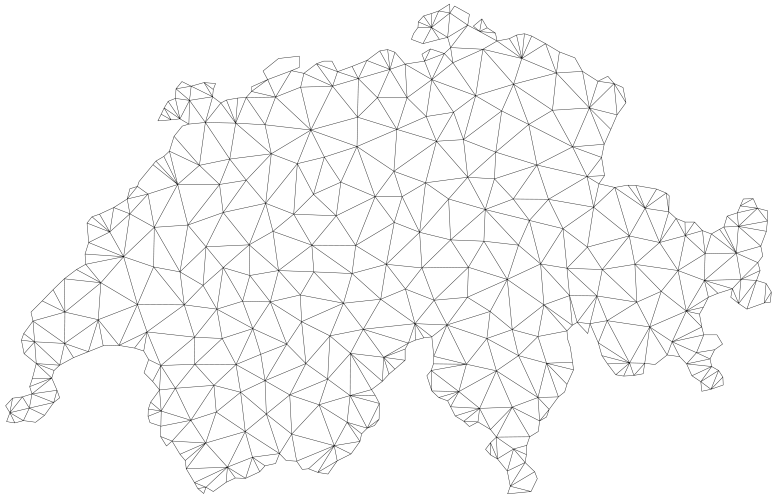
Signal Processing and Machine Learning on Graphs

Xiaowen Dong
Department of Engineering Science
University of Oxford

Summer School in Economic Networks
Oxford, June 2019

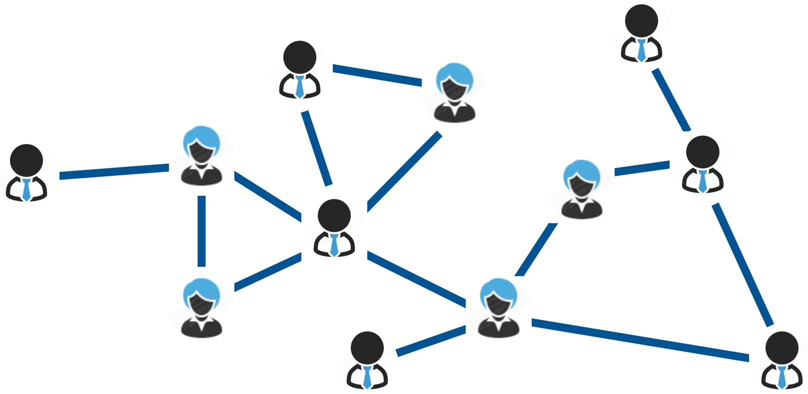
UNIVERSITY OF OXFORD

Networks are everywhere

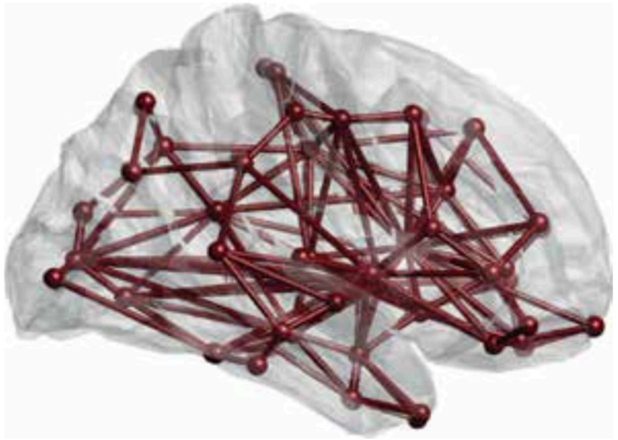


geographical network

traffic network

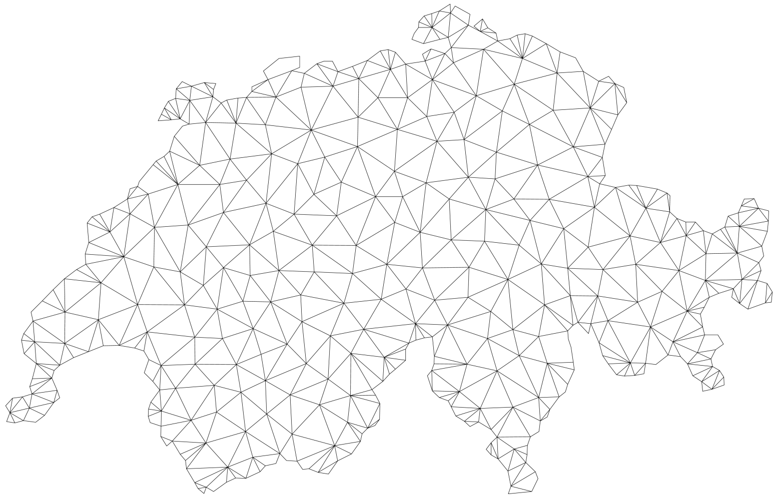


social network



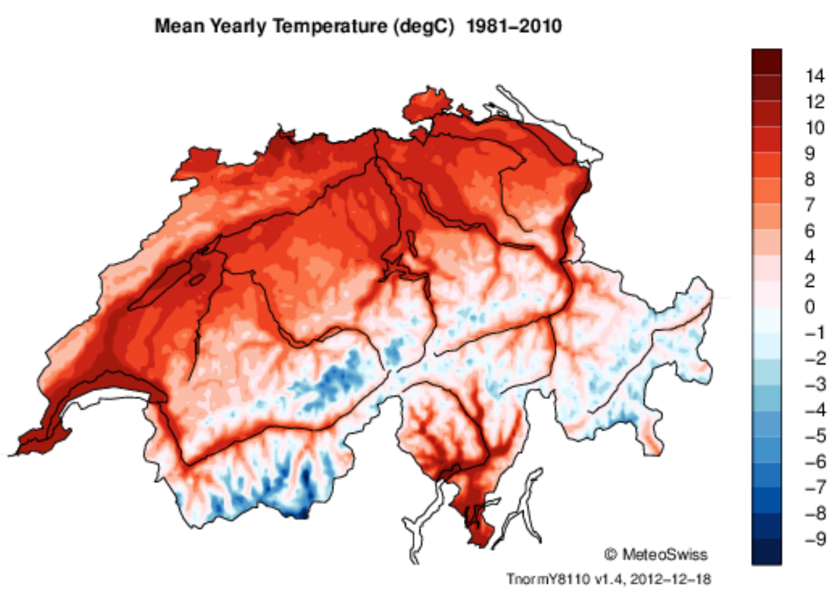
brain network [Huang18]

Network-structured data are everywhere



- vertices
  - geographical regions
- edges
  - geographical proximity between regions

Network-structured data are everywhere

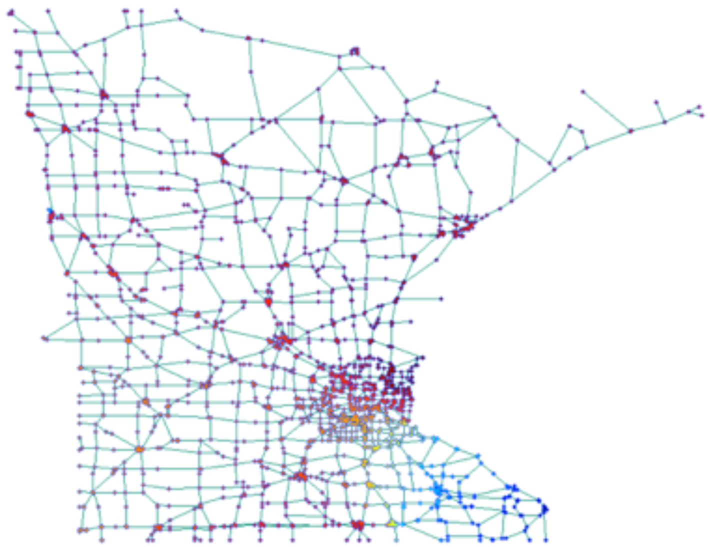


- vertices
  - geographical regions
- edges
  - geographical proximity between regions
- signal
  - temperature records in these regions

Network-structured data are everywhere

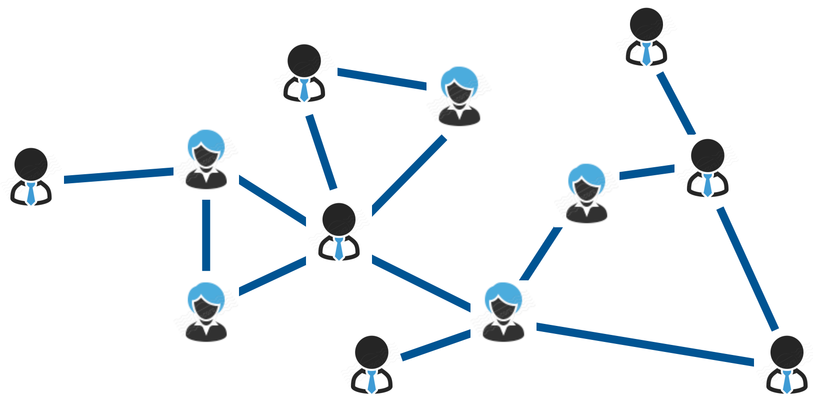
- vertices
  - road junctions
- edges
  - road network

Network-structured data are everywhere



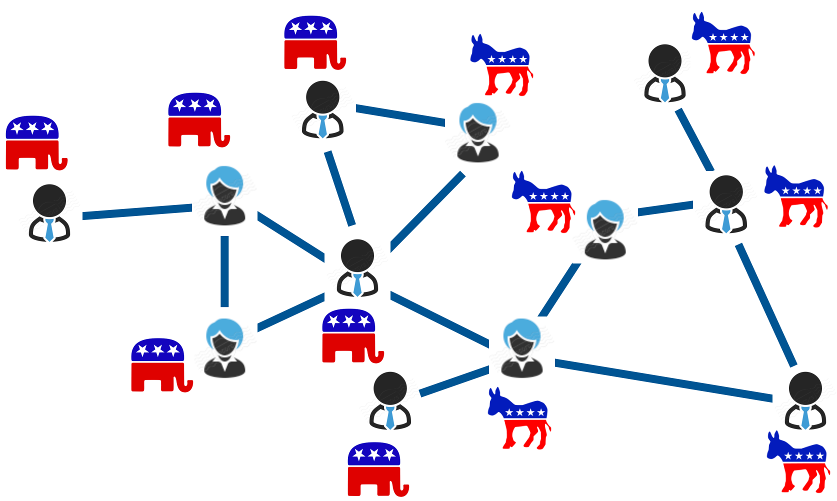
- vertices
  - road junctions
- edges
  - road network
- signal
  - traffic congestion at junctions

Network-structured data are everywhere



- vertices
  - individuals
- edges
  - friendship between individuals

Network-structured data are everywhere

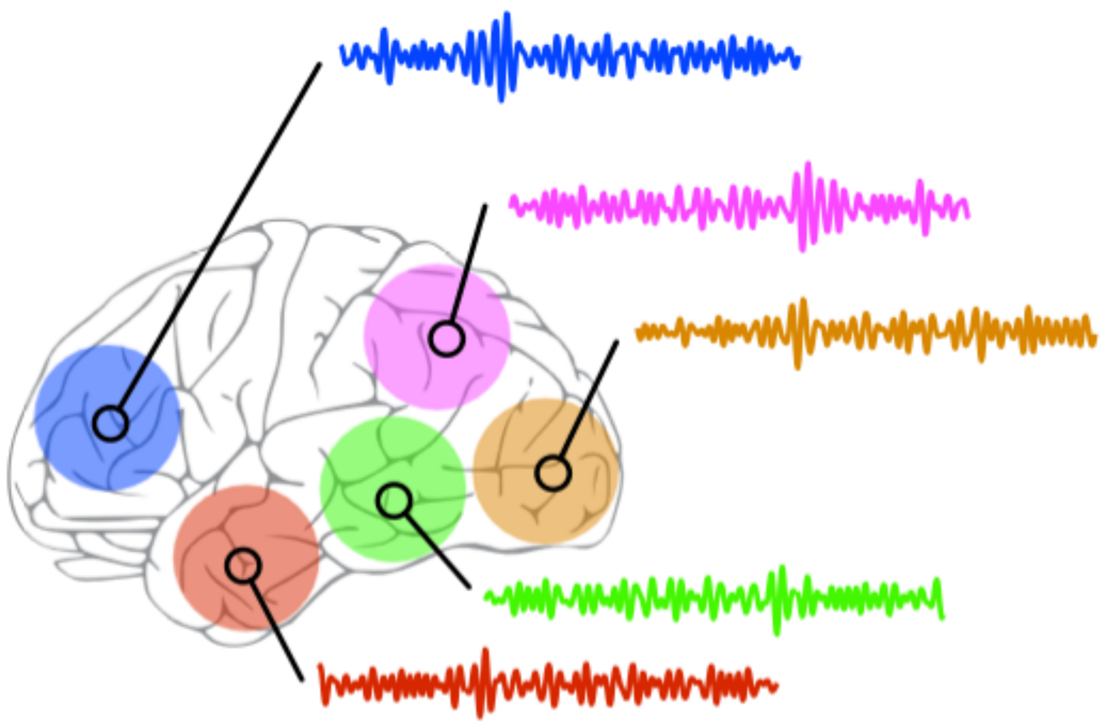


- vertices
  - individuals
- edges
  - friendship between individuals
- signal
  - political view

Network-structured data are everywhere

- vertices
  - brain regions
- edges
  - structural connectivity between brain regions

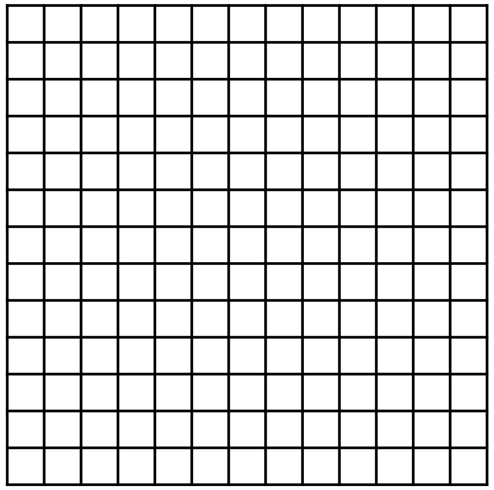
Network-structured data are everywhere



- vertices
  - brain regions
- edges
  - structural connectivity between brain regions
- signal
  - blood-oxygen-level-dependent (BOLD) time series

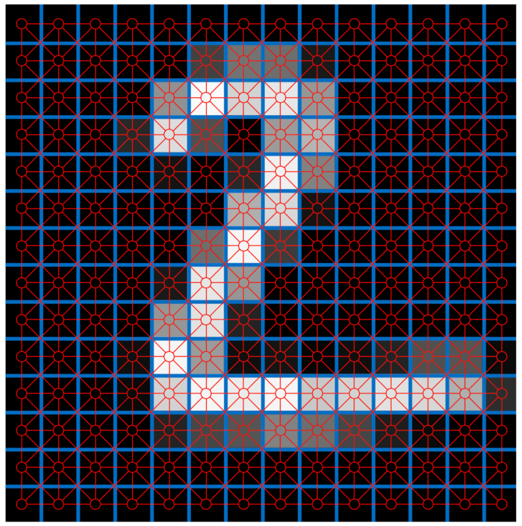
Richiardi et al., "Machine learning with brain graphs," IEEE SPM, 2013.

Network-structured data are everywhere



- vertices
  - pixels
- edges
  - spatial proximity between pixels

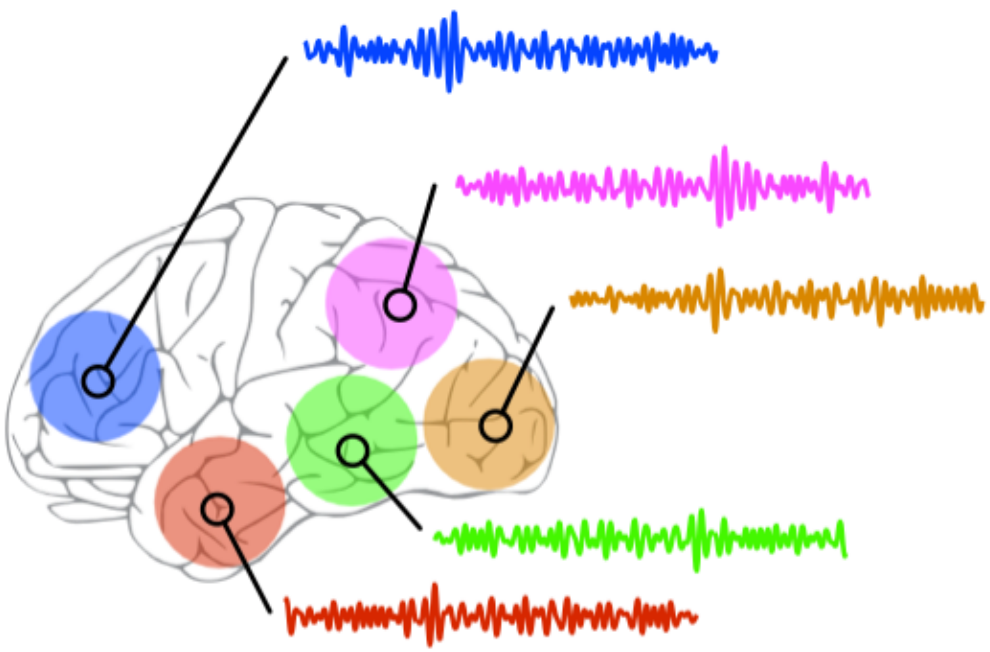
Network-structured data are everywhere



- vertices
  - pixels
- edges
  - spatial proximity between pixels
- signal
  - pixel values

Monti et al., "Geometric deep learning on graphs and manifolds using mixture model CNNs," CVPR, 2017.

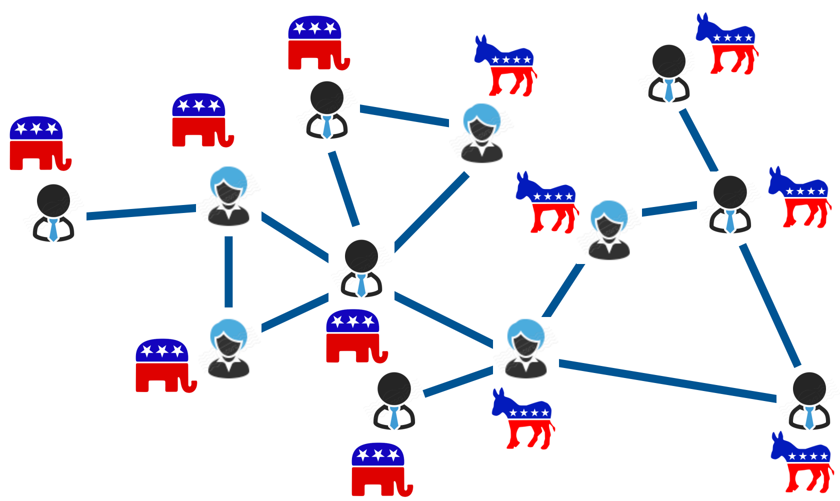
Learning with network-structured data



medical condition?
no medical condition?

network-wise classification

Learning with network-structured data



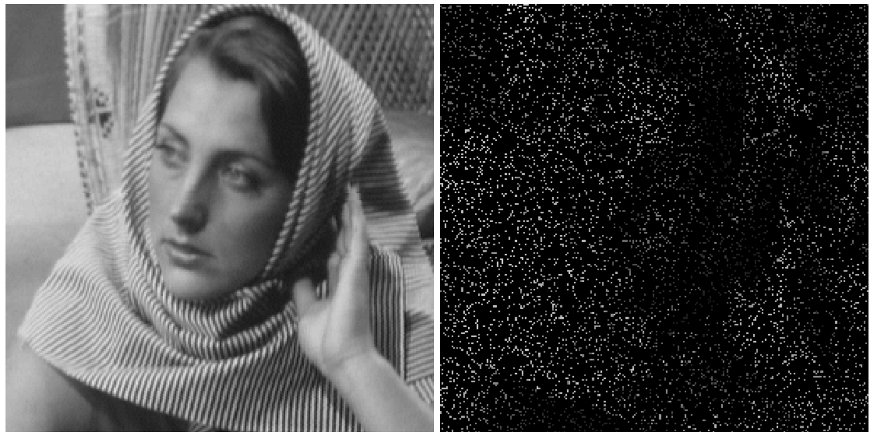
democratic?
republican?

vertex-wise classification

Learning with network-structured data

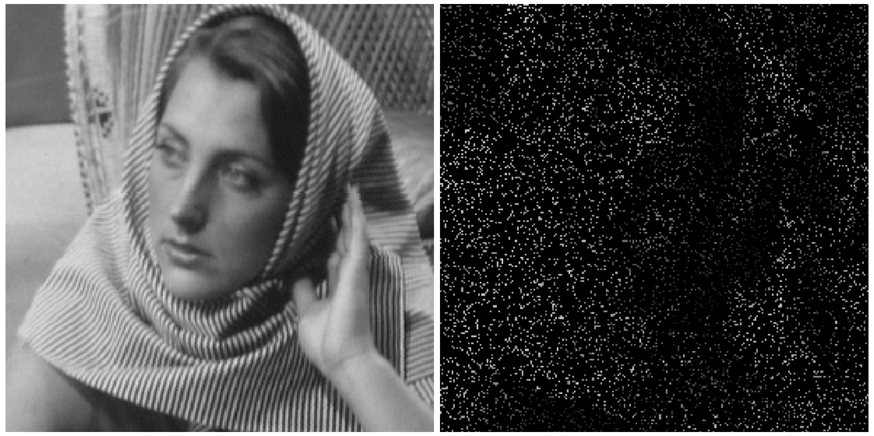
denoising and inpainting (inspired by image processing)

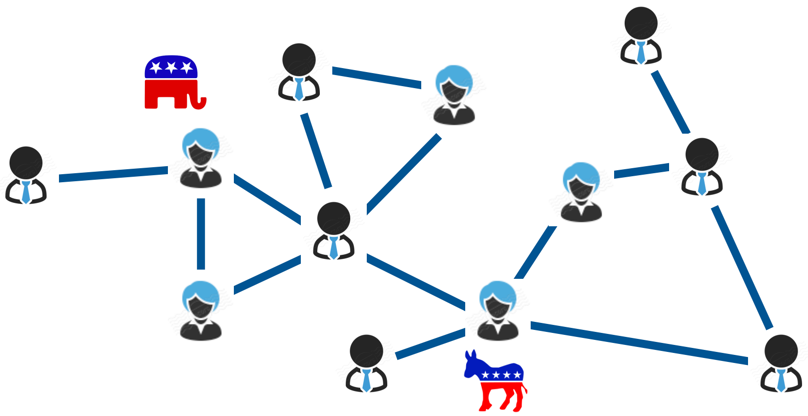
Learning with network-structured data



denoising and inpainting (inspired by image processing)

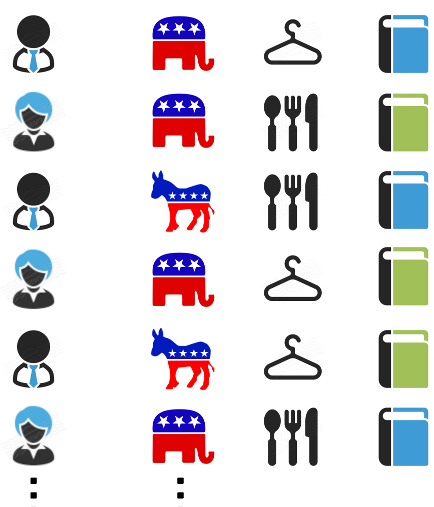
Learning with network-structured data





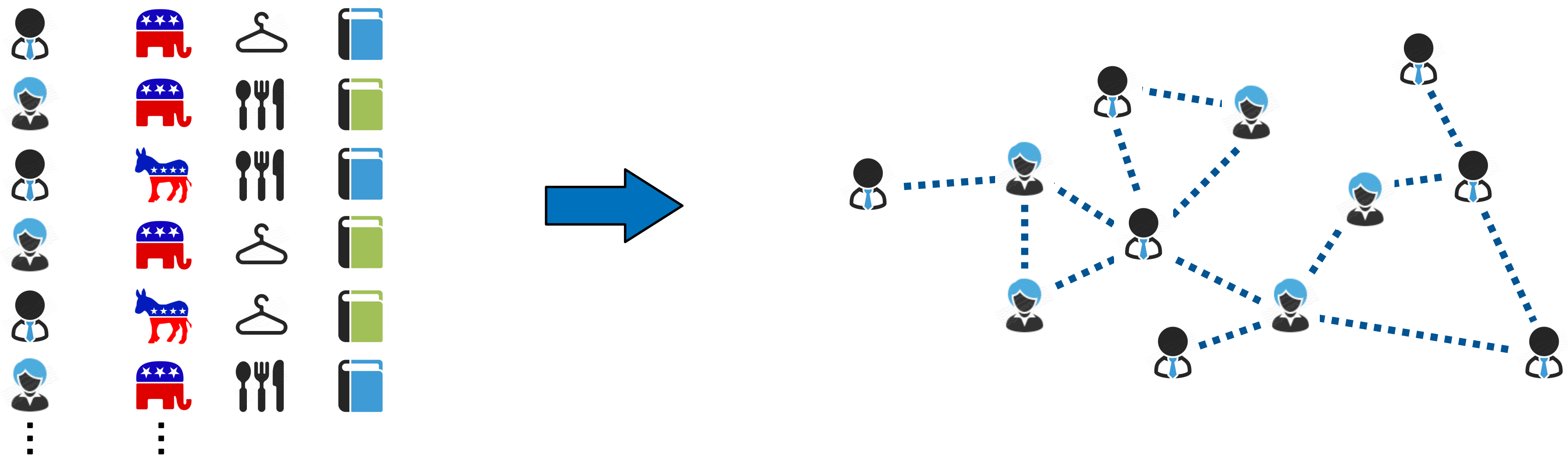
denoising and inpainting (inspired by image processing)

Learning with network-structured data



inferring network structure from data

Learning with network-structured data



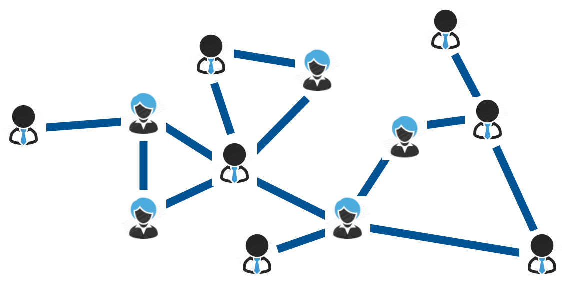
inferring network structure from data

How to incorporate network into learning?

- Straightforward approach: embed the network into a Euclidean space

How to incorporate network into learning?

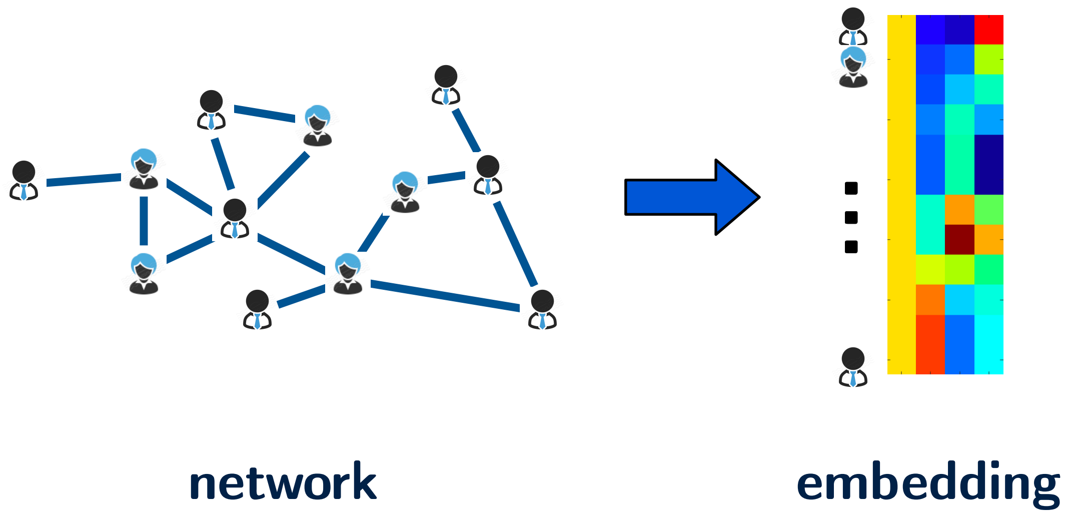
- Straightforward approach: embed the network into a Euclidean space



network

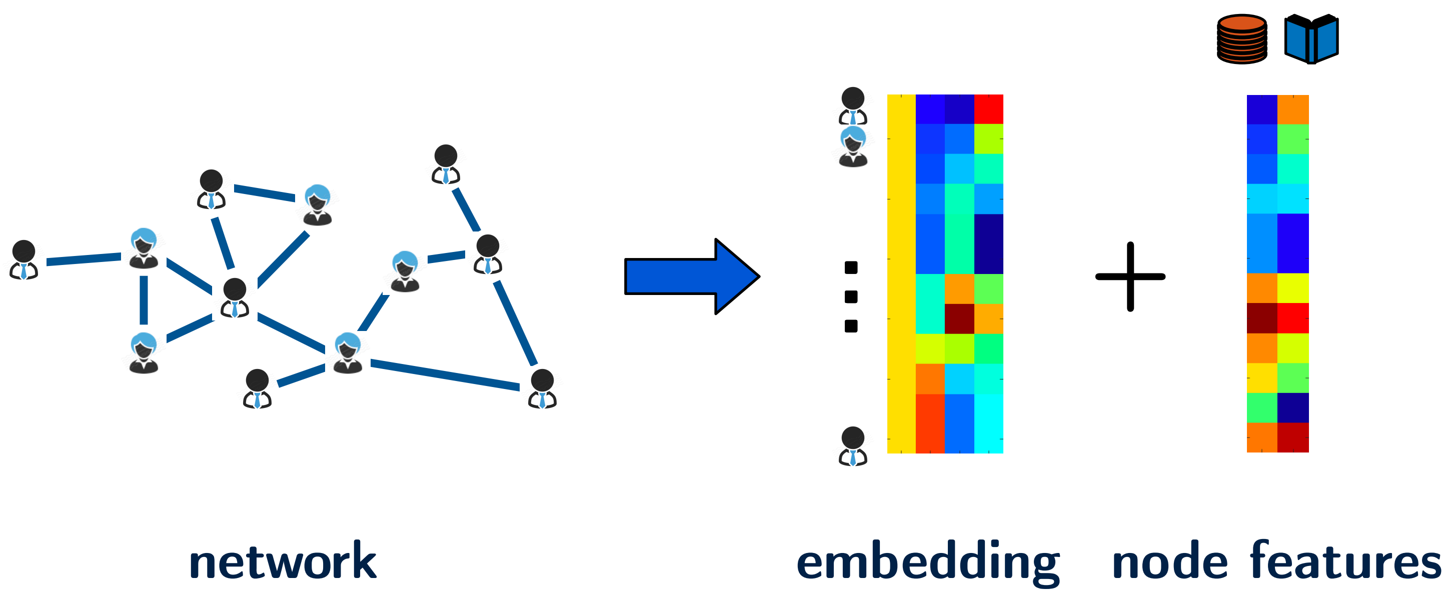
How to incorporate network into learning?

- Straightforward approach: embed the network into a Euclidean space



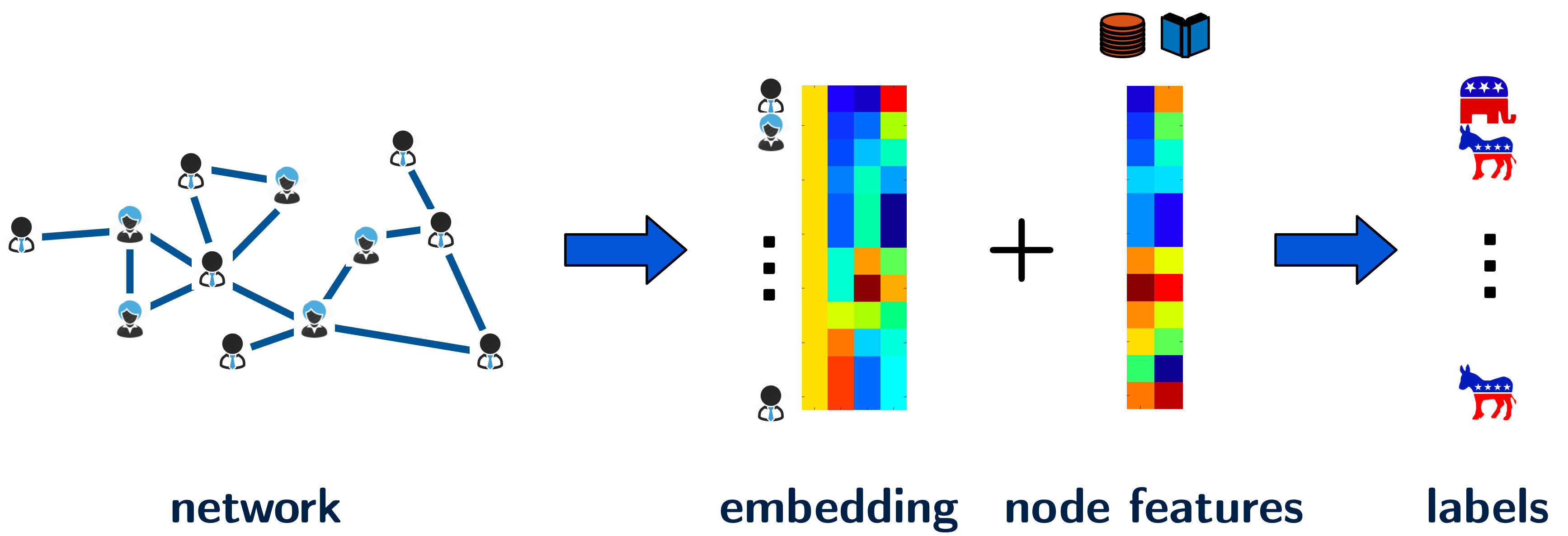
How to incorporate network into learning?

- Straightforward approach: embed the network into a Euclidean space



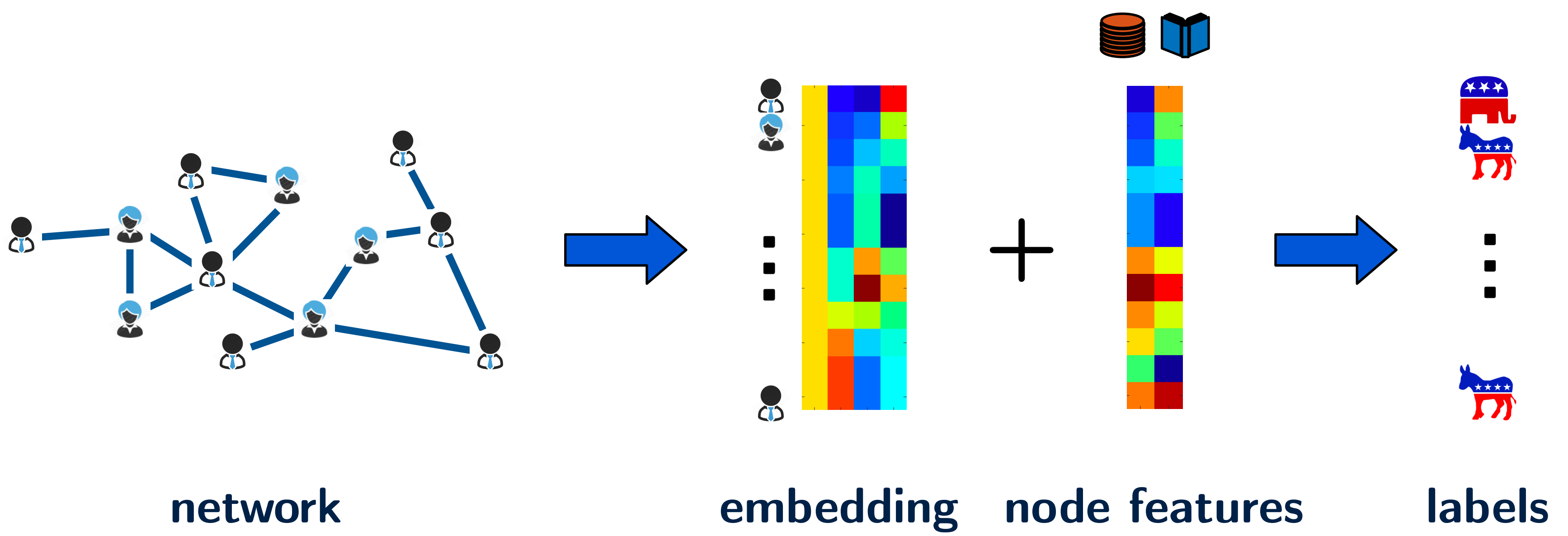
How to incorporate network into learning?

- Straightforward approach: embed the network into a Euclidean space



How to incorporate network into learning?

- Straightforward approach: embed the network into a Euclidean space



- embedding of network structure leads to information loss
- need for new models & tools that directly incorporate structure in data analysis

Outline

- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

Outline

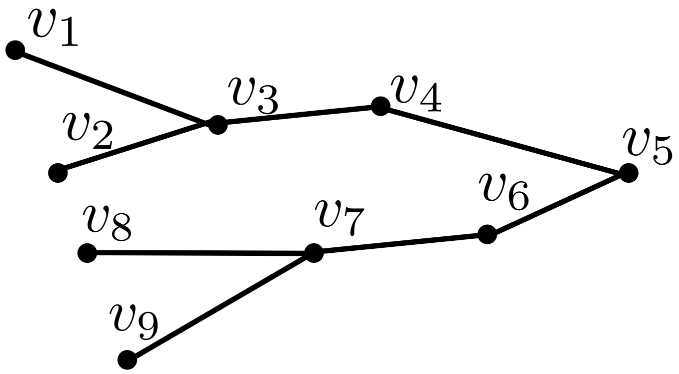
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

Graph signal processing

- Network-structured data can be represented by graph signals

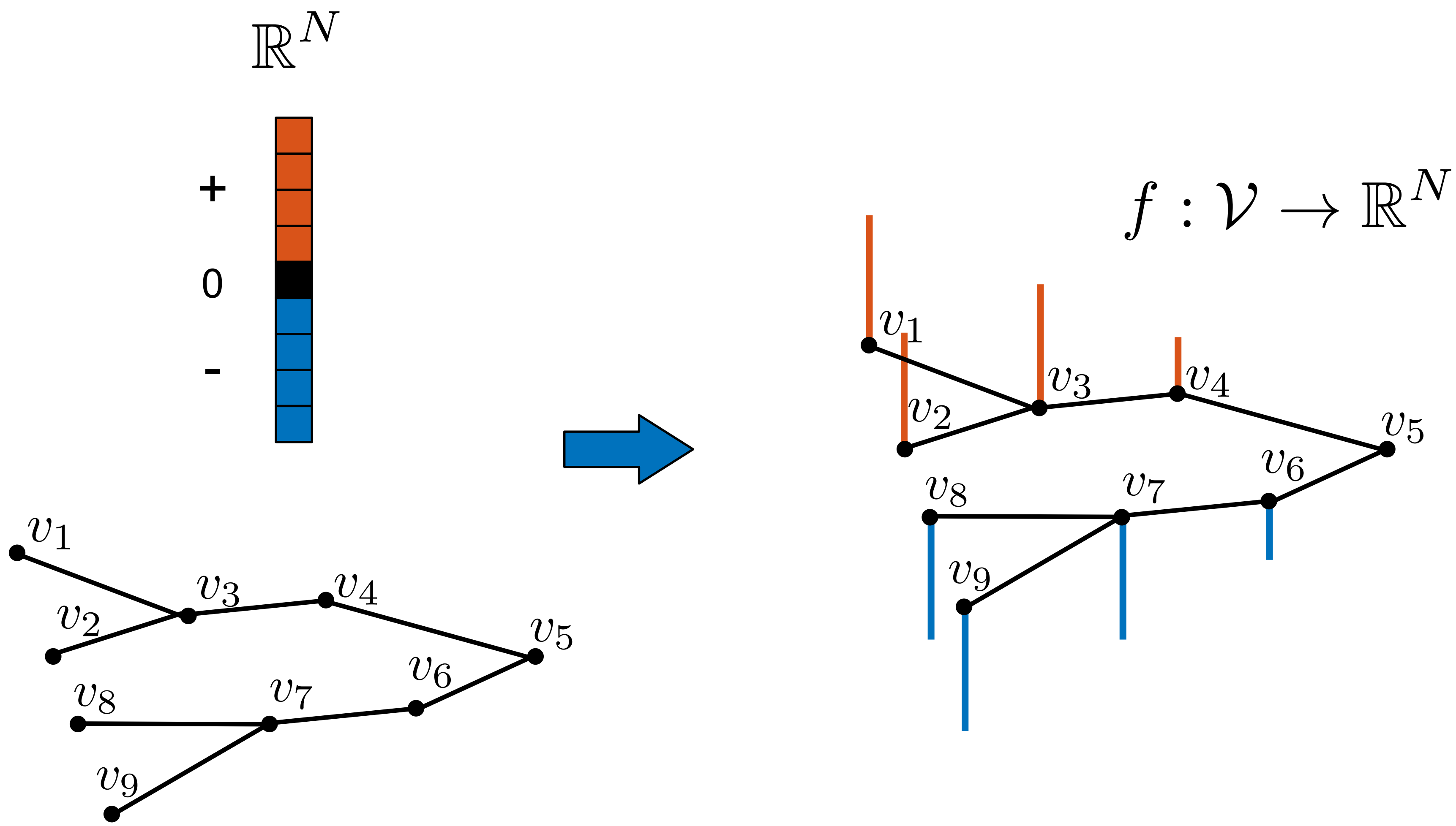
Graph signal processing

- Network-structured data can be represented by graph signals



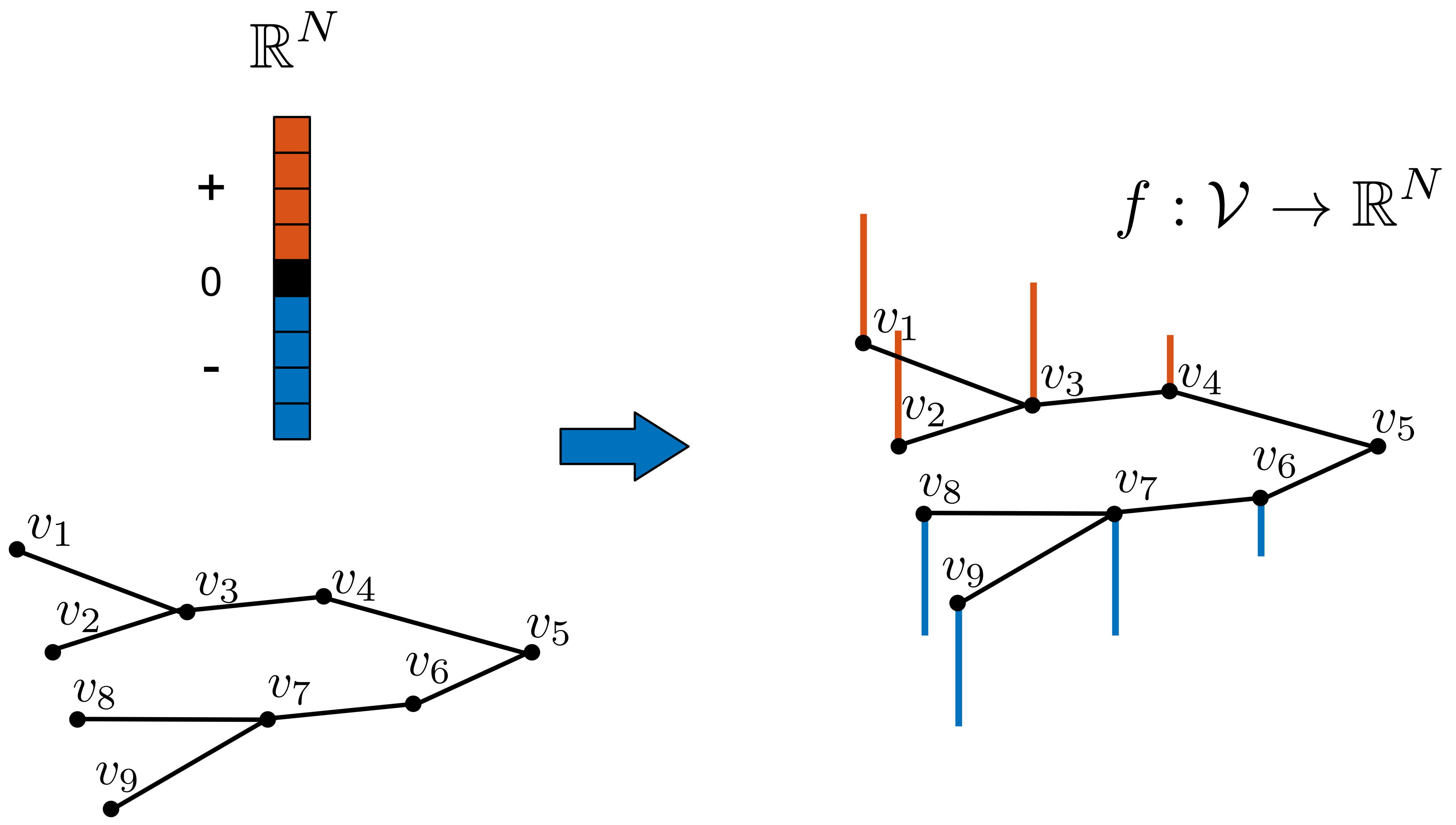
Graph signal processing

- Network-structured data can be represented by graph signals



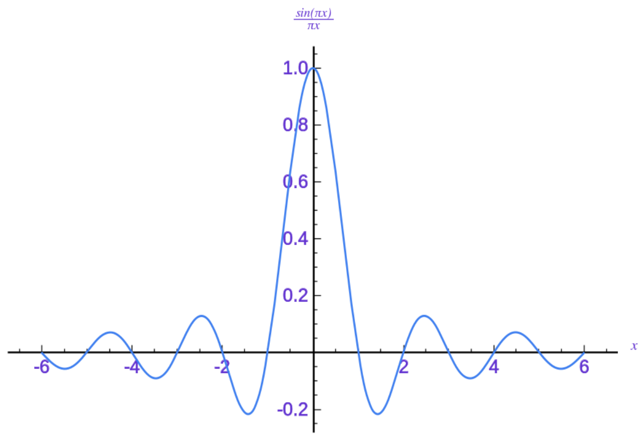
Graph signal processing

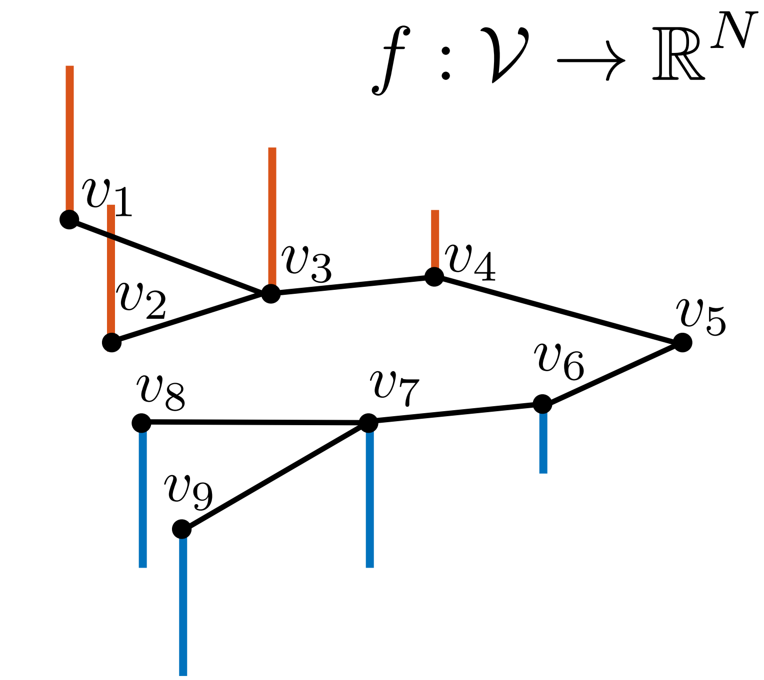
- Network-structured data can be represented by graph signals



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

Graph signal processing





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

Graph signal processing

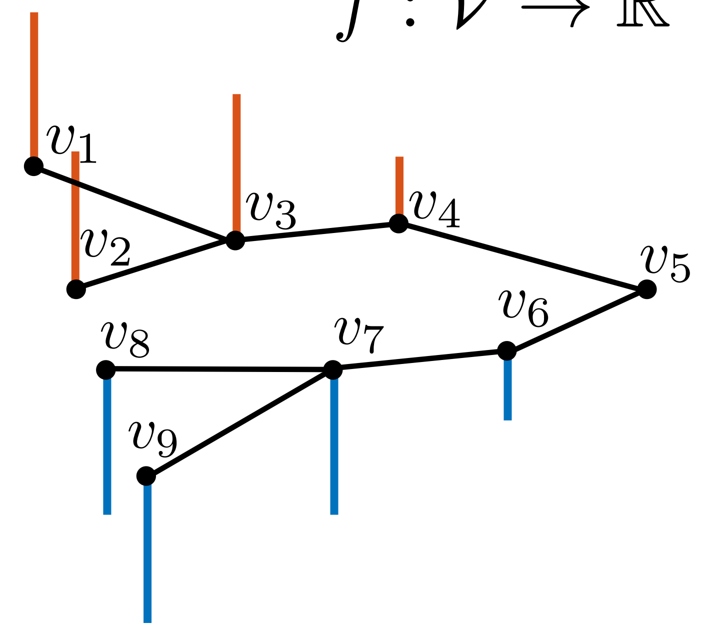
- Graph signals provide a nice compact format to encode structure within data

- Generalisation of classical signal processing tools can greatly benefit analysis of such data

- Numerous applications: Transportation, biomedical, social, economic network analysis

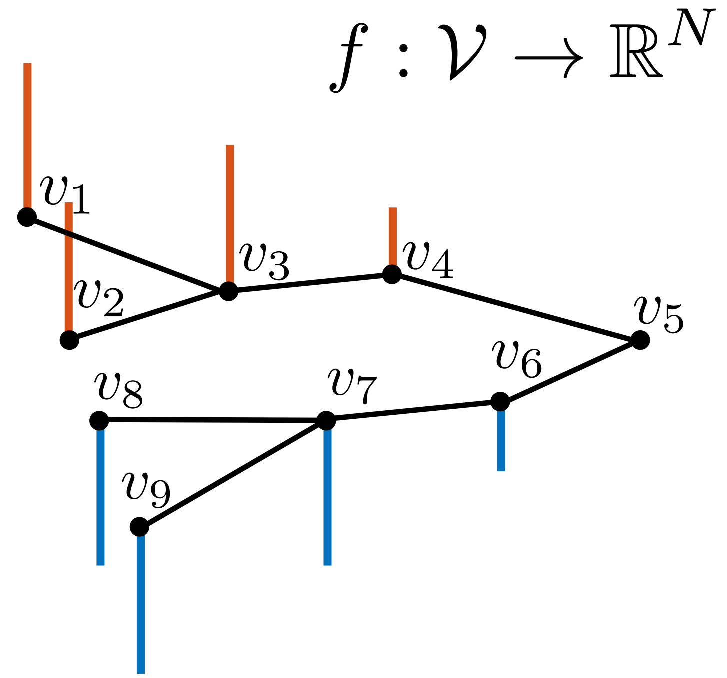
- An increasingly rich literature
  - classical signal processing
  - algebraic and spectral graph theory
  - computational harmonic analysis
  - machine learning

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



Two paradigms

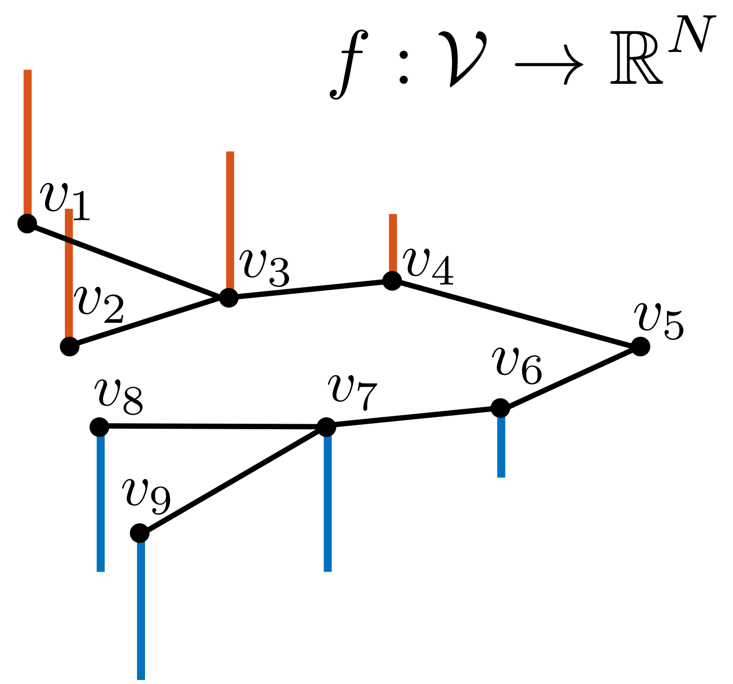
- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs



Two paradigms

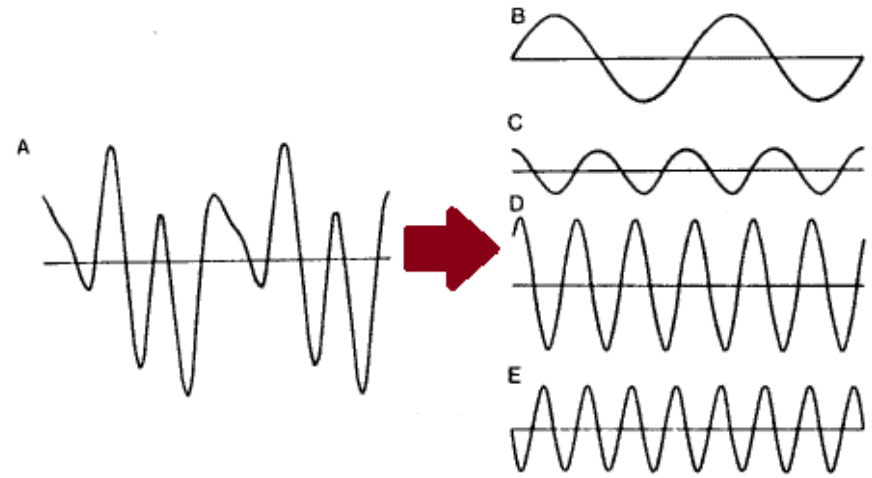
- Main GSP approaches can be categorised into two families:
  - vertex (spatial) domain designs
  - frequency (graph spectral) domain designs

important for analysis of signal properties



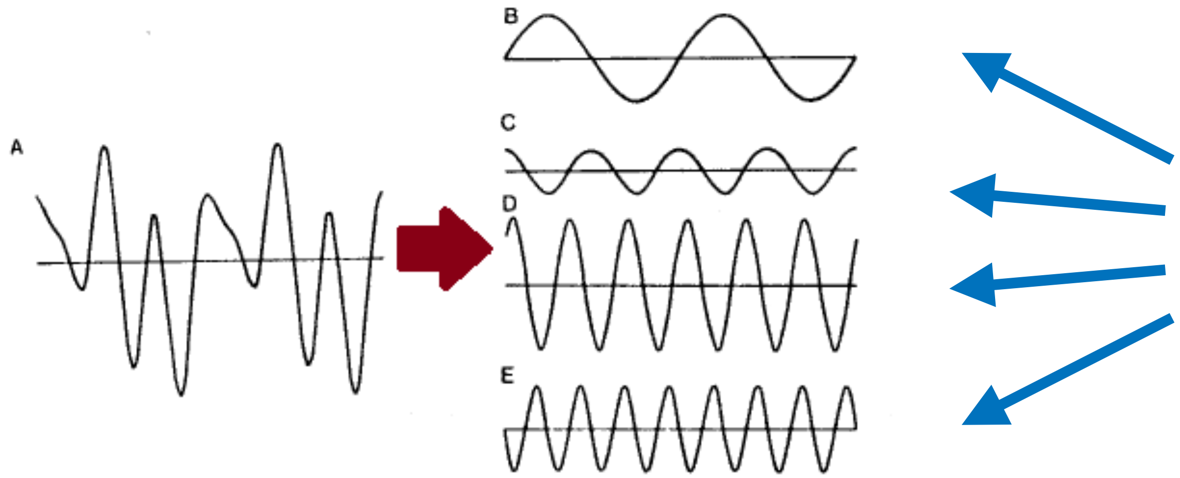
Need for notion of frequency

- Classical Fourier transform provides frequency domain representation of signals



Need for notion of frequency

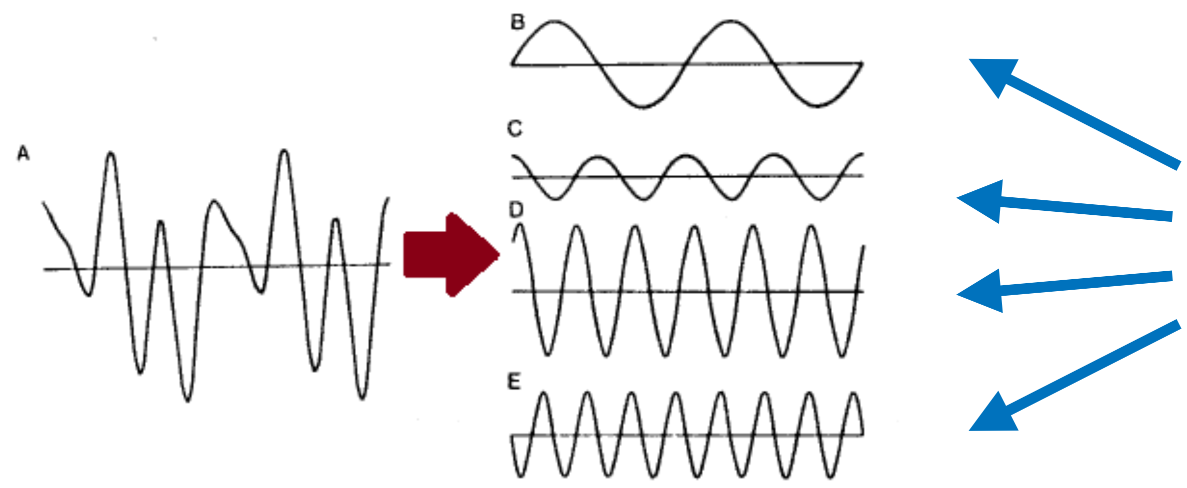
- Classical Fourier transform provides frequency domain representation of signals



- "building blocks" of signal
- different frequency (oscillation)

Need for notion of frequency

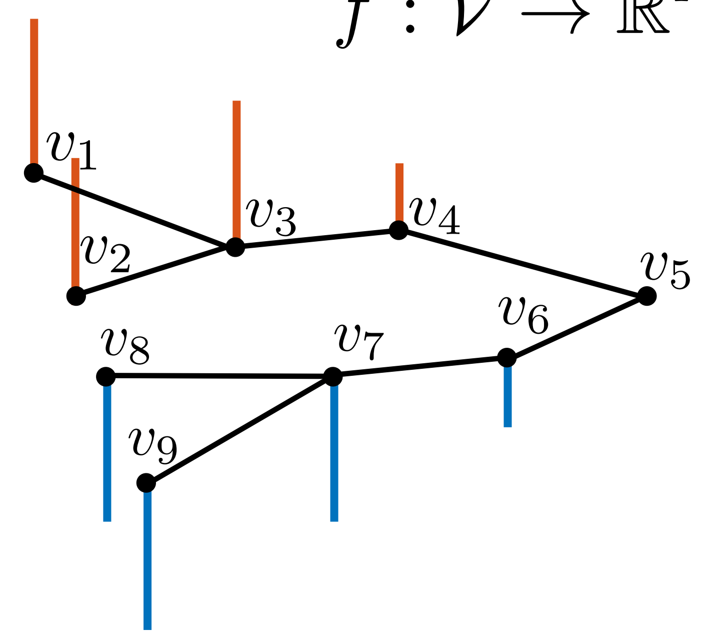
- Classical Fourier transform provides frequency domain representation of signals



- "building blocks" of signal
- different frequency (oscillation)

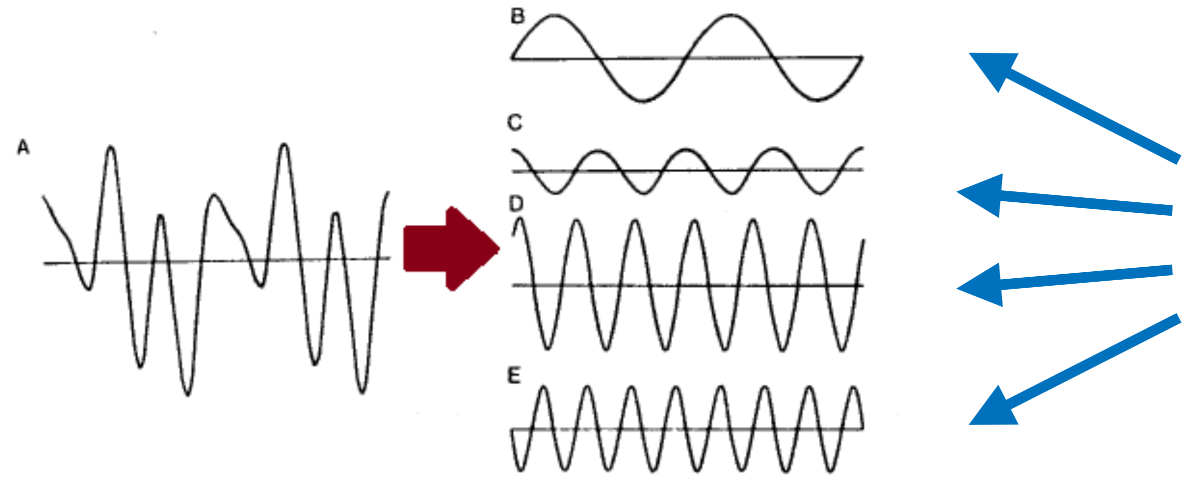
- What about a notion of frequency for graph signals?

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



Need for notion of frequency

- Classical Fourier transform provides frequency domain representation of signals

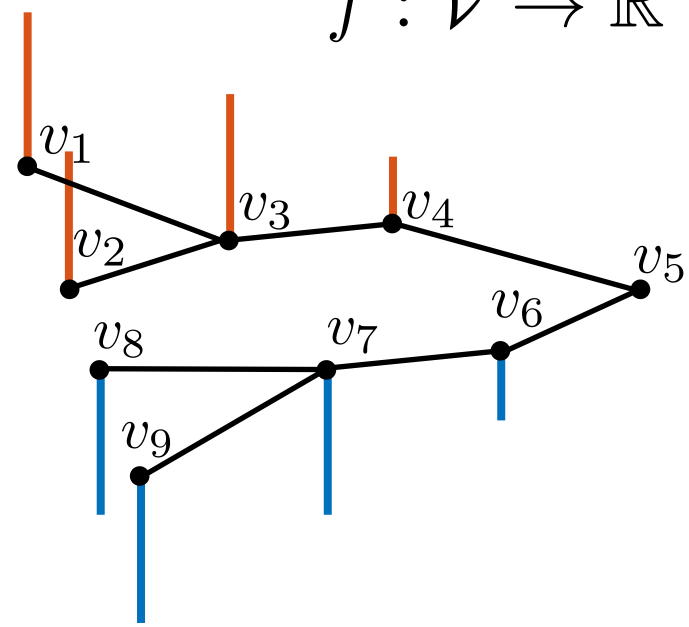


- "building blocks" of signal
- different frequency (oscillation)

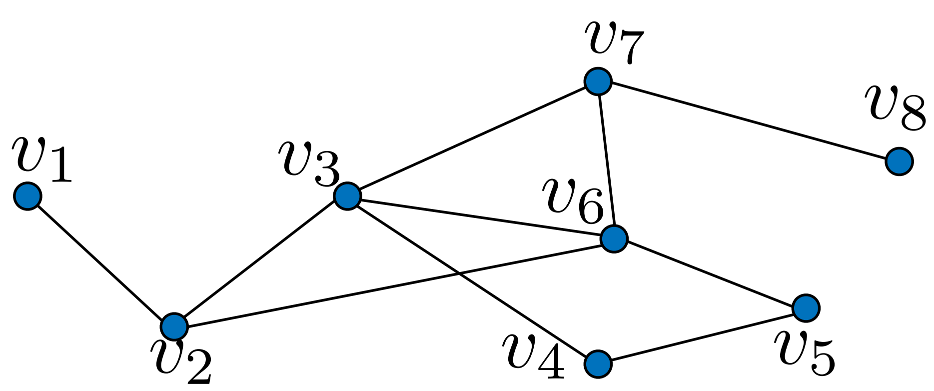
- What about a notion of frequency for graph signals?

  we need the graph Laplacian matrix

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



Graph Laplacian



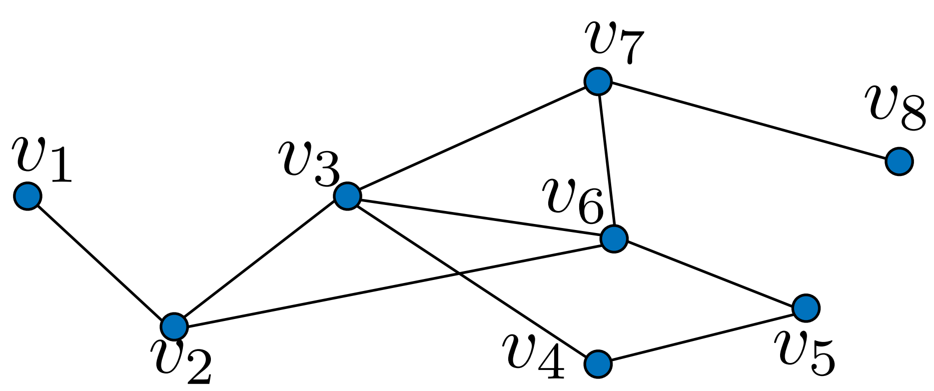
weighted and undirected graph:

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

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

W

Graph Laplacian



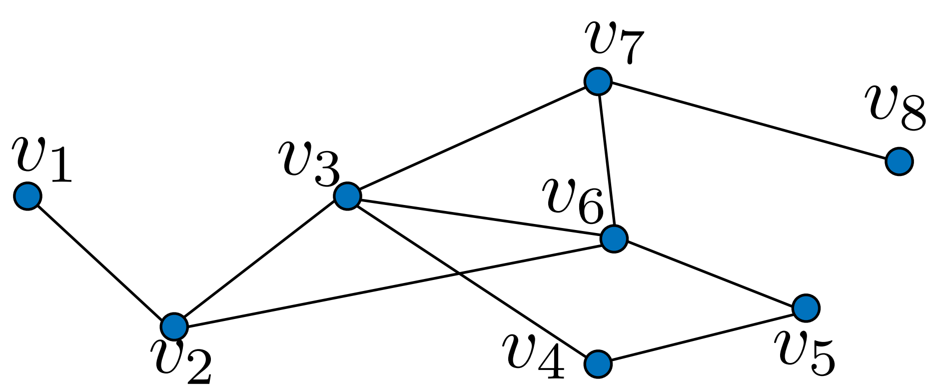
weighted and undirected graph:

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

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

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

Graph Laplacian



weighted and undirected graph:

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

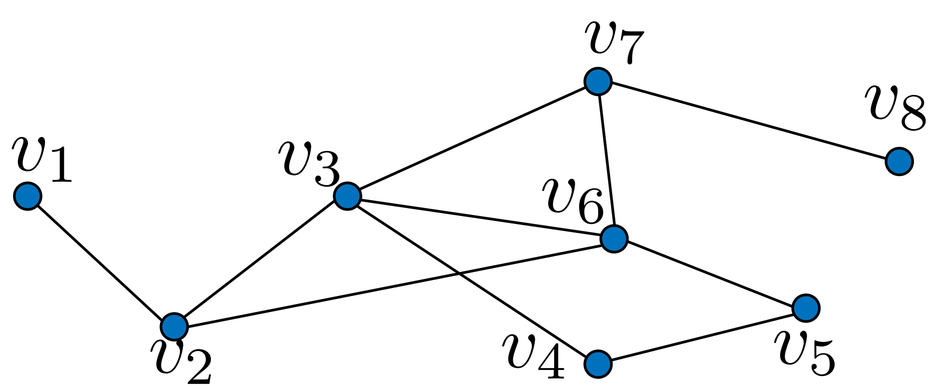
$$D = \mathrm{diag}(d(v_1), \cdots, d(v_N))$$

$$L = D - W$$ **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:

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

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

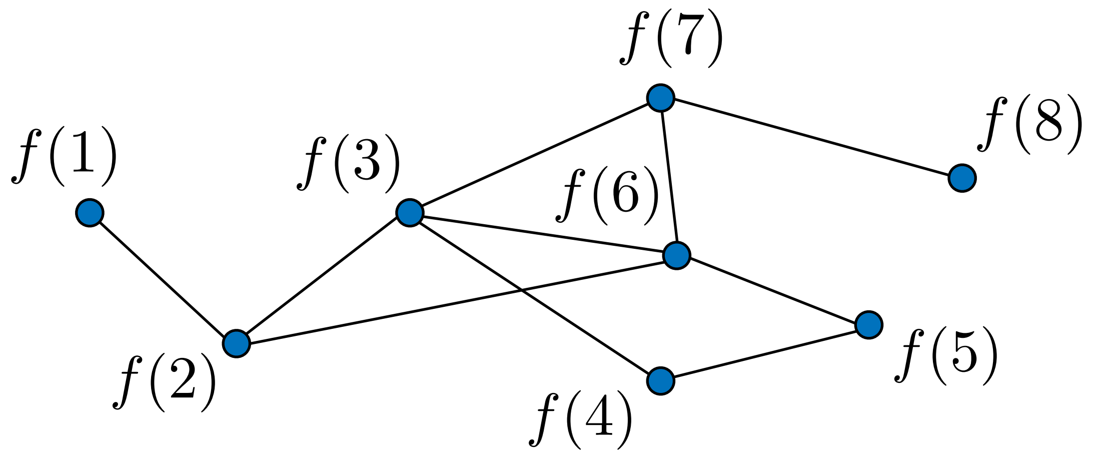
$$L = D - W$$ **equivalent to G!**

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

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

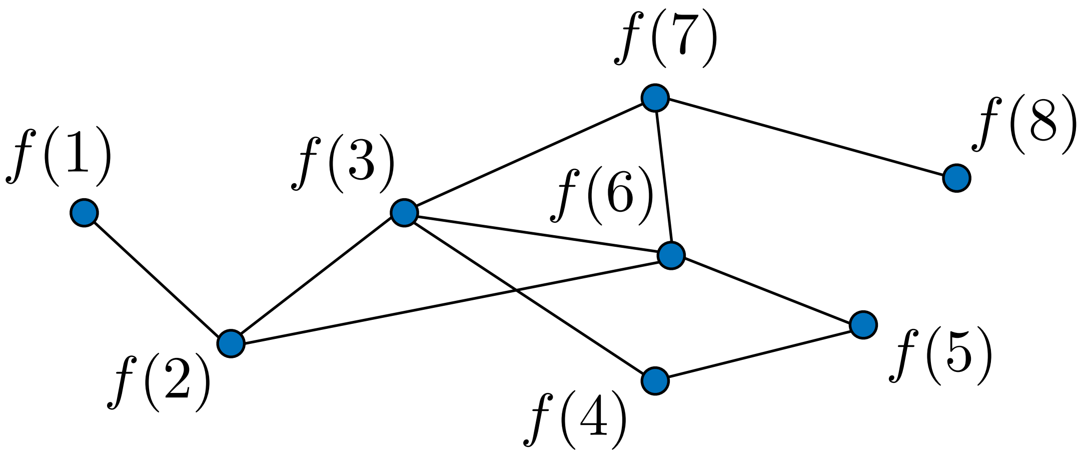
- symmetric
- off-diagonal entries non-positive
- rows sum up to zero

Graph Laplacian



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

Graph Laplacian

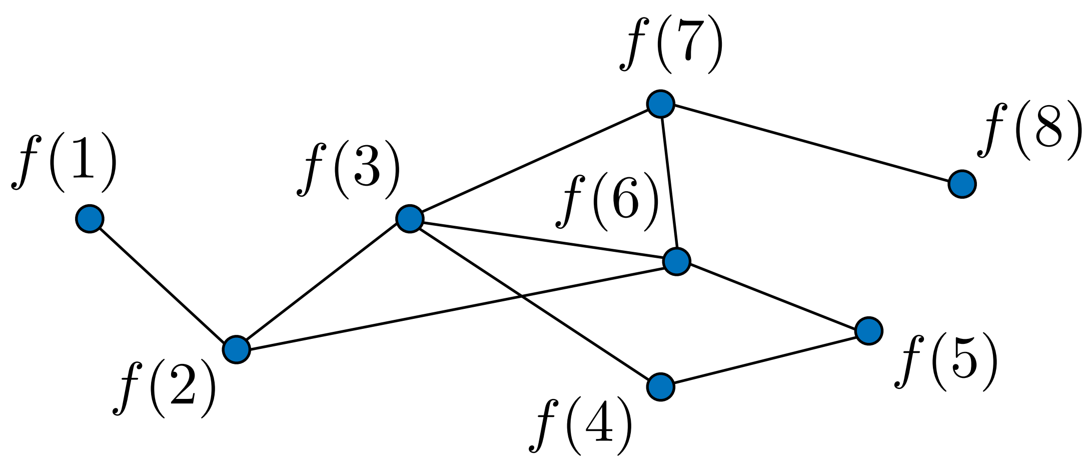


graph signal $f : \mathcal{V} \to \mathbb{R}^N$

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

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

Graph Laplacian



graph signal $f : \mathcal{V} \rightarrow \mathbb{R}^N$

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

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

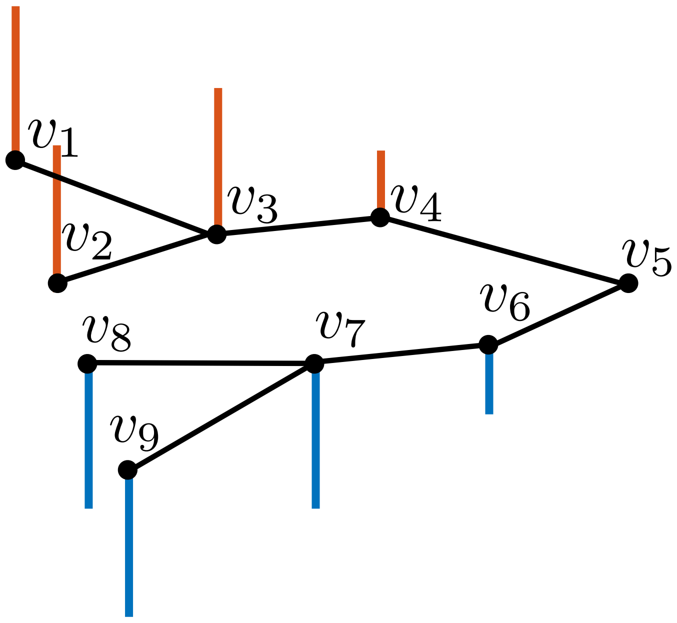
$$Lf(i) = \sum_{j=1}^{N} W_{ij}(f(i) - f(j))$$

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

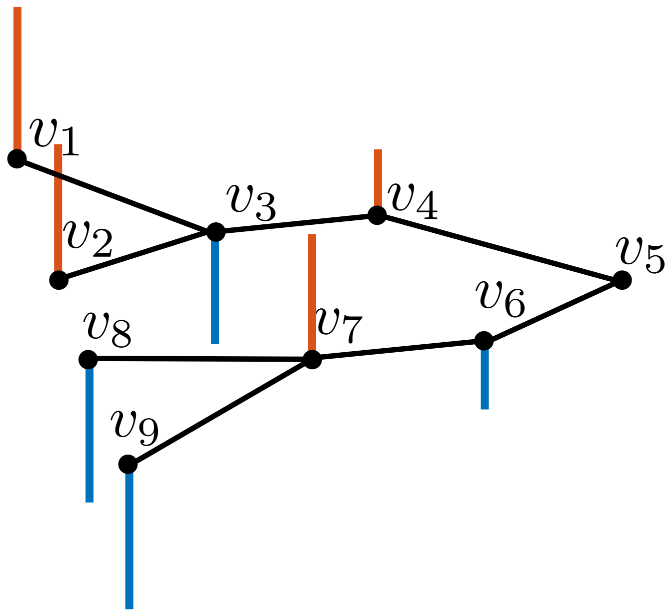
a measure of "smoothness"

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: $L = \chi \Lambda \chi^T$

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

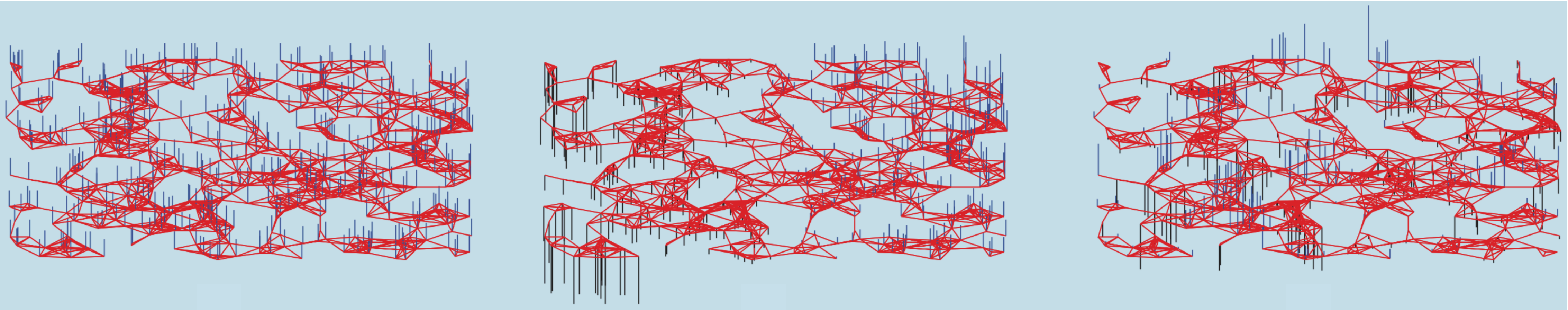
Graph Laplacian

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

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

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

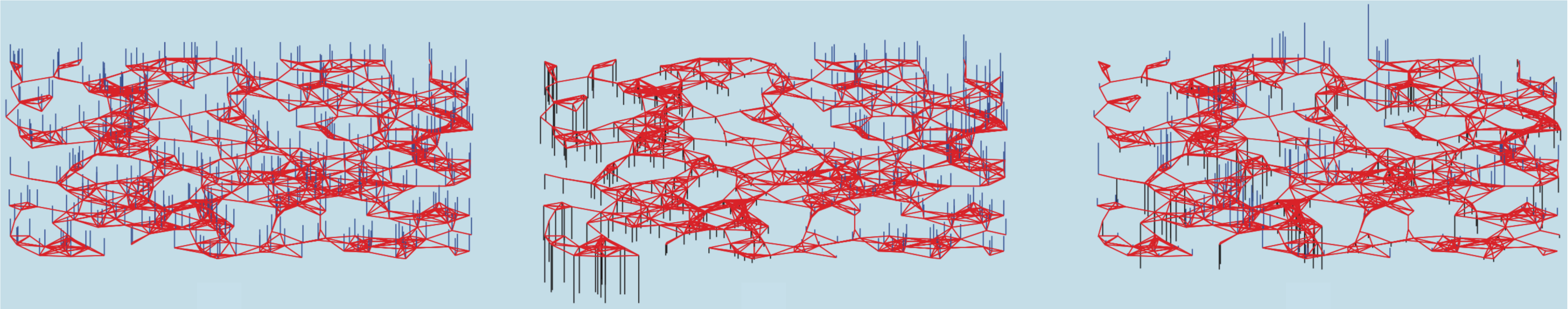
Graph Fourier transform



χ_0 χ_1 χ_{50}

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

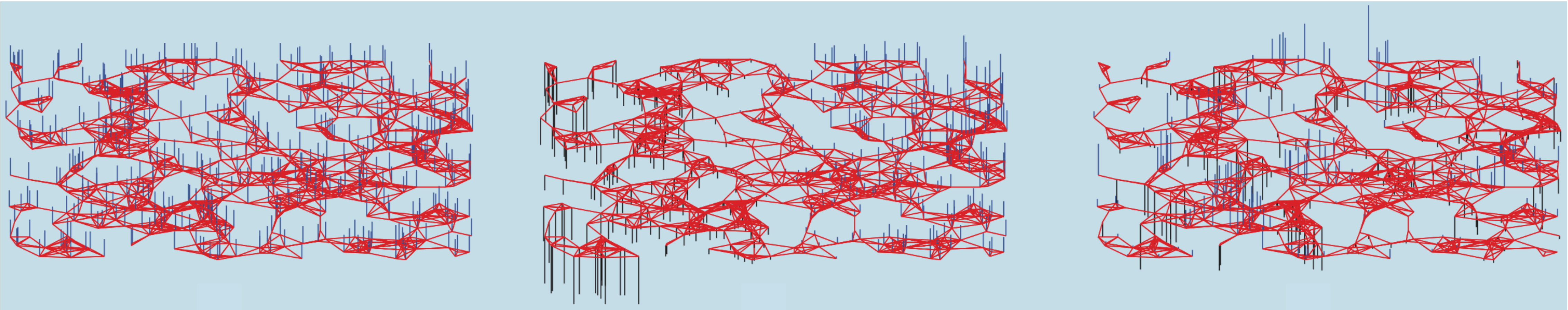
$L = \chi \Lambda \chi^T$

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

- 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

$$L = \chi \Lambda \chi^T$$

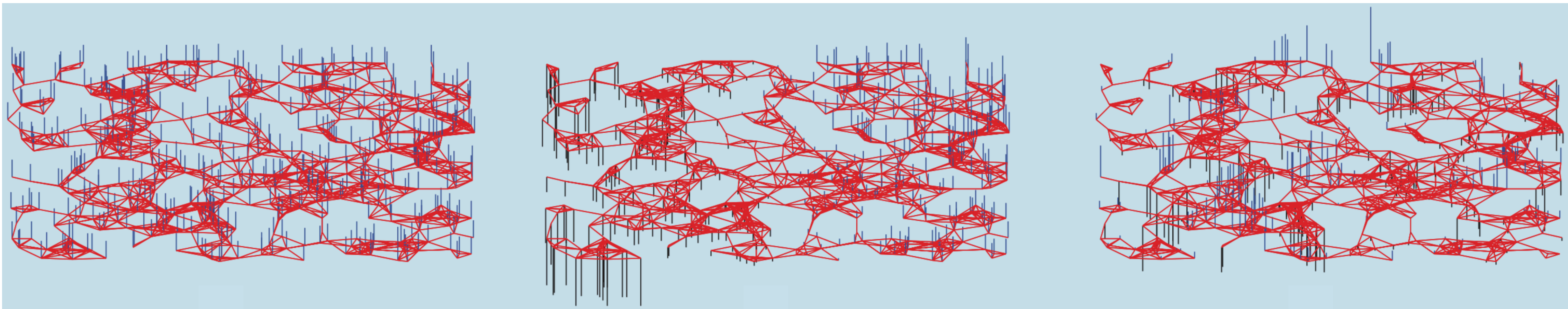
$\chi_0^T L \chi_0 = \lambda_0 = 0$ $\chi_{50}^T L \chi_{50} = \lambda_{50}$

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

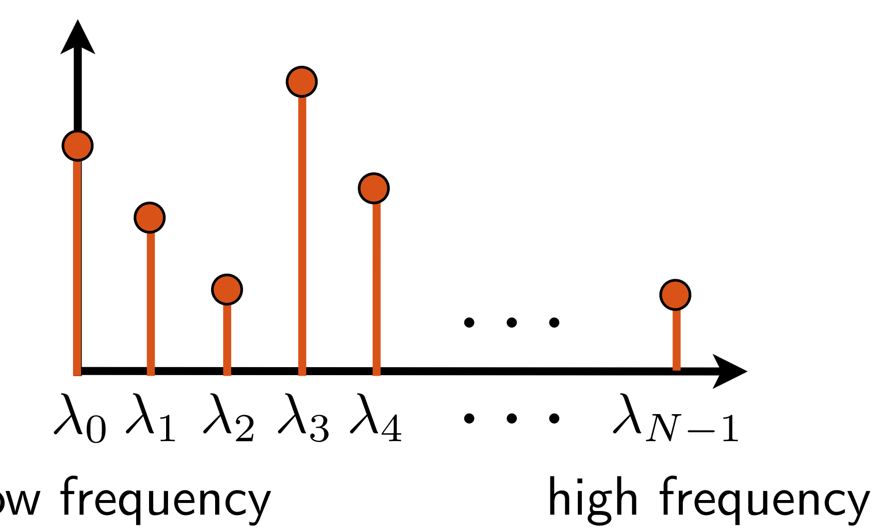
$L = \chi \Lambda \chi^T$

$\chi_0^T L \chi_0 = \lambda_0 = 0$

$\chi_{50}^T L \chi_{50} = \lambda_{50}$

graph Fourier transform:
[Hammond11]

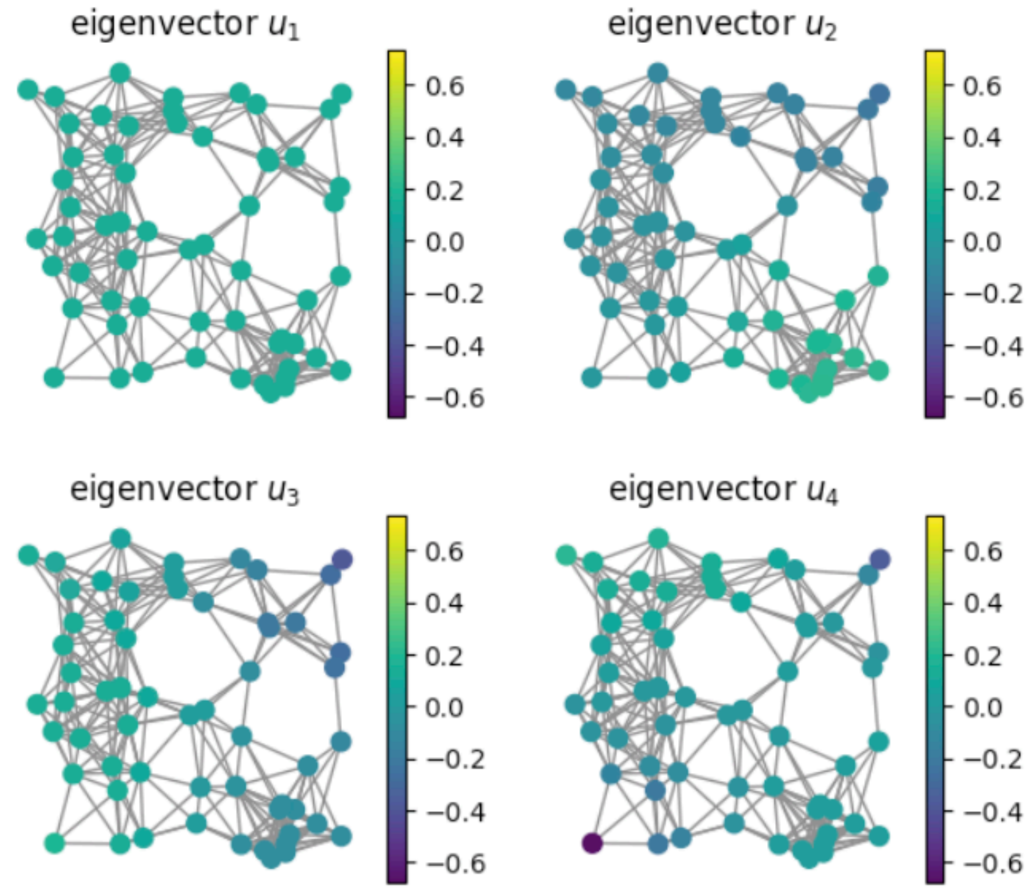
$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$



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

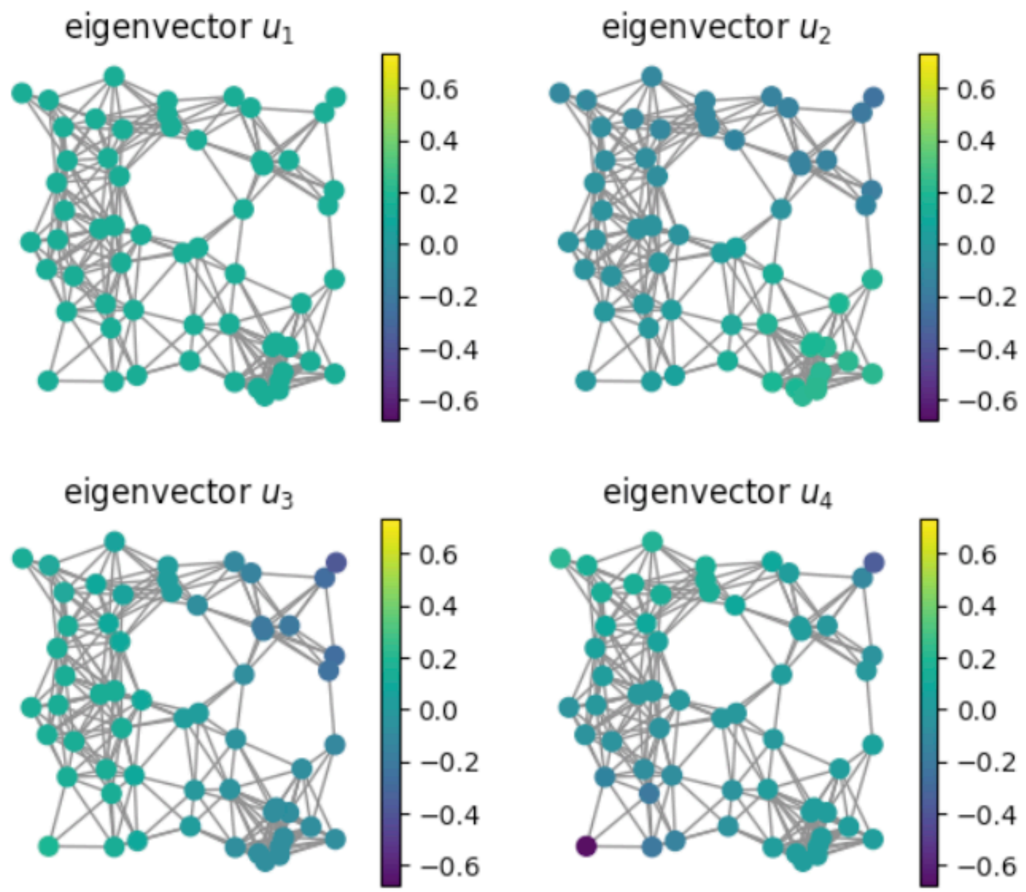
Graph Fourier transform

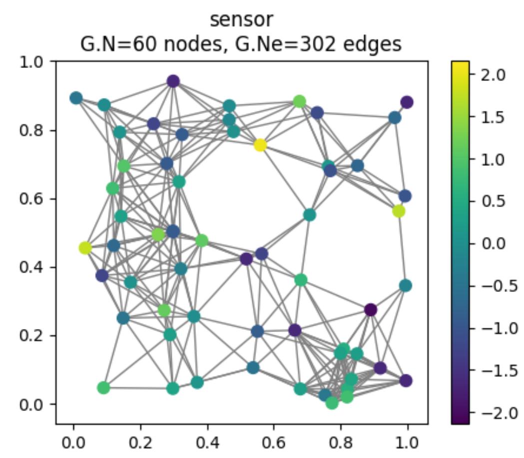
- Example on a simple graph

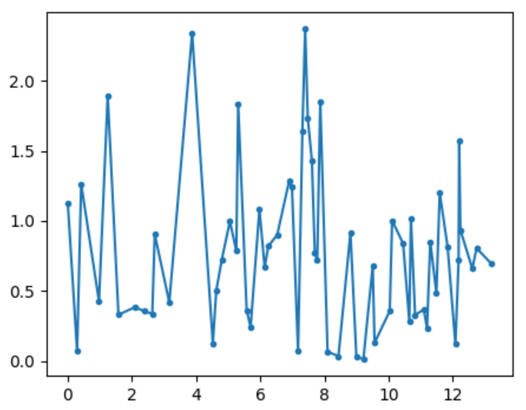


Graph Fourier transform

- Example on a simple graph

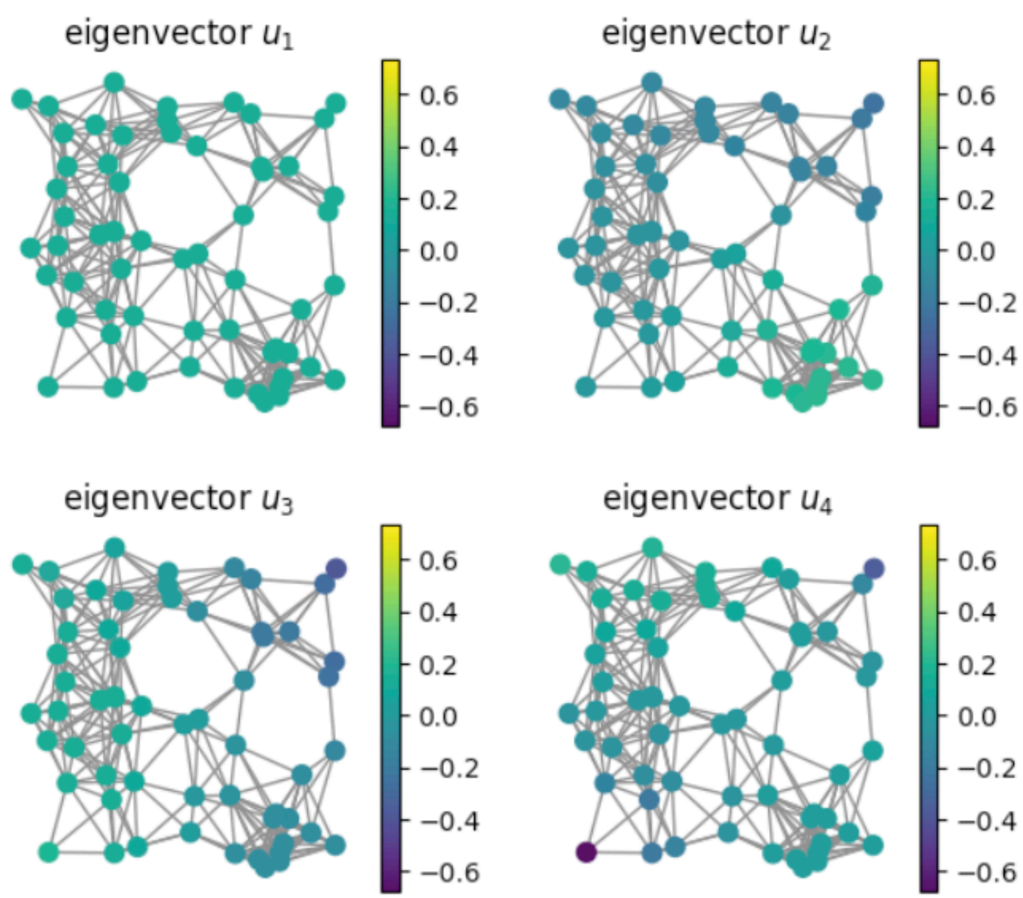


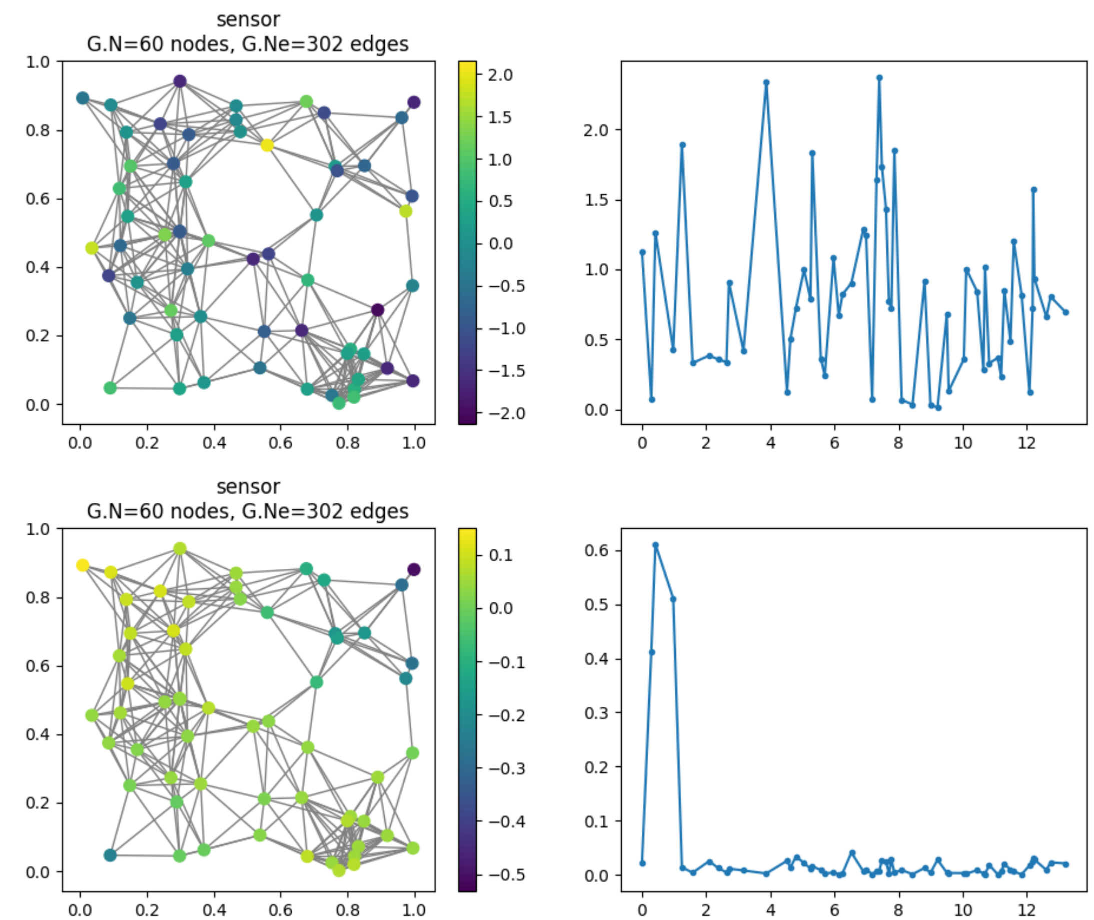




Graph Fourier transform

- Example on a simple graph





Outline

- Graph signal processing (GSP): Basic concepts
- Filtering of graph signals: Basic tool of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

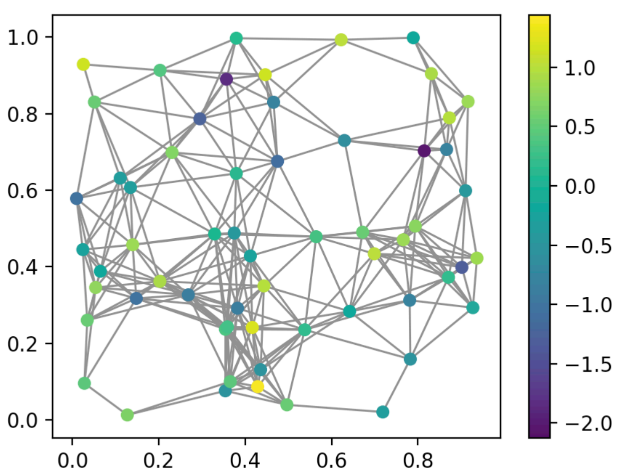
Filtering of graph signals

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

Filtering of graph signals

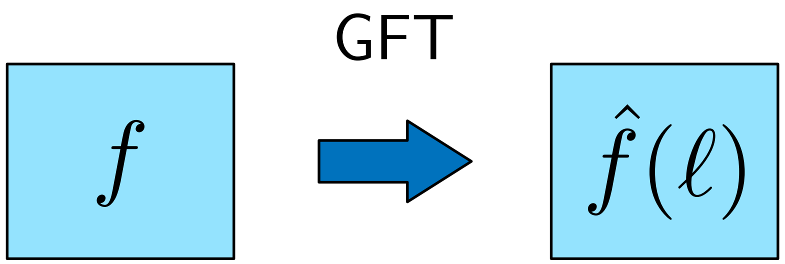
$$\text{GFT:} \quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

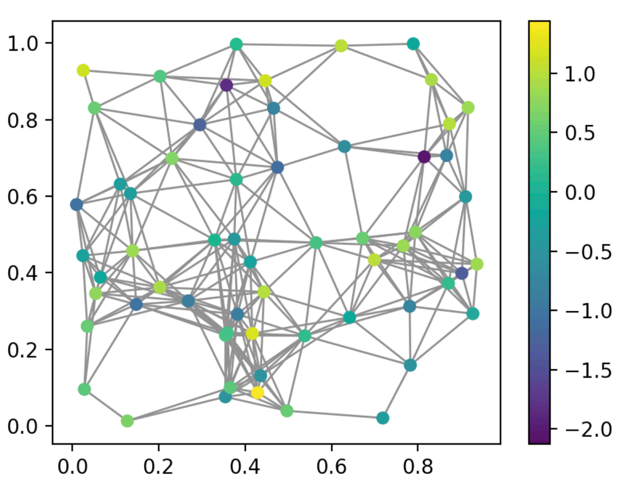
f

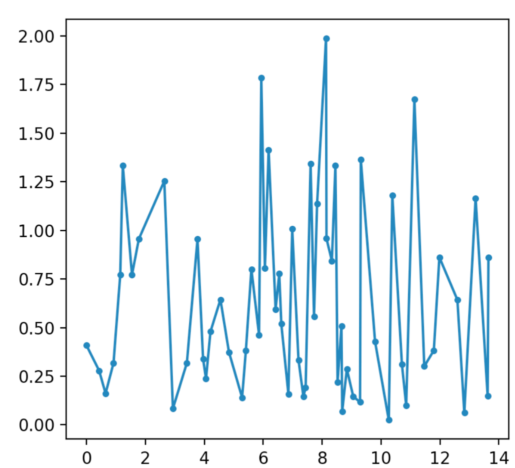


Filtering of graph signals

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

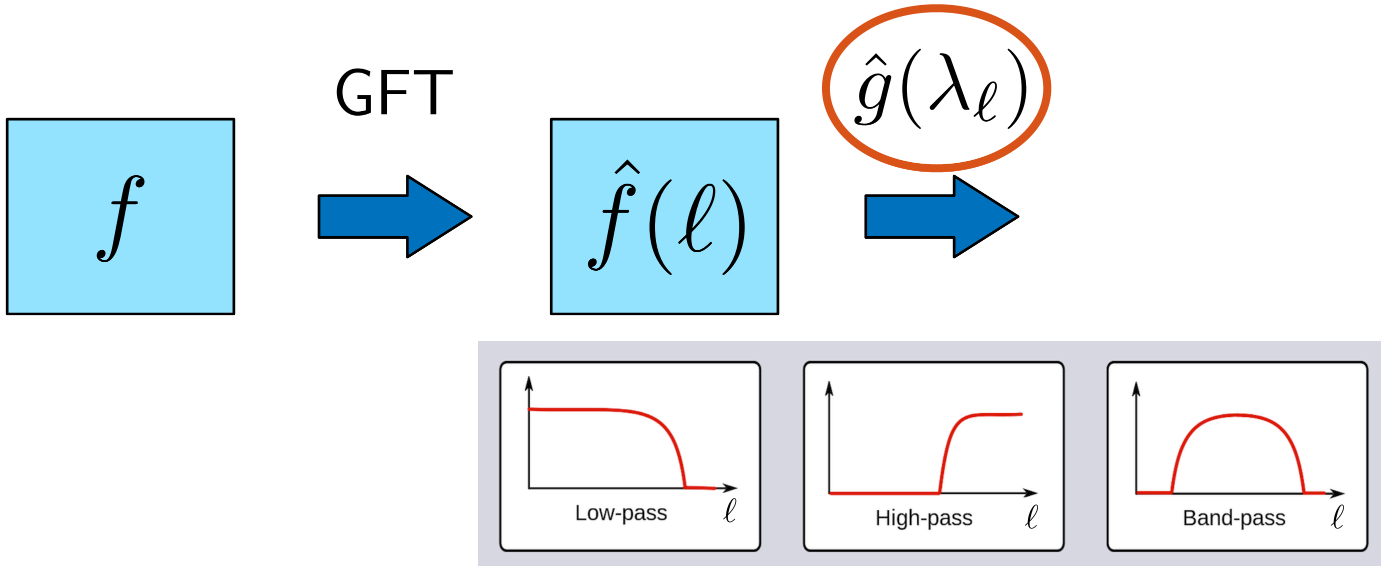


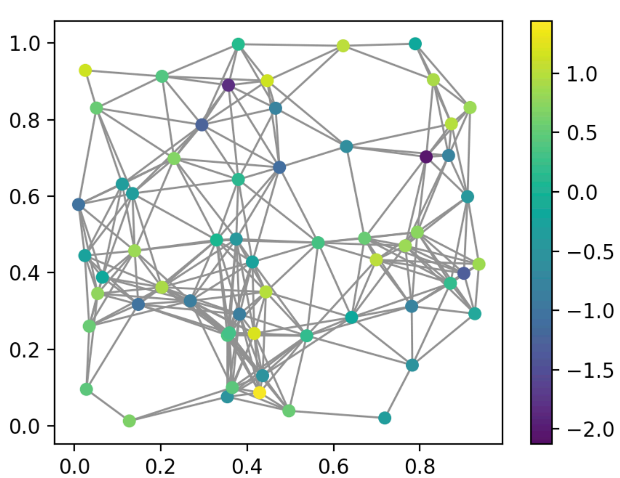


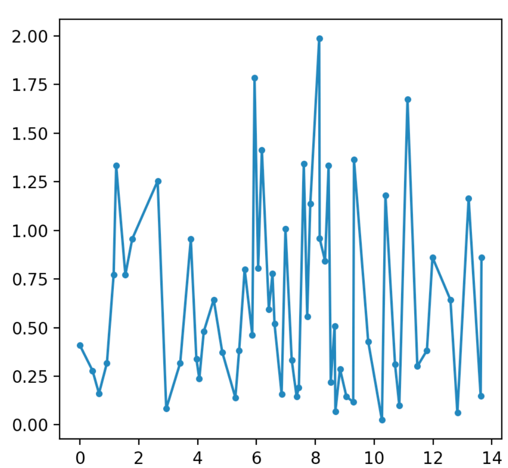


Filtering of graph signals

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

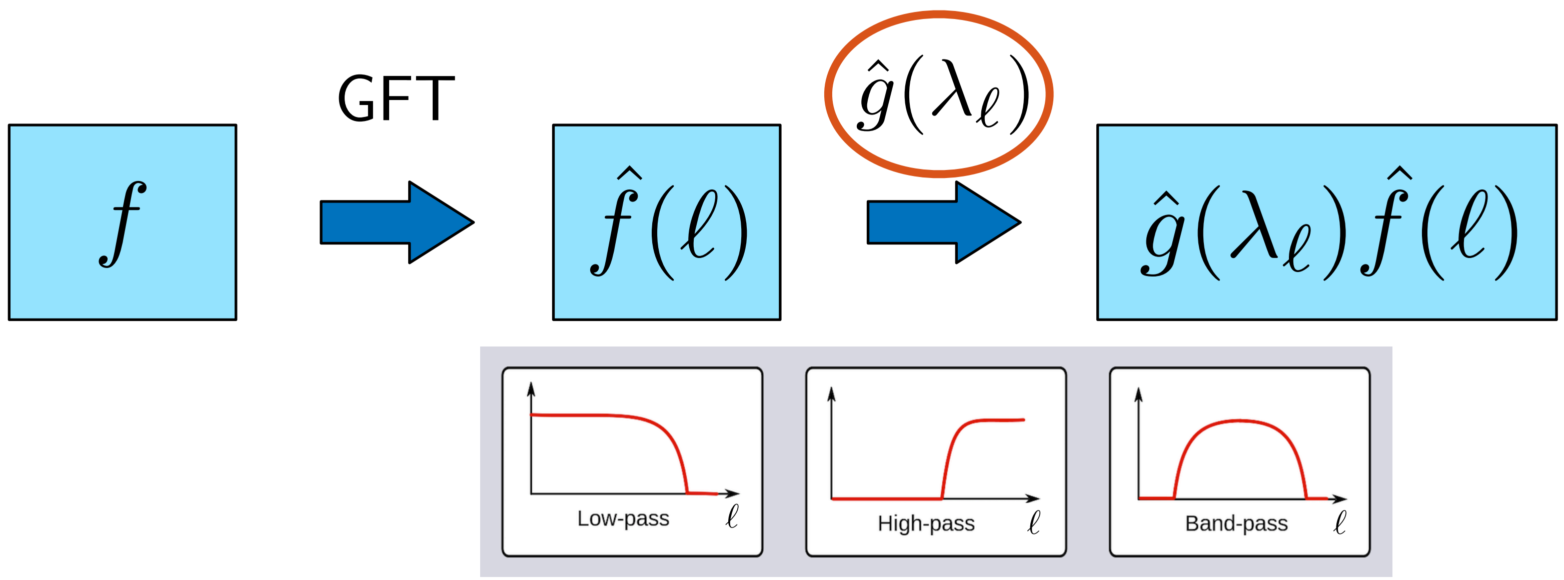


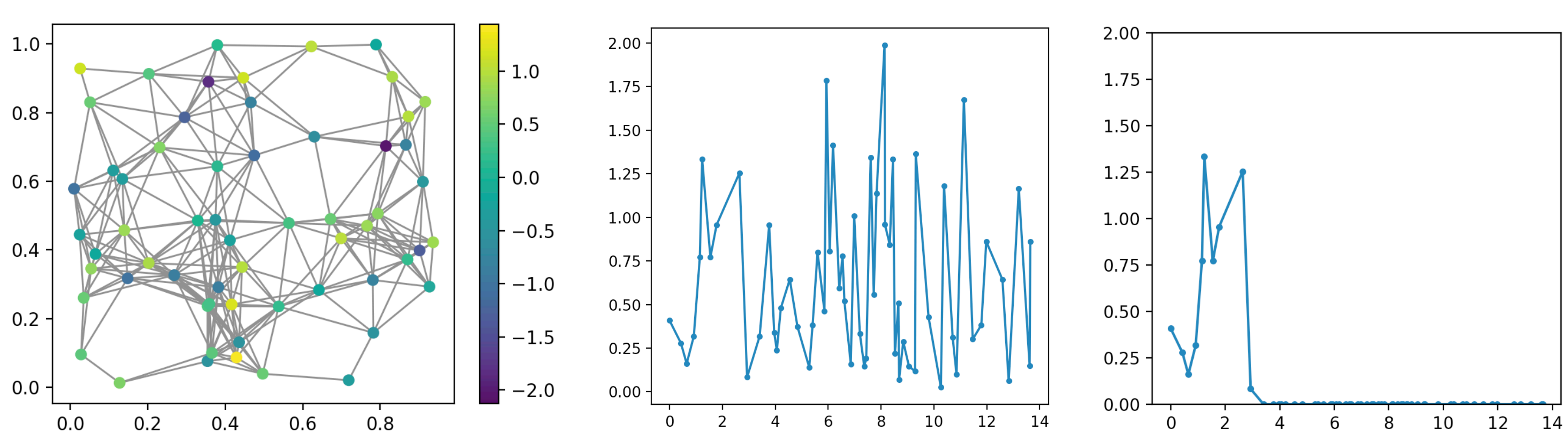




Filtering of graph signals

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

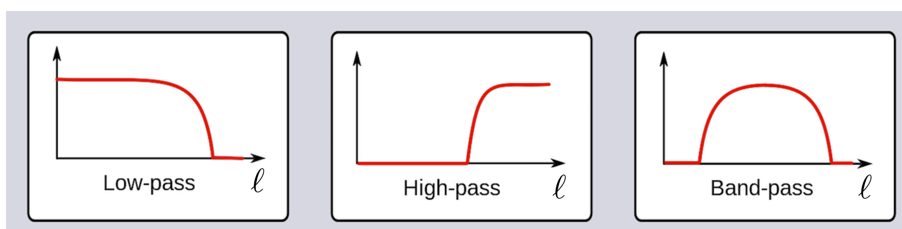


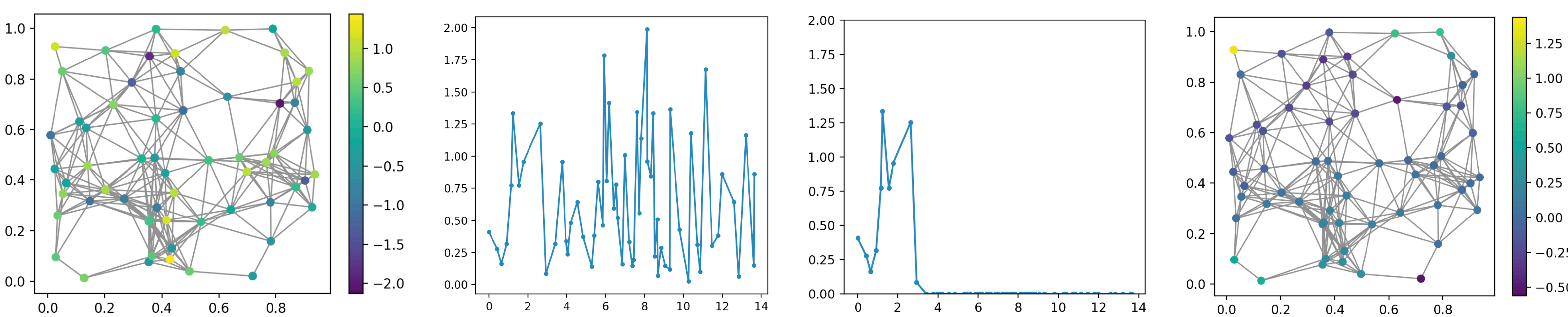


Filtering of graph signals

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

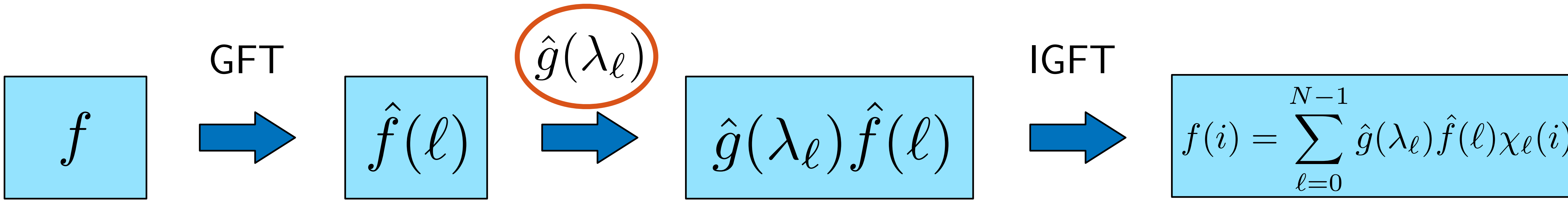
f → GFT → $\hat{f}(\ell)$ → $\hat{g}(\lambda_\ell)$ → $\hat{g}(\lambda_\ell)\hat{f}(\ell)$ → IGFT → $f(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)$

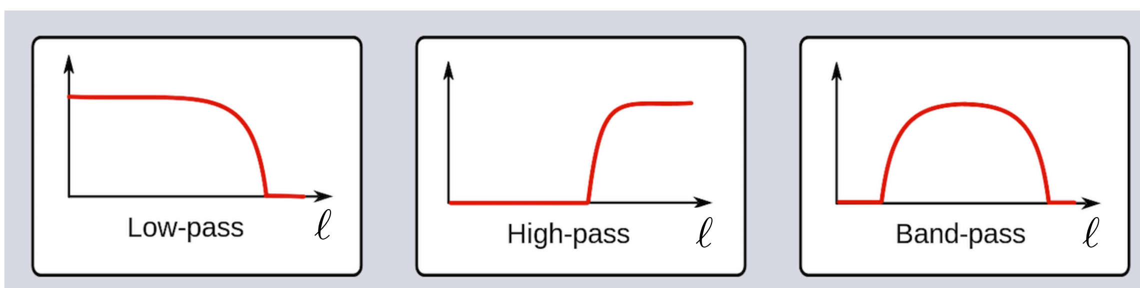




Filtering of graph signals

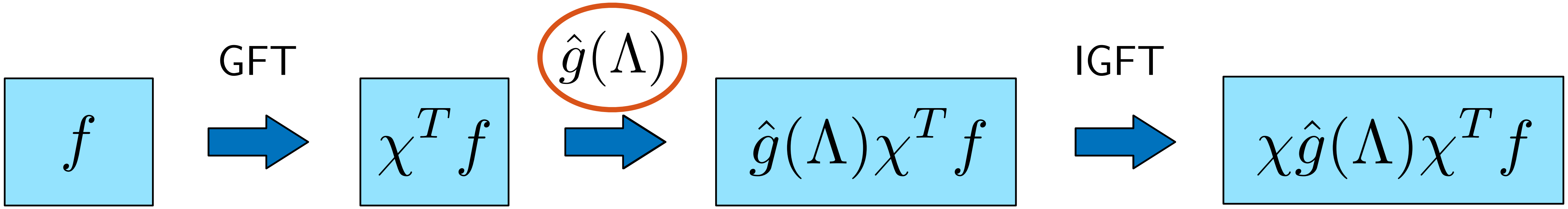
GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$





Filtering of graph signals

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



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

Filtering of graph signals

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

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

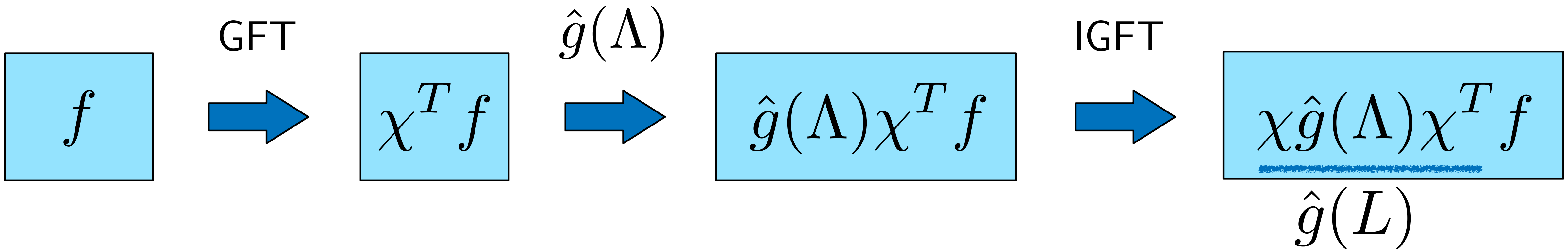
$\hat{g}(L)$**: functions of L!**

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

A practical example

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

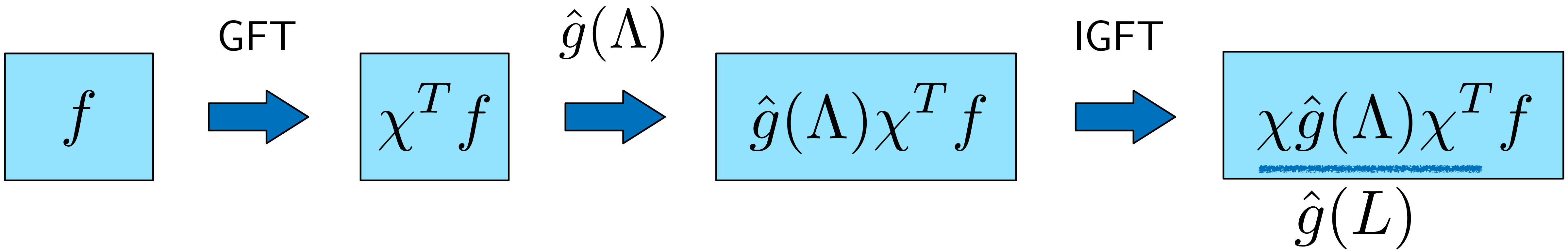
A practical example



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

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

A practical example



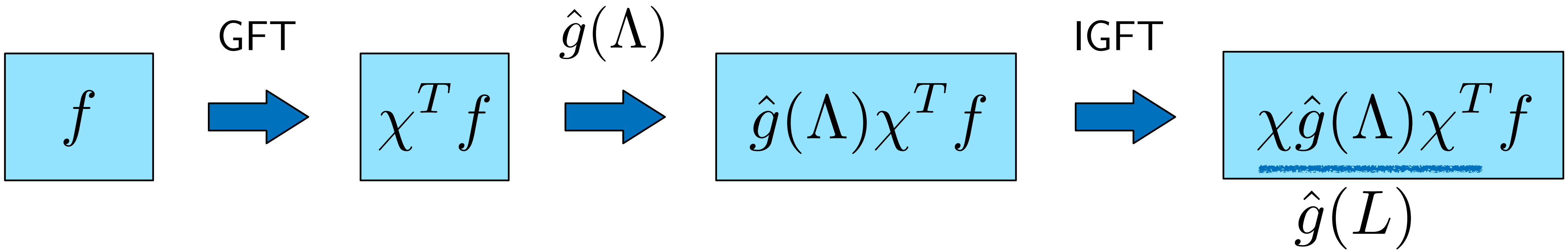
problem: we observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

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

data fitting term

"smoothness" assumption

A practical example



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

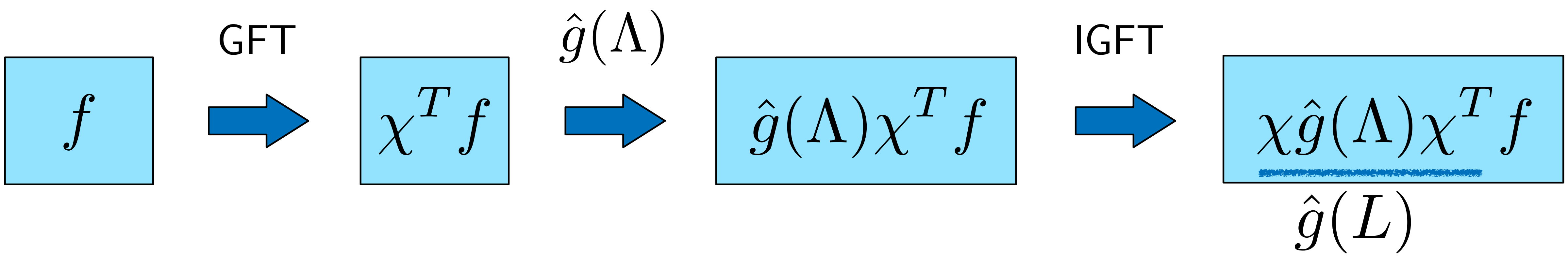
$$y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}$$

data fitting term: $||y - f||_2^2$

"smoothness" assumption: $\gamma y^T L y$

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

A practical example



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

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

data fitting term

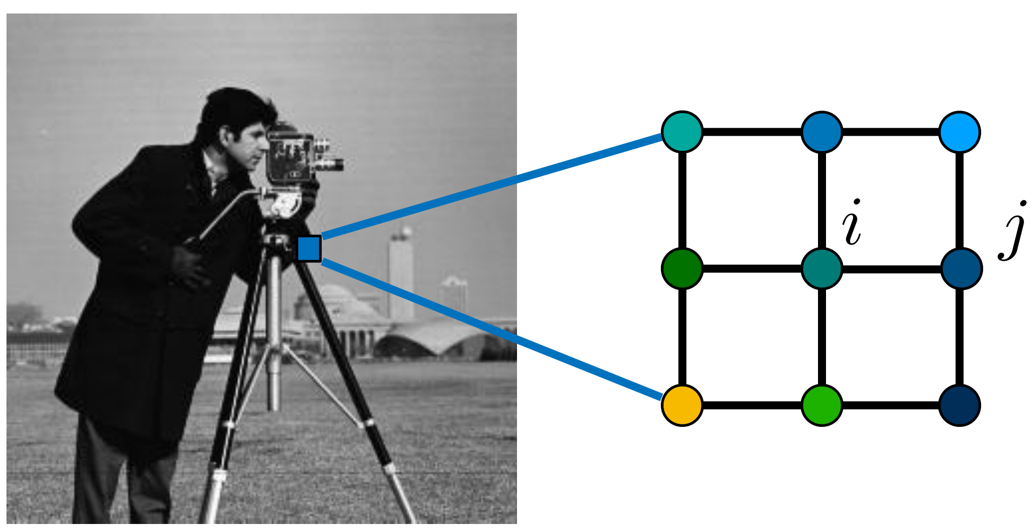
"smoothness" assumption

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

remove noise by low-pass filtering in graph spectral domain!

A practical example

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

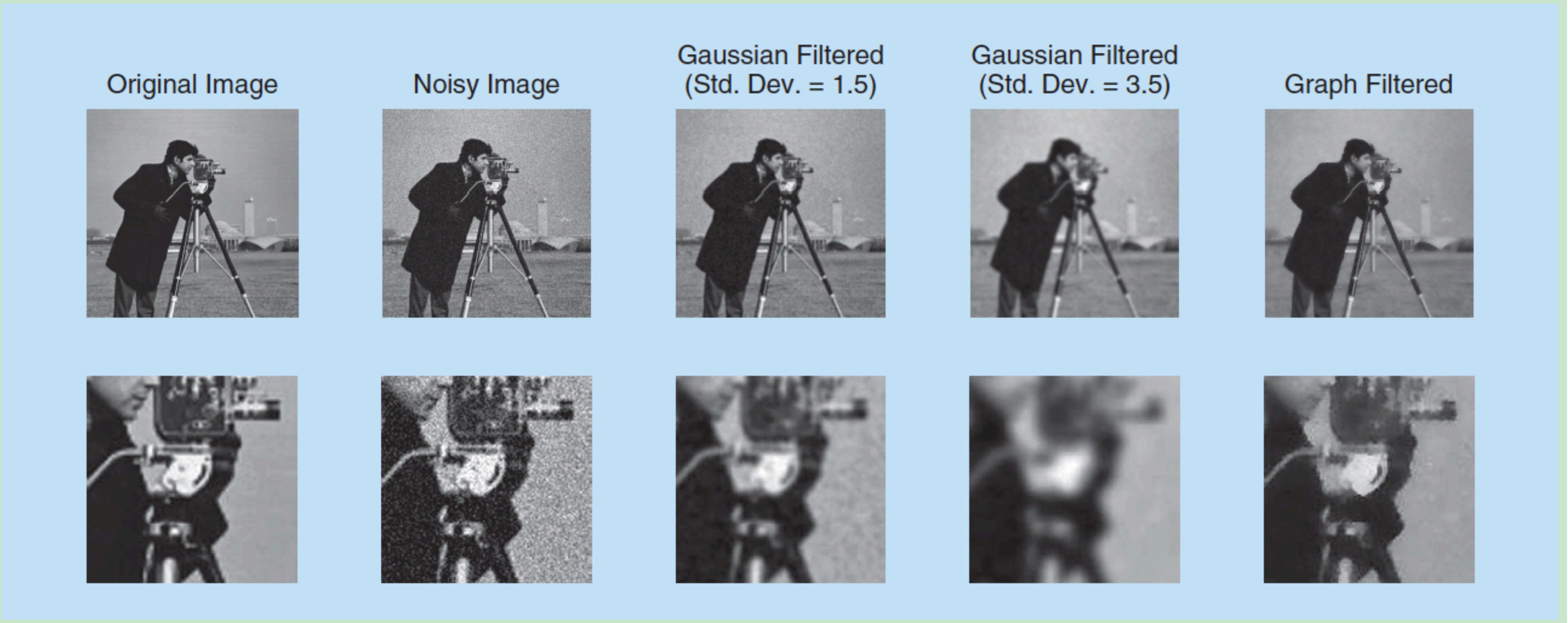


$$w_{ij} = \frac{1}{|f(i) - f(i)|}$$

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

A practical example

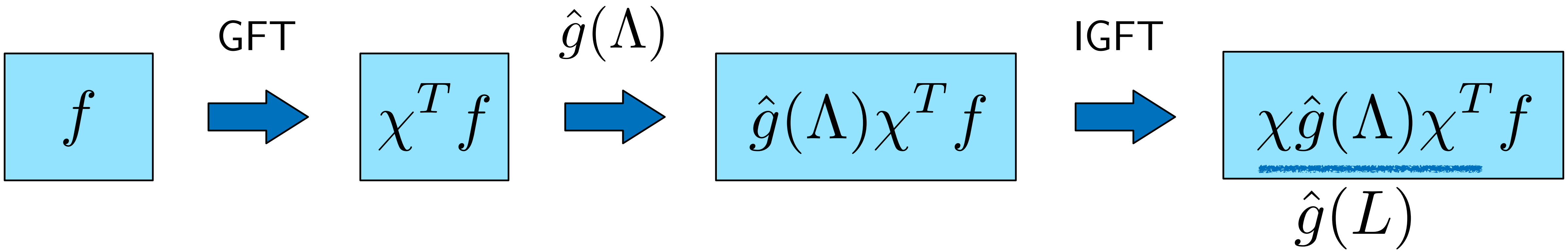
- noisy image as observed noisy graph signal
- regular grid graph (weights inversely proportional to pixel value difference)



[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.

More filtering operations



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

Outline

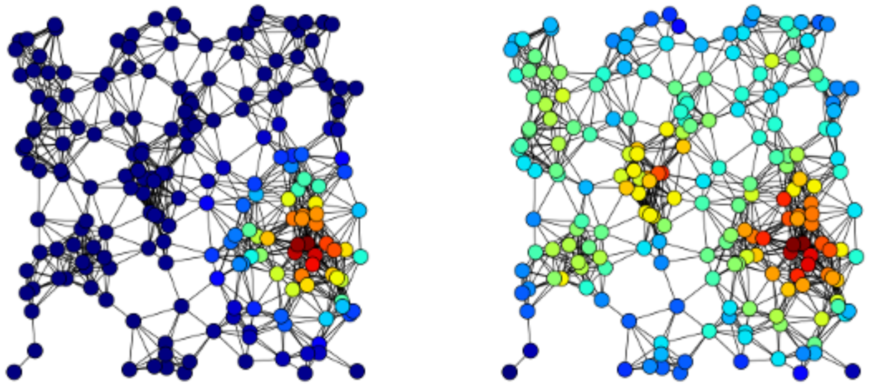
- Graph signal processing (GSP): Basic concepts
- Filtering of graph signals: Basic tool of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

GSP and the literature

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

GSP and the literature

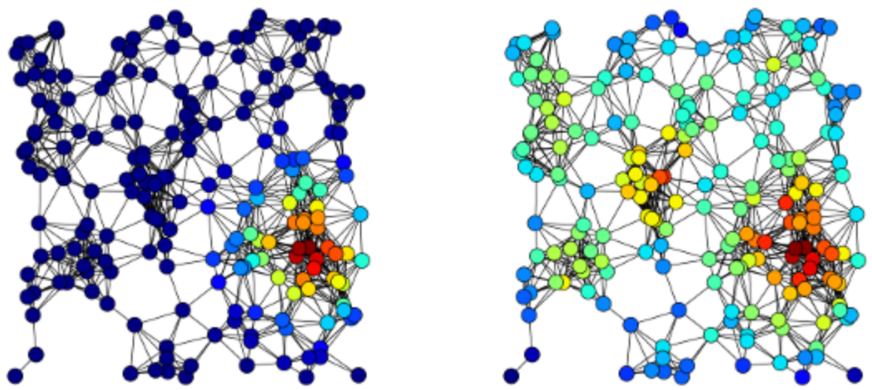
there is a rich literature about data analysis and learning on graphs



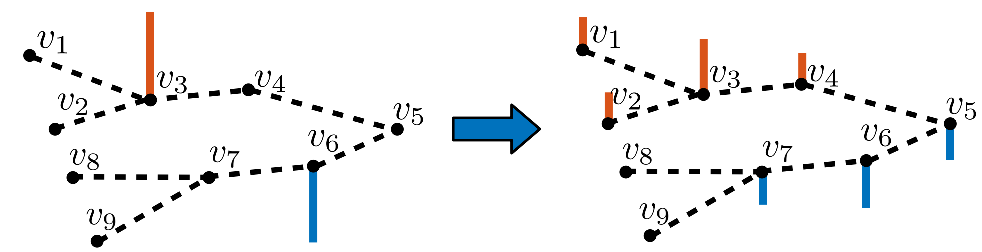
network science

GSP and the literature

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



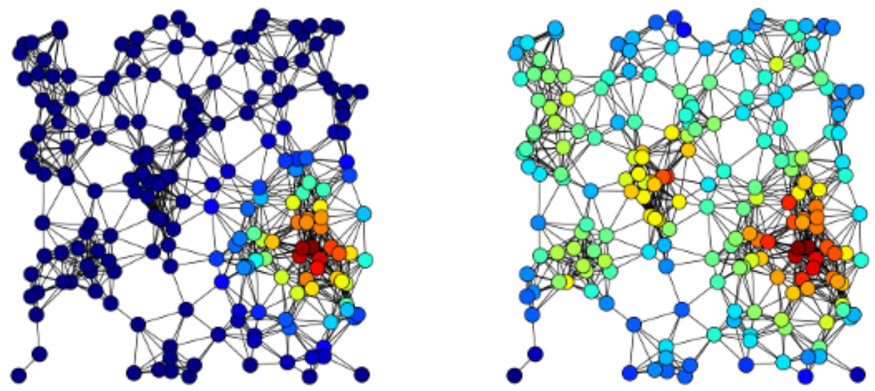
network science



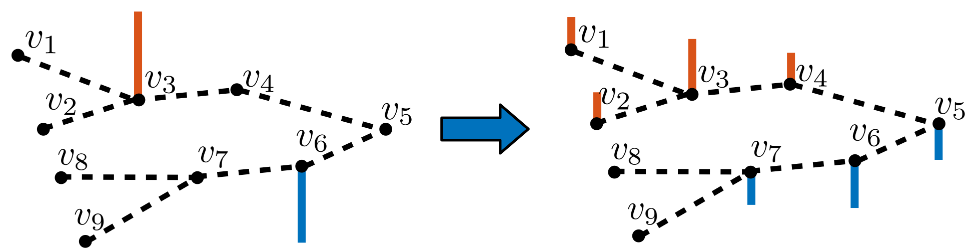
diffusion on graphs

GSP and the literature

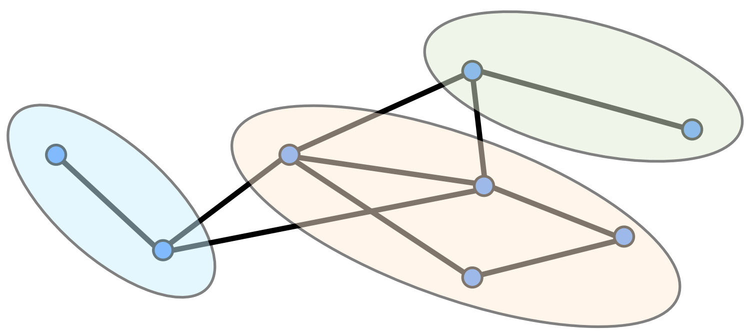
there is a rich literature about data analysis and learning on graphs



network science



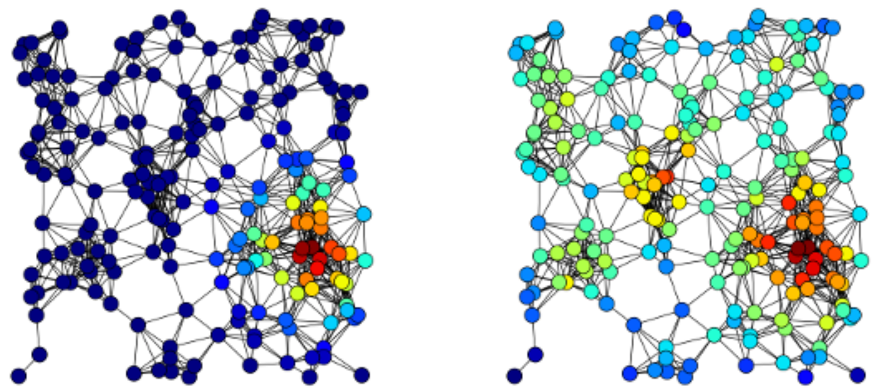
diffusion on graphs



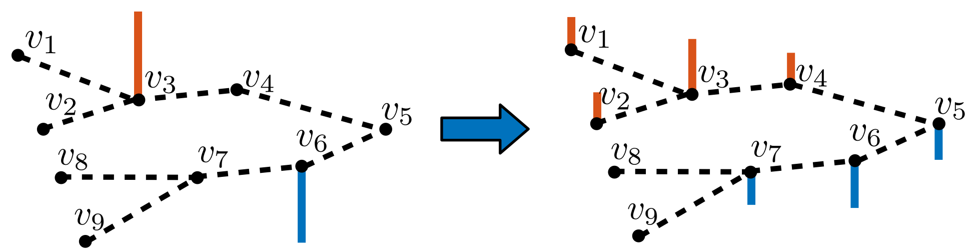
unsupervised learning (dimensionality reduction, clustering)

GSP and the literature

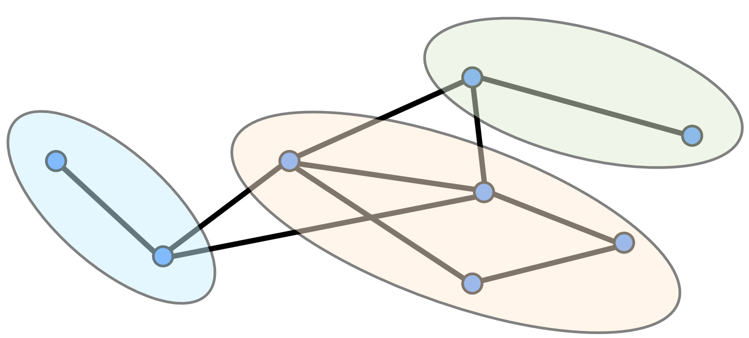
there is a rich literature about data analysis and learning on graphs



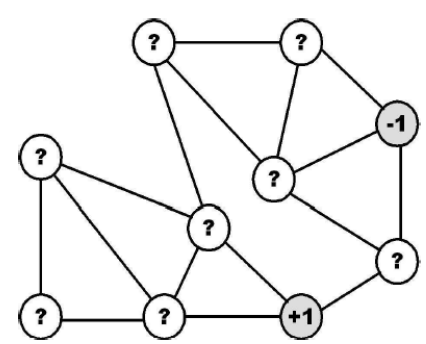
network science



diffusion on graphs



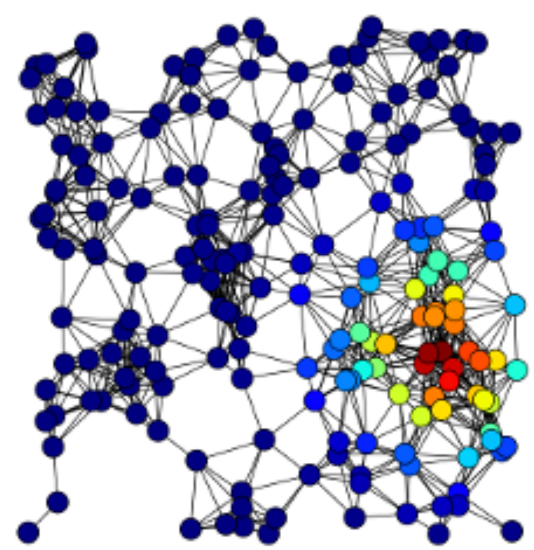
unsupervised learning (dimensionality reduction, clustering)



semi-supervised learning

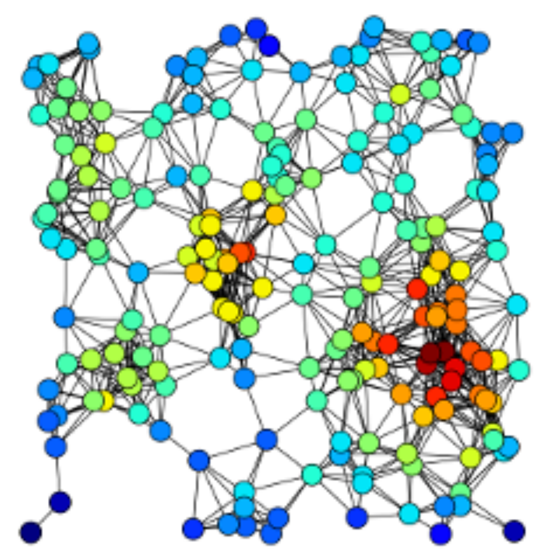
Network centrality

eigenvector centrality



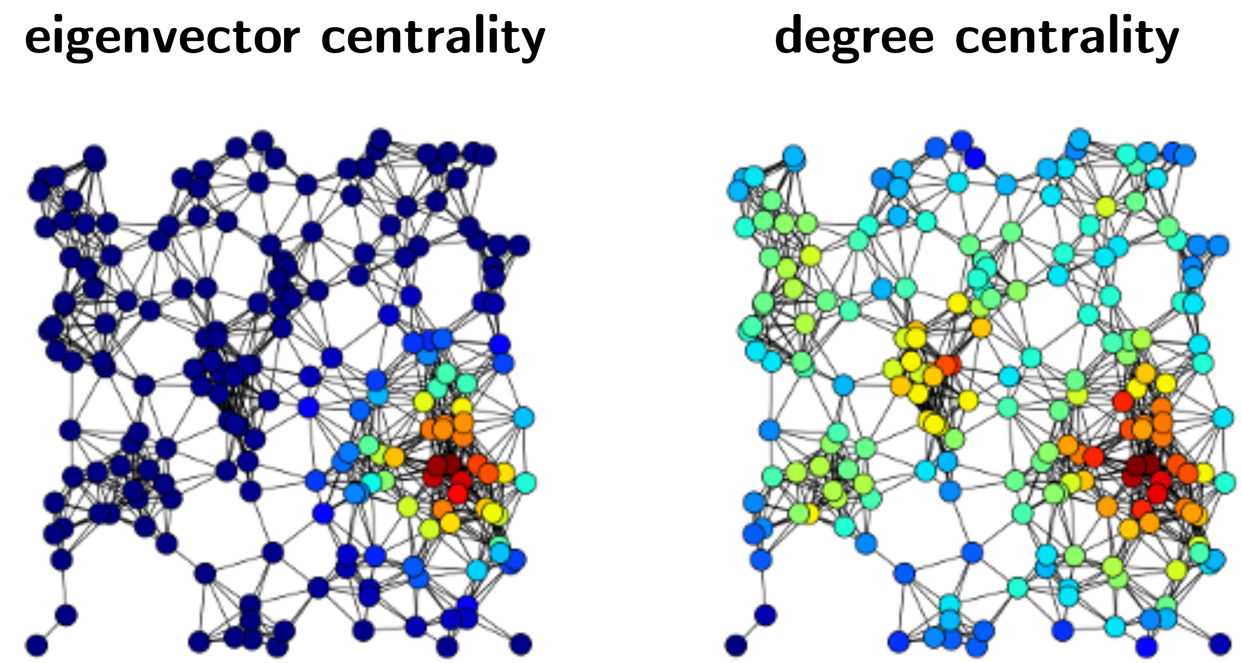
$$Wx = \lambda_{\max} x$$

degree centrality



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

Network centrality

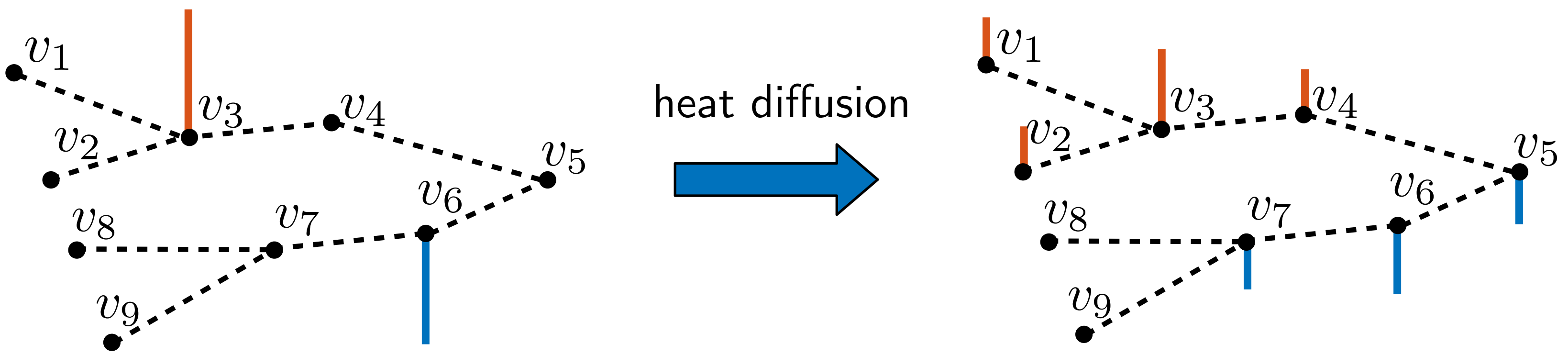


$$Wx = \lambda_{\max} x \qquad d = [d(v_1), \cdots, d(v_N)]$$

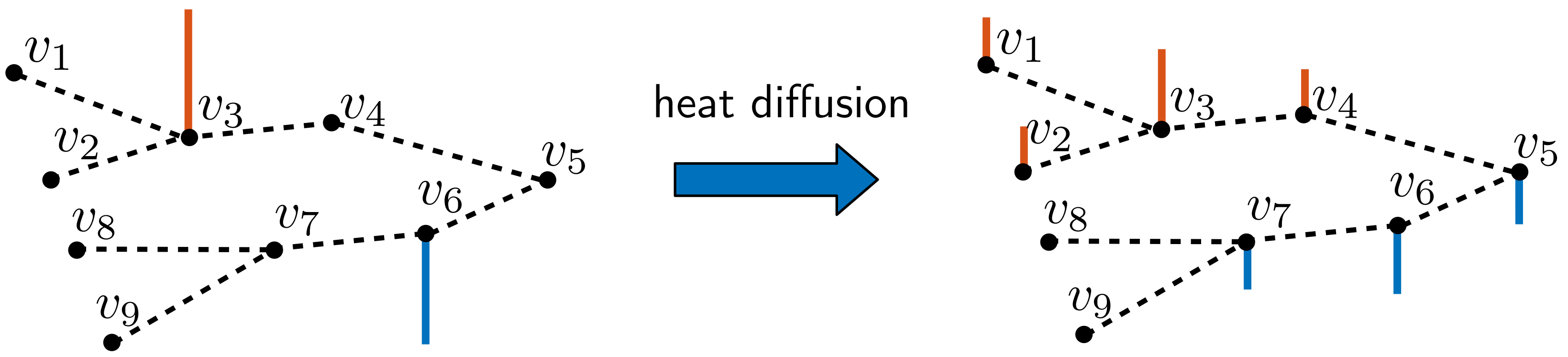
- Google's PageRank is a variant of eigenvector centrality
- eigenvectors of W can also be used to provide a frequency interpretation for graph signals

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

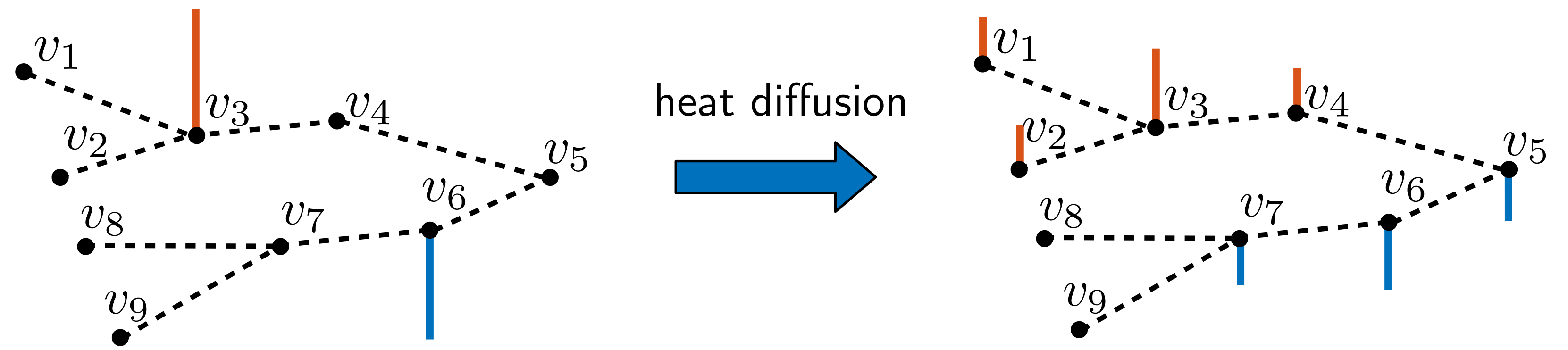


$$\frac{\partial x}{\partial \tau} - Lx = 0$$

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

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

Diffusion on graphs



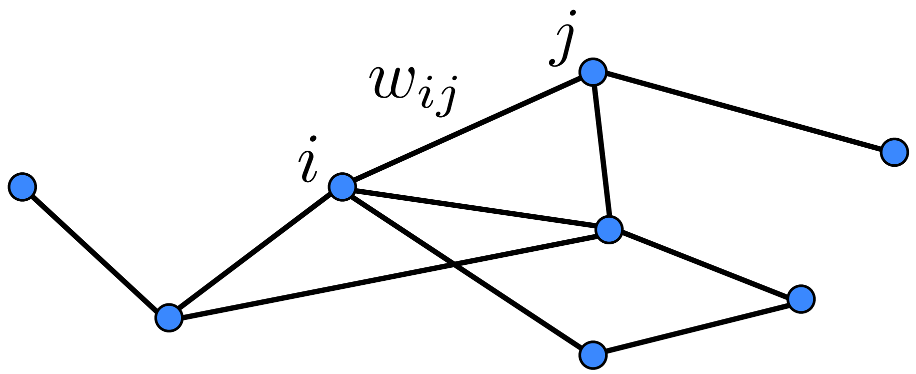
$$\frac{\partial x}{\partial \tau} - Lx = 0$$
$$x(v, 0) = x_0(v)$$

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

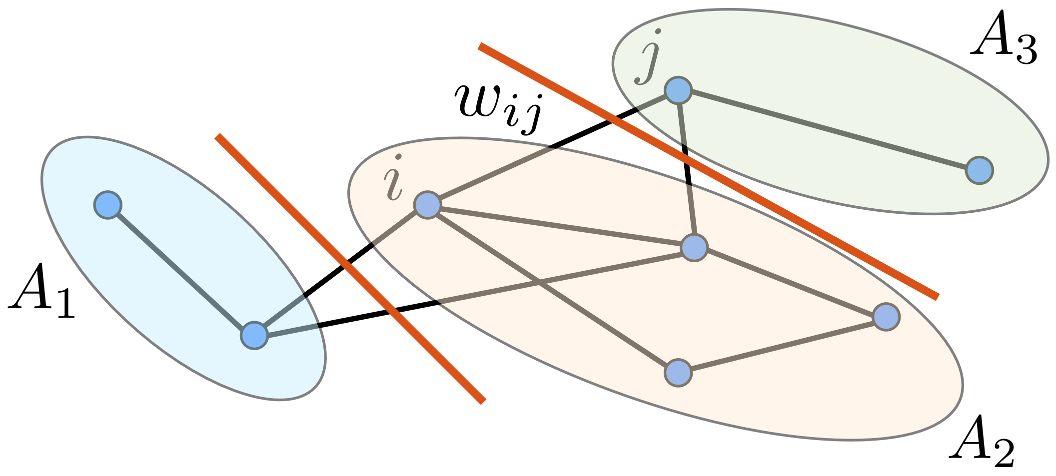
- heat diffusion on graphs is a typical physical process on graphs
- other possibilities exist (e.g., random walk on graphs)
- many have an interpretation of filtering on graphs

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

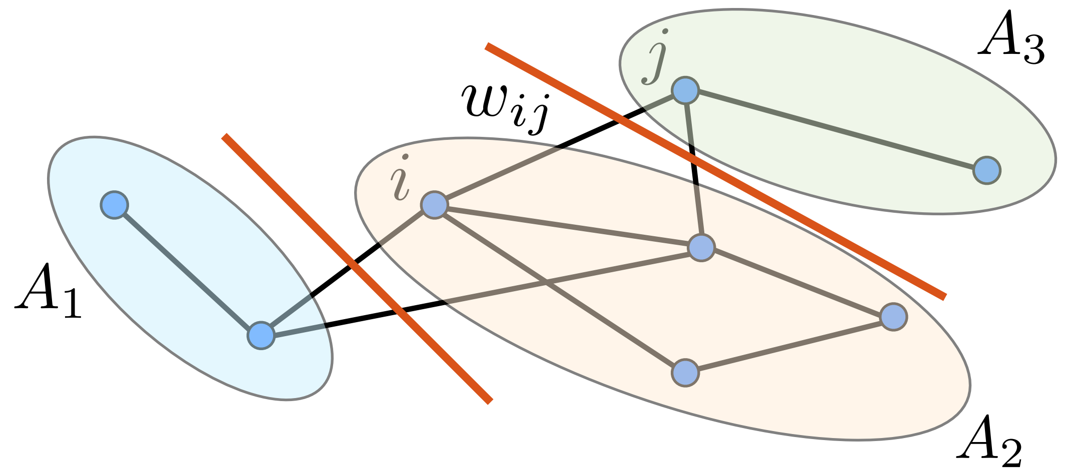
Graph clustering (community detection)



Graph clustering (community detection)

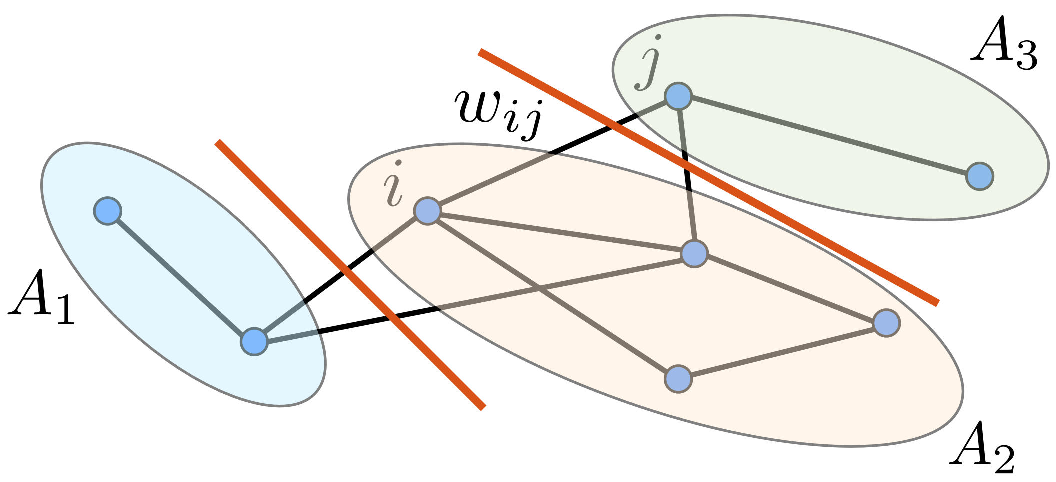


Graph clustering (community detection)



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

Graph clustering (community detection)

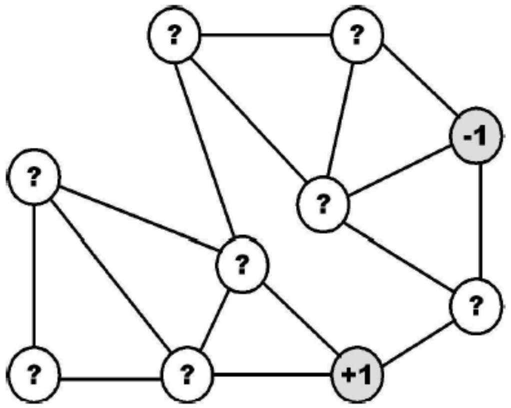


$$NCut(A_1, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{vol(A_i)}$$

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

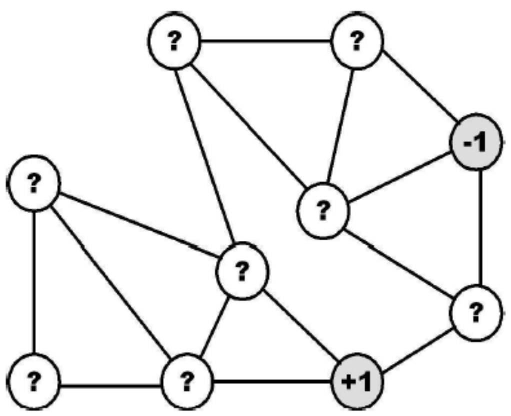
Ulrike von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, 2007.

Semi-supervised learning



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

Semi-supervised learning

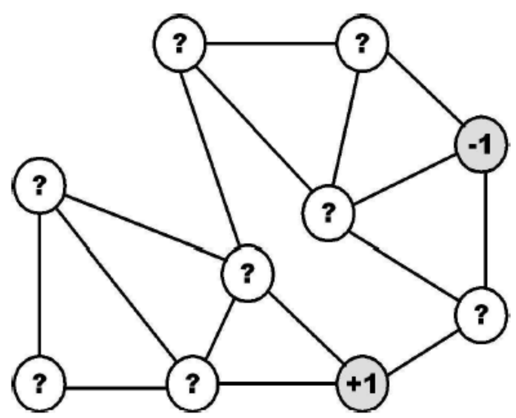


$$y : \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

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

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

Semi-supervised learning



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

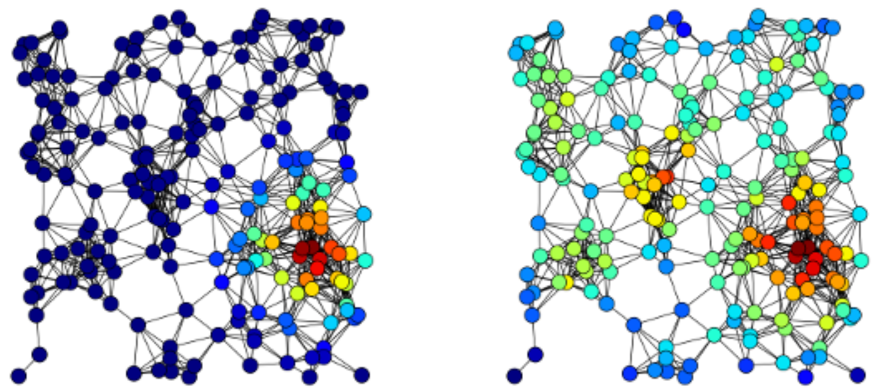
$$\min_{x \in \mathbb{R}^N} ||y - x||_2^2 + \alpha \, x^T L x,$$

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

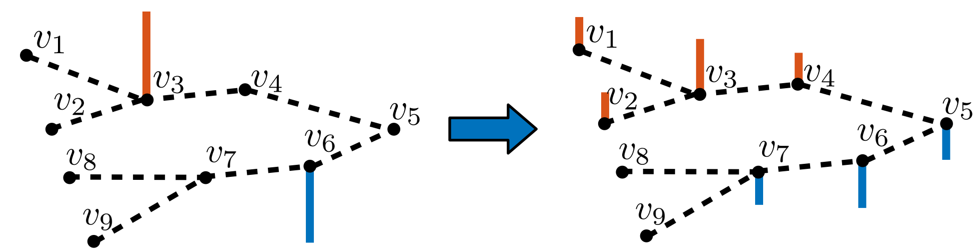
Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, CMU, 2005.

GSP and the literature

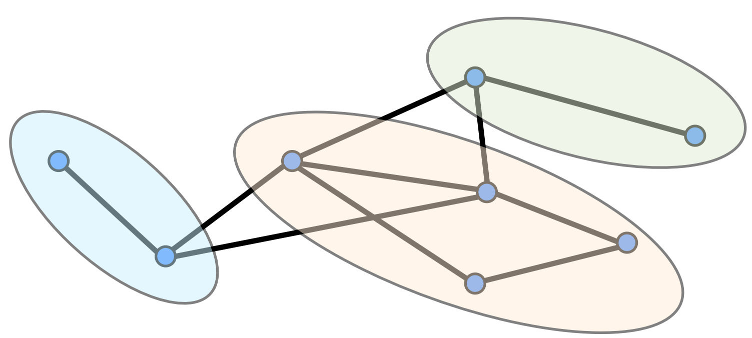
centrality, diffused information, cluster membership, node labels (and node features in general) can ALL be viewed as graph signals



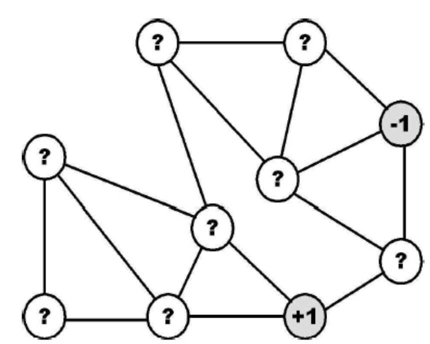
network science



diffusion on graphs



unsupervised learning (dimensionality reduction, clustering)

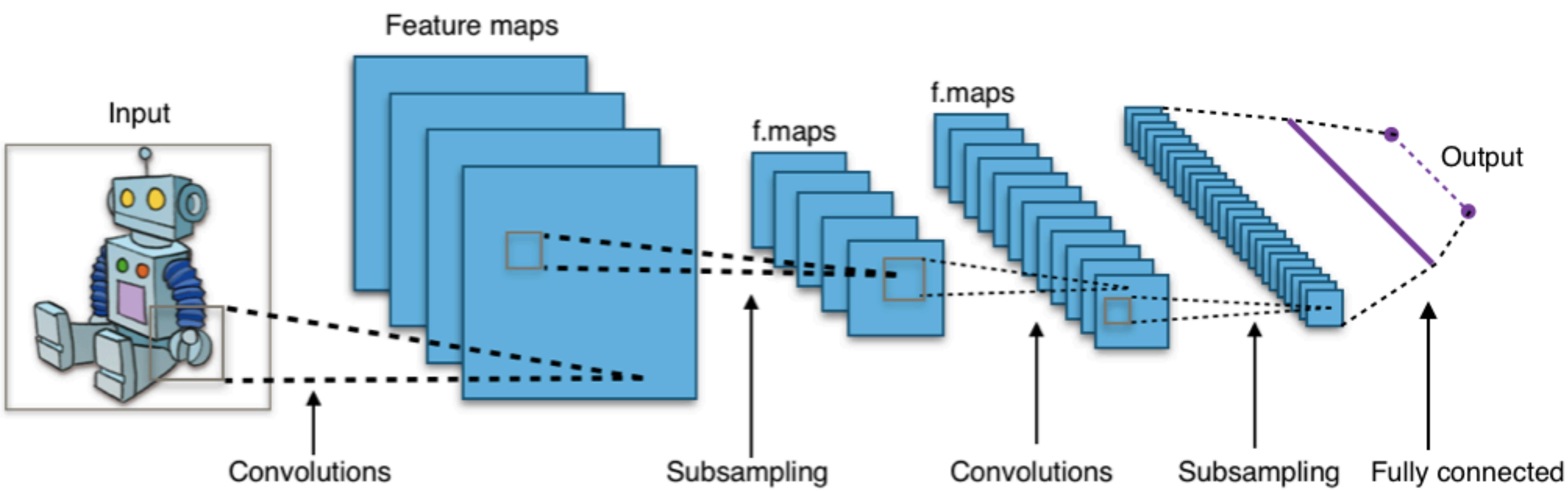


semi-supervised learning

Outline

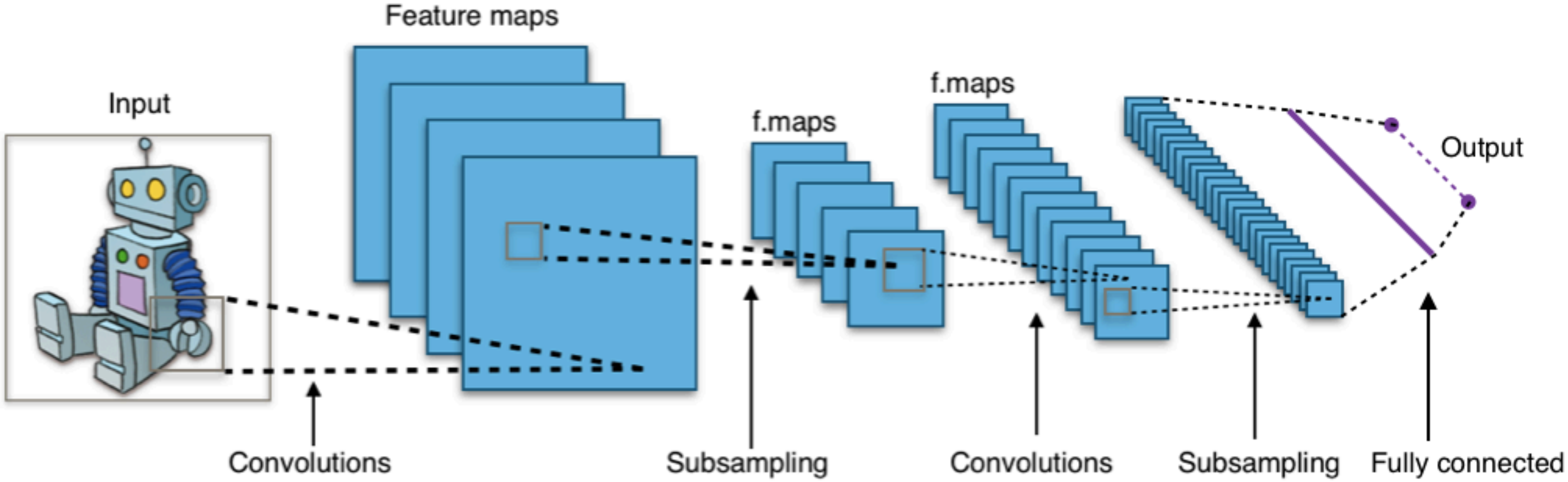
- Graph signal processing (GSP): Basic concepts
- Filtering of graph signals: Basic tool of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

Deep learning on graphs

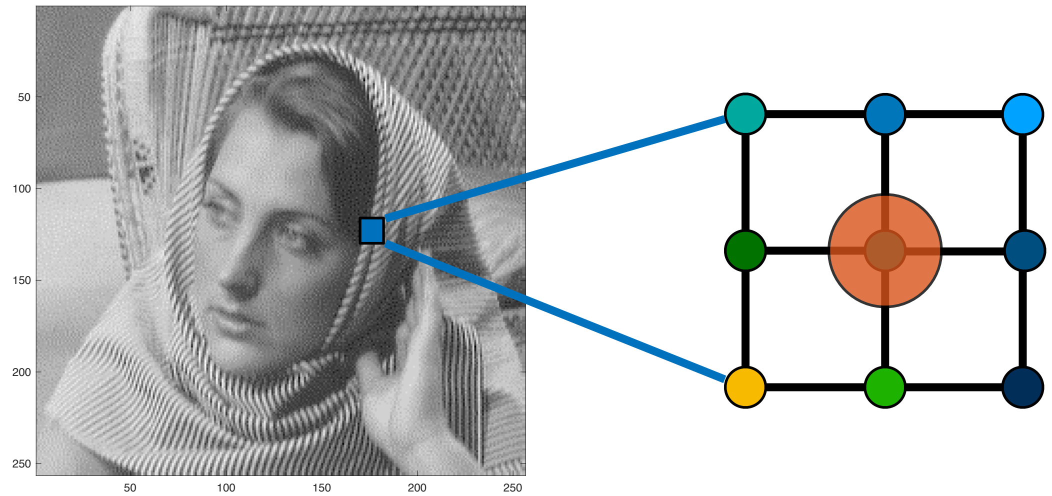


https://en.wikipedia.org/wiki/File:Typical_cnn.png

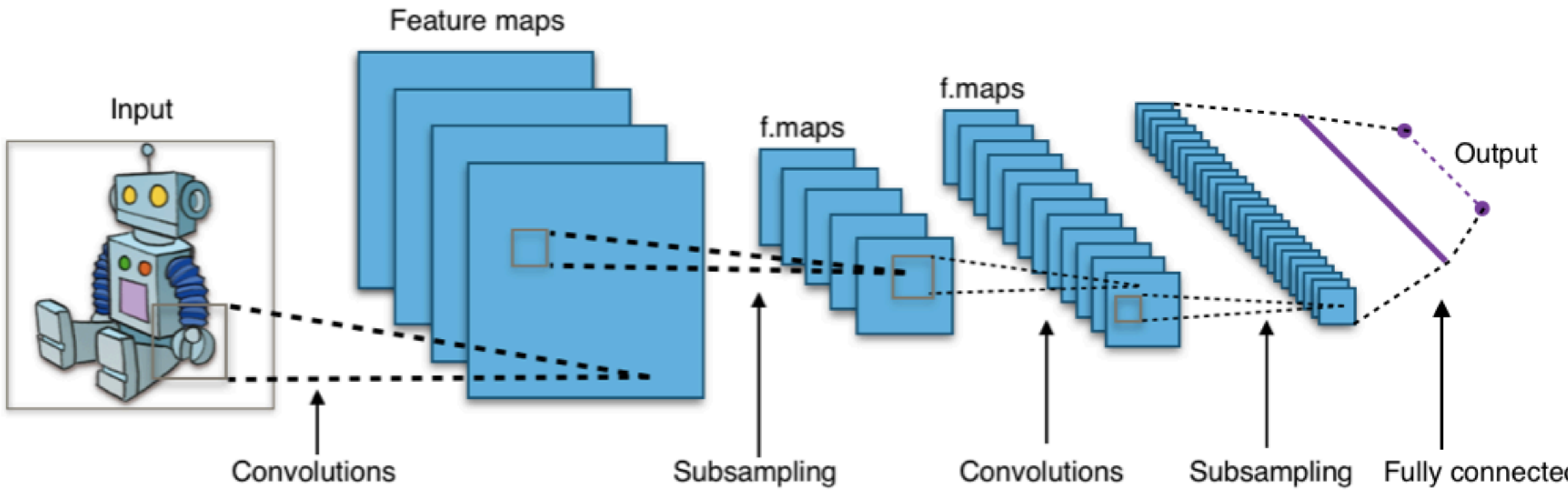
Deep learning on graphs



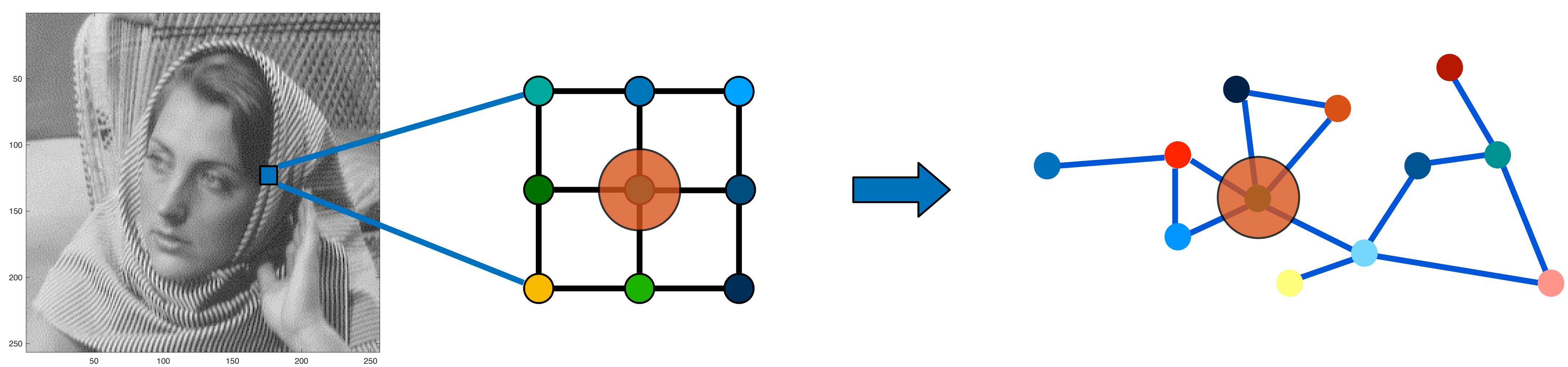
https://en.wikipedia.org/wiki/File:Typical_cnn.png



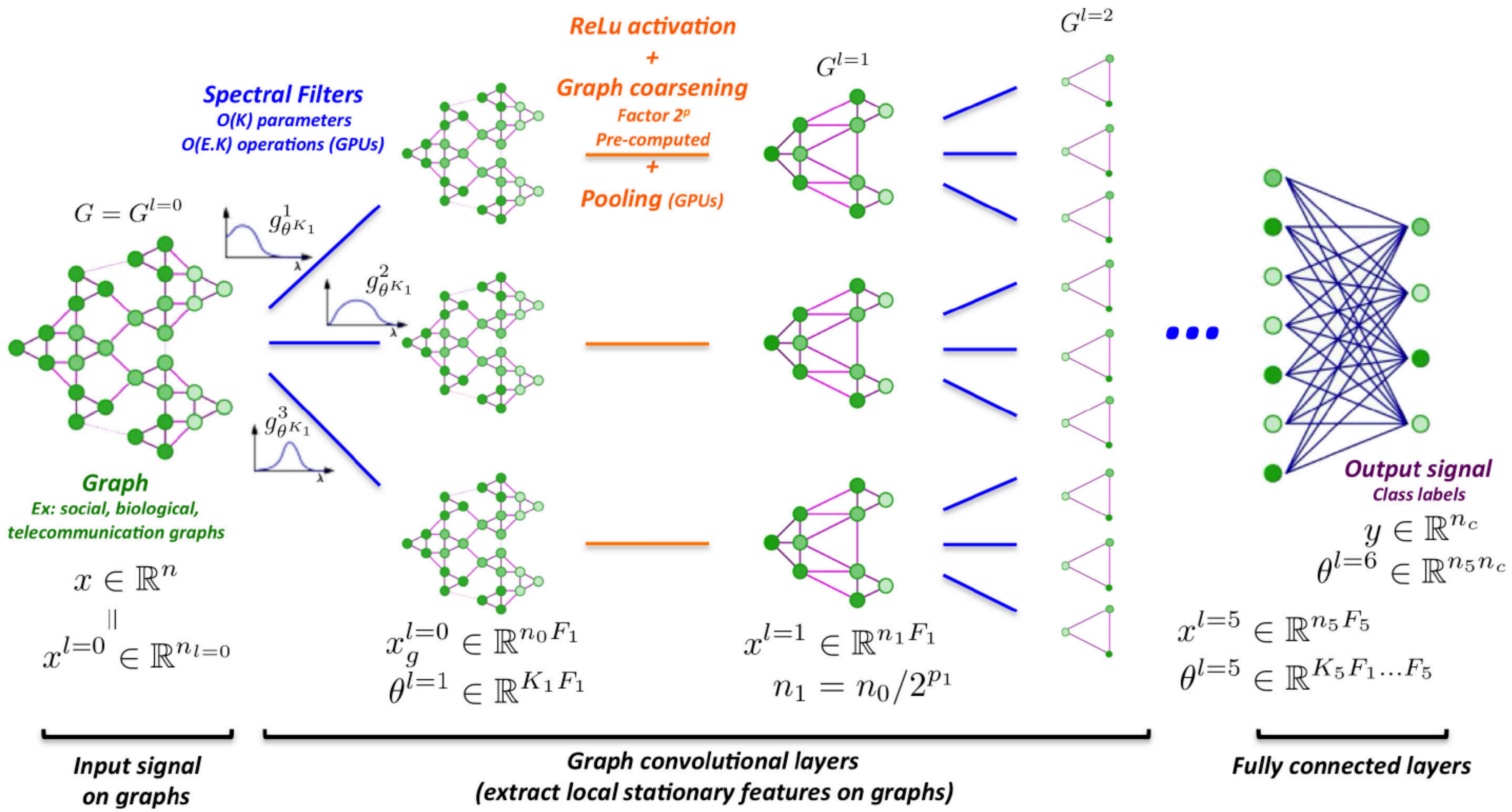
Deep learning on graphs



https://en.wikipedia.org/wiki/File:Typical_cnn.png

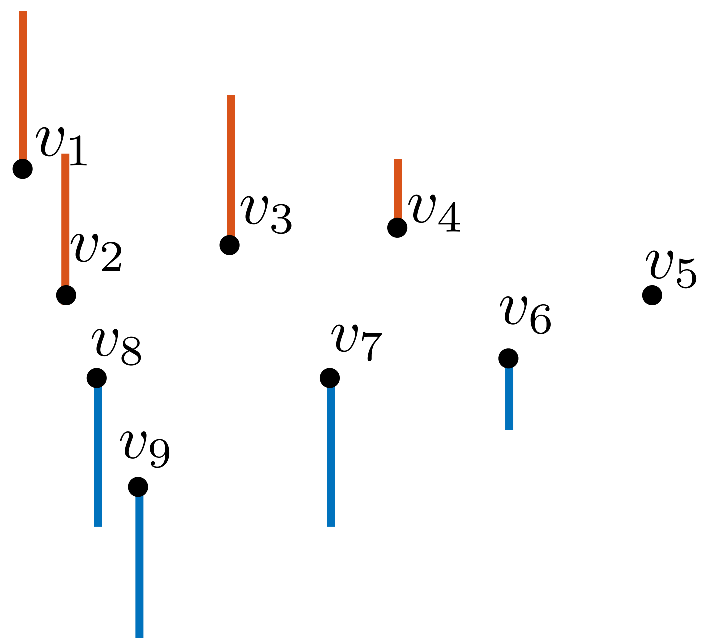


Deep learning on graphs

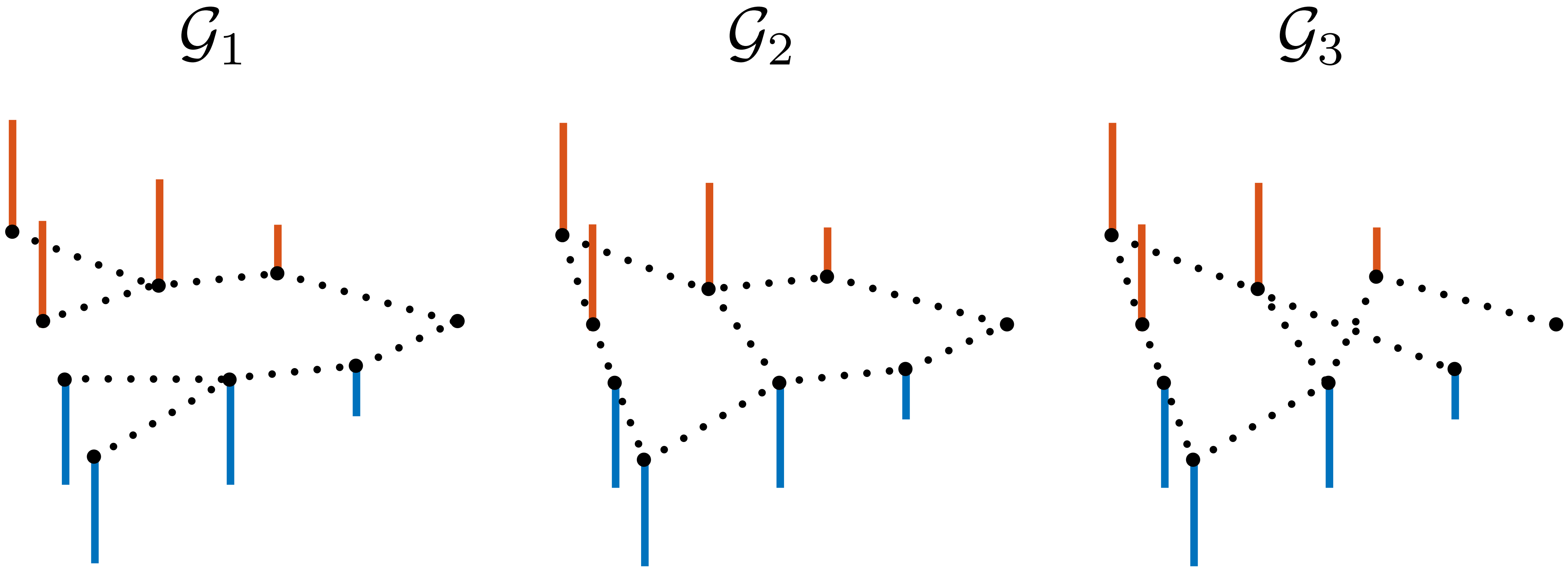


Defferrard et al., "Convolutional neural networks on graphs with fast localized spectral filtering," NIPS, 2016.

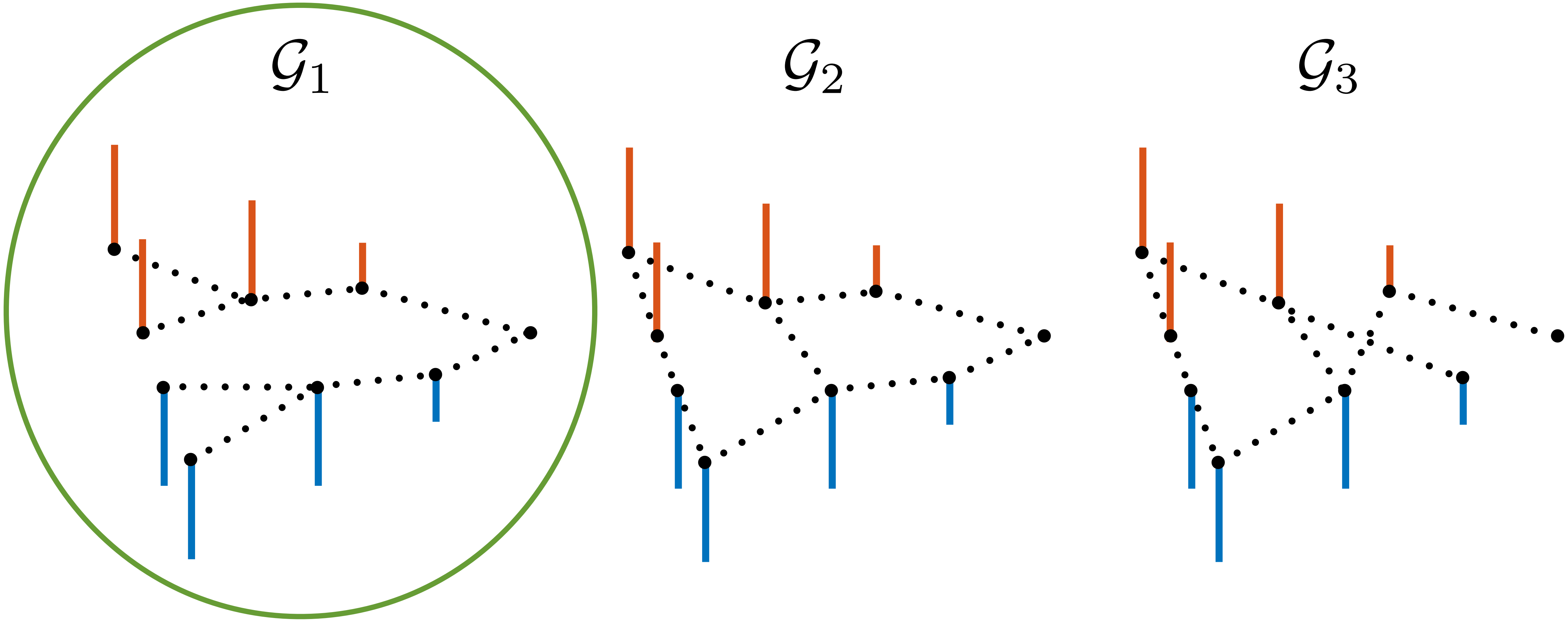
Inferring graph structure from data



Inferring graph structure from data



Inferring graph structure from data



idea: choose the structure that enforces certain signal characteristics

Dong et al., "Learning Laplacian matrix in smooth graph signal representations," IEEE TSP, 2016.

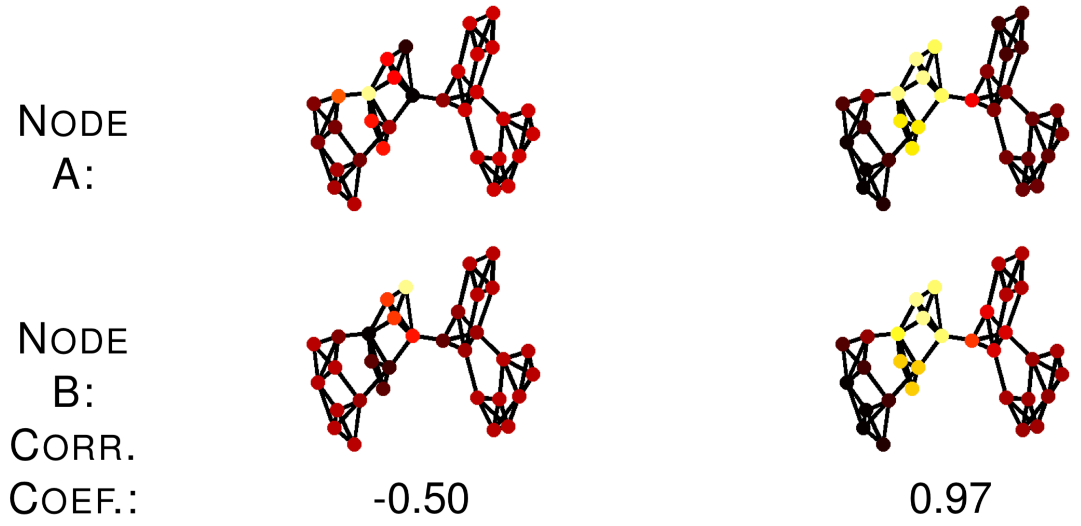
Outline

- Graph signal processing (GSP): Basic concepts
- Filtering of graph signals: Basic tool of GSP
- Connection with literature
- Research topics inspired by GSP
- Applications

Application I: Community detection

spectral graph wavelets
at different scales:

$$\boldsymbol{D}_s(a,b) = 1 - \frac{\psi_{s,a}^\top \psi_{s,b}}{||\psi_{s,a}||_2 ||\psi_{s,b}||_2}.$$



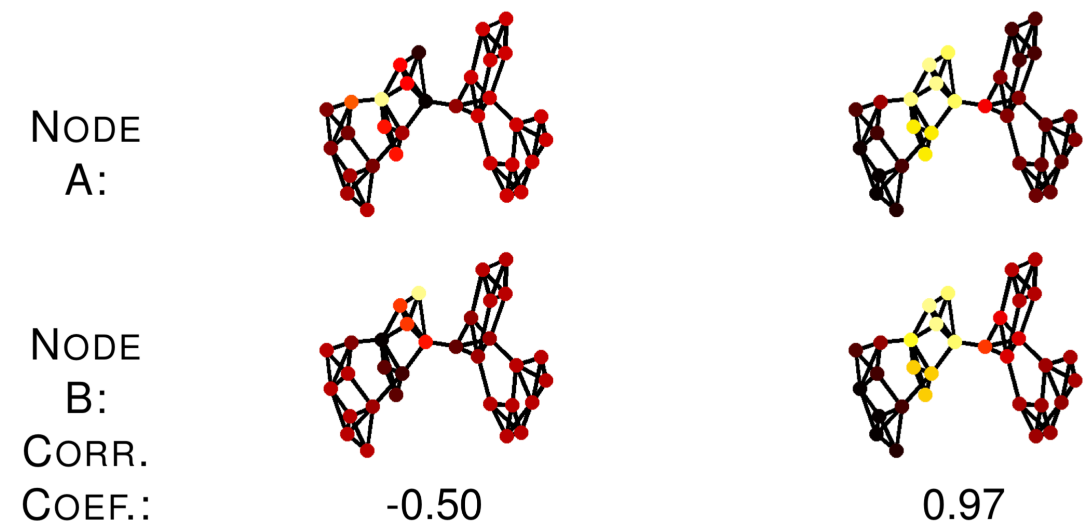
small scale **large scale**

Hammond et al., "Wavelets on graphs via spectral graph theory," Applied and Computational Harmonic Analysis, 2011.
Tremblay and Borgnat, "Graph wavelets for multiscale community mining," IEEE TSP, 2014.

Application I: Community detection

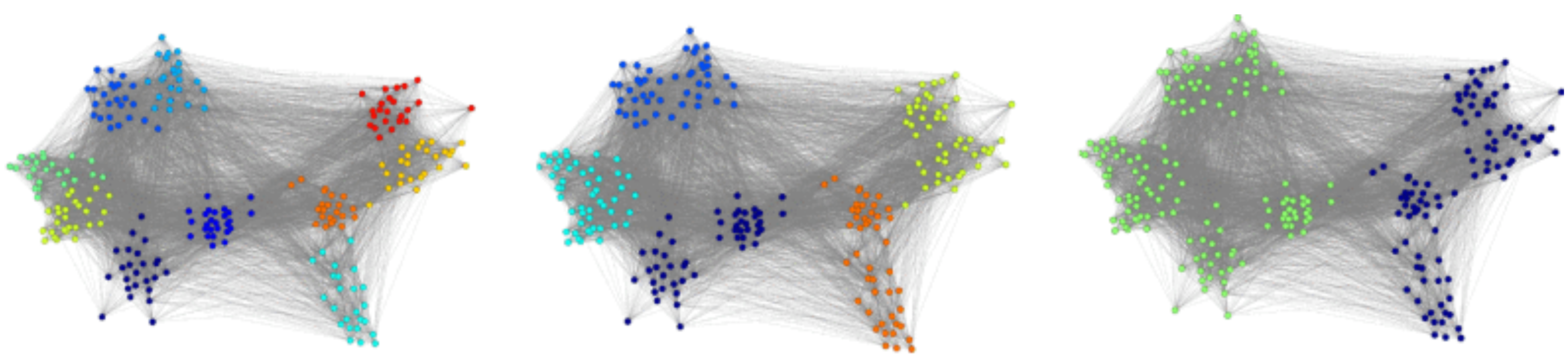
spectral graph wavelets at different scales:

$$\boldsymbol{D}_s(a,b) = 1 - \frac{\psi_{s,a}^{\top}\psi_{s,b}}{||\psi_{s,a}||_2 ||\psi_{s,b}||_2}.$$



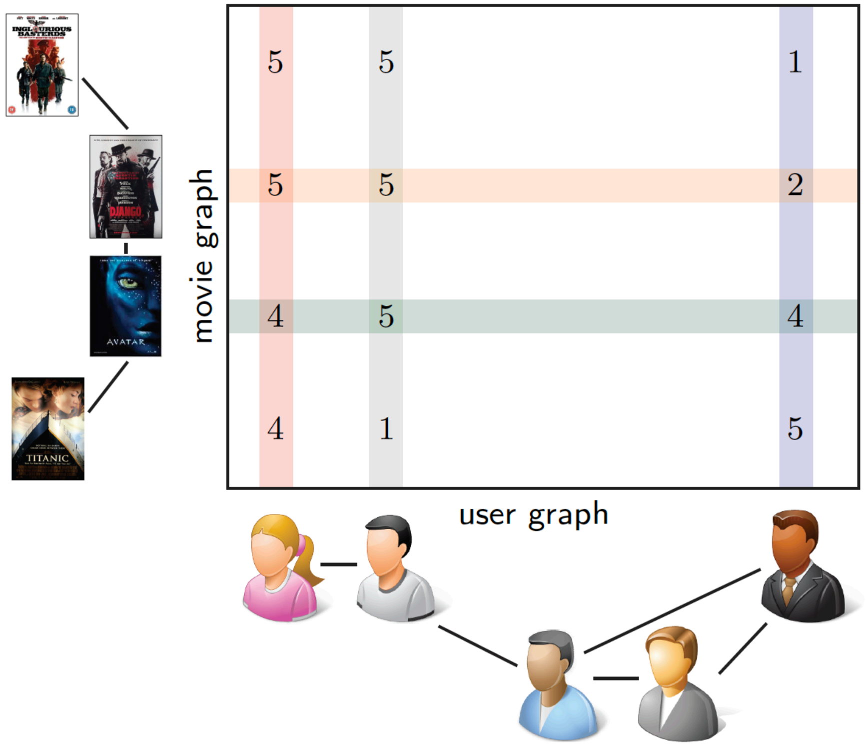
small scale **large scale**

multi-scale community detection:

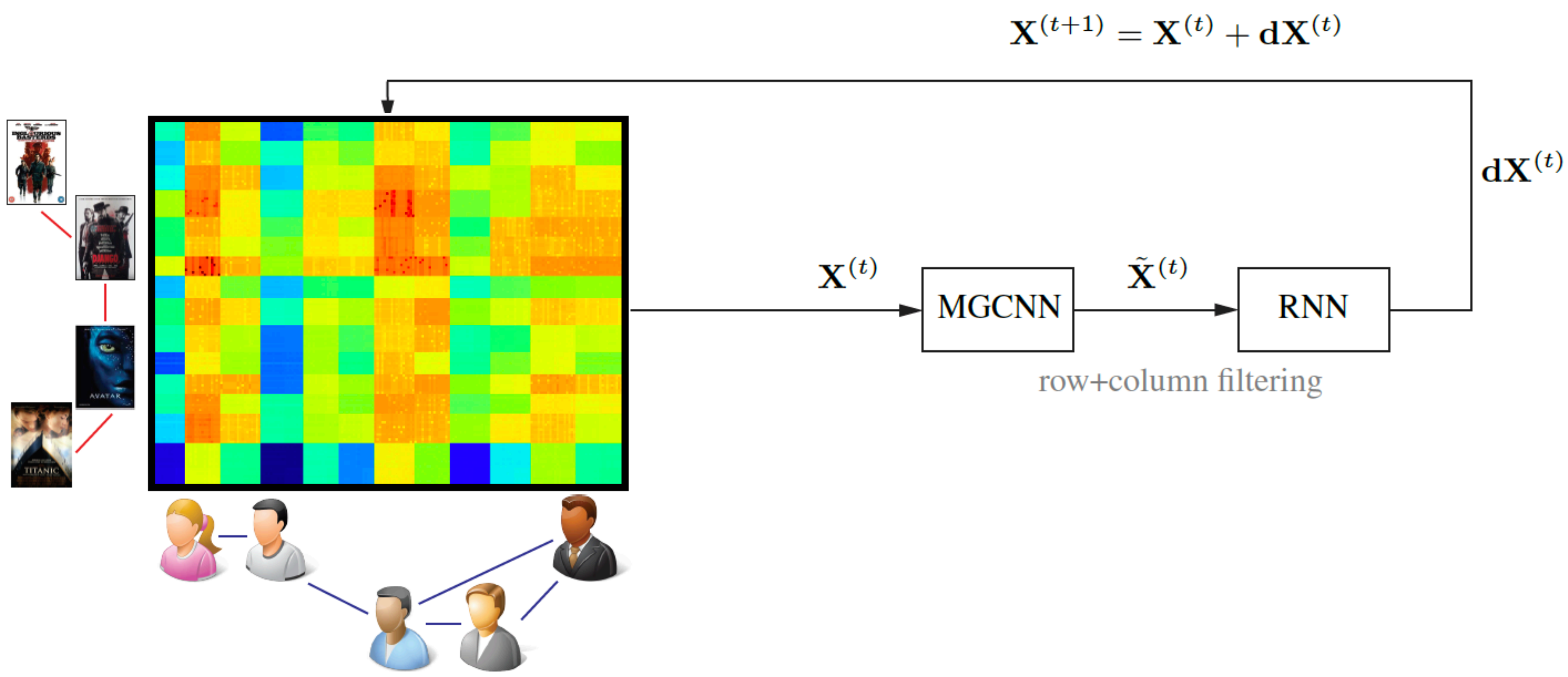


Hammond et al., "Wavelets on graphs via spectral graph theory," Applied and Computational Harmonic Analysis, 2011.
Tremblay and Borgnat, "Graph wavelets for multiscale community mining," IEEE TSP, 2014.

Application II: Recommender systems

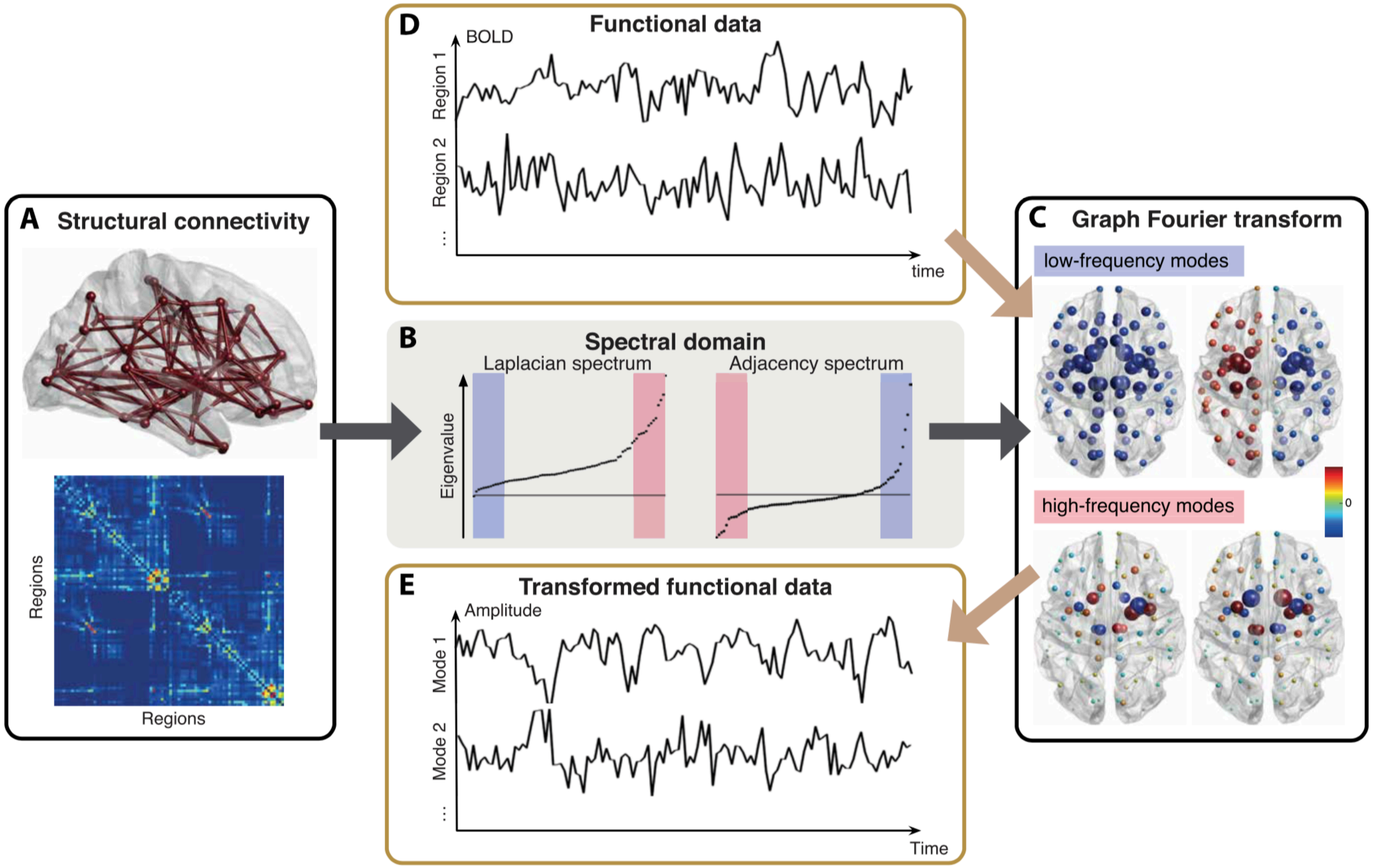


Application II: Recommender systems

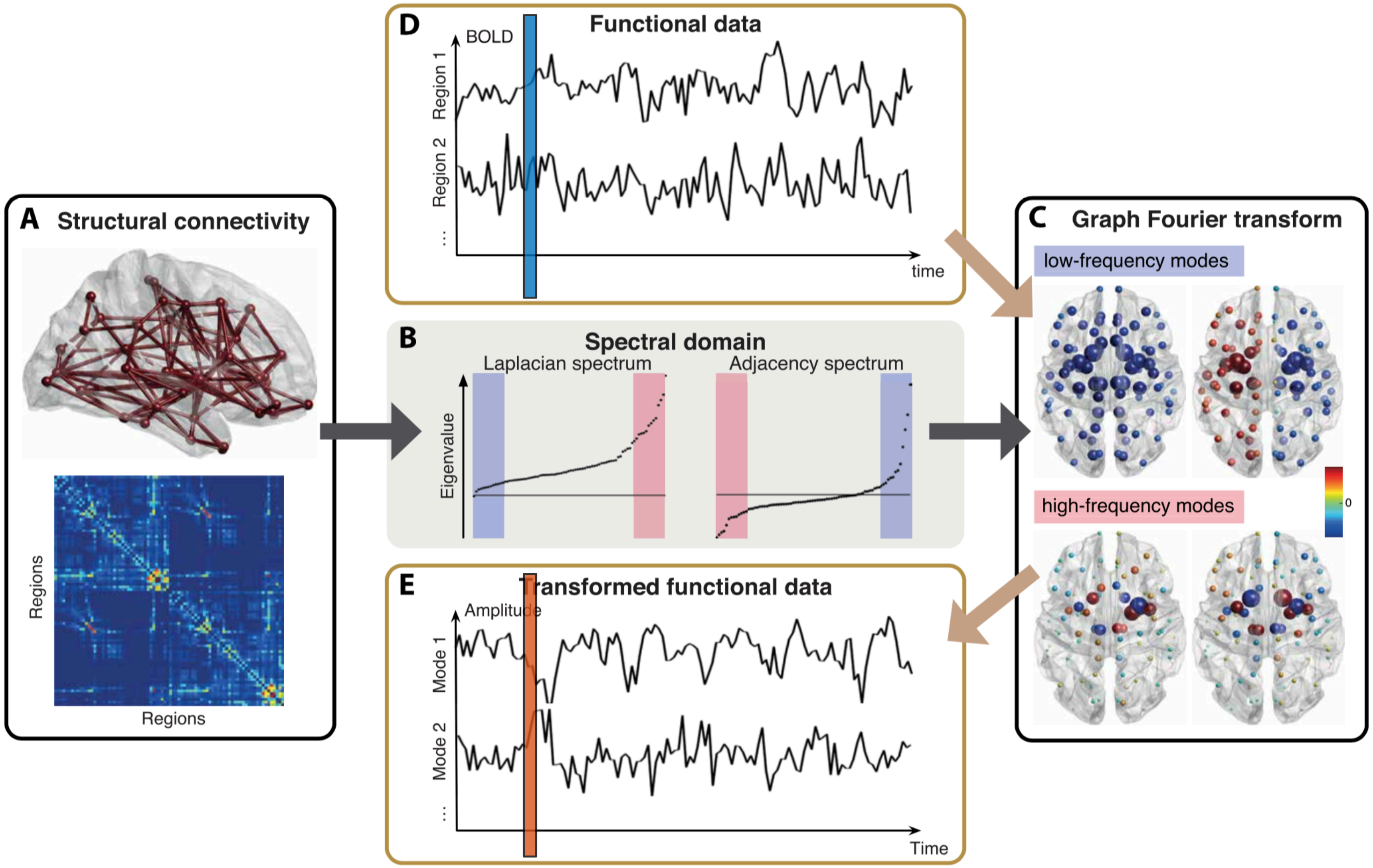


Monti et al., "Geometric matrix completion with recurrent multi-graph neural networks," NIPS, 2017.

Application III: Functional brain imaging

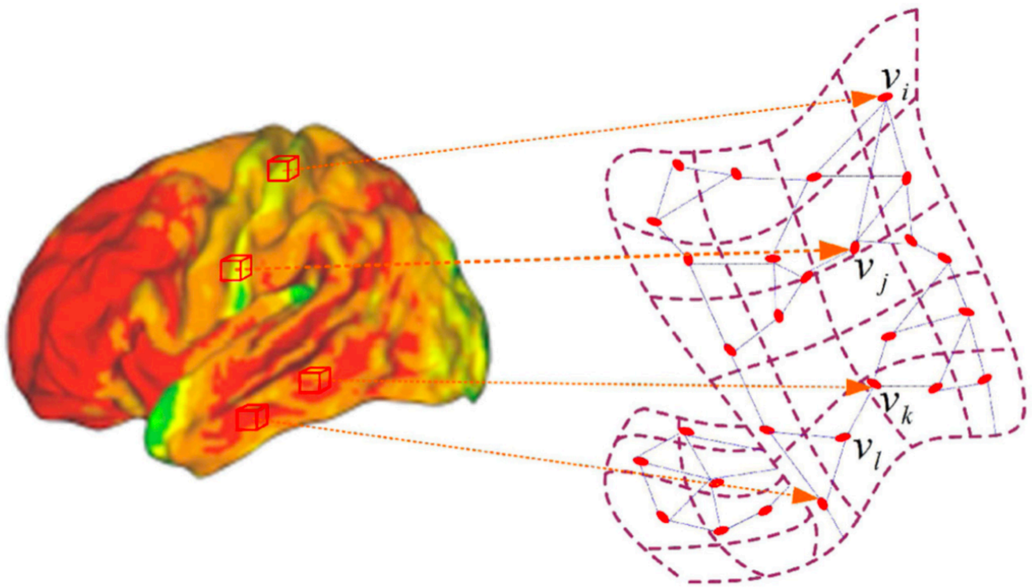


Application III: Functional brain imaging



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

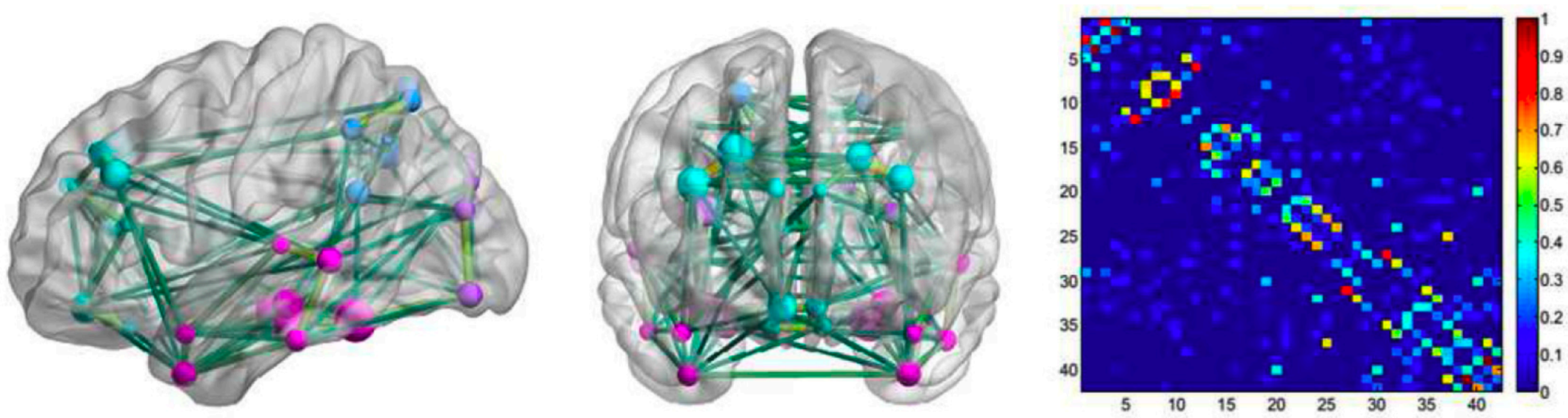
Application IV: Inferring brain connectivity



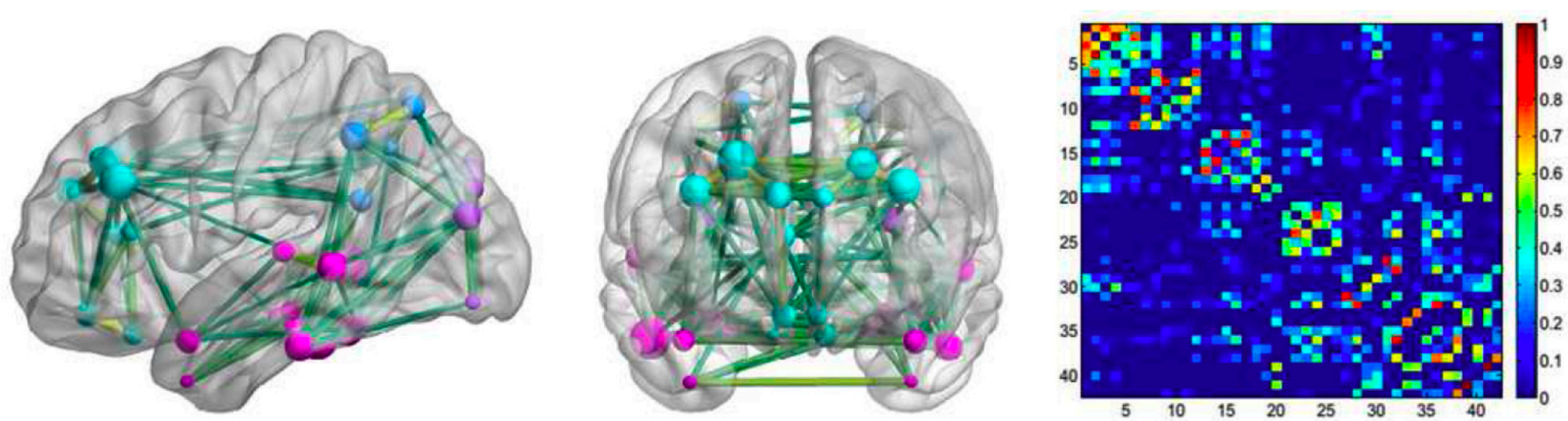
Hu et al., "A spectral graph regression model for learning brain connectivity of Alzheimer's disease," PLOS ONE, 2015.

Application IV: Inferring brain connectivity

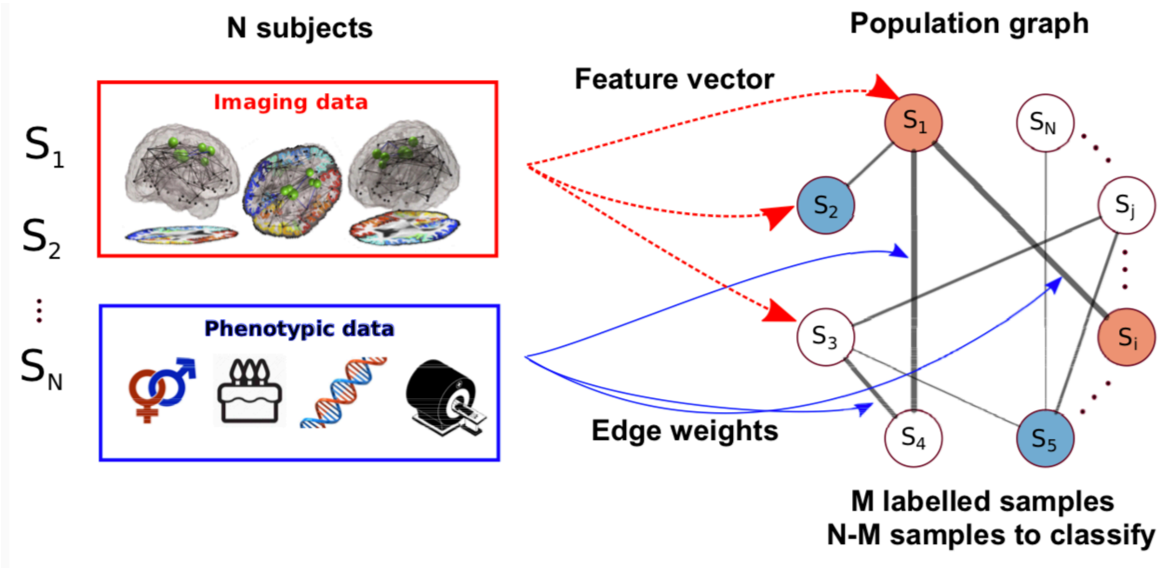
Alzheimer's disease



normal control

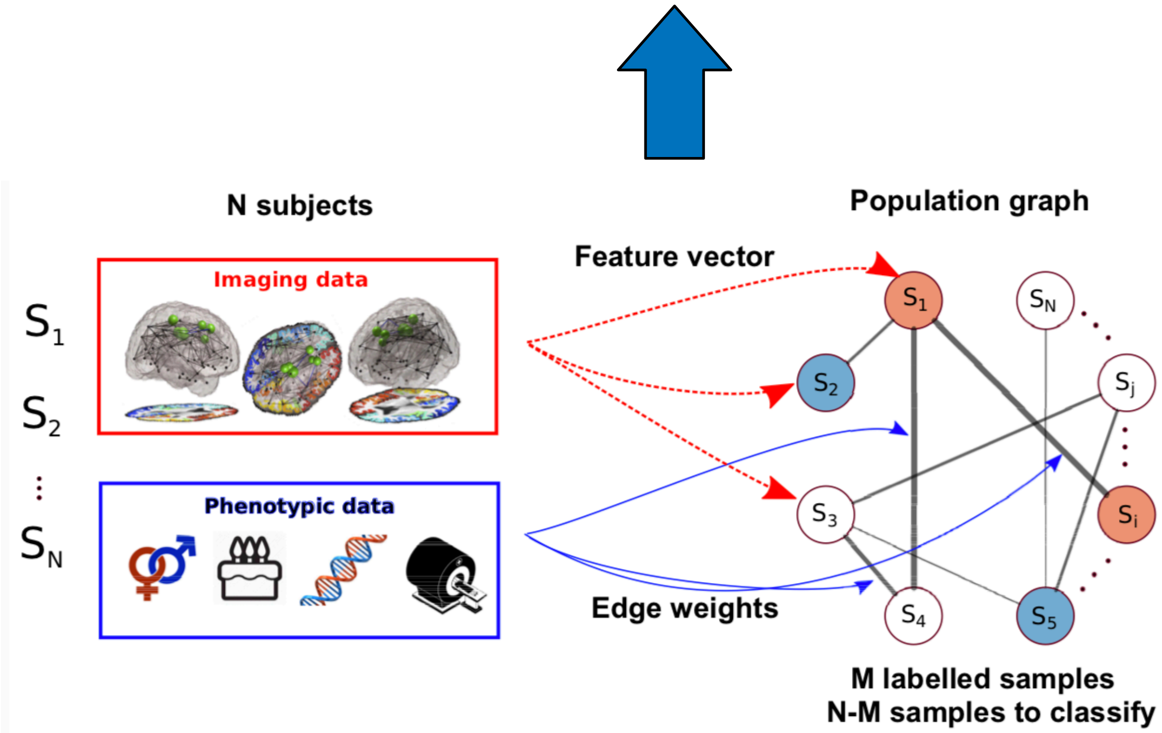


Application V: Disease classification



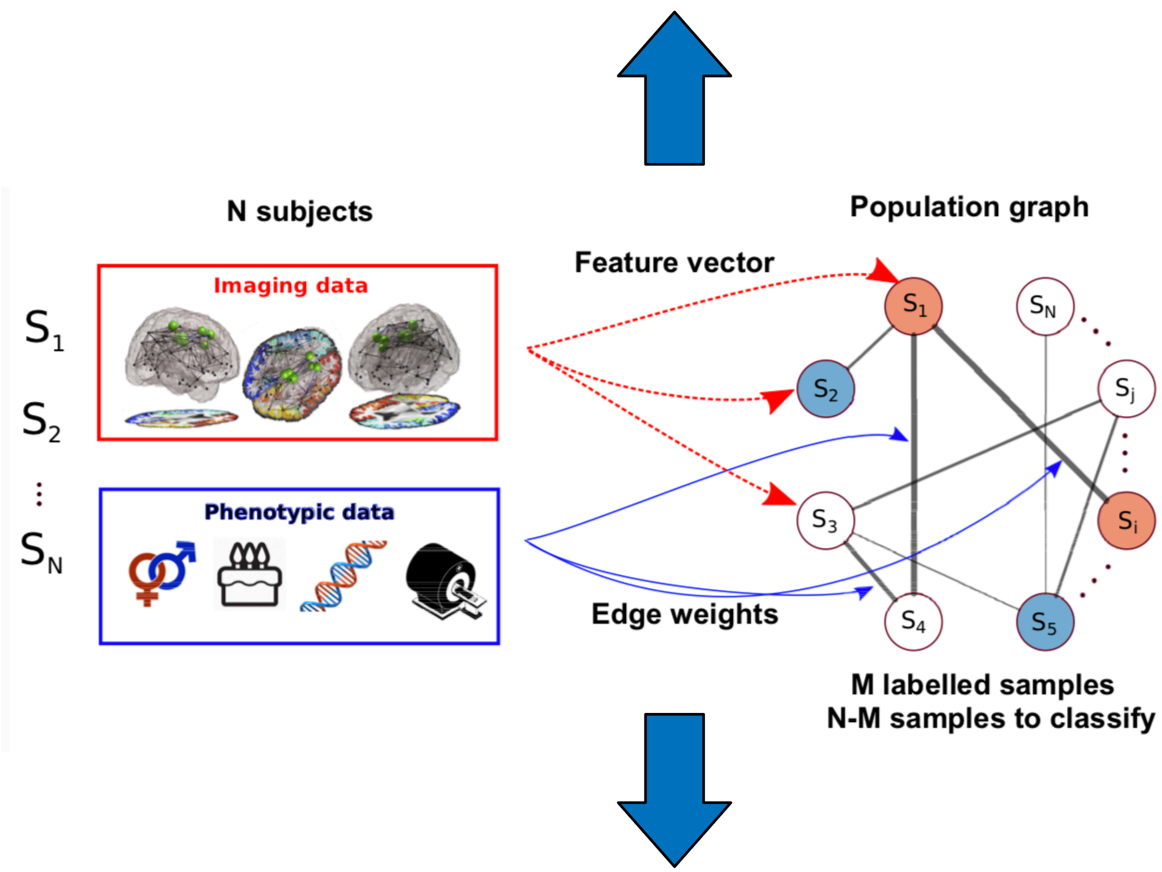
Application V: Disease classification

features extracted from brain analysis



Application V: Disease classification

features extracted from brain analysis

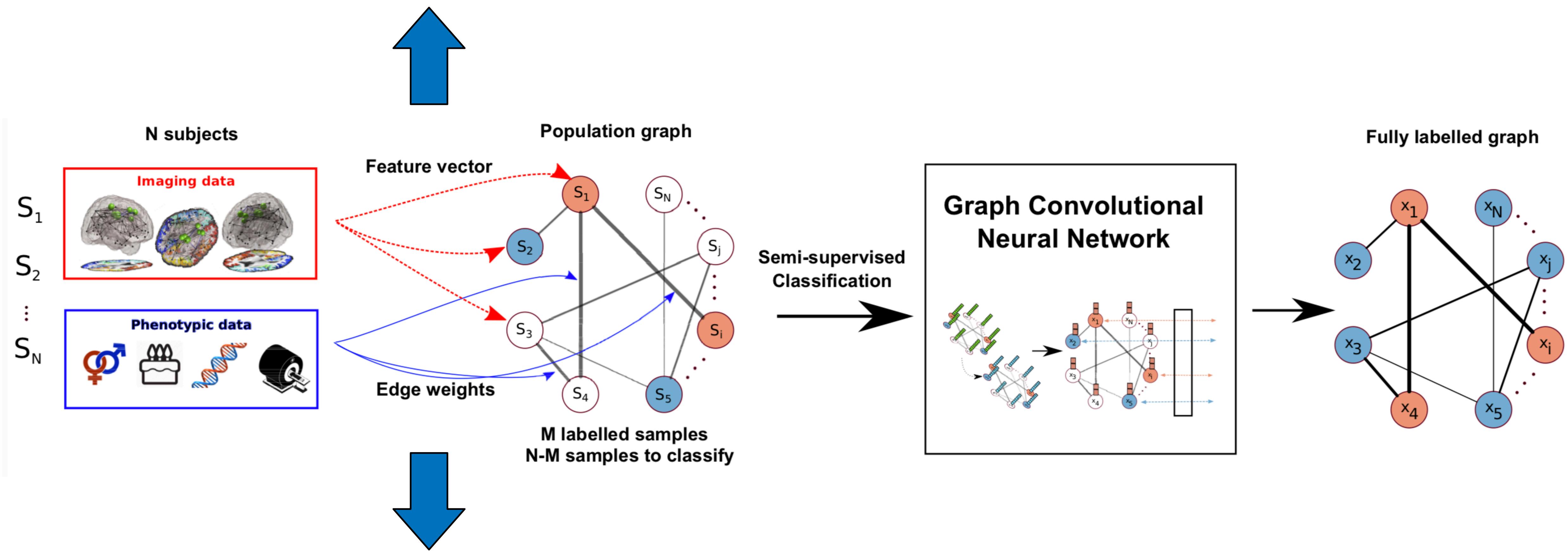


similarity in phenotypic data

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

Application V: Disease classification

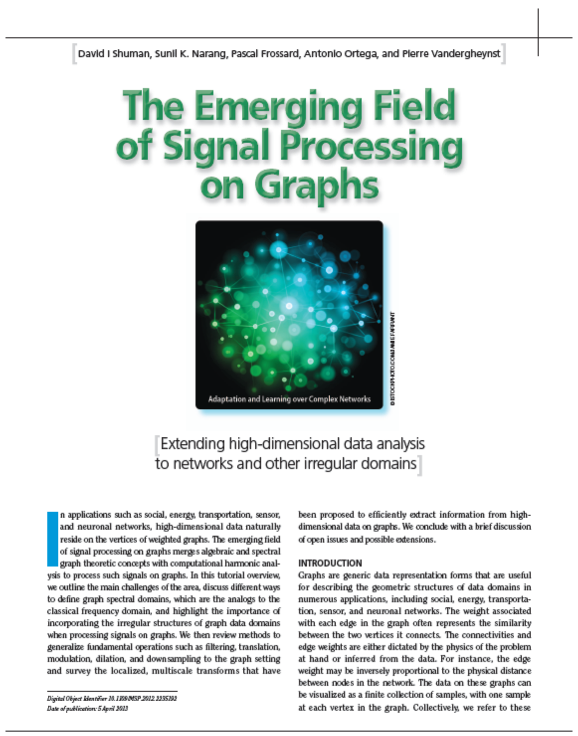
features extracted from brain analysis

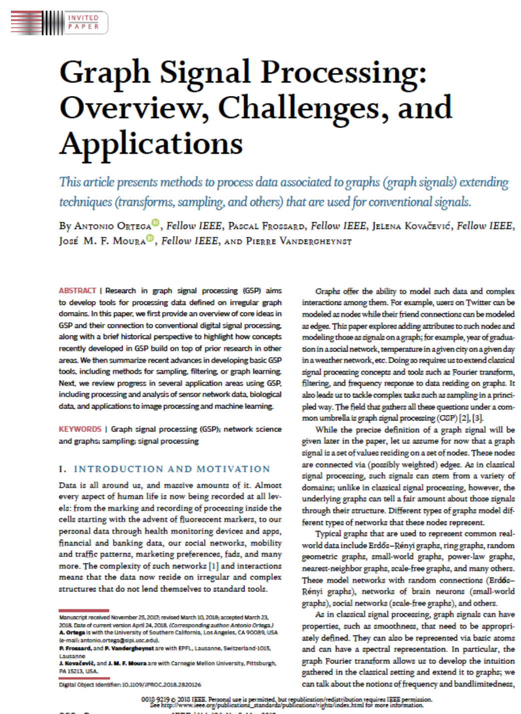


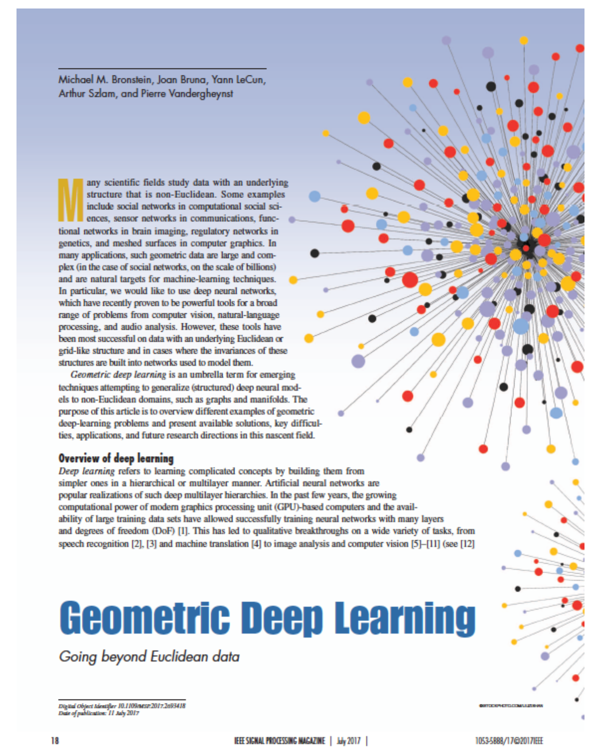
similarity in phenotypic data

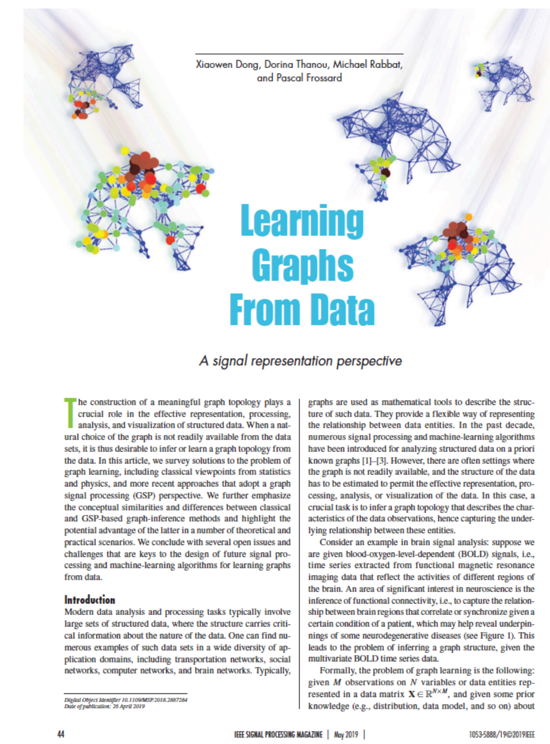
Papers & Resources

- Tutorial/overview papers:









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