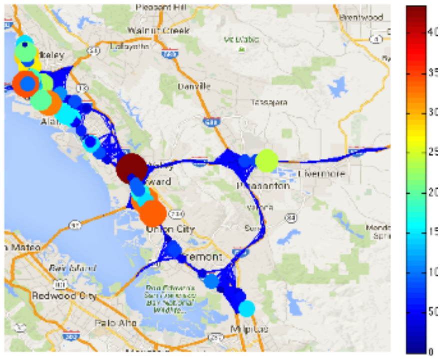
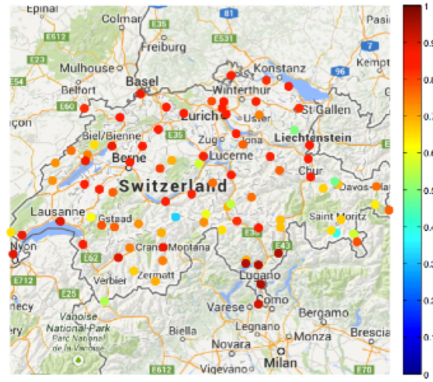
LEARNING GRAPHS FROM DATA: A GENERAL OVERVIEW

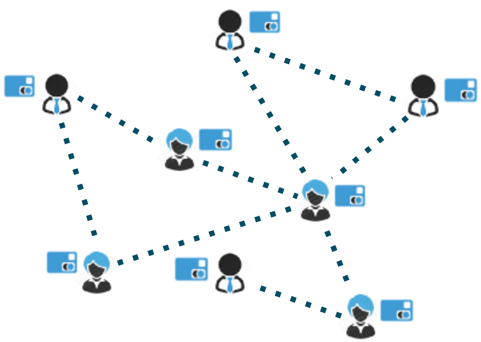
Dorina Thanou

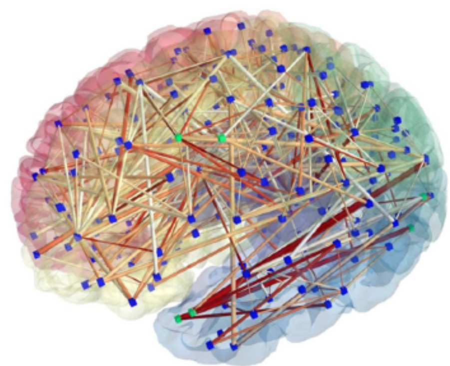
Swiss Data Science Center (SDSC), EPFL/ETH Zurich

Data is often structured



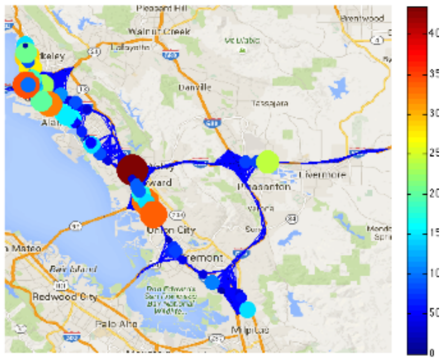


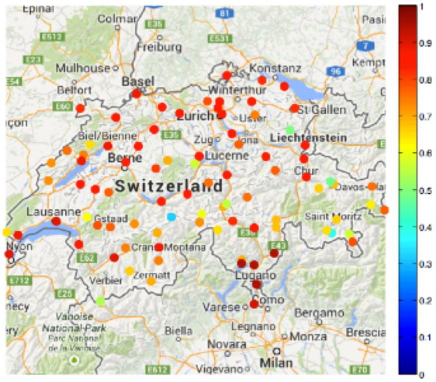


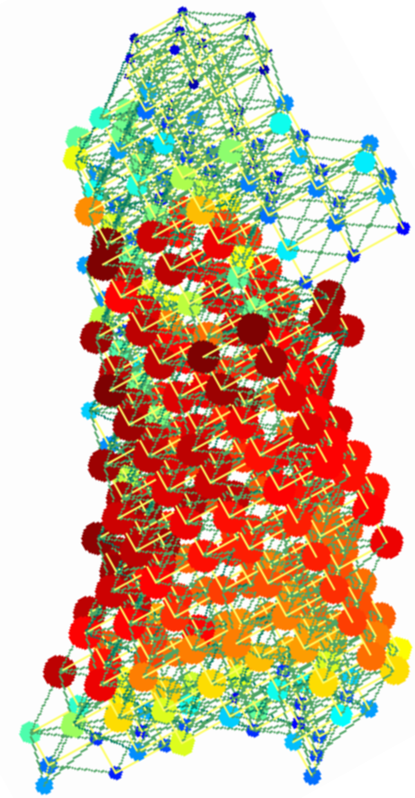


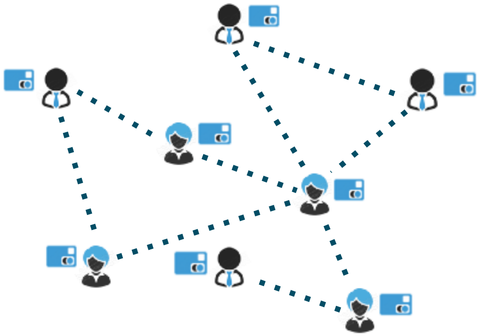
Graphs: flexible tools to represent the geometric structure of signals defined on irregular domains

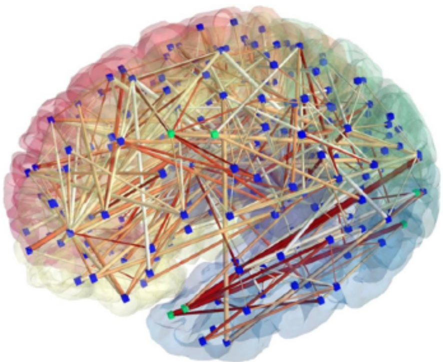
Data is often structured







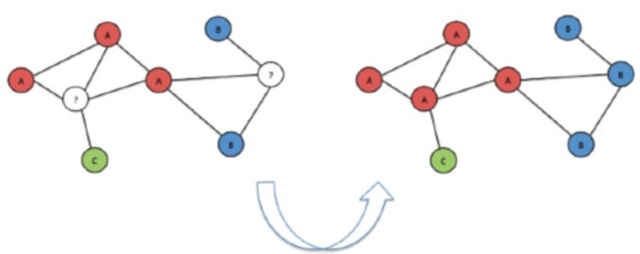




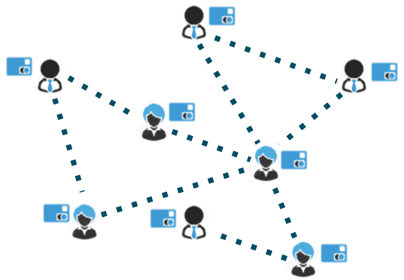
Graphs: flexible tools to represent the geometric structure of signals defined on irregular domains

The importance of the graph

- Graph signal models define the interplay between signals and structure
  - Such models are used for **effective data processing or understanding**
  - Useful for **future prediction/decision making**



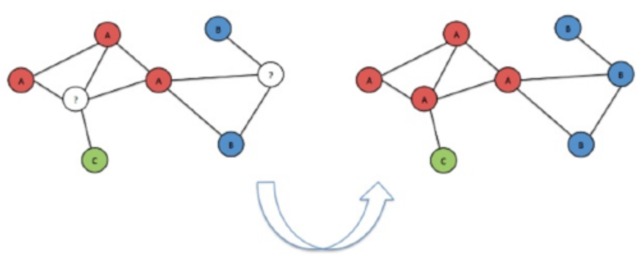
Objective: Inference of missing values



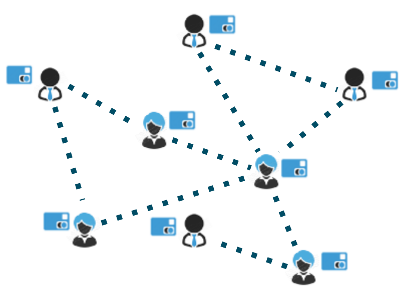
Objective: behavioural similarity/ influence between people

The importance of the graph

- Graph signal models define the interplay between signals and structure
  - Such models are used for **effective data processing or understanding**
  - Useful for **future prediction/decision making**



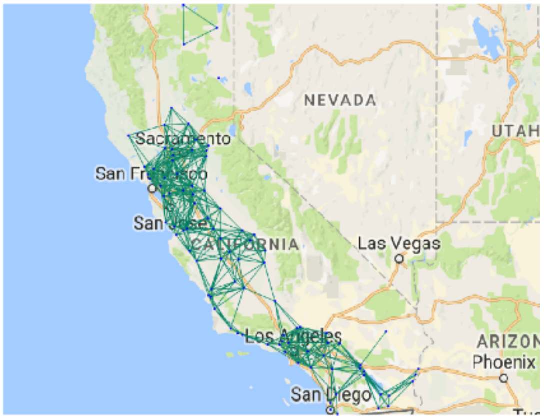
Objective: Inference of missing values



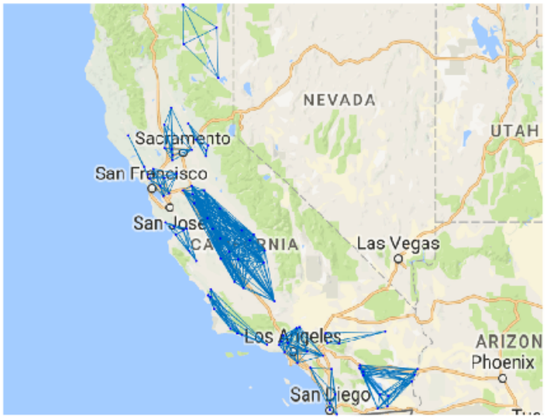
Objective: behavioural similarity/ influence between people

- The graph might (often) not be known a priori
- It becomes important to infer the proper structure

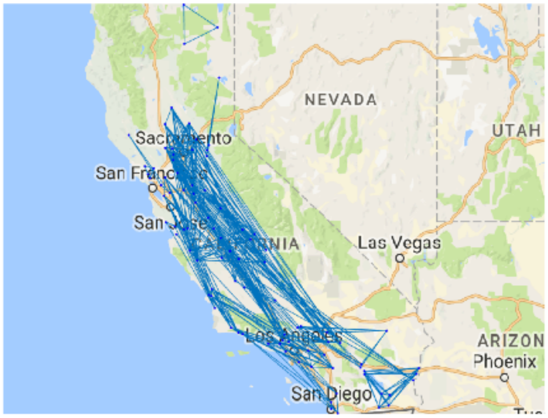
A simple example



Distance based

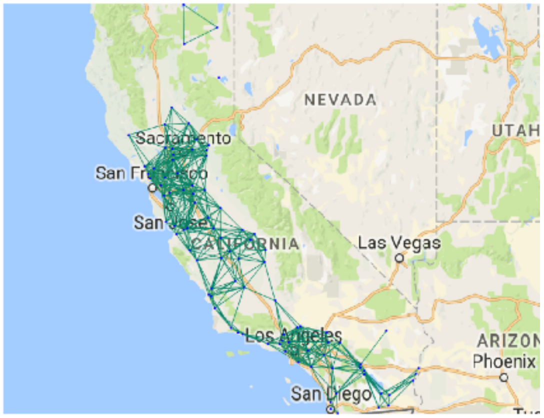


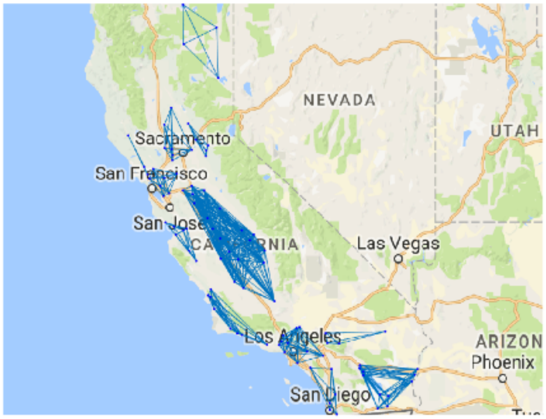
Region based

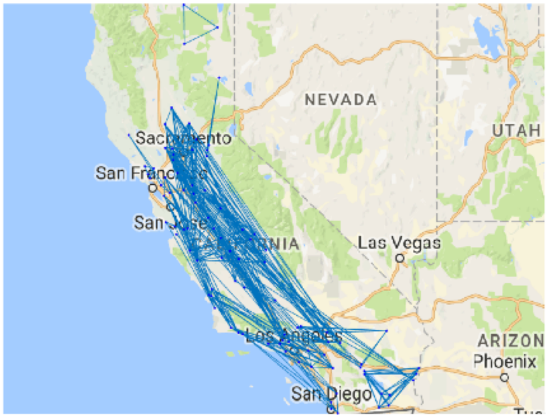


Evapotranspiration zone based

A simple example



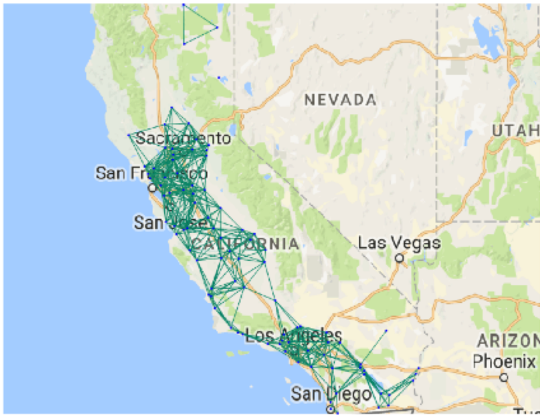




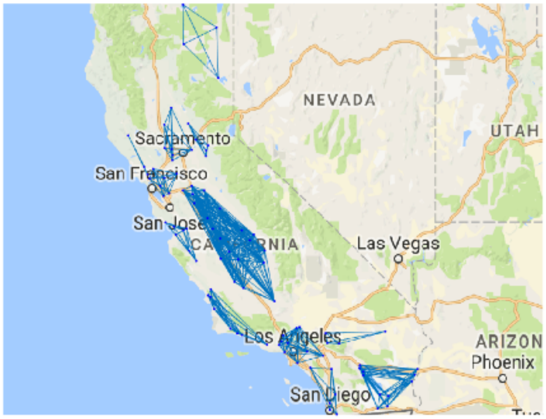
Distance based **Region based** **Evapotranspiration zone based**

- What is the structure of my data?

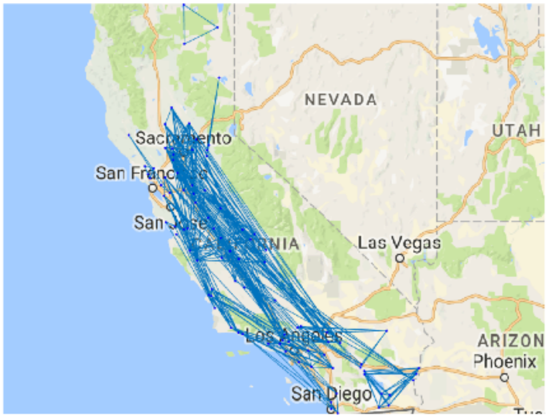
A simple example



Distance based



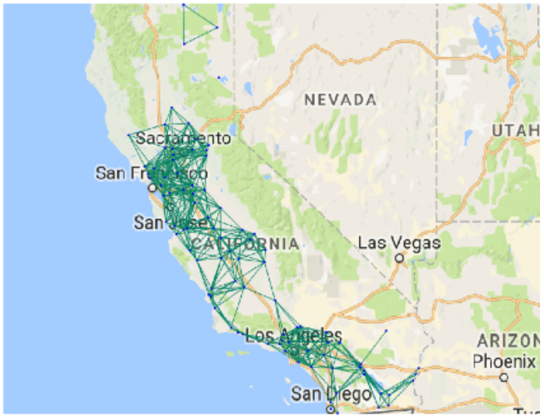
Region based



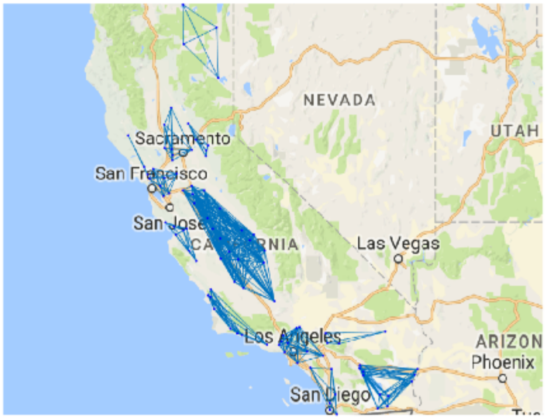
Evapotranspiration zone based

- What is the structure of my data?
- Can we use observations (e.g., temperature, humidity) to infer the right graph?

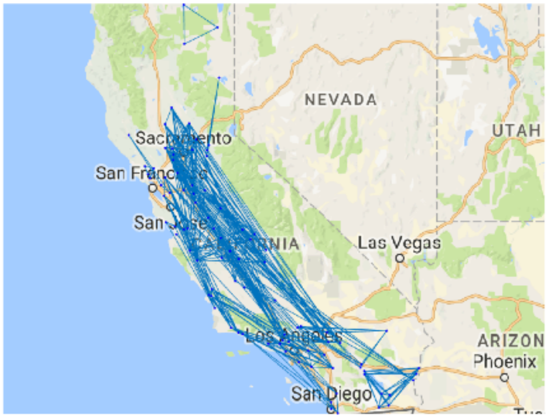
A simple example



Distance based



Region based

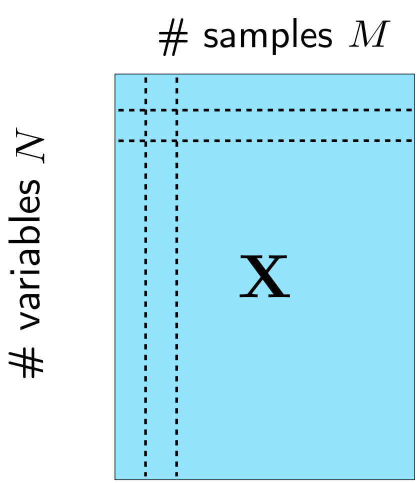


Evapotranspiration zone based

- What is the structure of my data?
- Can we use observations (e.g., temperature, humidity) to infer the right graph?
- What info about the data should the 'optimal' graph reveal?

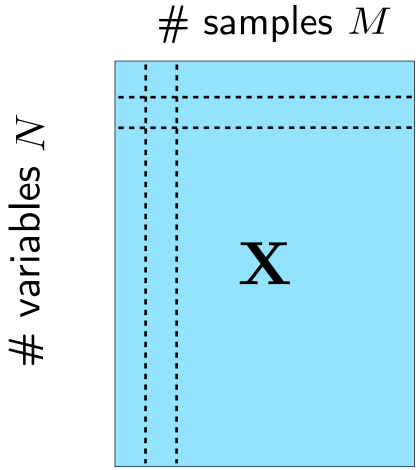
The topology inference problem

- **Different terminologies** in the literature:
  - covariance estimation
  - learning graphical models
  - network inference/graph learning
  - …



The topology inference problem

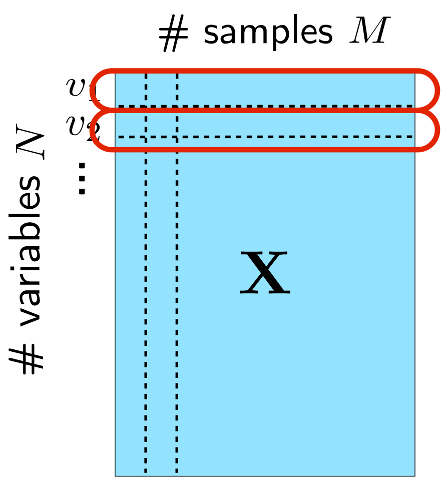
- **Different terminologies** in the literature:
  - covariance estimation
  - learning graphical models
  - network inference/graph learning
  - …
- **Common goal:** Discover structure in the data by building a measure of relations between variables

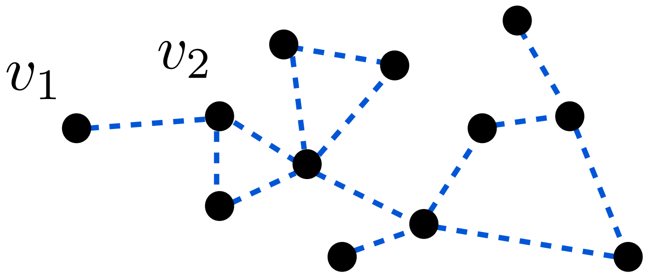


SDSC

The topology inference problem

- **Different terminologies** in the literature:
  - covariance estimation
  - learning graphical models
  - network inference/graph learning
  - …
- **Common goal:** Discover structure in the data by building a measure of relations between variables





How do we build or learn a graph?

Topology inference methods

- **Model-free:**
  - do not assume any mechanism to generate the data
  - usually use statistical measures
- **Model-based:**
  - assume a mathematical model to generate the data/ signals
  - determine the model parameters to fit observed data

Model-free methods

- **Sample correlation/Pearson correlation:**

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}$$

Rate of concurring values in two nodes

Model-free methods

- **Sample correlation/Pearson correlation:**

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}$$

Rate of concurring values in two nodes

- **K-NN graph:** Connect each node to its K nearest neighbors

Model-free methods

- **Sample correlation/Pearson correlation:**

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}$$

Rate of concurring values in two nodes

- **K-NN graph:** Connect each node to its K nearest neighbors

- **Kernel-based graph:** Define a similarity measure based on a graph kernel (e.g., Gaussian)

$$W_{i,j} = \begin{cases} exp\left(-\frac{[dist(i,j)]^2}{2\theta^2}\right), & \text{if } dist(i,j) \leq \kappa \\ 0, & \text{otherwise.} \end{cases}$$

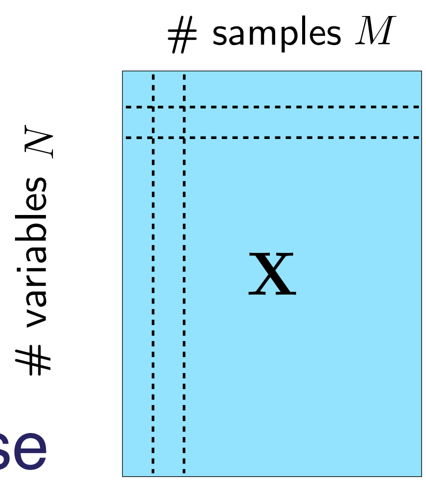
Model-free methods

✓ Computationally simple
✓ Intuitive

- Usually requires the definition of a good distance measure
- Need for parameter tuning
- Address pre/post-processing separately from connectivity inference
- Sensitive to noise

Model-based: Formal definition

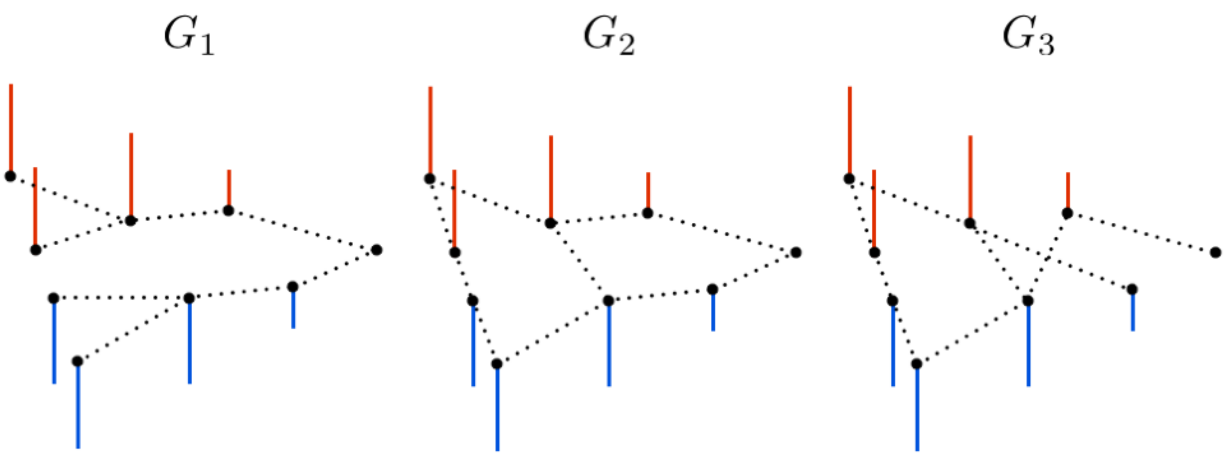
- **Given:**
  - M observations on N variables
  - some prior knowledge (e.g., distribution, data model, etc) about the data
- **Goal:**
  - Build or infer relationships between these variables that take the form of a graph



$$X = \mathcal{F}(G)$$

Model-based methods

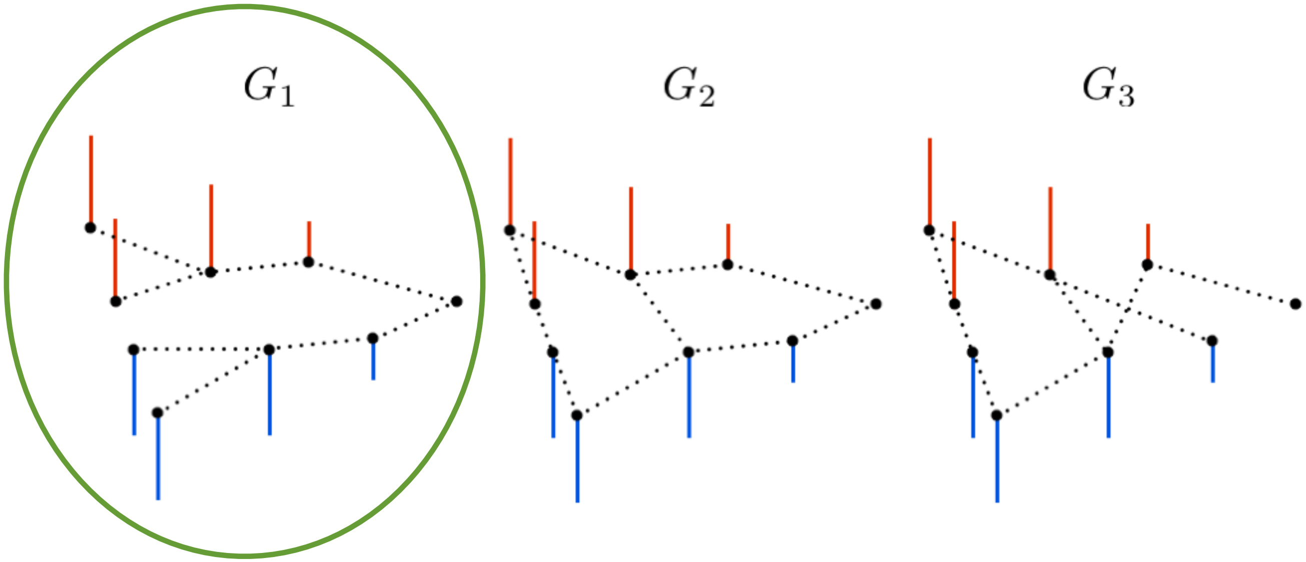
- ✓ Easier to incorporate priors
- ✓ More robust to noise



Model-based methods

✓ Easier to incorporate priors

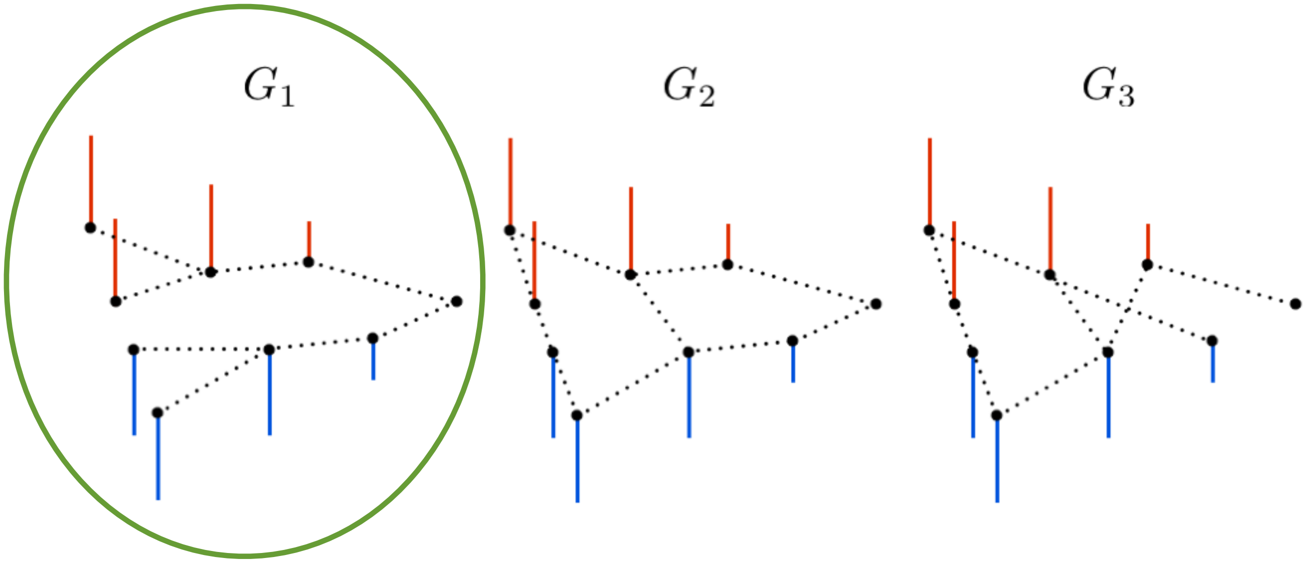
✓ More robust to noise



Model-based methods

✓ Easier to incorporate priors

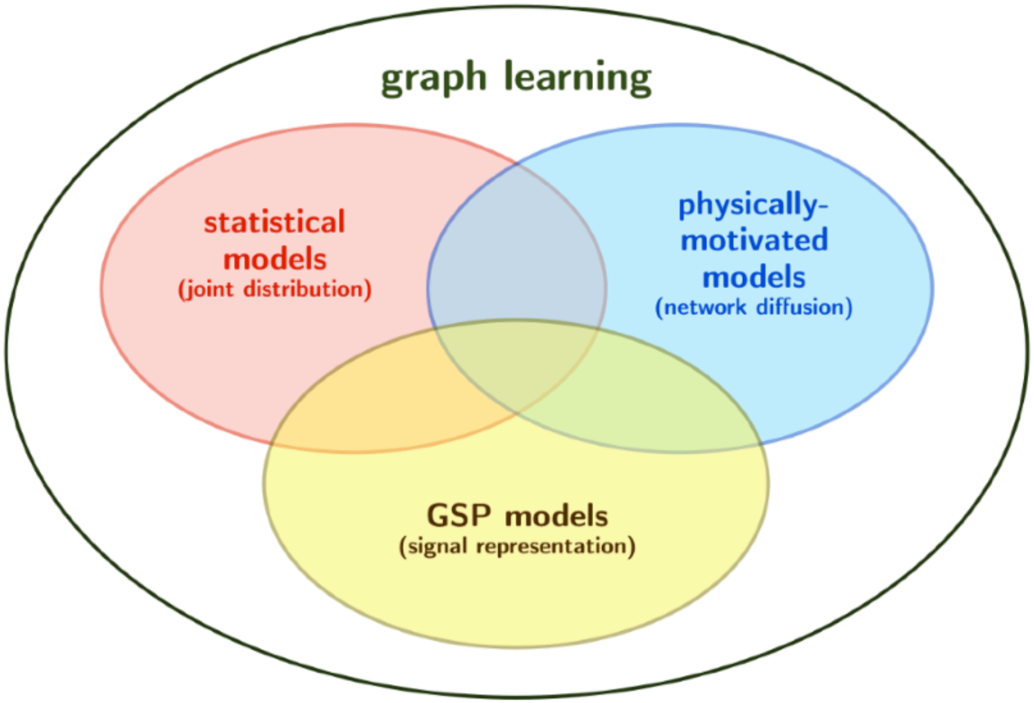
✓ More robust to noise



- Scalability issues: The solution of a complex optimization problem is usually required

Model-based methods in a nutshell

- Different approaches…



- … with however some similar assumptions

SDSC

Outline

- A (partial) historical overview of model-based methods
- A graph signal processing perspective
  - A quick reminder of GSP
  - GSP for topology inference
- Applications/Concluding remarks

SDSC

Outline

- **A (partial) historical overview of model-based methods**
- A graph signal processing perspective
  - A quick reminder of GSP
  - GSP for topology inference
- Applications/Concluding remarks

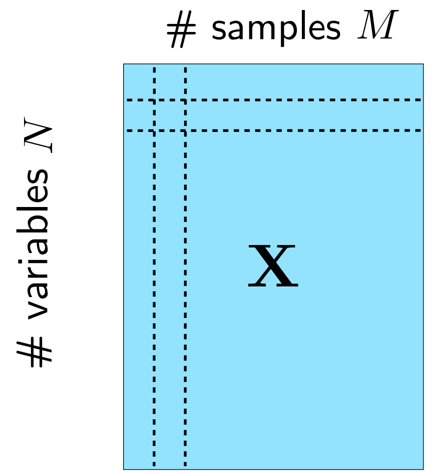
SDSC

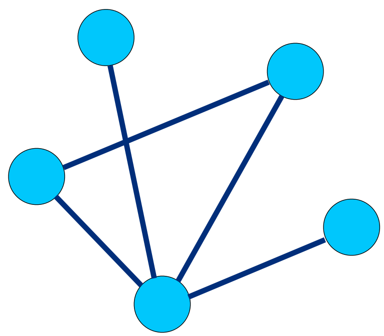
Historical overview

- Learning **graphical models**: probabilistic models for which the graph represent the conditional dependence between variables

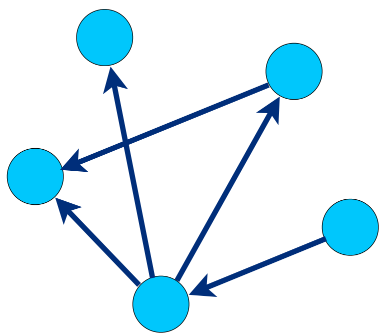
Historical overview

- Learning **graphical models**: probabilistic models for which the graph represent the conditional dependence between variables





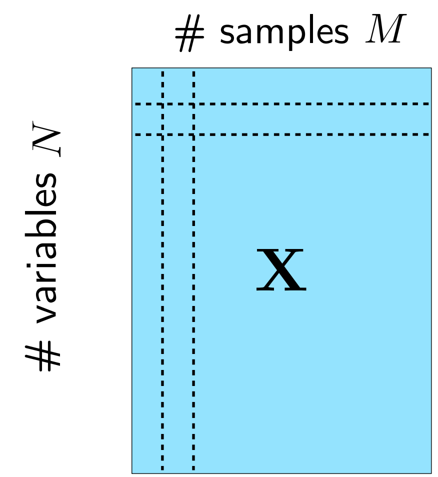
Undirected graphical models:
Markov random fields (MRF)

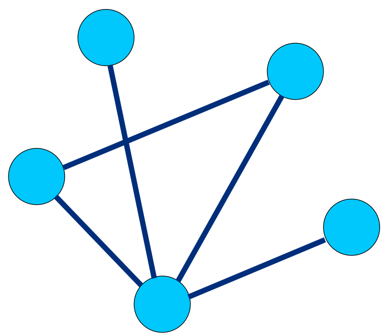


Directed graphical models:
Bayesian networks (BN)

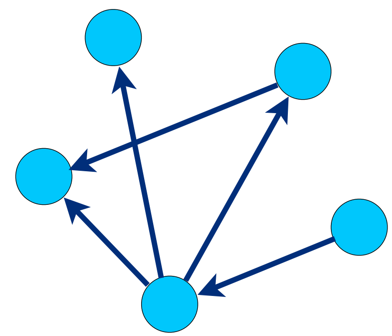
Historical overview

- Learning **graphical models**: probabilistic models for which the graph represent the conditional dependence between variables





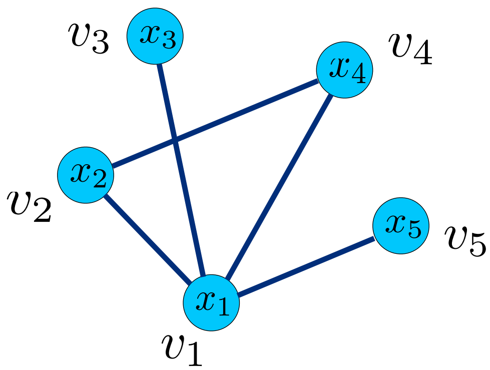
Undirected graphical models:
Markov random fields (MRF)



Directed graphical models:
Bayesian networks (BN)

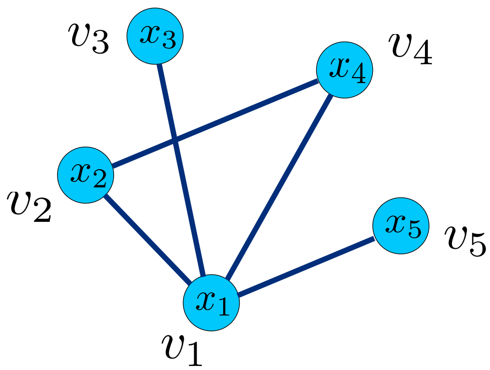
Historical overview

- Learning pairwise MRF



Historical overview

- Learning pairwise MRF

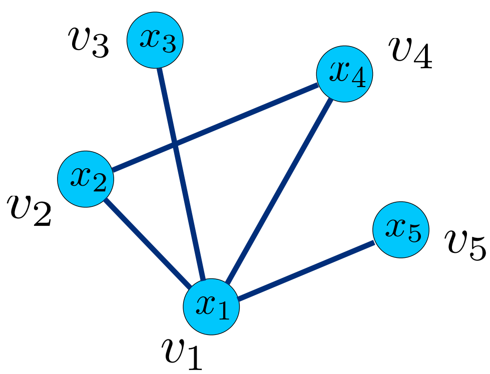


Conditional independence:

$$(i,j) \notin E \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

Historical overview

- Learning pairwise MRF



Conditional independence:

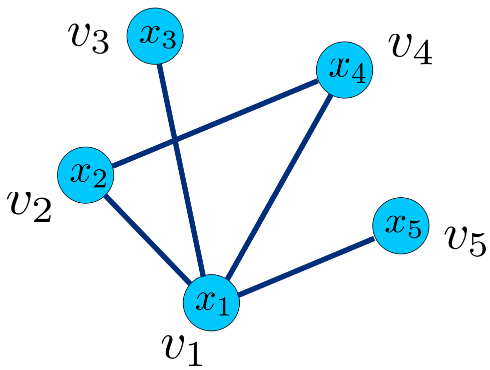
$$(i,j) \notin E \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

Probability parametrized by the precision matrix (inverse covariance) Θ:

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

Historical overview

- Learning pairwise MRF



Conditional independence:

$$(i,j) \notin E \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

Probability parametrized by the precision matrix (inverse covariance) Θ:

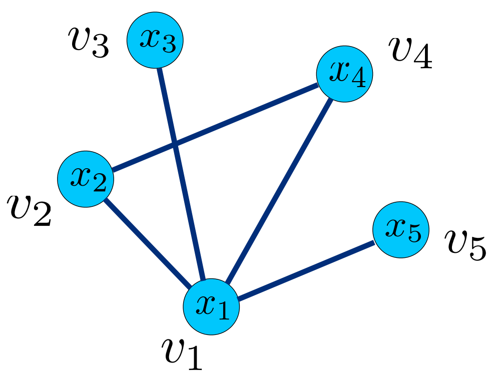
$$P(\mathbf{x}|\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} \exp\Big(\sum_{i \in V} \theta_{ii} x_i^2 + \sum_{(i,j) \in E} \theta_{ij} x_i x_j \Big)$$

Gaussian MRF with precision Θ:

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

Historical overview

- Learning pairwise MRF



Conditional independence:

$$(i,j) \notin E \Leftrightarrow x_i \perp x_j \mid \mathbf{x} \setminus \{x_i, x_j\}$$

Probability parametrized by the precision matrix (inverse covariance) Θ:

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

Gaussian MRF with precision Θ:

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

Learning a sparse Θ :

- **interactions are mostly local**
- **computationally more tractable**

Historical overview

Prune the smallest elements in precision (inverse covariance) matrix

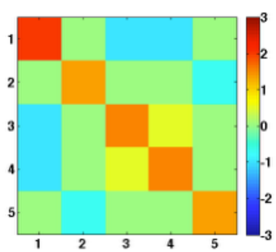
Historical overview

covariance selection

Dempster

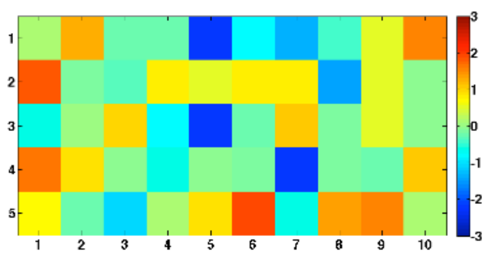
1972

Prune the smallest elements in precision (inverse covariance) matrix



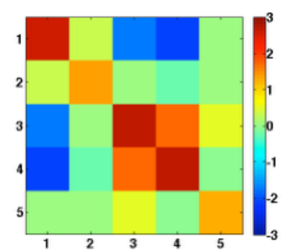
$\boldsymbol{\Theta}$

ground-truth precision



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

data matrix

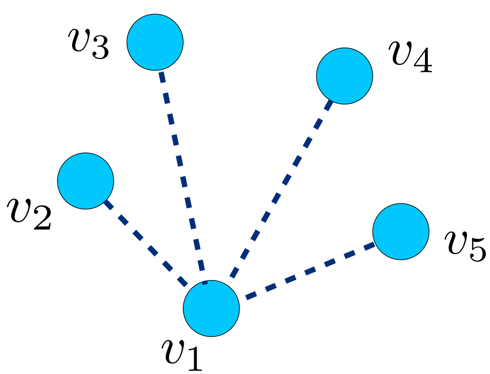


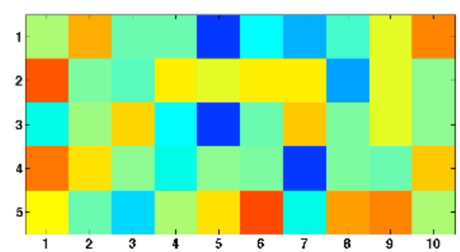
$\mathbf{S}^{-1}$

inverse of sample covariance

Historical overview

Learning a graph = learning neighbourhood of each node

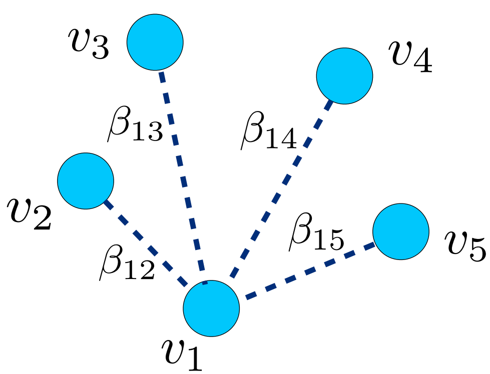


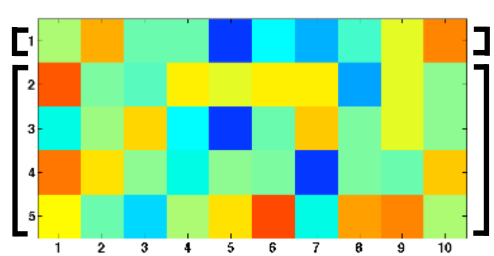


SDSC

Historical overview

Learning a graph = learning neighbourhood of each node



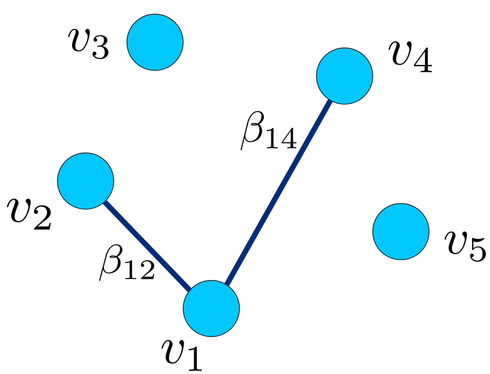


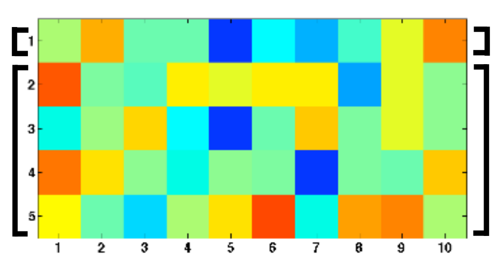
$\mathbf{X}_1{}^T$

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

Historical overview

Learning a graph = learning neighbourhood of each node





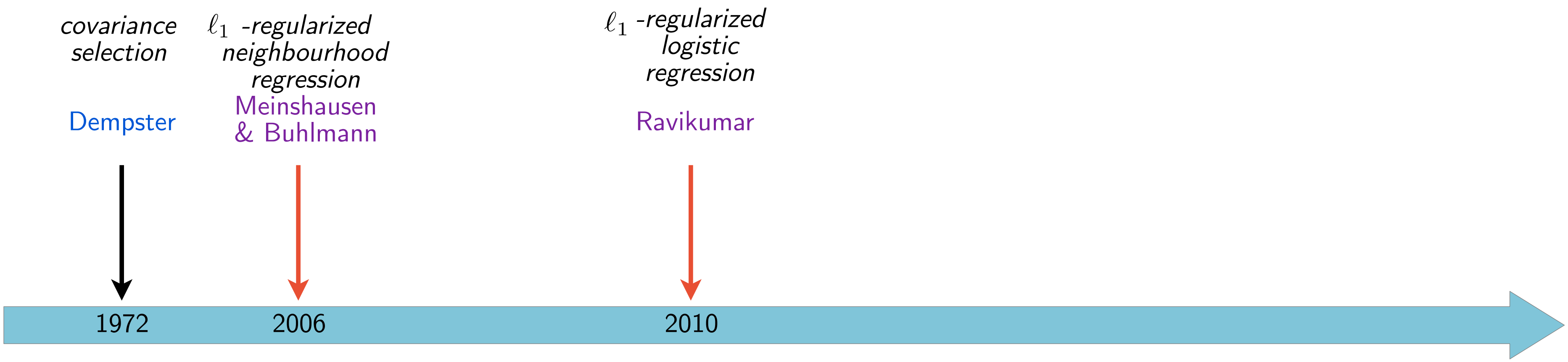
$\mathbf{X}_1{}^T$

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

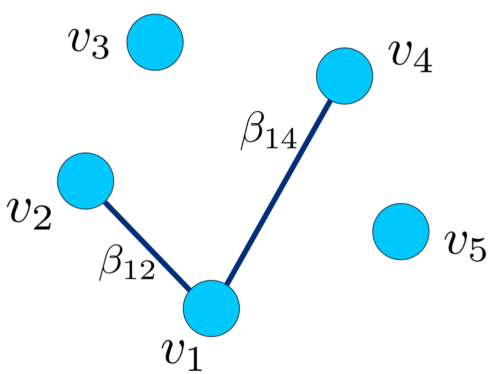
LASSO regression: $\min_{\boldsymbol{\beta}_1} ||\mathbf{X}_1 - \mathbf{X}_{\backslash 1}\boldsymbol{\beta}_1||^2 + \lambda||\boldsymbol{\beta}_1||_1$

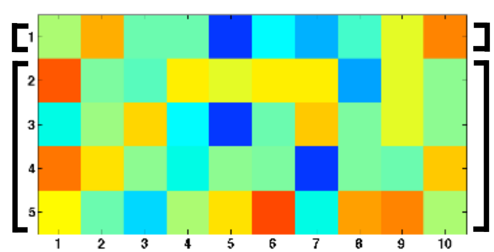
SDSC

Historical overview



Learning a graph = learning neighbourhood of each node





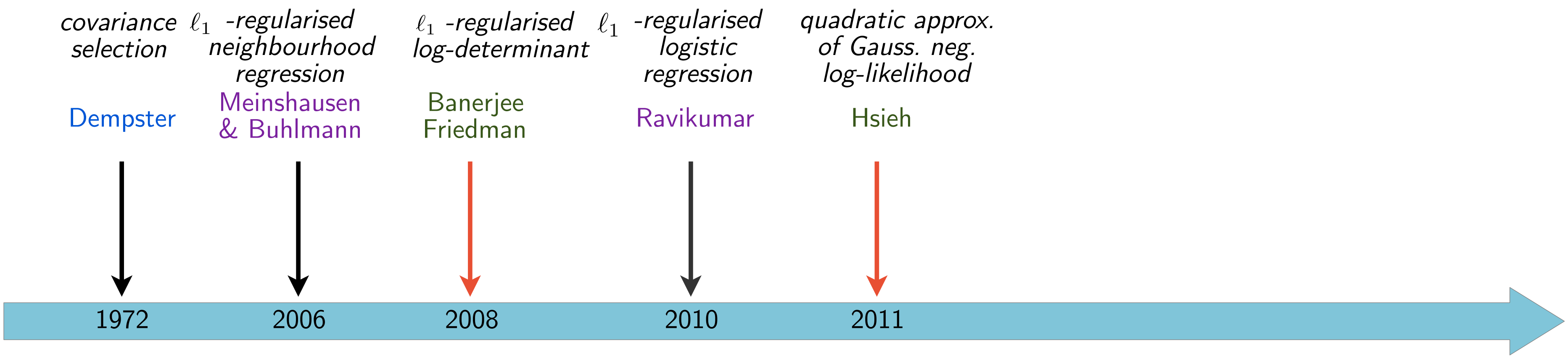
$\mathbf{X}_1{}^T$

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

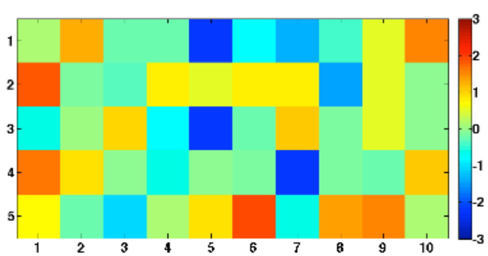
LASSO regression: $\min_{\boldsymbol{\beta}_1} ||\mathbf{X}_1 - \mathbf{X}_{\backslash 1}\boldsymbol{\beta}_1||^2 + \lambda||\boldsymbol{\beta}_1||_1$

Logistic regression for discrete variables

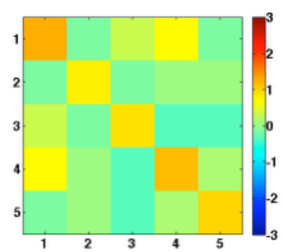
Historical overview



Estimation of sparse precision matrix



$\mathbf{X}$

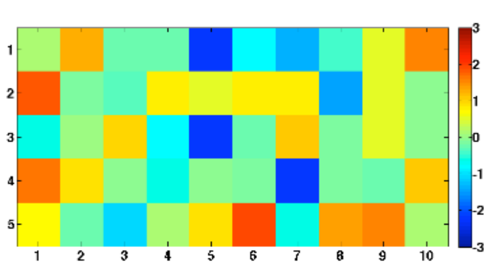


$\mathbf{S}$

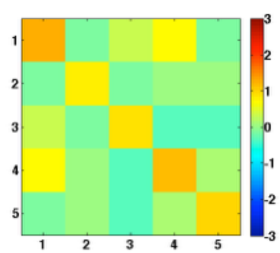
Historical overview

- 1972 — *covariance selection* — Dempster
- 2006 — *ℓ_1 -regularised neighbourhood regression* — Meinshausen & Buhlmann
- 2008 — *ℓ_1 -regularised log-determinant* — Banerjee Friedman
- 2010 — *ℓ_1 -regularised logistic regression* — Ravikumar
- 2011 — *quadratic approx. of Gauss. neg. log-likelihood* — Hsieh

Estimation of sparse precision matrix



$\mathbf{X}$



$\mathbf{S}$

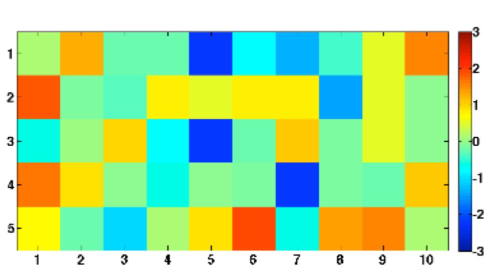
graphical LASSO maximizes likelihood of precision matrix $\boldsymbol{\Theta}$:

$$|\boldsymbol{\Theta}|^{M/2}\exp\Big(-\sum_{m=1}^{M}\frac{1}{2}\mathbf{X}(m)^T\boldsymbol{\Theta}\mathbf{X}(m)\Big)$$

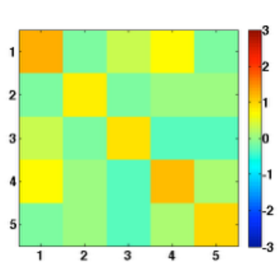
Historical overview

covariance selection	ℓ_1 -regularised neighbourhood regression	ℓ_1 -regularised log-determinant	ℓ_1 -regularised logistic regression	quadratic approx. of Gauss. neg. log-likelihood
Dempster	Meinshausen & Buhlmann	Banerjee Friedman	Ravikumar	Hsieh
1972	2006	2008	2010	2011

Estimation of sparse precision matrix



$\mathbf{X}$

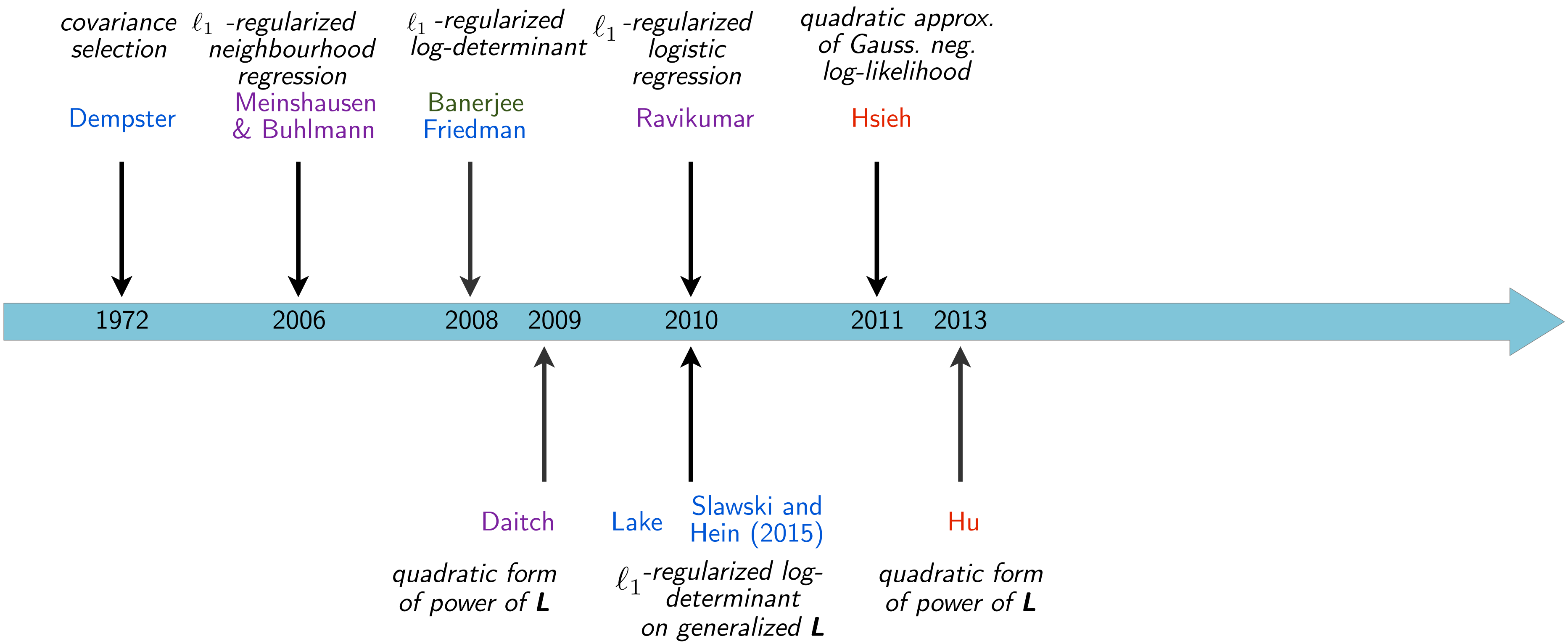


$\mathbf{S}$

graphical LASSO maximizes likelihood of precision matrix $\boldsymbol{\Theta}$:

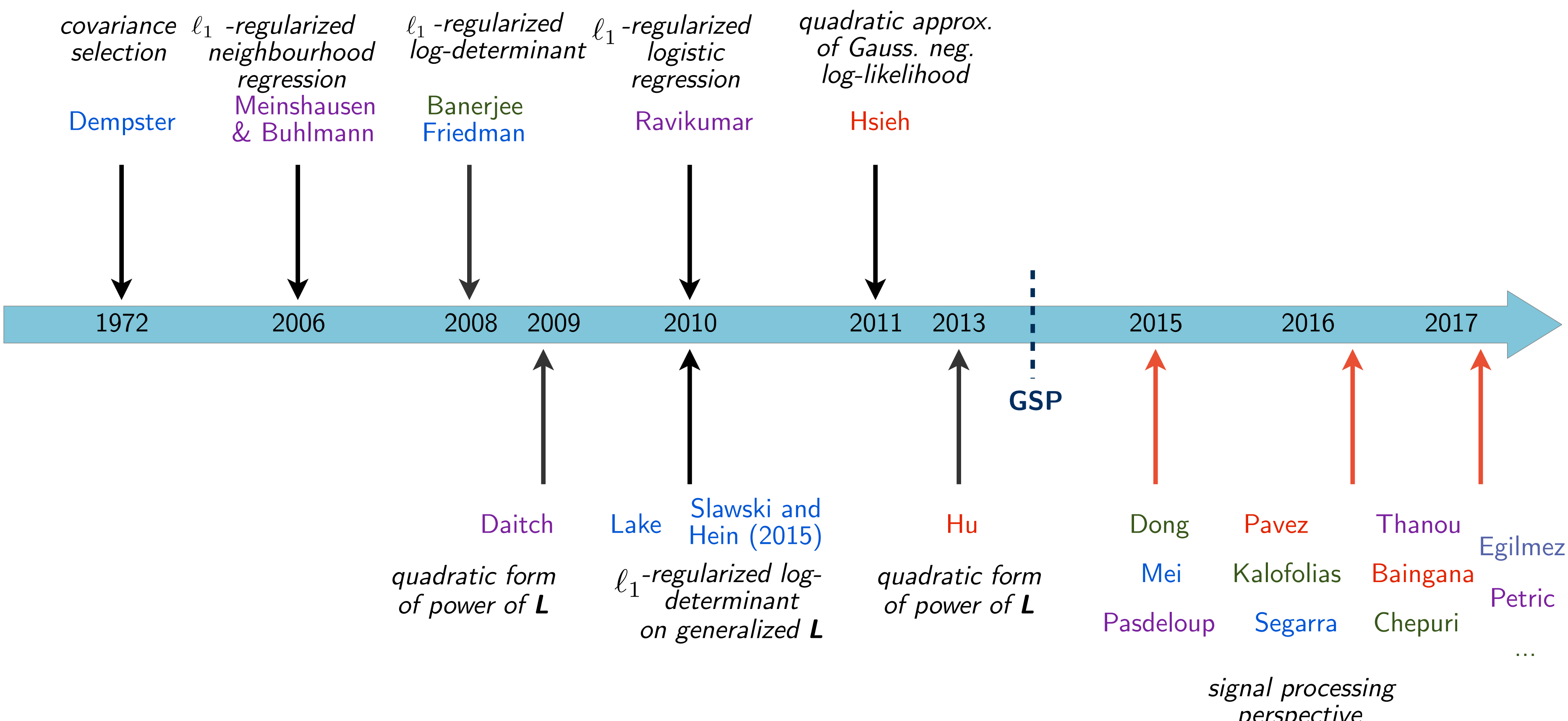
$$\max_{\boldsymbol{\Theta}} \underbrace{\log \det \boldsymbol{\Theta} - \mathrm{tr}(\mathbf{S}\boldsymbol{\Theta})}_{\text{log-likelihood function}} - \rho ||\boldsymbol{\Theta}||_1$$

Historical overview



From arbitrary precision matrix to graph Laplacian

Historical overview



From arbitrary precision matrix to graph Laplacian

Common setting in graph signal processing

SDSC

A signal processing approach?

- Existing approaches have limitations
  - Simple correlation/similarity functions are simplistic
  - There is no strong emphasis on signal/graph interaction with spectral/frequency domain interpretation

Outline

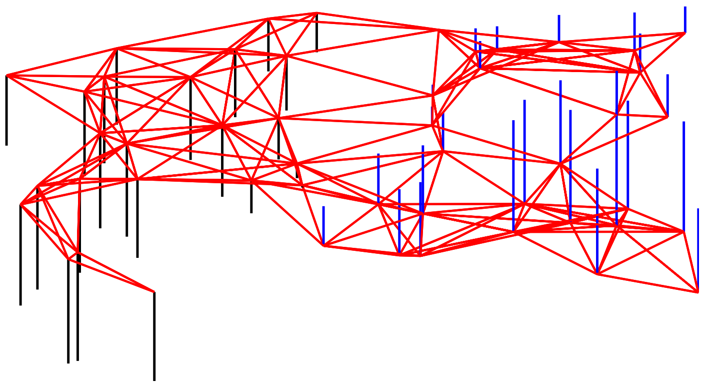
- A partial historical overview

- **A graph signal processing perspective**
  - **A quick reminder of GSP**
  - **GSP for topology inference**

- Applications/Concluding remarks

Graphs and signals on graphs

- **Graph:** A mathematical tool $G = (\mathcal{V}, \mathcal{E}, W)$ for modelling the irregular domain
  - directed/**undirected**
  - **weighted**/unweighted



- Connectivity pattern captured by:
  - Adjacency A or weight matrix W (positive weights)
  - Laplacian matrix L

- **Graph signal:** A function $x : \mathcal{V} \to \mathbb{R}$ that assigns real values to each vertex of the graph

The graph Laplacian operator

- It is defined as $L = D - W$
  - Complete set of eigenvectors $\chi = [\chi_0, \chi_1, ..., \chi_{N-1}]$
  - Non-negative eigenvalues $0 = \lambda_0 <= ... <= \lambda_{N-1}$

- Defines global smoothness on graphs

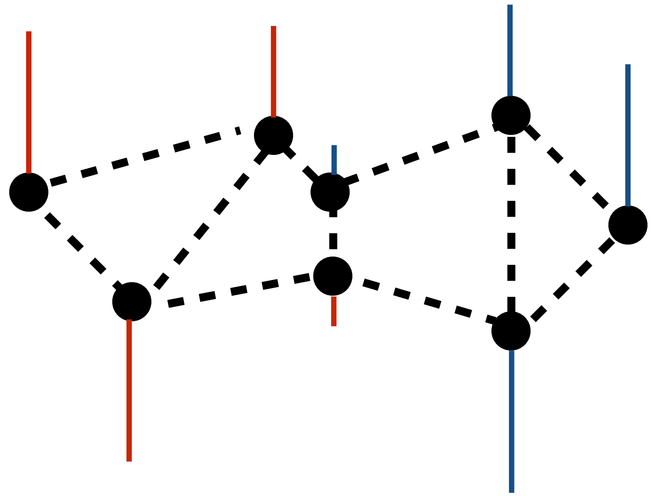
$$x^T L x = \sum_{m \in \mathcal{V}} \sum_{m \in \mathcal{N}_m} W_{m,n} [x(m) - x(n)]^2$$

SDSC

The graph Laplacian operator

- It is defined as $L = D - W$
  - Complete set of eigenvectors $\chi = [\chi_0, \chi_1, ..., \chi_{N-1}]$
  - Non-negative eigenvalues $0 = \lambda_0 <= ... <= \lambda_{N-1}$

- Defines global smoothness on graphs



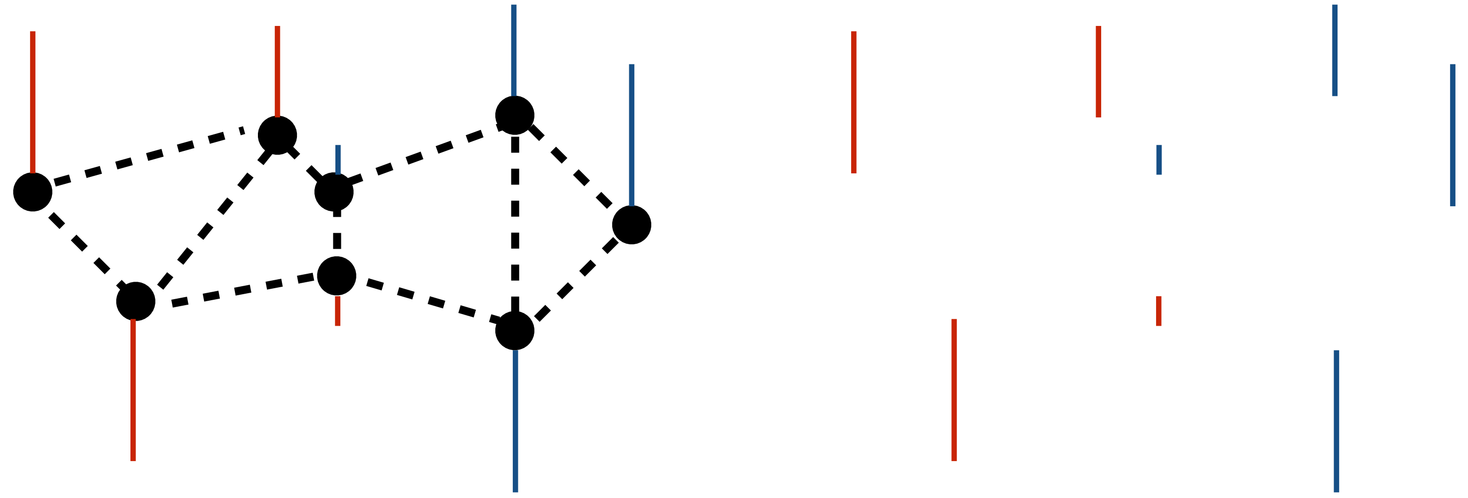
globally smooth signal

SDSC

The graph Laplacian operator

- It is defined as $L = D - W$
  - Complete set of eigenvectors $\chi = [\chi_0, \chi_1, ..., \chi_{N-1}]$
  - Non-negative eigenvalues $0 = \lambda_0 <= ... <= \lambda_{N-1}$

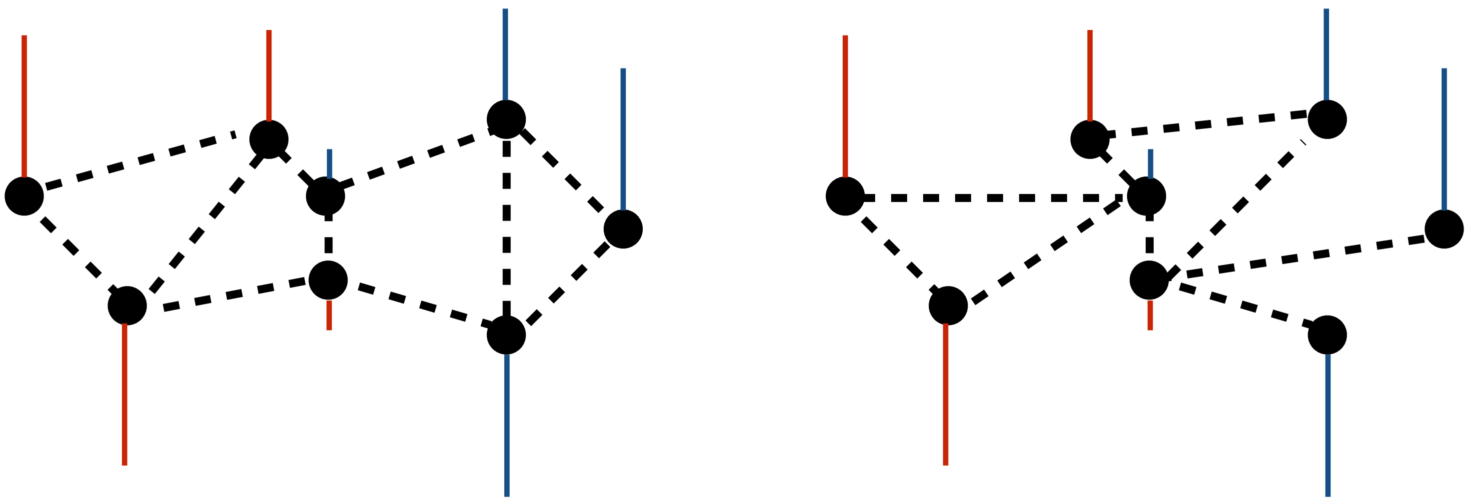
- Defines global smoothness on graphs



globally smooth signal

The graph Laplacian operator

- It is defined as $L = D - W$
  - Complete set of eigenvectors $\chi = [\chi_0, \chi_1, ..., \chi_{N-1}]$
  - Non-negative eigenvalues $0 = \lambda_0 <= ... <= \lambda_{N-1}$
- Defines global smoothness on graphs



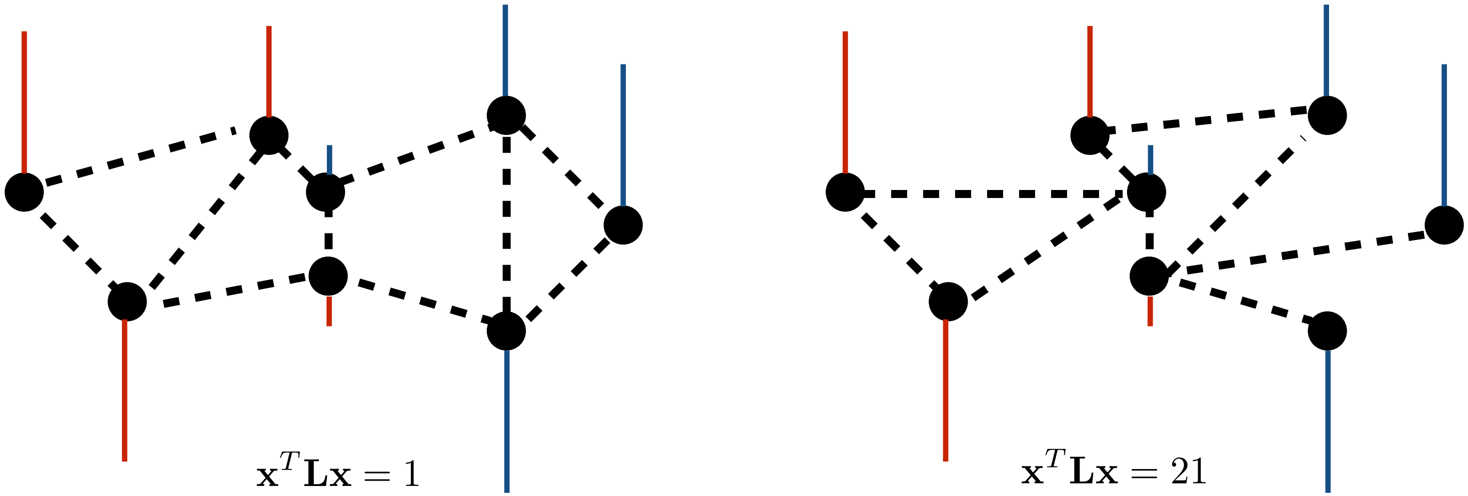
globally smooth signal

SDSC

The graph Laplacian operator

- It is defined as $L = D - W$
  - Complete set of eigenvectors $\chi = [\chi_0, \chi_1, ..., \chi_{N-1}]$
  - Non-negative eigenvalues $0 = \lambda_0 <= ... <= \lambda_{N-1}$

- Defines global smoothness on graphs



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

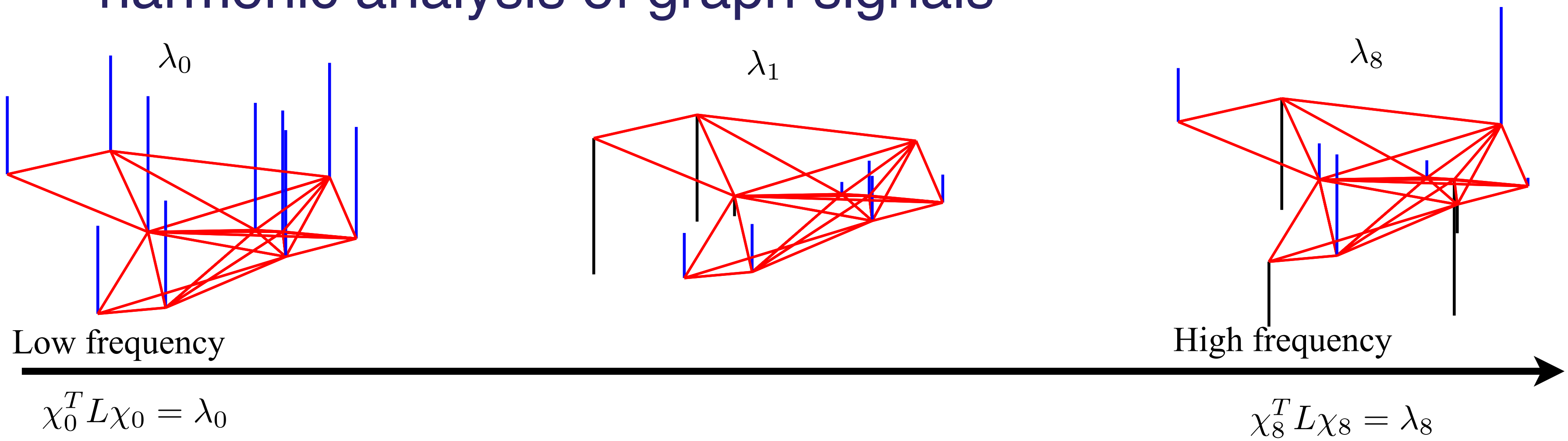
globally smooth signal

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

high frequency signal

The graph Fourier transform

- The eigenvectors of the Laplacian provide an harmonic analysis of graph signals



GFT $$\hat{x}(\lambda_\ell) = < x, \chi_\ell > = \sum_{n=1}^{N} x(n)\chi_\ell(n)$$

IGFT $$x(n) = \sum_{\ell=0}^{N-1} \hat{x}(\lambda_\ell)\chi_\ell(n)$$

Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

x_{in}

Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

$$x_{in} \xrightarrow{\text{GFT}} \chi^T x_{in}$$

Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

$$x_{in} \xrightarrow{\text{GFT}} \chi^T x_{in} \xrightarrow{\hat{g}(\Lambda)} \widehat{g}(\Lambda)\chi^T x_{in}$$

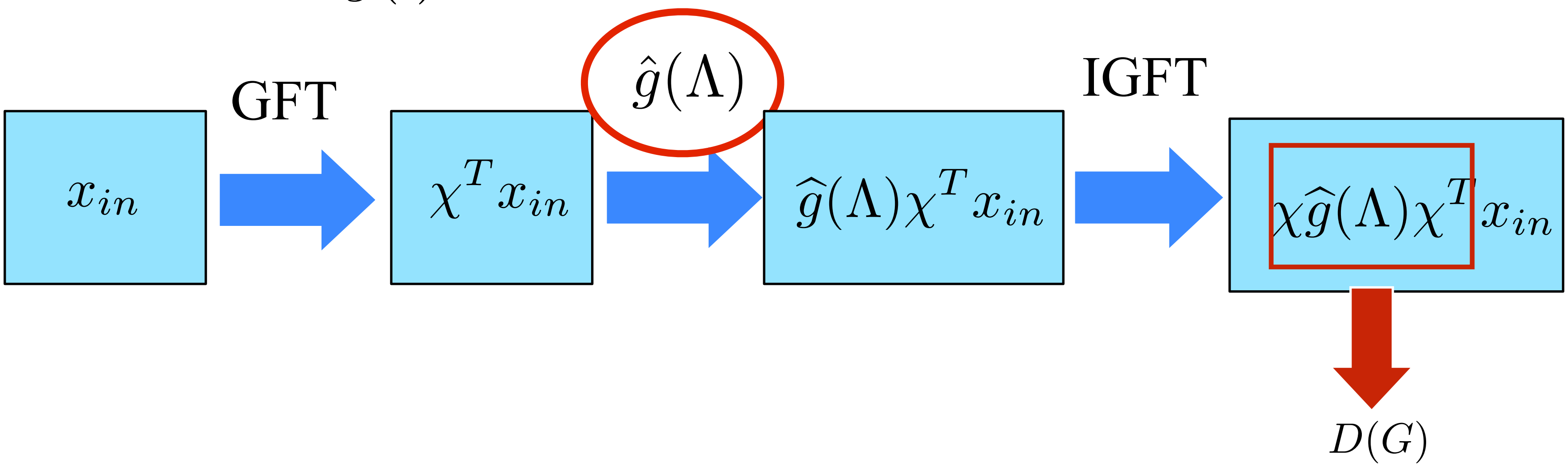
Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

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

Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

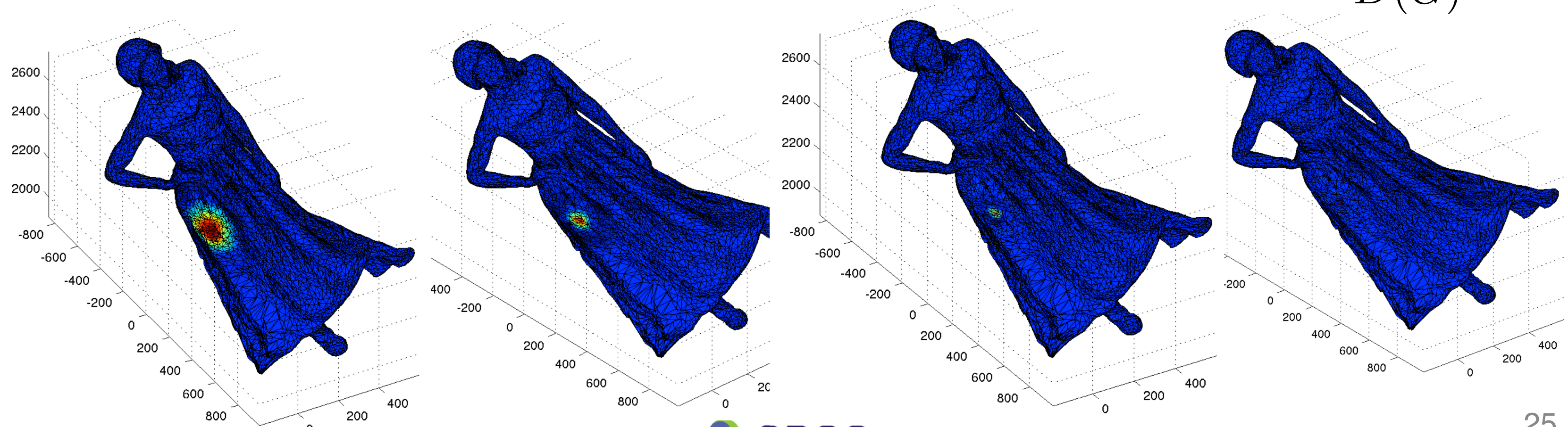


Filtering on graphs

- Filtering in the spectral domain with a transfer function $\hat{g}(\cdot)$

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

$$\boxed{\chi\hat{g}(\Lambda)\chi^T} \longrightarrow D(G)$$



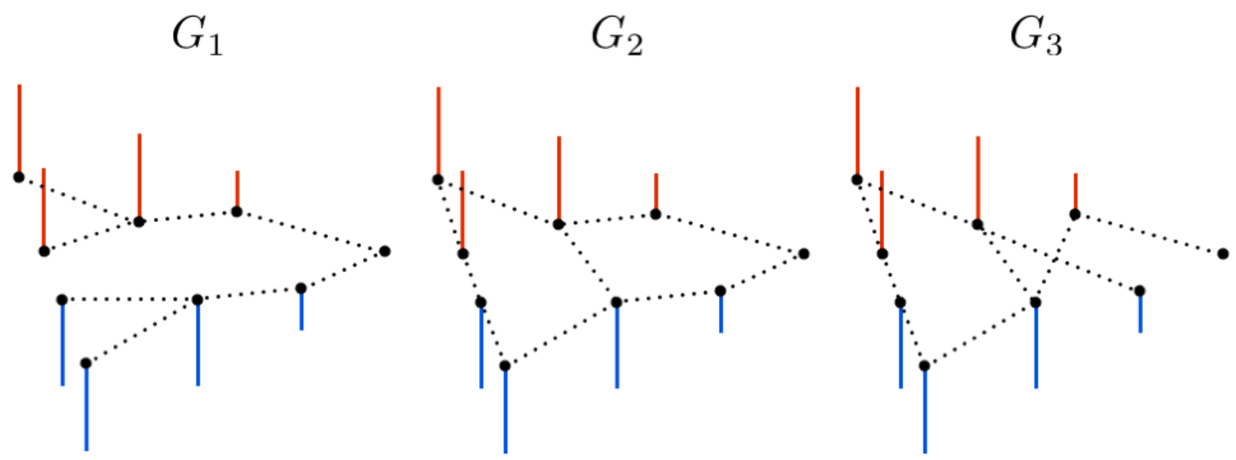
Topology inference revisited: A GSP perspective

- Define graph signal representation models
- Strong emphasis on signal/graph interaction with spectral/frequency-domain interpretation

SDSC

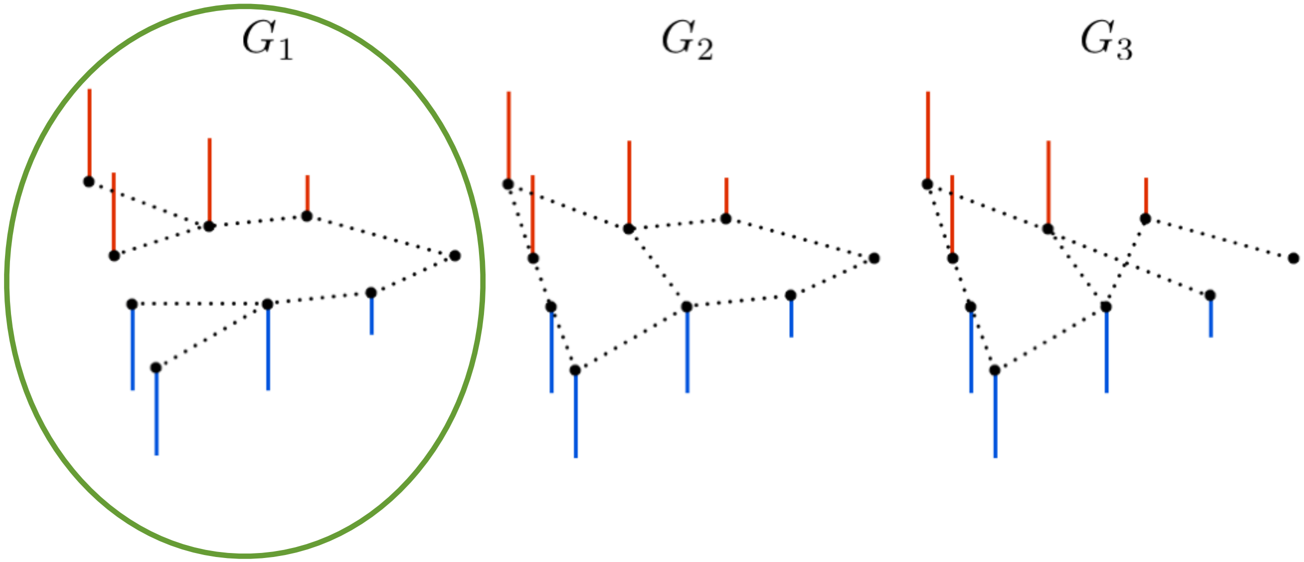
Topology inference revisited: A GSP perspective

- Define graph signal representation models
- Strong emphasis on signal/graph interaction with spectral/frequency-domain interpretation



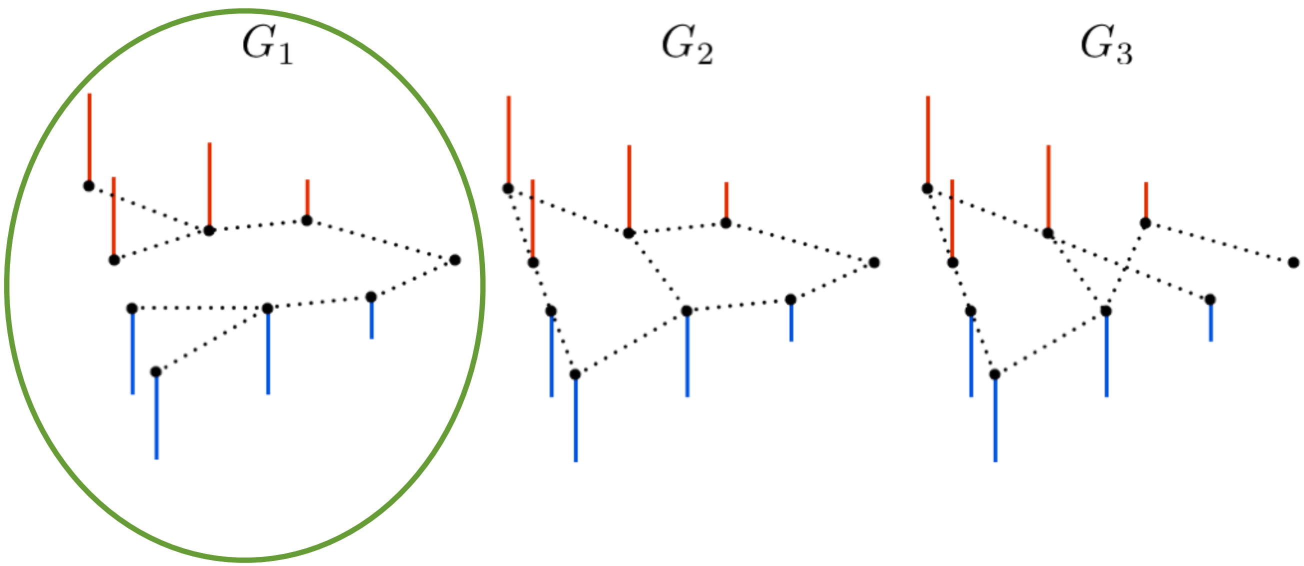
Topology inference revisited: A GSP perspective

- Define graph signal representation models
- Strong emphasis on signal/graph interaction with spectral/frequency-domain interpretation



Topology inference revisited: A GSP perspective

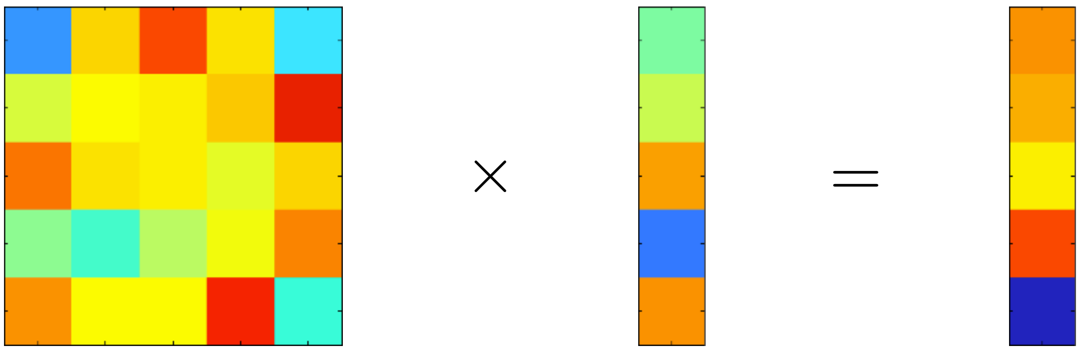
- Define graph signal representation models
- Strong emphasis on signal/graph interaction with spectral/frequency-domain interpretation



Can we exploit the interplay between graphs and signals on graphs to discover the topology?

Graph signal representations

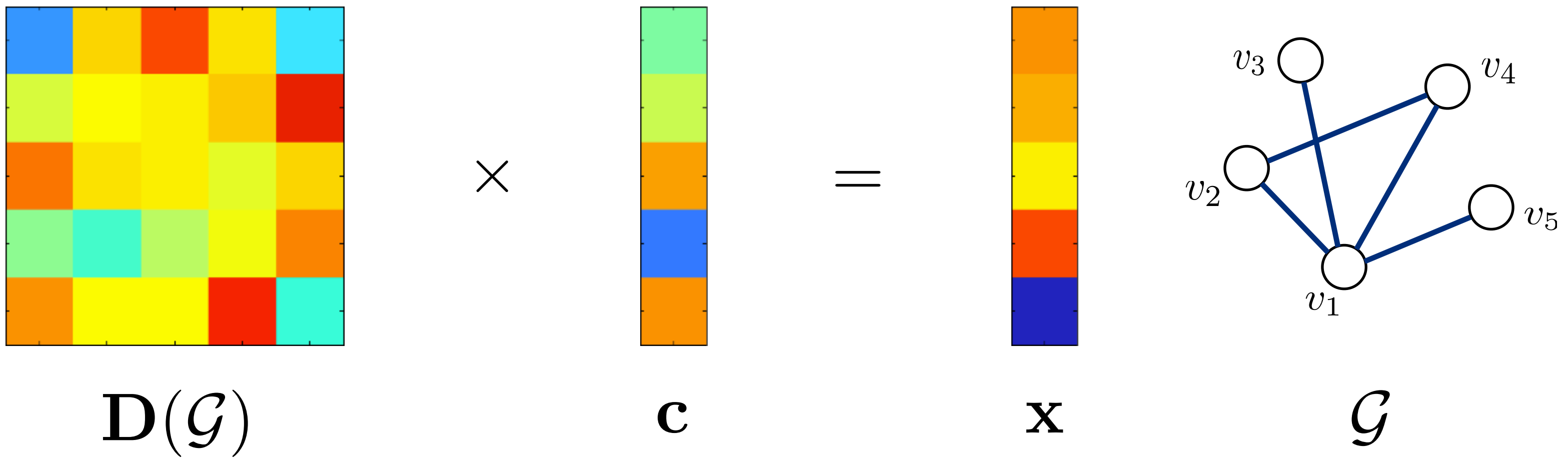
- Signal processing is about $Dc \approx x$



$$\mathbf{D} \times \mathbf{c} = \mathbf{x}$$

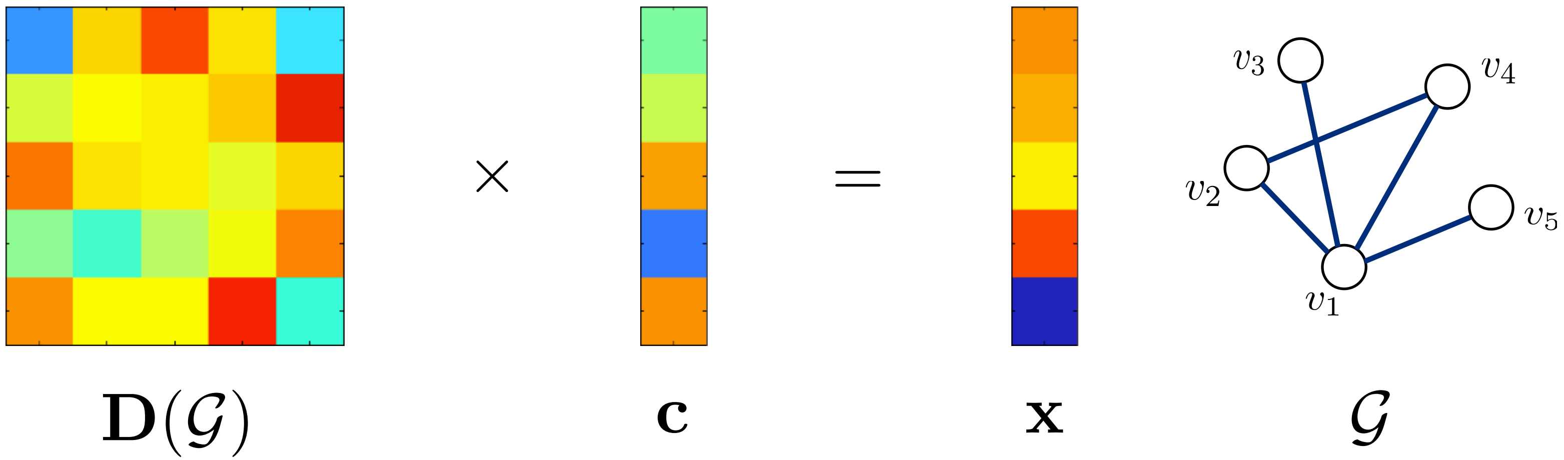
Graph signal representations

- Signal processing is about $Dc \approx x$
- Graph signal processing is about $D(G)c \approx x$

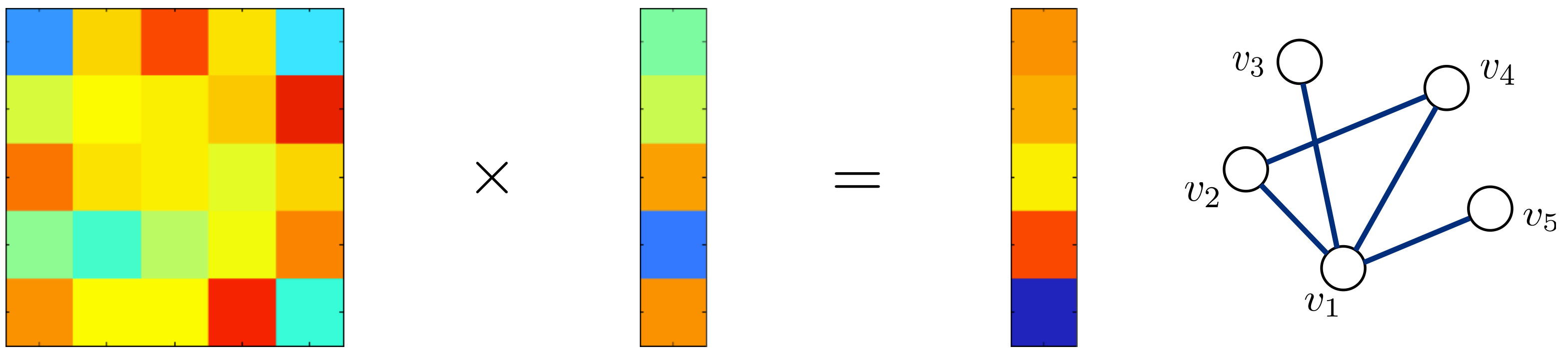


Graph signal representations

- Forward step: Given x, G, design D to study c



- Forward step: Given x, G, design D to study c

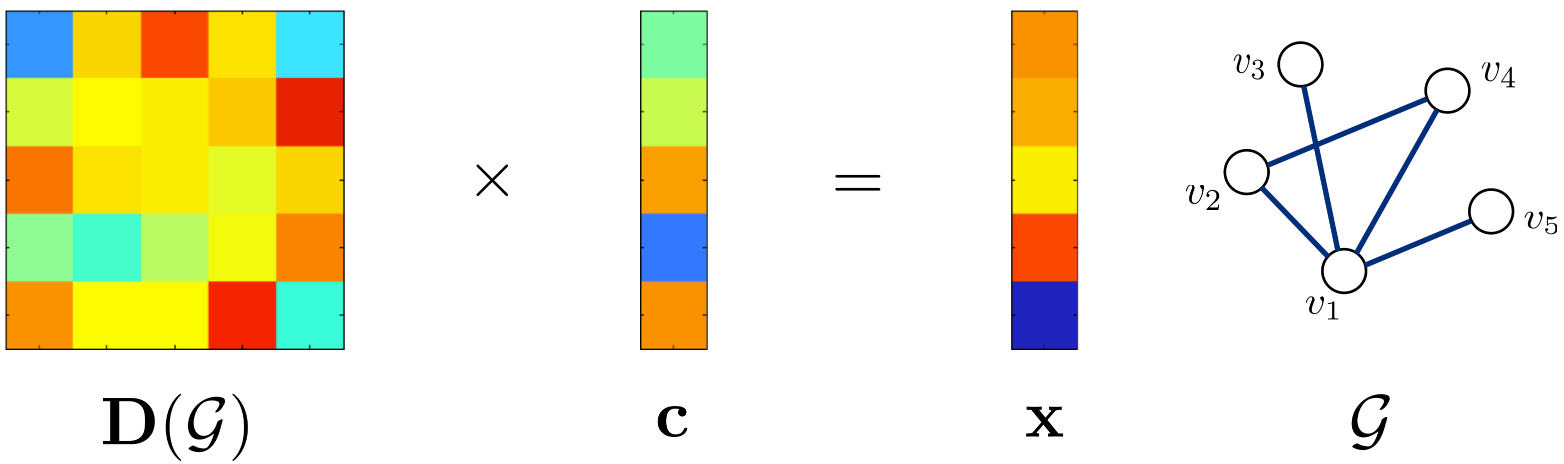


$$\mathbf{D}(\mathcal{G}) \quad \mathbf{c} \quad \mathbf{x} \quad \mathcal{G}$$

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

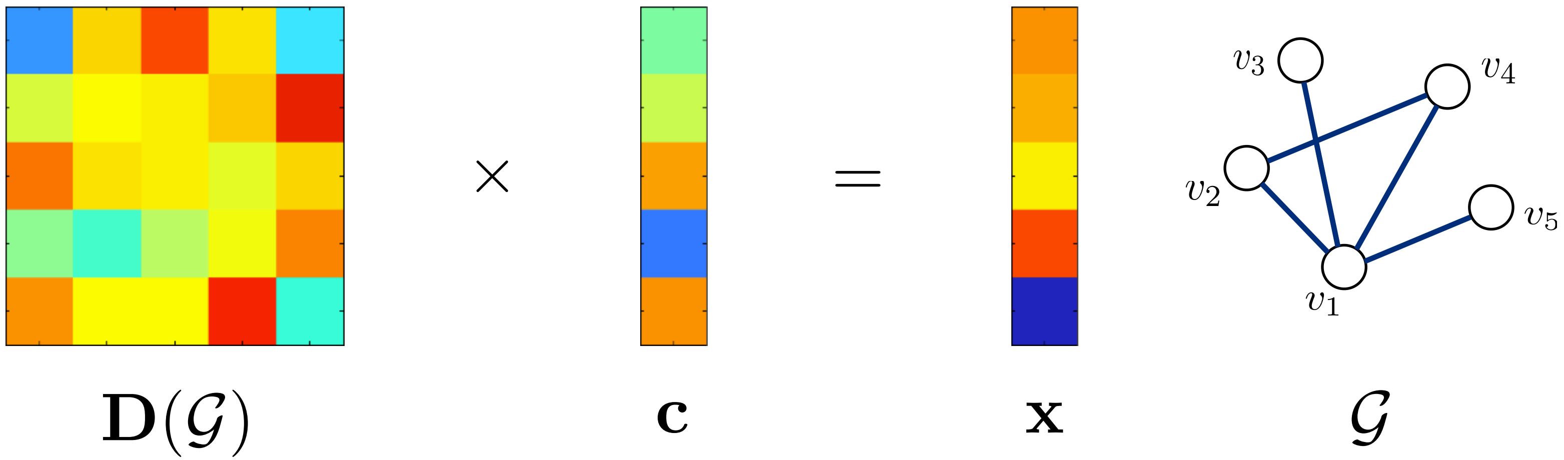
Graph signal representations

- Forward step: Given x, G, design D to study c



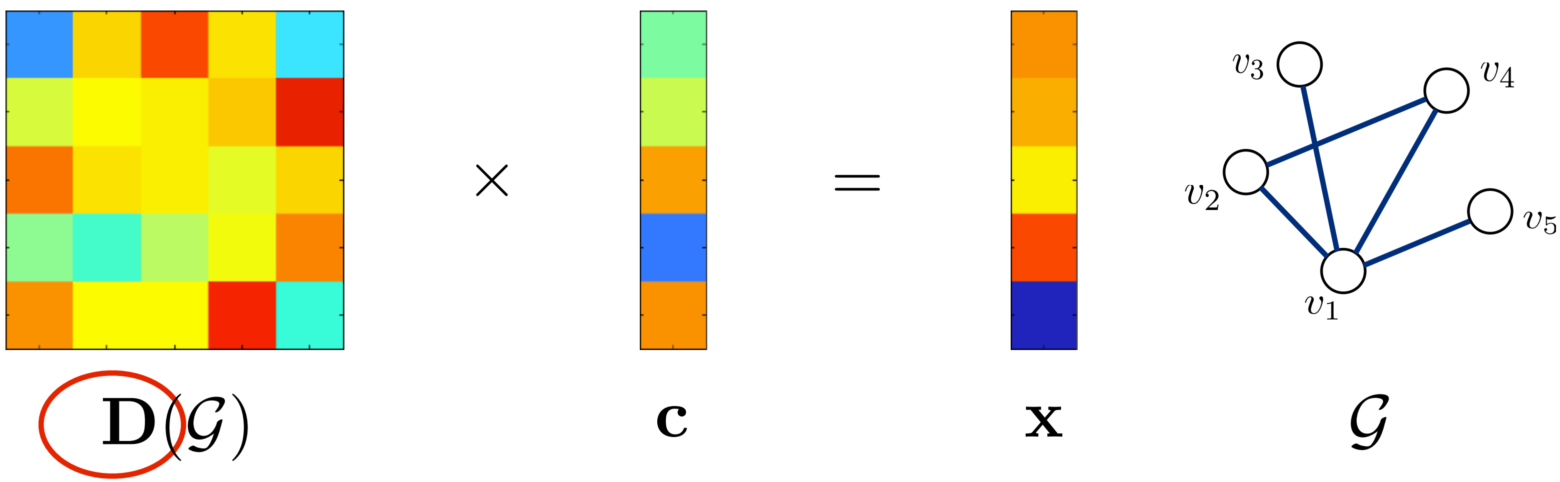
Graph signal representations

- Forward step: Given x, G, design D to study c
- Backward (topology inference): Given x, design D, c to infer G



Graph signal representations

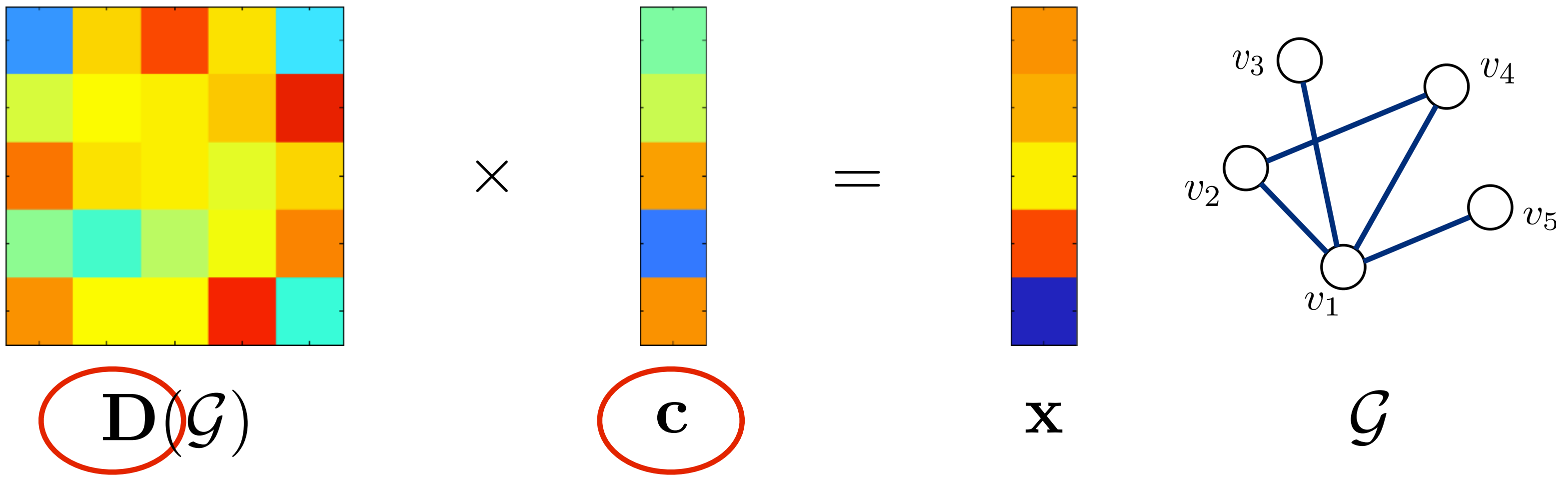
- Forward step: Given x, G, design D to study c
- Backward (topology inference): Given x , design D, c to infer G



- The key is a signal/graph model behind **D**
- Designed around graph operators (adjacency/Laplacian matrices, shift operators)

Graph signal representations

- Forward step: Given x, G, design D to study c
- Backward (topology inference): Given x , design D, c to infer G



- The key is a signal/graph model behind **D**
- Designed around graph operators (adjacency/Laplacian matrices, shift operators)
- Choice of/assumption on **c** often determines signal characteristics

Model 1: Global smoothness

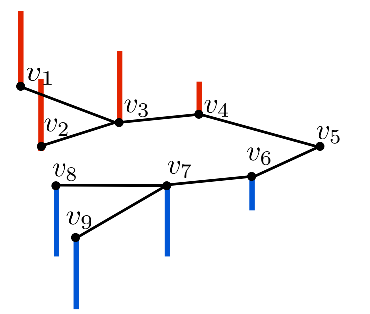
- Connected nodes on the graph have similar signal values
- Usually quantified by the Laplacian quadratic from:

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

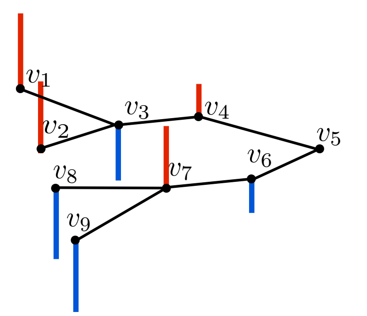
Model 1: Global smoothness

- Connected nodes on the graph have similar signal values
- Usually quantified by the Laplacian quadratic from:

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



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

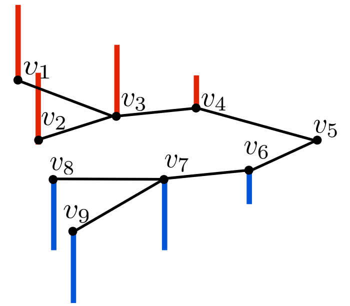


$$\mathbf{x}^T \mathbf{L} \mathbf{x} = 21$$

Model 1: Global smoothness

- Connected nodes on the graph have similar signal values
- Usually quantified by the Laplacian quadratic from:

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



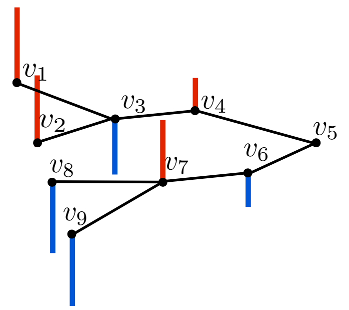
$\mathbf{x}^T\mathbf{L}\mathbf{x} = 1$

Similar to previous approaches:

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

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

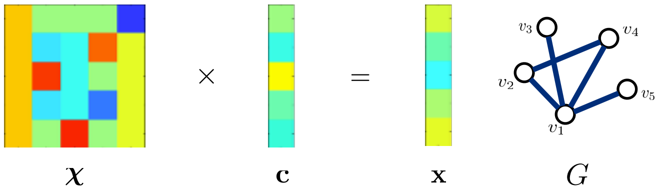
Hu (2013): $\min_{\mathbf{L}} \mathrm{tr}(\mathbf{X}^T\mathbf{L}^s\mathbf{X}) - \beta||\mathbf{W}||_F$



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

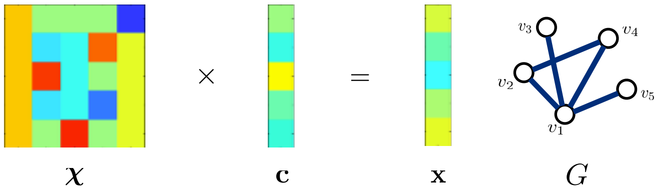
Model 1: Global smoothness

- Dong et al (2015), Kalofolias (2016)
  - $D(G) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$
  - Gaussian Markov Random Field
    $$x \sim \mathcal{N}(0, L^{\dagger} + \sigma_{\epsilon}^{2} I)$$



Model 1: Global smoothness

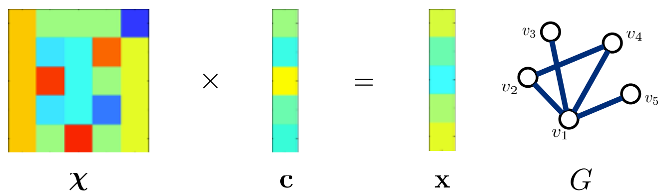
- Dong et al (2015), Kalofolias (2016)
  - $D(G) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$
  - Gaussian Markov Random Field
    $$x \sim \mathcal{N}(0, L^{\dagger} + \sigma_{\epsilon}^{2} I)$$
  - Maximum a posteriori (MAP) estimation of c leads to minimization of the Laplacian quadratic form:



$$
\begin{aligned}
c_{\mathrm{MAP}}(x) &:= \arg\max_{c} p(c|x) = \arg\max_{c} p(x|c)p(c) \\
&= \arg\min_{c} \left(-\log\, p_{\epsilon}(x - \chi c) - \log\, p_{C}(c)\right) \\
&= \arg\min_{c} \|x - \chi c\|_2^2 + \alpha c^T \Lambda c
\end{aligned}
$$

Model 1: Global smoothness

- Dong et al (2015), Kalofolias (2016)



  - $D(G) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$
  - Gaussian Markov Random Field

$$x \sim \mathcal{N}(0, L^{\dagger} + \sigma_{\epsilon}^2 I)$$

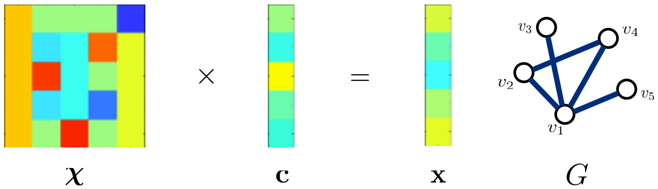
  - Maximum a posteriori (MAP) estimation of c leads to minimization of the Laplacian quadratic form:

$$\begin{aligned} c_{\mathrm{MAP}}(x) &:= \arg\max_c p(c|x) = \arg\max_c p(x|c)p(c) \\ &= \arg\min_c \left(-\log\, p_{\epsilon}(x - \chi c) - \log\, p_C(c)\right) \\ &= \arg\min_c \|x - \chi c\|_2^2 + \alpha c^T \Lambda c \end{aligned}$$

$$\min_{\mathbf{L},\mathbf{Y}} \underbrace{||\mathbf{X} - \mathbf{Y}||_F^2}_{\text{data fidelity}} + \alpha \underbrace{\mathrm{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y})}_{\text{smoothness on } \mathbf{Y}} + \beta \underbrace{||\mathbf{L}||_F^2}_{\text{regularization}}$$

Model 1: Global smoothness

- Dong et al (2015), Kalofolias (2016)
  - $D(G) = \chi$ (eigenvector matrix of **L**)
  - Gaussian assumption on **c**: $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$
  - Gaussian Markov Random Field
    $$x \sim \mathcal{N}(0, L^{\dagger} + \sigma_{\epsilon}^2 I)$$
  - Maximum a posteriori (MAP) estimation of c leads to minimization of the Laplacian quadratic form:



$$\begin{aligned} c_{\mathrm{MAP}}(x) &:= \arg\max_c p(c|x) = \arg\max_c p(x|c)p(c) \\ &= \arg\min_c \left(-\log\, p_{\epsilon}(x - \chi c) - \log\, p_C(c)\right) \\ &= \arg\min_c \|x - \chi c\|_2^2 + \alpha c^T \Lambda c \end{aligned}$$

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

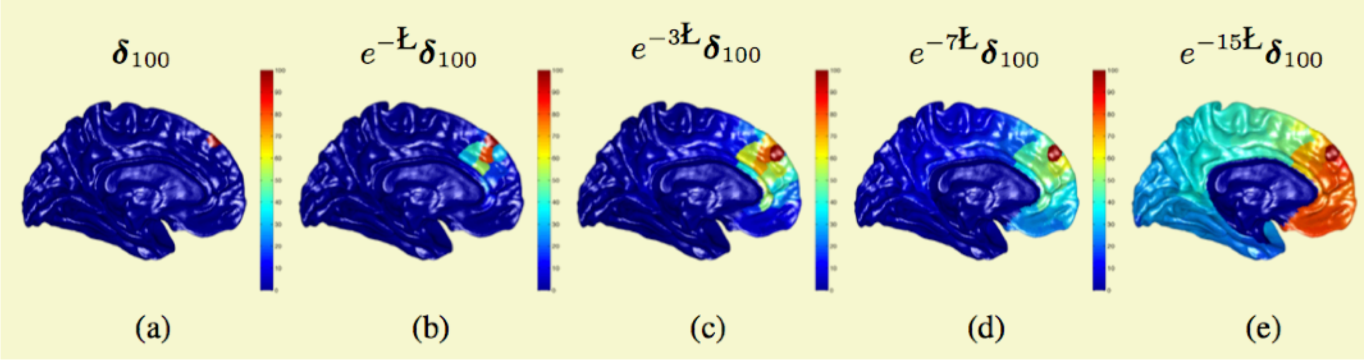
data fidelity smoothness on **Y** regularization

Learning enforces signal property (global smoothness)!

SDSC

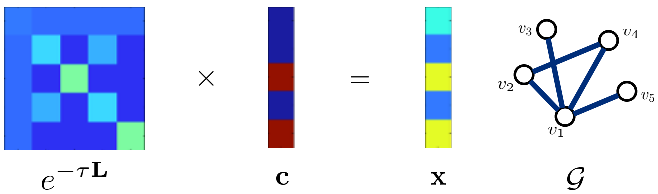
Model 2: Diffusion process

- Signals are the outcome of a diffusion process on the graph (more of local smoothness than global one!)
- Example: Movement of people/ vehicles
- Fully characterized by diffusion operators (graph filters)



Heat diffusion operators on the brain [7]

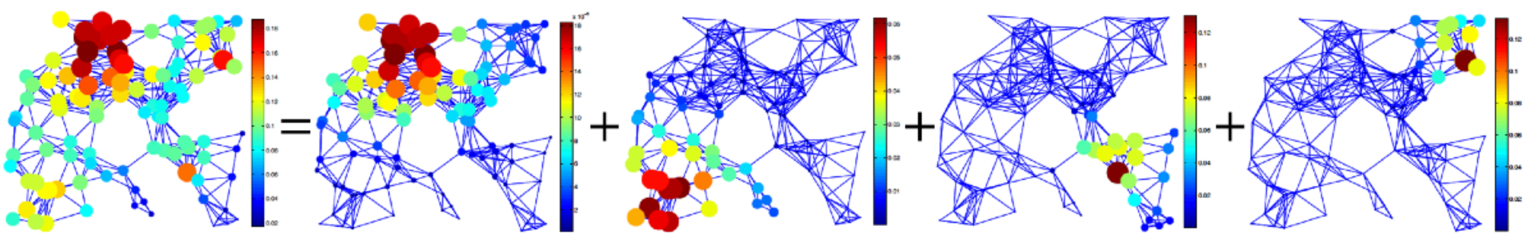
Model 2: Diffusion process



- Thanou et al. (2016)
  - Dictionary of heat diffusion operators

$$D(G) = e^{-\tau L}$$

  - Sparsity assumption on c
  - Each signal is a linear combination of heat diffusion processes

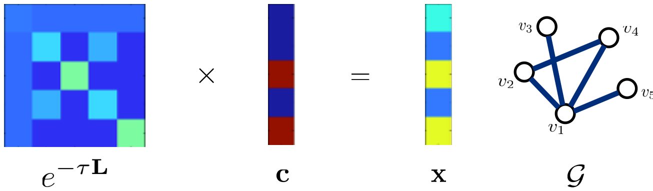


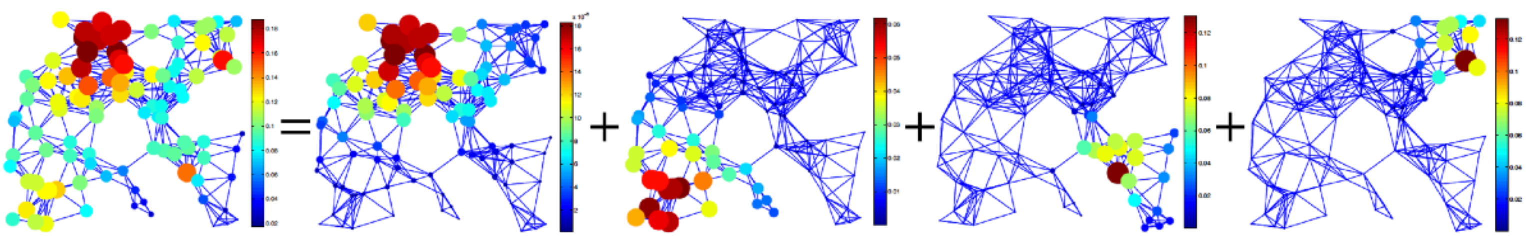
Model 2: Diffusion process

- Thanou et al. (2016)
  - Dictionary of heat diffusion operators

$$D(G) = e^{-\tau L}$$

  - Sparsity assumption on c
  - Each signal is a linear combination of heat diffusion processes



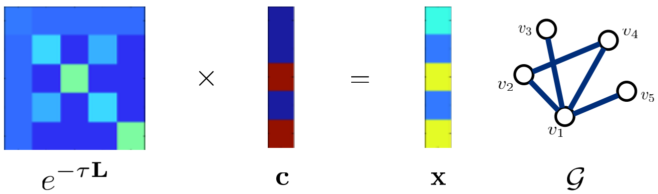


$$
\begin{aligned}
c_{\mathrm{MAP}}(x) &:= \arg\max_c p(c|x) = \arg\max_c p(x|c)p(c) \\
&= \arg\min_c \left(-\log\, p_\epsilon(x - e^{-\tau L}c) - \log\, p_C(c)\right) \\
&= \arg\min_c \|x - e^{-\tau L}c\|_2^2 + \alpha\|c\|_1
\end{aligned}
$$

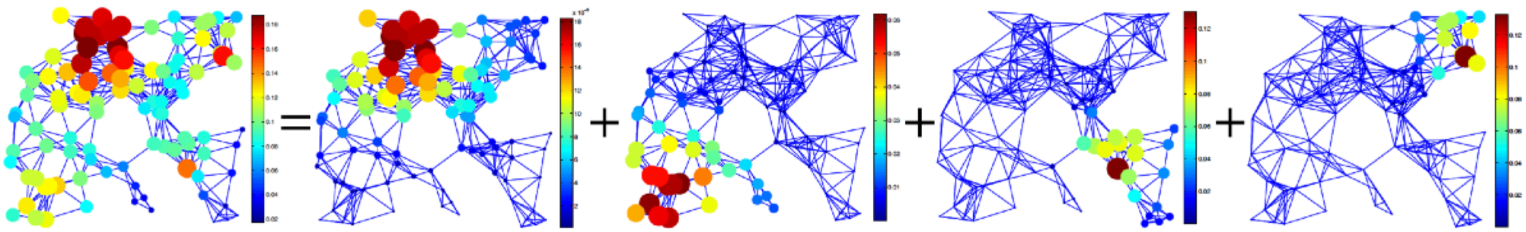
Model 2: Diffusion process

- Thanou et al. (2016)
  - Dictionary of heat diffusion operators

$$D(G) = e^{-\tau L}$$



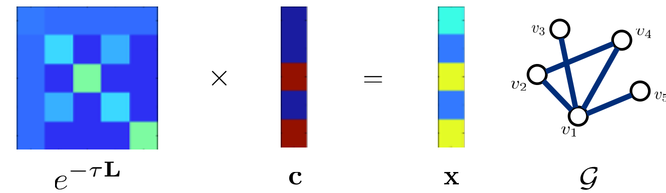
  - Sparsity assumption on c
  - Each signal is a linear combination of heat diffusion processes

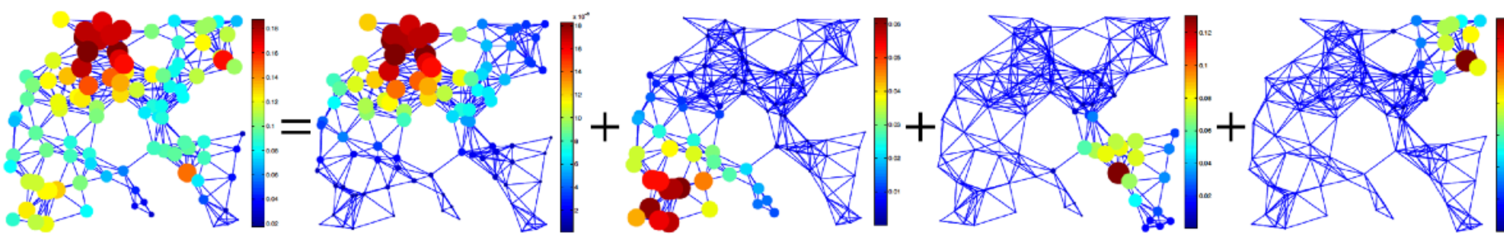


$$\min_{\mathbf{L},\mathbf{C},\tau} ||\mathbf{X} - \mathbf{D}(\mathbf{L})\mathbf{C}||_F^2 + \alpha \sum_{m=1}^{M} ||\mathbf{c}_m||_1 + \beta ||\mathbf{L}||_F^2 \quad \text{s.t.} \quad \mathbf{D} = [e^{-\tau_1 \mathbf{L}}, ..., e^{-\tau_S \mathbf{L}}]$$

Model 2: Diffusion process

- Thanou et al. (2016)
  - Dictionary of heat diffusion operators
    $$D(G) = e^{-\tau L}$$
  - Sparsity assumption on c
  - Each signal is a linear combination of heat diffusion processes





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

data fidelity　　sparsity on **c**　　regularization

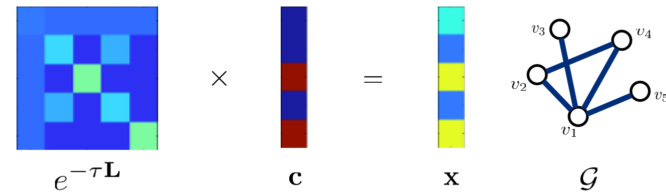
SDSC

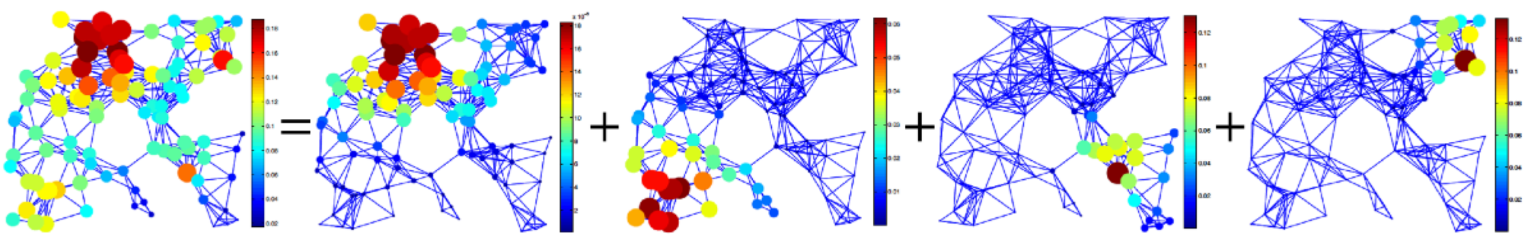
Model 2: Diffusion process

- Thanou et al. (2016)
  - Dictionary of heat diffusion operators

$$D(G) = e^{-\tau L}$$

  - Sparsity assumption on c
  - Each signal is a linear combination of heat diffusion processes





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

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

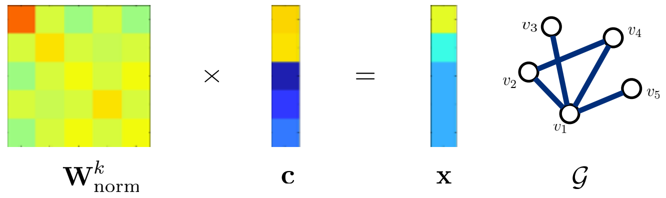
Can be extended to general polynomial case (Petric Maretic et al. 2017)

SDSC

Model 2: Diffusion process

- Pasdeloup et al. (2016), Segarra et al. (2016)
  - Dictionary: powers of the normalized weight matrix

$$\mathbf{D}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}(m)} = \mathbf{W}_{\text{norm}}^{\mathbf{k}(m)}$$



  - i.i.d. assumption on the coefficients c

  - Two stage approach for estimating the topology:

    - Estimate the eigenvectors from the sample covariance

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

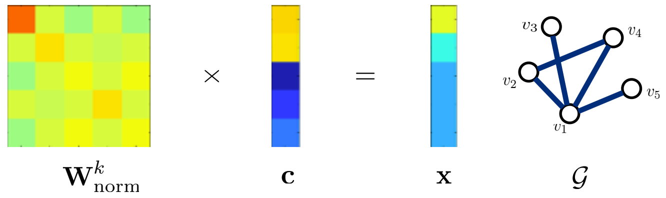
    - Optimize the eigenvalues given constraints on the topology (sparsity, non-negativity)

Model 2: Diffusion process

- Pasdeloup et al. (2016), Segarra et al. (2016)
  - Dictionary: powers of the normalized weight matrix

    $$\mathbf{D}(\mathcal{G}) = \mathbf{T}^{\mathbf{k}(m)} = \mathbf{W}_{\text{norm}}^{\mathbf{k}(m)}$$

  - i.i.d. assumption on the coefficients c



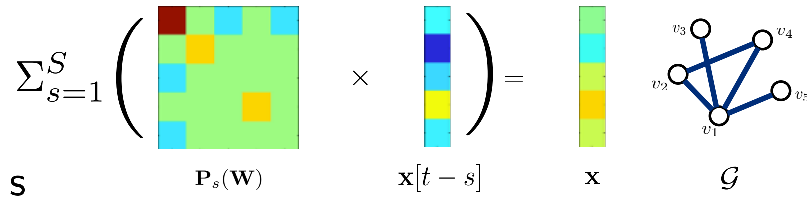
  - Two stage approach for estimating the topology:
    - Estimate the eigenvectors from the sample covariance

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

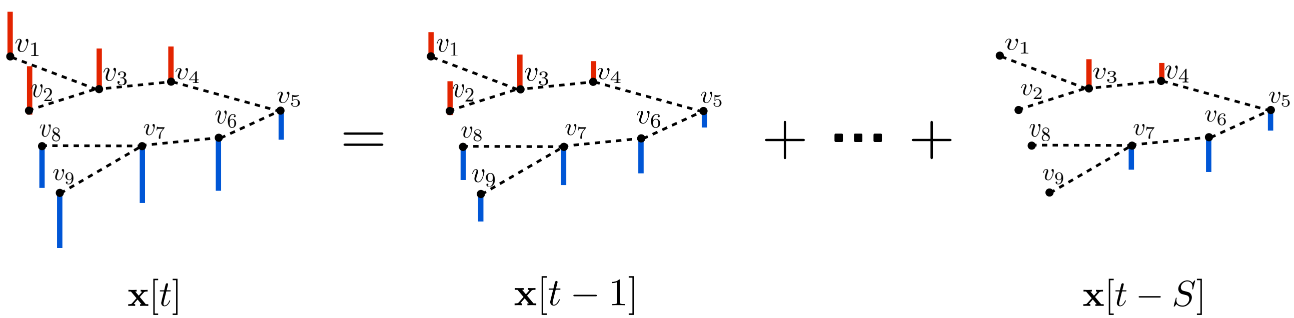
    - Optimize the eigenvalues given constraints on the topology (sparsity, non-negativity)

More a "graph-centric" framework: Cost on graph components instead of signals

Model 3: Time varying signals



- Mei et al. (2016)
  - $D_s(G) = P_s(W)$: polynomial of **W** of degree s
  - Define $c_s = x[t-s]$

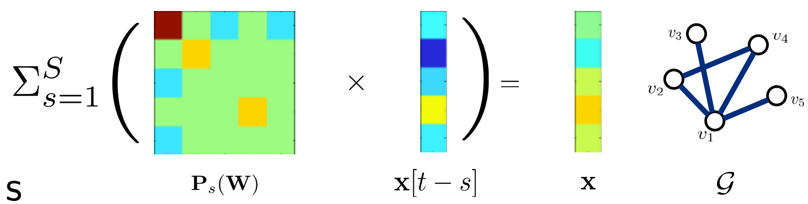


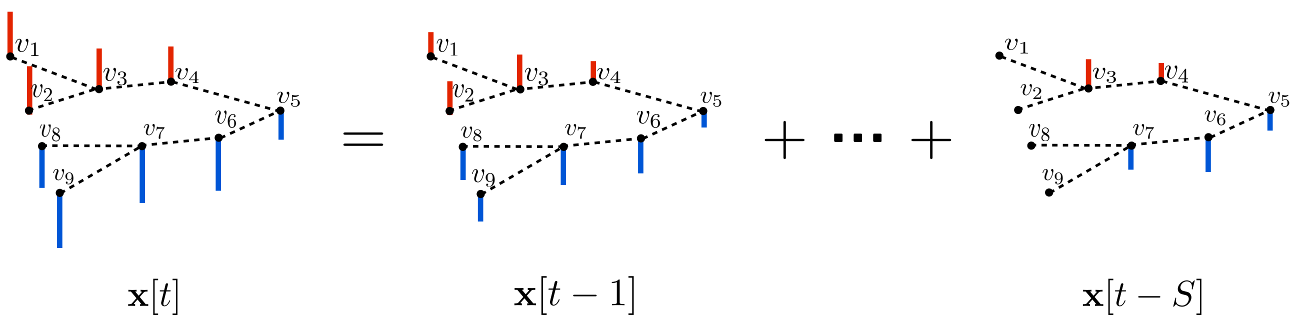
$$\min_{\mathbf{W},\mathbf{a}} \frac{1}{2} \sum_{k=S+1}^{K} ||\mathbf{x}[k] - \sum_{s=1}^{S} \mathbf{P}_s(\mathbf{W})\mathbf{x}[k-s]||_2^2 + \lambda_1 ||\text{vec}(\mathbf{W})||_1 + \lambda_2 ||\mathbf{a}||_1$$

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

Model 3: Time varying signals

- Mei et al. (2016)
  - $D_s(G) = P_s(W)$: polynomial of **W** of degree s
  - Define $c_s = x[t-s]$





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

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

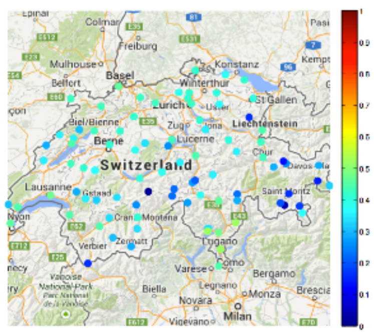
Good for inferring causal relations between signals

Kernel version (nonlinear): Shen et al. (2016)

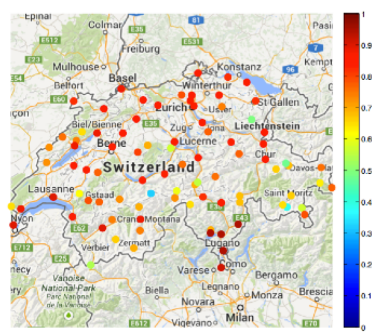
Application 1: Understanding meteorological data

- Signals: 12 monthly temperature records at 89 measuring stations [9]

February

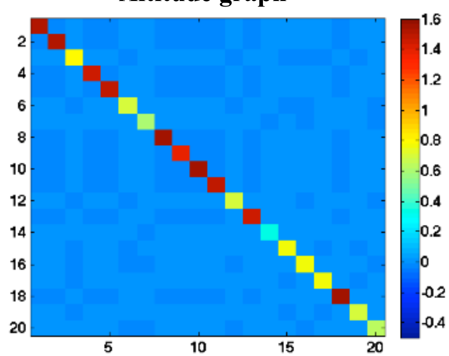


June

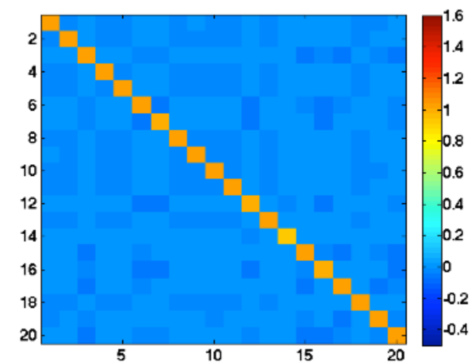


October

Altitude graph



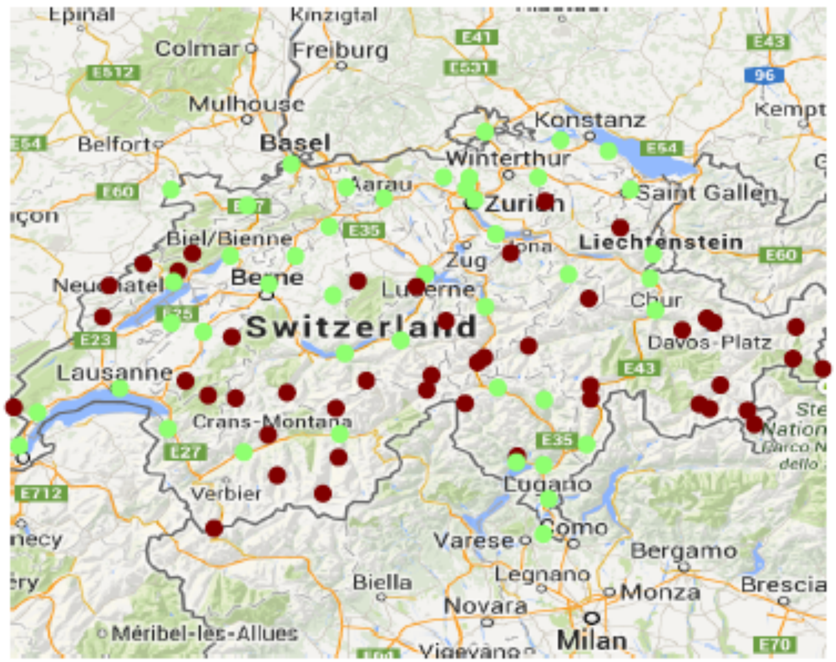
Learned graph



The learned graph captures the altitude relation between stations

Application 1: Understanding meteorological data

- The learned graph can be used for partitioning nodes into several clusters:

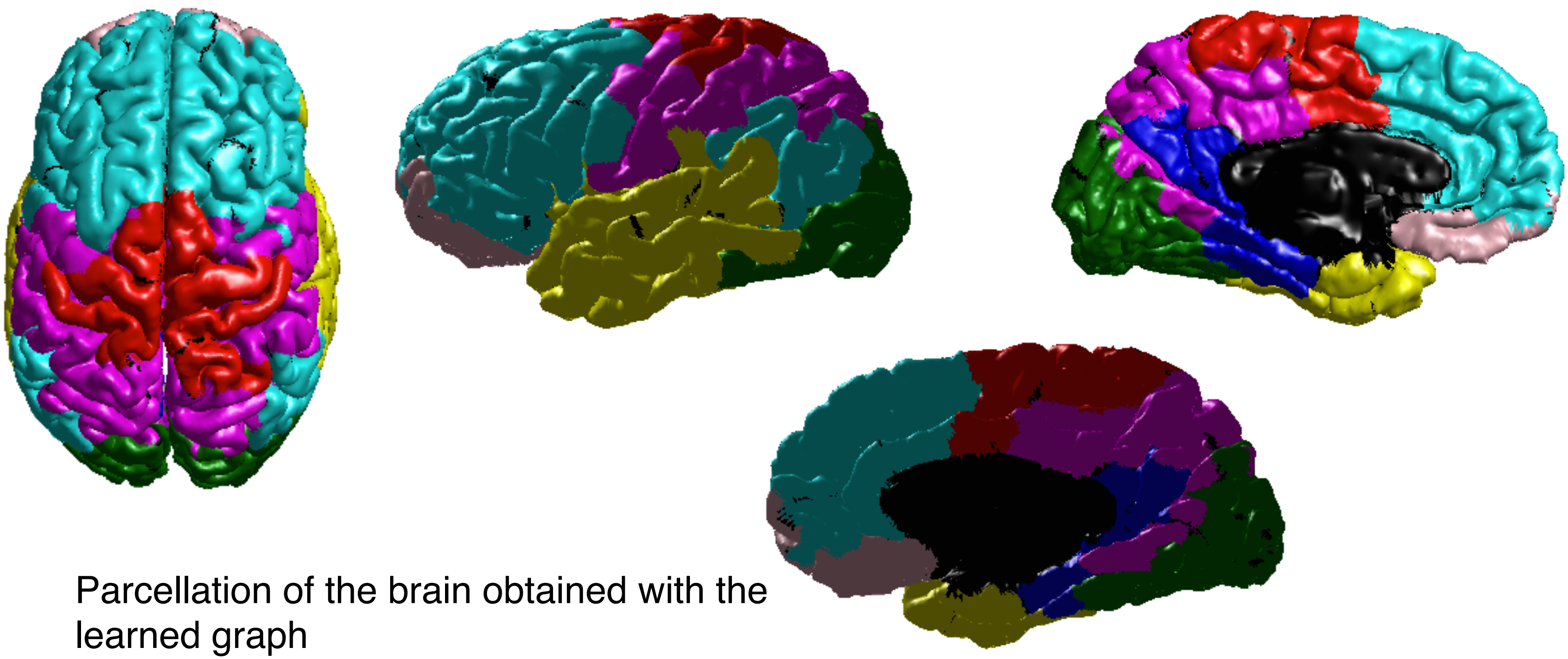


Mountain regions (Jura, Alps)

Flat regions and valleys

Application 2: Infer brain connections

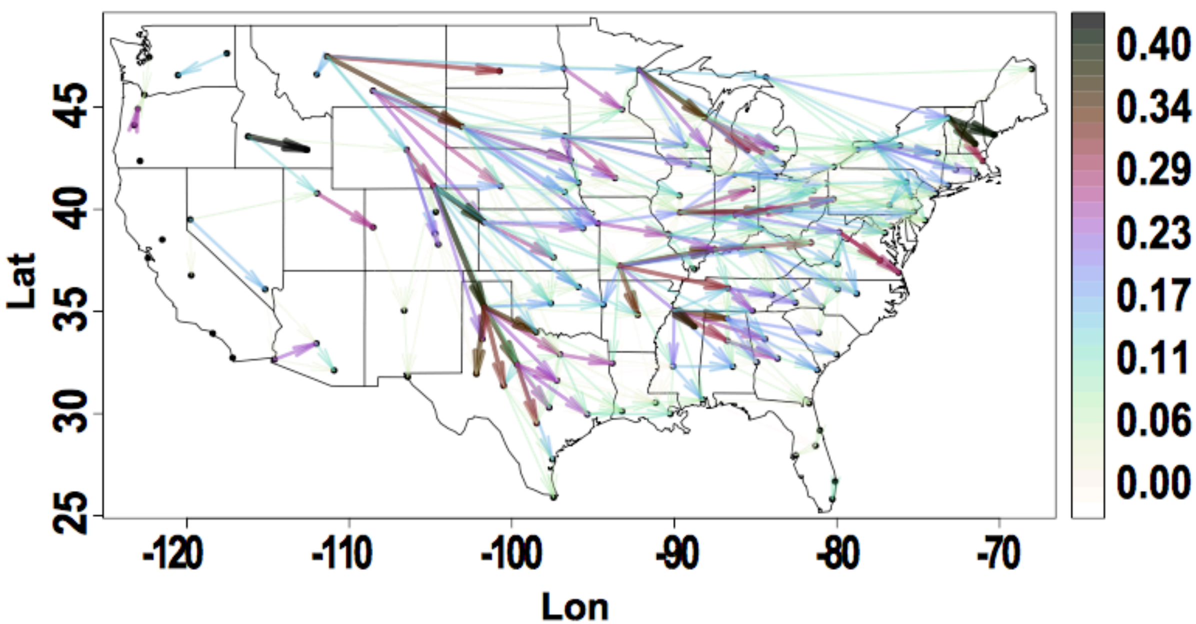
- Signals: time series recorded in MRI scan while the subjects are at rest



Parcellation of the brain obtained with the learned graph

Application 3: Learning wind directions

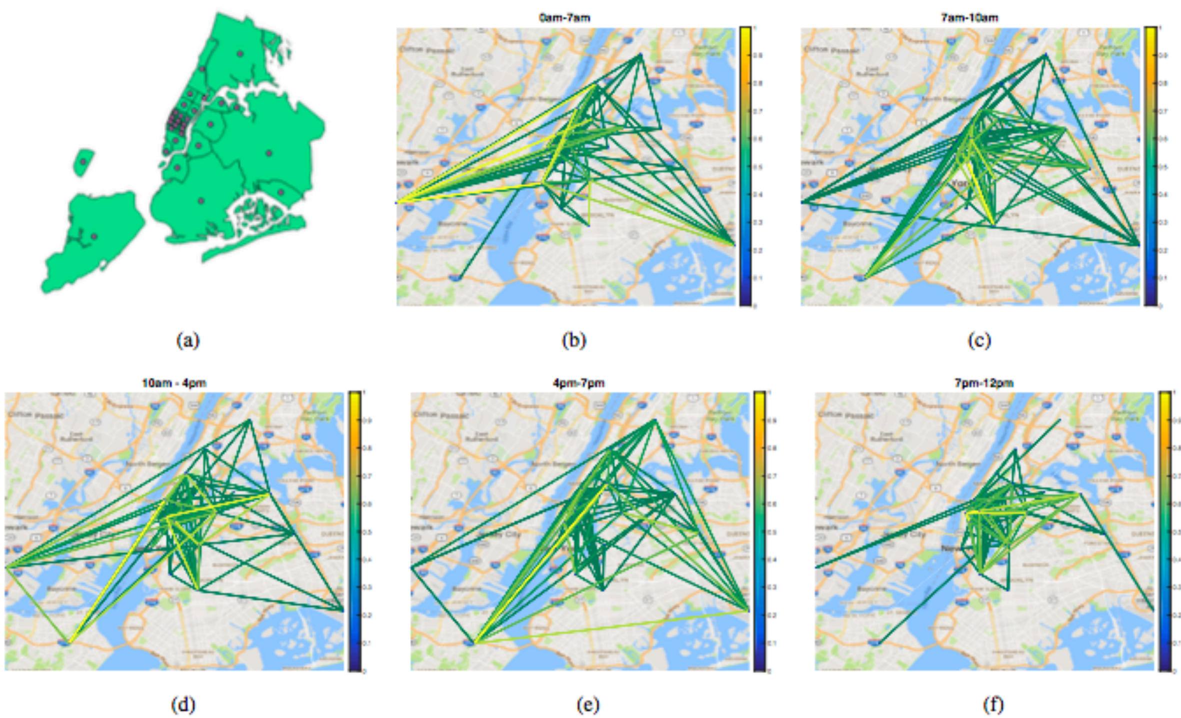
- Signals: daily temperature taken over 365 days at 150 cities in USA [20]



- The graph learned the predominant west-to-east and north-northwest-to-south-southeast directions

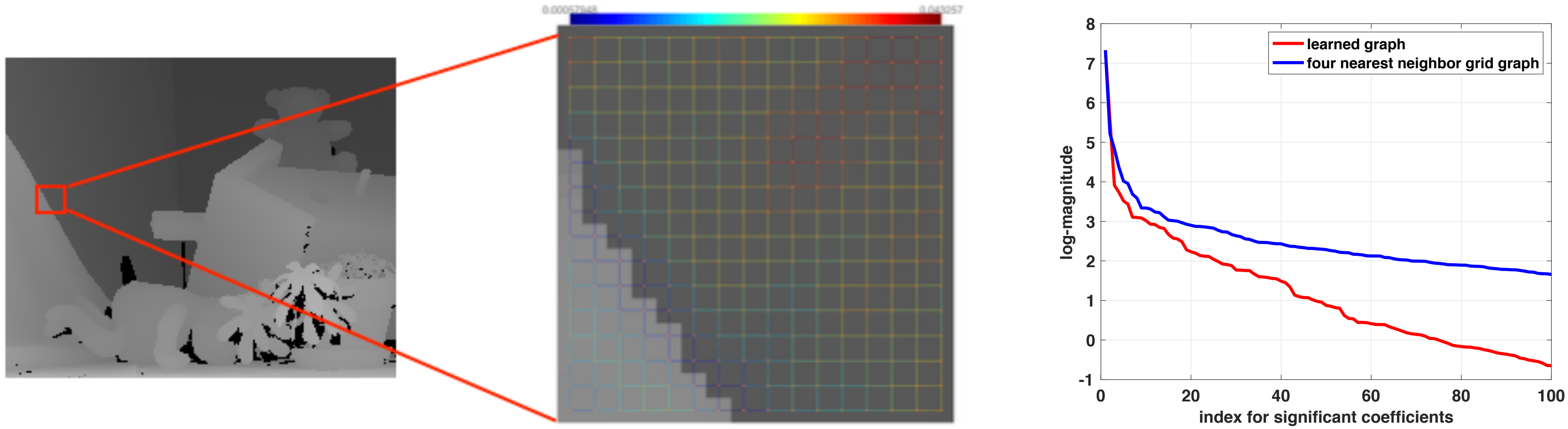
Application 4: Mining traffic behavior

- Learn the traffic patterns in NCY using a heat diffusion model [35]
  - nodes: centroids of taxi zones
  - signals: the hourly number of Uber pickups in each zone



Application 5: Image coding

- Graph learning can be used for efficient coding of image signals
- Additional constraints can be used to reduce the transmission cost of the graph



SDSC

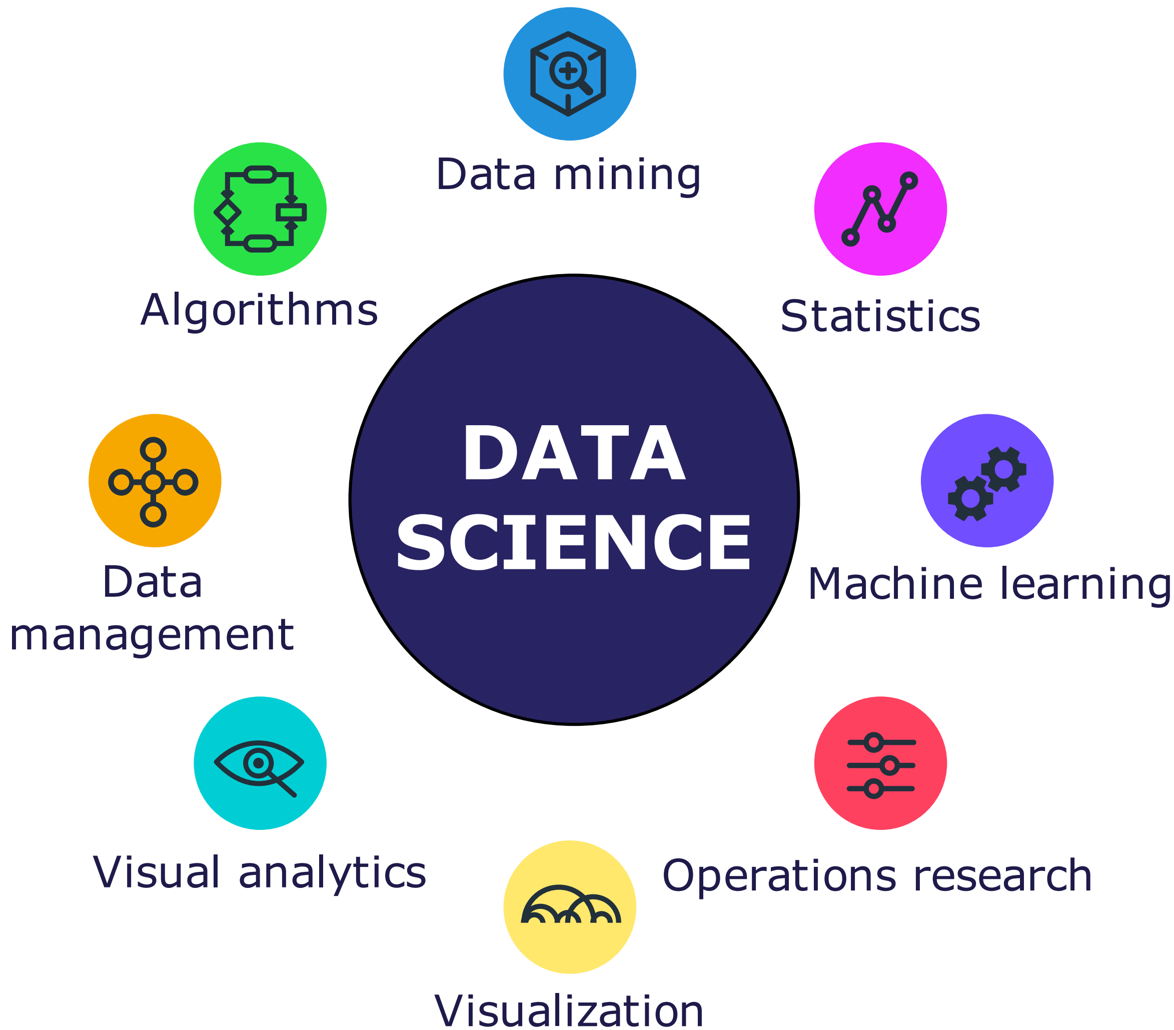
Comparison of GSP methods

Methods	Signal model	Assumption	Learning output	Edge direction	Inference
Dong (2015)	Global smoothness	Gaussian	Laplacian	Undirected	Signal-centric
Thanou (2016)	Heat diffusion	Sparsity	Laplacian	Undirected	Signal-centric
Pasdeloup (2015)	Diffusion by Adj. matrix	Stationary	Normalised Adj./ Laplacian	Undirected	Graph-centric
Mei (2015)	Time-varying	Dependent on previous states	Adj. matrix	Directed	Signal-centric

Take-home messages

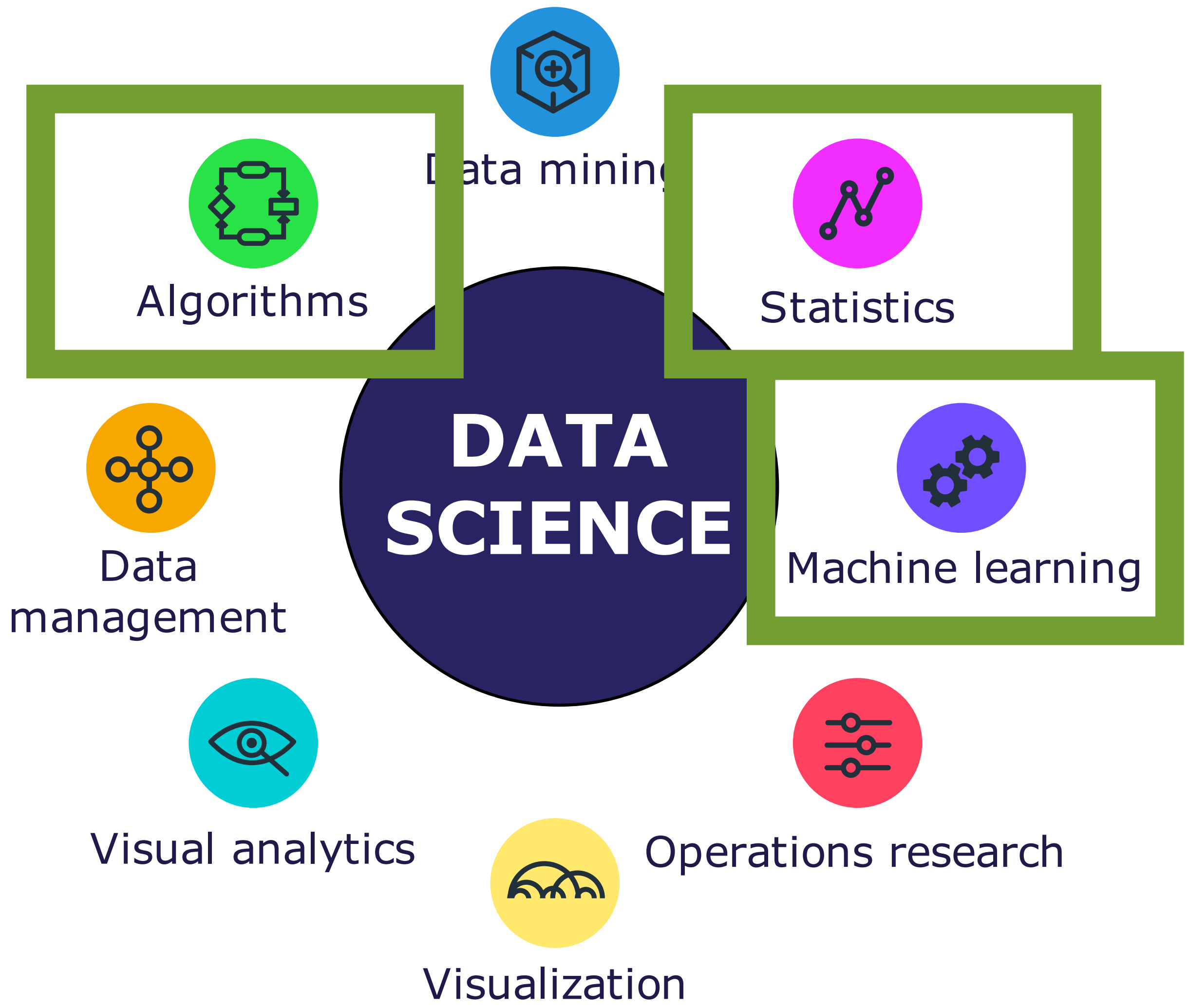
- Data is often structured
- **Graphs**: a flexible tool for modeling many real-world data
- If the structure if not known, **learn** it from the data
- The choice of the learning algorithm depends on many parameters (signal model, application)

Network inference: Powerful tool for Data Science

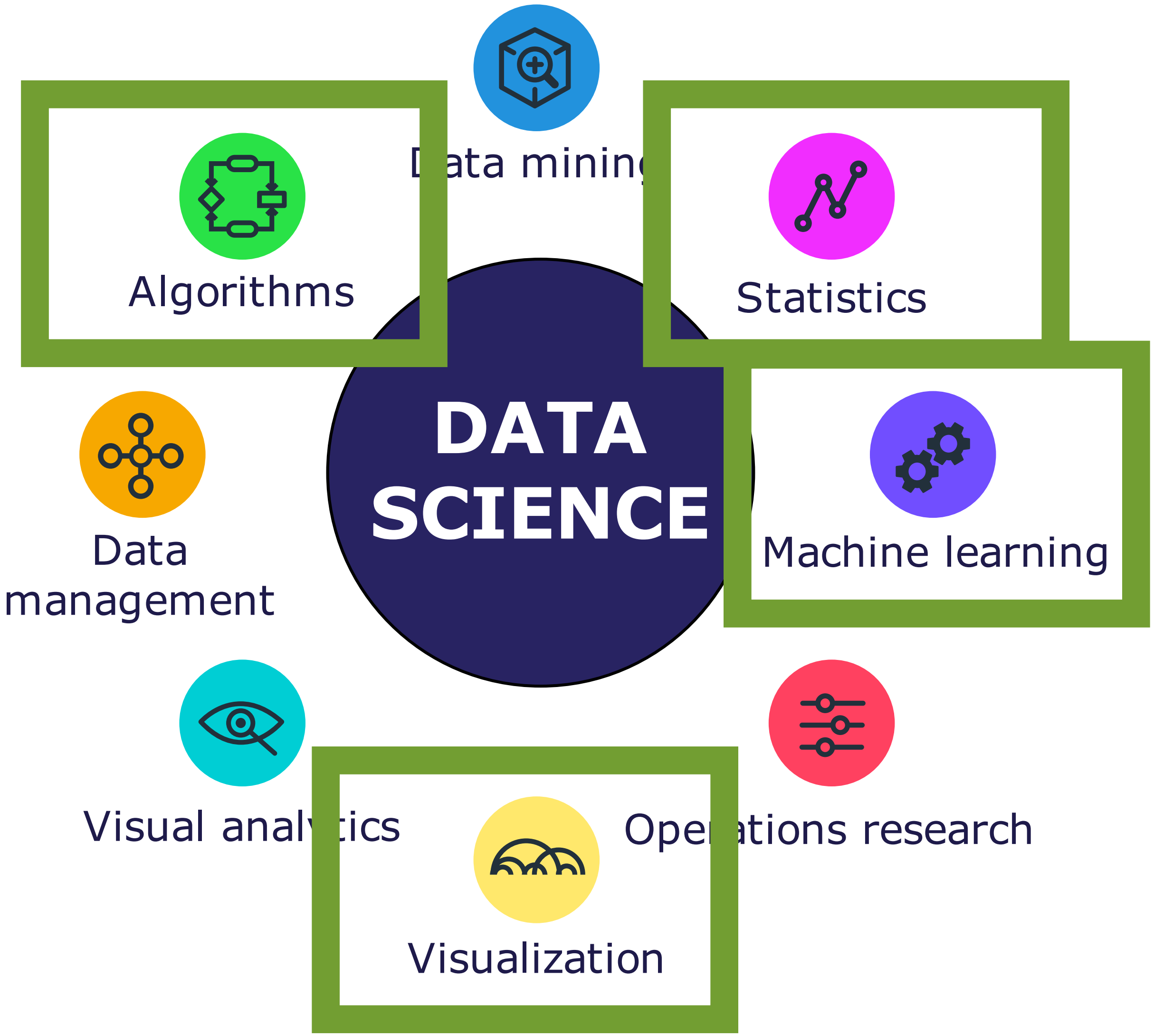


SDSC

Network inference: Powerful tool for Data Science



Network inference: Powerful tool for Data Science



SDSC

Overview paper

Learning Graphs from Data: A Signal Representation Perspective

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

The construction of a meaningful graph topology plays a crucial role in the effective representation, processing, analysis and visualization of structured data. When a natural choice of the graph is not readily available from the datasets, it is thus desirable to infer or learn a graph topology from the data. In this tutorial overview, we survey solutions to the problem of graph learning, including classical viewpoints from statistics and physics, and more recent approaches that adopt a graph signal processing (GSP) perspective. We further emphasize the conceptual similarities and differences between classical and GSP graph inference methods and highlight the potential advantage of the latter in a number of theoretical and practical scenarios. We conclude with several open issues and challenges that are keys to the design of future signal processing and machine learning algorithms for learning graphs from data.

I. Introduction

Modern data analysis and data processing tasks typically involve large sets of structured data, where the structure carries critical information about the nature of the data. One can find numerous examples of such datasets in a wide diversity of application domains, including transportation networks, social networks, computer networks, and brain networks. Typically, graphs are used as mathematical tools to describe the structure of such data. They provide a flexible way for representing relationships between data entities. Numerous signal processing and machine learning algorithms have been introduced in the past decade for analyzing structured data on *a priori* known graphs [1], [2]. However, there are often settings where the graph is not readily available, and the structure of the data has to be estimated in order to permit effective representation, processing, analysis or visualization of graph data. In this case, a crucial task is to infer a graph topology that describes the characteristics of the data observations, hence capturing the underlying relationship between these entities.

The problem of graph learning is the following: given M observations of N variables or data entities, represented in a data matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, and given some prior knowledge (e.g., distribution, data model,

*Authors contributed equally.

QUESTIONS?

References

1. B. Baingana and G. B. Giannakis. Tracking switching network topologies from propagating graph signals. In Graph Signal Processing Workshop, 2016.
2. B. Baingana and G. B. Giannakis. Tracking switched dynamic network topologies from information cascades. IEEE Transactions on Signal Processing, 65(4):985-997, 2017.
3. O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. The Journal of Machine Learning Research, 9:485-516, 2008.
4. S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero III. Learning sparse graphs under smoothness prior. arXiv:1609.03448, 2016.
5. D. I Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83-98, 2013.
6. S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In Proceedings of the International Conference on Machine Learning, 201-208, 2009.
7. A. P. Dempster. Covariance selection. Biometrics, 28(1):157-175, 1972.
8. X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Laplacian matrix learning for smooth graph signal representation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 3736-3740, 2015.
9. X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning Laplacian matrix in smooth graph signal representations. IEEE Transactions on Signal Processing, 64(23):6160-6173, 2016.

References

10. H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under structural and laplacian constraints. IEEE Journ. of Selected Topics in Sign. Process., 2017.
11. G. Fracastoro, D. Thanou, and P. Frossard. Graph transform learning for image compression. In Proceedings of the Picture Coding Symposium, 2016.
12. J.Friedman, T.Hastie, and R.Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432-441, 2008.
13. D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2):129-150, 2011.
14. C.-J. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In Advances in Neural Information Processing Systems 24, 2330-2338, 2011.
15. C. Hu, L. Cheng, J. Sepulcre, G. E. Fakhri, Y. M. Lu, and Q. Li. A graph theoretical regression model for brain connectivity learning of Alzheimer's disease. In Proceedings of the IEEE International Symposium on Biomedical Imaging, 616-619, 2013.
16. V. Kalofolias. How to learn a graph from smooth signals. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 920-929, 2016.
17. V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard. Learning time varying graphs. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2826-2830, 2017.
18. B. Lake and J. Tenenbaum. Discovering structure by learning sparse graph. In Proceedings of the Annual Cognitive Science Conference, 2010.

References

19.H. Petric Maretic, D. Thanou, and P. Frossard. Graph learning under sparsity priors. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 6523-6527, 2017.

20. J. Mei and J. M. F. Moura. Signal processing on graphs: Estimating the structure of a graph. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 5495-5499, 2015.

21. N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. Annals of Statistics, 34(3):1436-1462, 2006.

22. S. K. Narang and A. Ortega. Lifting based wavelet transforms on graphs. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 441-444, 2009.

23. B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat. Characterization and inference of weighted graph topologies from observations of diffused signals. IEEE Trans. on Signal and Inform. Process. over Networks, 2017.

24.B. Pasdeloup, M. Rabbat, V. Gripon, D. Pastor, and G. Mercier. Graph reconstruction from the observation of diffused signals. In Proceedings of the Annual Allerton Conference, 1386-1390, 2015.

25. P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional Ising model selection using l1-regularized logistic regression. Annals of Statistics, 38(3):1287-1319, 2010.

26. I. Rotondo, G. Cheung, A. Ortega, and H. E. Egilmez. Designing sparse graphs via structure tensor for block transform coding of images. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 571-574, 2015.

27. A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. IEEE Transactions on Signal Processing, 61(7):1644-1656, 2013.

References

30. S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro. Network topology inference from spectral templates. IEEE Trans. on Signal and Inform. Process. over Networks, 2017.
31. Y. Shen, B. Baingana, and G. B. Giannakis. Nonlinear structural vector autoregressive models for inferring effective brain network connectivity. arXiv:1610.06551, 2016.
32. Y. Shen, B. Baingana, and G. B. Giannakis. Kernel-based structural equation models for topology identification of directed networks. IEEE Transactions on Signal Processing, 65(10):2503-2516, 2017.
33. M. Slawski and M. Hein. Estimation of positive definite M-matrices and structure learning for attractive Gaussian Markov random fields. Linear Algebra and its Applications, 473:145-179, 2015.
34. D. Thanou, D. I Shuman, and P. Frossard. Learning parametric dictionaries for signals on graphs. IEEE Transactions on Signal Processing, 62(15):3849-3862, 2014.
35. D. Thanou, X. Dong, D. Kressner, and P. Frossard. Learning heat diffusion graphs. IEEE Trans. on Signal and Inform. Process. over Networks, 2017.
36. Y. Yankelevsky and M. Elad. Dual graph regularized dictionary learning. IEEE Transactions on Signal and Information Processing over Networks, 2(4):611-624, 2016.
37. X. Zhang, X. Dong, and P. Frossard. Learning of structured graph dictionaries. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 3373-3376, 2012.

References

38. N. Du, L. Song, and M. Yuan, "Learning networks of heterogeneous influence," in Advances in Neural Information Processing Systems 25, 2012, pp. 2780–2788
39. P. Netrapalli and S. Sanghavi, "Learning the graph of epidemic cascades," in Proceedings of the ACM SIGMET- RICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, 2012, pp. 211–222.
40. B. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi, "Trace complexity of network inference," in Proceedings of the IEEE International Symposium on Information Theory, 2013, pp. 491–499.
41. V. Gripon and M. Rabbat, "Reconstructing a graph from path traces," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 2488–2492.
42. H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Scholkopf, "Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm," in Proceedings of the International Conference on Machine Learning, 2014.