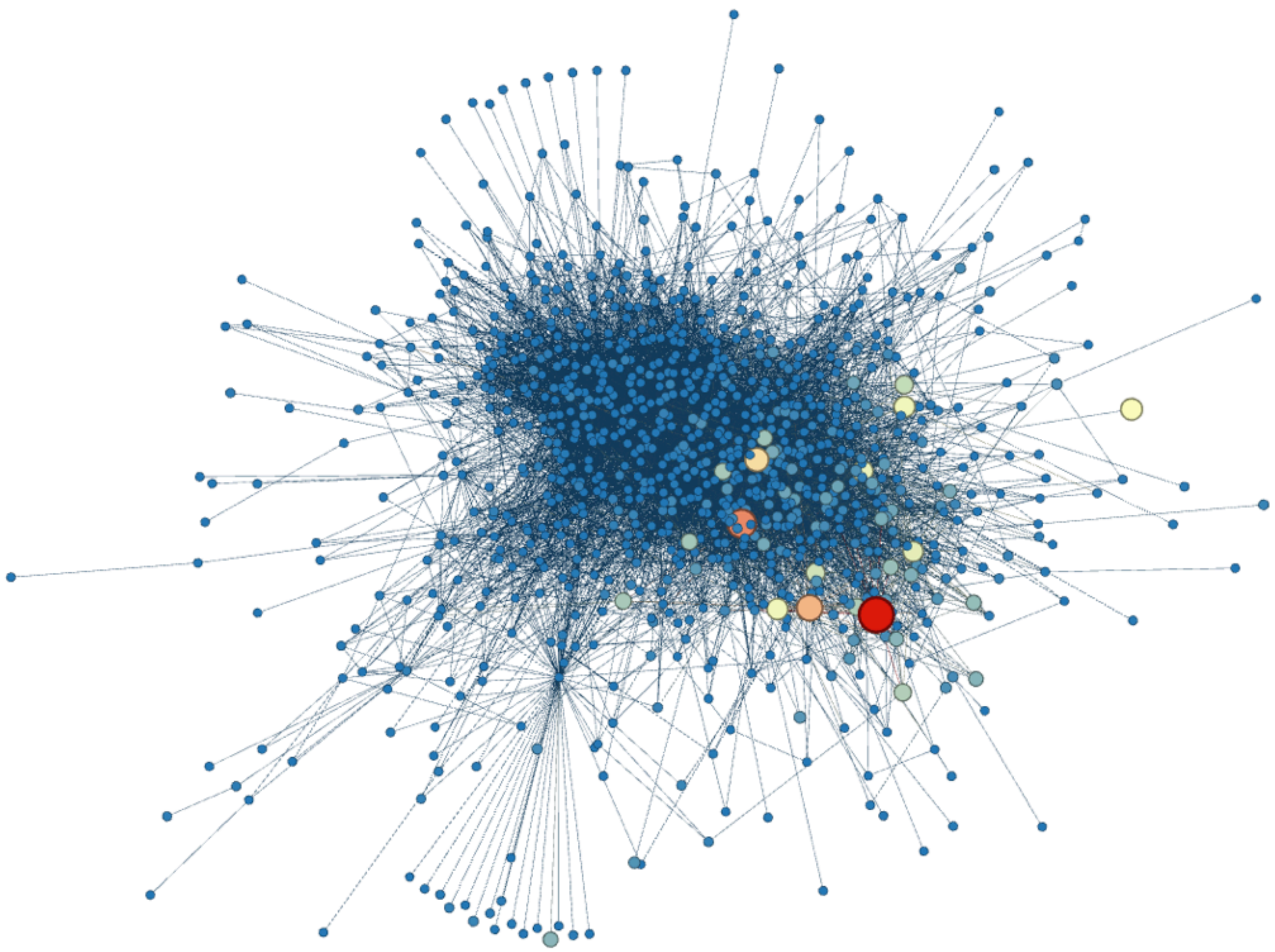
Graph signals are pervasive



- Vertices:
  - 9000 grid cells in London
- Edges:
  - geographical proximity of grid cells
- Signal:
  - # Flickr users who have taken photos in two and a half year

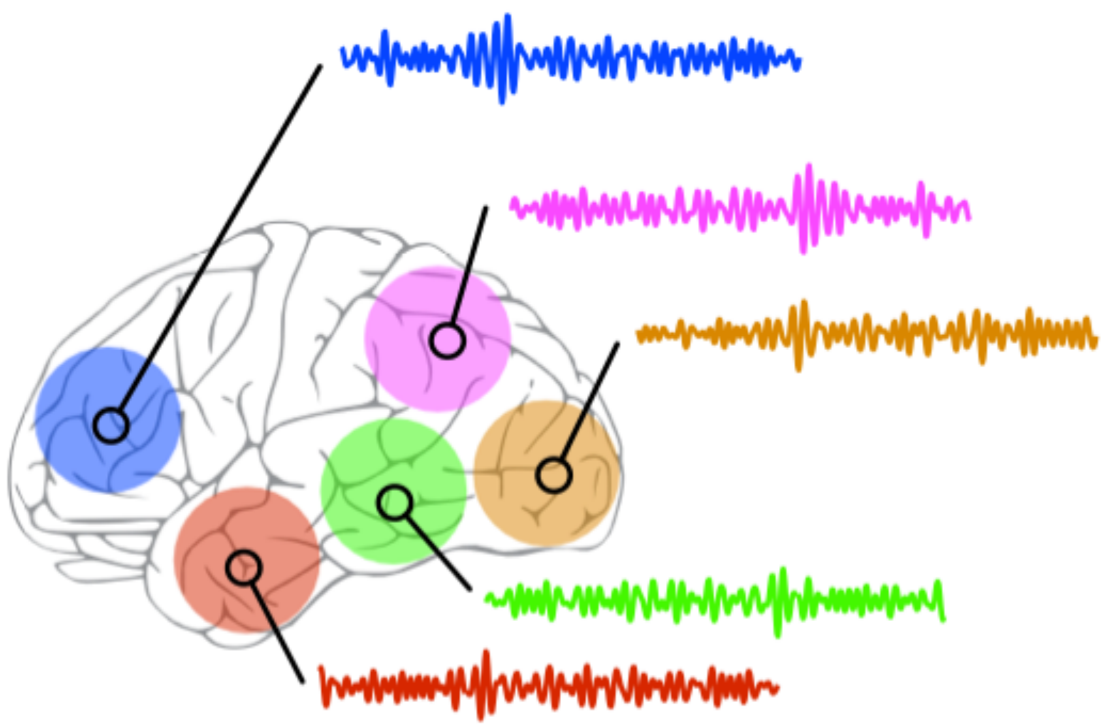


Graph signals are pervasive



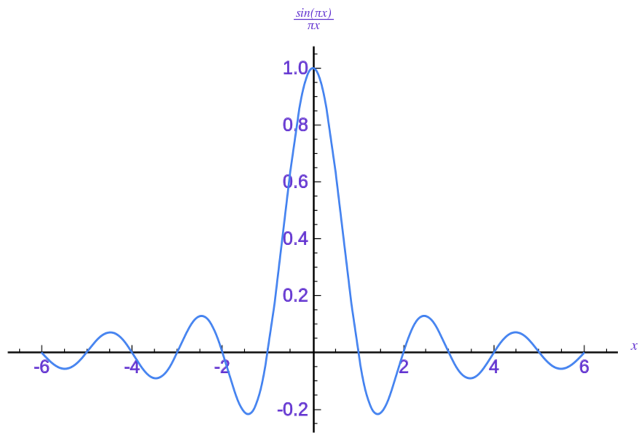
- Vertices:
  - 1000 Twitter users
- Edges:
  - following relationship among users
- Signal:
  - # Apple-related hashtags they have posted in six weeks

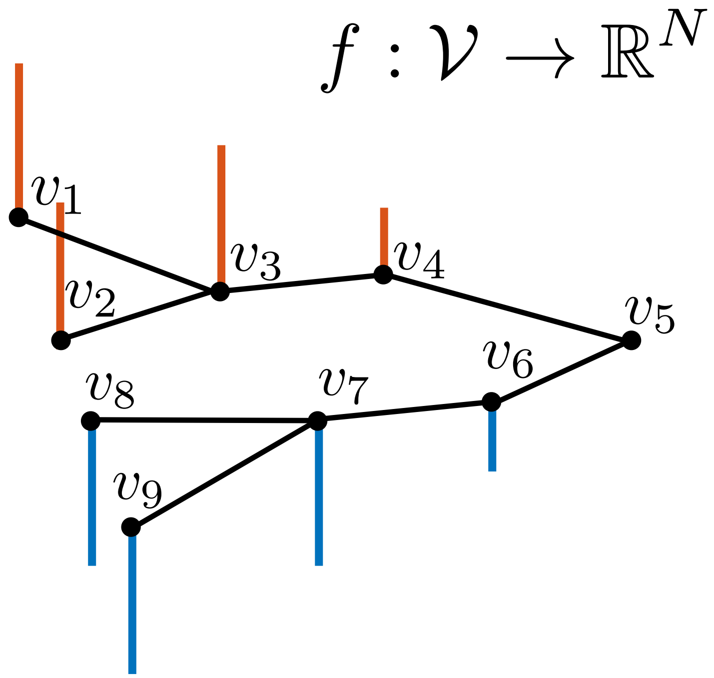
Graph signals are pervasive



- Vertices:
  - brain regions
- Edges:
  - structural connectivity between brain regions
- Signal:
  - blood-oxygen-level-dependent (BOLD) time series

Research challenges

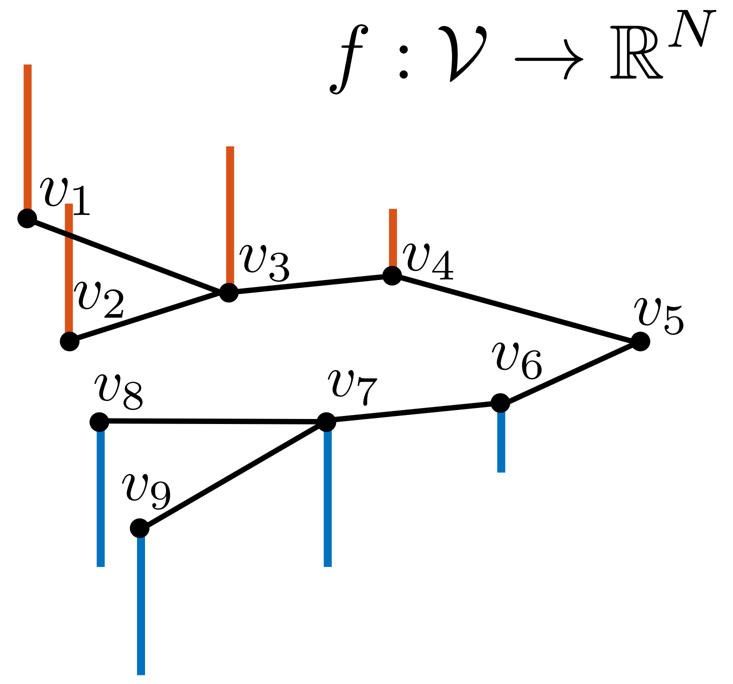




How to generalise classical signal processing tools on irregular domains such as graphs?

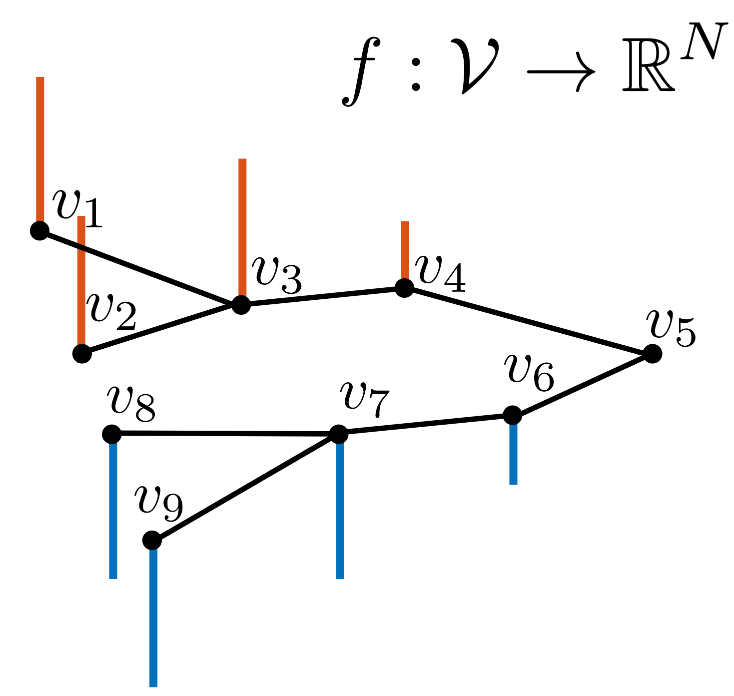
Graph signal processing

- Graph signals provide a nice compact format to encode structure within data
- Generalisation of classical signal processing tools can greatly benefit analysis of such data
- Numerous applications: Transportation, biomedical, social network analysis, etc.



Graph signal processing

- Graph signals provide a nice compact format to encode structure within data
- Generalisation of classical signal processing tools can greatly benefit analysis of such data
- Numerous applications: Transportation, biomedical, social network analysis, etc.
- An increasingly rich literature
  - classical signal processing
  - algebraic and spectral graph theory
  - computational harmonic analysis
  - machine learning

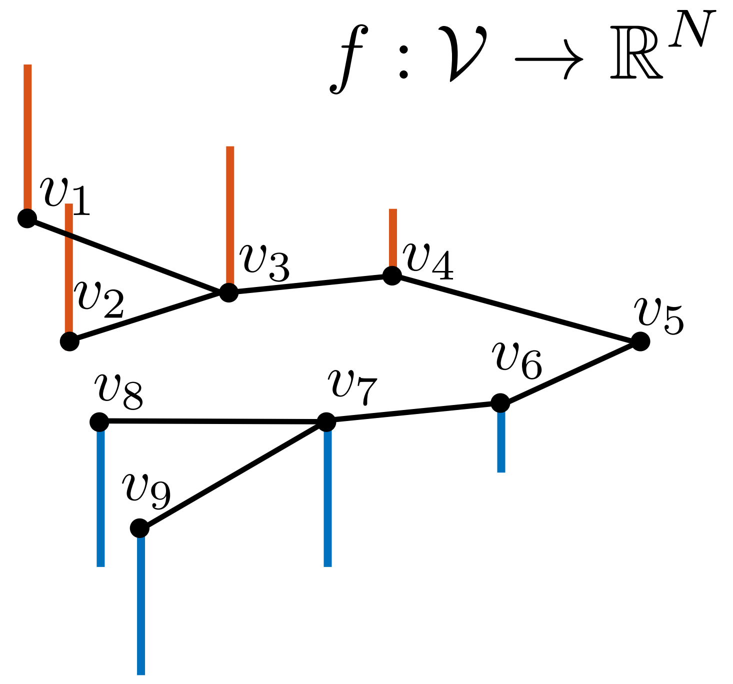


Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Two paradigms

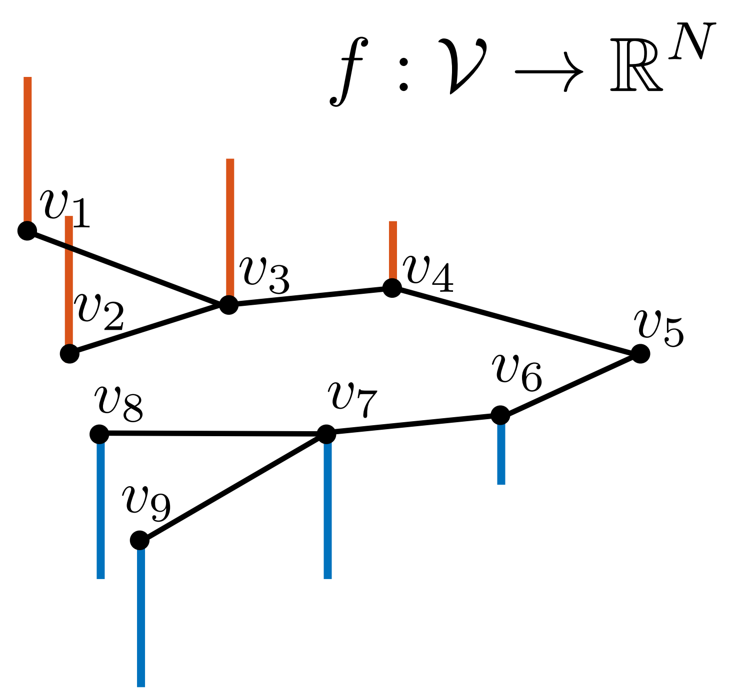
- The main approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs



Two paradigms

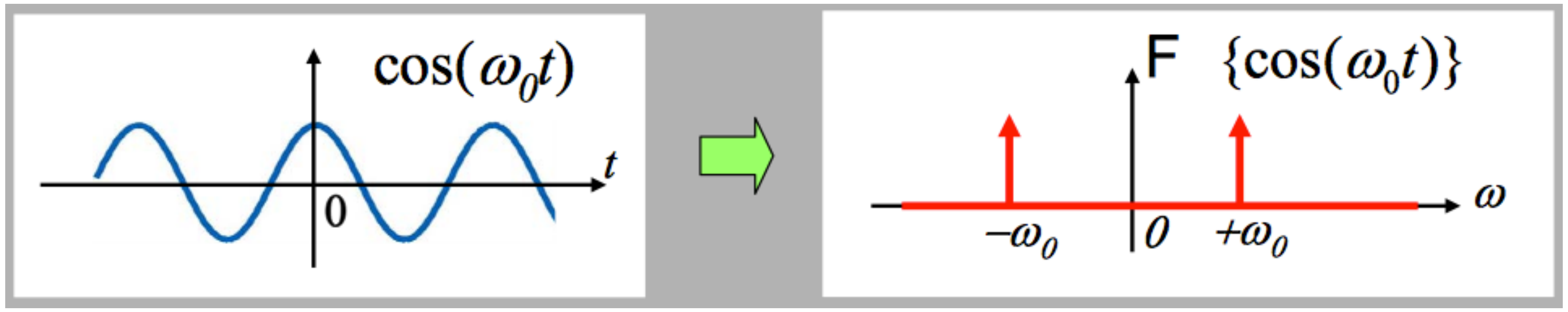
- The main approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

Important for analysis of signal properties

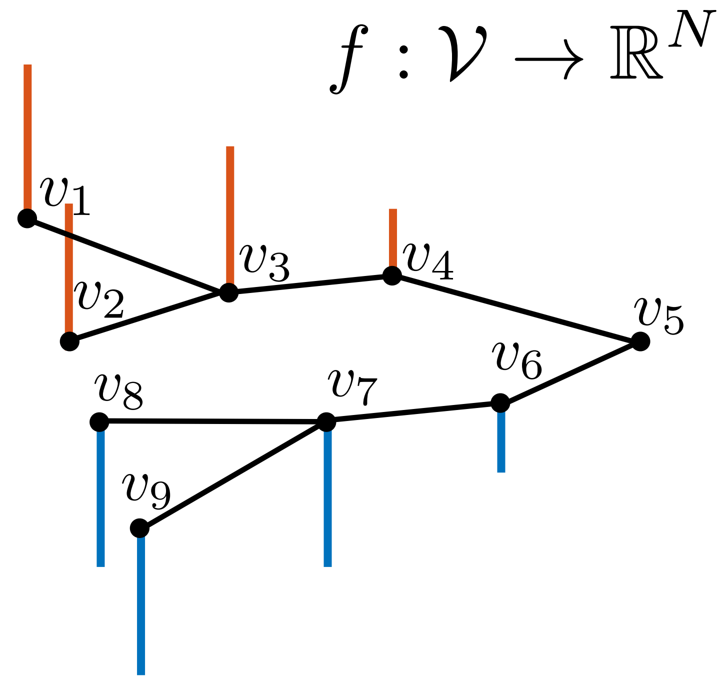


Need for frequency

- Classical Fourier transform provides the frequency domain representation of the signals

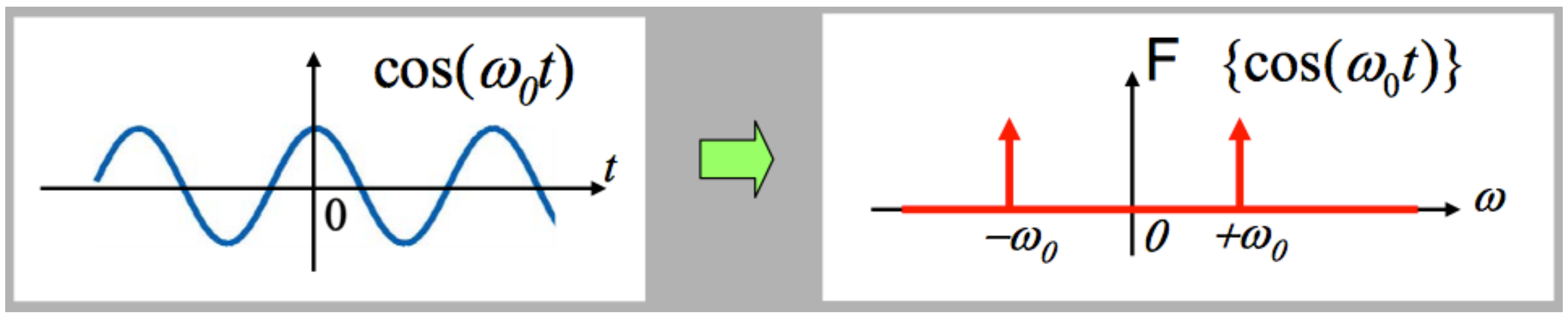


Source: http://www.physik.uni-kl.de



Need for frequency

- Classical Fourier transform provides the frequency domain representation of the signals

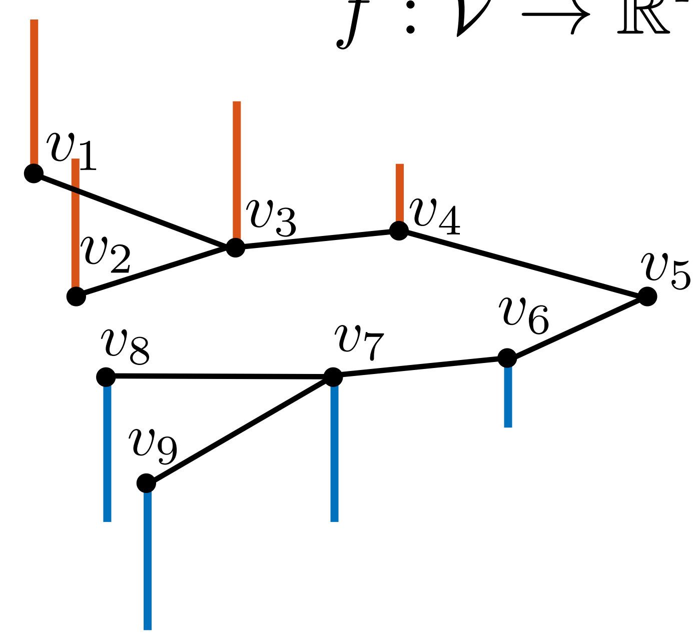


Source: http://www.physik.uni-kl.de

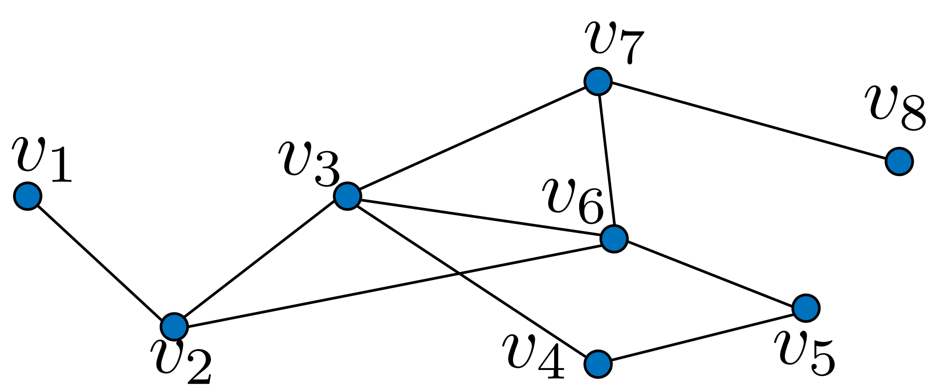
A notion of frequency for graph signals:

We need the graph Laplacian matrix

$$f : \mathcal{V} \to \mathbb{R}^N$$



Graph Laplacian



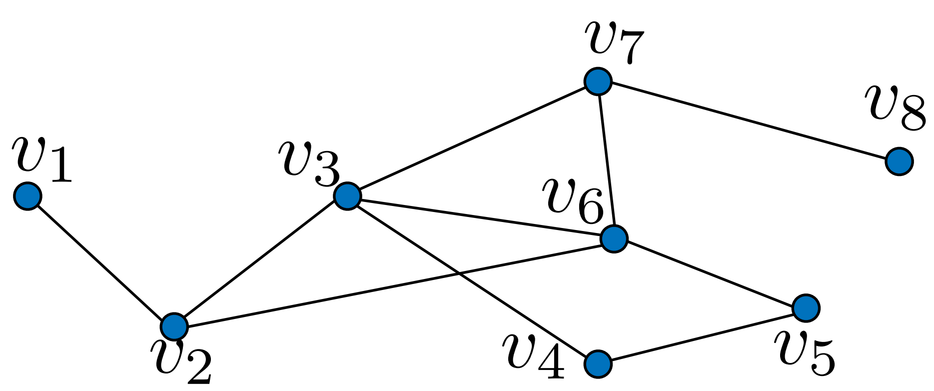
Weighted and undirected graph:

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

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

$$W$$

Graph Laplacian



Weighted and undirected graph:

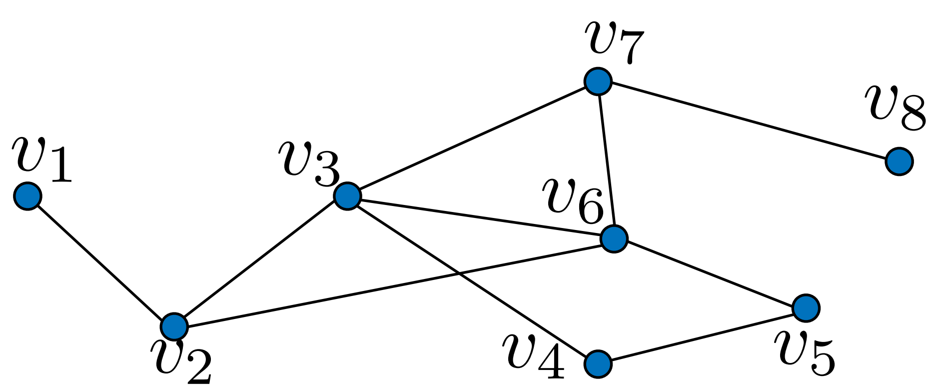
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D \qquad\qquad\qquad W$$

Graph Laplacian



Weighted and undirected graph:

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

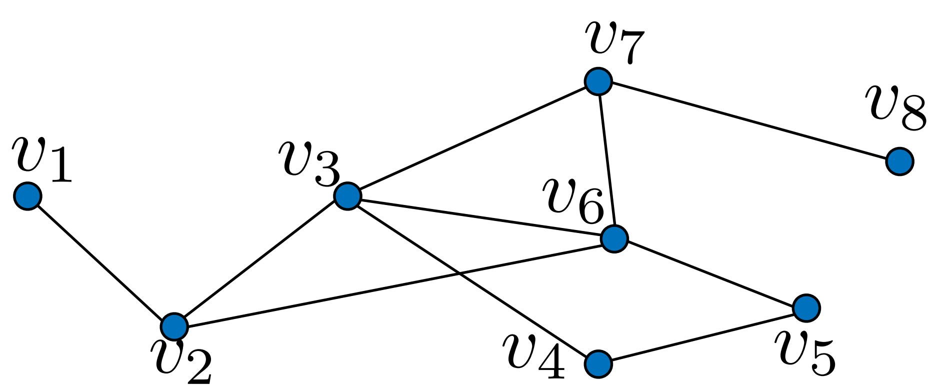
$$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$$

$$L = D - W$$ **Equivalent to G!**

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}}_{L}$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

Graph Laplacian



Weighted and undirected graph:

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

$$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$$

$$L = D - W$$ **Equivalent to G!**

$$L_{\text{norm}} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{W} = \underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}}_{L}$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

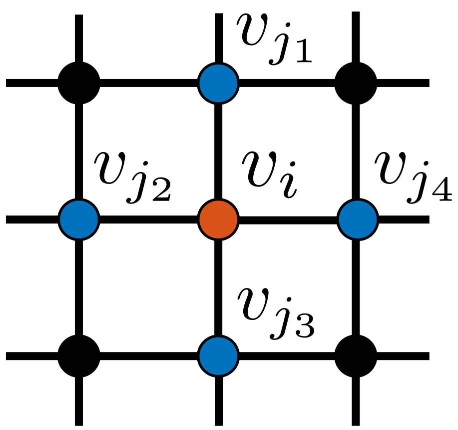
Graph Laplacian

Why graph Laplacian?

Graph Laplacian

Why graph Laplacian?

- approximation of the Laplace operator



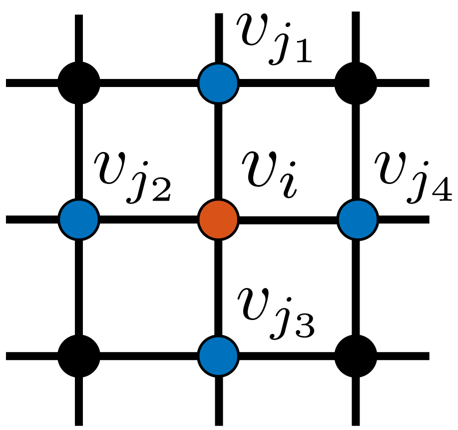
$$(Lf)(i) = 4f(i) - [f(j_1) + f(j_2) + f(j_3) + f(j_4)]$$

standard 5-point stencil for approximating $-\nabla^2 f$

Graph Laplacian

Why graph Laplacian?

- approximation of the Laplace operator

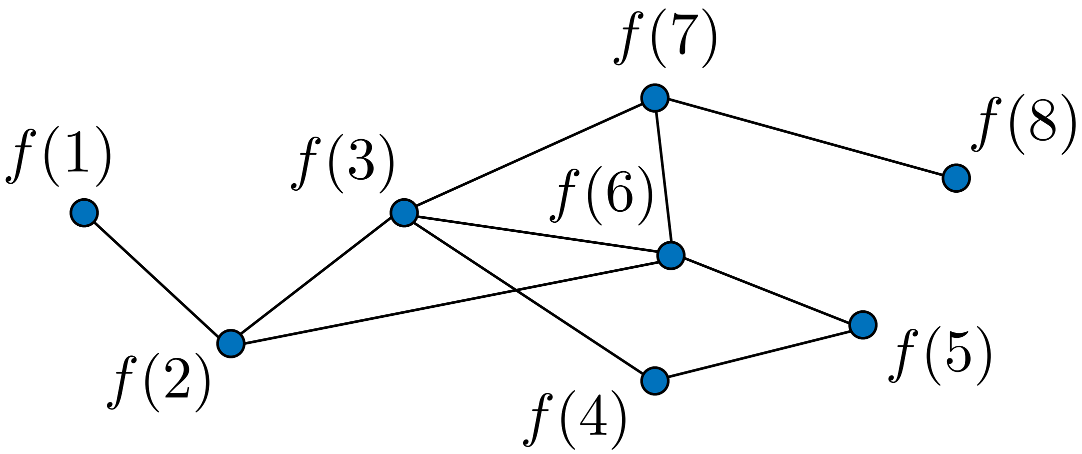


$$(Lf)(i) = 4f(i) - [f(j_1) + f(j_2) + f(j_3) + f(j_4)]$$

standard 5-point stencil for approximating $-\nabla^2 f$

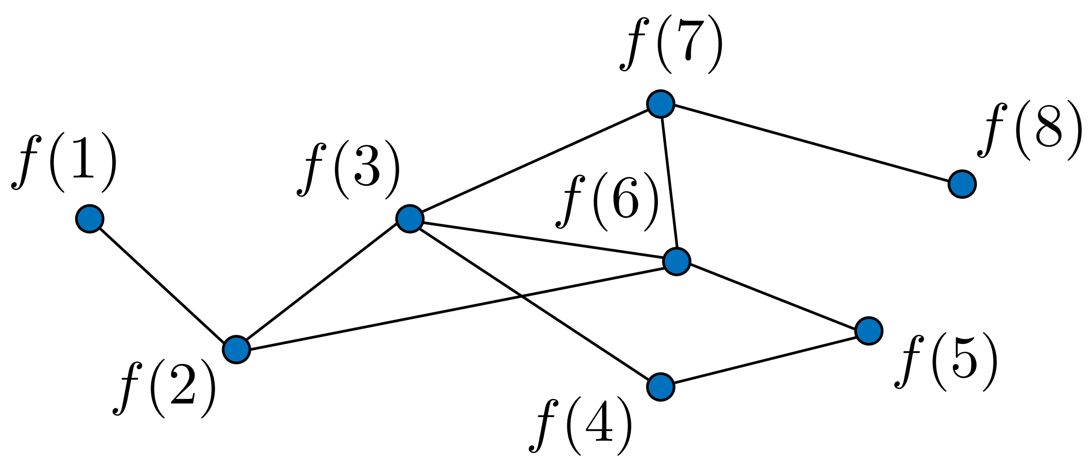
- converges to the Laplace-Beltrami operator (given certain conditions)
- provides a notion of "frequency" on graphs

Graph Laplacian



Graph signal $f : \mathcal{V} \to \mathbb{R}^N$

Graph Laplacian

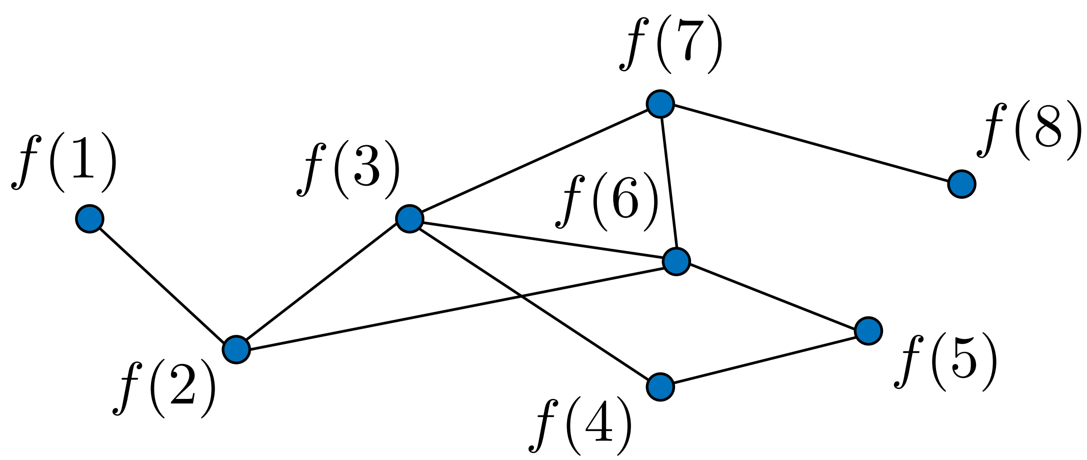


Graph signal $f : \mathcal{V} \to \mathbb{R}^N$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}$$

$$Lf(i) = \sum_{j=1}^{N} W_{ij}(f(i) - f(j))$$

Graph Laplacian



Graph signal $f : \mathcal{V} \to \mathbb{R}^N$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}$$

$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}$$

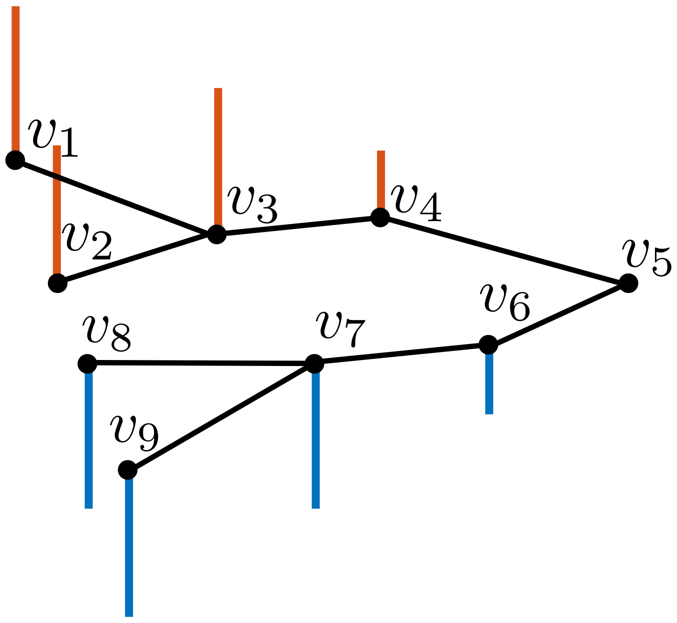
$$Lf(i) = \sum_{j=1}^{N} W_{ij}(f(i) - f(j))$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left(f(i) - f(j)\right)^2$$

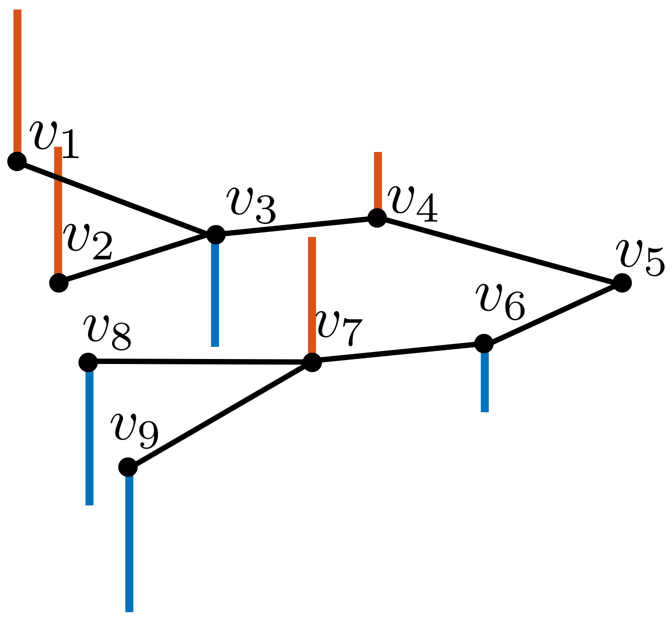
A measure of "smoothness"

Zhou and Schölkopf, "A regularization framework for learning from graph data," ICML Workshop, 2004.

Graph Laplacian



$$f^T L f = 1$$



$$f^T L f = 21$$

Graph Laplacian

- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

$$L = \underbrace{\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}}_{\chi} \underbrace{\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} — & \chi_0 & — \\ & \cdots & \\ — & \chi_{N\text{-}1} & — \end{bmatrix}}_{\chi^T}$$

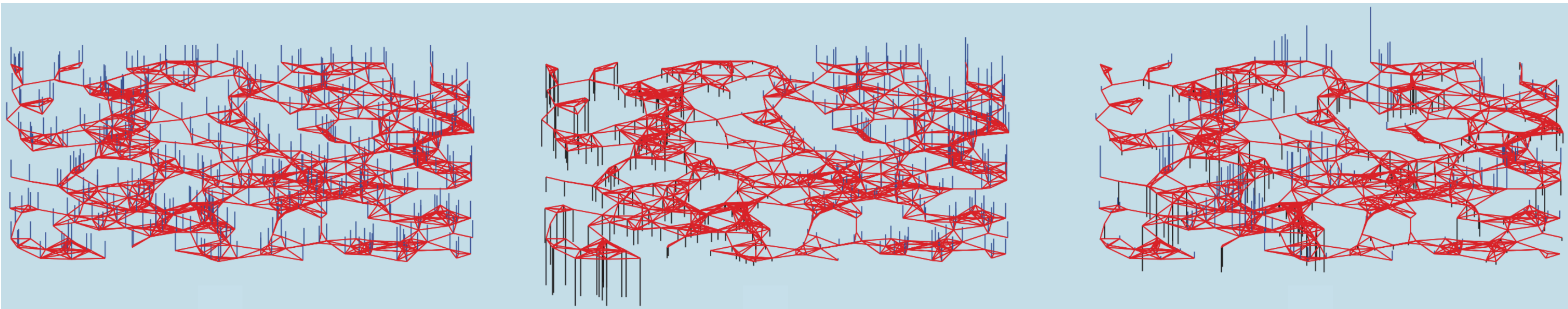
Graph Laplacian

- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

$$L = \underbrace{\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}}_{\chi} \underbrace{\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \text{---} & \chi_0 & \text{---} \\ & \cdots & \\ \text{---} & \chi_{N\text{-}1} & \text{---} \end{bmatrix}}_{\chi^T}$$

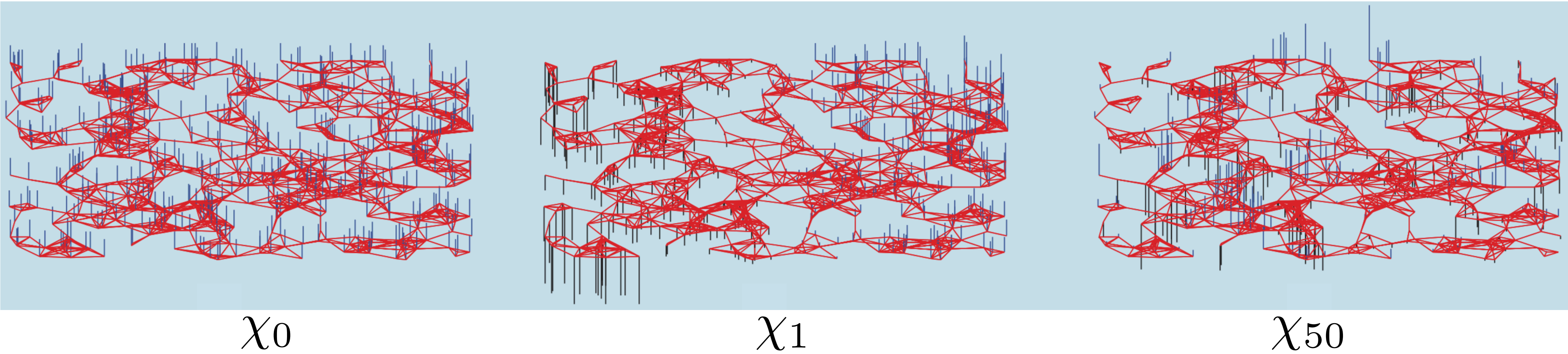
- Eigenvalues are usually sorted increasingly: $0 = \lambda_0 < \lambda_1 \leq \ldots \leq \lambda_{N-1}$

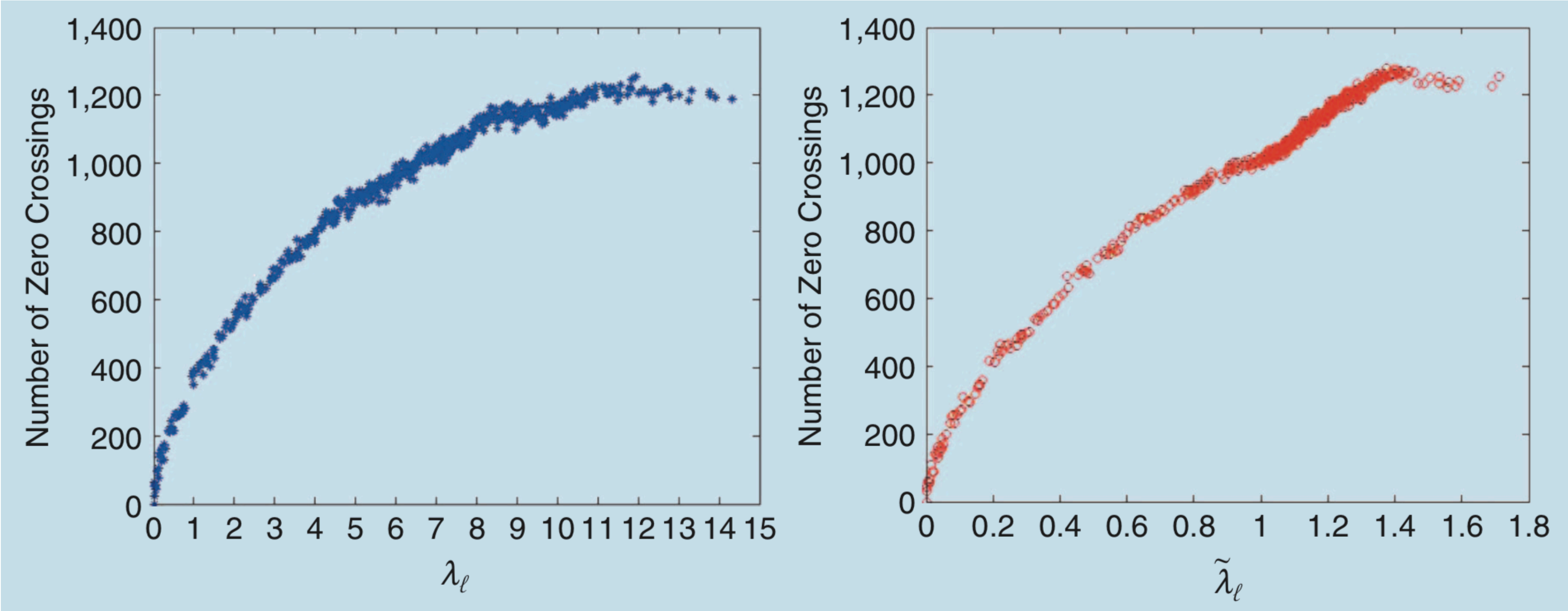
Graph Fourier transform



χ_0 χ_1 χ_{50}

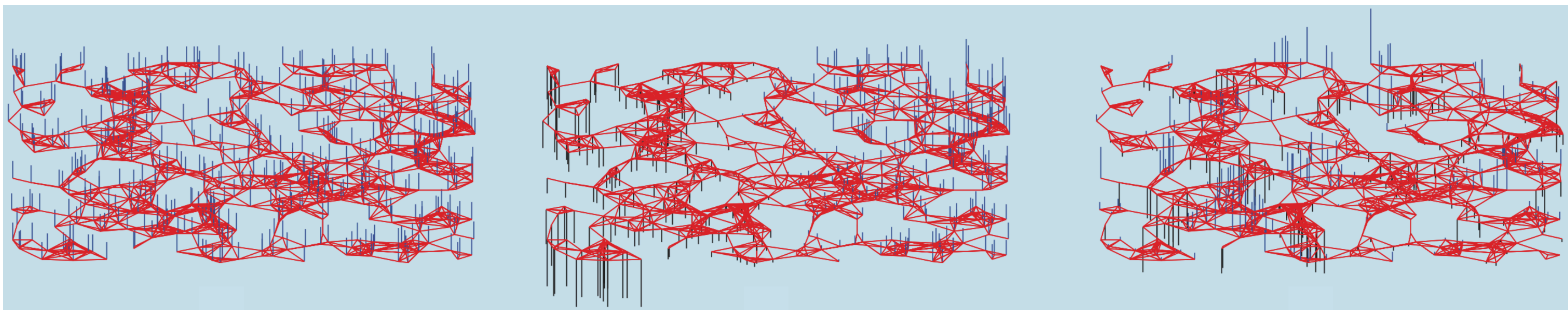
Graph Fourier transform





Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform



χ_0 χ_1 χ_{50}

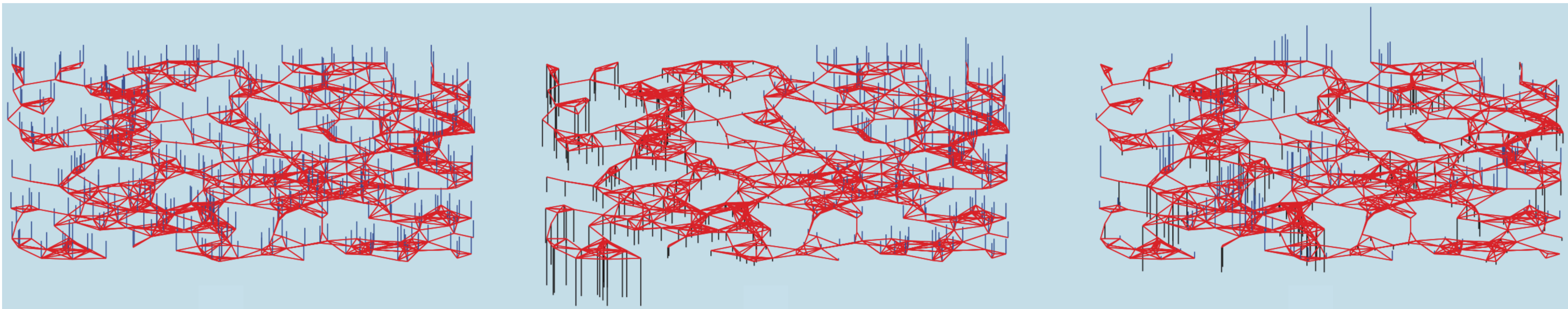
Low frequency → High frequency

$\boxed{L = \chi \Lambda \chi^T}$ $\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

- Eigenvectors associated with smaller eigenvalues have values that vary less rapidly along the edges

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform



χ_0 χ_1 χ_{50}

Low frequency → High frequency

$\boxed{L = \chi \Lambda \chi^T}$

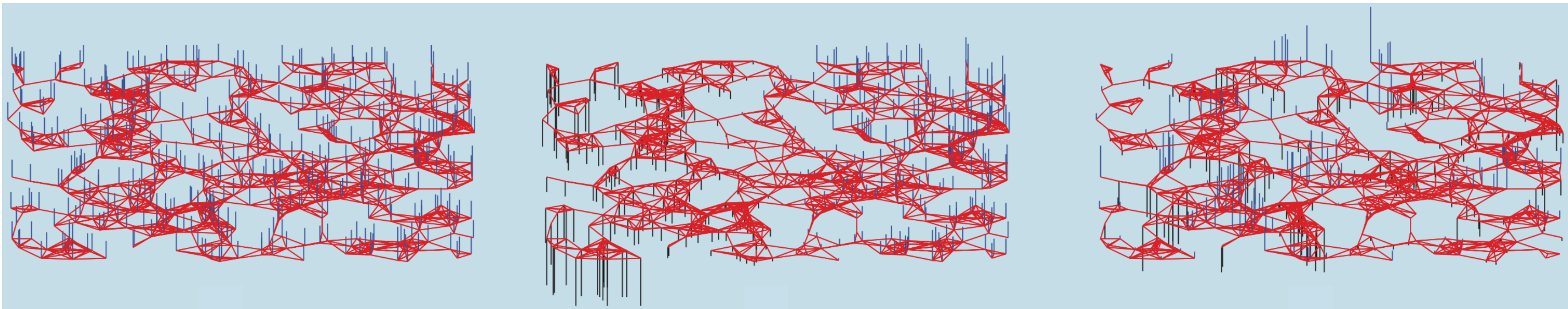
$\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

Graph Fourier transform:
[Hammond11]

$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Graph Fourier transform



χ_0 χ_1 χ_{50}

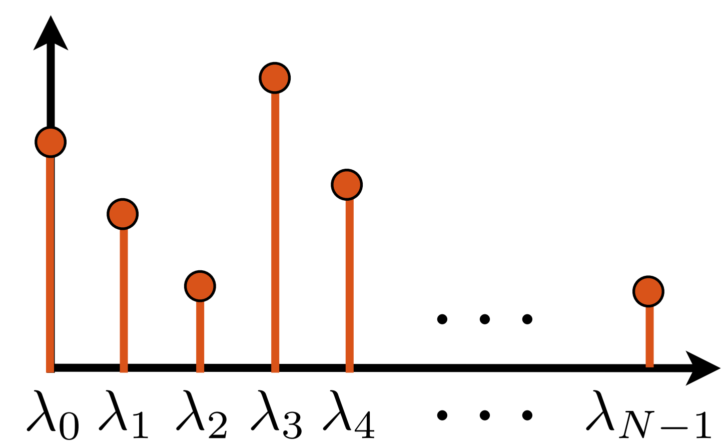
Low frequency High frequency

$\boxed{L = \chi \Lambda \chi^T}$

$\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

<u>Graph Fourier transform:</u>
[Hammond11]

$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{bmatrix} | \\ f \\ | \end{bmatrix}$$



Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

eigenvectors: χ_ℓ

$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i)$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \boxed{\int} \boxed{(e^{j\omega x})^*} \boxed{f(x)} dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

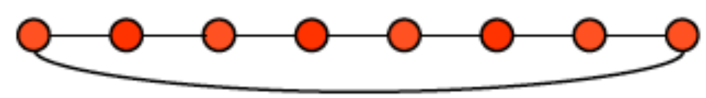
eigenvectors: χ_ℓ

$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \boxed{\sum_{i=1}^{N}} \boxed{\chi_\ell^*(i)} \boxed{f(i)}$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

Two special cases



- (Unordered) Laplacian eigenvalues: $\lambda_\ell = 2 - 2\cos\left(\frac{2\ell\pi}{N}\right)$

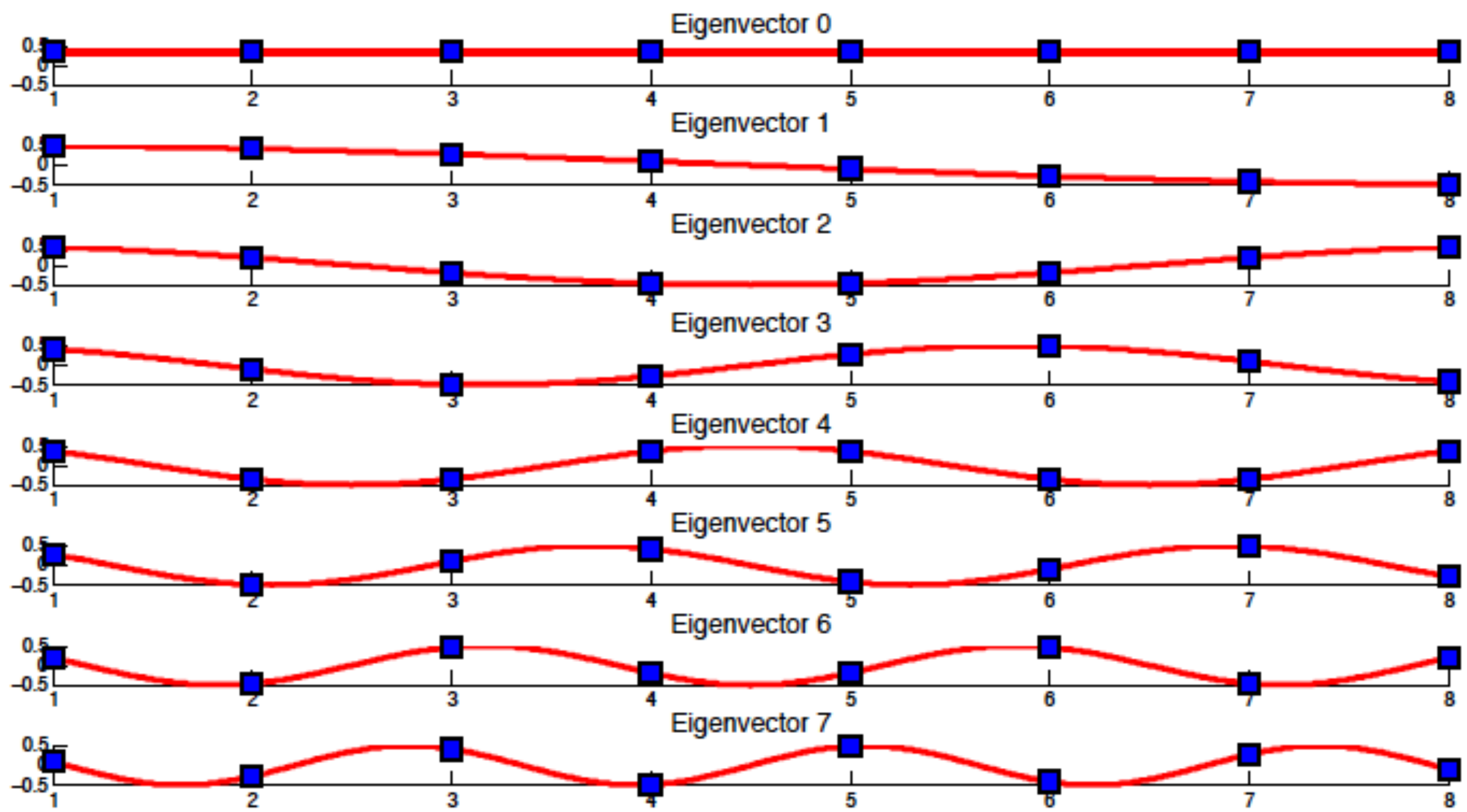
- One possible choice of orthogonal Laplacian eigenvectors:

$$\chi_\ell = \left[1, \omega^\ell, \omega^{2\ell}, \ldots, \omega^{(N-1)\ell}\right], \text{ where } \omega = e^{\frac{2\pi j}{N}}$$

- $\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Fourier Transform (DFT) matrix

Two special cases

$\lambda_\ell = 2 - 2\cos\left(\frac{\pi\ell}{N}\right)$ $\chi_0(i) = \frac{1}{\sqrt{N}}, \; \chi_\ell(i) = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi\ell(i-0.5)}{N}\right), \; \ell = 1, 2, \ldots, N-1$



$\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Cosine Transform matrix (DCT-II, Strang, 1999), which is used in JPEG image compression

Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x)dx \qquad f(x) = \frac{1}{2\pi}\int \hat{f}(\omega)e^{j\omega x}d\omega$

Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f

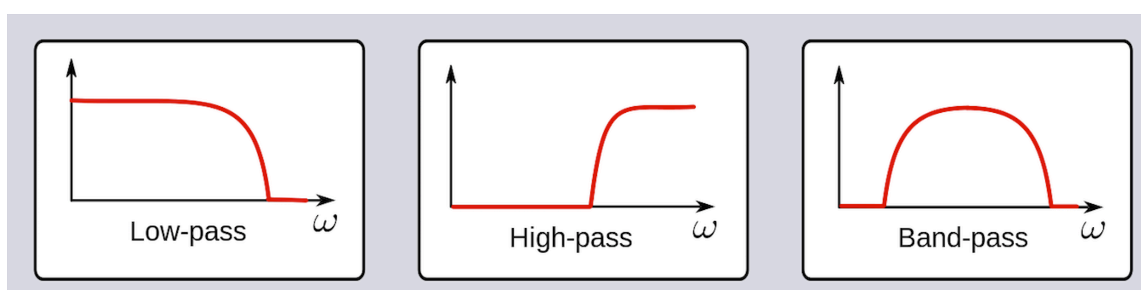
$$\boxed{f} \xrightarrow{\text{FT}} \boxed{\hat{f}(\omega)} \xrightarrow{\hat{g}(\omega)} \boxed{\hat{g}(\omega)\hat{f}(\omega)} \xrightarrow{\text{IFT}} \boxed{f * g}$$

Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f

$$f \xrightarrow{\text{FT}} \hat{f}(\omega) \xrightarrow{\hat{g}(\omega)} \hat{g}(\omega)\hat{f}(\omega) \xrightarrow{\text{IFT}} f * g$$



Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$

Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \to \mathbb{R}^N$

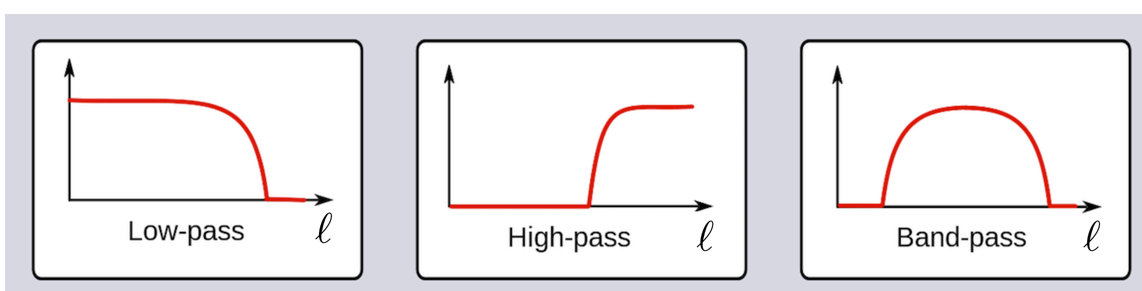
$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\hat{f}(\ell)} \xrightarrow{\hat{g}(\lambda_\ell)} \boxed{\hat{g}(\lambda_\ell)\hat{f}(\ell)} \xrightarrow{\text{IGFT}} \boxed{f(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)}$$

Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \to \mathbb{R}^N$

$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\hat{f}(\ell)} \xrightarrow{\hat{g}(\lambda_\ell)} \boxed{\hat{g}(\lambda_\ell)\hat{f}(\ell)} \xrightarrow{\text{IGFT}} \boxed{f(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)}$$



Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$

Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \to \mathbb{R}^N$

$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\chi \hat{g}(\Lambda)\chi^T f}$$

$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

Graph Laplacian revisited

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda) = \Lambda} \Lambda \chi^T f \xrightarrow{\text{IGFT}} \boxed{\chi \Lambda \chi^T} f$$

$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

Graph Laplacian revisited

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda) = \Lambda} \Lambda \chi^T f \xrightarrow{\text{IGFT}} \boxed{\chi \Lambda \chi^T} f$$

$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

The Laplacian operator filters the signal in the spectral domain by its eigenvalues!

The Laplacian quadratic form: $f^T L f = ||L^{\frac{1}{2}} f||_2 = ||\chi \Lambda^{\frac{1}{2}} \chi^T f||_2$

Graph transform/dictionary design

- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \chi\hat{g}(\Lambda)\chi^T f$$

Graph transform/dictionary design

- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)\text{: functions of L!}} f$$

Graph transform/dictionary design

- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!

$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\underline{\chi\hat{g}(\Lambda)\chi^T} f}$$

$\hat{g}(L)$**: functions of L!**

- Important properties can be achieved by properly defining $\hat{g}(L)$, such as localisation of atoms

- Closely related to kernels and regularisation on graphs

Smola and Kondor, "Kernels and regularization on graphs", COLT, 2003.

A simple example

$$\boxed{f} \xrightarrow{\text{GFT}} \boxed{\chi^T f} \xrightarrow{\hat{g}(\Lambda)} \boxed{\hat{g}(\Lambda)\chi^T f} \xrightarrow{\text{IGFT}} \boxed{\underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f}$$

A simple example

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f$$

Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$\boxed{y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}}$$

A simple example

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi \hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f$$

Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$y^* = \arg\min_{y} \{ \underbrace{||y - f||_2^2}_{\text{Data fitting term}} + \gamma \underbrace{y^T L y}_{\text{“Smoothness” assumption}} \}$$

A simple example

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f$$

Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

Data fitting term

$$y^* = \arg\min_y \{ ||y - f||_2^2 + \gamma y^T L y \}$$

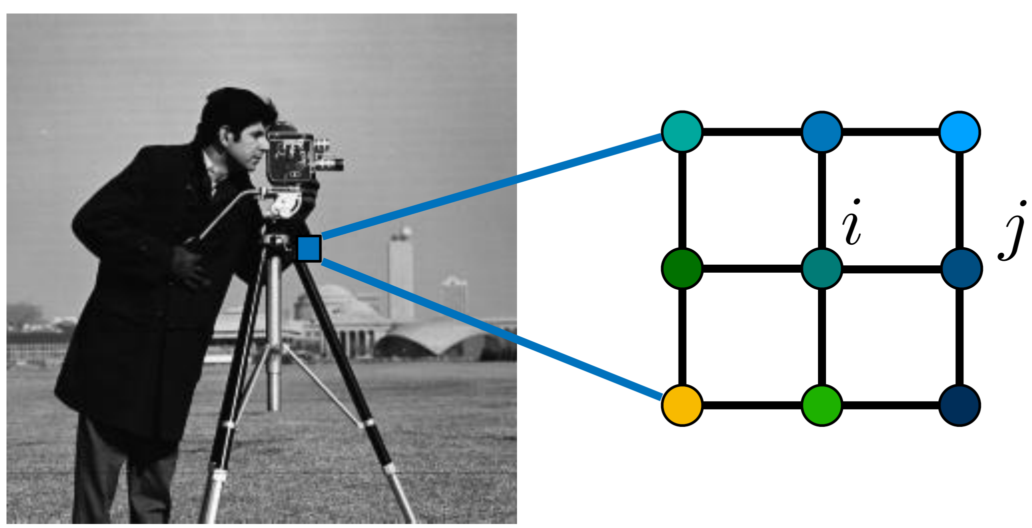
"Smoothness" assumption

$$y^* = \underbrace{(I + \gamma L)^{-1}}_{\hat{g}(L)} f$$

Laplacian (Tikhonov) regularisation is equivalent to low-pass filtering in the graph spectral domain!

A simple example

- noisy image as the observed noisy graph signal
- regular grid graph (weights inversely proportional to pixel value difference)

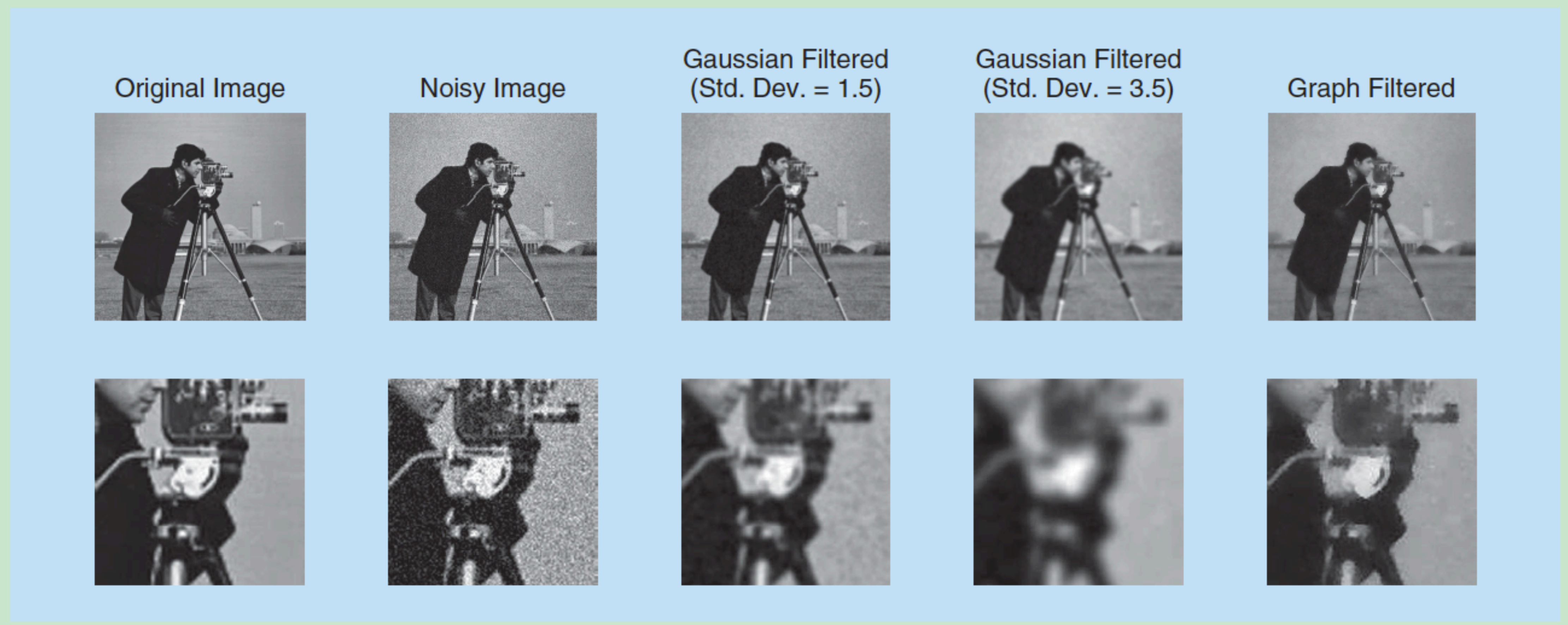


$$w_{ij} = \frac{1}{|f(i) - f(i)|}$$

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

A simple example

- noisy image as the observed noisy graph signal
- regular grid graph (weights inversely proportional to pixel value difference)



[FIGS2] Image denoising via classical Gaussian filtering and graph spectral filtering.

Shuman et al., "The emerging field of signal processing on graphs," IEEE SPM, 2013.

Example designs

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi \hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f$$

Low-pass filters: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma\Lambda)^{-1}\chi^T$

Example designs

$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \underbrace{\chi\hat{g}(\Lambda)\chi^T}_{\hat{g}(L)} f$$

Low-pass filters: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma\Lambda)^{-1}\chi^T$

Window kernel: Windowed graph Fourier transform

Shifted and dilated band-pass filters: Spectral graph wavelets $\hat{g}(sL)$

Adapted kernels: Learn values of $\hat{g}(L)$ directly from data

Parametric polynomials: $\hat{g}_s(L) = \sum_{k=0}^{K} \alpha_{sk} L^k = \chi(\sum_{k=0}^{K} \alpha_{sk}\Lambda^k)\chi^T$

Shuman et al., "Dictionary design for graph signal processing," GSP Workshop, 2016.

Outline

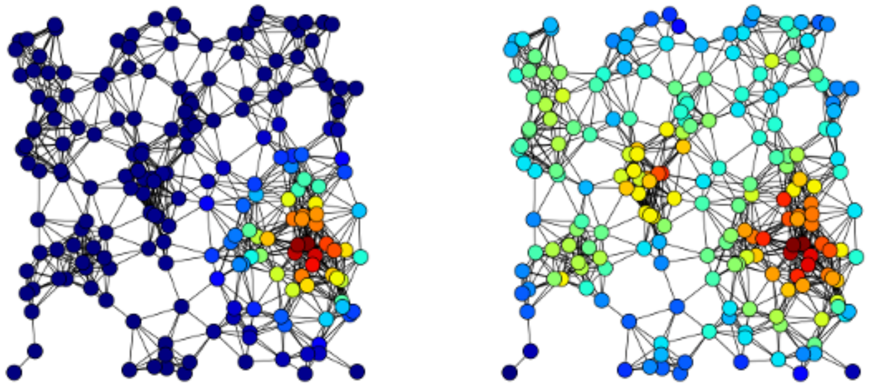
- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

GSP and the literature

There is a rich literature about data analysis and learning on graphs

GSP and the literature

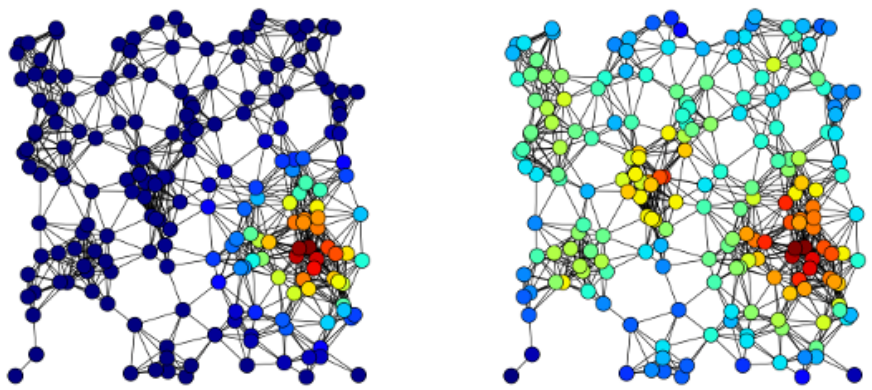
There is a rich literature about data analysis and learning on graphs



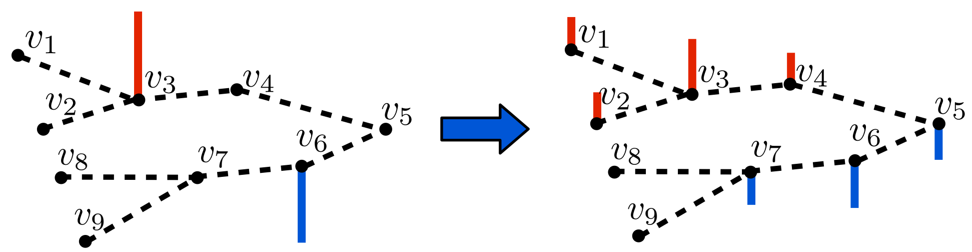
network science

GSP and the literature

There is a rich literature about data analysis and learning on graphs



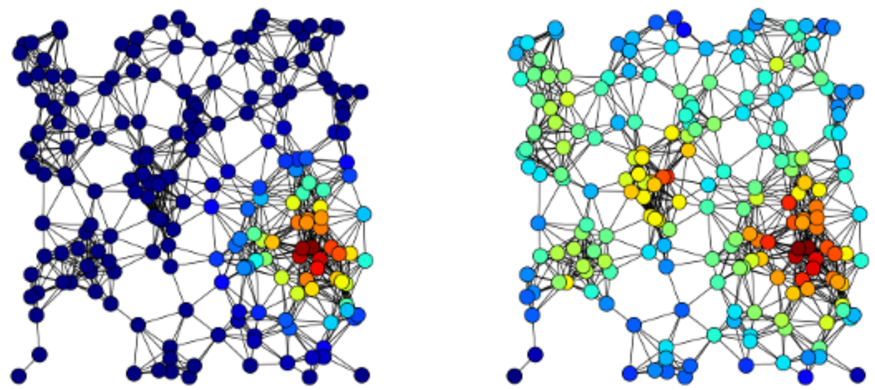
network science



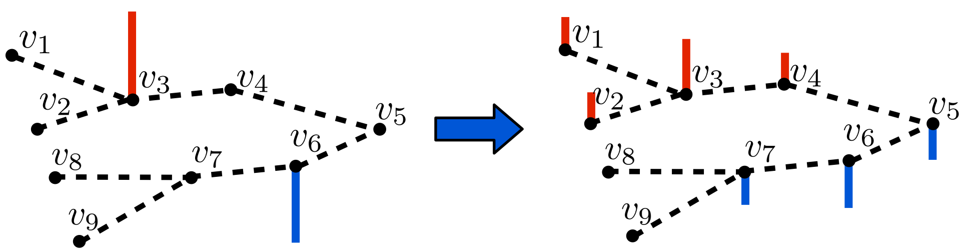
diffusion on graphs

GSP and the literature

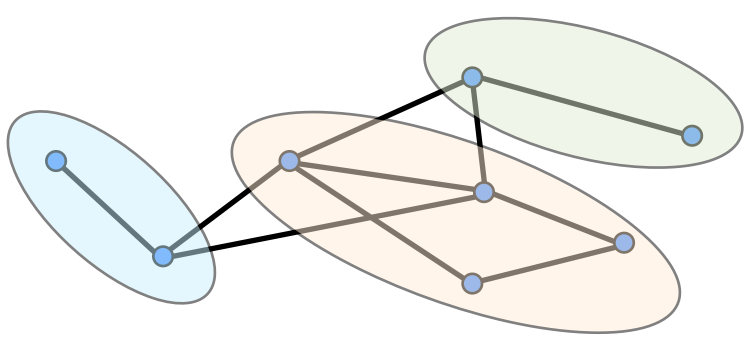
There is a rich literature about data analysis and learning on graphs



network science



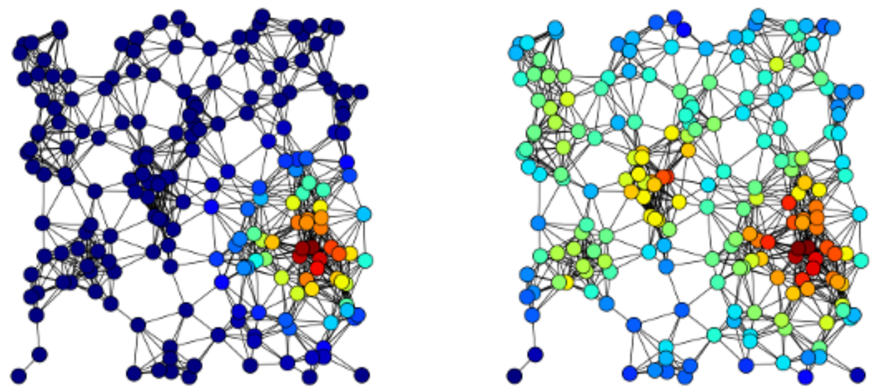
diffusion on graphs



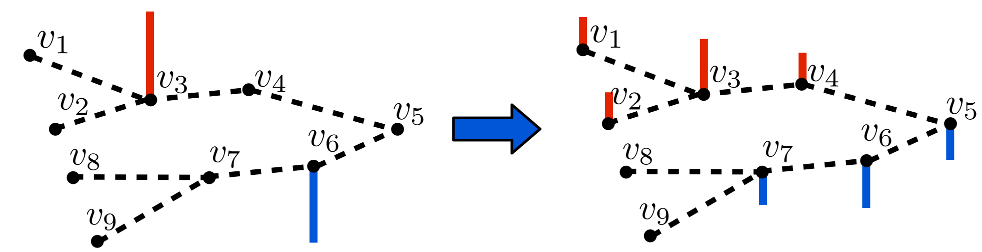
unsupervised learning (dimensionality reduction, clustering)

GSP and the literature

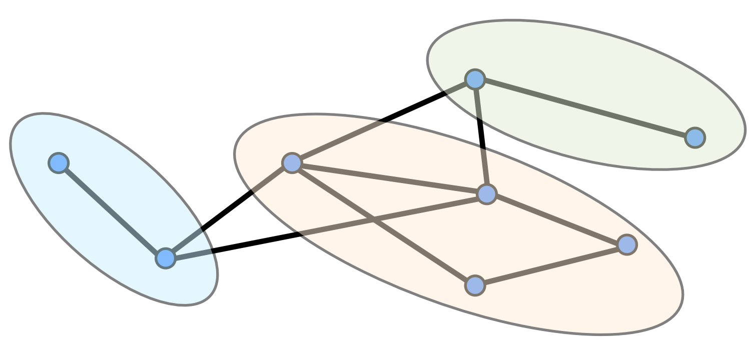
There is a rich literature about data analysis and learning on graphs



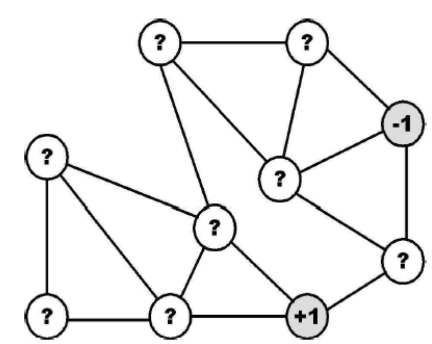
network science



diffusion on graphs



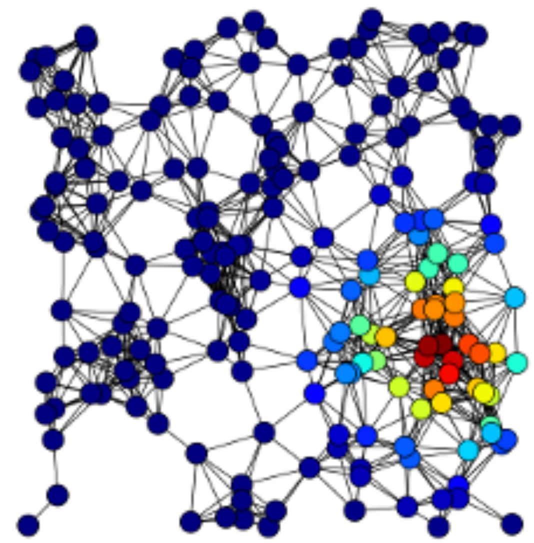
unsupervised learning (dimensionality reduction, clustering)



semi-supervised learning

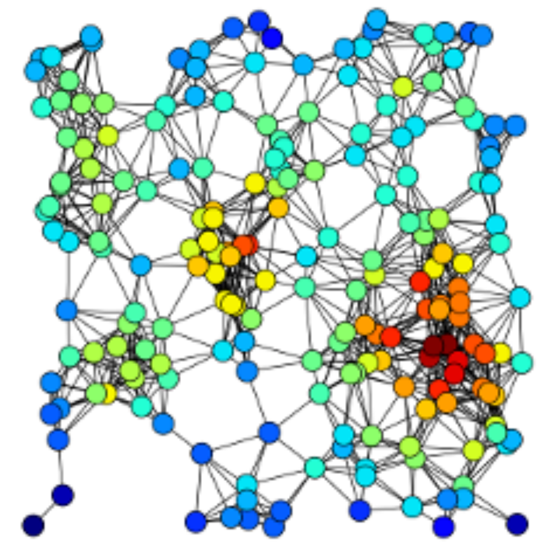
Network centrality

eigenvector centrality



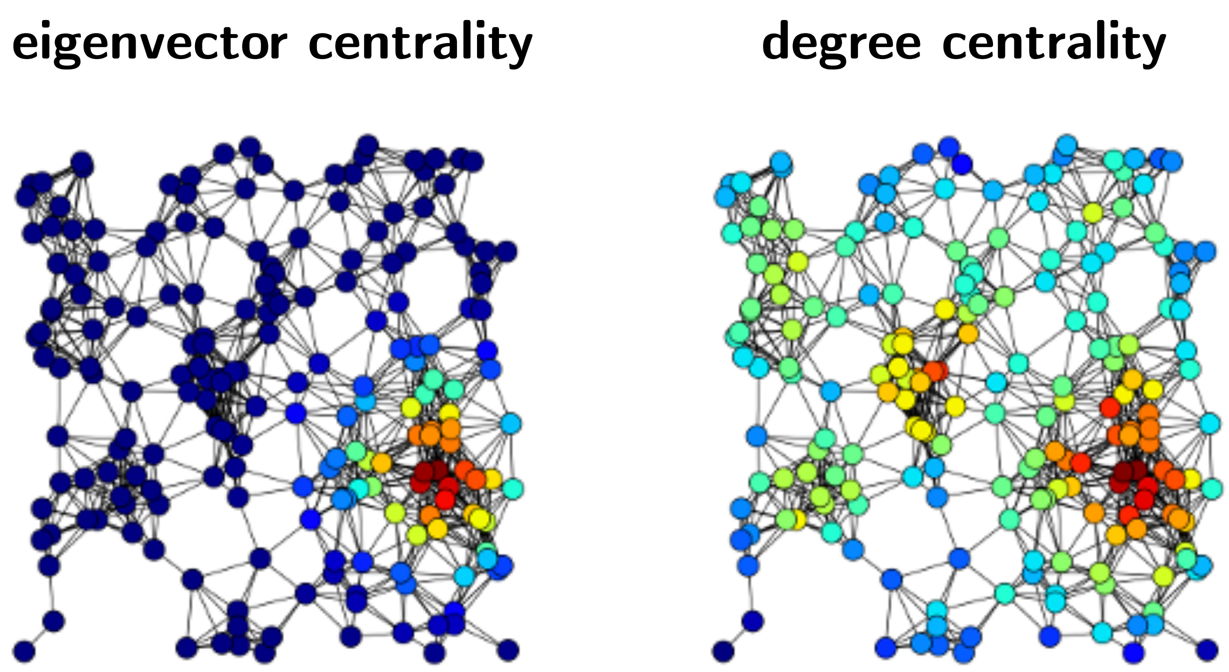
$$Wx = \lambda_{\max} x$$

degree centrality



$$d = [d(v_1), \cdots, d(v_N)]$$

Network centrality

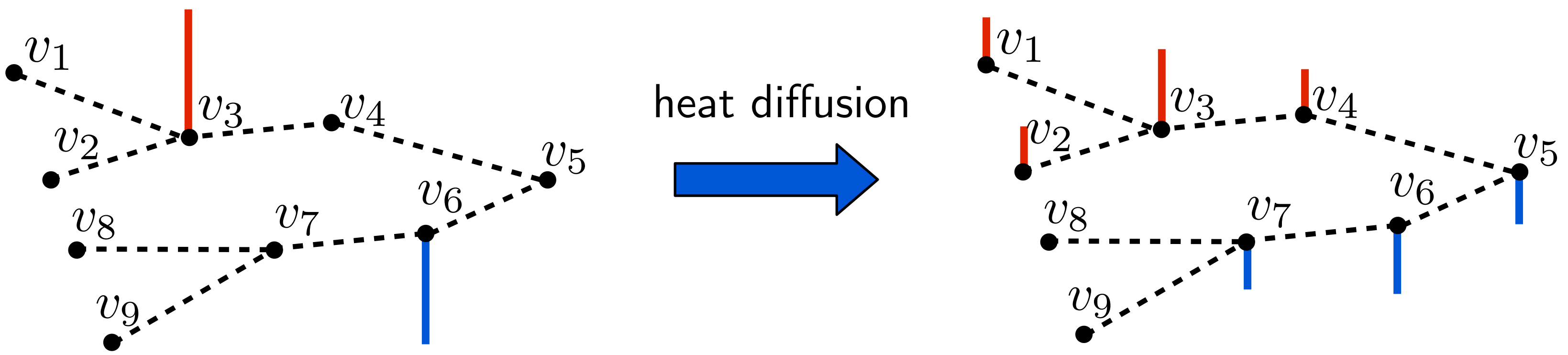


$$Wx = \lambda_{\max} x \qquad d = [d(v_1), \cdots, d(v_N)]$$

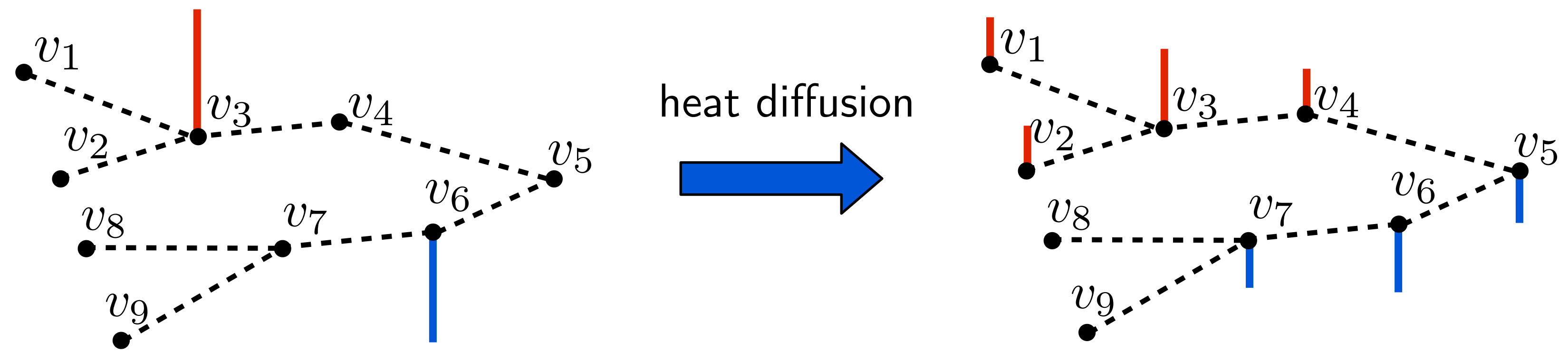
- Google's PageRank is a variant of eigenvector centrality
- eigenvectors of W can also be used to provide a frequency interpretation for graph signals

PageRank: http://www.ams.org/publicoutreach/feature-column/fcarc-pagerank
Sandryhaila and Moura, "Discrete signal processing on graphs", IEEE TSP, 2013.

Diffusion on graphs



Diffusion on graphs

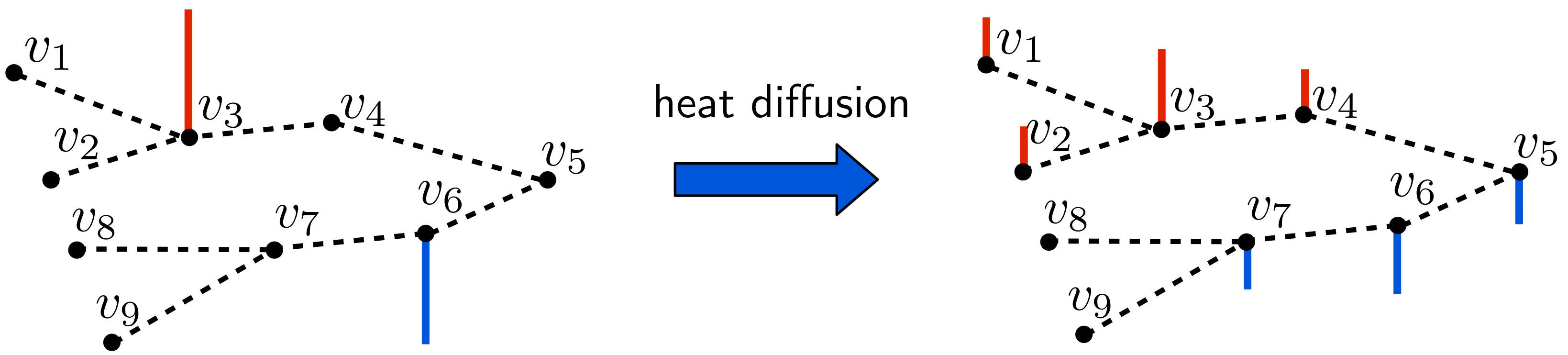


$$\frac{\partial x}{\partial \tau} - Lx = 0$$

$$x(v, 0) = x_0(v)$$

$$\Longrightarrow \quad x(v, \tau) = e^{-\tau L} x_0(v)$$

Diffusion on graphs



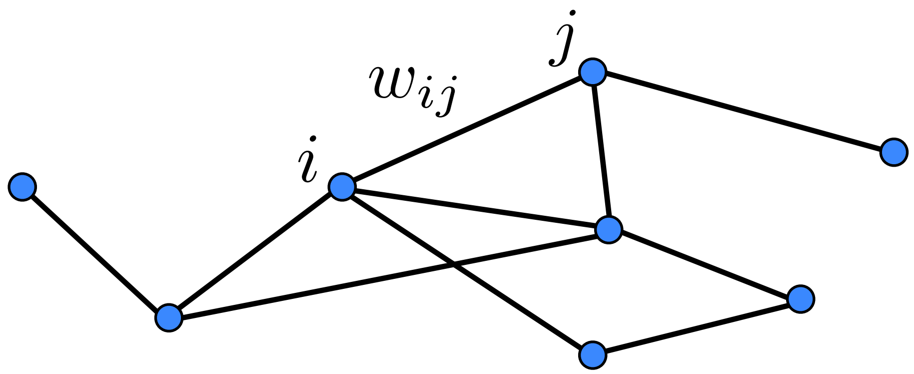
$$\frac{\partial x}{\partial \tau} - Lx = 0$$
$$x(v, 0) = x_0(v)$$

$$x(v, \tau) = e^{-\tau L} x_0(v)$$

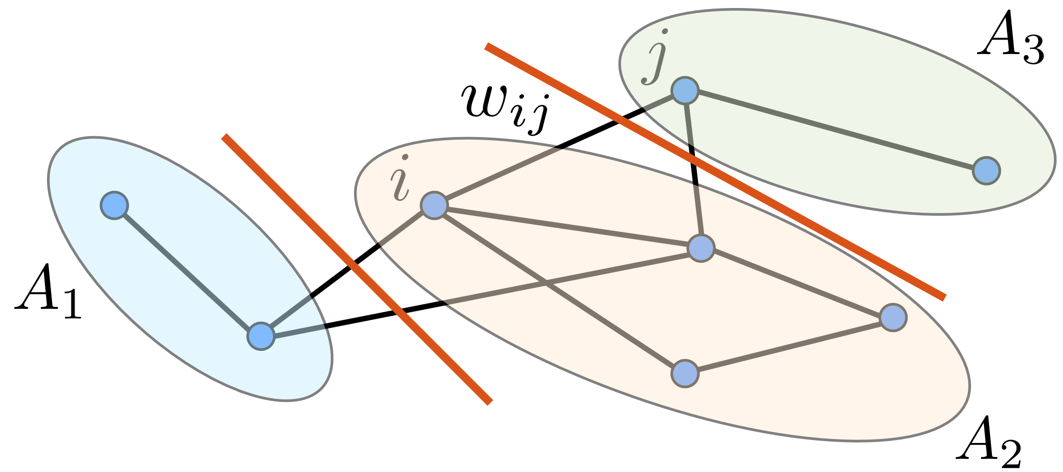
- heat diffusion on graphs is a typical physical process on graphs
- other possibilities exist (e.g., random walk on graphs)
- many have an interpretation of filtering on graphs

Smola and Kondor, "Kernels and regularization on graphs", COLT, 2003.

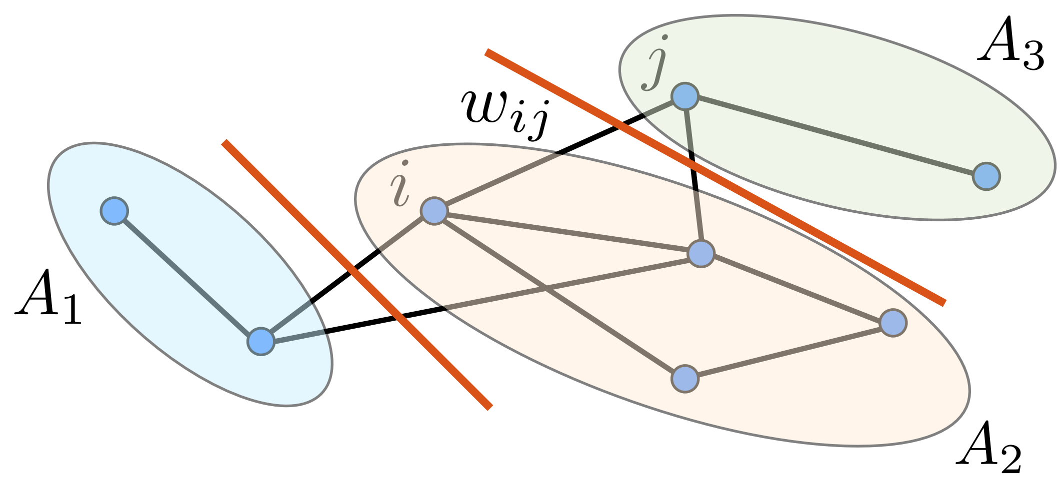
Graph clustering (community detection)



Graph clustering (community detection)

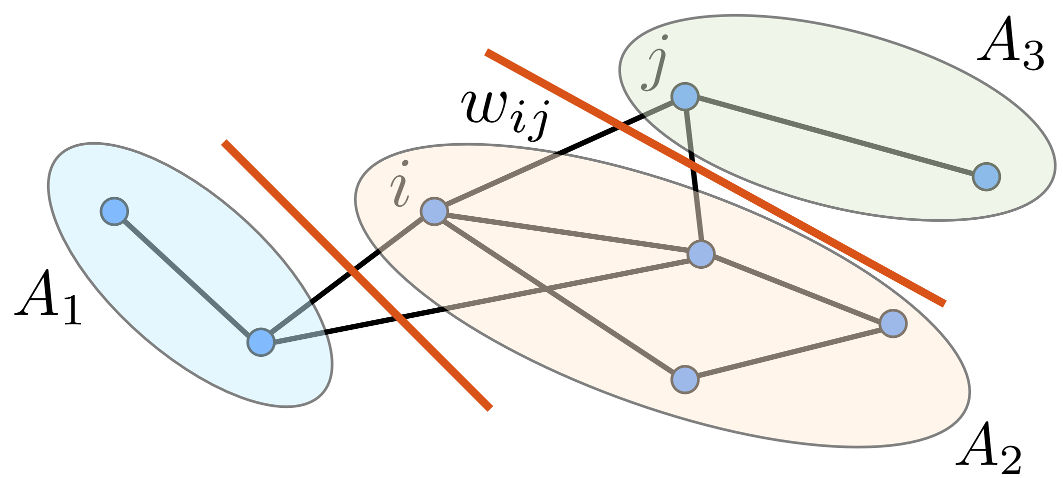


Graph clustering (community detection)



$$NCut(A_1, ..., A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \overline{A_i})}{vol(A_i)}$$

Graph clustering (community detection)

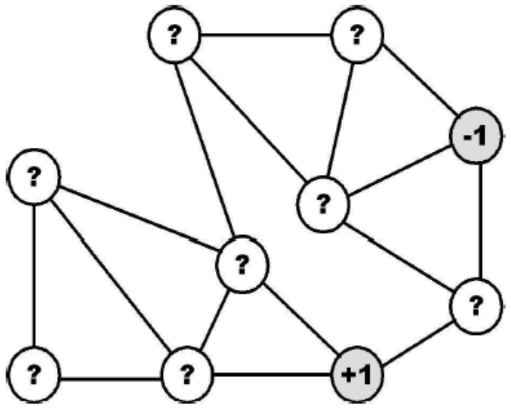


$$NCut(A_1, ..., A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \overline{A_i})}{vol(A_i)}$$

- first k eigenvectors of graph Laplacian minimise the graph cut
- eigenvectors of graph Laplacian enable a Fourier-like analysis for graph signals

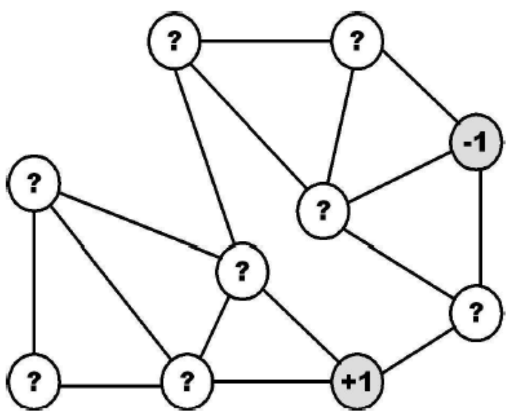
Ulrike von Luxburg, "A tutorial on spectral clustering", Statistics and Computing, 2007.

Semi-supervised learning



Zhu, "Semi-supervised learning with graphs", Ph.D. dissertation, CMU, 2005.

Semi-supervised learning

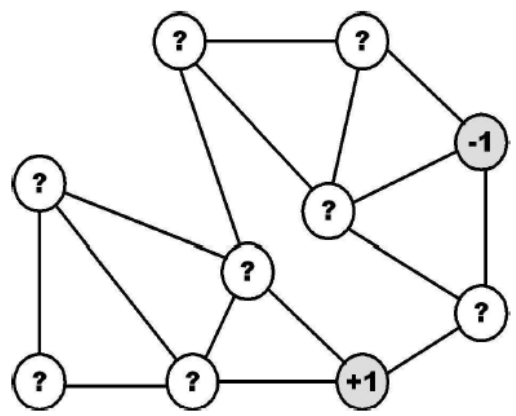


$$y : \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\min_{x \in \mathbb{R}^N} ||y - x||_2^2 + \alpha \; x^T L x,$$

Zhu, "Semi-supervised learning with graphs", Ph.D. dissertation, CMU, 2005.

Semi-supervised learning



$$y : \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

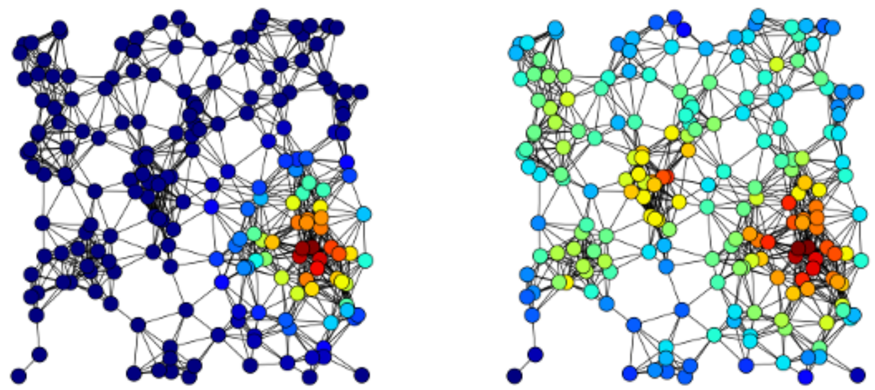
$$\min_{x \in \mathbb{R}^N} ||y - x||_2^2 + \alpha \, x^T L x,$$

- learning by assuming smoothness of predicted labels
- this is equivalent to a denoising problem for graph signal y

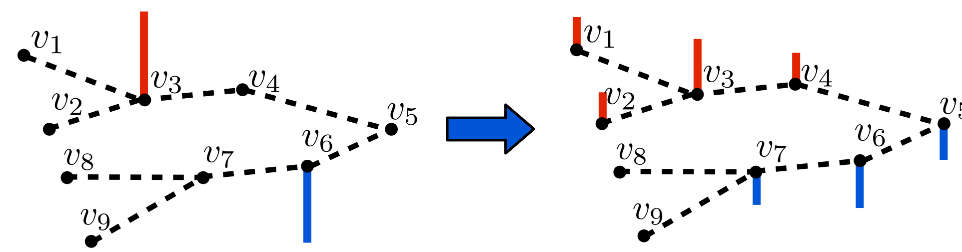
Zhu, "Semi-supervised learning with graphs", Ph.D. dissertation, CMU, 2005.

GSP and the literature

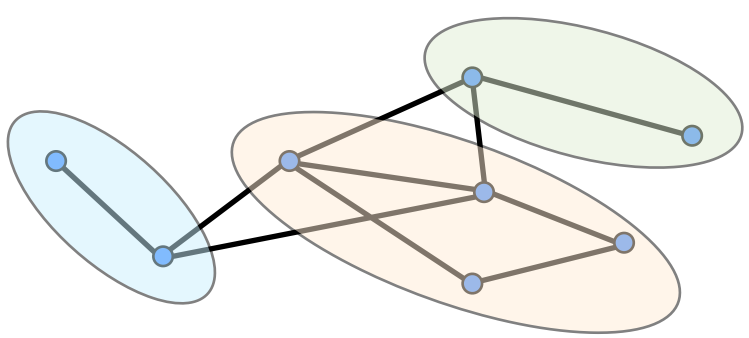
centrality, diffused information, class membership, node labels (and node-level features in general) can ALL be viewed as graph signals



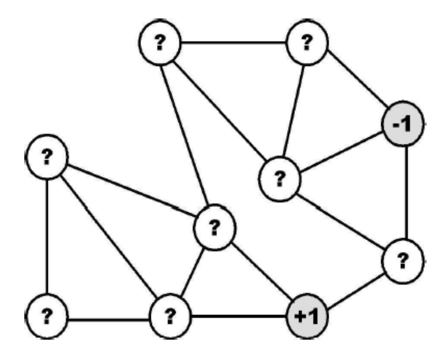
network science



network diffusion



unsupervised learning (dimensionality reduction, clustering)

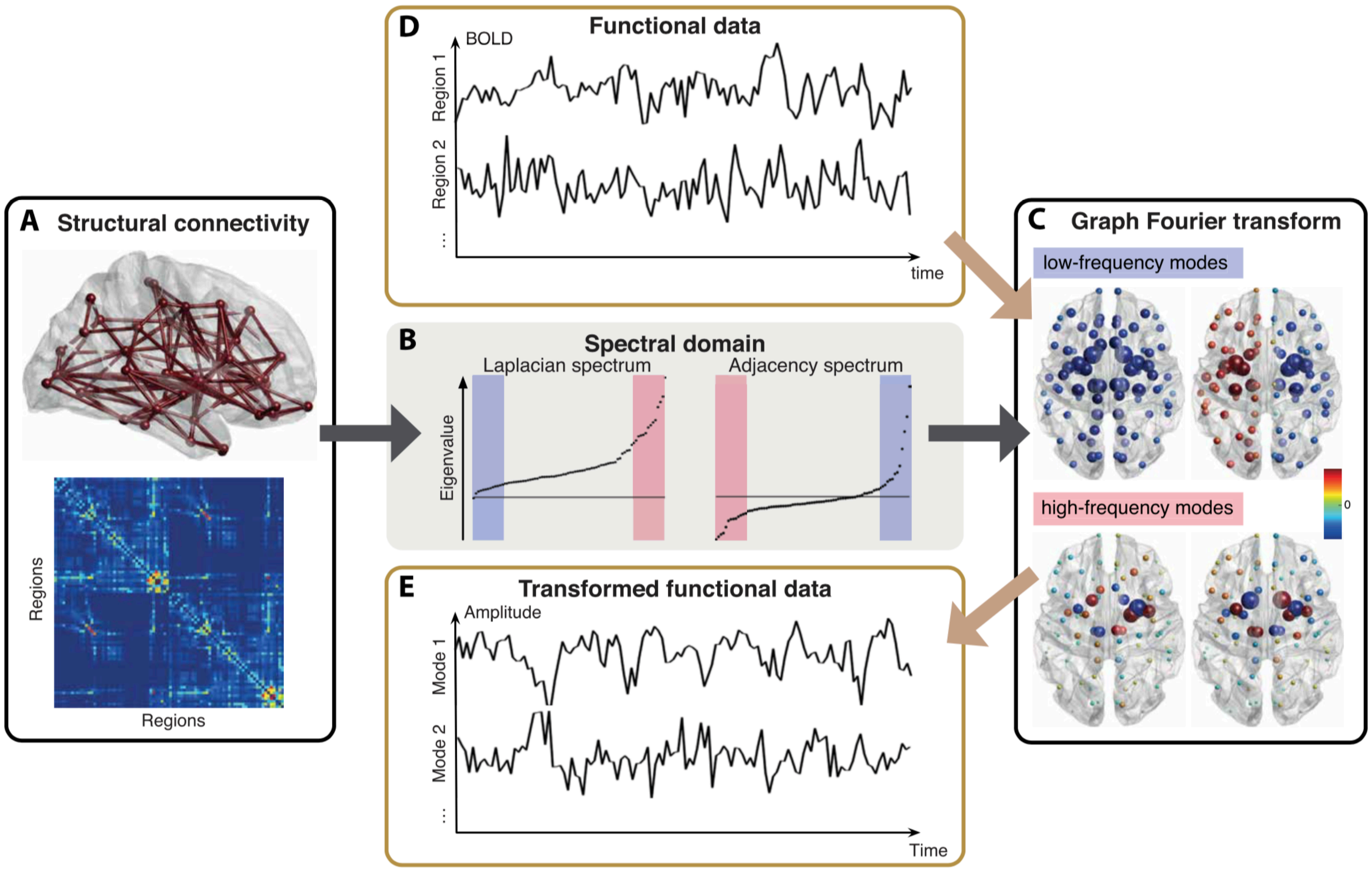


semi-supervised learning

Outline

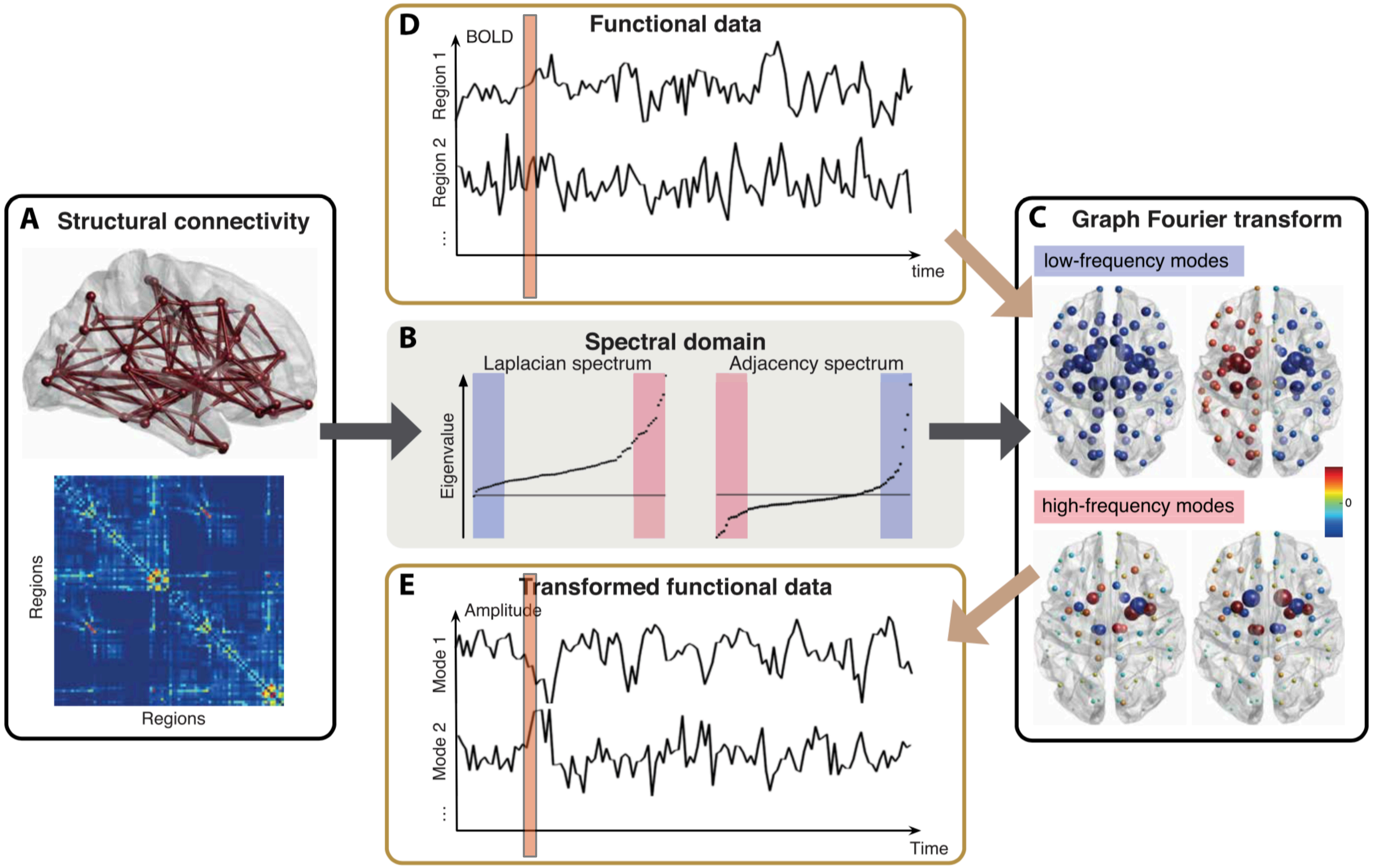
- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Applications in neuroscience

A typical analysis framework



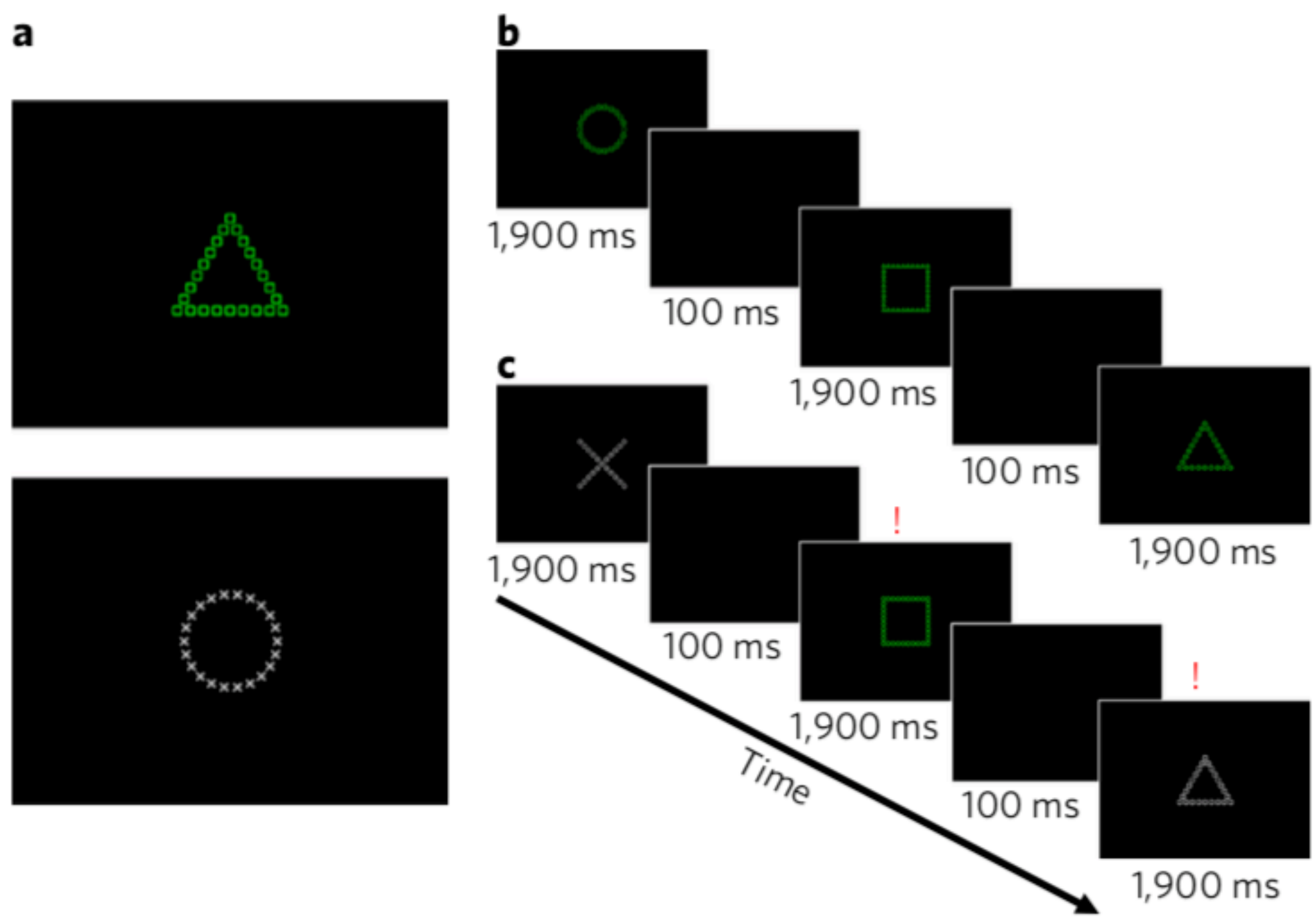
Huang et al., "A graph signal processing perspective on functional brain imaging", Proc. IEEE, 2018.

A typical analysis framework



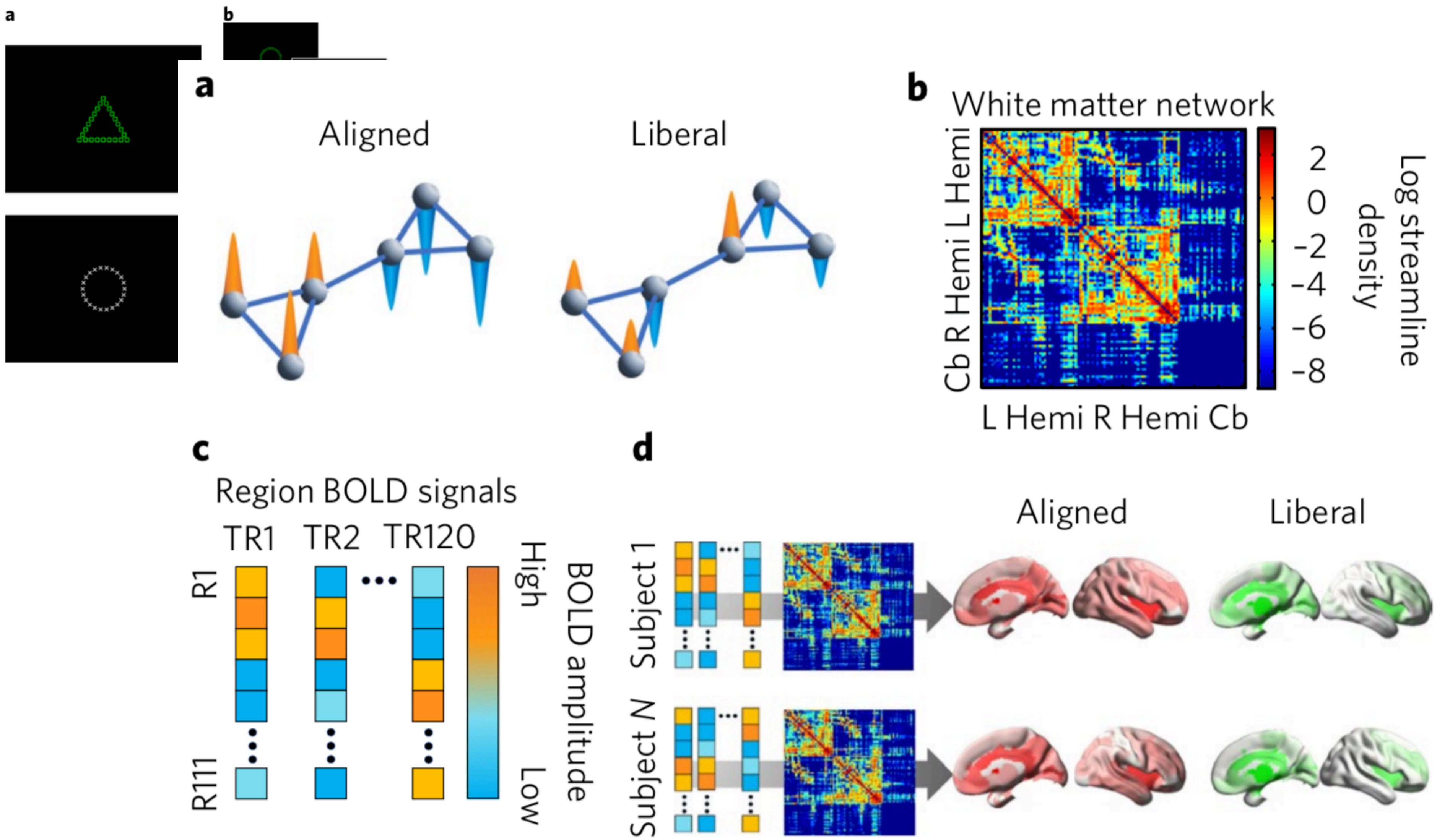
Huang et al., "A graph signal processing perspective on functional brain imaging", Proc. IEEE, 2018.

Application I: Understanding brain functioning



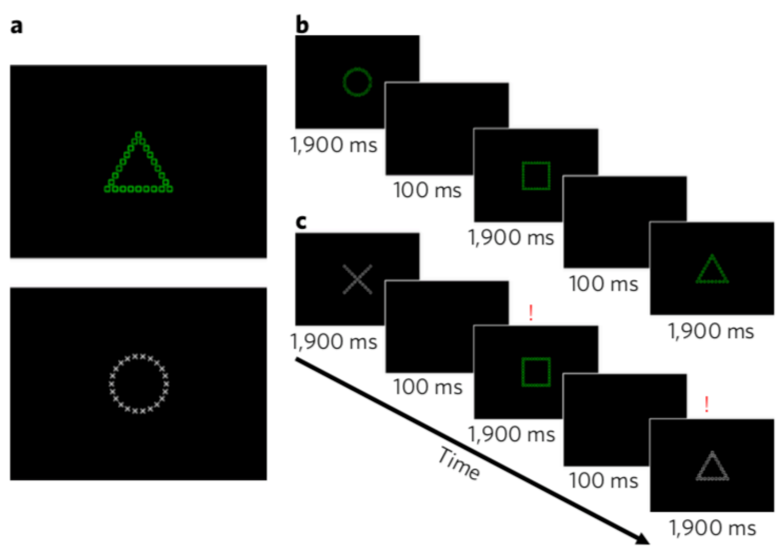
Medaglia et al., "Functional alignment with anatomical networks is associated with cognitive flexibility", Nature Human Behaviour, 2018.

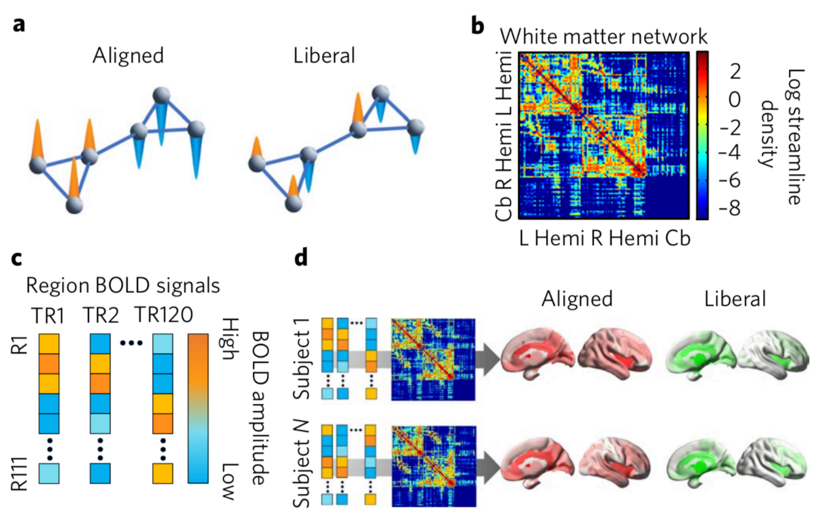
Application I: Understanding brain functioning

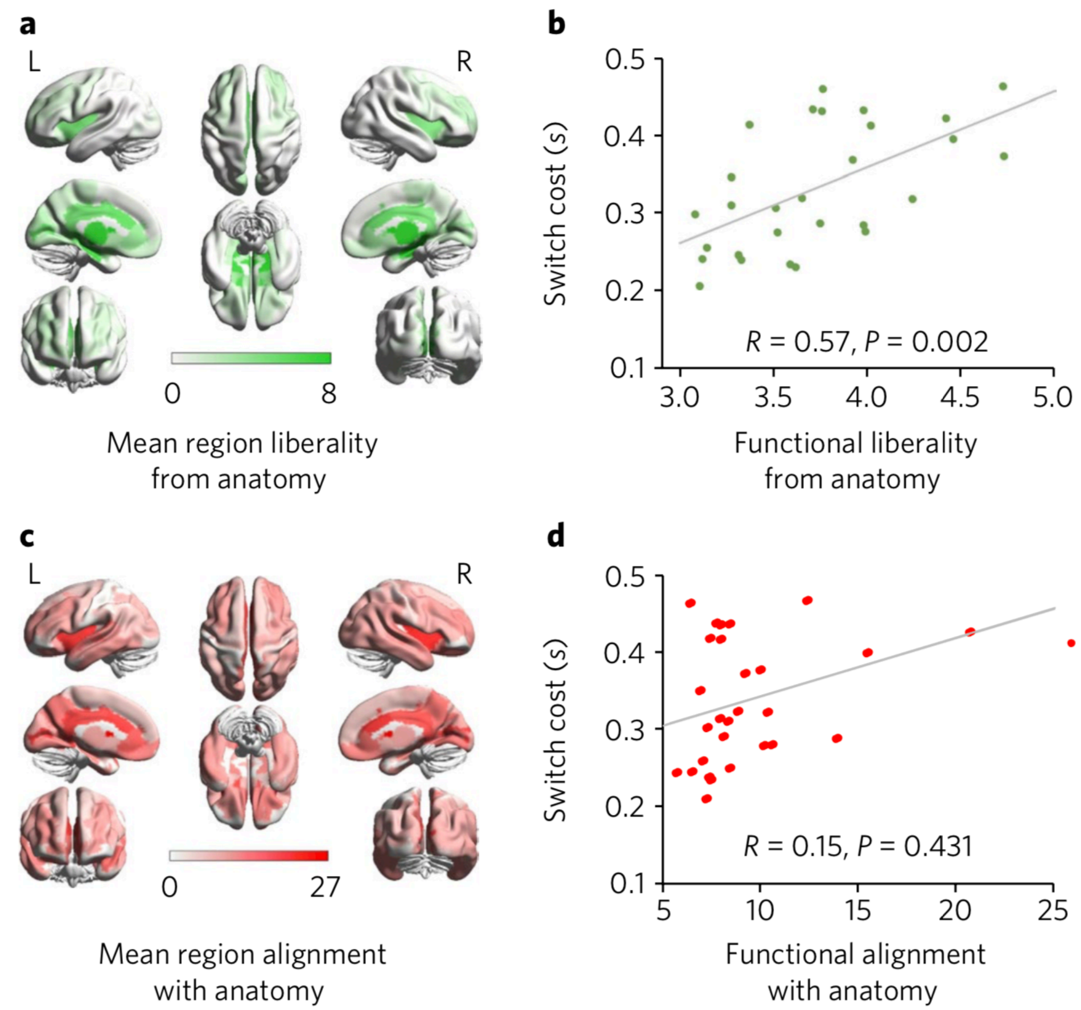


Medaglia et al., "Functional alignment with anatomical networks is associated with cognitive flexibility", Nature Human Behaviour, 2018.

Application I: Understanding brain functioning



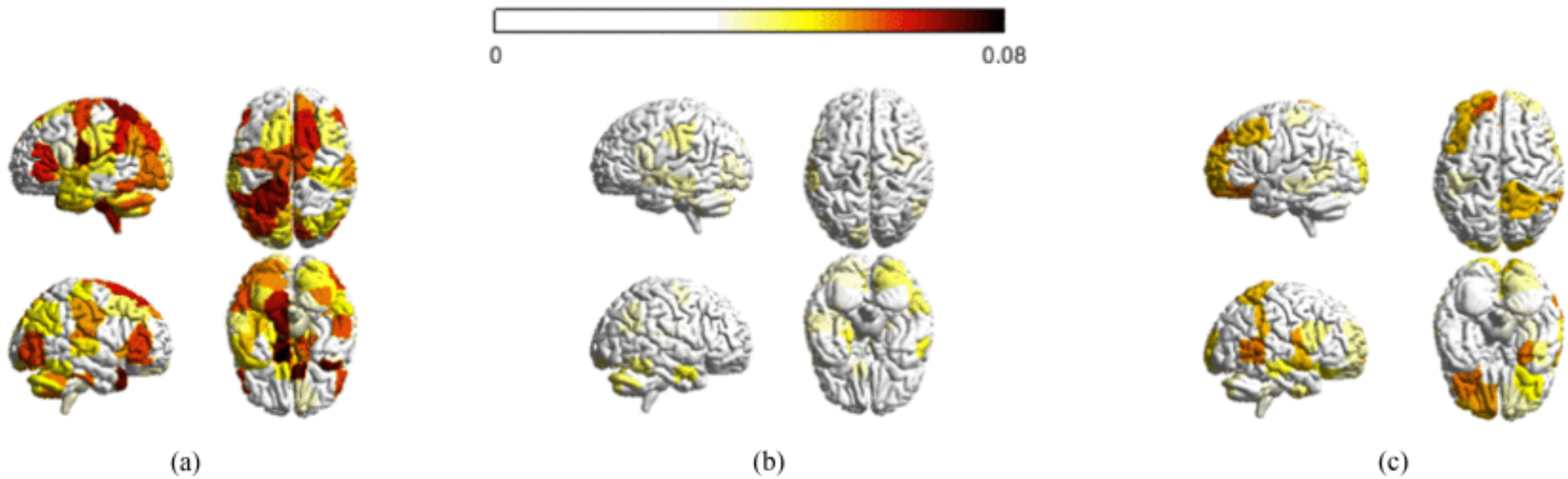




liberality (large high-freq. components) associated with high switching cost

Medaglia et al., "Functional alignment with anatomical networks is associated with cognitive flexibility", Nature Human Behaviour, 2018.

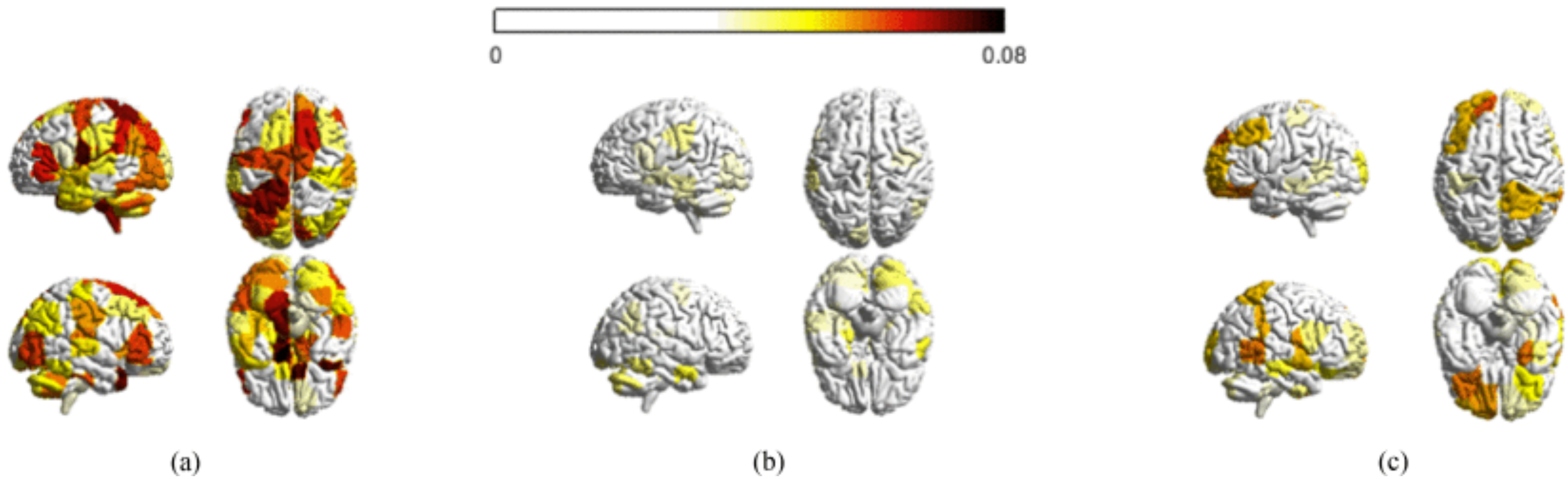
Application I: Understanding brain functioning



low-freq. components	**mid-freq. components**	**high-freq. components**

- record BOLD signals while responding to sequentially presented stimuli

Application I: Understanding brain functioning

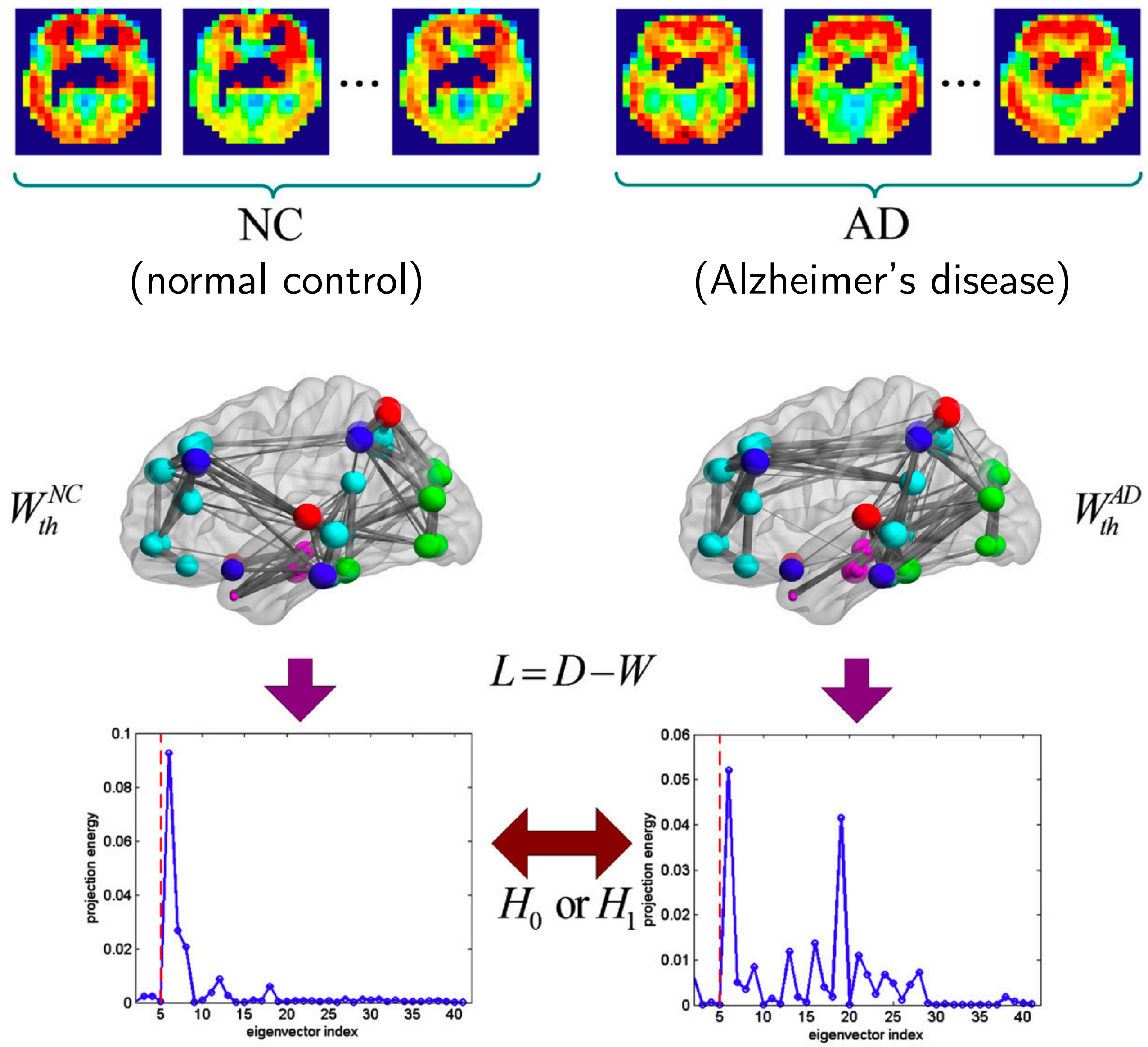


low-freq. components | **mid-freq. components** | **high-freq. components**

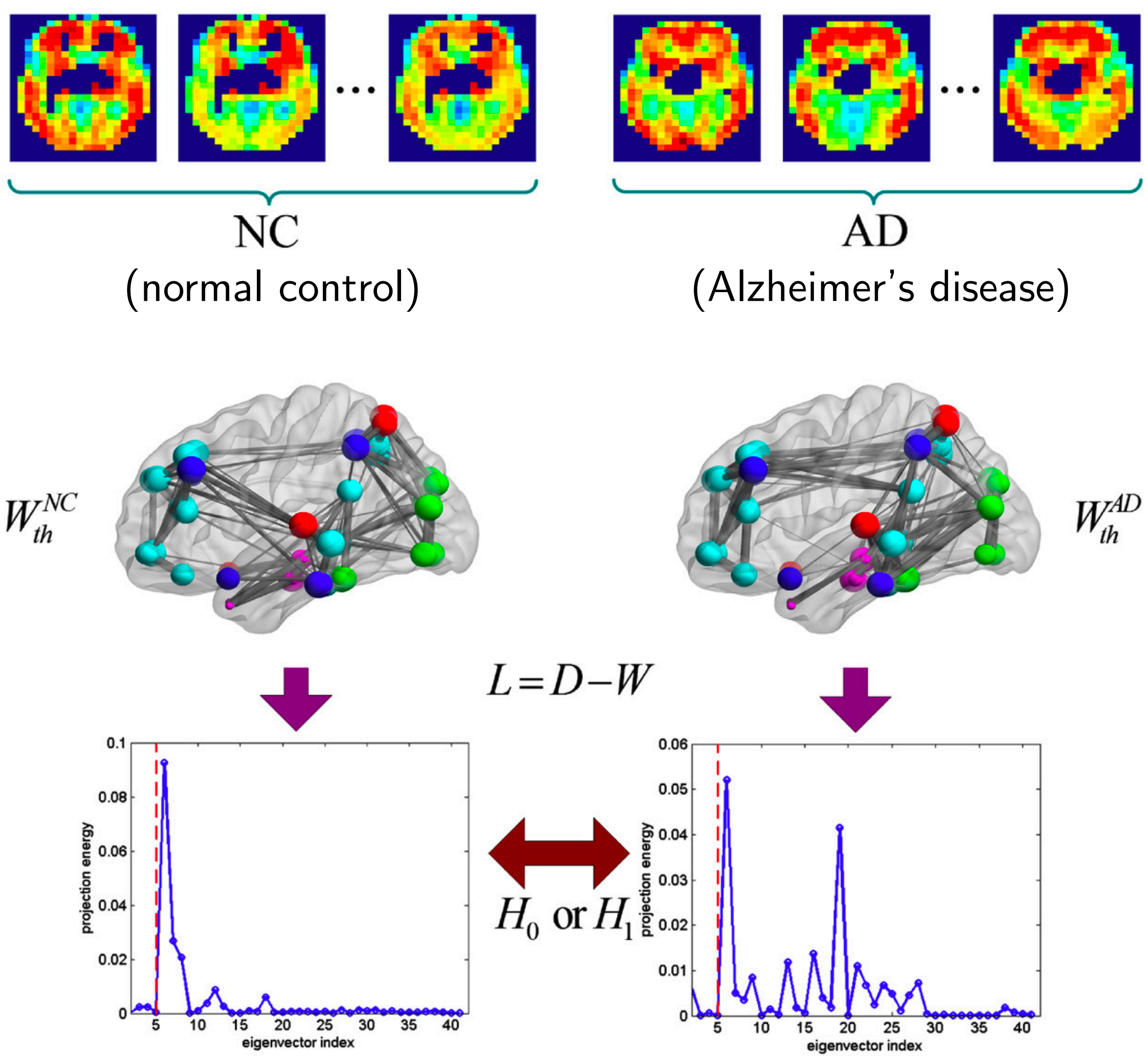
- record BOLD signals while responding to sequentially presented stimuli
- it favours learning to have
  - **smooth, spread** signals (low-freq.) when facing **unfamiliar** task
  - **varied, spiking** signals (high-freq.) when task becomes **familiar**

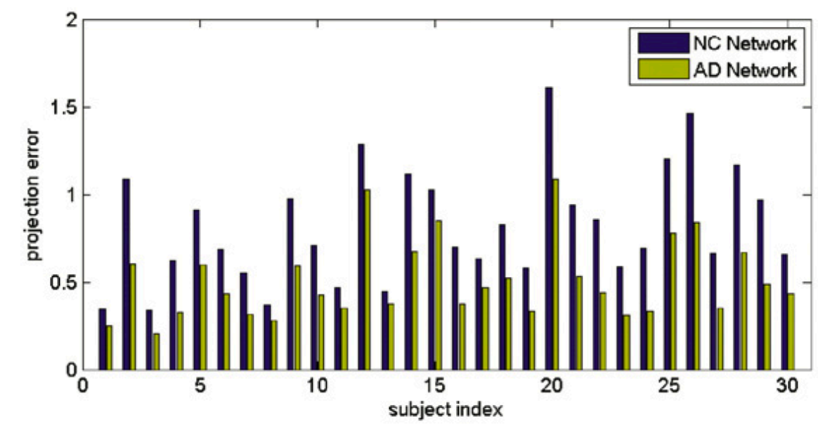
Huang et al., "Graph frequency analysis of brain signals", IEEE JSTSP, 2016.

Application II: Disease classification

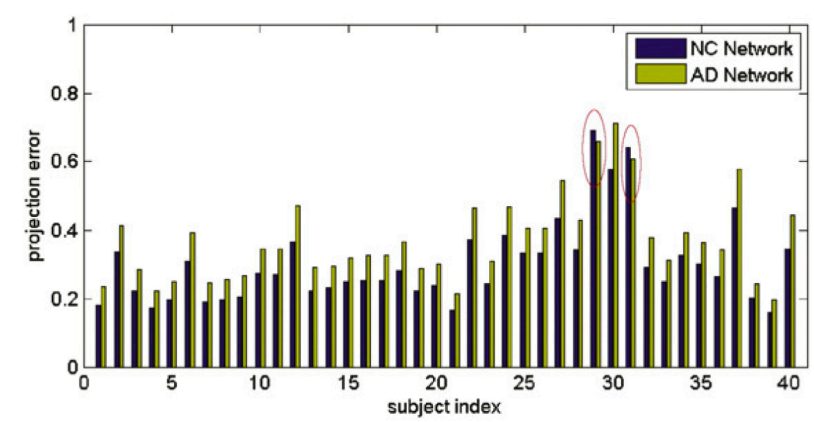


Application II: Disease classification





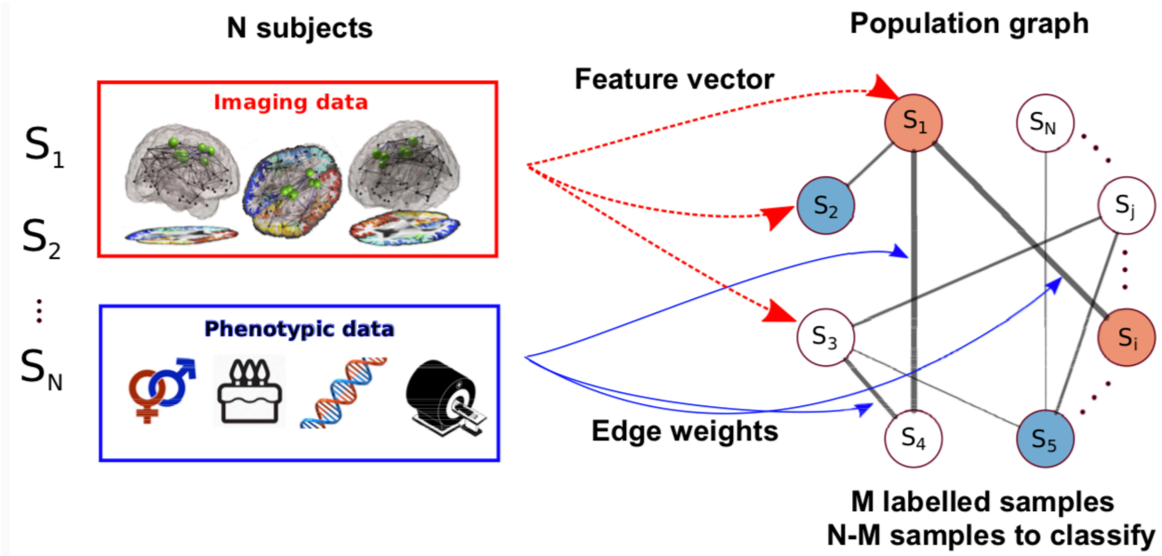
(a) AD group projection error



(b) NC group projection error

Hu et al., "Matched signal detection on graphs", NeuroImage, 2016.

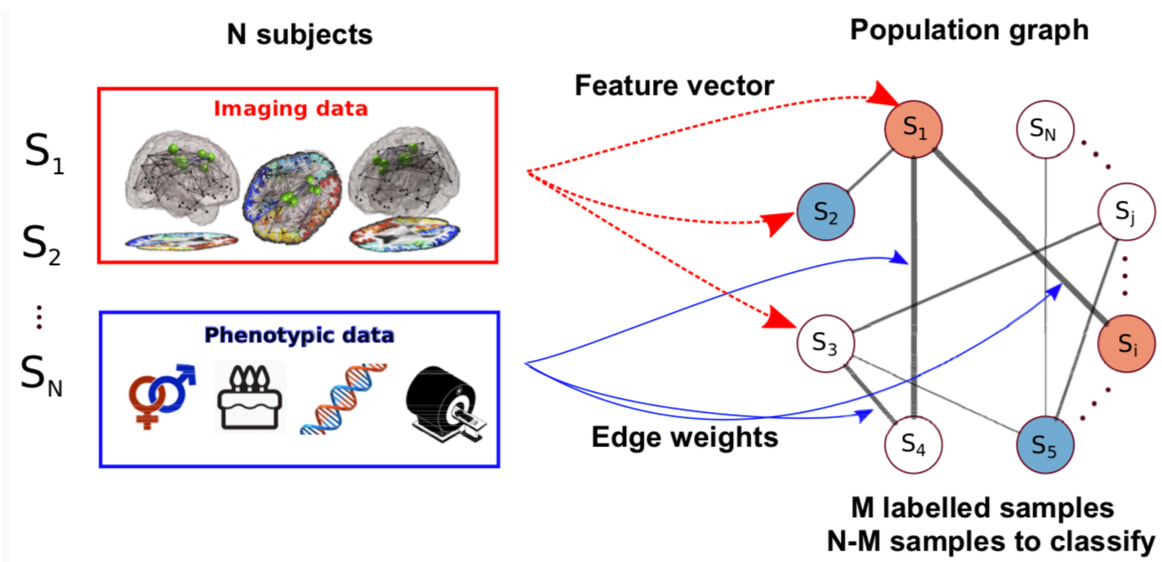
Application II: Disease classification



Parisot et al., "Disease prediction using graph convolutional networks", Medical Image Analysis, 2018.

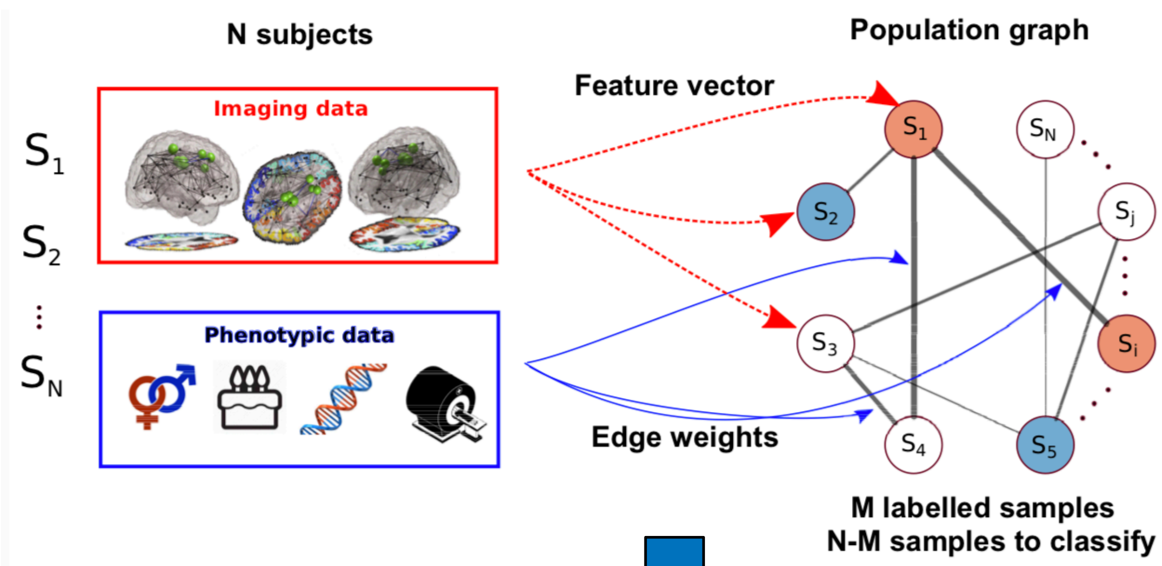
Application II: Disease classification

ADNI (structural MRI): volumes of brain structures
ABIDE (fMRI): off-diagonal of functional connectivity



Application II: Disease classification

ADNI (structural MRI): volumes of brain structures
ABIDE (fMRI): off-diagonal of functional connectivity

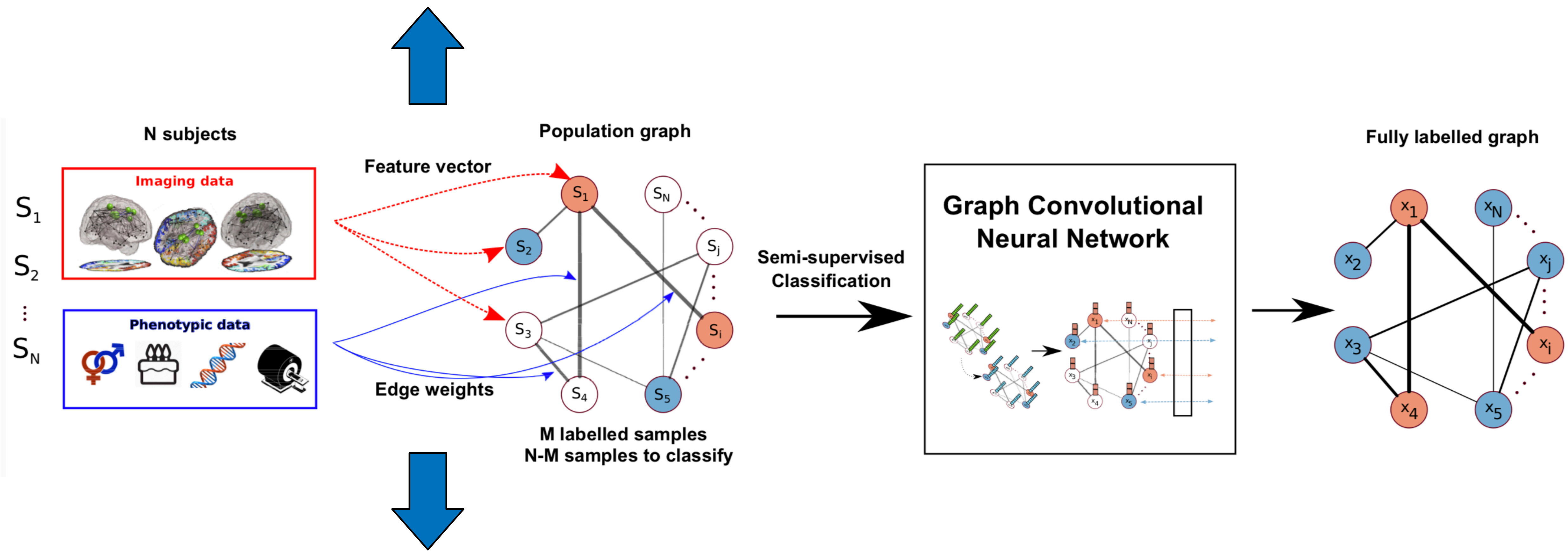


similarity in phenotypic data

Parisot et al., "Disease prediction using graph convolutional networks", Medical Image Analysis, 2018.

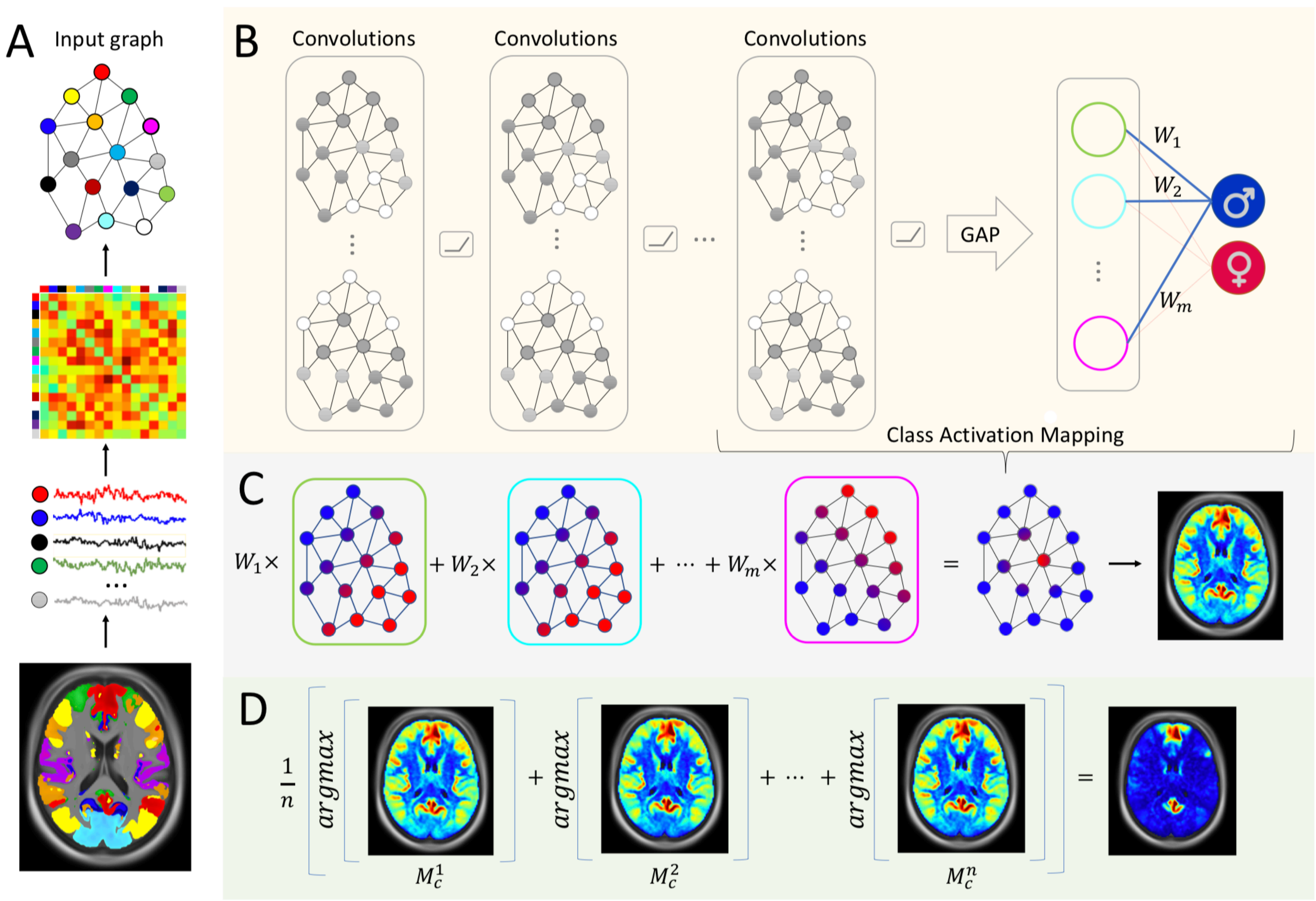
Application II: Disease classification

ADNI (structural MRI): volumes of brain structures
ABIDE (fMRI): off-diagonal of functional connectivity



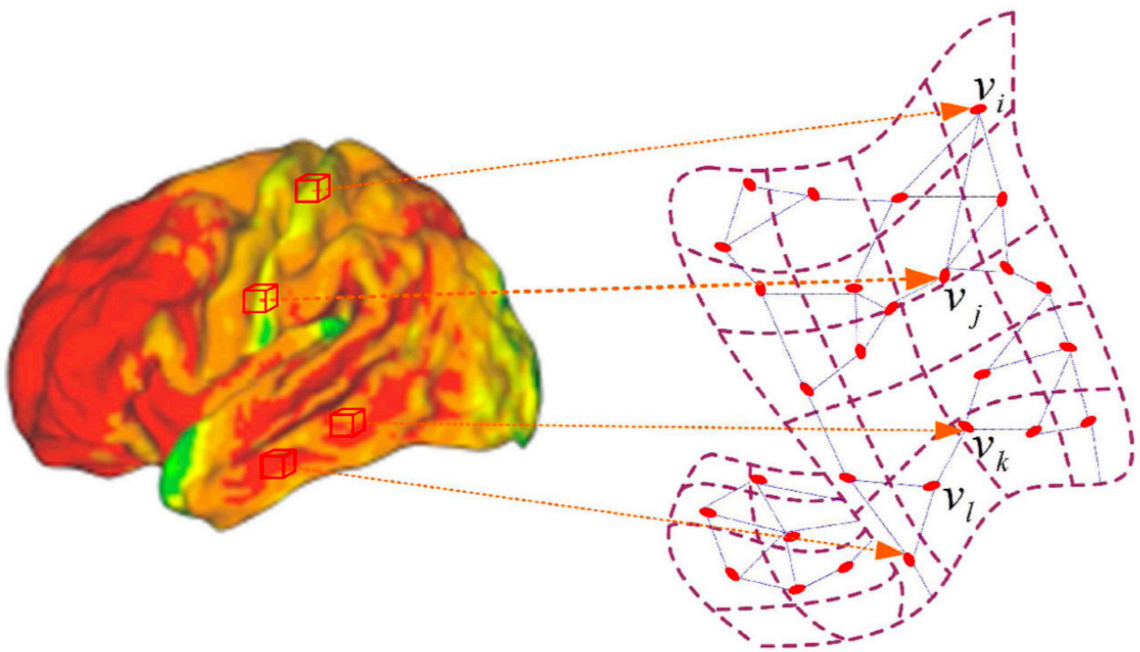
similarity in phenotypic data

Application III: Gender classification

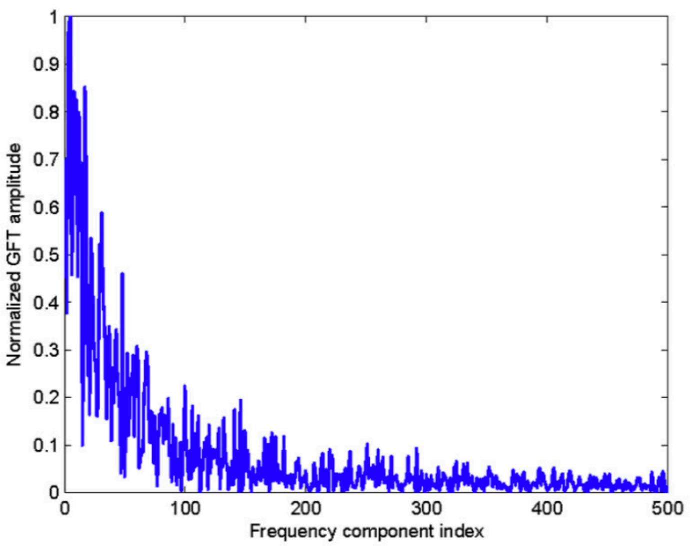


Arslan et al., "Graph saliency maps through spectral convolutional networks", GRAIL, 2018.

Application IV: Inferring brain connectivity



(a) Mapping from image voxels (cubes) to vertices (ellipses) of a graph.

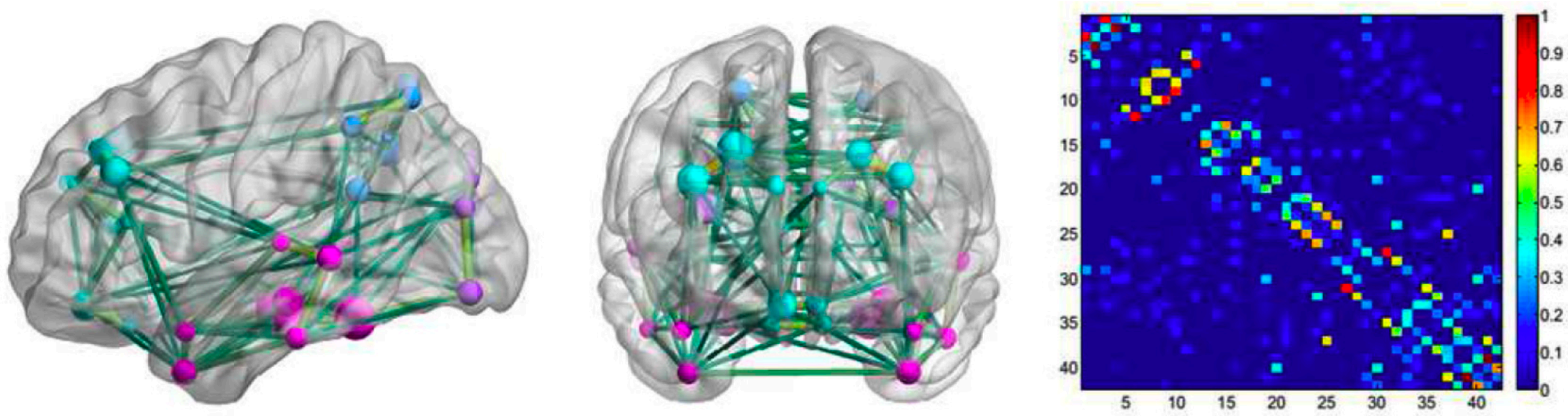


(b) Amplitudes of the GFT coefficients.

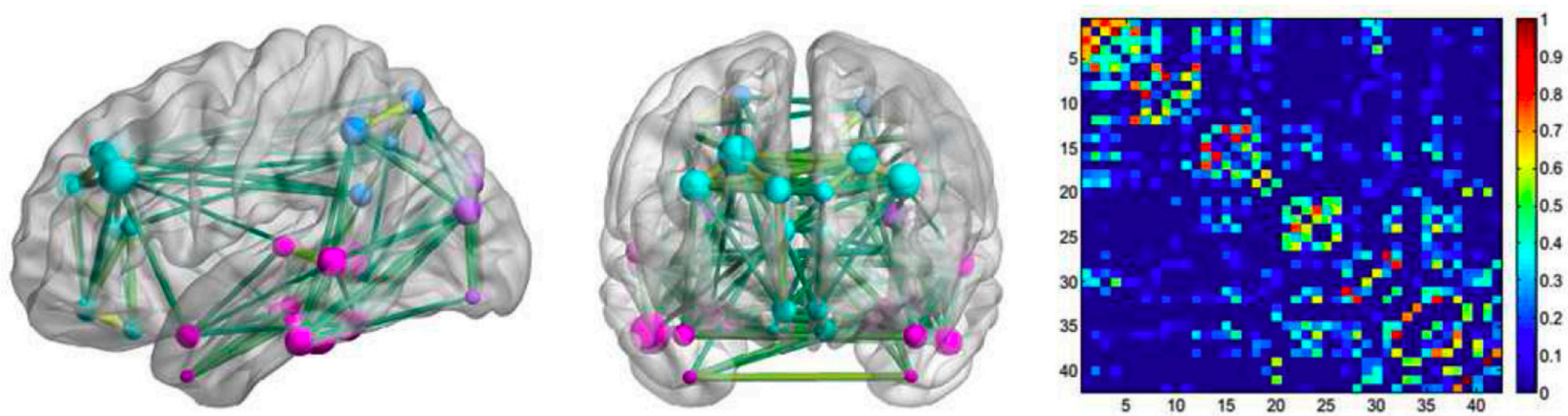
Hu et al., "A spectral graph regression model for learning brain connectivity of Alzheimer's disease", PLOS ONE, 2015.
Shen et al., "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity", 2016.

Application IV: Inferring brain connectivity

Alzheimer's disease



normal control



Hu et al., "A spectral graph regression model for learning brain connectivity of Alzheimer's disease", PLOS ONE, 2015.
Shen et al., "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity", 2016.

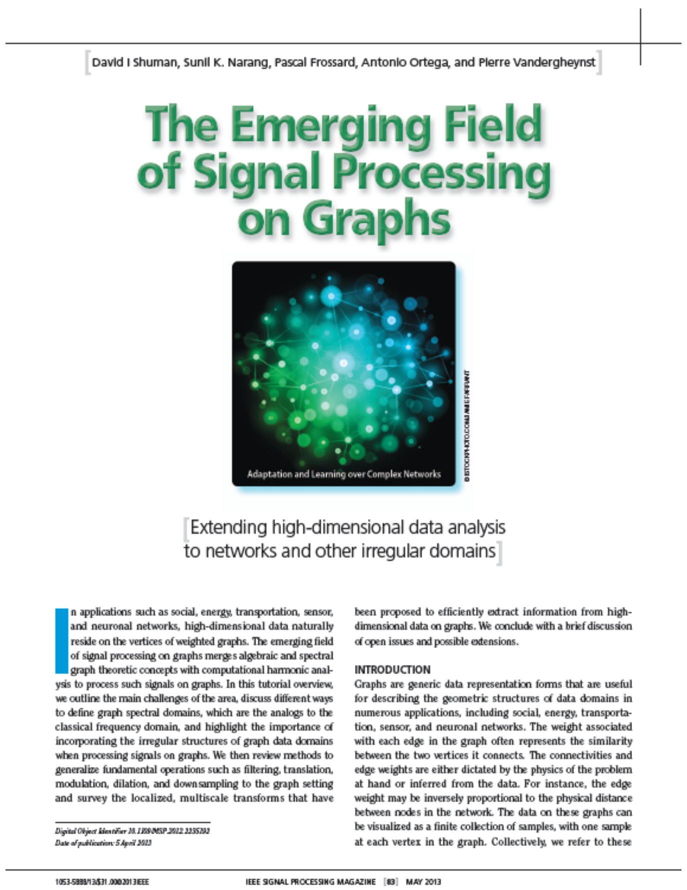
Future of GSP

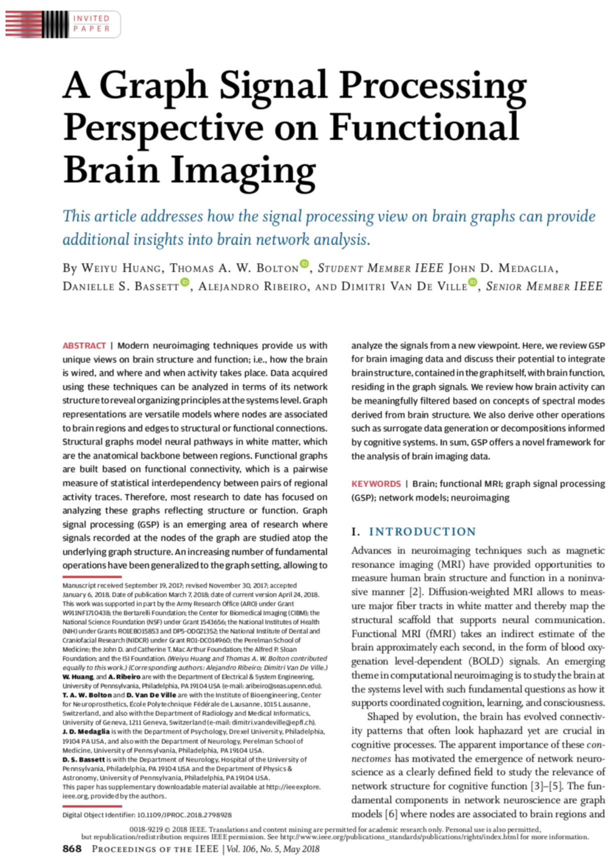
- Mathematical models for graph signals
  - global and local smoothness / regularity
  - underlying physical processes
- Graph construction
  - how to infer topologies given observed data?
- Fast implementation
  - fast graph Fourier transform
  - distributed processing
- Connection to / combination with other fields
  - statistical machine learning
  - deep learning on graphs and manifolds
- Key applications

Bronstein et al., "Geometric deep learning", IEEE SPM, 2017.
Wu et al., "A comprehensive survey on graph neural networks", arXiv, 2019.

Resources

- Three tutorial/overview papers:





- More available at: http://web.media.mit.edu/~xdong/resource.html