Learning Graphs From Data

A Signal Processing Perspective

Xiaowen Dong

Oxford-Man Institute

Department of Engineering Science

OxCSML Seminar, Oxford, May 2019

OXFORD-MAN INSTITUTE

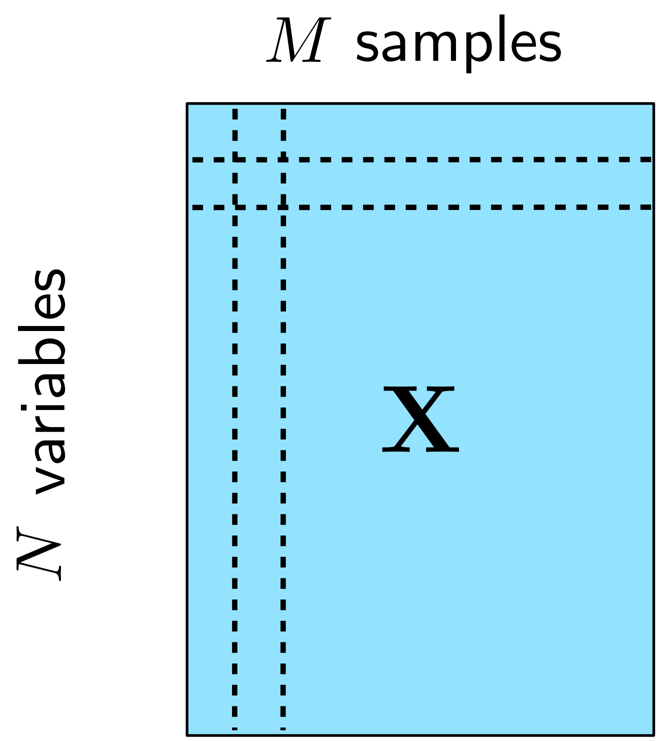
UNIVERSITY OF OXFORD

Introduction

- What is the problem of graph learning?

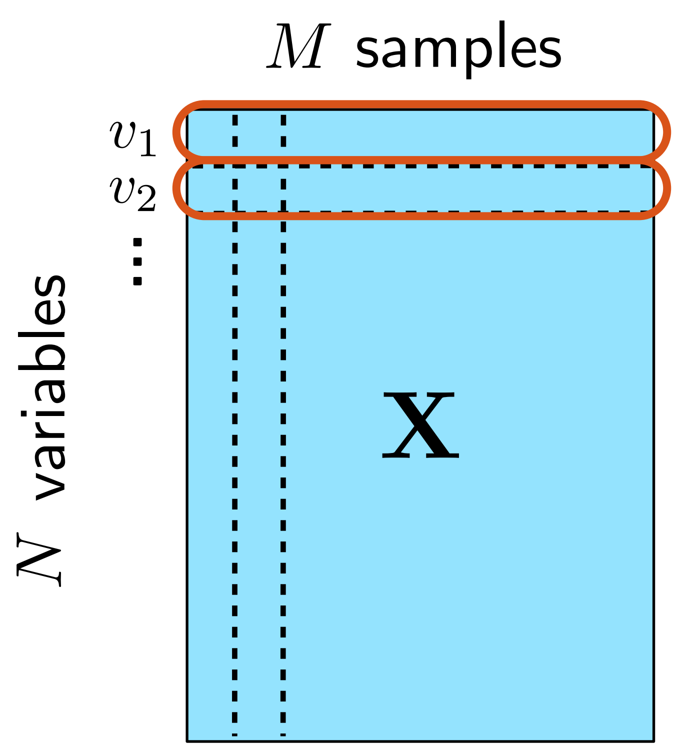
Introduction

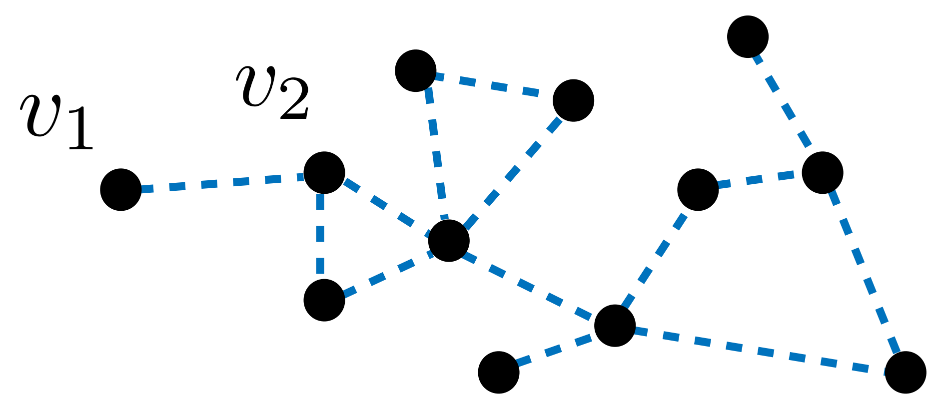
- What is the problem of graph learning?
  - given observations on a number of variables and some prior knowledge (distribution, model, etc)



Introduction

- What is the problem of graph learning?
  - given observations on a number of variables and some prior knowledge (distribution, model, etc)
  - build/learn a measure of pairwise relation between variables (correlation/covariance, graph topology/operator or equivalent)





Introduction

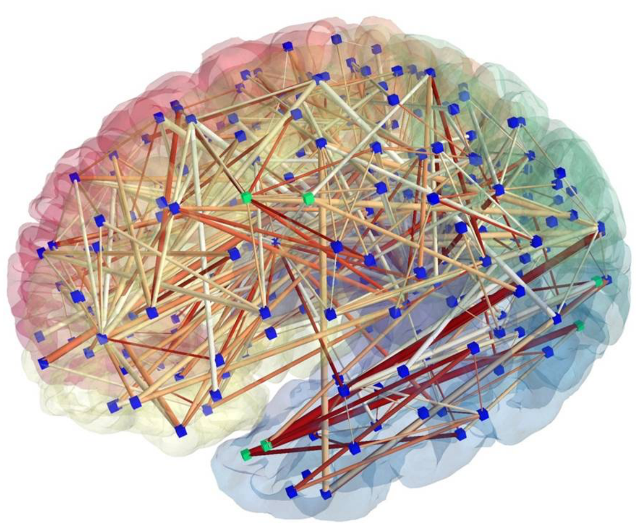
- Why is it important?
  - learned relation (graph) captures underlying structure of data

Introduction

- Why is it important?
  - learned relation (graph) captures underlying structure of data
  - learning relation between entities benefits numerous application domains

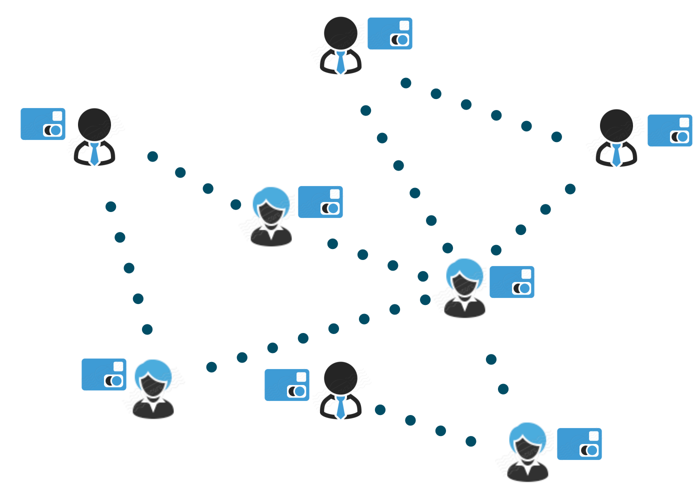
Introduction

- Why is it important?
  - learned relation (graph) captures underlying structure of data
  - learning relation between entities benefits numerous application domains



Input: fMRI recordings in brain regions

Objective: functional connectivity between brain regions



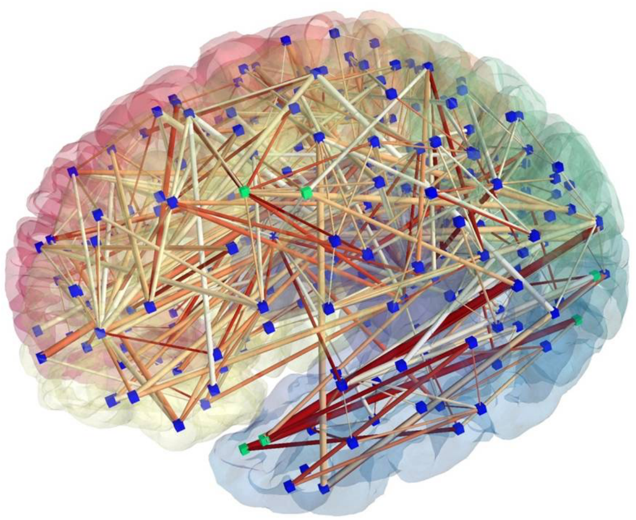
Input: history of individual activities

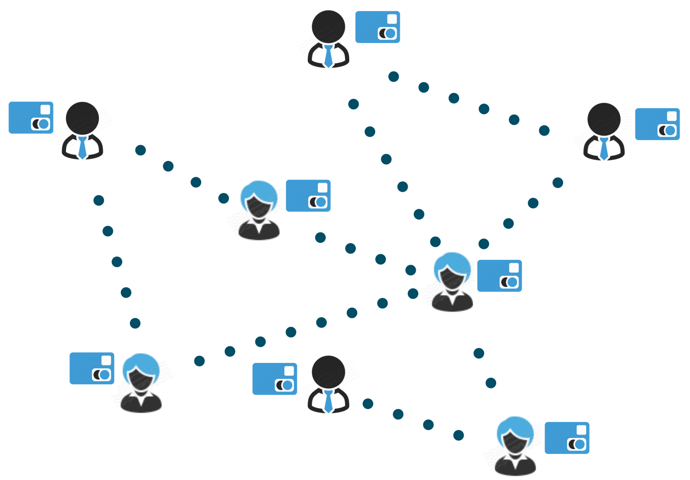
Objective: behavioural similarity/ influence between people

image credit: http://blog.myesr.org/mri-reveals-the-human-connectome/, https://www.iconexperience.com

Introduction

- Why is it important?
  - learned relation (graph) captures underlying structure of data
  - learning relation between entities benefits numerous application domains
  - learned relation may help predict future observations





Input: fMRI recordings in brain regions

Objective: functional connectivity between brain regions

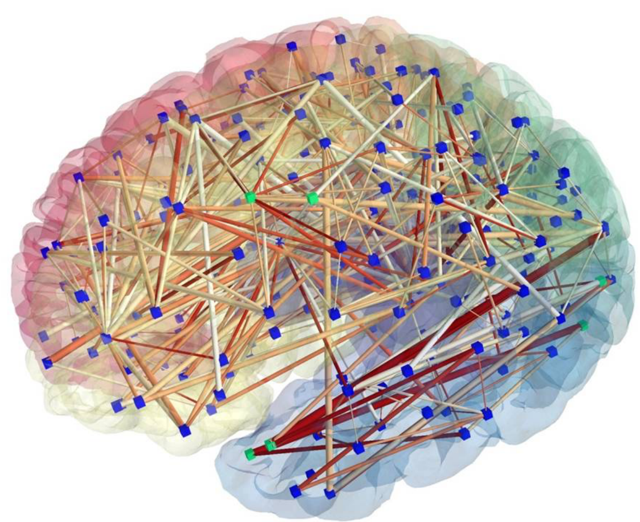
Input: history of individual activities

Objective: behavioural similarity/ influence between people

image credit: http://blog.myesr.org/mri-reveals-the-human-connectome/, https://www.iconexperience.com

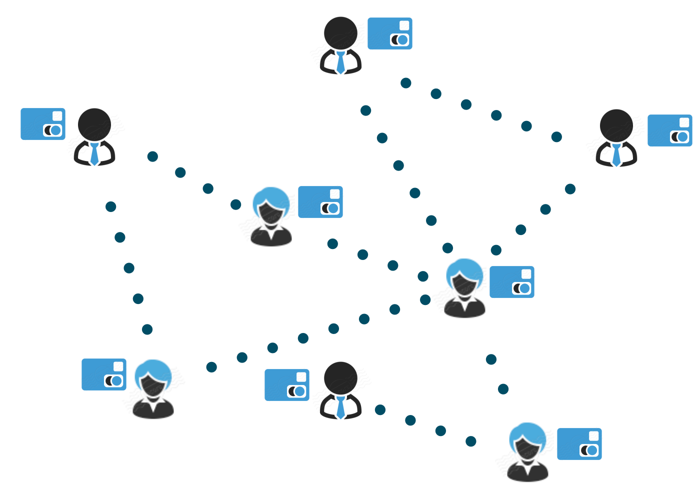
Introduction

- Why is it important?
  - learned relation (graph) captures underlying structure of data
  - learning relation between entities benefits numerous application domains
  - learned relation may help predict future observations



Input: fMRI recordings in brain regions

Objective: functional connectivity between brain regions



Input: history of individual activities

Objective: behavioural similarity/ influence between people

how do we build/learn the graph?

image credit: http://blog.myesr.org/mri-reveals-the-human-connectome/, https://www.iconexperience.com

Outline

- A (very partial) literature overview
- A signal processing perspective
  - A brief introduction to graph signal processing (GSP)
  - GSP approaches for graph learning
- Concluding remarks

Outline

- A (very partial) literature overview
- A signal processing perspective
  - A brief introduction to graph signal processing (GSP)
  - GSP approaches for graph learning
- Concluding remarks

A (very partial) literature overview

- Simple and intuitive methods
  - sample correlation
  - similarity function (e.g., Gaussian RBF)

A (very partial) literature overview

- Simple and intuitive methods
  - sample correlation
  - similarity function (e.g., Gaussian RBF)

- Need for a meaningful data model: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$

A (very partial) literature overview

- Simple and intuitive methods
  - sample correlation
  - similarity function (e.g., Gaussian RBF)

- Need for a meaningful data model: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$

- Two general approaches in the literature
  - statistical models: **F** draws realisations from a distribution determined by **G** (e.g., probabilistic graphical models)

A (very partial) literature overview

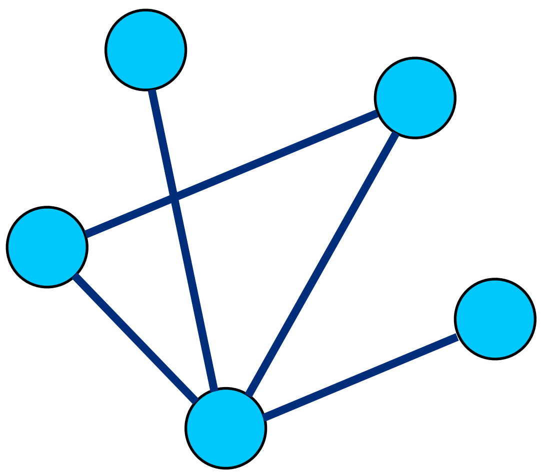
- Simple and intuitive methods
  - sample correlation
  - similarity function (e.g., Gaussian RBF)

- Need for a meaningful data model: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$

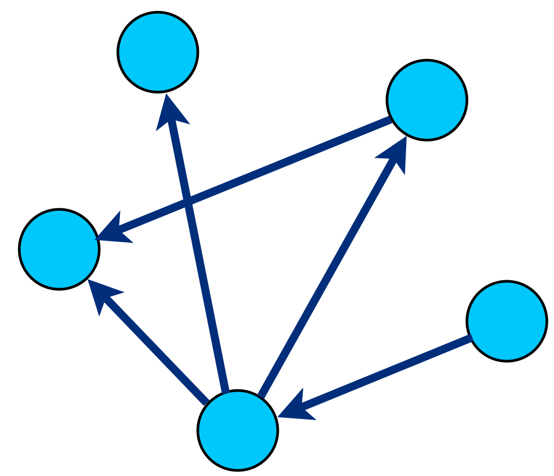
- Two general approaches in the literature
  - statistical models: **F** draws realisations from a distribution determined by **G** (e.g., probabilistic graphical models)
  - physically motivated models: **F** is based on a physical generative process on **G** (e.g., diffusion processes on graphs)

A (very partial) literature overview

- Learning graphical models



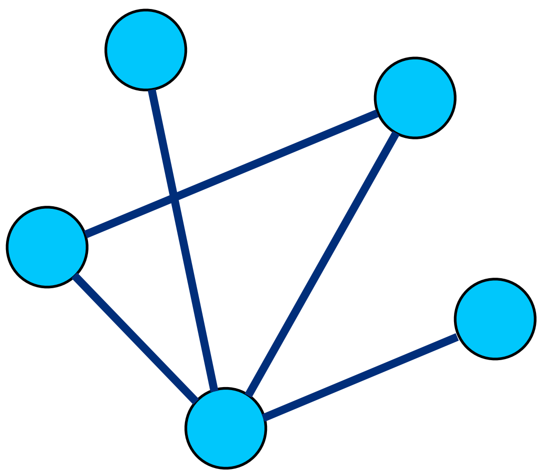
undirected graphical models:
Markov random fields (MRFs)



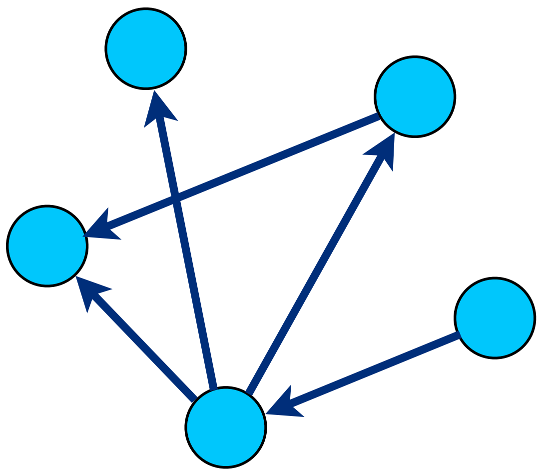
directed graphical models:
Bayesian networks (BNs)

A (very partial) literature overview

- Learning graphical models



undirected graphical models:
Markov random fields (MRFs)



directed graphical models:
Bayesian networks (BNs)

A (very partial) literature overview

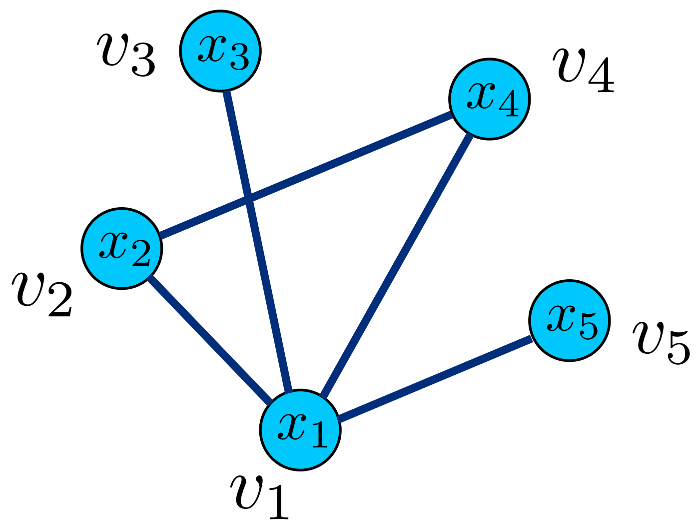
- Learning pairwise MRF

A (very partial) literature overview

- Learning pairwise MRF

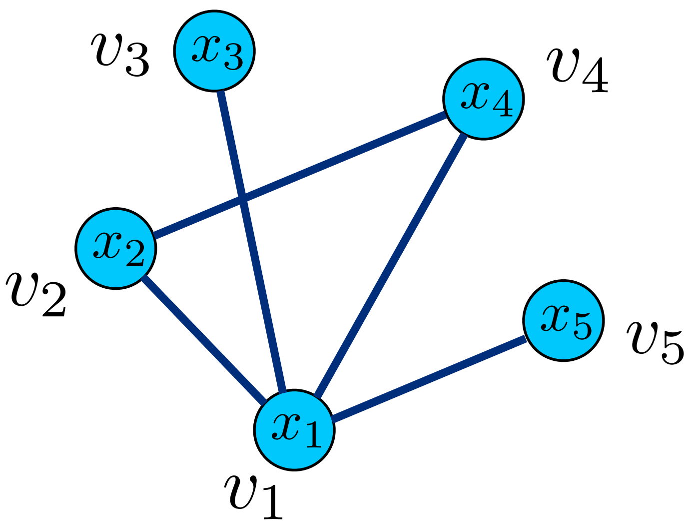
conditional independence:

$$(v_i, v_j) \notin \mathcal{E} \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$



A (very partial) literature overview

- Learning pairwise MRF



conditional independence:

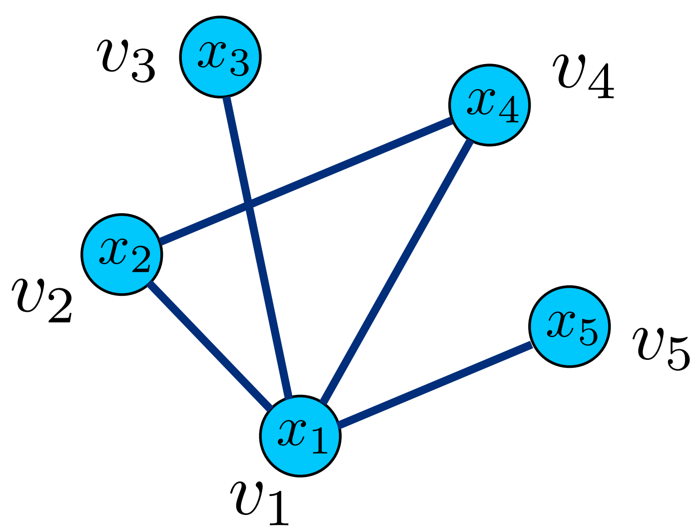
$$(v_i, v_j) \notin \mathcal{E} \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

probability parameterised by $\mathbf{\Theta}$:

$$P(\mathbf{x}|\mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\left(\sum_{v_i \in \mathcal{V}} \theta_{i,i} x_i^2 + \sum_{(v_i, v_j) \in \mathcal{E}} \theta_{i,j} x_i x_j\right)$$

A (very partial) literature overview

- Learning pairwise MRF



conditional independence:

$$(v_i, v_j) \notin \mathcal{E} \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

probability parameterised by $\boldsymbol{\Theta}$:

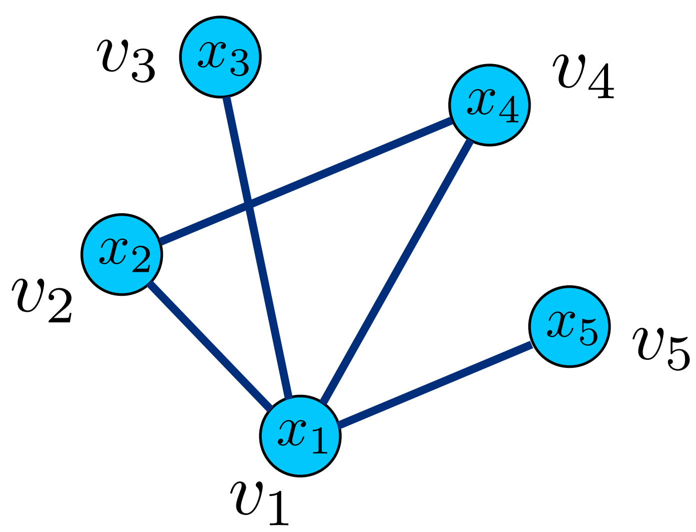
$$P(\mathbf{x}|\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} \exp\Big(\sum_{v_i \in \mathcal{V}} \theta_{i,i} x_i^2 + \sum_{(v_i,v_j) \in \mathcal{E}} \theta_{i,j} x_i x_j\Big)$$

Gaussian MRF with precision $\boldsymbol{\Theta}$:

$$P(\mathbf{x}|\boldsymbol{\Theta}) = \frac{|\boldsymbol{\Theta}|^{1/2}}{(2\pi)^{N/2}} \exp\Big(-\frac{1}{2}\mathbf{x}^T \boldsymbol{\Theta} \mathbf{x}\Big)$$

A (very partial) literature overview

- Learning pairwise MRF



conditional independence:

$$(v_i, v_j) \notin \mathcal{E} \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

probability parameterised by $\mathbf{\Theta}$:

$$P(\mathbf{x}|\mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\Big(\sum_{v_i \in \mathcal{V}} \theta_{i,i} x_i^2 + \sum_{(v_i, v_j) \in \mathcal{E}} \theta_{i,j} x_i x_j\Big)$$

Gaussian MRF with precision $\mathbf{\Theta}$:

$$P(\mathbf{x}|\mathbf{\Theta}) = \frac{|\mathbf{\Theta}|^{1/2}}{(2\pi)^{N/2}} \exp\Big(-\frac{1}{2}\mathbf{x}^T \mathbf{\Theta} \mathbf{x}\Big)$$

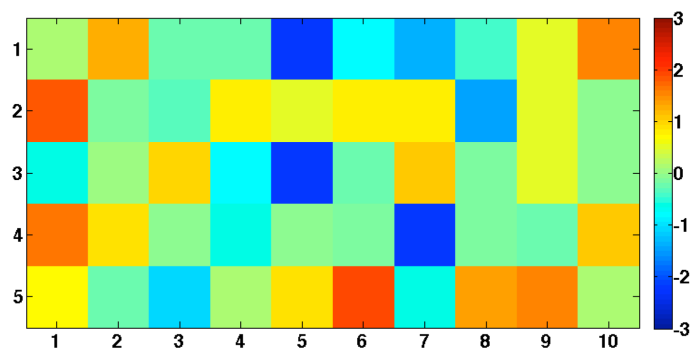
learning a sparse $\mathbf{\Theta}$:

- **interactions are mostly local**
- **feasible in high-dimensional space**

A (very partial) literature overview

covariance selection

Dempster



$$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$$

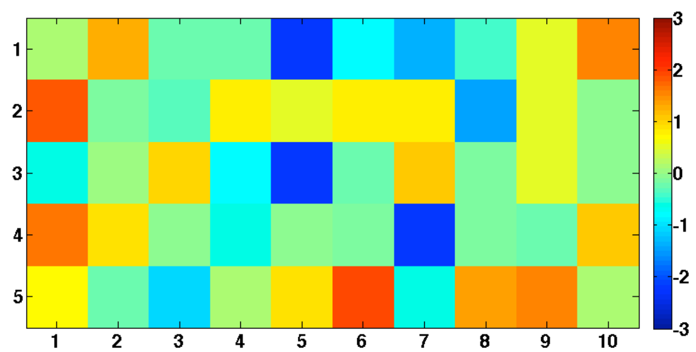
data matrix

A (very partial) literature overview

covariance selection

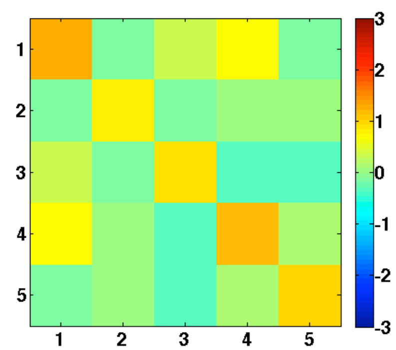
Dempster

choosing covariance that agrees with **S** in set J (precision is zero in complementary set I)



$$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$$

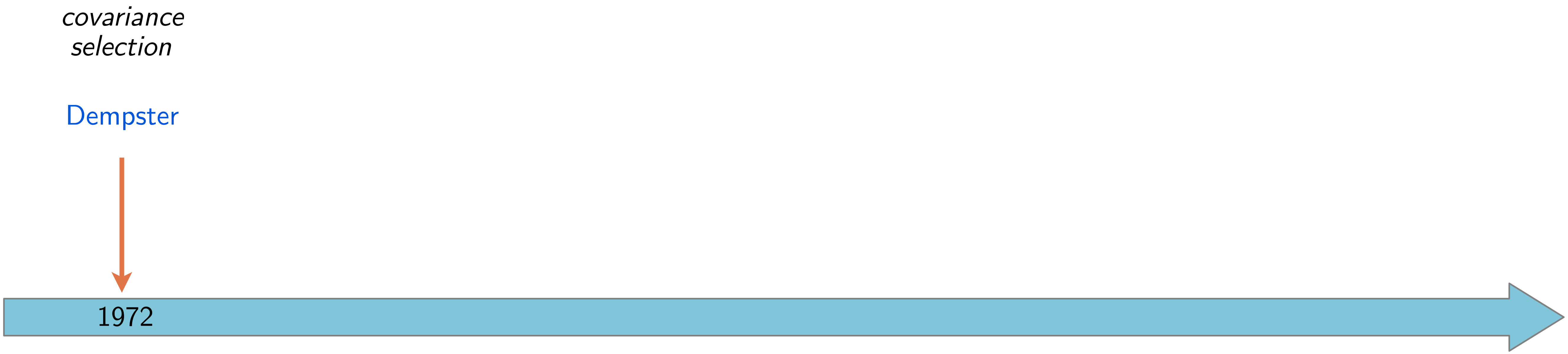
data matrix



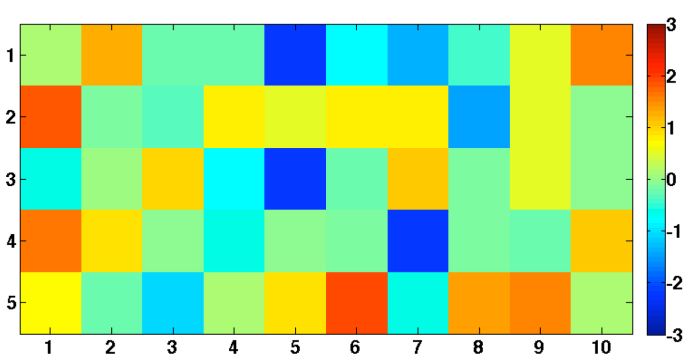
$$\mathbf{S}$$

sample covariance

A (very partial) literature overview

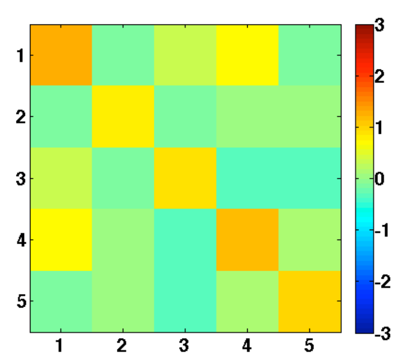


sequentially pruning elements in set I in sample precision



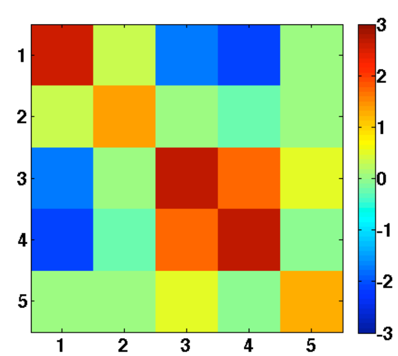
$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Theta})$

data matrix



$\mathbf{S}$

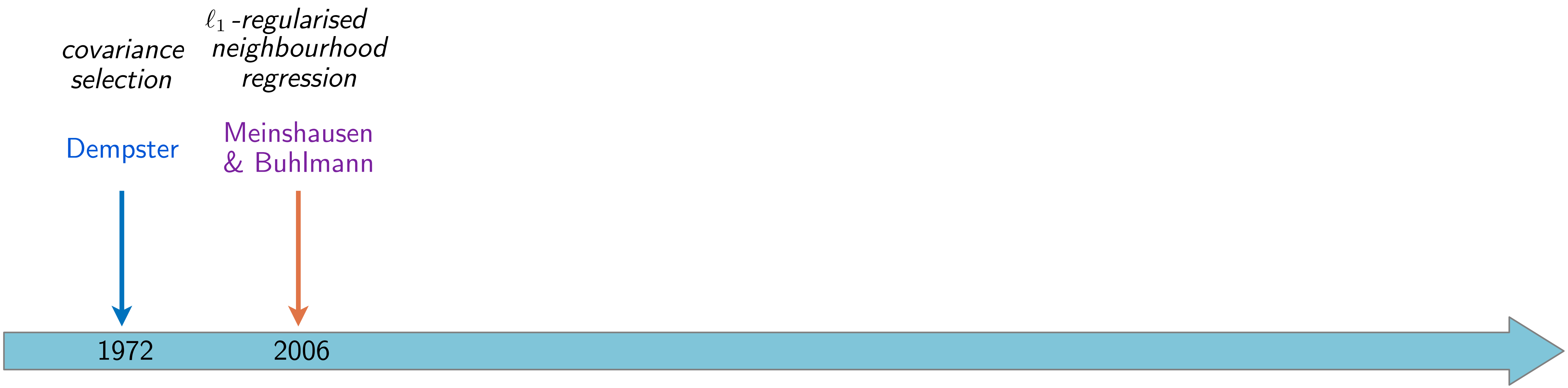
sample covariance



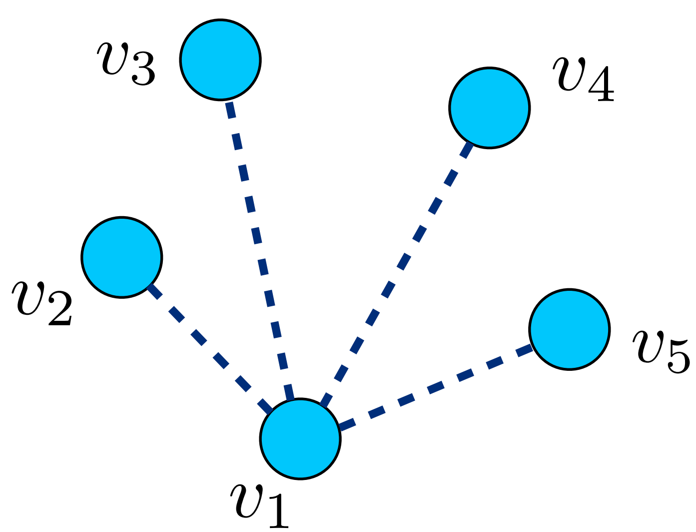
$\mathbf{S}^{-1}$

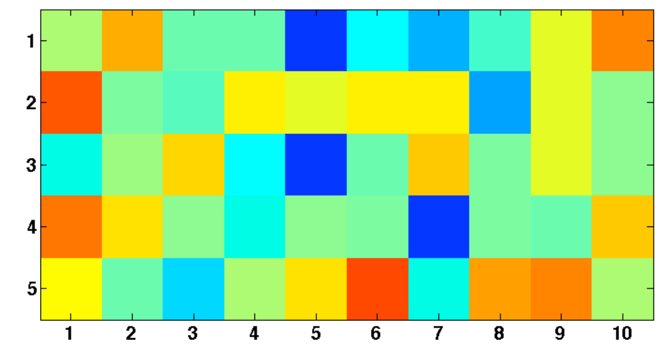
sample precision

A (very partial) literature overview

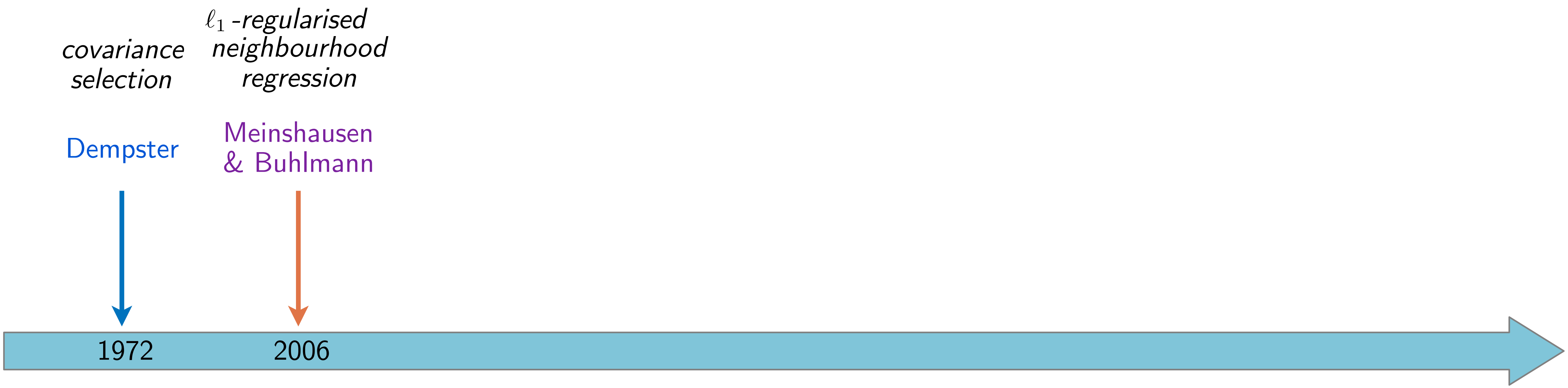


neighbourhood selection: learning neighbourhood of each node

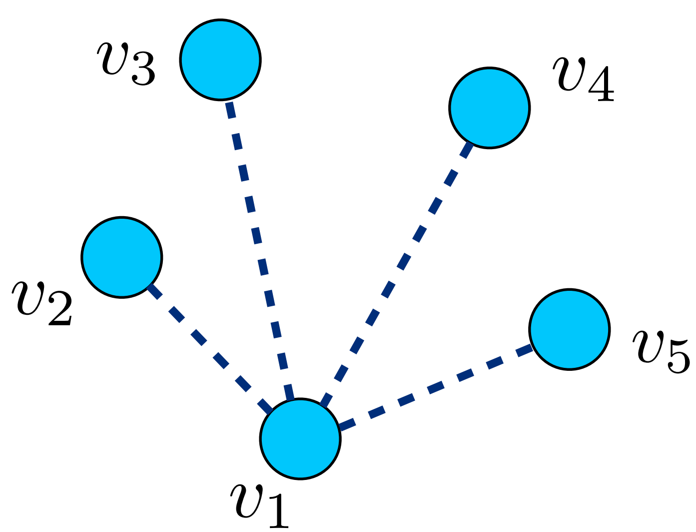


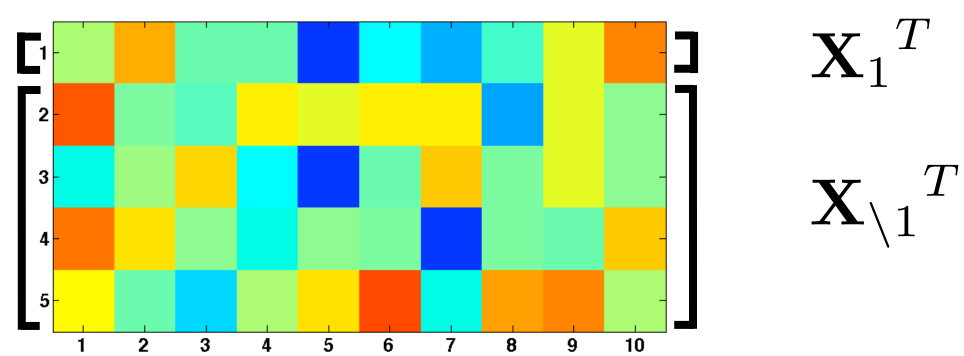


A (very partial) literature overview

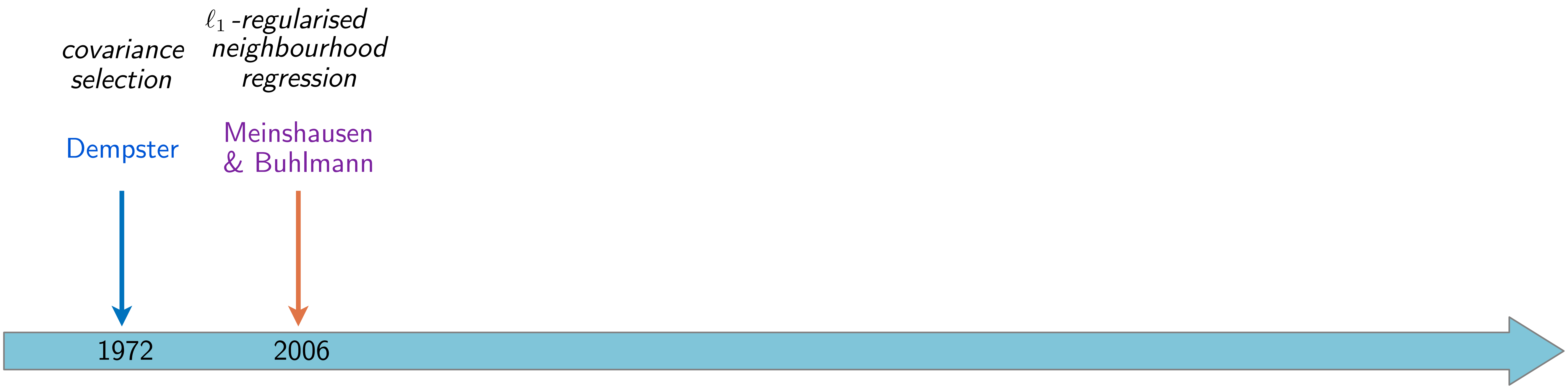


neighbourhood selection: learning neighbourhood of each node

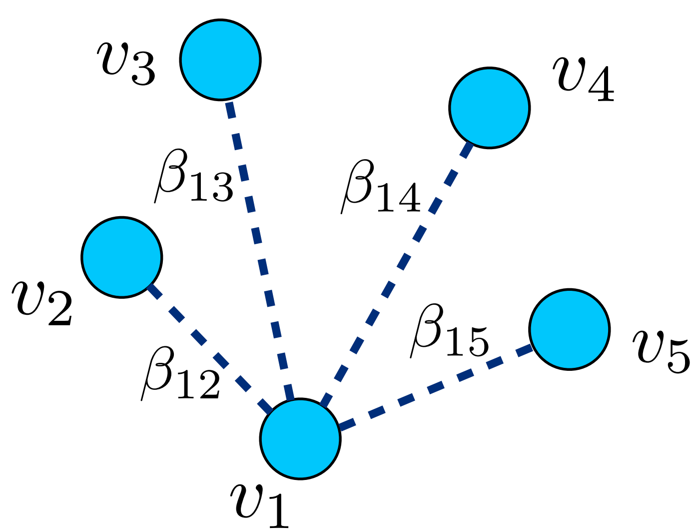


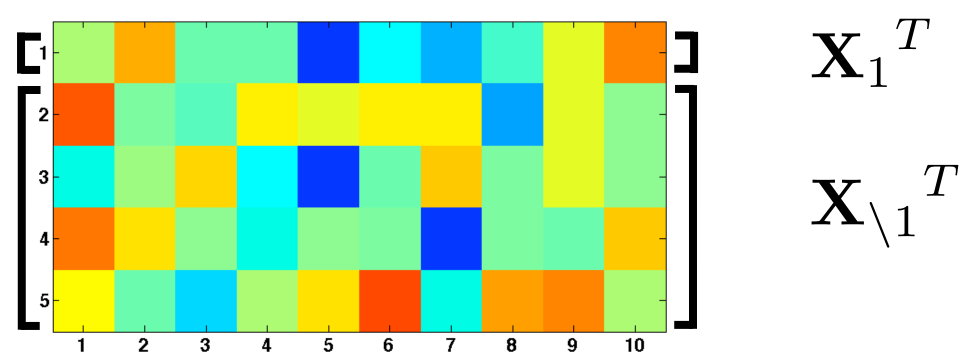


A (very partial) literature overview



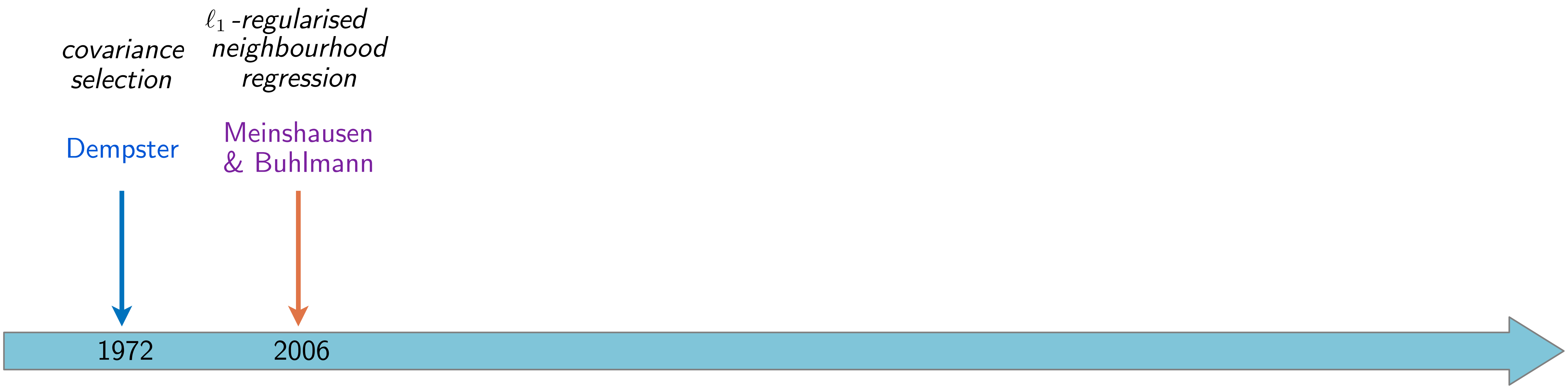
neighbourhood selection: learning neighbourhood of each node



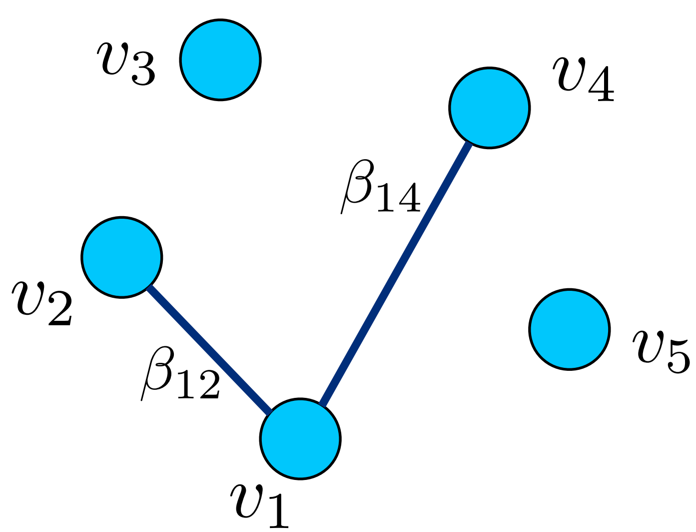


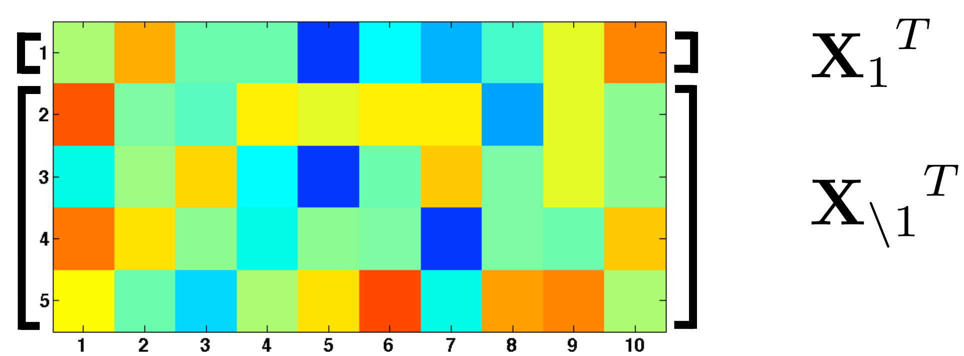
Lasso regression: $\min_{\boldsymbol{\beta}_1} ||\mathbf{X}_1 - \mathbf{X}_{\setminus 1}\boldsymbol{\beta}_1||^2 + \lambda||\boldsymbol{\beta}_1||_1$

A (very partial) literature overview



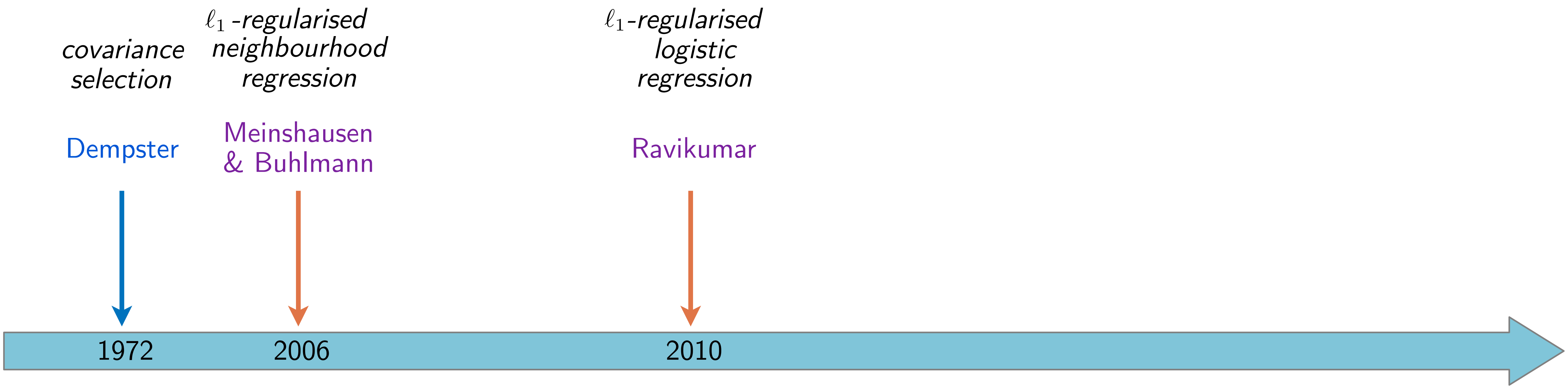
neighbourhood selection: learning neighbourhood of each node



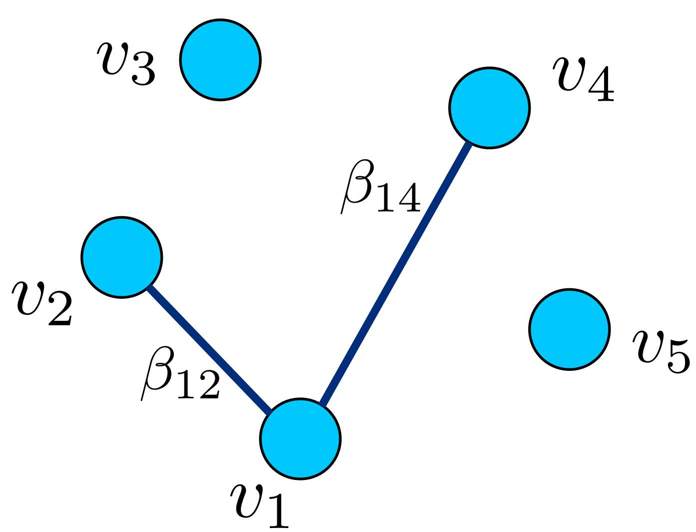


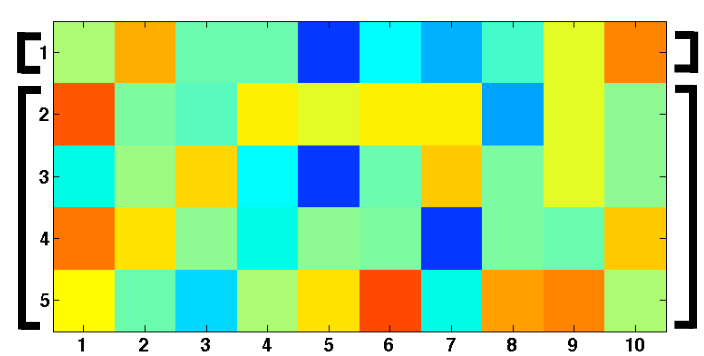
Lasso regression: $\min_{\boldsymbol{\beta}_1} ||\mathbf{X}_1 - \mathbf{X}_{\backslash 1}\boldsymbol{\beta}_1||^2 + \lambda||\boldsymbol{\beta}_1||_1$

A (very partial) literature overview



neighbourhood selection: learning neighbourhood of each node





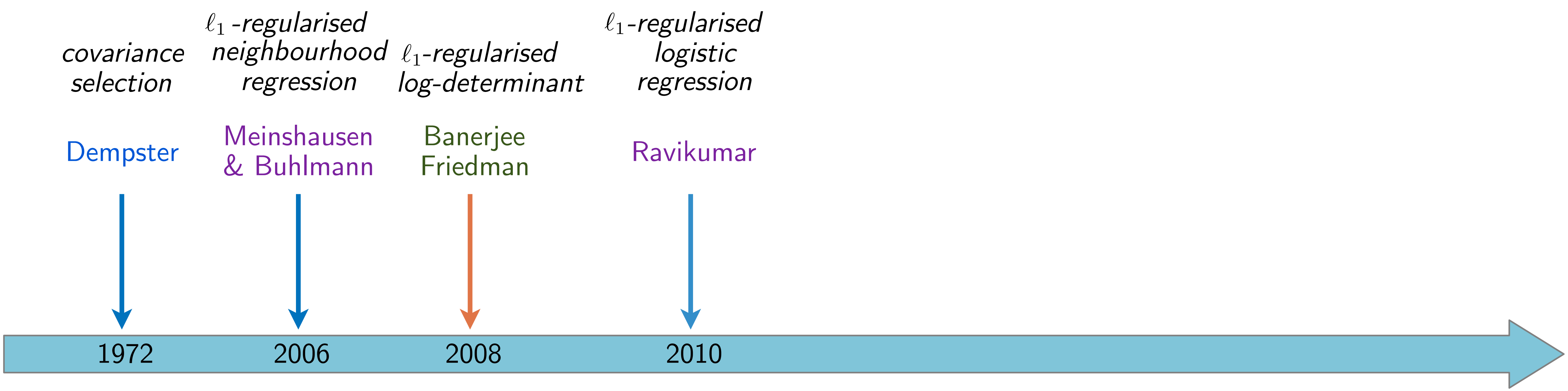
$\mathbf{X}_1{}^T$

$\mathbf{X}_{\setminus 1}{}^T$

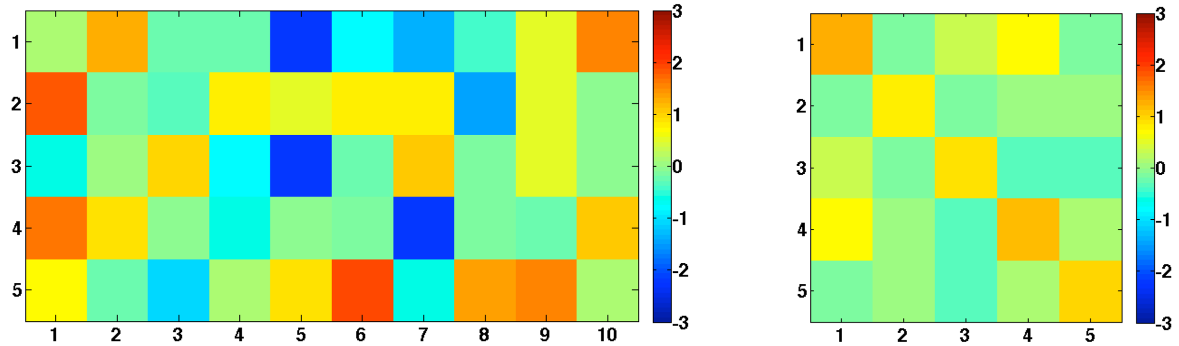
Lasso regression: $\min_{\boldsymbol{\beta}_1} ||\mathbf{X}_1 - \mathbf{X}_{\setminus 1}\boldsymbol{\beta}_1||^2 + \lambda||\boldsymbol{\beta}_1||_1$

logistic regression for discrete variables

A (very partial) literature overview

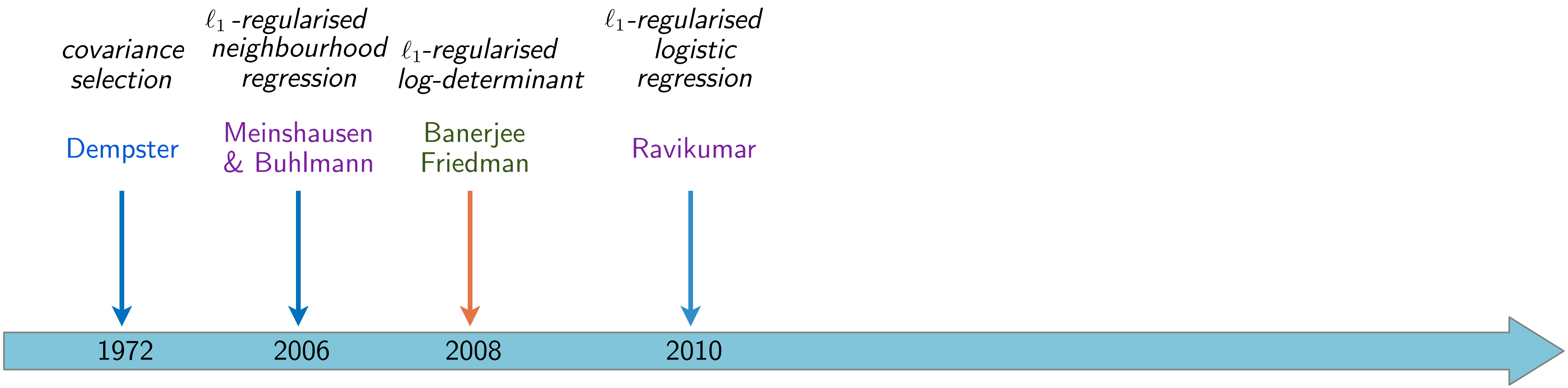


graphical Lasso: estimation of sparse precision matrix

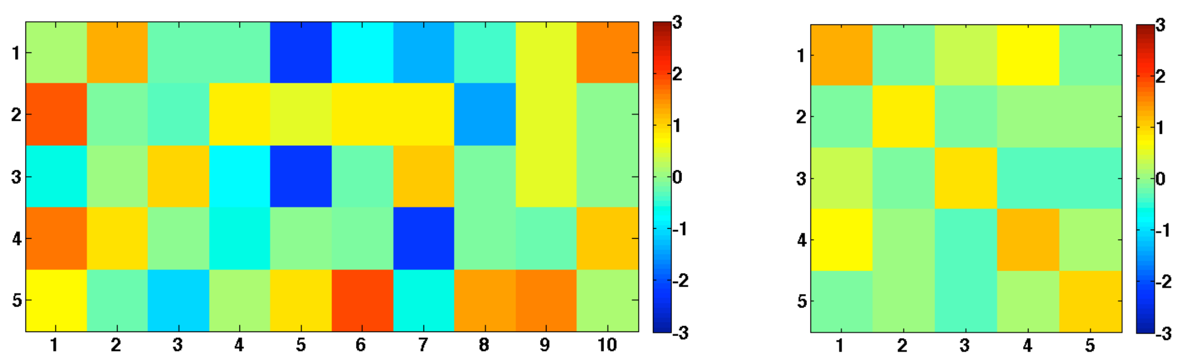


$\mathbf{X}$ $\mathbf{S}$

A (very partial) literature overview



graphical Lasso: estimation of sparse precision matrix



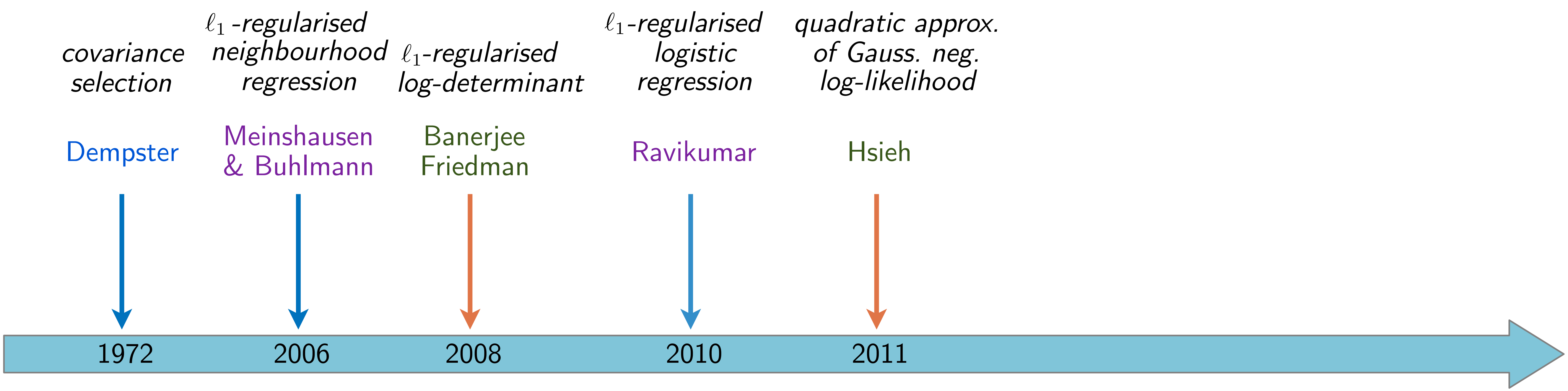
$\mathbf{X}$ $\mathbf{S}$

graphical Lasso maximises likelihood of precision matrix $\mathbf{\Theta}$:

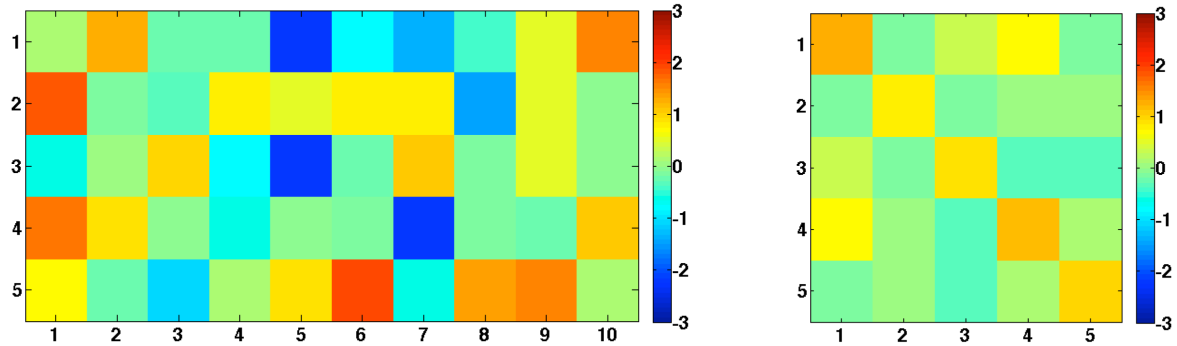
$$\max_{\mathbf{\Theta}} \underline{\log \det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S}\mathbf{\Theta})} - \rho ||\mathbf{\Theta}||_1$$

log-likelihood function

A (very partial) literature overview



graphical Lasso: estimation of sparse precision matrix



$\mathbf{X}$ $\mathbf{S}$

graphical Lasso maximises likelihood of precision matrix $\mathbf{\Theta}$:

$$\max_{\mathbf{\Theta}} \underbrace{\log \det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S}\mathbf{\Theta})}_{\text{log-likelihood function}} - \rho ||\mathbf{\Theta}||_1$$

A (very partial) literature overview

- Learning graphical models
  - classical approaches lead to both positive/negative relations
  - learning graphs with non-negative weights?

A (very partial) literature overview

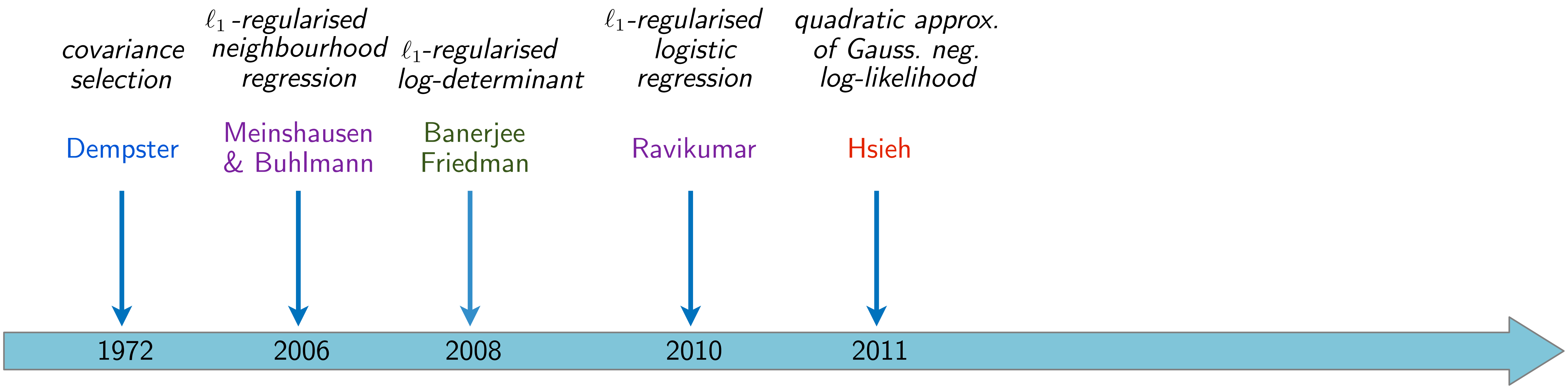
- Learning graphical models
  - classical approaches lead to both positive/negative relations
  - learning graphs with non-negative weights?
- Learning graphs with non-negative weights
  - M-matrices (symmetric, positive definite, non-positive off-diagonals) have been used as precision, leading to attractive GMRF [Slawski2015]

A (very partial) literature overview

- Learning graphical models
  - classical approaches lead to both positive/negative relations
  - learning graphs with non-negative weights?

- Learning graphs with non-negative weights
  - M-matrices (symmetric, positive definite, non-positive off-diagonals) have been used as precision, leading to attractive GMRF [Slawski2015]
  - combinatorial **graph Laplacian L** belongs to M-matrices and is equivalent to graph topology

A (very partial) literature overview

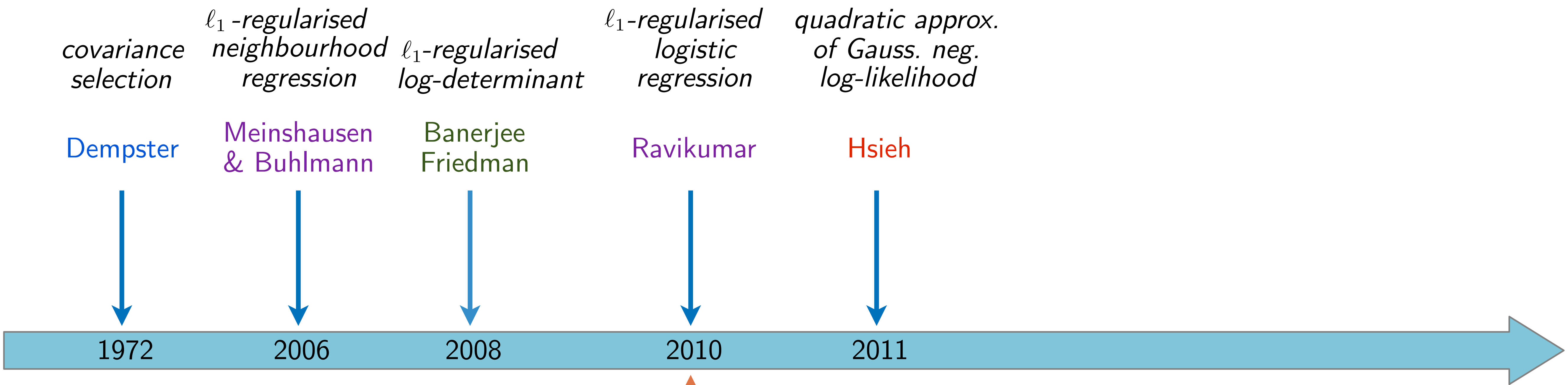


$$\max_{\boldsymbol{\Theta}} \log \det \boldsymbol{\Theta} - \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta}) - \rho ||\boldsymbol{\Theta}||_1$$

graph Laplacian **L** can be the precision,
BUT it is singular

from arbitrary precision matrix to graph Laplacian

A (very partial) literature overview



$$\max_{\boldsymbol{\Theta}} \log \det \boldsymbol{\Theta} - \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta}) - \rho ||\boldsymbol{\Theta}||_1$$

$$\text{s.t. } \boldsymbol{\Theta} = \mathbf{L} + \frac{1}{\sigma^2}\mathbf{I}$$

graph Laplacian **L** can be the precision,
BUT it is singular

from arbitrary precision matrix to graph Laplacian

A (very partial) literature overview

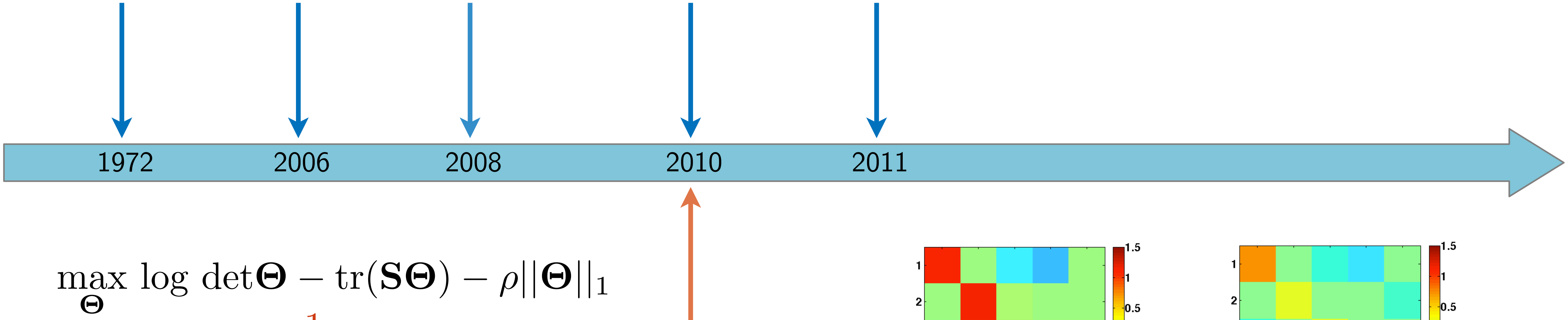
covariance selection — Dempster

ℓ_1 -regularised neighbourhood regression — Meinshausen & Buhlmann

ℓ_1-regularised log-determinant — Banerjee Friedman

ℓ_1-regularised logistic regression — Ravikumar

quadratic approx. of Gauss. neg. log-likelihood — Hsieh



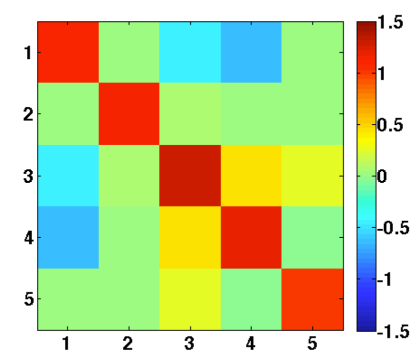
$$\max_{\mathbf{\Theta}} \log \det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S}\mathbf{\Theta}) - \rho ||\mathbf{\Theta}||_1$$

$$\text{s.t. } \mathbf{\Theta} = \mathbf{L} + \frac{1}{\sigma^2}\mathbf{I}$$

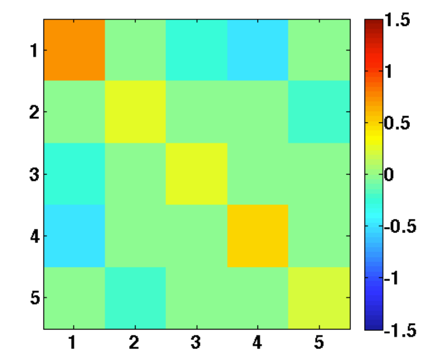
graph Laplacian **L** can be the precision, BUT it is singular

Lake

log-determinant program on regularised ***L***



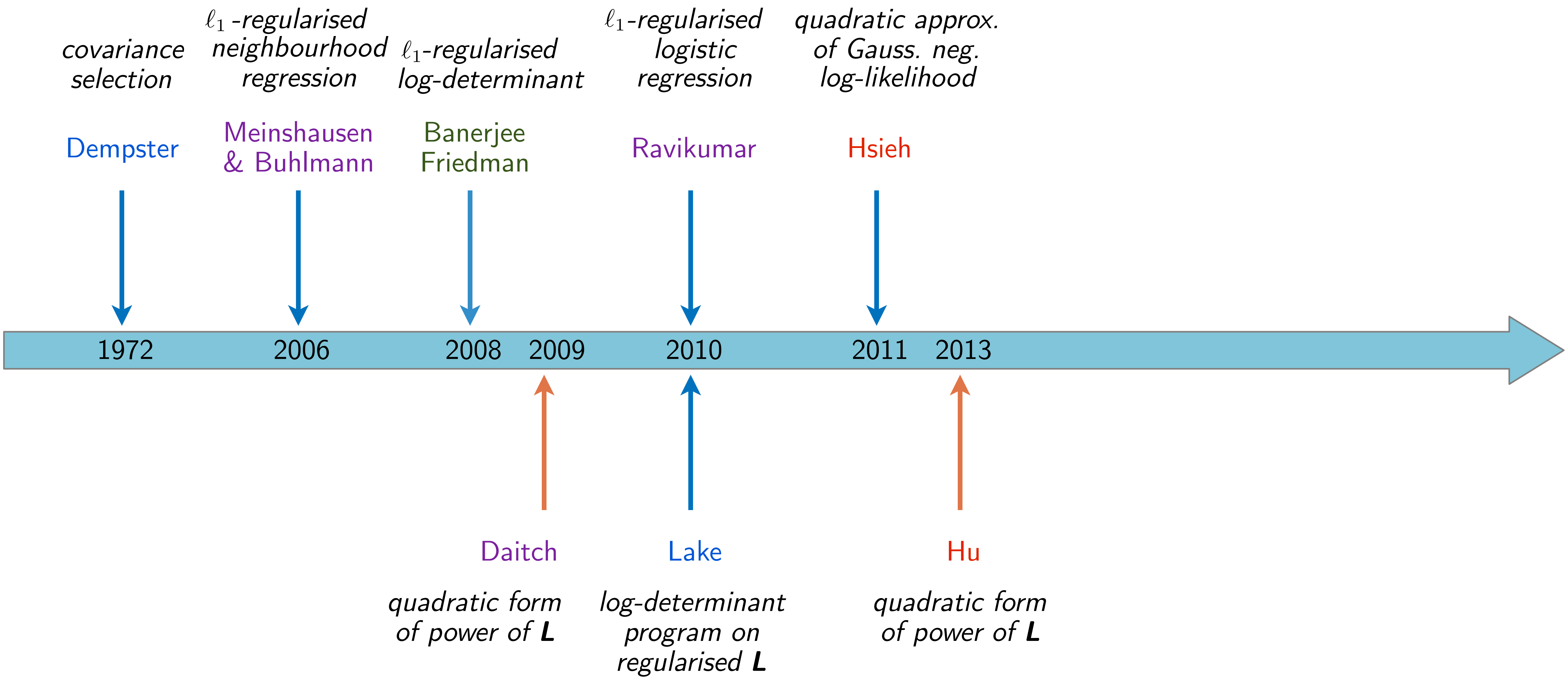
precision by graphical Lasso



Laplacian by Lake et al.

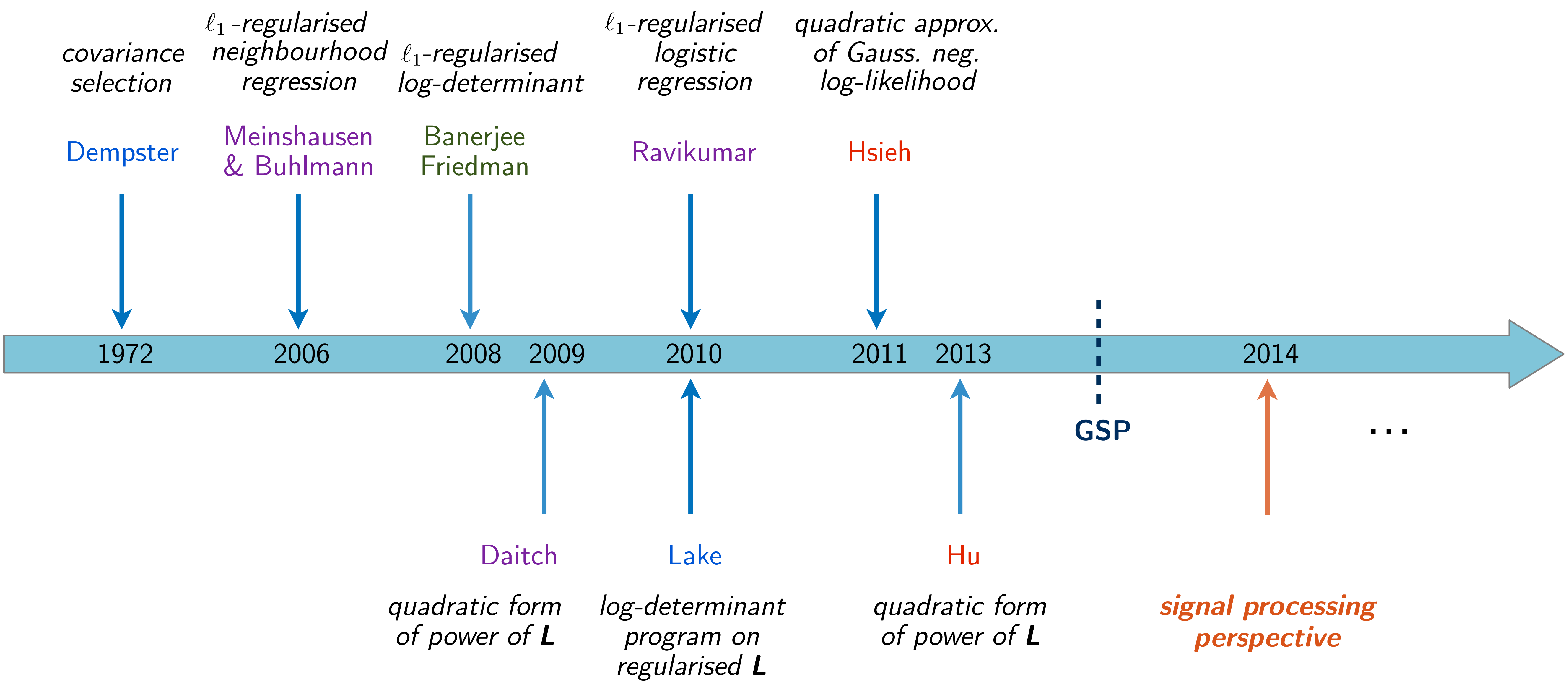
from arbitrary precision matrix to graph Laplacian

A (very partial) literature overview



from arbitrary precision matrix to graph Laplacian

A (very partial) literature overview



from arbitrary precision matrix to graph Laplacian

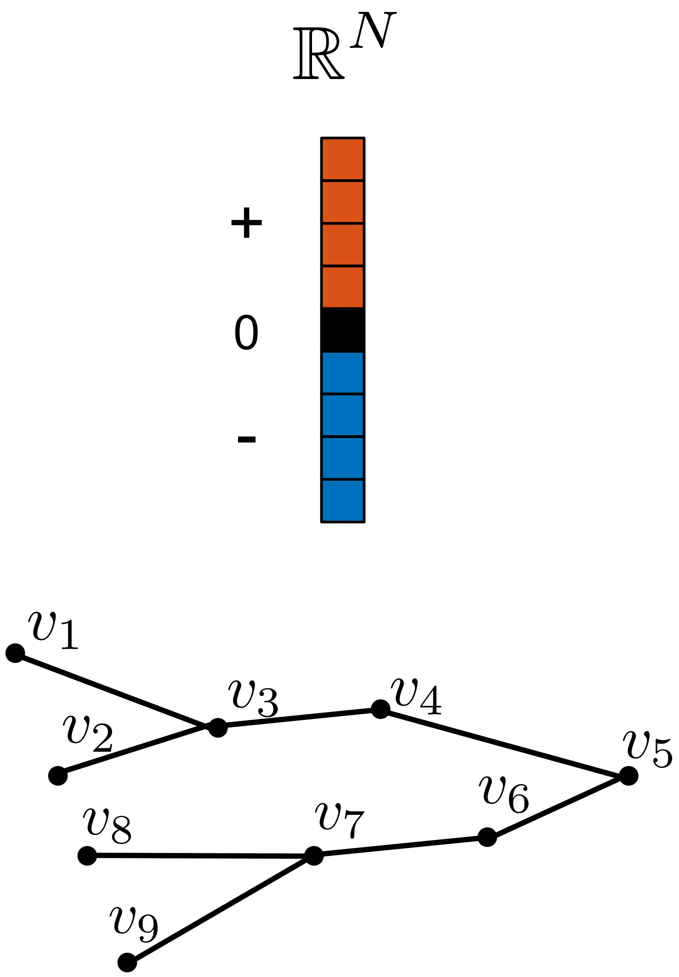
common setting in graph signal processing (GSP)

Outline

- A (very partial) literature overview
- A signal processing perspective
  - A brief introduction to graph signal processing (GSP)
  - GSP approaches for graph learning
- Concluding remarks

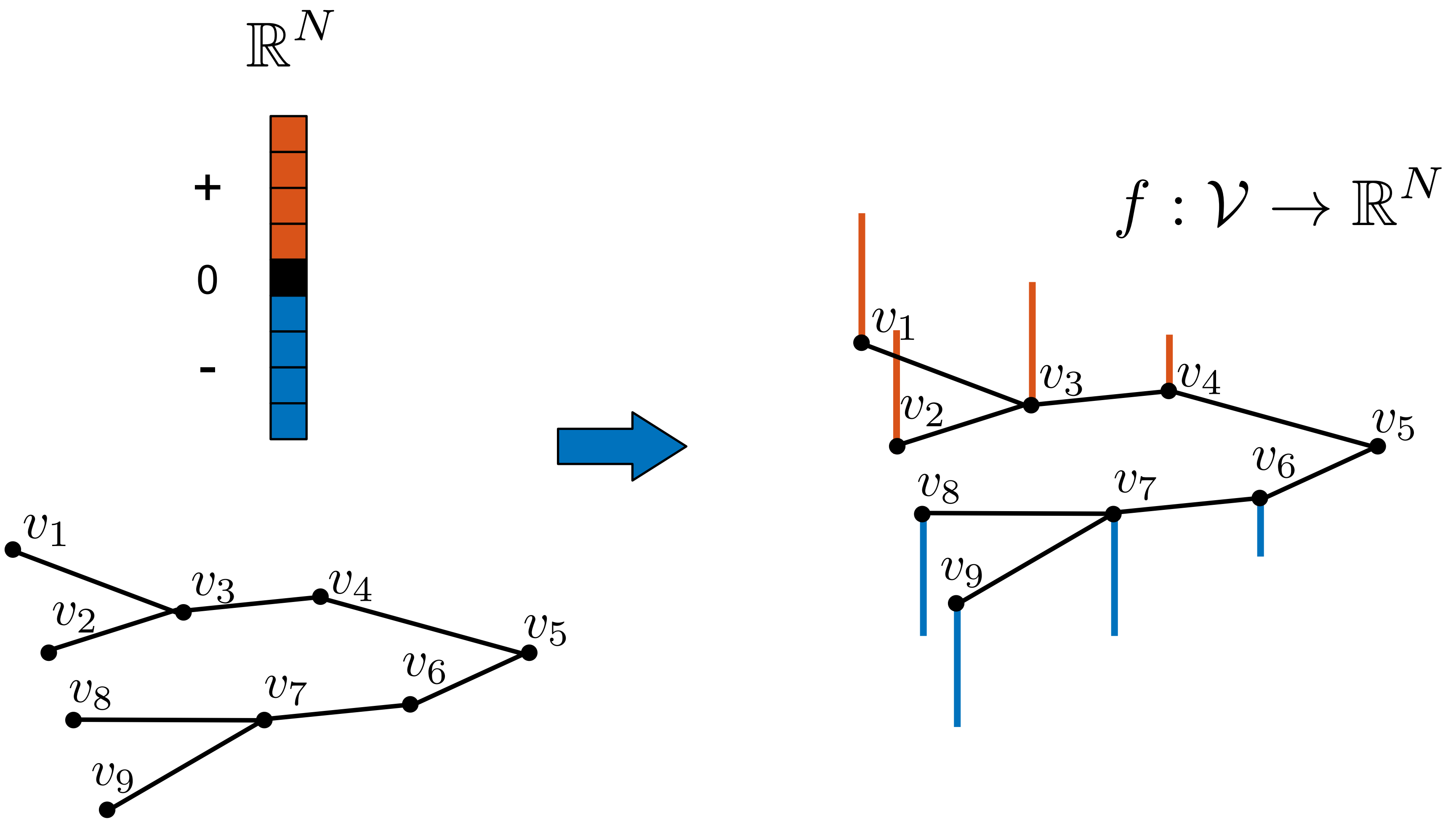
Graph signals

- Structured data can be represented by graph signals



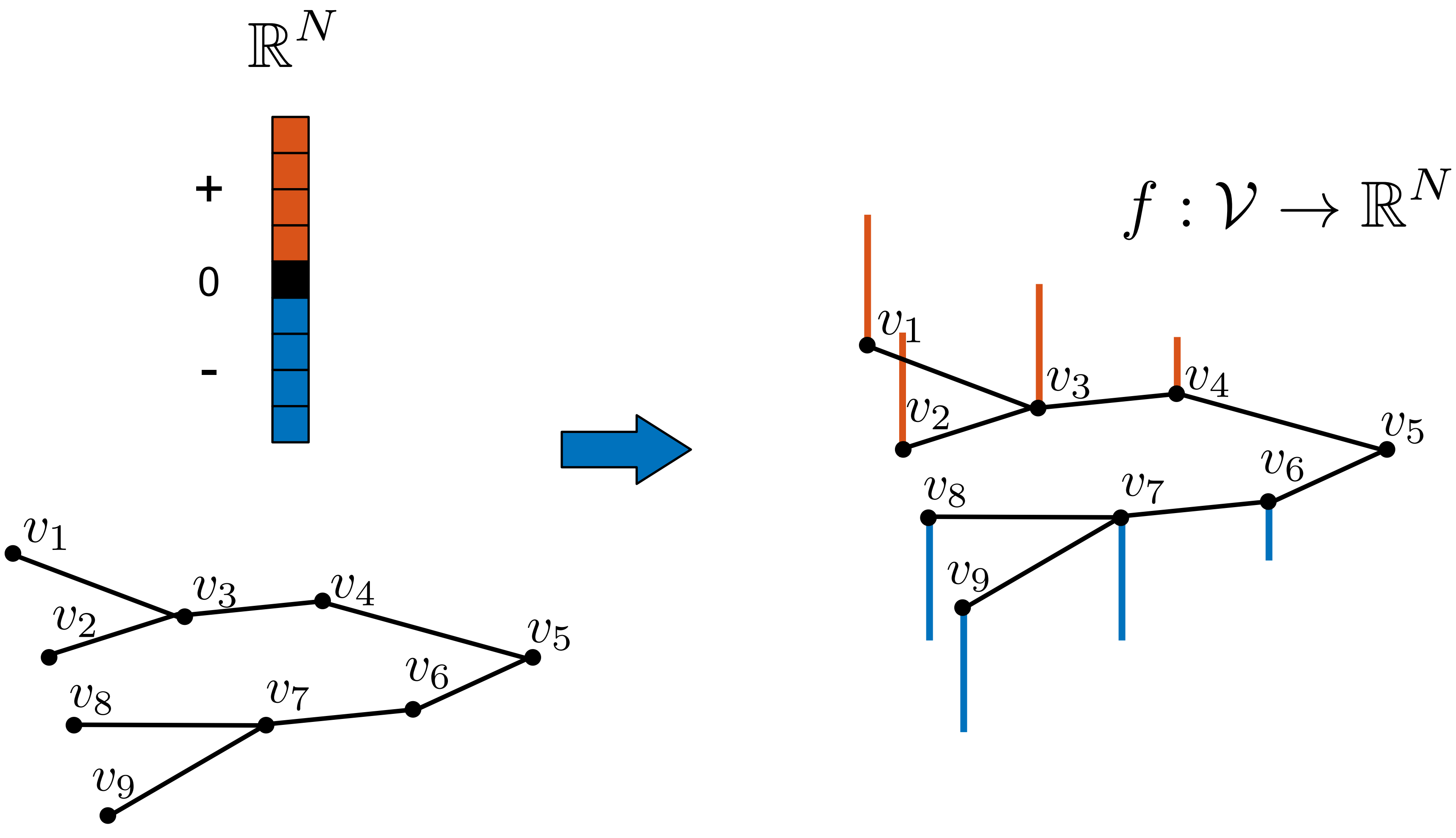
Graph signals

- Structured data can be represented by graph signals



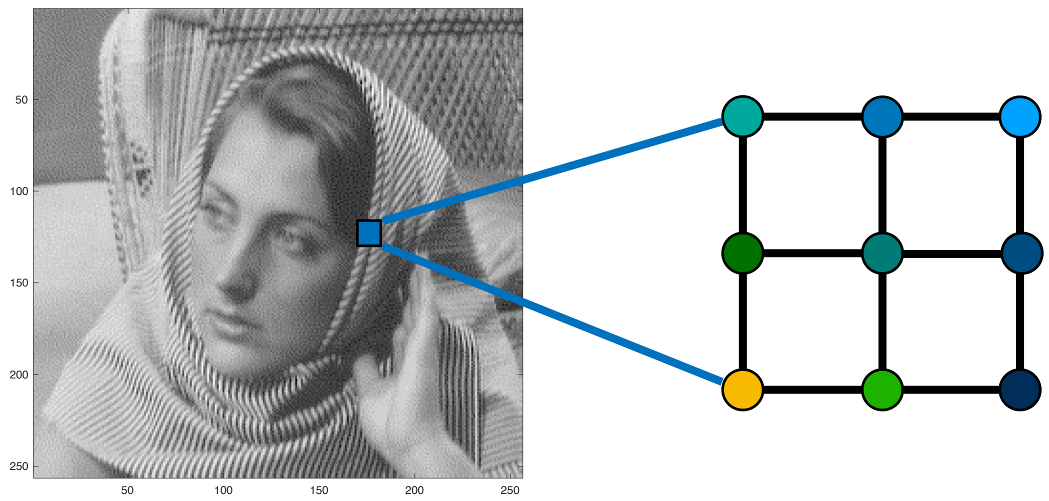
Graph signals

- Structured data can be represented by graph signals



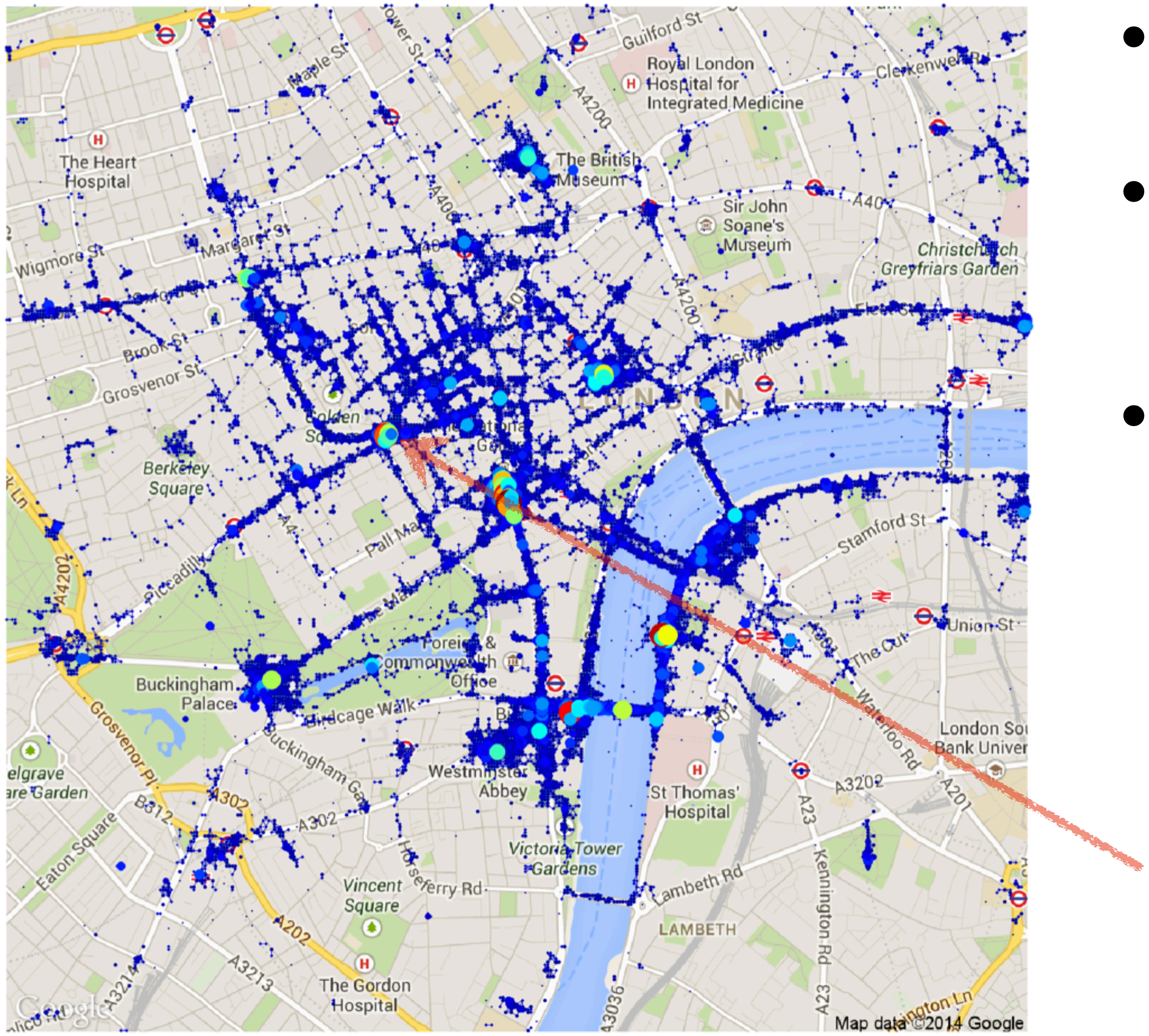
takes into account both structure (edges) and data (values at vertices)

Graph signals are pervasive

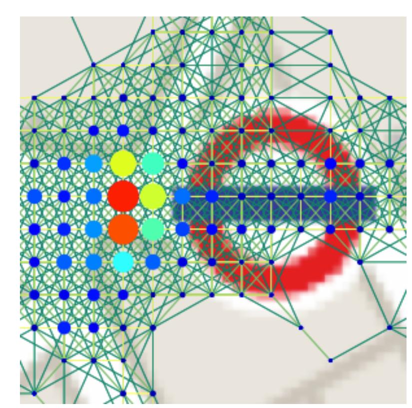


- Vertices:
  - regular grid
- Edges:
  - 4-nearest neighbour connection
- Signal:
  - pixel values

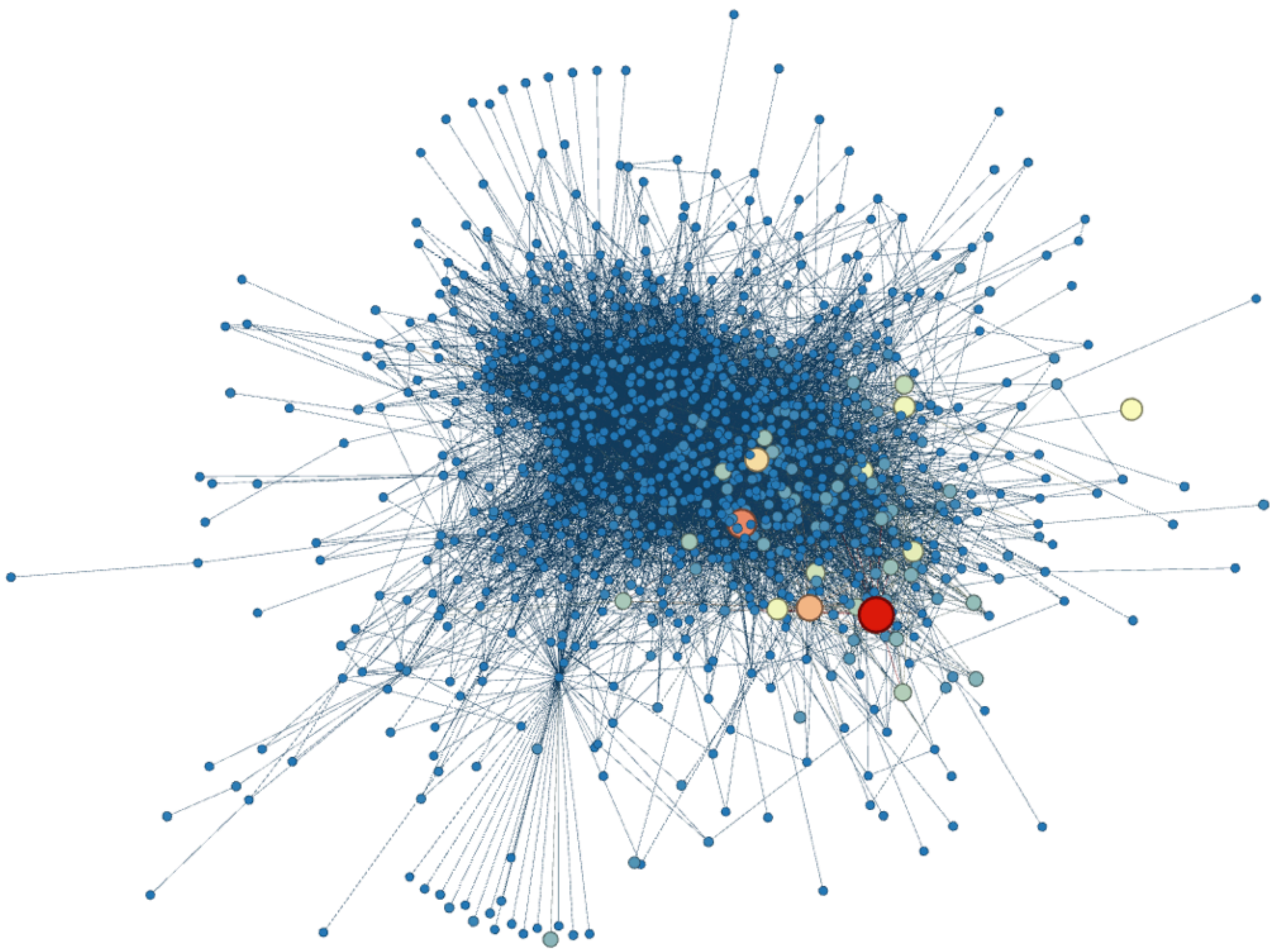
Graph signals are pervasive



- 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

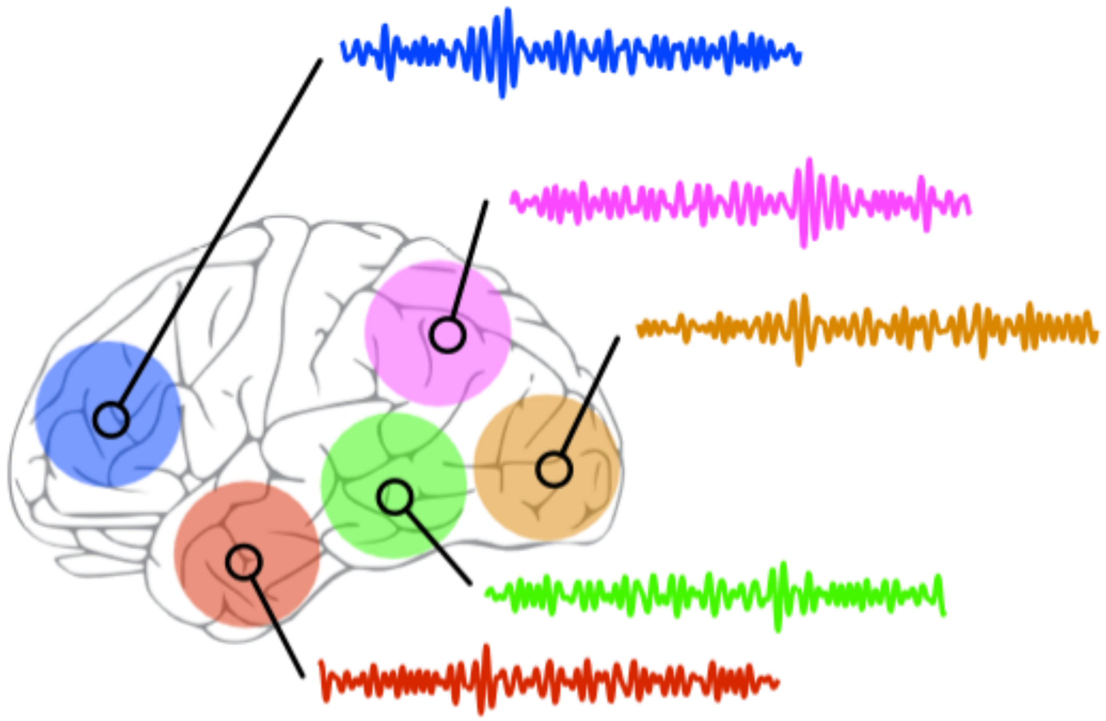


Graph signals are pervasive



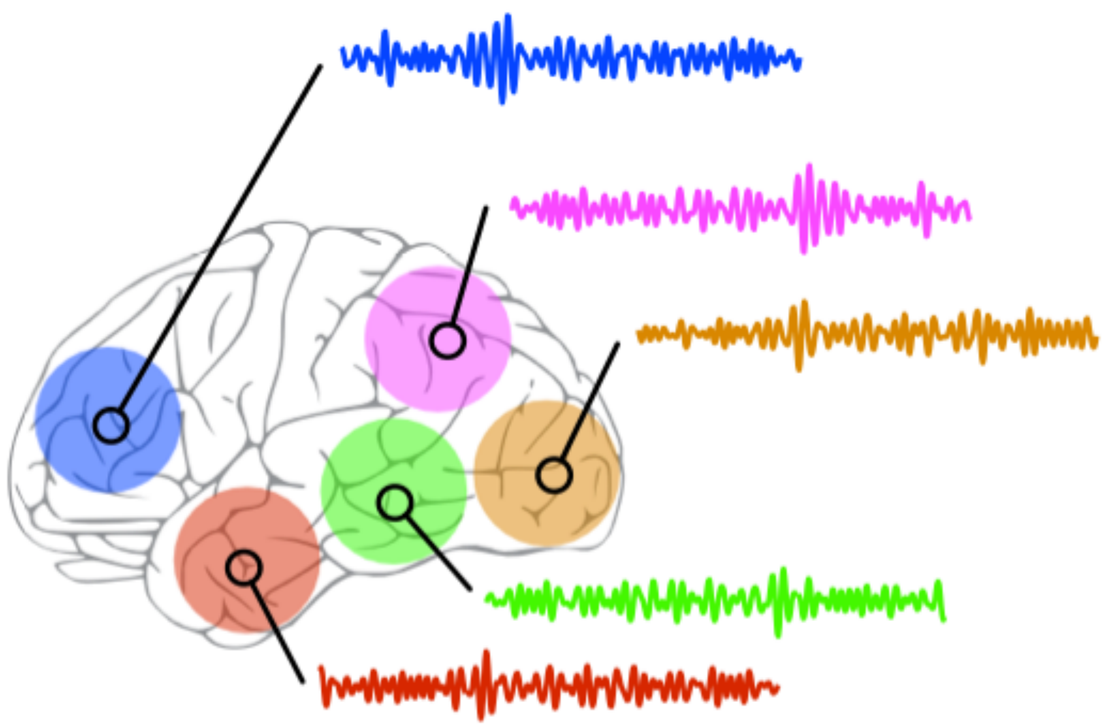
- Vertices:
  - 1000 Twitter users
- Edges:
  - following relationship among users
- Signal:
  - # Apple-related hashtags they have posted in six weeks

Graph signals are pervasive



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

Graph signals are pervasive

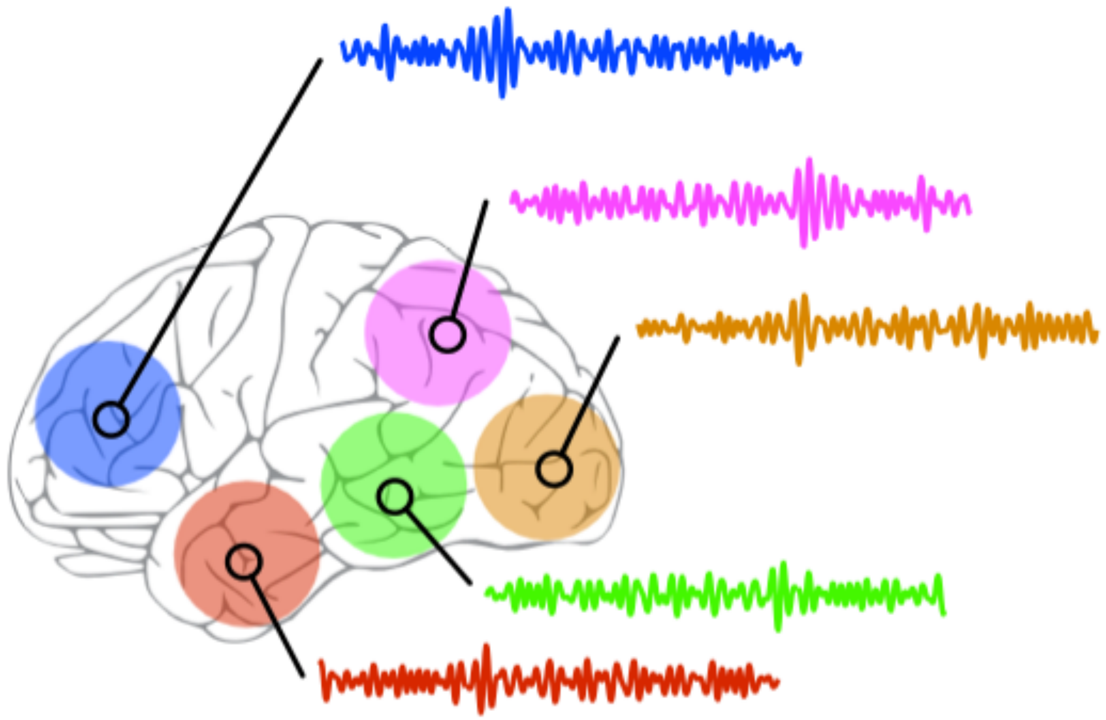


- Vertices:
  - brain regions
- Edges:
  - structural connectivity (via diffusion spectrum imaging) between brain regions
- Signal:
  - blood-oxygen-level-dependent (BOLD) time series

how to generalise signal processing tools on graphs?

- **notion of shift invariance?**
- **notion of frequency?**

Graph signals are pervasive

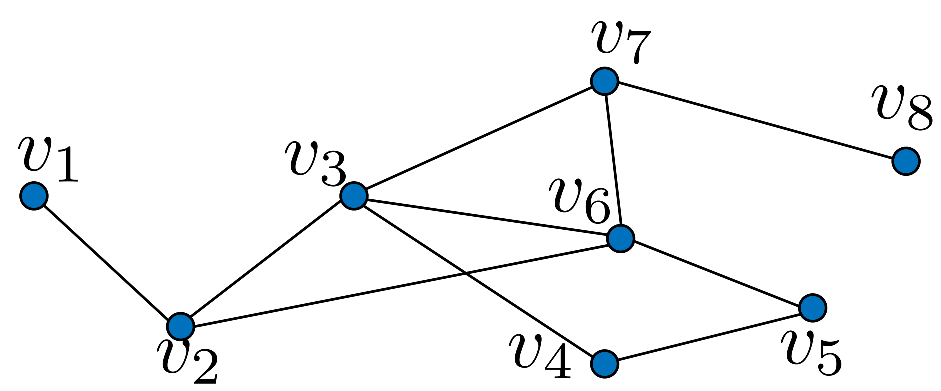


- Vertices:
  - brain regions
- Edges:
  - structural connectivity (via diffusion spectrum imaging) between brain regions
- Signal:
  - blood-oxygen-level-dependent (BOLD) time series

how to generalise signal processing tools on graphs?

- **notion of shift invariance? graph shift operator**
- **notion of frequency? graph Laplacian**

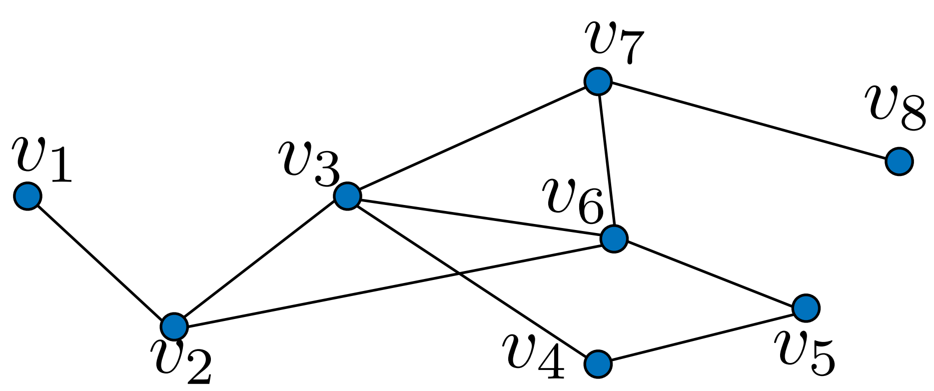
Graph Laplacian



Weighted and undirected graph:

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

Graph Laplacian



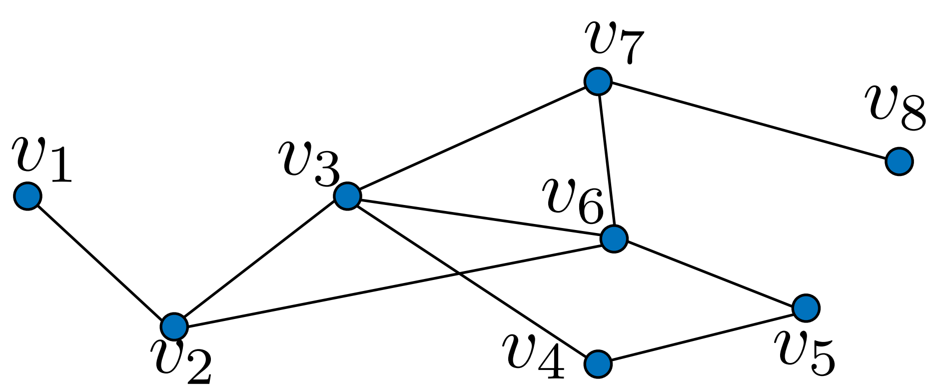
Weighted and undirected graph:

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

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

W

Graph Laplacian



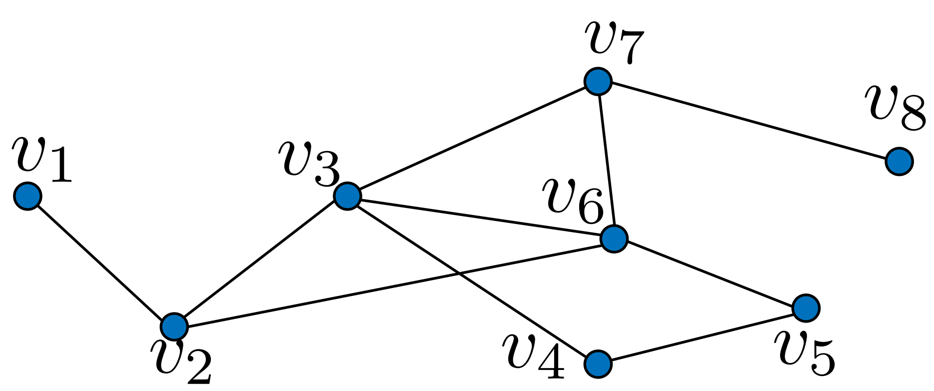
Weighted and undirected graph:

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

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

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

Graph Laplacian



Weighted and undirected graph:

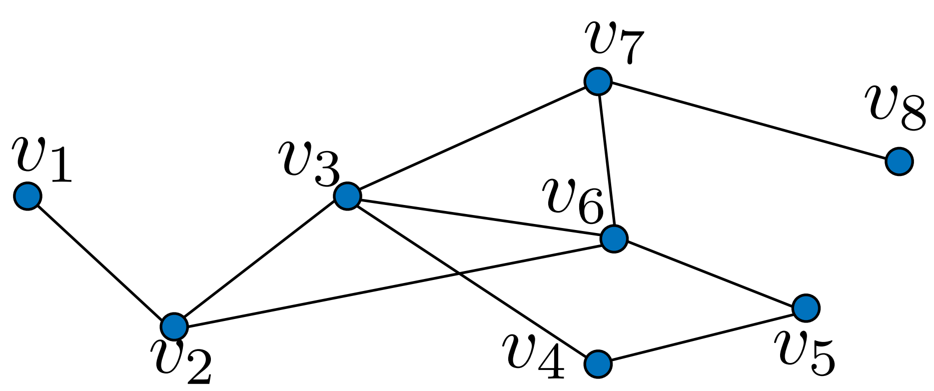
$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$

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

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

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

Graph Laplacian



Weighted and undirected graph:

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

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

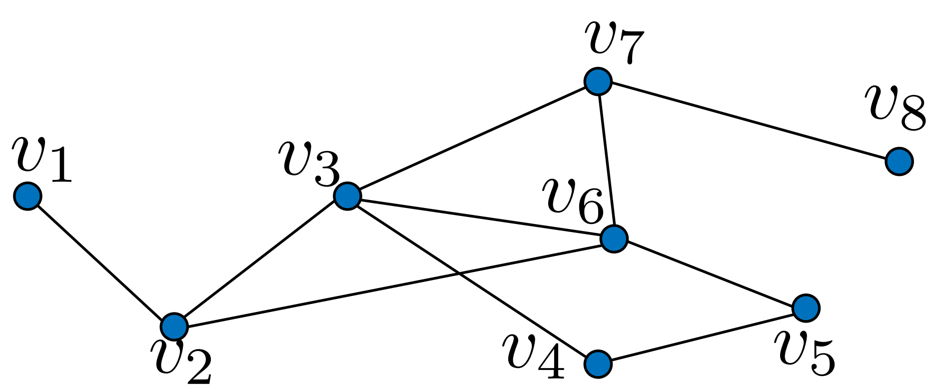
$$L = D - W$$ **Equivalent to G!**

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

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

D $\qquad$ W $\qquad$ L

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!**

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

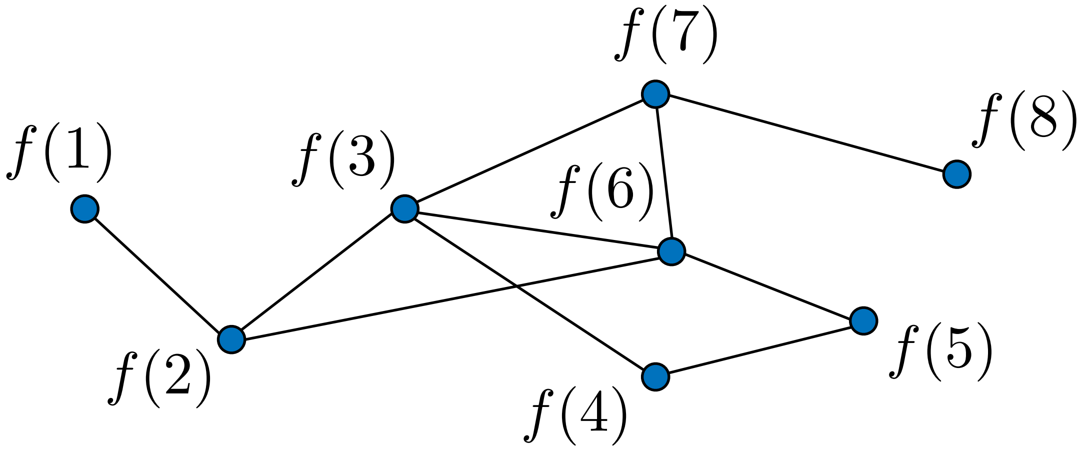
$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{D}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

why graph Laplacian?

- **standard stencil approximation of the Laplace operator**
- **provides a notion of "frequency" on graphs**

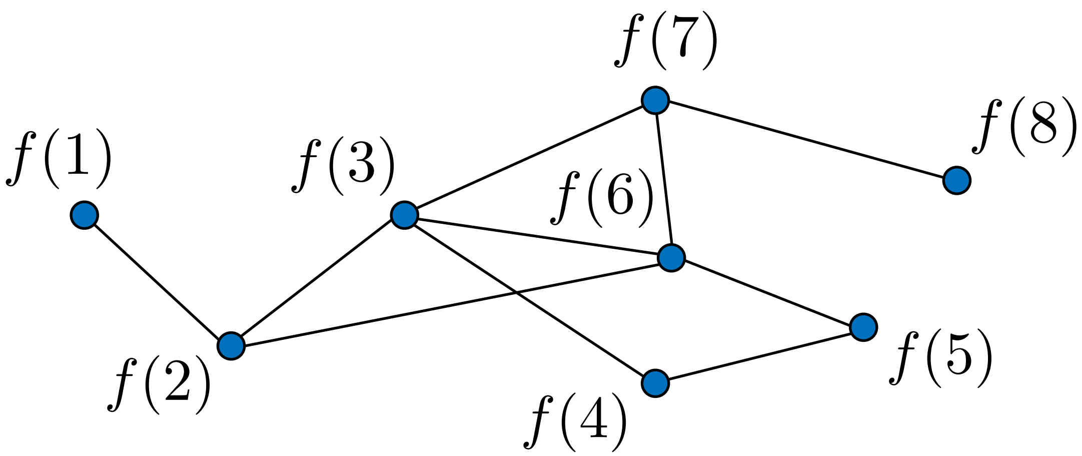
Graph Laplacian

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



Graph Laplacian

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

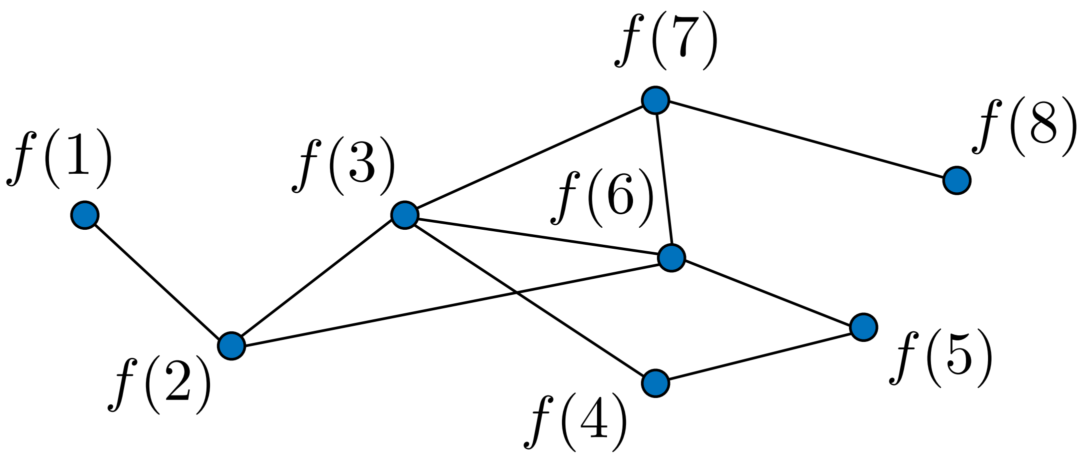


$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}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

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

Graph Laplacian

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



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

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

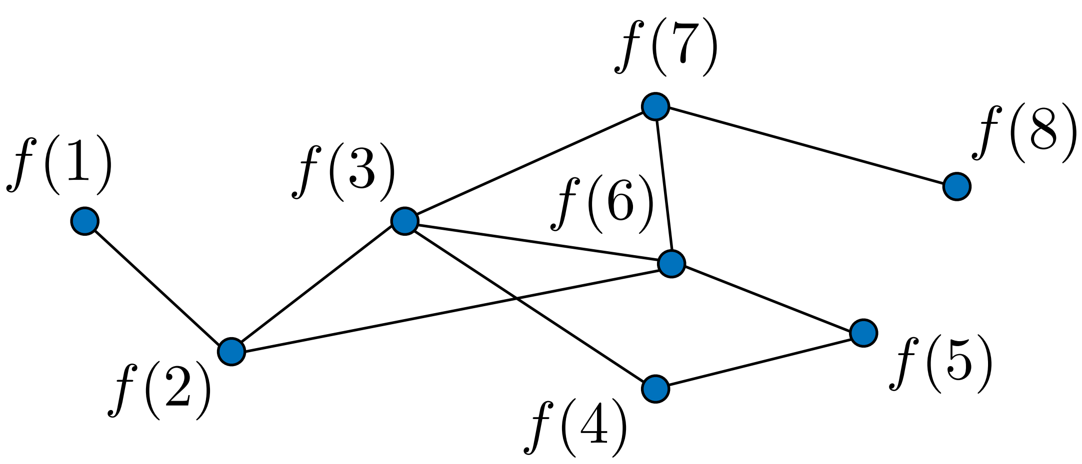
$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 &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

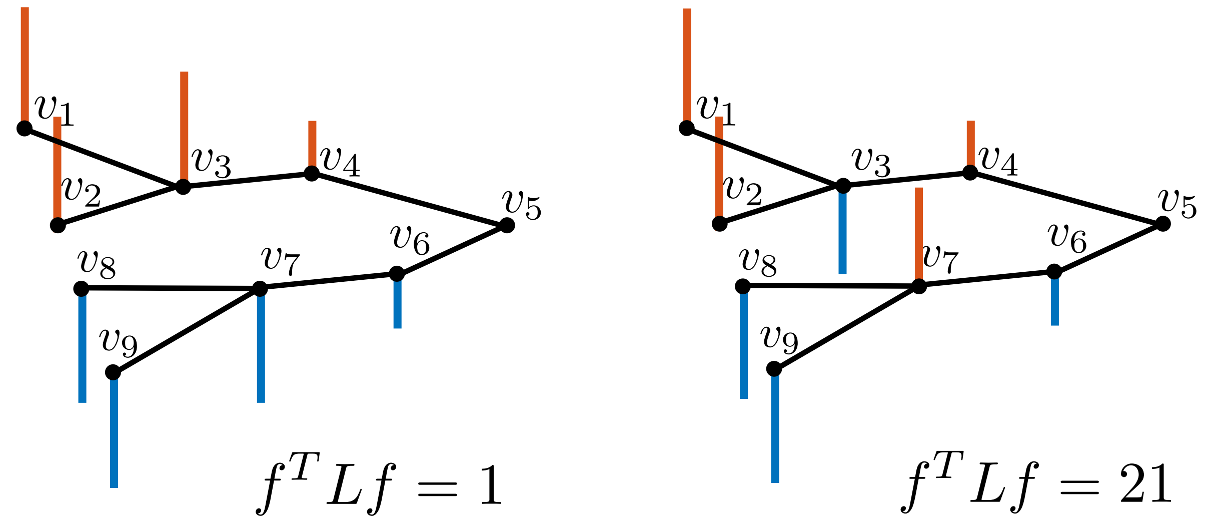
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left(f(i) - f(j) \right)^2$$

measure of "smoothness" [Zhou04]

Graph Laplacian

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





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

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

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

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

measure of "smoothness" [Zhou04]

Graph Laplacian

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

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

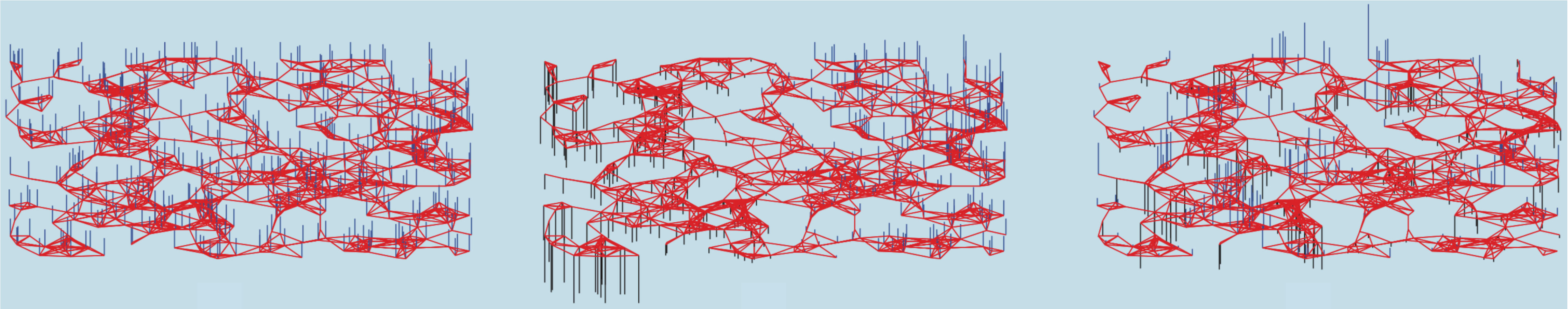
Graph Laplacian

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

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

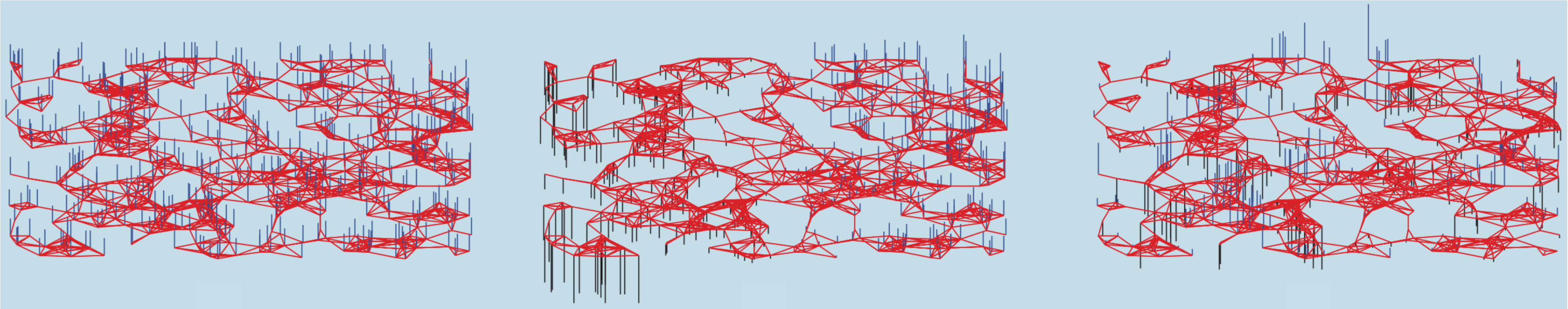
- Eigenvalues are usually sorted increasingly: $0 = \lambda_0 < \lambda_1 \leq \ldots \leq \lambda_{N-1}$

Graph Fourier transform

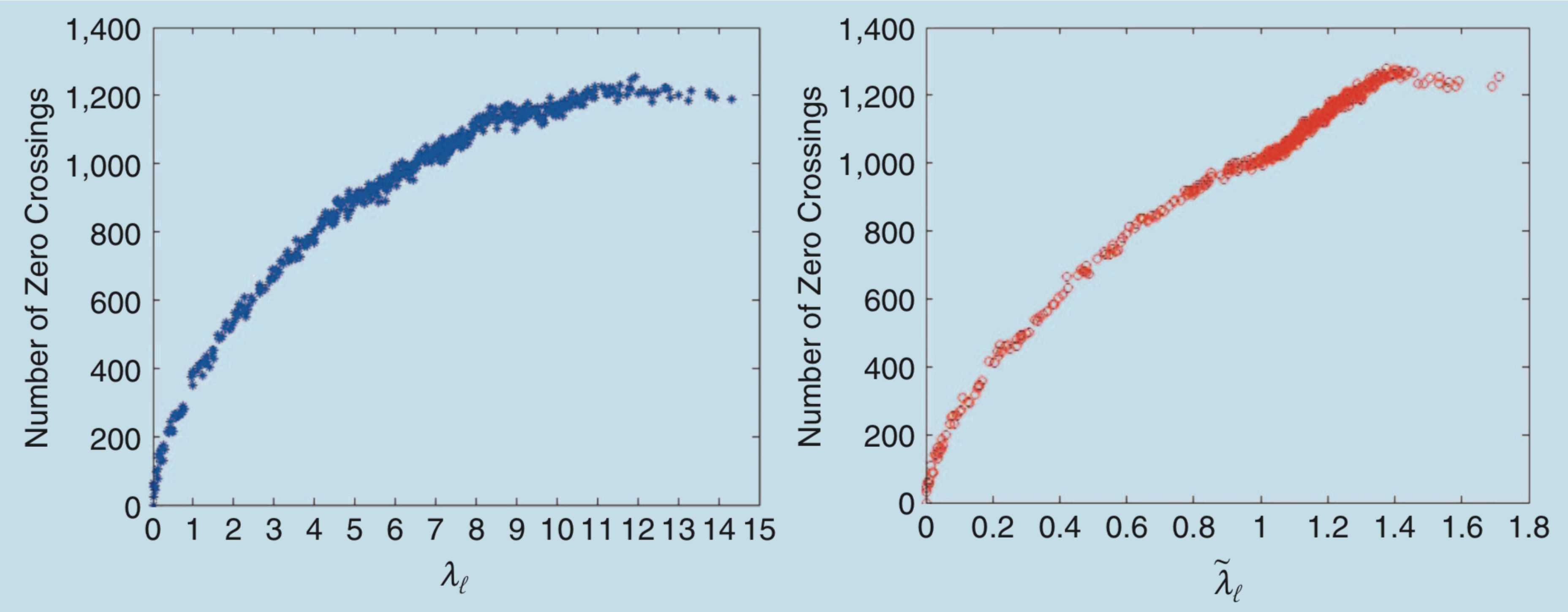


χ_0 χ_1 χ_{50} [Shuman13]

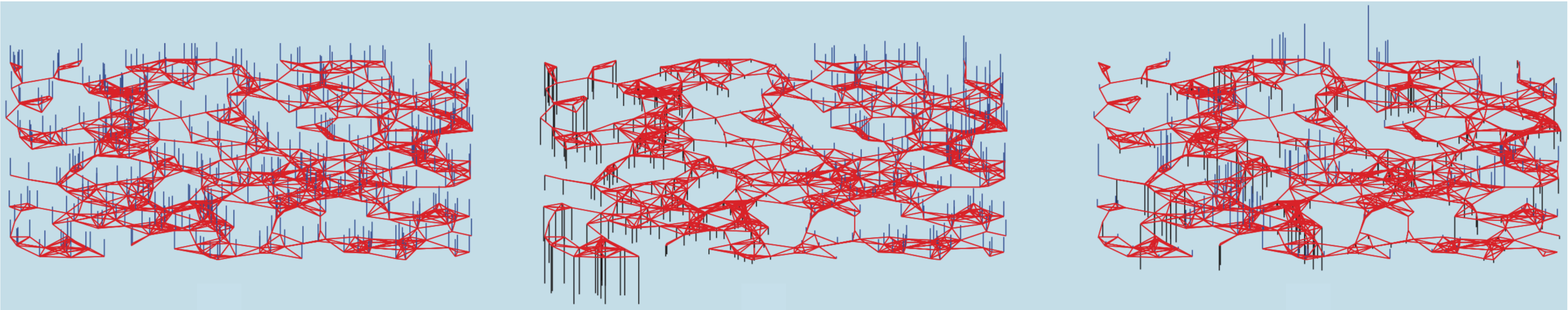
Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]



Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

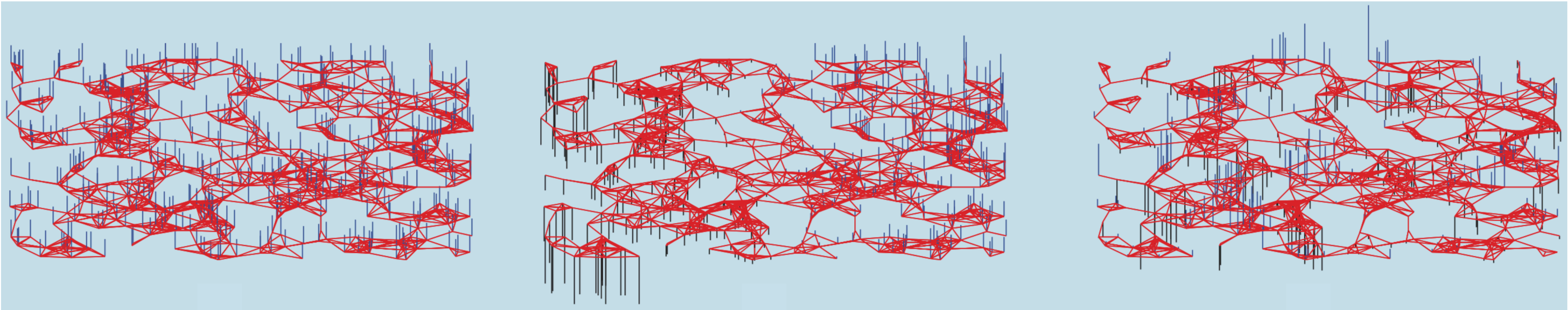
$L = \chi \Lambda \chi^T$

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

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

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

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

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

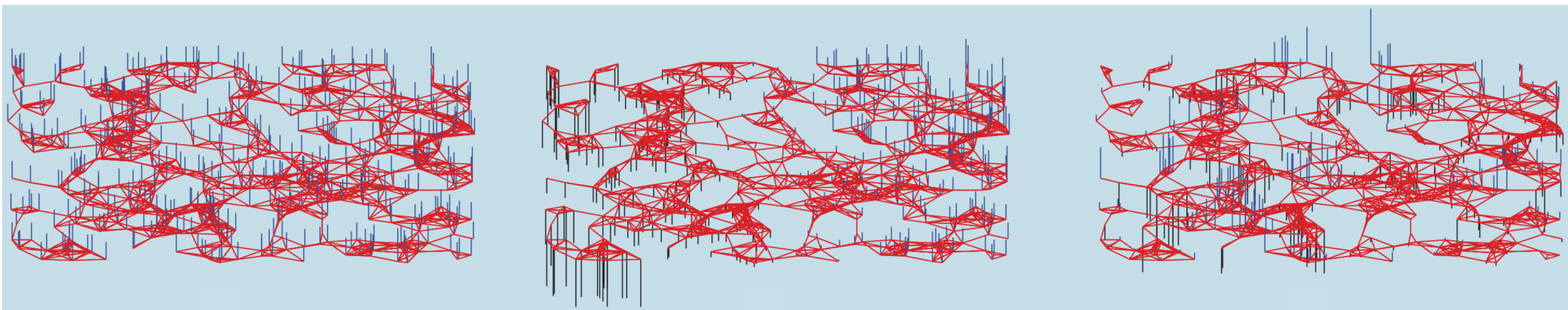
$\chi_0^T L \chi_0 = \lambda_0 = 0$

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

Graph Fourier transform:
[Hammond11]

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

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

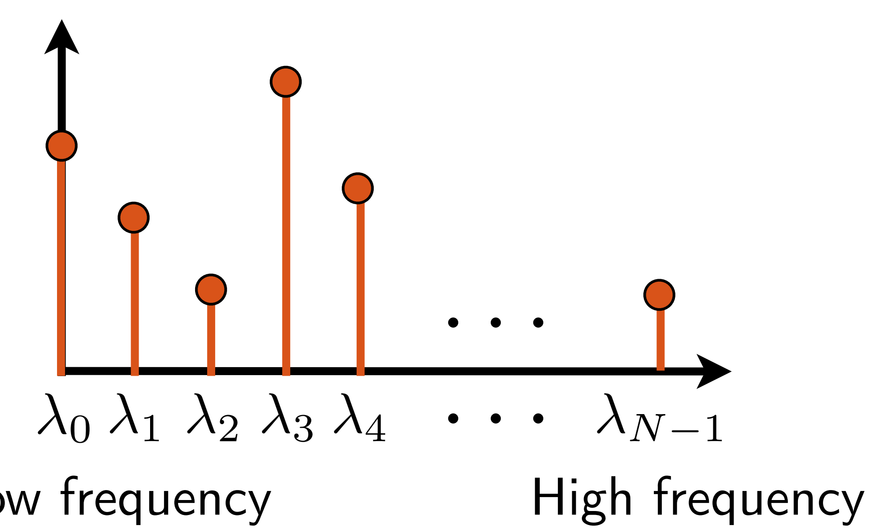
$L = \chi \Lambda \chi^T$

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

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

Graph Fourier transform:
[Hammond11]

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



Graph Fourier transform

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

Graph Fourier transform

- 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

- 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

- 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

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

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

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

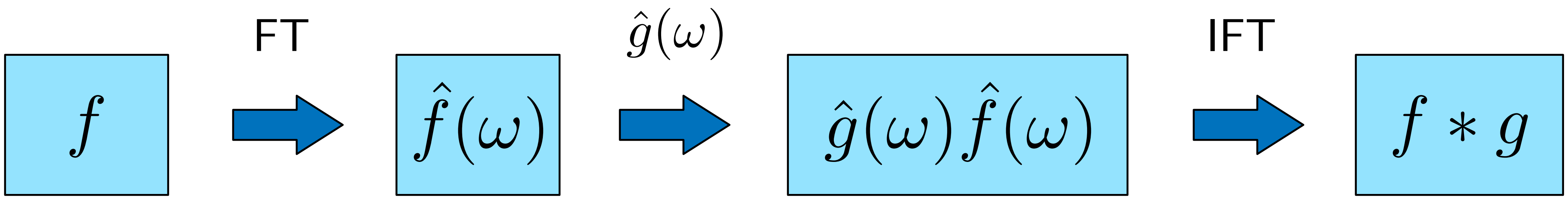
Classical frequency filtering

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

Classical frequency filtering

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

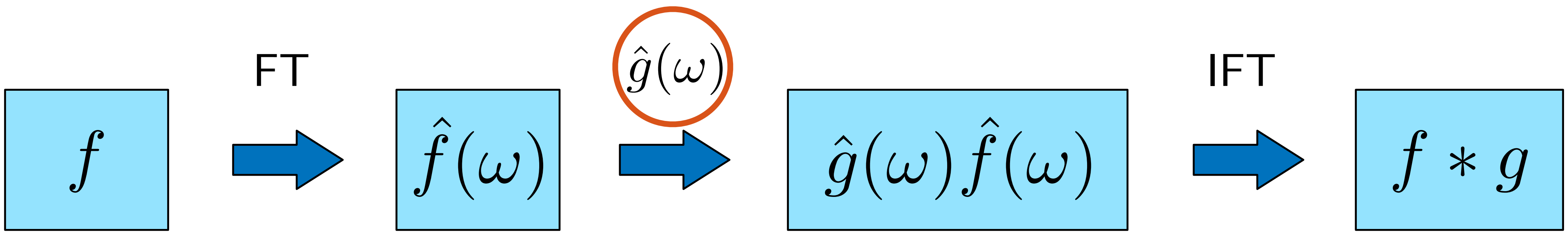
apply filter with transfer function $\hat{g}(\cdot)$ to a signal f

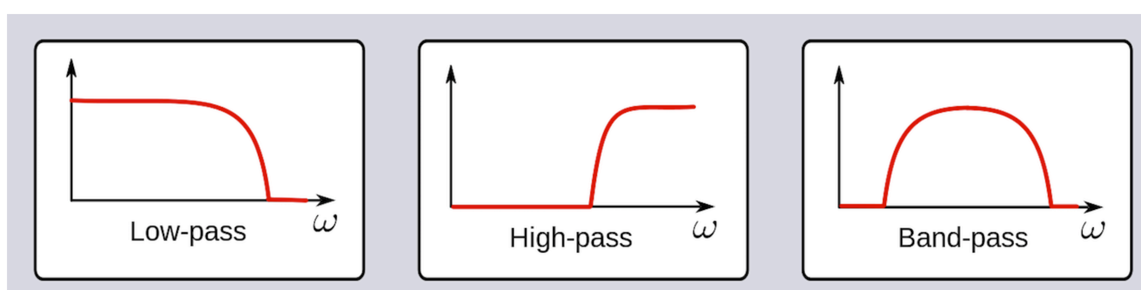


Classical frequency filtering

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





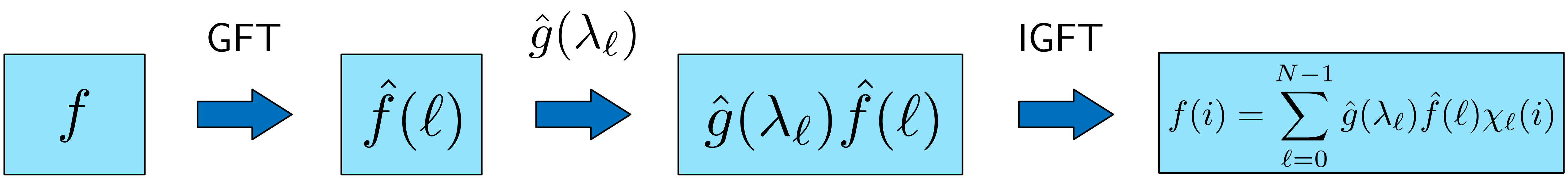
Graph spectral filtering

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

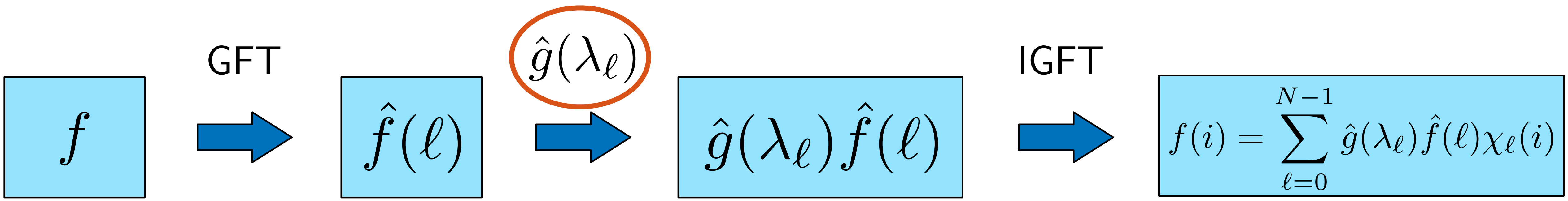
apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : \mathcal{V} \to \mathbb{R}^N$

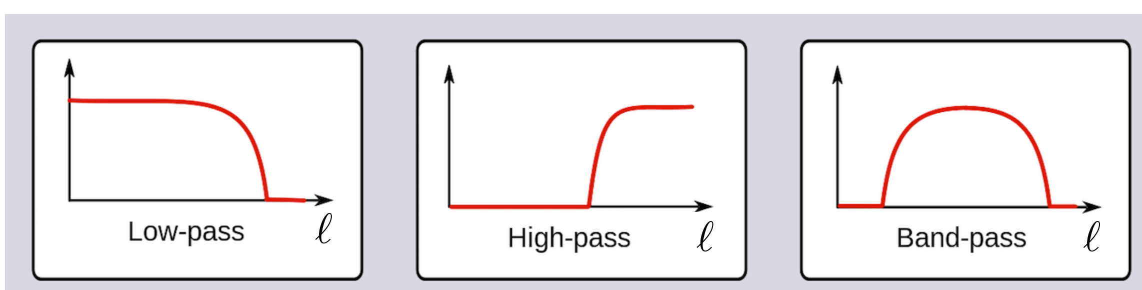


Graph spectral filtering

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

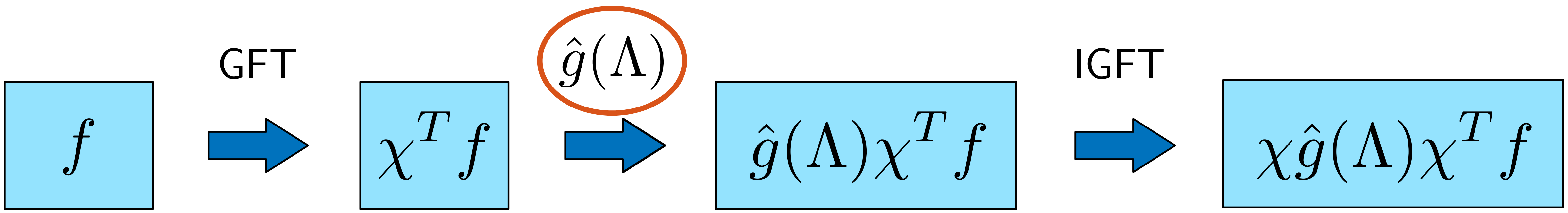




Graph spectral filtering

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



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

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

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

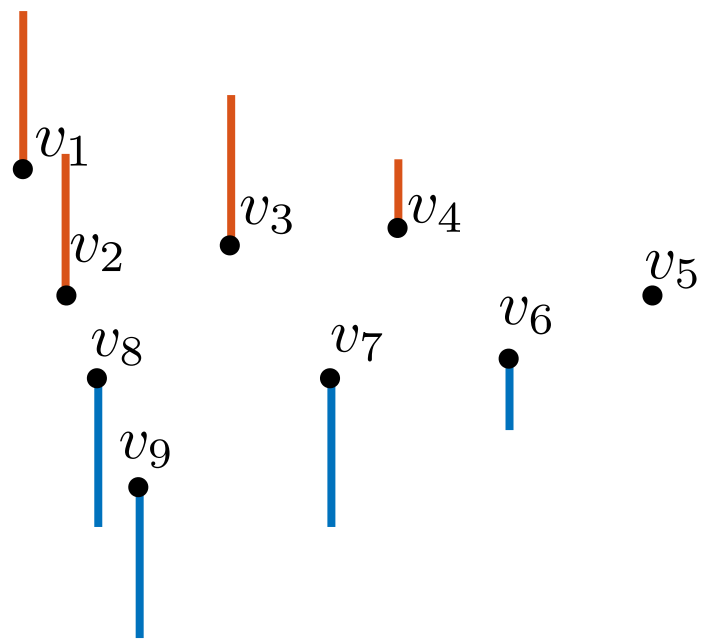
$\hat{g}(L)$: **function of L!**

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

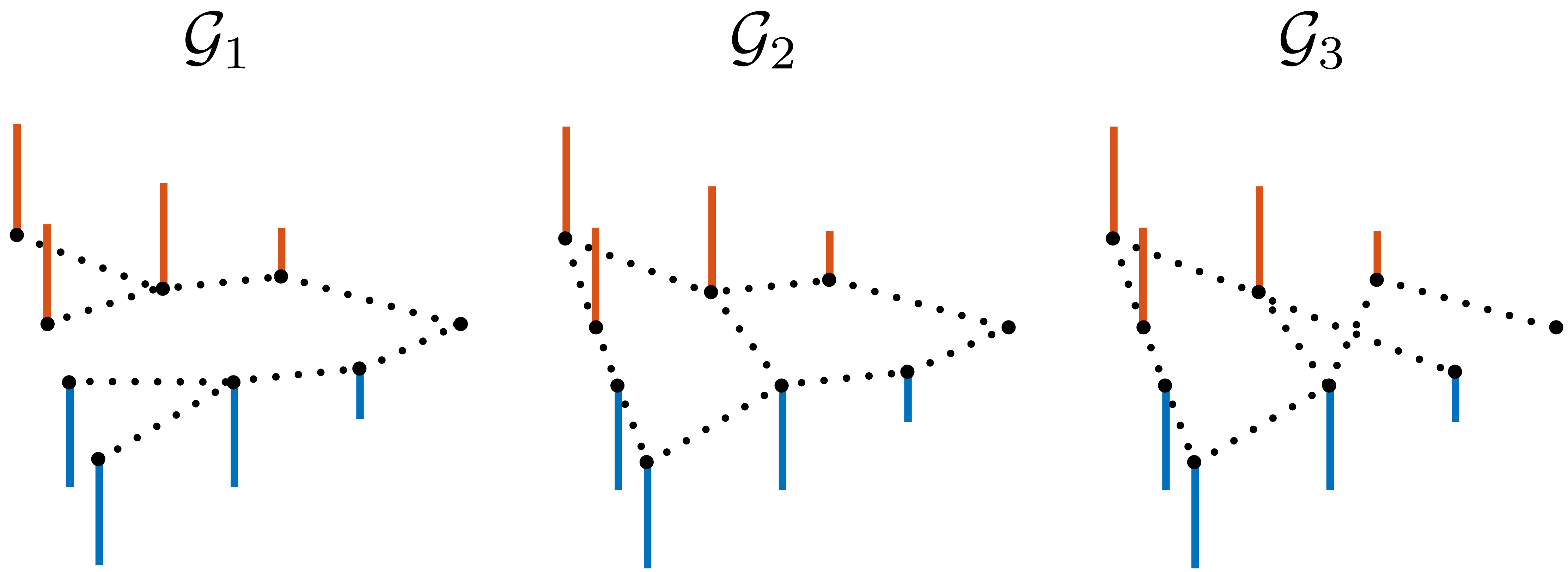
Outline

- A (very partial) literature overview
- A signal processing perspective
  - A brief introduction to graph signal processing (GSP)
  - GSP approaches for graph learning
- Concluding remarks

GSP for graph learning

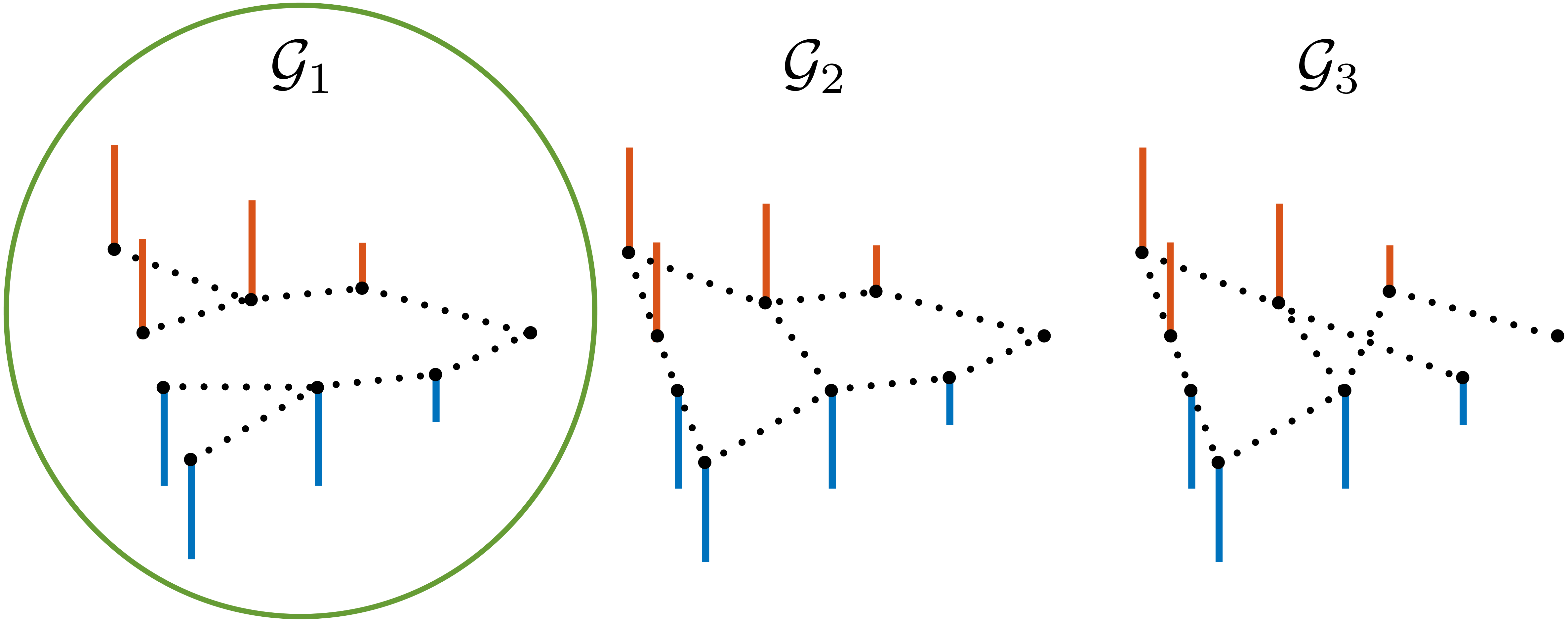


GSP for graph learning



which graph to choose?

GSP for graph learning



which graph to choose?

- **depends on the signal/graph model**
- **idea: choose one that enforces certain signal characteristics**

GSP for graph learning

- Existing approaches have limitations
  - simple correlation or similarity function is not enough
  - statistical models do not always lead to non-negative edge weights
  - many impose a “global” distribution or behaviour

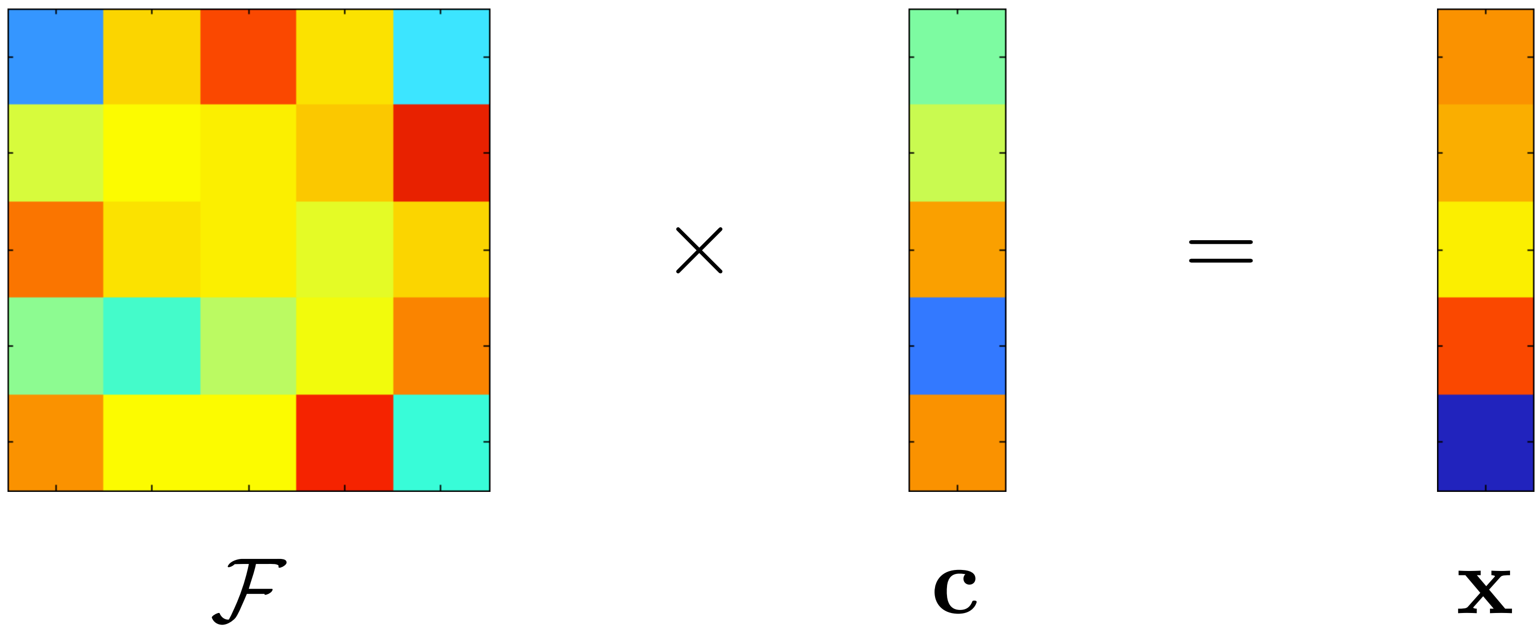
GSP for graph learning

- Existing approaches have limitations
  - simple correlation or similarity function is not enough
  - statistical models do not always lead to non-negative edge weights
  - many impose a "global" distribution or behaviour

- Opportunity and challenge for GSP
  - GSP tools offer another "regulariser" for complicated inference: frequency or spectral representation
  - filtering-based approaches can provide generative models for signals with complex (non-Gaussian) behaviour

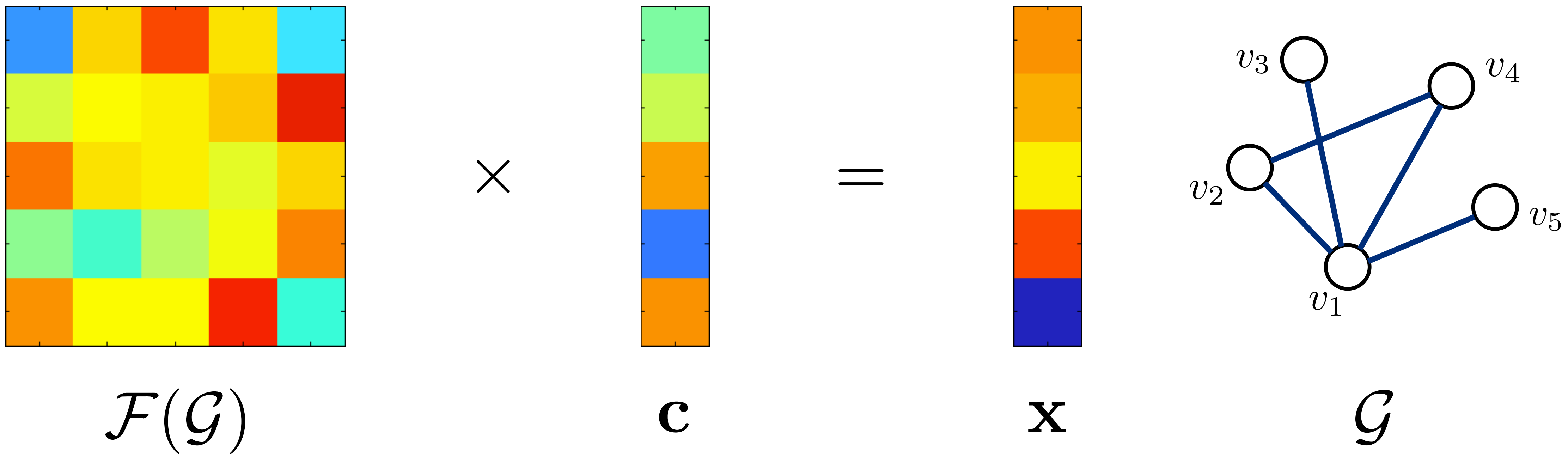
GSP for graph learning

- Signal processing is about $\mathbf{F}\ \mathbf{c} = \mathbf{x}$



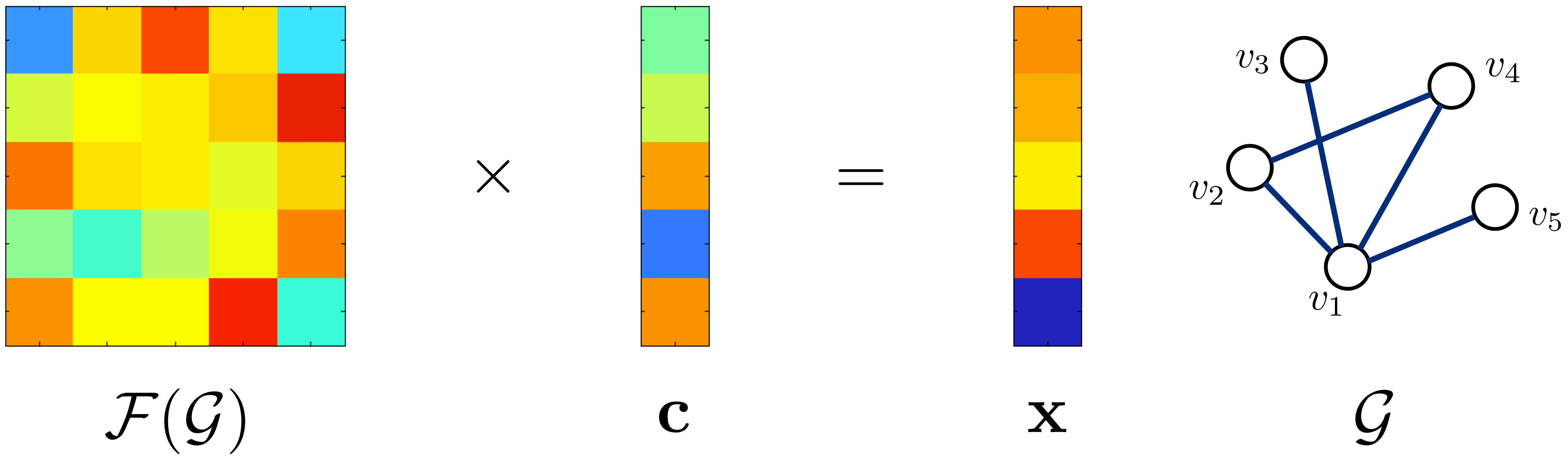
GSP for graph learning

- Graph signal processing is about $\mathbf{F}(\mathbf{G})\ \mathbf{c} = \mathbf{x}$



GSP for graph learning

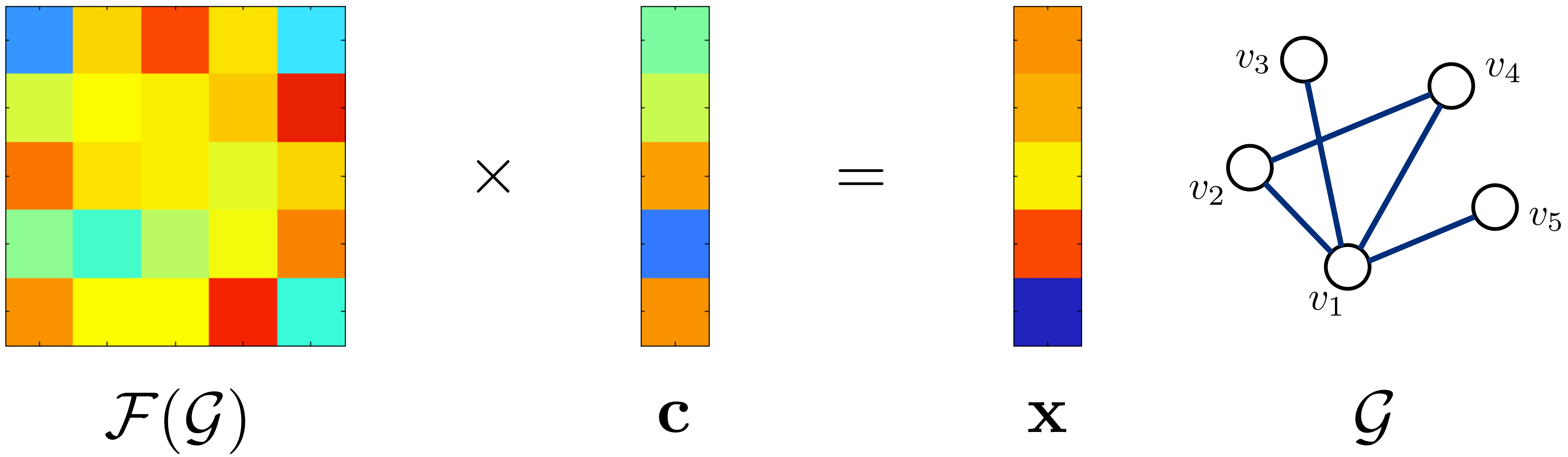
- Forward: given **G** and **x**, design **F** to study **c**



Fourier/wavelet atoms	graph Fourier/ wavelet coefficient	[Coifman06,Narang09,Hammond11, Shuman13,Sandryhaila13]
trained dictionary atoms	graph dictionary coefficient	[Zhang12,Thanou14]

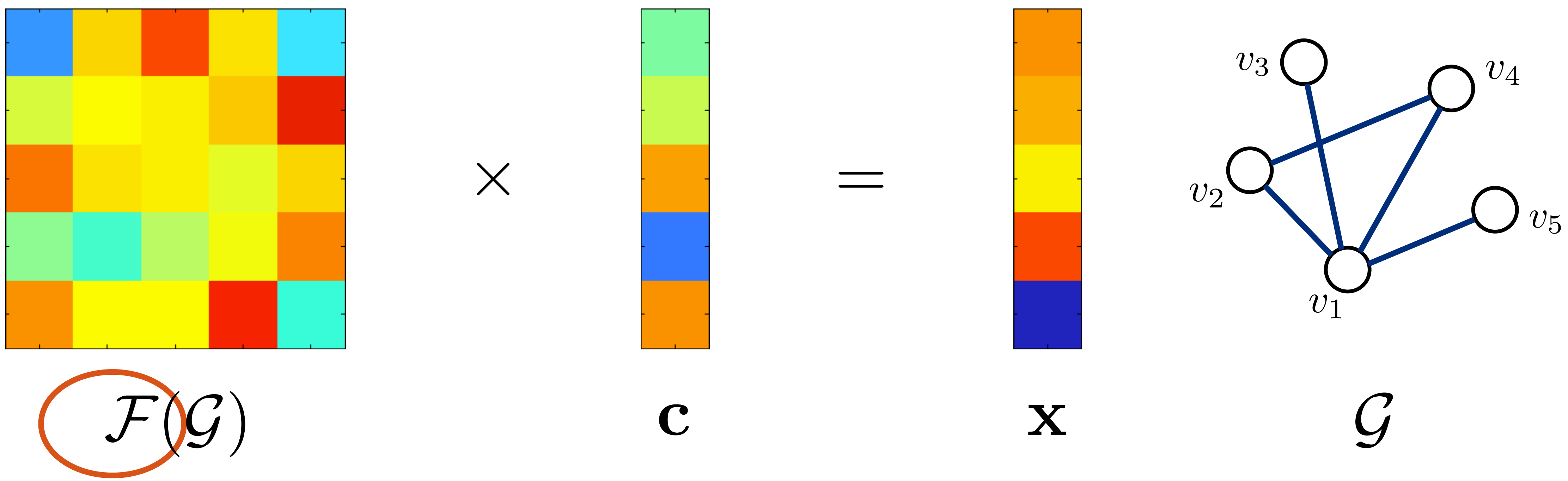
GSP for graph learning

- Backward (graph learning): given **x**, design **F** and **c** to infer **G**



GSP for graph learning

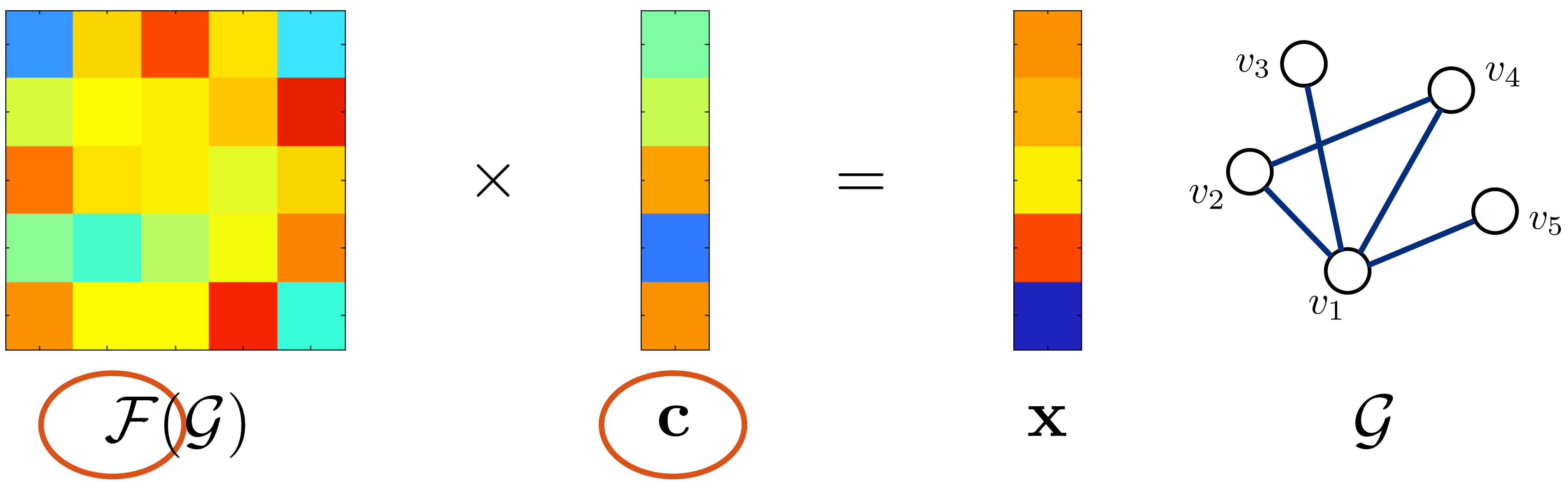
- Backward (graph learning): given **x**, design **F** and **c** to infer **G**



- **key is signal/graph model behind F**
- **via graph operators (adjacency/Laplacian or shift operators)**

GSP for graph learning

- Backward (graph learning): given **x**, design **F** and **c** to infer **G**



- key is signal/graph model behind F
- via graph operators (adjacency/Laplacian or shift operators)
- assumption on c also determines signal characteristics

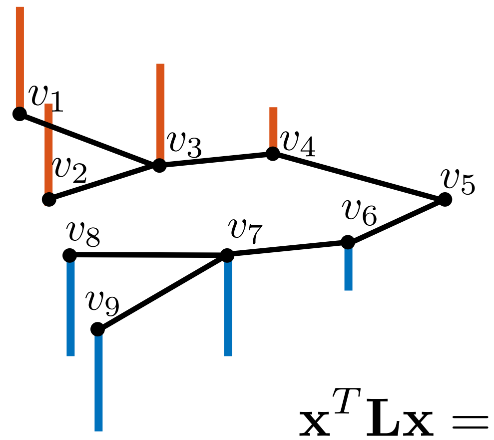
Model 1: Global smoothness

- Signal varies smoothly between all pairs of nodes that are connected
- Example: temperature of different locations in a flat geographical region

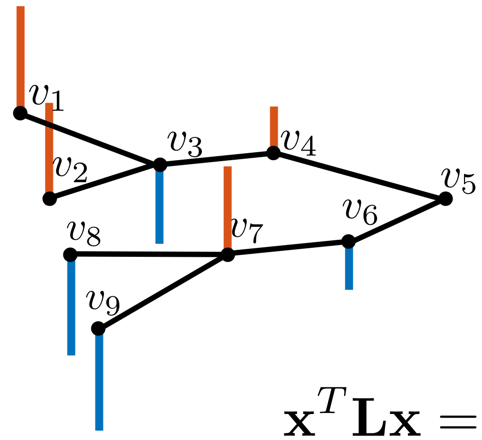
Model 1: Global smoothness

- Signal varies smoothly between all pairs of nodes that are connected
- Example: temperature of different locations in a flat geographical region
- Usually quantified by the Laplacian quadratic form:

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i,j} \mathbf{W}_{ij} \left(\mathbf{x}(i) - \mathbf{x}(j) \right)^2$$



$\mathbf{x}^T \mathbf{L} \mathbf{x} = 1$

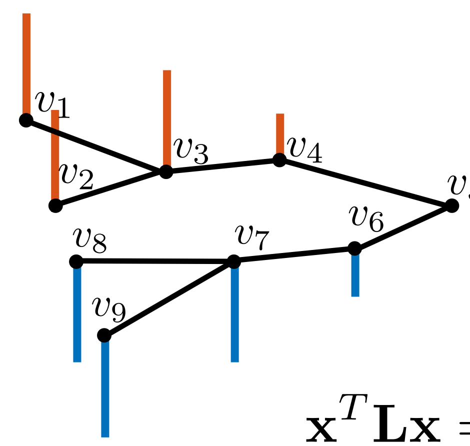


$\mathbf{x}^T \mathbf{L} \mathbf{x} = 21$

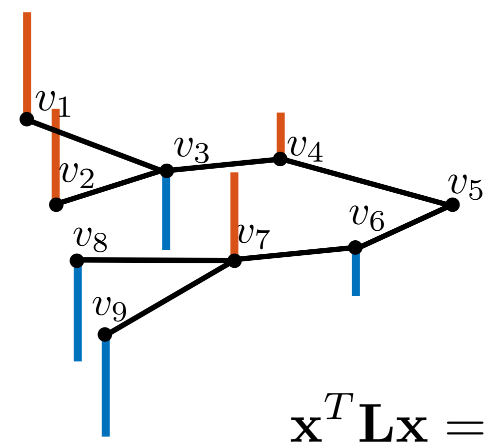
Model 1: Global smoothness

- Signal varies smoothly between all pairs of nodes that are connected
- Example: temperature of different locations in a flat geographical region
- Usually quantified by the Laplacian quadratic form:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j}\mathbf{W}_{ij}\left(\mathbf{x}(i) - \mathbf{x}(j)\right)^2$$



$\mathbf{x}^T\mathbf{L}\mathbf{x} = 1$



$\mathbf{x}^T\mathbf{L}\mathbf{x} = 21$

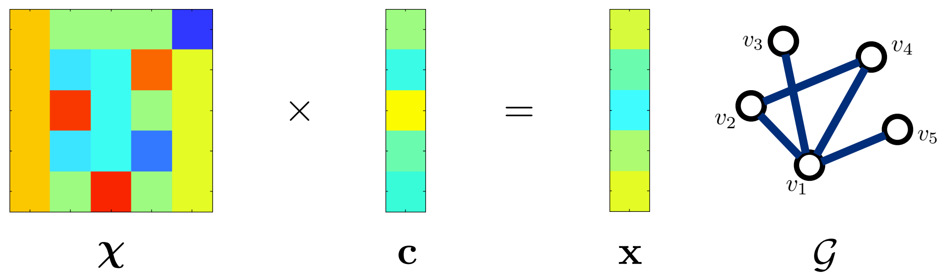
similar to previous approaches:

Lake (2010): $\max_{\boldsymbol{\Theta}=\mathbf{L}+\frac{1}{\sigma^2}\mathbf{I}} \log\det\boldsymbol{\Theta} - \frac{1}{M}\mathrm{tr}(\mathbf{X}\mathbf{X}^T\boldsymbol{\Theta}) - \rho||\boldsymbol{\Theta}||_1$

Daitch (2009): $\min_{\mathbf{L}} \mathbf{X}^T\mathbf{L}^2\mathbf{X}$

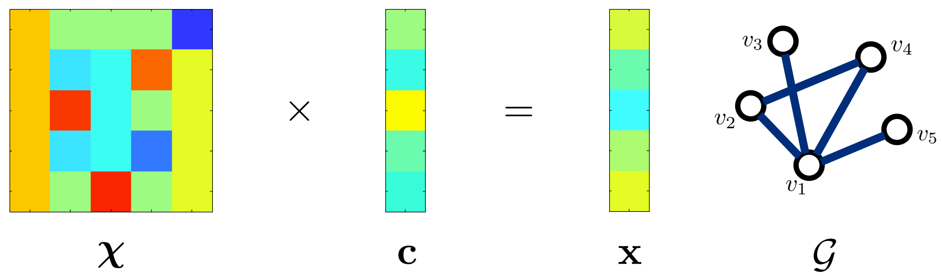
Hu (2013): $\min_{\mathbf{L}} \mathrm{tr}(\mathbf{X}^T\mathbf{L}^s\mathbf{X}) - \beta||\mathbf{W}||_F$

Model 1: Global smoothness



- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \boldsymbol{\chi}$ (eigenvector matrix of **L**)
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{\dagger})$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger} + \sigma_{\epsilon}^{2}\mathbf{I})$: **Gaussian Markov random field**

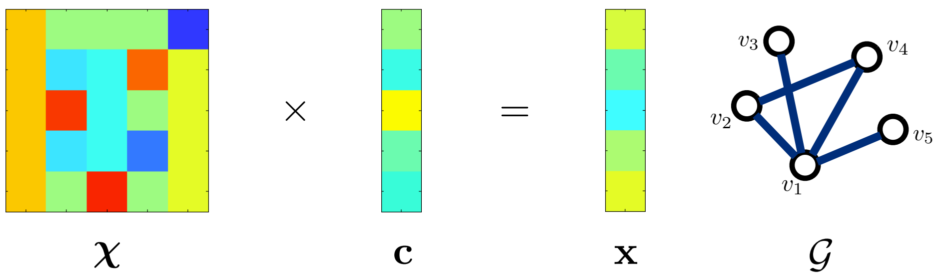
Model 1: Global smoothness



- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \boldsymbol{\chi}$ (eigenvector matrix of **L**)
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{\dagger})$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger} + \sigma_{\epsilon}^{2}\mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of $\mathbf{c}$ leads to minimisation of Laplacian quadratic form:

$$\min_{\mathbf{c}} ||\mathbf{x} - \boldsymbol{\chi}\mathbf{c}||_2^2 + \alpha\ \mathbf{c}^T \Lambda \mathbf{c}$$

Model 1: Global smoothness



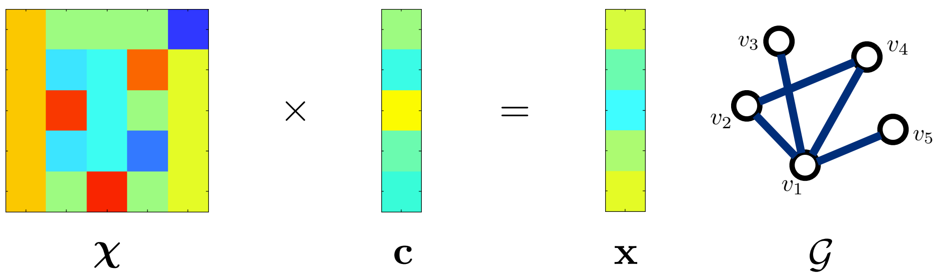
- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \boldsymbol{\chi}$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^\dagger)$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger + \sigma_\epsilon^2 \mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of **c** leads to minimisation of Laplacian quadratic form:

$$\min_{\mathbf{c}} ||\mathbf{x} - \boldsymbol{\chi}\mathbf{c}||_2^2 + \alpha\ \mathbf{c}^T \Lambda \mathbf{c}$$

$$\Downarrow\ \mathbf{y} = \boldsymbol{\chi}\mathbf{c}$$

$$\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_2^2 + \alpha\ \mathbf{y}^T \mathbf{L} \mathbf{y}$$

Model 1: Global smoothness

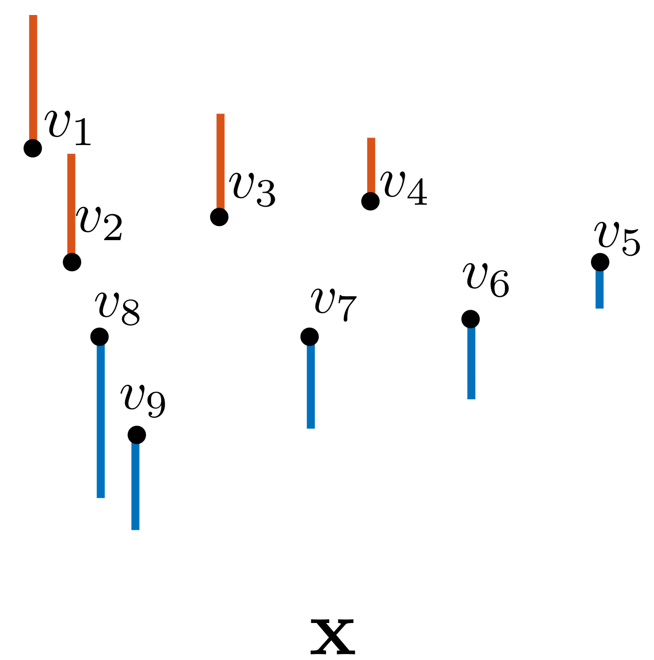


- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^\dagger)$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger + \sigma_\epsilon^2 \mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of **c** leads to minimisation of Laplacian quadratic form:

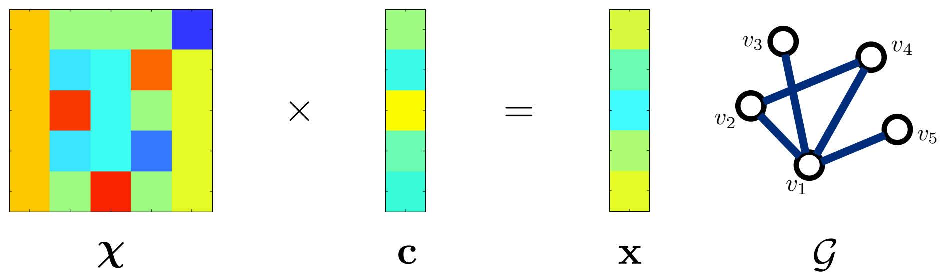
$$\min_{\mathbf{c}} ||\mathbf{x} - \boldsymbol{\chi}\mathbf{c}||_2^2 + \alpha\ \mathbf{c}^T \Lambda \mathbf{c}$$

$$\Downarrow \quad \mathbf{y} = \boldsymbol{\chi}\mathbf{c}$$

$$\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_2^2 + \alpha\ \mathbf{y}^T \mathbf{L} \mathbf{y}$$



Model 1: Global smoothness

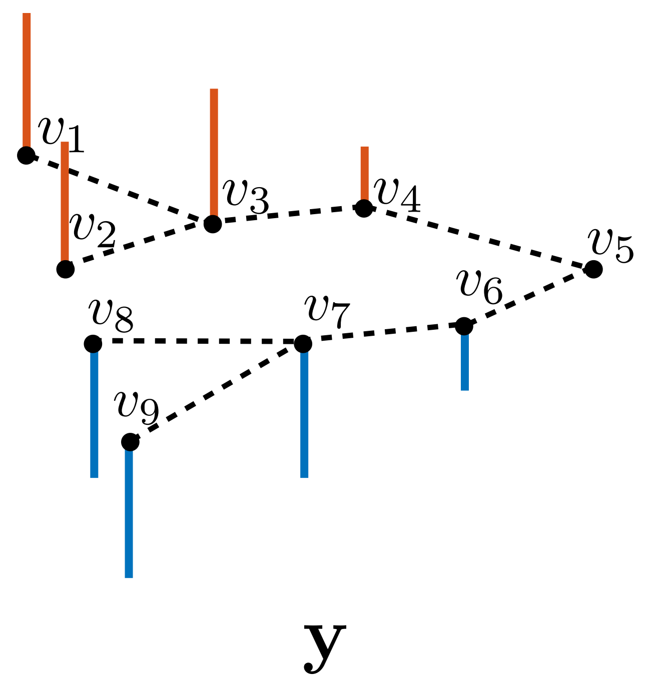


- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{\dagger})$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger} + \sigma_{\epsilon}^2 \mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of **c** leads to minimisation of Laplacian quadratic form:

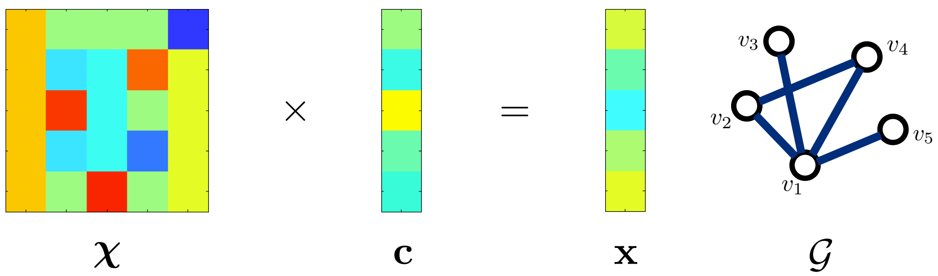
$$\min_{\mathbf{c}} ||\mathbf{x} - \chi \mathbf{c}||_2^2 + \alpha \ \mathbf{c}^T \Lambda \mathbf{c}$$

$$\mathbf{y} = \chi \mathbf{c}$$

$$\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_2^2 + \alpha \ \mathbf{y}^T \mathbf{L} \mathbf{y}$$



Model 1: Global smoothness



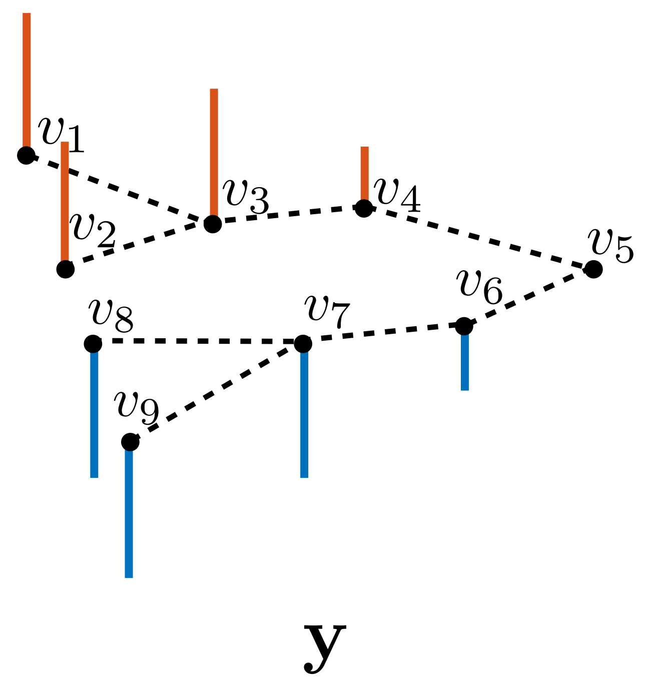
- Dong et al. (2016)
  - $\mathcal{F}(\mathcal{G}) = \boldsymbol{\chi}$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{\dagger})$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger} + \sigma_{\epsilon}^{2}\mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of **c** leads to minimisation of Laplacian quadratic form:

$$\min_{\mathbf{c}} ||\mathbf{x} - \boldsymbol{\chi}\mathbf{c}||_2^2 + \alpha\ \mathbf{c}^T \Lambda \mathbf{c}$$

$$\mathbf{y} = \boldsymbol{\chi}\mathbf{c}$$

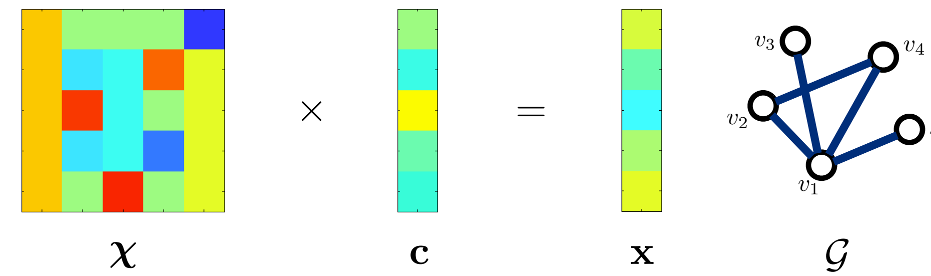
$$\min_{\mathbf{L},\mathbf{Y}} ||\mathbf{X} - \mathbf{Y}||_F^2 + \alpha\ \mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) + \beta||\mathbf{L}||_F^2$$

data fidelity   smoothness on **Y**   regularisation



Model 1: Global smoothness

- Dong et al. (2016)



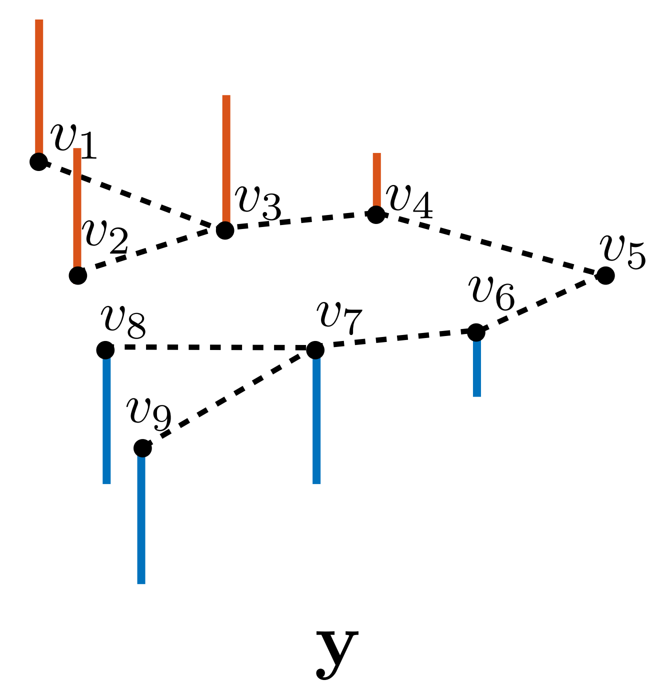
  - $\mathcal{F}(\mathcal{G}) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{\dagger})$
  - $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^{\dagger} + \sigma_{\epsilon}^{2}\mathbf{I})$: **Gaussian Markov random field**
  - maximum a posteriori (MAP) estimate of **c** leads to minimisation of Laplacian quadratic form:

$$\min_{\mathbf{c}} ||\mathbf{x} - \chi\mathbf{c}||_2^2 + \alpha\, \mathbf{c}^T \Lambda \mathbf{c}$$

$$\mathbf{y} = \chi\mathbf{c}$$

$$\min_{\mathbf{L},\mathbf{Y}} ||\mathbf{X} - \mathbf{Y}||_F^2 + \alpha\, \mathrm{tr}(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) + \beta||\mathbf{L}||_F^2$$

data fidelity — smoothness on **Y** — regularisation



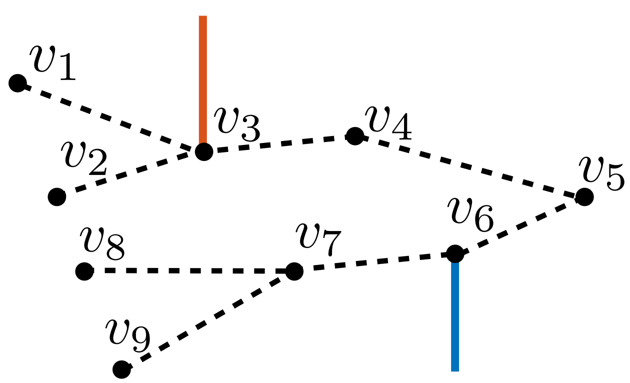
learning enforces signal property (global smoothness)

Model 2: Spectral filtering

- Signals are outcome of applying filtering to latent (input) signals
- Filtering often corresponds to a diffusion process on graphs (different spectral characteristics or localisation properties)
- Example: movement of people/vehicles in transportation network

Model 2: Spectral filtering

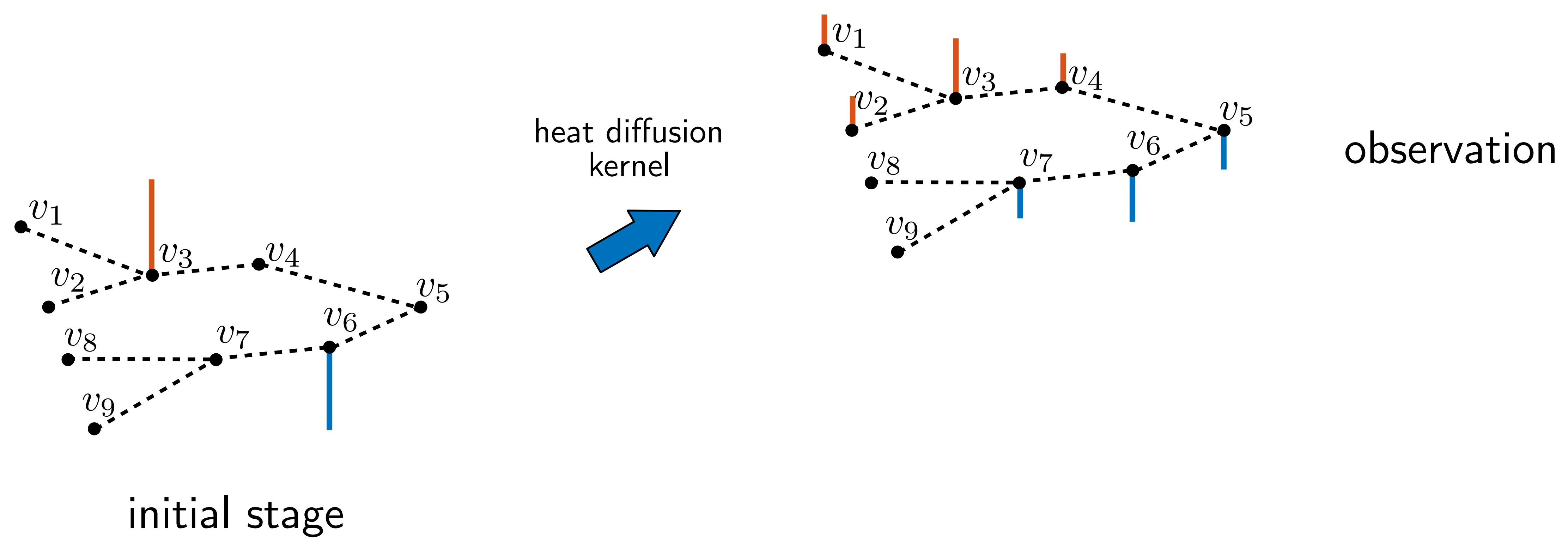
- Signals are outcome of applying filtering to latent (input) signals
- Filtering often corresponds to a diffusion process on graphs (different spectral characteristics or localisation properties)
- Example: movement of people/vehicles in transportation network



initial stage

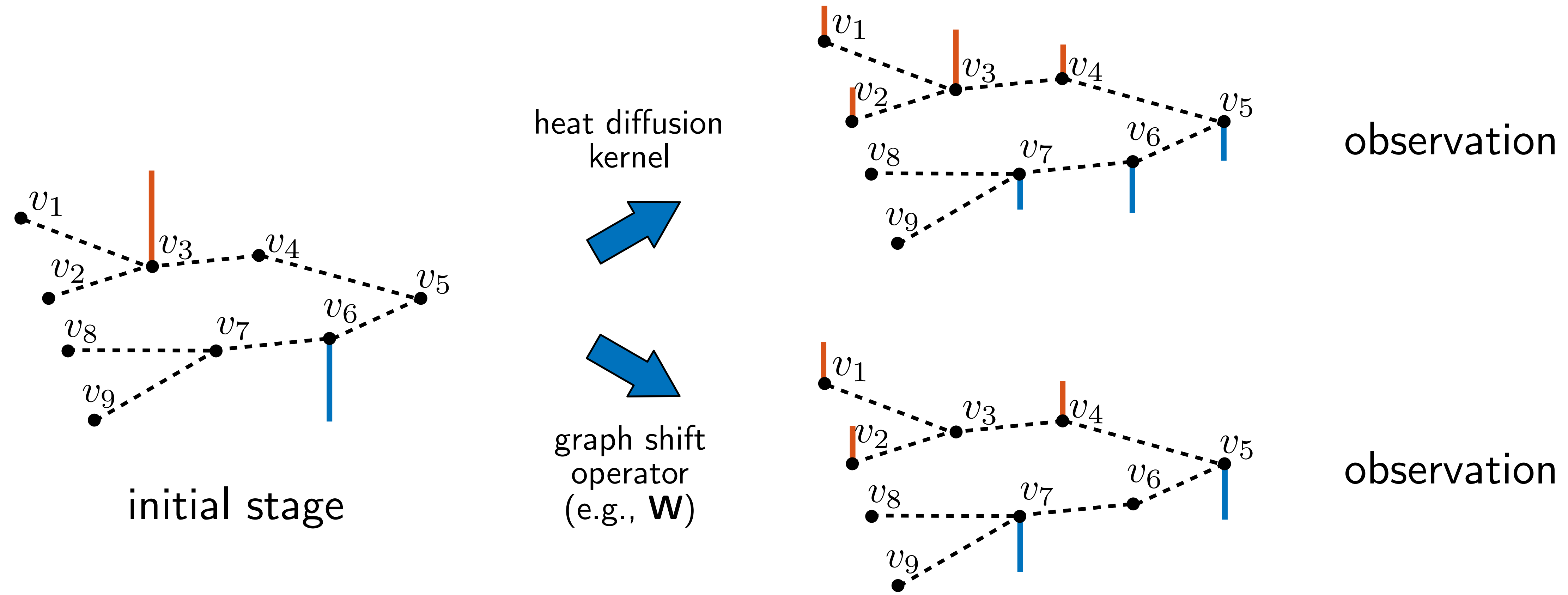
Model 2: Spectral filtering

- Signals are outcome of applying filtering to latent (input) signals
- Filtering often corresponds to a diffusion process on graphs (different spectral characteristics or localisation properties)
- Example: movement of people/vehicles in transportation network



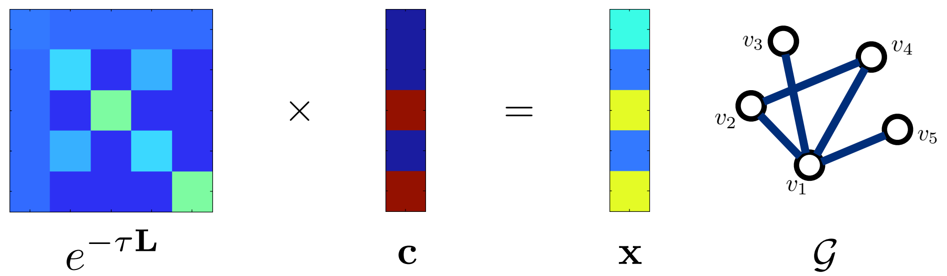
Model 2: Spectral filtering

- Signals are outcome of applying filtering to latent (input) signals
- Filtering often corresponds to a diffusion process on graphs (different spectral characteristics or localisation properties)
- Example: movement of people/vehicles in transportation network

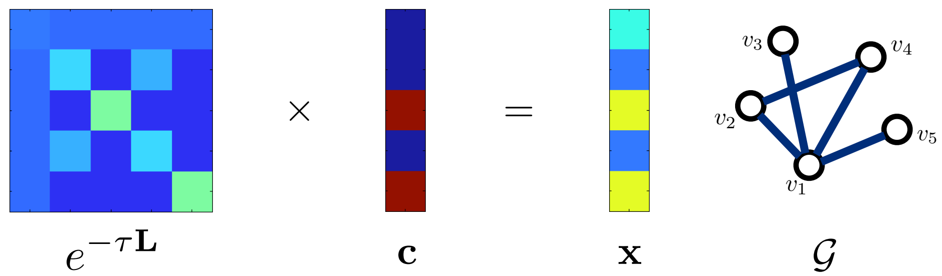


Model 2: Spectral filtering

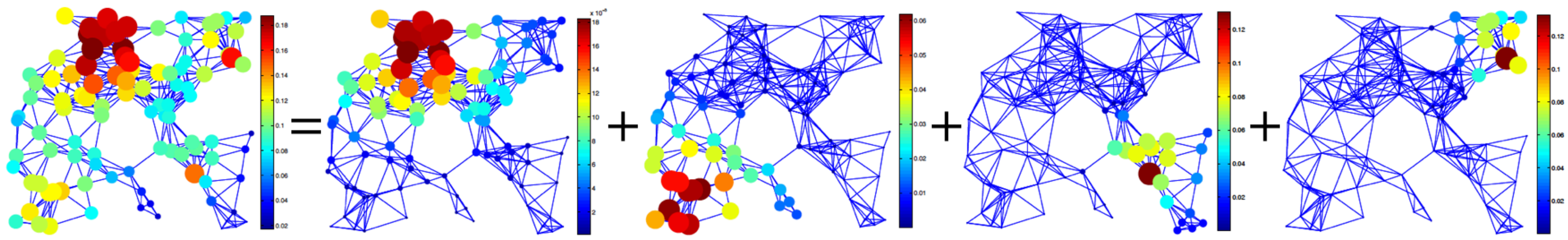
- Thanou et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = e^{-\tau \mathbf{L}}$ (localisation in vertex domain)
  - sparsity assumption on $\mathbf{c}$



Model 2: Spectral filtering



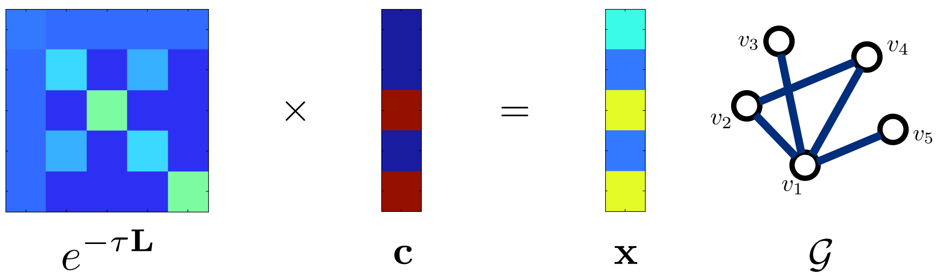
- Thanou et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = e^{-\tau \mathbf{L}}$ (localisation in vertex domain)
  - sparsity assumption on $\mathbf{c}$
  - each signal is a combination of several heat diffusion processes at time τ

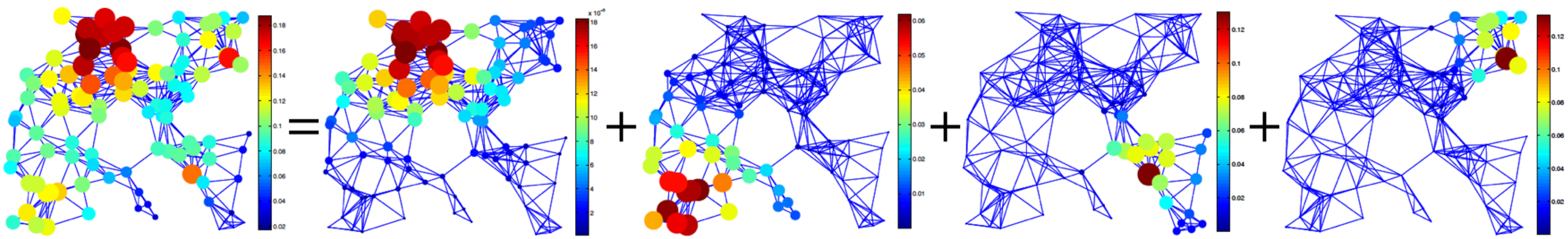


[Thanou17]

Model 2: Spectral filtering

- Thanou et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = e^{-\tau \mathbf{L}}$ (localisation in vertex domain)
  - sparsity assumption on $\mathbf{c}$
  - each signal is a combination of several heat diffusion processes at time τ



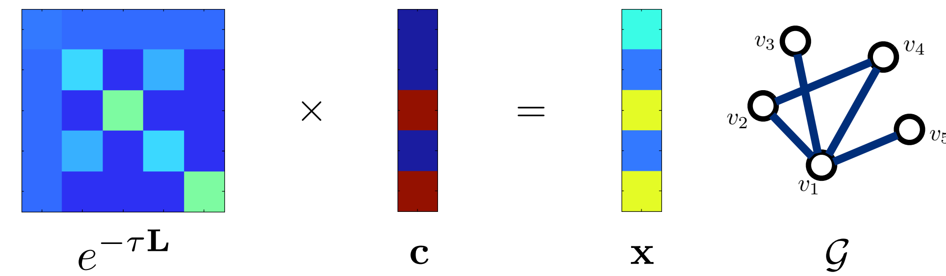


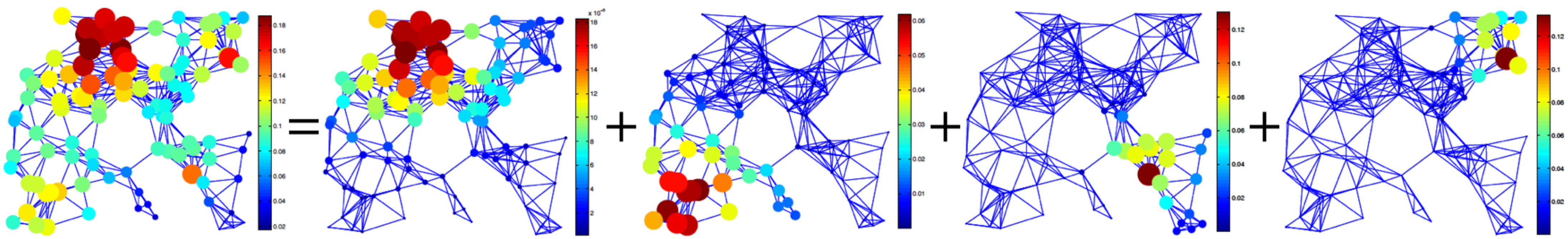
[Thanou17]

$$\min_{\mathbf{L},\mathbf{C},\tau} ||\mathbf{X} - \mathcal{F}\mathbf{C}||_F^2 + \alpha \sum_{m=1}^{M} ||\mathbf{c}_m||_1 + \beta ||\mathbf{L}||_F^2 \qquad \text{s.t.} \quad \mathcal{F} = [e^{-\tau_1 \mathbf{L}}, e^{-\tau_2 \mathbf{L}}, \ldots, e^{-\tau_S \mathbf{L}}]$$

Model 2: Spectral filtering

- Thanou et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = e^{-\tau\mathbf{L}}$ (localisation in vertex domain)
  - sparsity assumption on $\mathbf{c}$
  - each signal is a combination of several heat diffusion processes at time τ





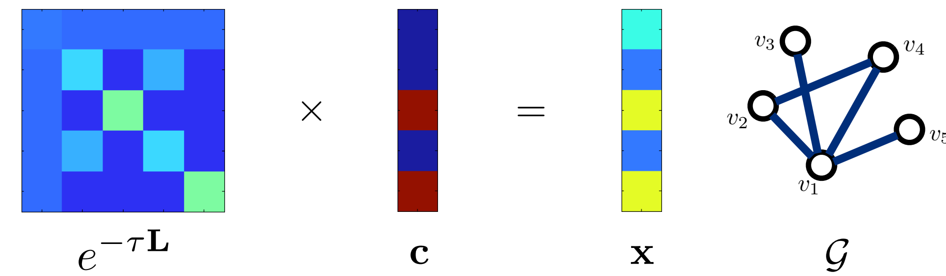
[Thanou17]

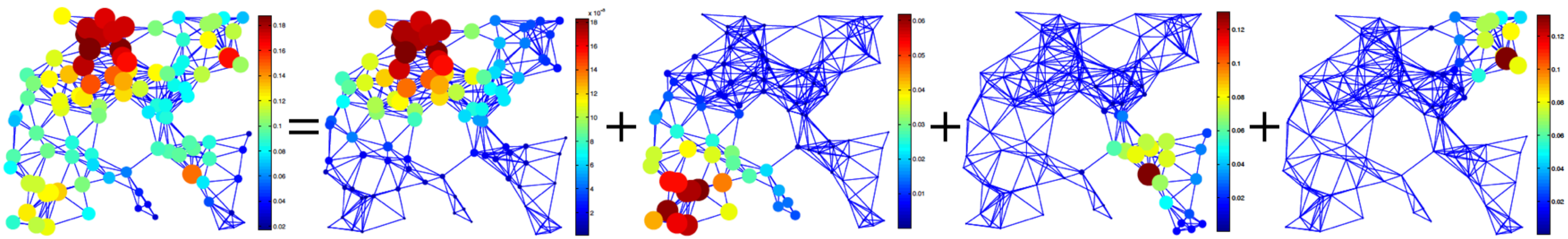
$$\min_{\mathbf{L},\mathbf{C},\tau} ||\mathbf{X} - \mathcal{F}\mathbf{C}||_F^2 + \alpha \sum_{m=1}^{M} ||\mathbf{c}_m||_1 + \beta ||\mathbf{L}||_F^2 \qquad \text{s.t.} \quad \mathcal{F} = [e^{-\tau_1\mathbf{L}}, e^{-\tau_2\mathbf{L}}, \ldots, e^{-\tau_S\mathbf{L}}]$$

data fidelity — sparsity on $\mathbf{c}$ — regularisation

Model 2: Spectral filtering

- Thanou et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = e^{-\tau \mathbf{L}}$ (localisation in vertex domain)
  - sparsity assumption on $\mathbf{c}$
  - each signal is a combination of several heat diffusion processes at time τ





[Thanou17]

$$\min_{\mathbf{L},\mathbf{C},\tau} ||\mathbf{X} - \mathcal{F}\mathbf{C}||_F^2 + \alpha \sum_{m=1}^{M} ||\mathbf{c}_m||_1 + \beta ||\mathbf{L}||_F^2 \qquad \text{s.t.} \quad \mathcal{F} = [e^{-\tau_1 \mathbf{L}}, e^{-\tau_2 \mathbf{L}}, \dots, e^{-\tau_S \mathbf{L}}]$$

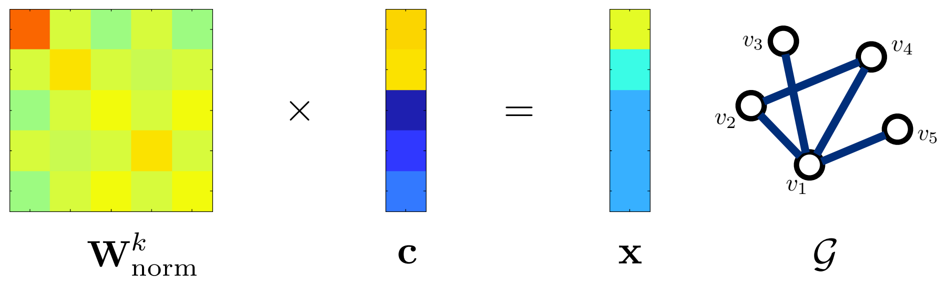
data fidelity   sparsity on $\mathbf{c}$ regularisation

local (instead of global) signal characteristics

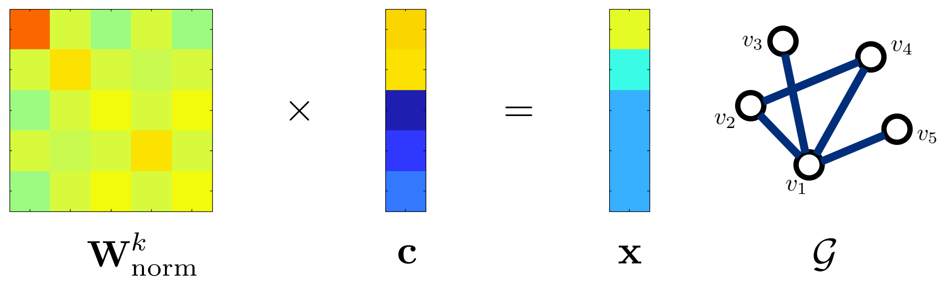
can be extended to general polynomial case

Model 2: Spectral filtering

- Pasdeloup et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}} = \mathbf{W}^{\mathbf{k}}_{\text{norm}}$
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$



Model 2: Spectral filtering

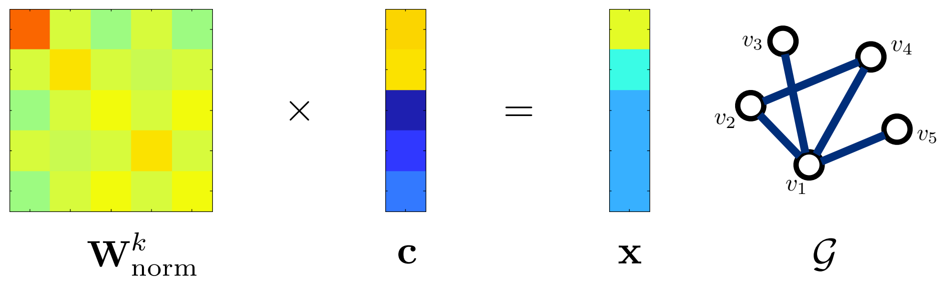


- Pasdeloup et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}} = \mathbf{W}^{\mathbf{k}}_{\text{norm}}$
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - two-step approach:
    - estimate eigenvector matrix of graph operator from sample covariance:

$$\mathbf{\Sigma} = \mathbb{E}\Big[\sum_{m=1}^{M} \mathbf{X}(m)\mathbf{X}(m)^T\Big] = \sum_{m=1}^{M} \mathbf{W}^{2\mathbf{k}(m)}_{\text{norm}}$$ (polynomial of $\mathbf{W}_{\text{norm}}$)

Model 2: Spectral filtering

- Pasdeloup et al. (2017)

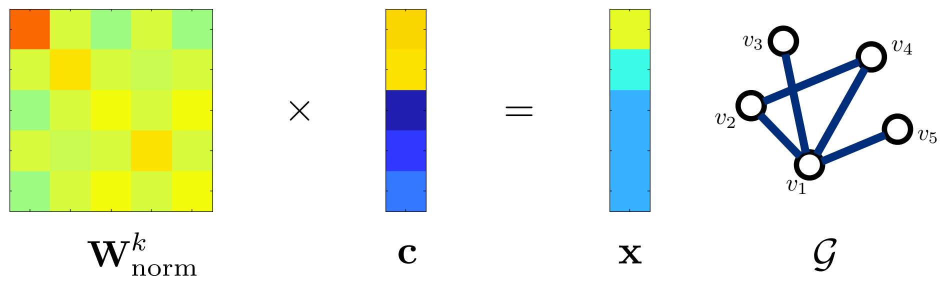


  - $\mathcal{F}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}} = \mathbf{W}_{\mathrm{norm}}^{\mathbf{k}}$
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - two-step approach:
    - estimate eigenvector matrix of graph operator from sample covariance:

$$\boldsymbol{\Sigma} = \mathbb{E}\Big[\sum_{m=1}^{M} \mathbf{X}(m)\mathbf{X}(m)^{T}\Big] = \sum_{m=1}^{M} \mathbf{W}_{\mathrm{norm}}^{2\mathbf{k}(m)} \quad \text{(polynomial of } \mathbf{W}_{\mathrm{norm}}\text{)}$$

    - optimise for eigenvalues given constraints of $\mathbf{W}_{\mathrm{norm}}$ (e.g., non-negativity of off-diagonals and range of eigenvalues) and certain priors (e.g., sparsity)

Model 2: Spectral filtering



- Pasdeloup et al. (2017)
  - $\mathcal{F}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}} = \mathbf{W}_{\text{norm}}^{\mathbf{k}}$
  - Gaussian assumption on $\mathbf{c}$: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - two-step approach:
    - estimate eigenvector matrix of graph operator from sample covariance:

$$\boldsymbol{\Sigma} = \mathbb{E}\Big[\sum_{m=1}^{M} \mathbf{X}(m)\mathbf{X}(m)^T\Big] = \sum_{m=1}^{M} \mathbf{W}_{\text{norm}}^{2\mathbf{k}(m)}$$ (polynomial of $\mathbf{W}_{\text{norm}}$)

    - optimise for eigenvalues given constraints of $\mathbf{W}_{\text{norm}}$ (e.g., non-negativity of off-diagonals and range of eigenvalues) and certain priors (e.g., sparsity)

diffusion process based on different operator

statistical vs structural (Thanou et al.) assumption on c

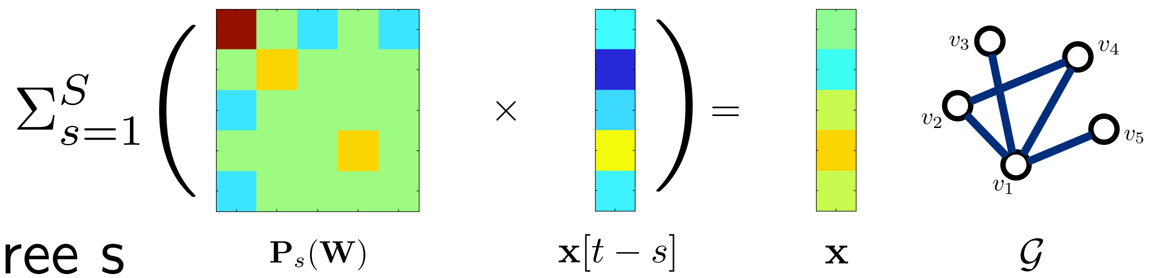
“graph-centric”: cost on graph operators instead of signals

Model 3: Causal dependency on graphs

- Signals are causal outcome of current or past observations (spectral characteristics depending on dependence structure)
- Example: evolution of individual behaviour due to influence of different friends at different timestamps
- Characterised by vector autoregressive models (VARMs) or structural equation models (SEMs)
  - VARMs exploits relation between present and past
  - SEMs exploits relation between vertices at present

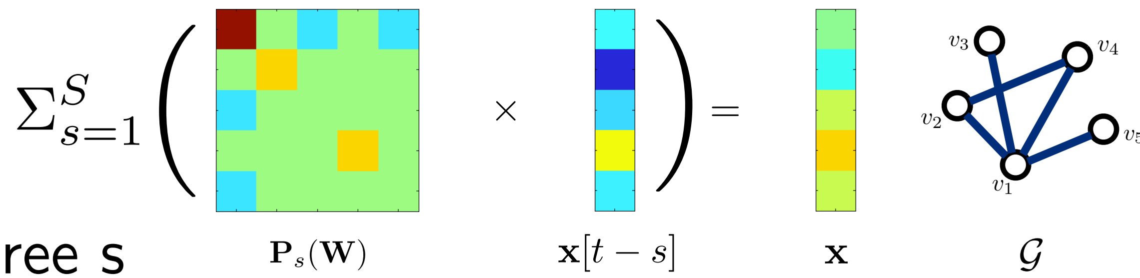
Model 3: Causal dependency on graphs

- Mei and Moura (2017)
  - $\mathcal{F}_s(\mathcal{G}) = \mathbf{P}_s(\mathbf{W})$: polynomial of **W** of degree s
  - define $\mathbf{c}_s$ as $\mathbf{x}[t-s]$

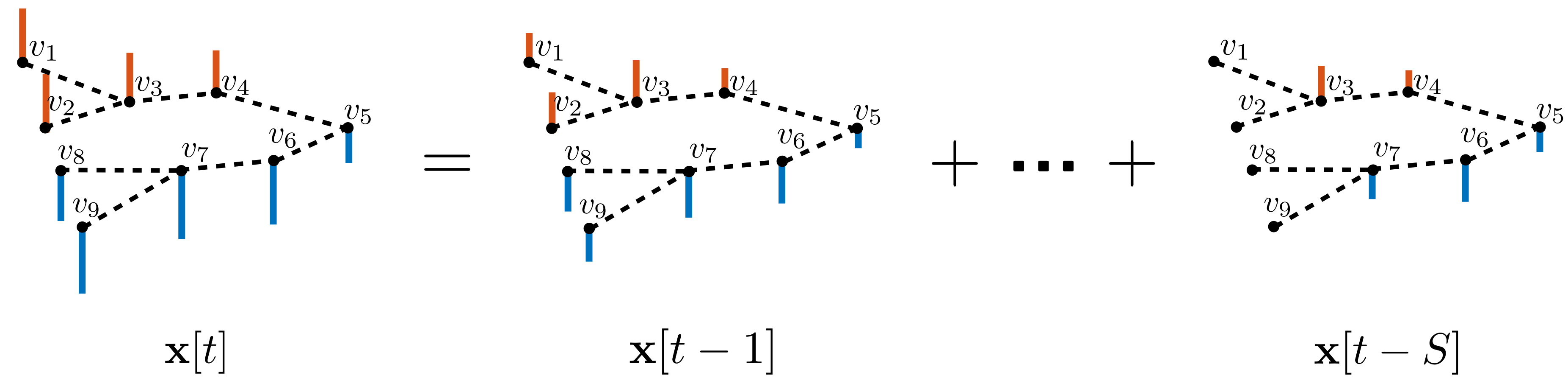


Model 3: Causal dependency on graphs

- Mei and Moura (2017)

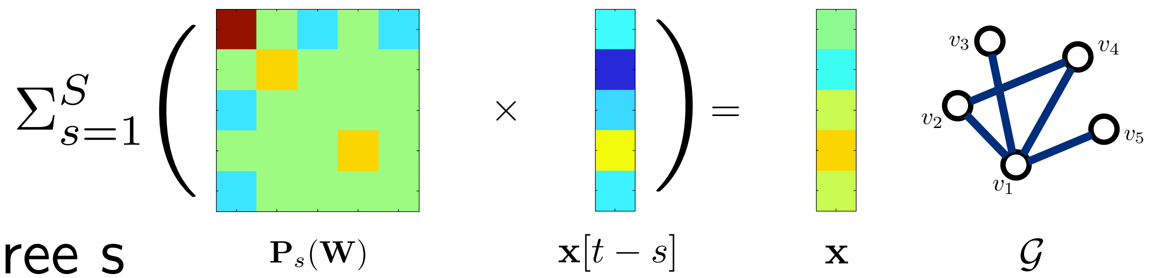


  - $\mathcal{F}_s(\mathcal{G}) = \mathbf{P}_s(\mathbf{W})$: polynomial of **W** of degree s
  - define $\mathbf{c}_s$ as $\mathbf{x}[t-s]$

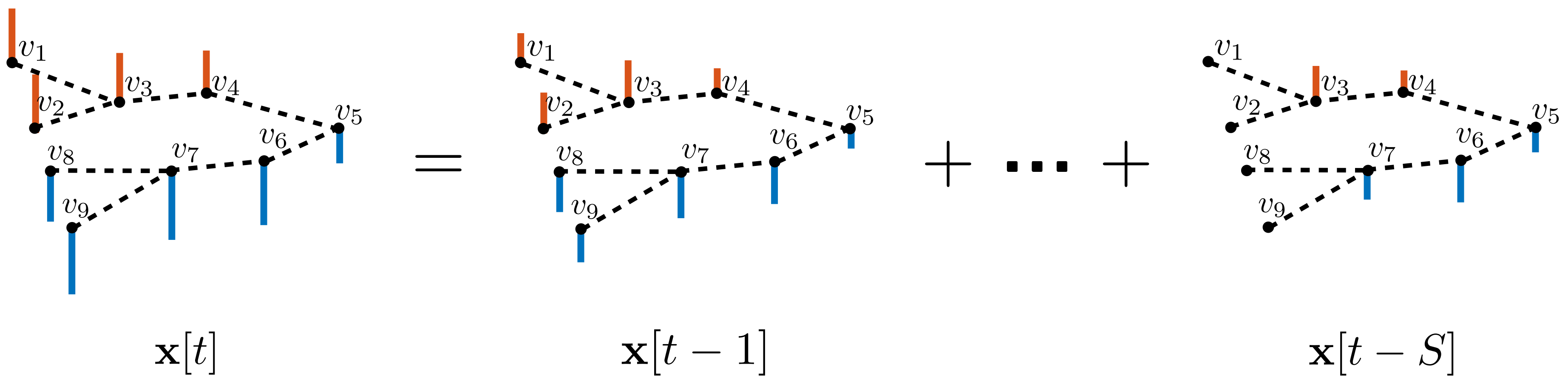


Model 3: Causal dependency on graphs

- Mei and Moura (2017)



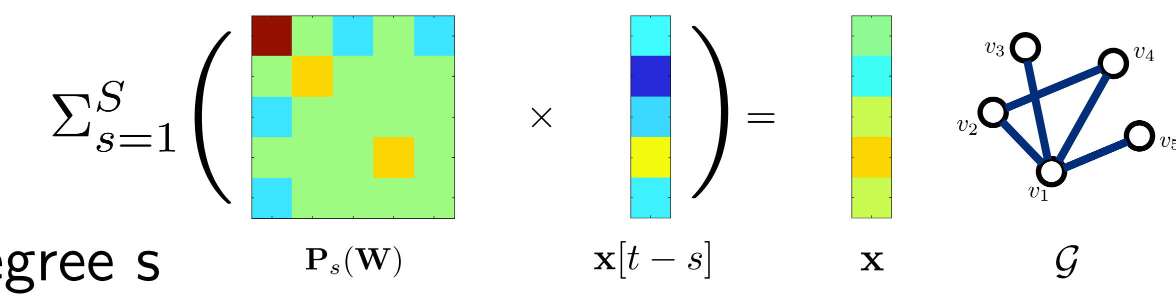
  - $\mathcal{F}_s(\mathcal{G}) = \mathbf{P}_s(\mathbf{W})$: polynomial of **W** of degree s
  - define $\mathbf{c}_s$ as $\mathbf{x}[t-s]$

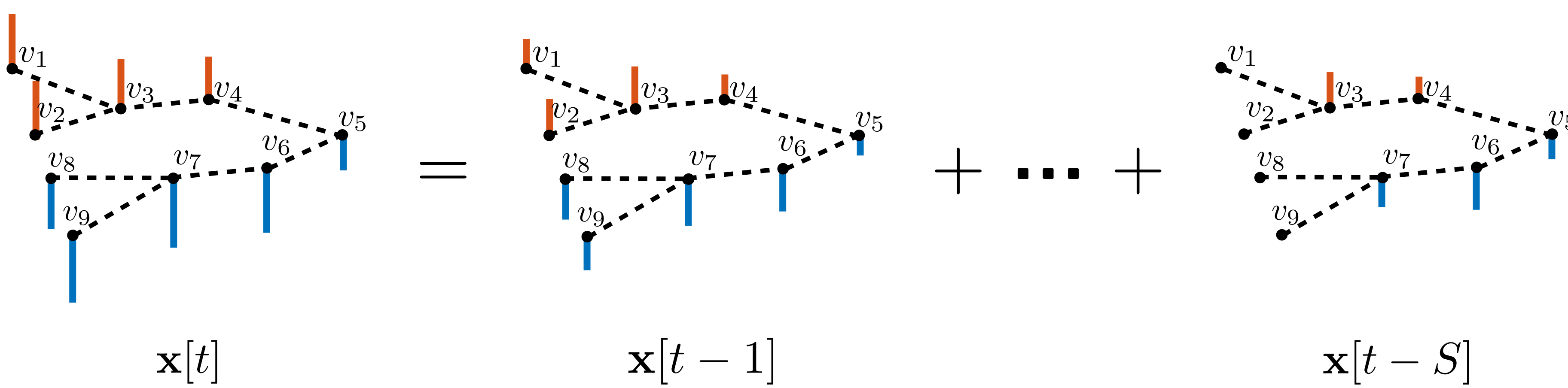


$$\min_{\mathbf{W},\mathbf{a}} \frac{1}{2} \sum_{k=S+1}^{K} ||\mathbf{x}[k] - \sum_{s=1}^{S} \mathbf{P}_s(\mathbf{W})\mathbf{x}[k-s]||_2^2 + \lambda_1 ||\text{vec}(\mathbf{W})||_1 + \lambda_2 ||\mathbf{a}||_1$$

Model 3: Causal dependency on graphs

- Mei and Moura (2017)
  - $\mathcal{F}_s(\mathcal{G}) = \mathbf{P}_s(\mathbf{W})$: polynomial of **W** of degree s
  - define $\mathbf{c}_s$ as $\mathbf{x}[t-s]$



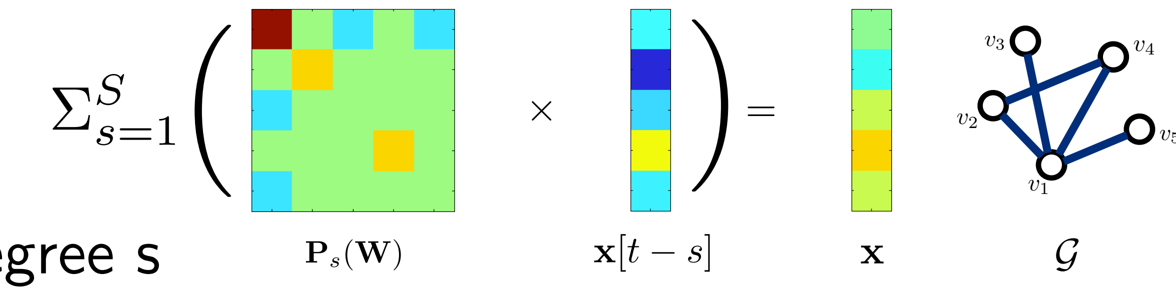


$$\min_{\mathbf{W},\mathbf{a}} \frac{1}{2} \sum_{k=S+1}^{K} ||\mathbf{x}[k] - \sum_{s=1}^{S} \mathbf{P}_s(\mathbf{W})\mathbf{x}[k-s]||_2^2 + \lambda_1 ||\mathrm{vec}(\mathbf{W})||_1 + \lambda_2 ||\mathbf{a}||_1$$

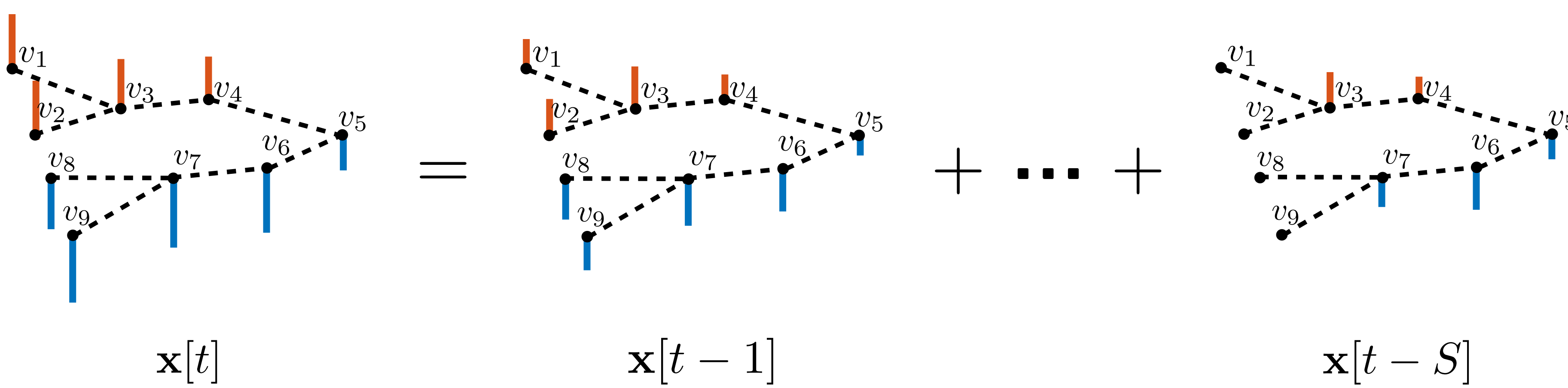
data fidelity — sparsity on **W** — sparsity on **a**

Model 3: Causal dependency on graphs

- Mei and Moura (2017)



  - $\mathcal{F}_s(\mathcal{G}) = \mathbf{P}_s(\mathbf{W})$: polynomial of **W** of degree s
  - define $\mathbf{c}_s$ as $\mathbf{x}[t-s]$



$$\min_{\mathbf{W},\mathbf{a}} \frac{1}{2} \sum_{k=S+1}^{K} ||\mathbf{x}[k] - \sum_{s=1}^{S} \mathbf{P}_s(\mathbf{W})\mathbf{x}[k-s]||_2^2 + \lambda_1 ||\text{vec}(\mathbf{W})||_1 + \lambda_2 ||\mathbf{a}||_1$$

data fidelity — sparsity on **W** — sparsity on **a**

good for inferring causal relations between signals

can be combined with SEMs and kernelised

Comparison of different GSP methods

Table 1. Comparisons between different GSP-based approaches to graph learning.

Method	Signal Model	Assumption		Learning Output	Edge Directionality
		$\mathcal{F}(\mathcal{G})$	c		
Dong et al. [39]	Global smoothness	Eigenvector matrix	i.i.d. Gaussian	Laplacian	Undirected
Kalofolias et al. [40]	Global smoothness	Eigenvector matrix	i.i.d. Gaussian	Adjacency matrix	Undirected
Egilmez et al. [41]	Global smoothness	Eigenvector matrix	i.i.d. Gaussian	Generalized Laplacian	Undirected
Chepuri et al. [42]	Global smoothness	Eigenvector matrix	i.i.d. Gaussian	Adjacency matrix	Undirected
Pasdeloup et al. [46]	Spectral filtering (diffusion by adjacency)	Normalized adjacency matrix	i.i.d. Gaussian	Normalized adjacency matrix normalized Laplacian	Undirected
Segarra et al. [45]	Spectral filtering (diffusion by graph shift operator)	Graph shift operator	i.i.d. Gaussian	Graph shift operator	Undirected
Thanou et al. [47]	Spectral filtering (heat diffusion)	Heat kernel	Sparsity	Laplacian	Undirected
Mei and Moura [55]	Causal dependency (SVAR)	Polynomials of adjacency matrix	Past signals	Adjacency matrix	Directed
Baingana et al. [62]	Causal dependency (SEM)	Adjacency matrix	Present signal	Time-varying adjacency matrix	Directed
Shen et al. [54]	Causal dependency (SVARM)	Polynomials of adjacency matrix	Past and present signals	Adjacency matrix	Directed

[Dong19]

Connection with broad literature

- Global smoothness of graph signals is also promoted in Graphical Lasso

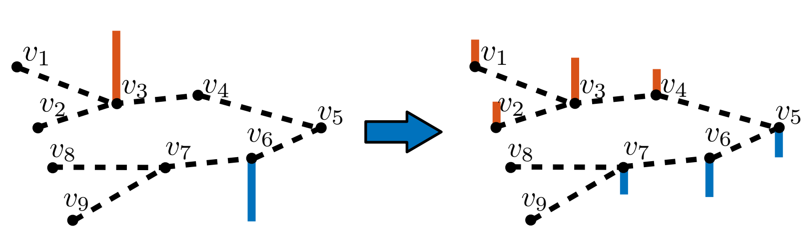
Lake (2010): $$\max_{\mathbf{\Theta}=\mathbf{L}+\frac{1}{\sigma^2}\mathbf{I}} \log\ \det\mathbf{\Theta} - \frac{1}{M}\mathrm{tr}(\mathbf{X}\mathbf{X}^T\mathbf{\Theta}) - \rho||\mathbf{\Theta}||_1$$

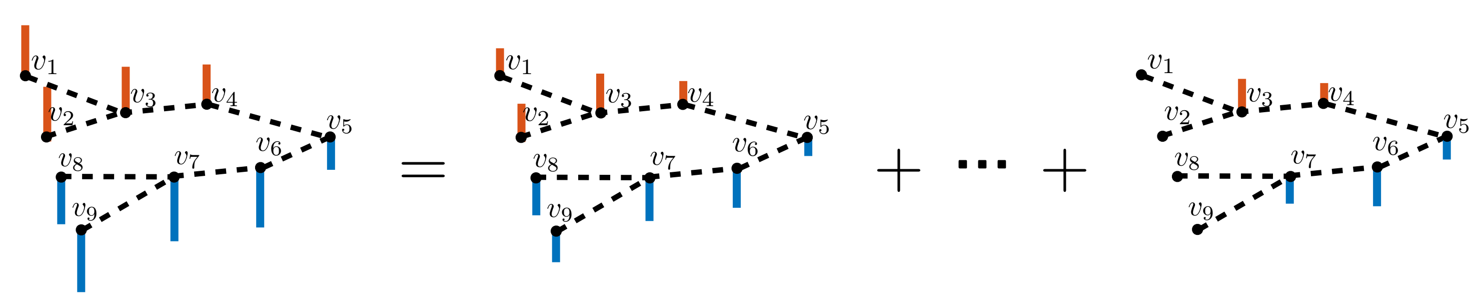
Connection with broad literature

- Global smoothness of graph signals is also promoted in Graphical Lasso

Lake (2010): $$\max_{\mathbf{\Theta}=\mathbf{L}+\frac{1}{\sigma^2}\mathbf{I}} \log \det\mathbf{\Theta} - \frac{1}{M}\mathrm{tr}(\mathbf{X}\mathbf{X}^T\mathbf{\Theta}) - \rho||\mathbf{\Theta}||_1$$

- Models based on spectral filtering or causal dependency lead to generative process of signals, similarly to traditional physically motivated models



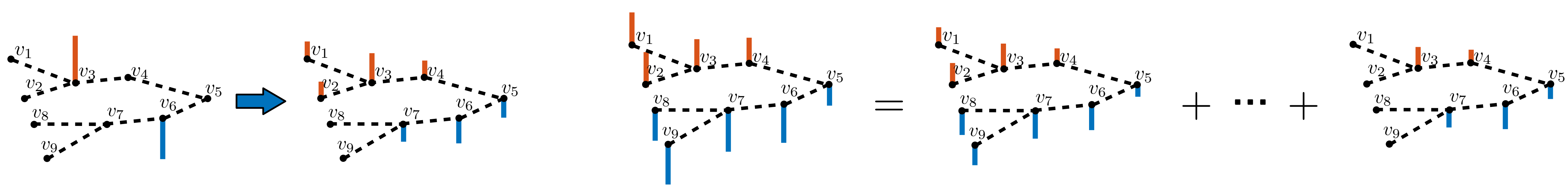


Connection with broad literature

- Global smoothness of graph signals is also promoted in Graphical Lasso

Lake (2010): $$\max_{\boldsymbol{\Theta}=\mathbf{L}+\frac{1}{\sigma^2}\mathbf{I}} \log \det \boldsymbol{\Theta} - \frac{1}{M}\text{tr}(\mathbf{X}\mathbf{X}^T\boldsymbol{\Theta}) - \rho||\boldsymbol{\Theta}||_1$$

- Models based on spectral filtering or causal dependency lead to generative process of signals, similarly to traditional physically motivated models



- GSP approaches offer design flexibility (via **F** and **c**) and extend beyond a Gaussian statistical model or a simple diffusion model

Applications

- Image coding and compression (review of [Chung18])
  - images are natural graph signals on regular grid
  - learning adaptive edge weights for structure-aware transform coding
  - more efficient image compression

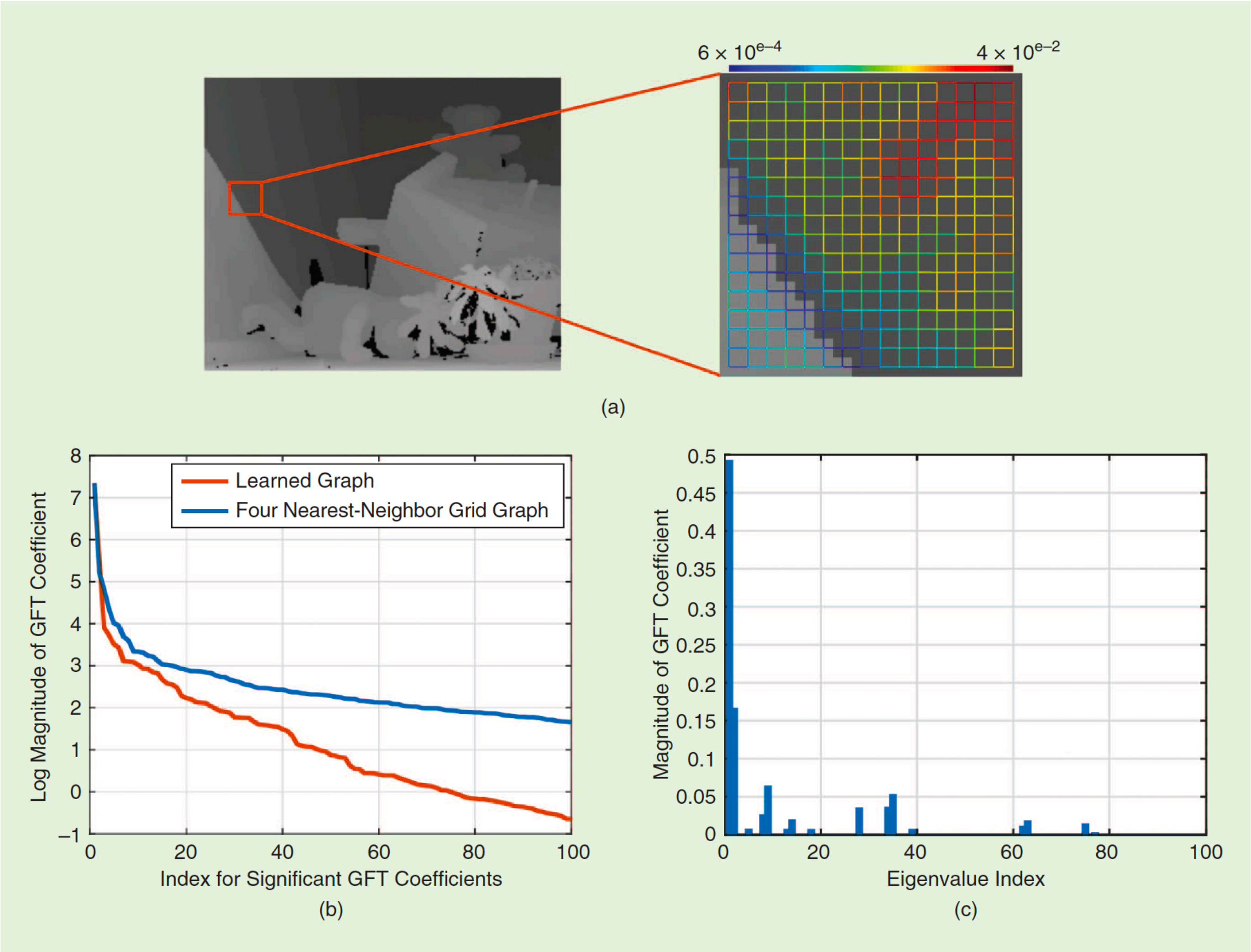
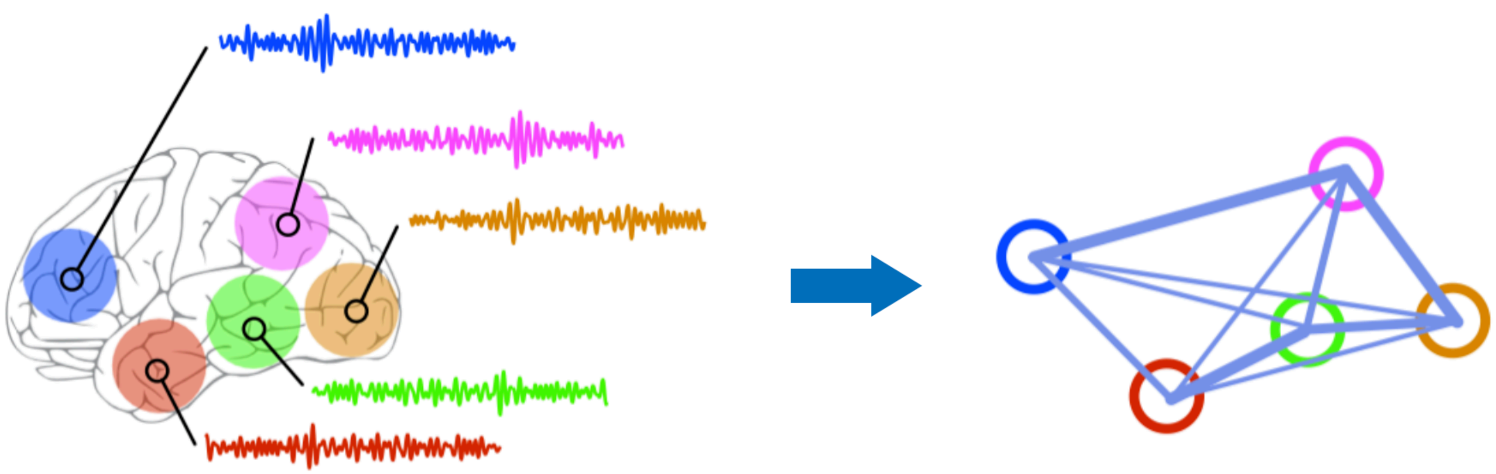


FIGURE 10. Inferring a graph for image coding. (a) The graph learned on a random patch of the image Teddy using [69]. (b) A comparison between the GFT coefficients of the image signal on the learned graph and the four nearest-neighbor grid graph. The coefficients are ordered decreasingly by log magnitude. (c) The GFT coefficients of the graph weights.

[Fracastoro17]

Applications

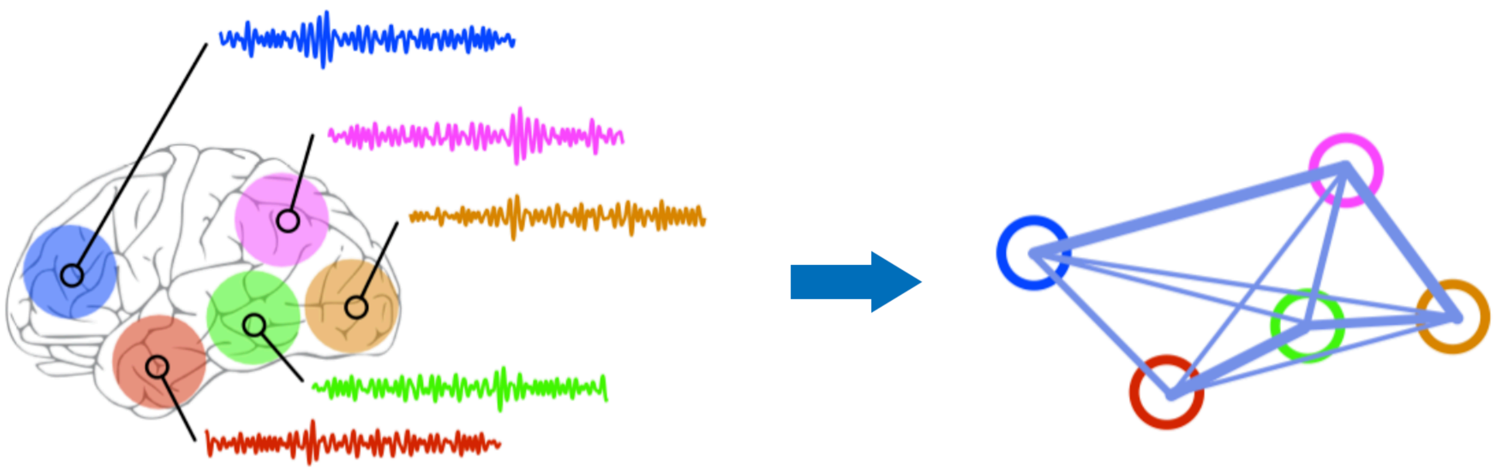
- Brain signal analysis (review of [Huang18])
  - learning functional connectivity of brain regions



[Richiardi13]

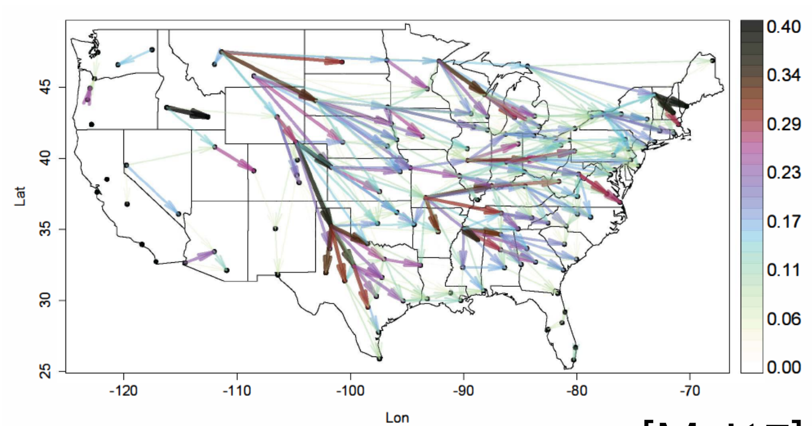
Applications

- Brain signal analysis (review of [Huang18])
  - learning functional connectivity of brain regions



[Richiardi13]

- Other application domains
  - learning meteorology graph using temperatures
  - learning commuting graph using traffic volume
  - learning political relations using voting data

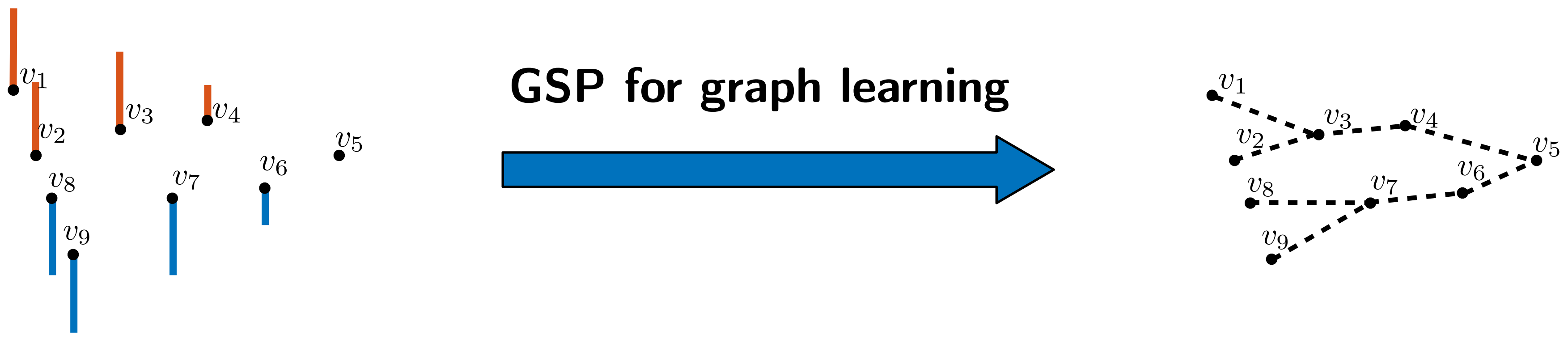


[Mei17]

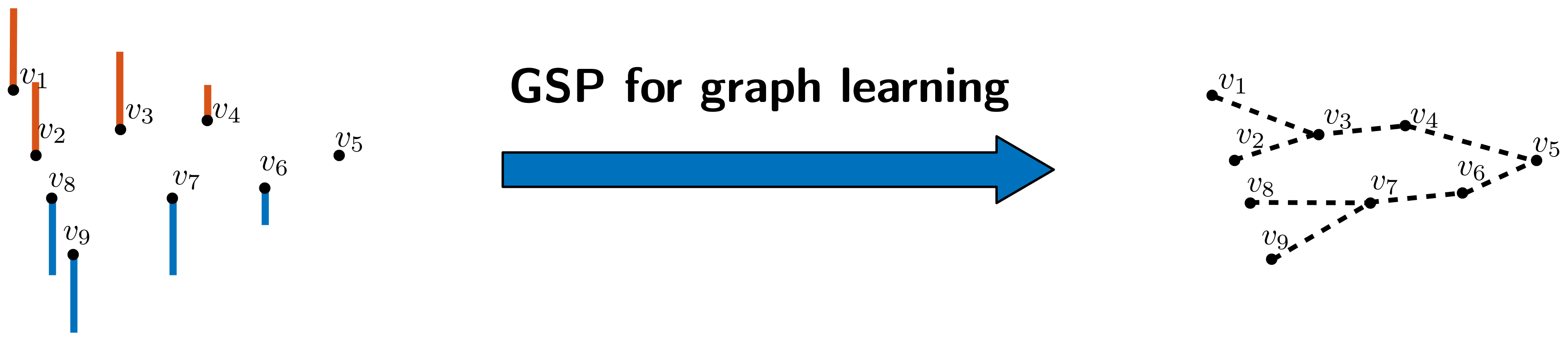
Outline

- A (very partial) literature overview
- A signal processing perspective
  - A brief introduction to graph signal processing (GSP)
  - GSP approaches for graph learning
- Concluding remarks

Concluding remarks



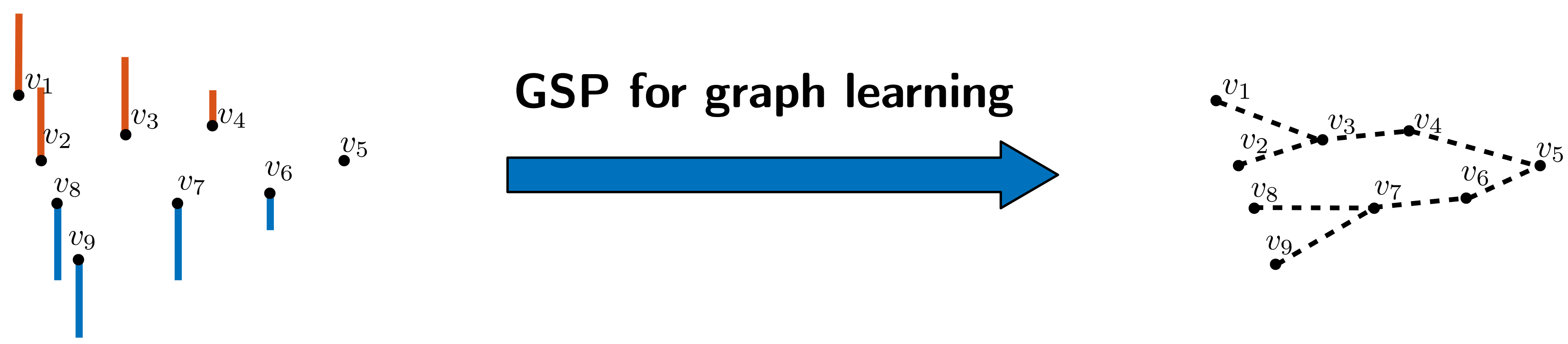
Concluding remarks



input signals

- **partial observations**
- **sequential observations**

Concluding remarks



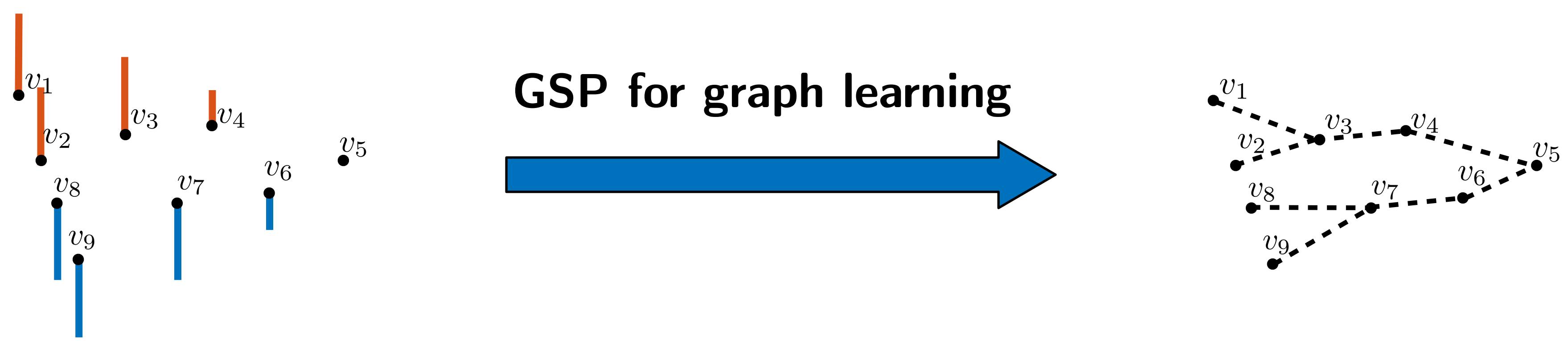
input signals

- **partial observations**
- **sequential observations**

learning outcome

- **directed graphs**
- **time-varying (dynamic) graphs**
- **graphs with certain properties**
- **intermediate graph representation**
- **uncertainly in learned structure**

Concluding remarks



input signals

- **partial observations**
- **sequential observations**

signal/graph model

- **beyond smoothness: localisation in vertex-frequency domain**
- **adapt to specific input/output**

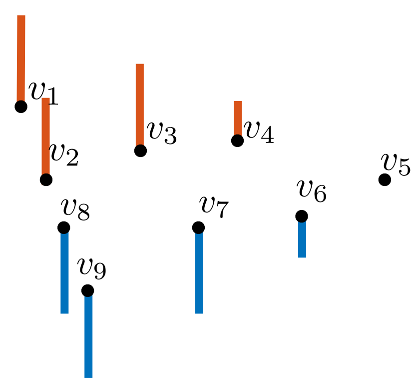
learning outcome

- **directed graphs**
- **time-varying (dynamic) graphs**
- **graphs with certain properties**
- **intermediate graph representation**
- **uncertainly in learned structure**

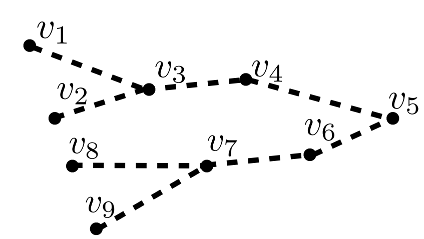
Concluding remarks

theoretical consideration

- performance guarantee
- computational efficiency



GSP for graph learning →



input signals

- partial observations
- sequential observations

signal/graph model

- beyond smoothness: localisation in vertex-frequency domain
- adapt to specific input/output

learning outcome

- directed graphs
- time-varying (dynamic) graphs
- graphs with certain properties
- intermediate graph representation
- uncertainly in learned structure

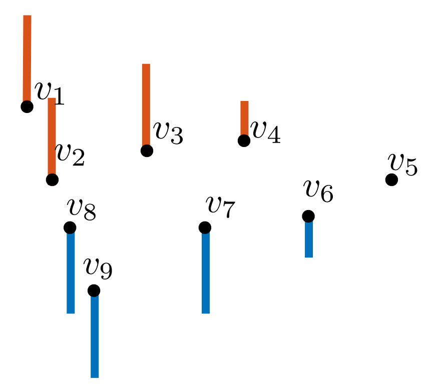
Concluding remarks

theoretical consideration

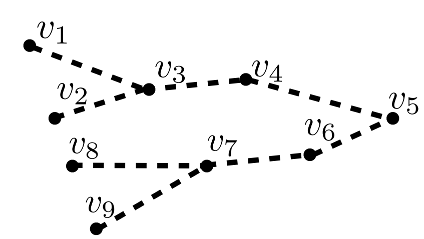
- performance guarantee
- computational efficiency

objective of graph learning

- for traditional graph-based learning, e.g., clustering, dim. reduction, ranking
- integrate inference with subsequent data analysis (targeted applications)



GSP for graph learning



input signals

- partial observations
- sequential observations

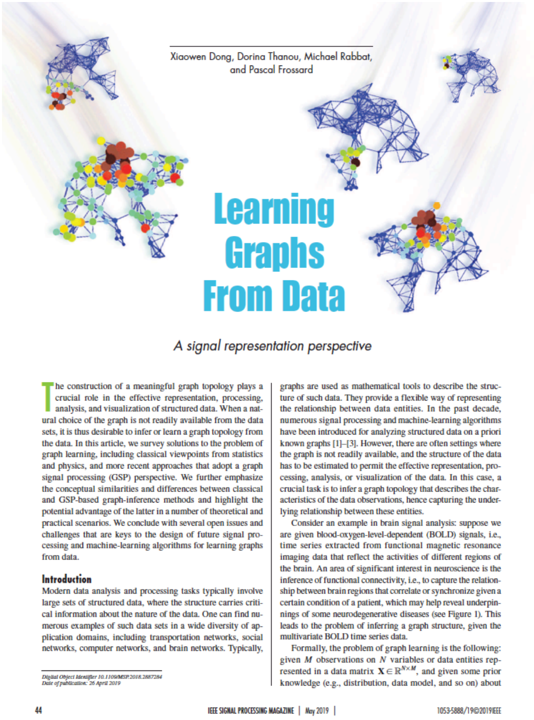
signal/graph model

- beyond smoothness: localisation in vertex-frequency domain
- adapt to specific input/output

learning outcome

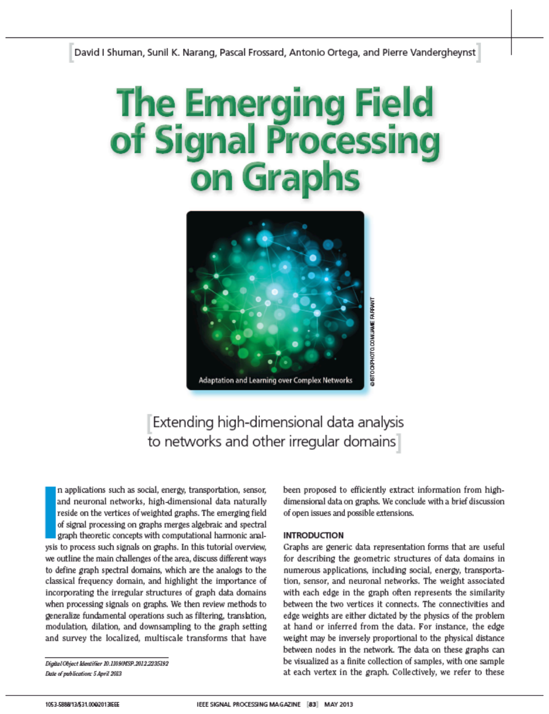
- directed graphs
- time-varying (dynamic) graphs
- graphs with certain properties
- intermediate graph representation
- uncertainly in learned structure

Papers & Resources & Acknowledgement


Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard

Learning Graphs From Data

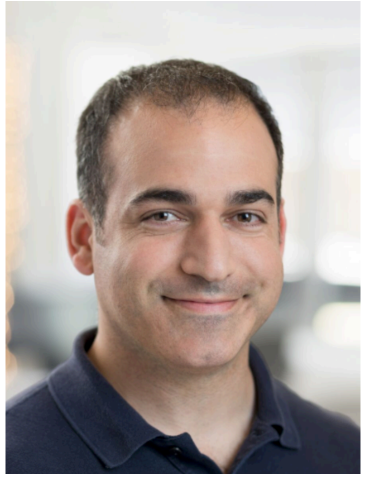
A signal representation perspective


David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst

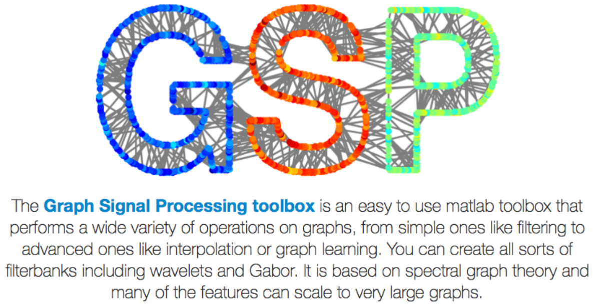
The Emerging Field of Signal Processing on Graphs

Extending high-dimensional data analysis to networks and other irregular domains

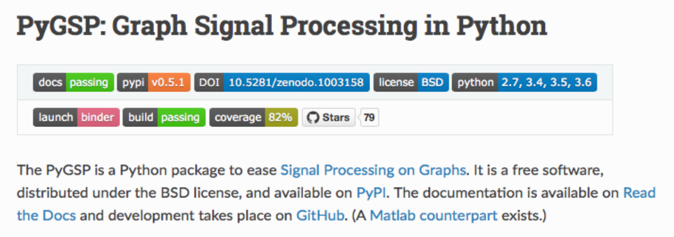

Dorina Thanou


Mike Rabbat


Pascal Frossard



https://epfl-lts2.github.io/gspbox-html/



https://pygsp.readthedocs.io/en/stable/

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

Thank you!