ELL 888: Graph ML a Big Picture

Prof. Sandeep Kumar

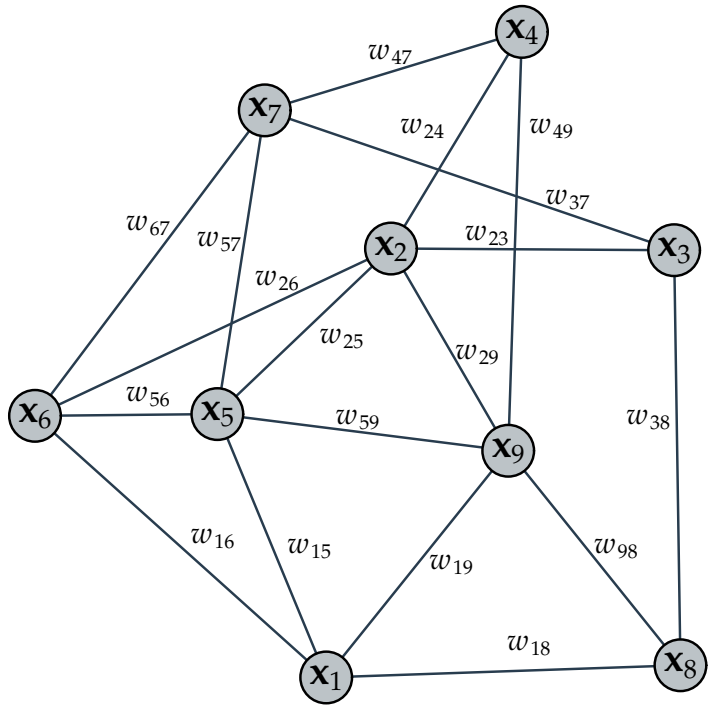
Indian Institute of Technology Delhi

January 18, 2022

Outline

Outline

Representing Knowledge Through Graphical Models



Graph is a simple mathematical structure of form $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$

- **Nodes** $\mathcal{V} = \{1, 2, 3, \ldots, p\}$ correspond to the entities (variables), and
- **Edges** $\mathcal{E} = \{(1,2), (1,3), \ldots, (i,j), \ldots\}$ encode the relationships between entities ($\{i,j\}$) (dependencies between the variables),
- **Weights** $\{w_{12}, w_{13}, \ldots, w_{ij}, \ldots\}$ encode the relationships strengths.

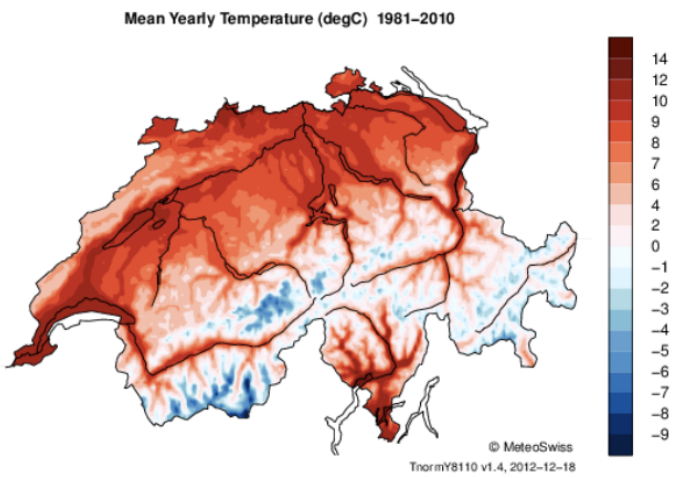
Importance of Graphical Models

- Graphs are intuitive way of representing and visualising the **relationships**.
- Graphs allow us to abstract out the **independence** relationships between the variables from the details of their **parametric** forms.
- Graphs can serve as the **final goal** (e.g., connectivity information) as well as an **intermediary tool** with that some other goals are aimed, e.g., classification, prediction, smoothing, structural inference, Graph CNN applications, Graph signal processing applications.
- Graphs offer a **language** through which different **disciplines** can seamlessly **interact** with each other.
- Graph-based approaches with big data and machine learning are driving the current research frontiers.

Applications: **High-dimensional probabilistic modelling, Machine Learning, Networks, Bioinformatics, Deep learning, Graph signal processing, Statistical physics.**

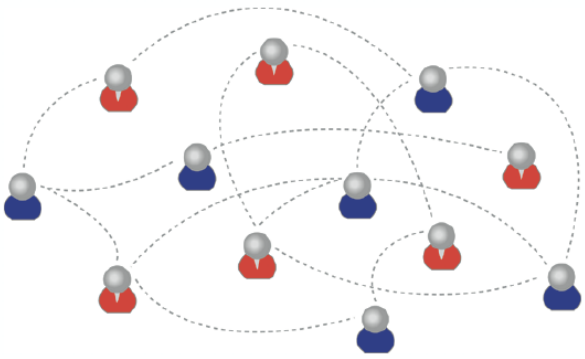
$$\text{Graphical Models} = \text{Statistics} \times \text{Graph Theory} \times \text{Optimization} \times \text{Engineering}$$

Structured (Irregular) Data



Temperature data

Traffic data

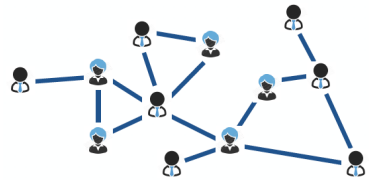


Social network data

Neuroimaging data

Networks are Pervasive

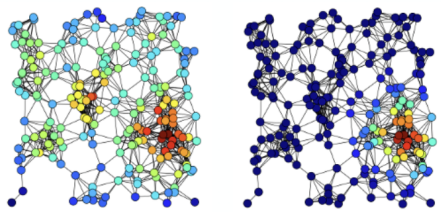
traffic network



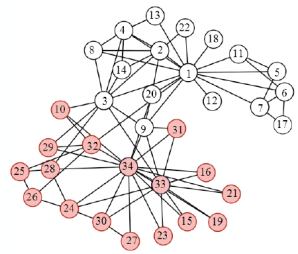
social network

brain network

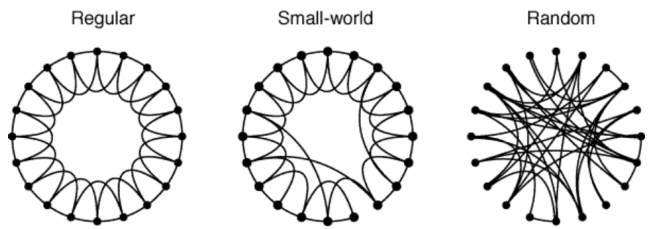
network centrality



community detection



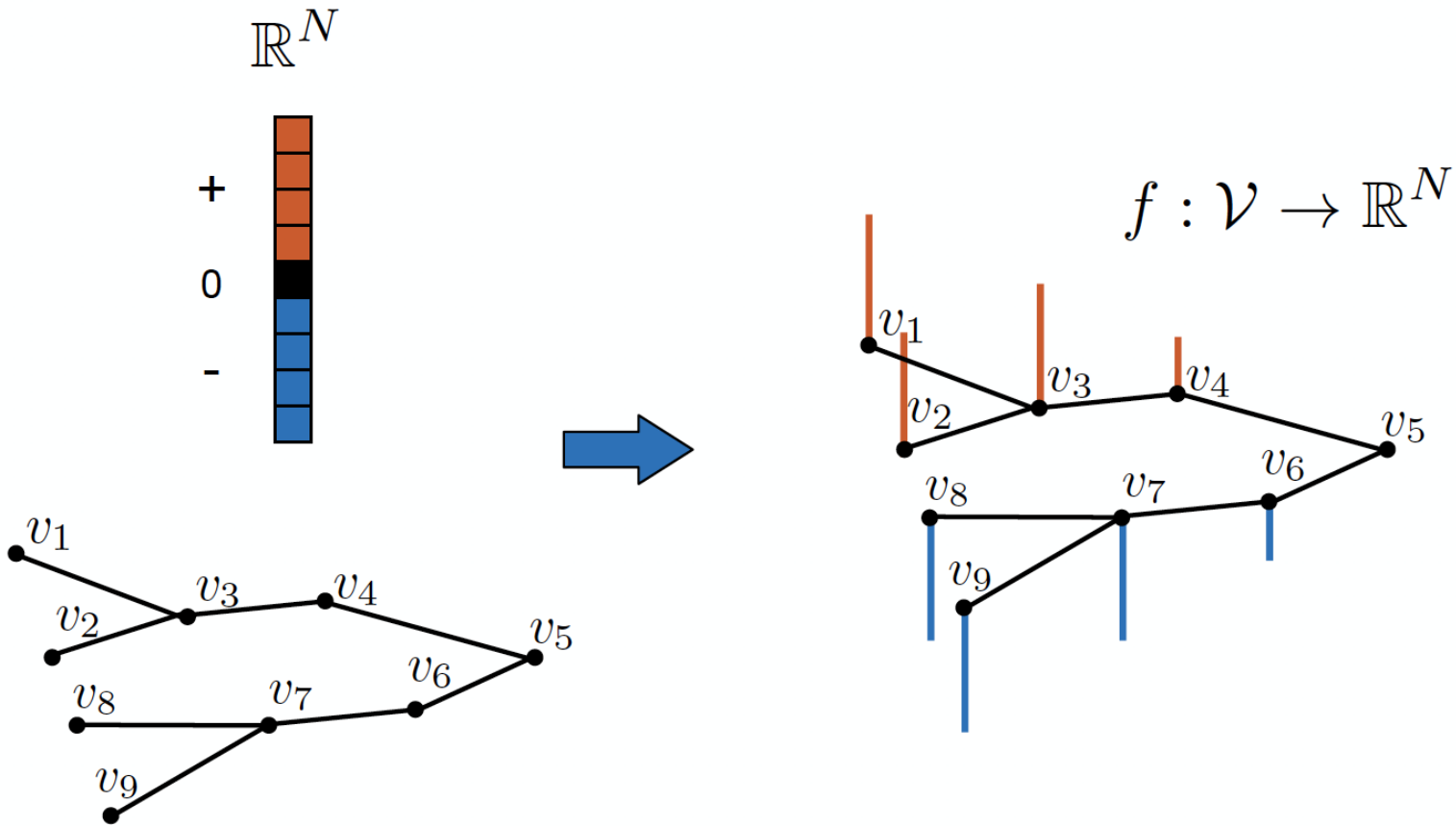
random graph models



Network Science: It's an all inclusive field of study

[Barabási et al., 2016]Barabási, A. L. (2016). Network science. Cambridge university press.

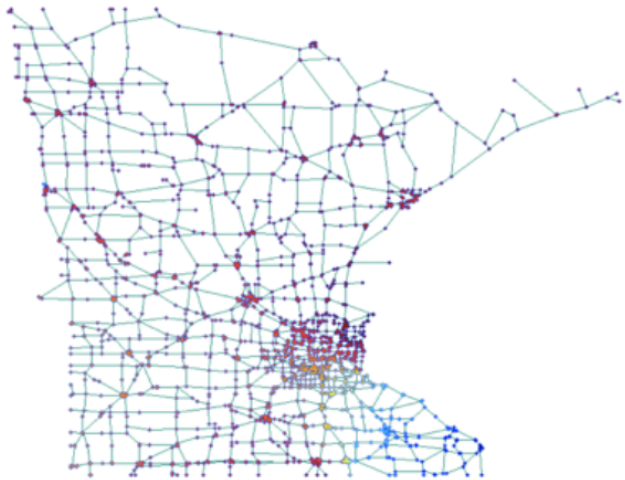
Graph structured data (graph signal)



Graph structured data : Traffic data

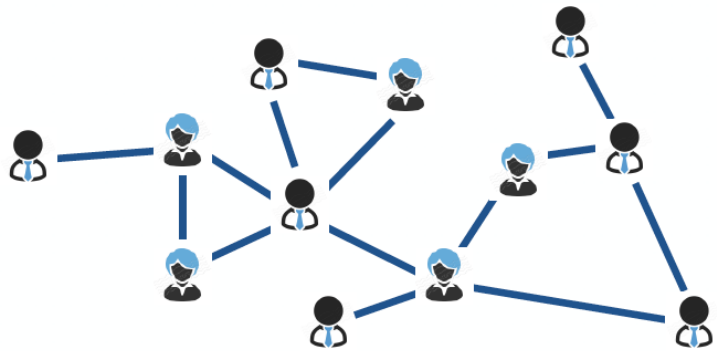
- nodes: road junctions
- edges: road connections

Graph structured data : Traffic



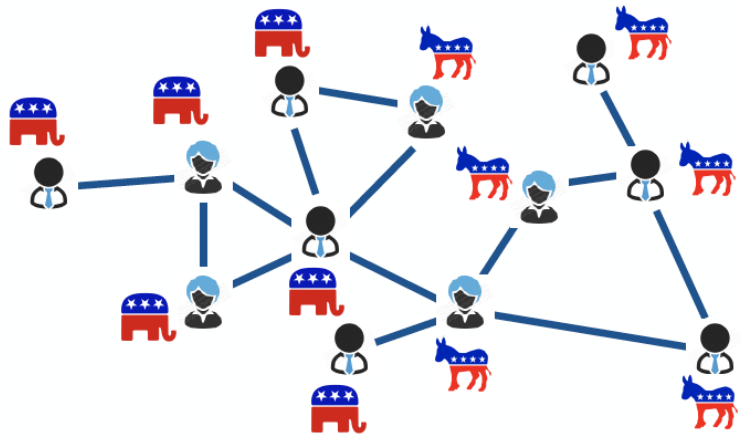
- nodes: road junctions
- edges: road connections
- signal: traffic congestion at junctions

Graph structured data : Social network



- nodes: individuals
- edges: friendships, preferences, behavioural dynamics

Graph structured data : Social network data

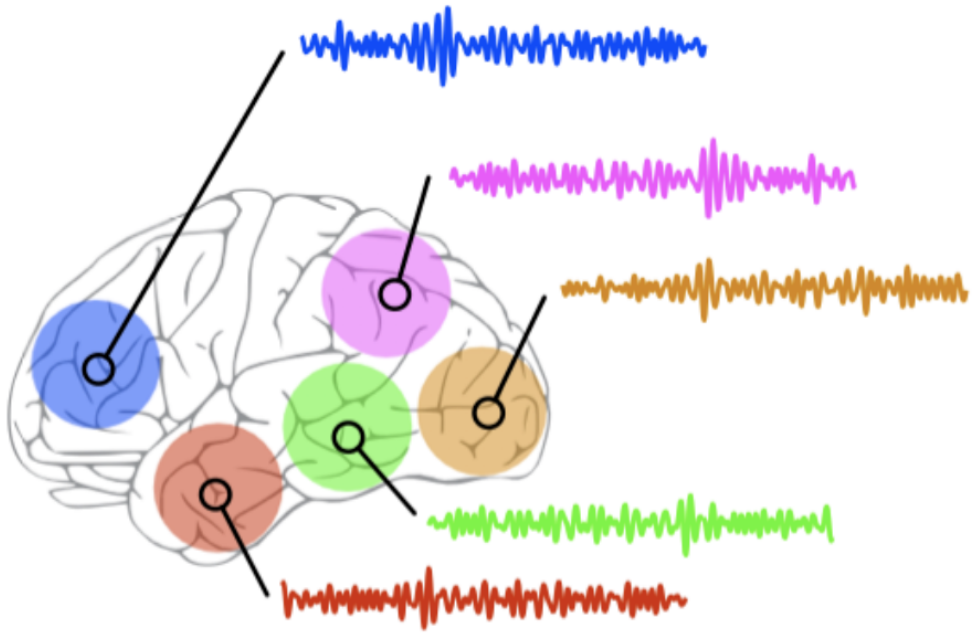


- nodes: individuals
- edges: friendships, association
- signals: preferences, behavioural dynamics, political view

Graph structured data : Brain network

- nodes: brain regions
- edges: structural connectivity between brain regions

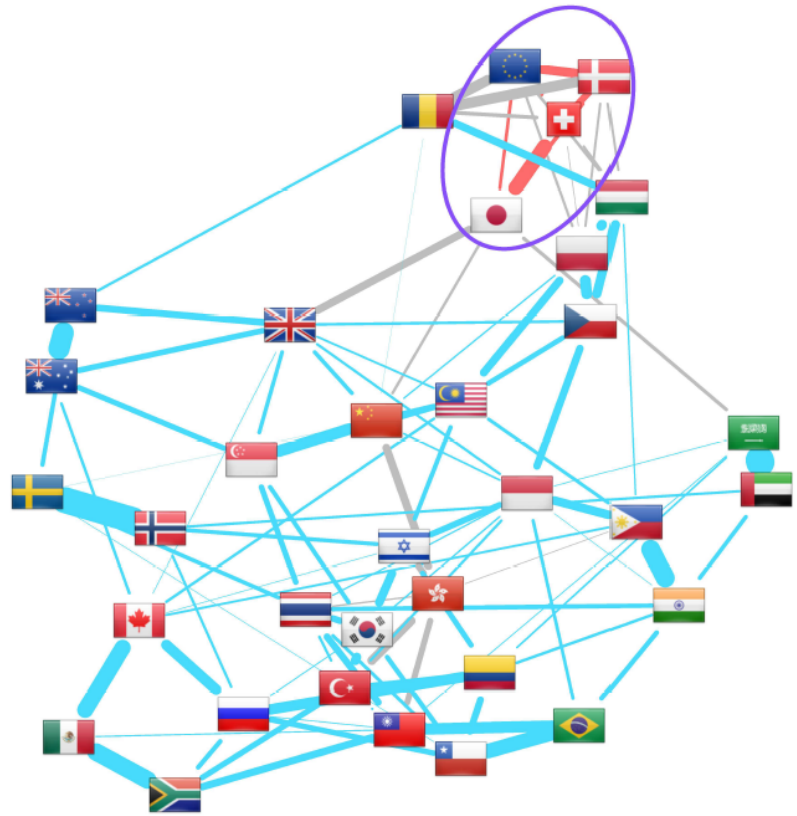
Graph structured data : Brain network



- nodes: individuals
- edges: friendships, association
- signals: blood-oxygen-level dependent signals

Graph structured data : Financial network

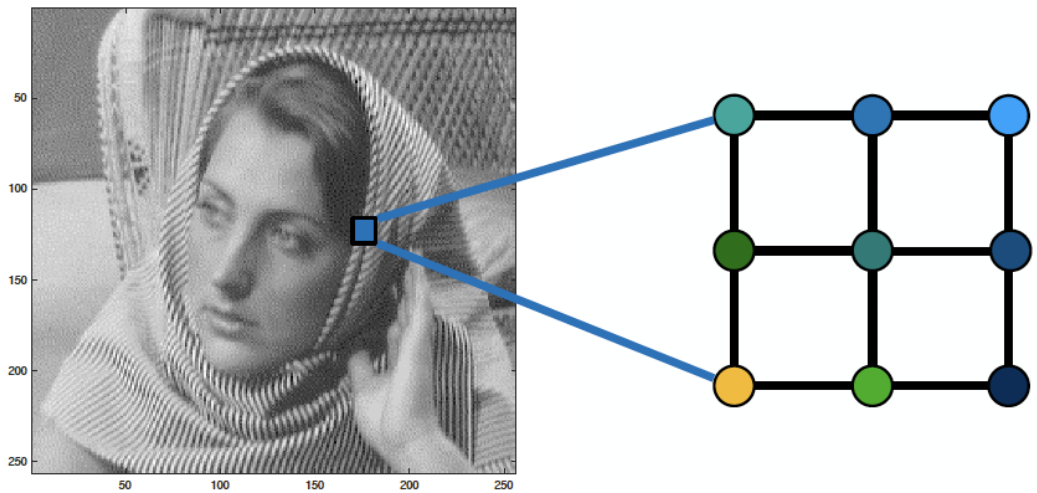
(a) Currency network estimated using data from Feb. 1st to May 1st 2019 (pre-covid).



(b) Currency network estimated using data from Feb. 3rd to May 1st 2020 (including covid).

- nodes: country
- edges: economic connections
- signals: volumes of trades, commodity exchange

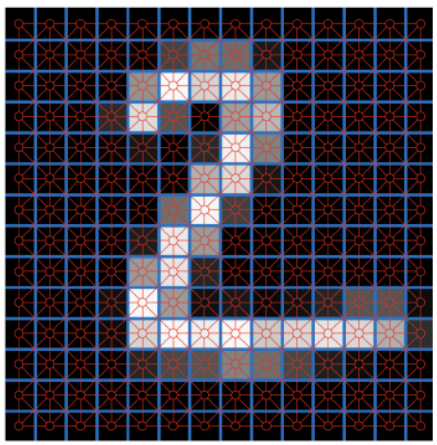
Graph structured data : Image network



- nodes: pixels
- edges: spatial proximity between pixels
- signals: pixel values

Outline

Learning with Graph Structured Data

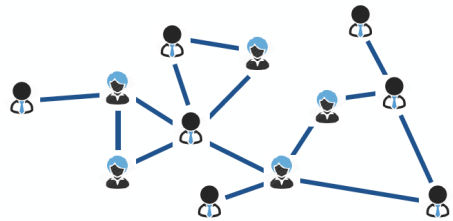


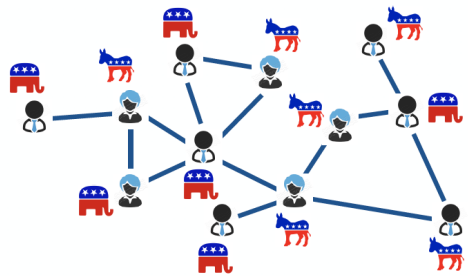
is it a 2?
is it a 4?

condition?
no condition?

Identification Problem.

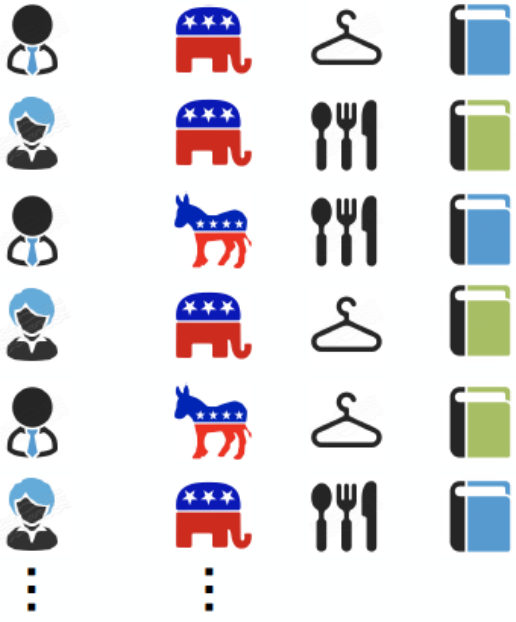
Learning with Graph Structured Data

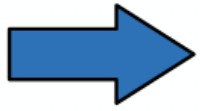


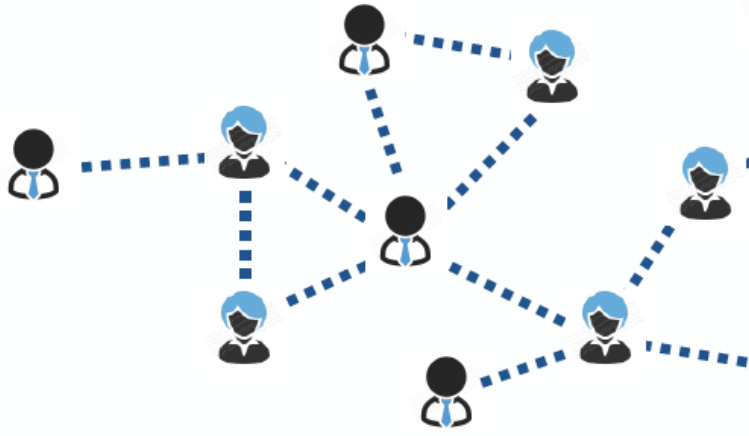


Node-wise classification problem (Semi-supervised learning)

Graph Learning from Data

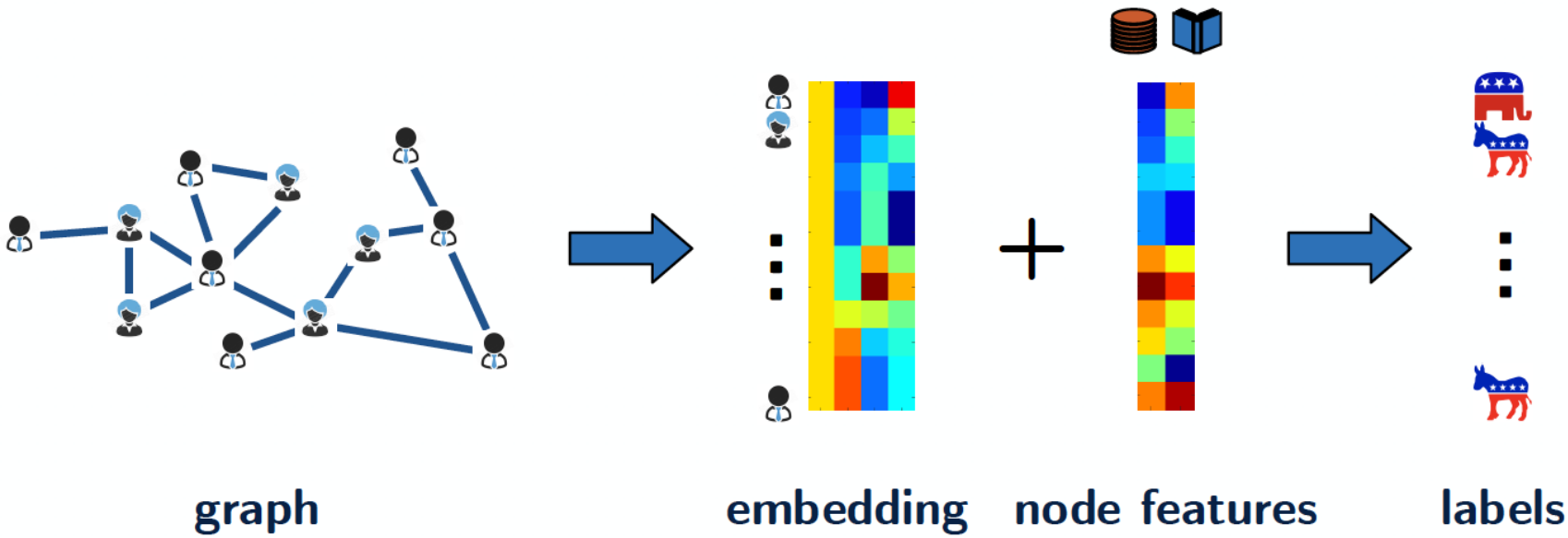






learning graph structure from data

Graph Based Learning



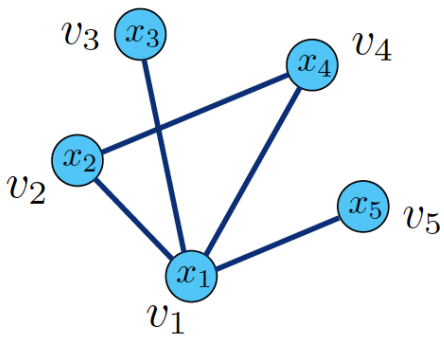
Need for new models that directly incorporate structure in data analysis

- Graph signal processing (GSP)
- Graph neural networks (GNN)
- Manifold regularization/ Graph Regularization

Outline

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



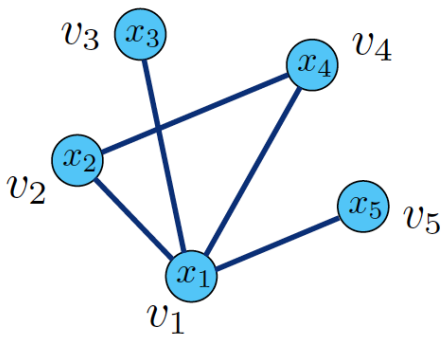
PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



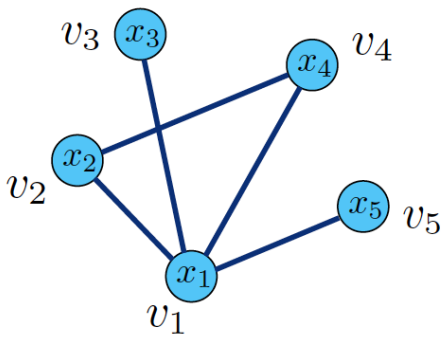
PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



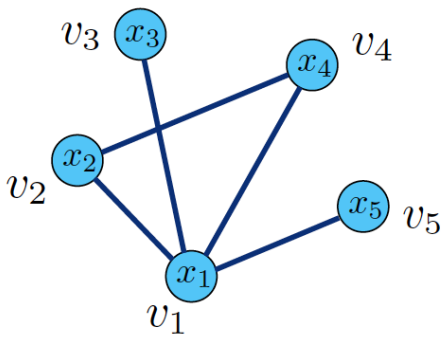
PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



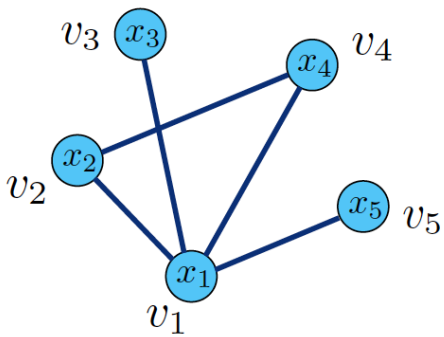
PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



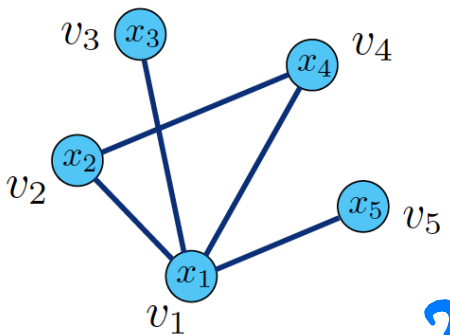
PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Types of Graphical Models

- Models encoding **pairwise dependencies**: simple and intuitive.
  - **Sample correlation** based graph: two entities i and j are connected if the pairwise coorelation is greater than certain threshold ($\rho_{ij} > \alpha$,) and vice versa.
  - **Similarity function** (e.g., Gaussian RBF) based graph, k-nn graphs, $[\mathbf{W}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2)$, if $i \neq 0$, and $\mathbf{W}_{ii} = 0$. Here, the scaling parameter σ^2 controls how rapidly the affinity $[\mathbf{W}]_{ij}$ falls off with distances between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Models based on some **assumption** on the data: $\mathbf{x} \sim \mathcal{F}(\mathcal{G})$.
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs, smooth signals defined over graphs).
  - **Probabilistic graphical models (PGM)**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov random field).



PGM captures the conditional independnce property (Markov property)

$$(i,j) \notin \mathcal{E} \iff x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

$$(4,5) \notin \mathcal{E} \implies x_4 \perp\!\!\!\perp x_5 | \mathbf{x}_{/x_4,x_5}$$

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\boldsymbol{\Theta}) = h(\mathbf{x})\exp\{\langle\boldsymbol{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\boldsymbol{\Theta})\}$$

- $\phi(\mathbf{x})$ is the sufficient statistic and Θ is the associated parameter.
- The **nonzero pattern** of Θ directly encodes the Markov property.

$$\Theta_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \Theta)$, if its density follows:

$$P(\mathbf{x};\Theta) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\Theta\mathbf{x}\right)$$

  - $\phi(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\Theta = [\Theta_{11}, \Theta_{12}, \Theta_{13}, \ldots, \Theta_{ij}, \ldots]$
  - Estimation of Θ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\mathbf{\Theta}) = h(\mathbf{x})\exp\left\{\langle\mathbf{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\mathbf{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\mathbf{\Theta}$ is the associated parameter.
- The **nonzero pattern** of Θ directly encodes the Markov property.

$$\Theta_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \Theta)$, if its density follows:

$$P(\mathbf{x};\Theta) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\Theta\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\Theta = [\Theta_{11}, \Theta_{12}, \Theta_{13}, \ldots, \Theta_{ij}, \ldots]$
  - Estimation of Θ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\mathbf{\Theta}) = h(\mathbf{x})\exp\left\{\langle\mathbf{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\mathbf{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\mathbf{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\mathbf{\Theta}$ directly encodes the Markov property.

$$\mathbf{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(0, \Theta)$, if its density follows:

$$P(\mathbf{x};\Theta) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\Theta\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\Theta = [\Theta_{11}, \Theta_{12}, \Theta_{13}, \ldots, \Theta_{ij}, \ldots]$
  - Estimation of Θ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\mathbf{\Theta}) = h(\mathbf{x})\exp\left\{\langle\mathbf{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\mathbf{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\mathbf{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\mathbf{\Theta}$ directly encodes the Markov property.

$$\mathbf{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- GMRF:
  - A random vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ is called a GMRF with parameters $(0, \Theta)$, if its density follows:

$$P(\mathbf{x};\Theta) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\Theta\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\Theta = [\Theta_{11}, \Theta_{12}, \Theta_{13}, \ldots, \Theta_{ij}, \ldots]$
  - Estimation of Θ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\boldsymbol{\Theta}) = h(\mathbf{x})\exp\left\{\langle\boldsymbol{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\boldsymbol{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\boldsymbol{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\boldsymbol{\Theta}$ directly encodes the Markov property.

$$\boldsymbol{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \Theta)$, if its density follows:

$$P(\mathbf{x};\Theta) = \frac{\det(\Theta)^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\Theta\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\Theta = [\Theta_{11}, \Theta_{12}, \Theta_{13}\ldots, \Theta_{ij}, \ldots]$
  - Estimation of Θ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\boldsymbol{\Theta}) = h(\mathbf{x})\exp\left\{\langle\boldsymbol{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\boldsymbol{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\boldsymbol{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\boldsymbol{\Theta}$ directly encodes the Markov property.

$$\boldsymbol{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \boldsymbol{\Theta})$, if its density follows:

$$P(\mathbf{x};\boldsymbol{\Theta}) = \frac{\det(\boldsymbol{\Theta})^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\boldsymbol{\Theta}\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_{11}, \boldsymbol{\Theta}_{12}, \boldsymbol{\Theta}_{13}\ldots, \boldsymbol{\Theta}_{ij}, \ldots]$
  - Estimation of $\boldsymbol{\Theta}$ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\boldsymbol{\Theta} = \boldsymbol{\Theta}^\top, \boldsymbol{\Theta} \in \mathbb{S}^p_{++}, \boldsymbol{\Theta}_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\boldsymbol{\Theta}) = h(\mathbf{x})\exp\left\{\langle\boldsymbol{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\boldsymbol{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\boldsymbol{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\boldsymbol{\Theta}$ directly encodes the Markov property.

$$\boldsymbol{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \boldsymbol{\Theta})$, if its density follows:

$$P(\mathbf{x};\boldsymbol{\Theta}) = \frac{\det(\boldsymbol{\Theta})^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\boldsymbol{\Theta}\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_{11}, \boldsymbol{\Theta}_{12}, \boldsymbol{\Theta}_{13}\ldots., \boldsymbol{\Theta}_{ij}, \ldots]$
  - Estimation of $\boldsymbol{\Theta}$ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\Theta = \Theta^\top, \Theta \in \mathbb{S}^p_{++}, \Theta_{ij} \in \mathbb{R}\right\}$.

Exponential Families of Distribution for PGM

- Many useful graphical models are naturally viewed as **exponential families** of distribution:

$$P(\mathbf{x};\mathbf{\Theta}) = h(\mathbf{x})\exp\left\{\langle\mathbf{\Theta},\boldsymbol{\phi}(\mathbf{x})\rangle - \log A(\mathbf{\Theta})\right\}$$

- $\boldsymbol{\phi}(\mathbf{x})$ is the sufficient statistic and $\mathbf{\Theta}$ is the associated parameter.
- The **nonzero pattern** of $\mathbf{\Theta}$ directly encodes the Markov property.

$$\mathbf{\Theta}_{ij} = 0 \iff (i,j) \notin \mathcal{E} : x_i \perp\!\!\!\perp x_j | \mathbf{x}_{/x_i,x_j}$$

- **Examples**: Gaussian Markov Random field (GMRF), Boltzman machine, Ising model, Potts model, and Maximum entropy model.
- **GMRF**:
  - A random vector $\mathbf{x} = (x_1, x_2, ..., x_p)$ is called a GMRF with parameters $(\mathbf{0}, \mathbf{\Theta})$, if its density follows:

$$P(\mathbf{x};\mathbf{\Theta}) = \frac{\det(\mathbf{\Theta})^{1/2}}{(2\pi)^{p/2}}\exp\left(-\frac{1}{2}\mathbf{x}^\top\mathbf{\Theta}\mathbf{x}\right)$$

  - $\boldsymbol{\phi}(\mathbf{x}) = [x_1^2, x_1x_2, x_1x_3, \ldots, x_ix_j, \ldots]$, $\mathbf{\Theta} = [\mathbf{\Theta}_{11}, \mathbf{\Theta}_{12}, \mathbf{\Theta}_{13}\ldots., \mathbf{\Theta}_{ij}, \ldots]$
  - Estimation of $\mathbf{\Theta}$ is simply the **precision (inverse covariance)** matrix estimation problem: $\mathcal{S}_\Theta = \left\{\mathbf{\Theta} = \mathbf{\Theta}^\top, \mathbf{\Theta} \in \mathbb{S}^p_{++}, \mathbf{\Theta}_{ij} \in \mathbb{R}\right\}$.

Graph and its Matrix Representation

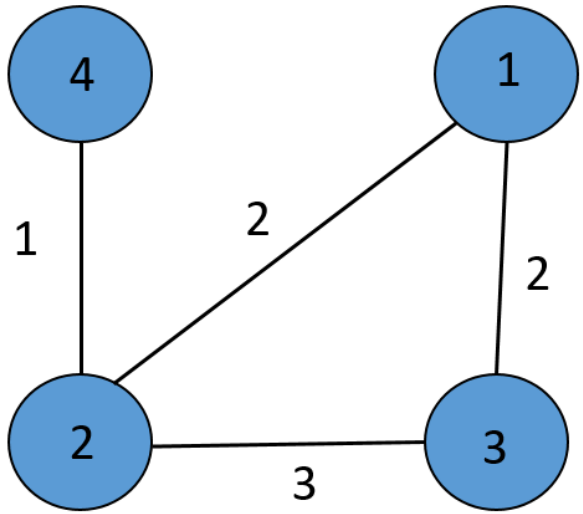


Figure 1: $\mathcal{V} = \{1, 2, 3, 4\}$, $\mathcal{E} = \{(1, 2), (1, 3), (2, 3), (2, 4)\}$

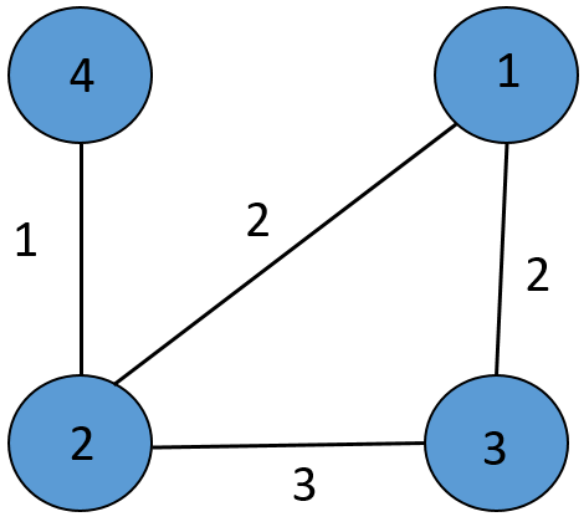
$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 0 & 3 & 1 \\ 2 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Connectivity matrix $\mathbf{C}$, **(Weight) Adjacency** matrix $\mathbf{A}$, **Degree** matrix $\mathbf{D}$.

Laplacian Matrix

- The adjacency matrix $\mathbf{W} = [w_{ij}]$ and the Laplacian matrix $\mathbf{L}$ both are symmetric, and represent the same weighted graph related $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- $\mathbf{L}$ and $\mathbf{W}$ have different mathematical properties, e.g., $\mathbf{L}$ is PSD, while $\mathbf{W}$ is not.

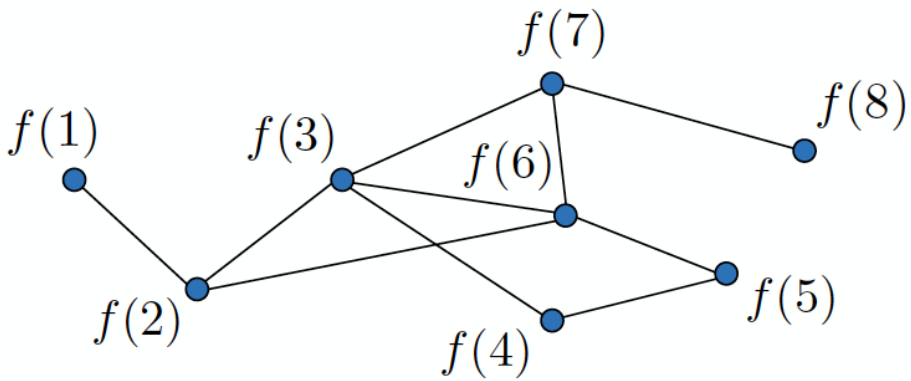
Laplacian matrix $\mathbf{L}$.



$$\mathbf{L} = \mathbf{D} - \mathbf{W} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 0 & 3 & 1 \\ 2 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -2 & -2 & 0 \\ -2 & 6 & -3 & -1 \\ -2 & -3 & 5 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

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

Laplacian Matrix for Physically inspired Model



- Consider an unweighted graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$.
- Let $f : \mathcal{V} \to \mathbb{R}^N$ captures the signal defined over graph.
- $\mathbf{L}$ is the Laplacian matrix.

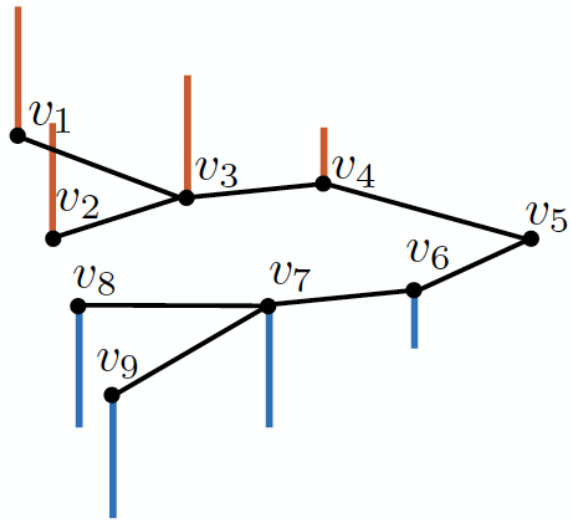
$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}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

$$\mathbf{L}f = \sum_{j=1}^{N} \mathbf{W}_{ij}(f(i) - f(j))$$

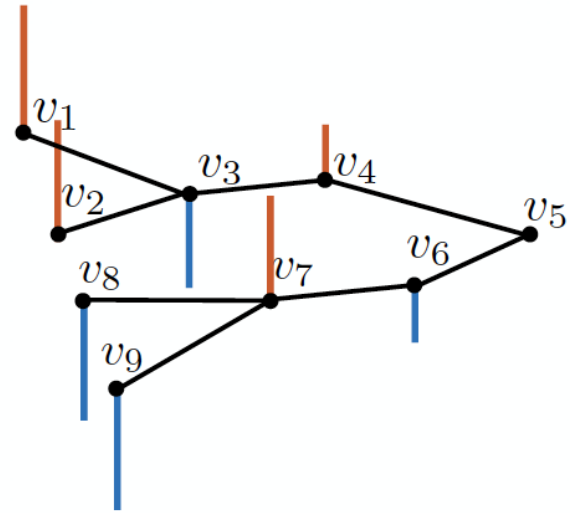
$$\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}^T \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 &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

$$f^\top \mathbf{L} f = \sum_{j=1}^{N} \mathbf{W}_{ij}(f(i) - f(j))^2,$$

A measure of smoothness.



$$\mathbf{L}f = \sum_{j=1}^{N} \mathbf{W}_{ij}(f(i) - f(j)) = 1$$



$$f^\top \mathbf{L}f = \sum_{j=1}^{N} \mathbf{W}_{ij}(f(i) - f(j))^2 = 21$$

Laplacian matrix play a fundamental role in understanding of the physically defined graphical models:

- In statistical mechanics: $\mathbf{E}(f) = \sum_{j=1}^{N} \mathbf{W}_{ij}(f(i) - f(j))^2$ is also called energy of the system.
- A low energy system is stable while a high energy system is unstable.
- Laplace matrix also approximates the Laplacian operator.
- Laplace can also capture the notion of frequency.

Graphical Models: Major Research Agenda

Using Graphs for Learning and Inference Task:

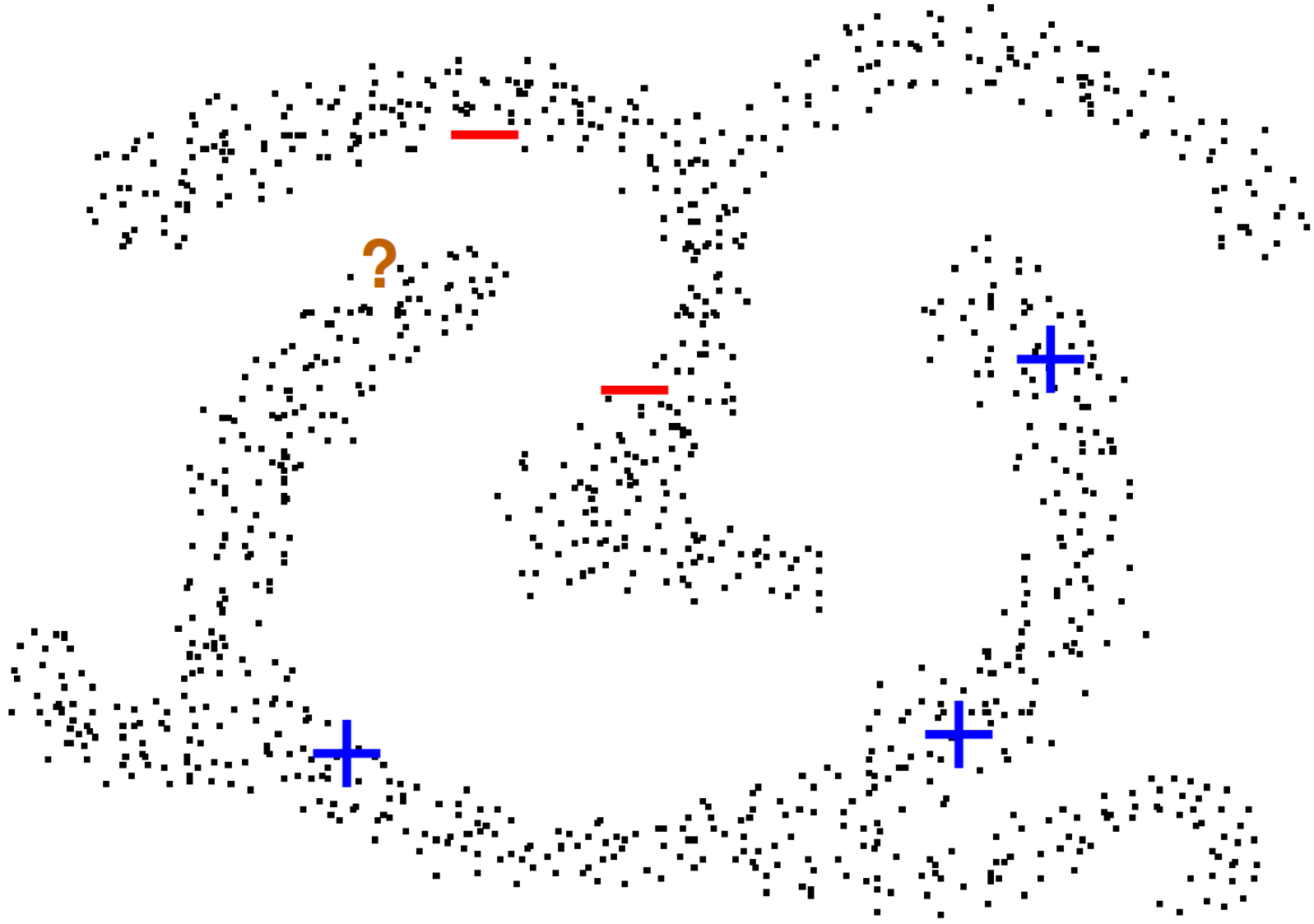
- Semi-supervised Learning, transductive learning, graph regularization based algorithms.
- Graph signal processing: this branch attempts to integrate the widely popular and well-tested signal processing algorithms within the graph domain.
- Network science: using network propertis, e.g., small world, network centrality, eigenvector centrality, and other properties to study the behaviour of complex networks

Learning Graph from Data is the Main Task:

- Probabilistic graph learning
- Graph learning under practical assumptions, e.g., smoothness, diffusivity, and heterogeneity, and etc.

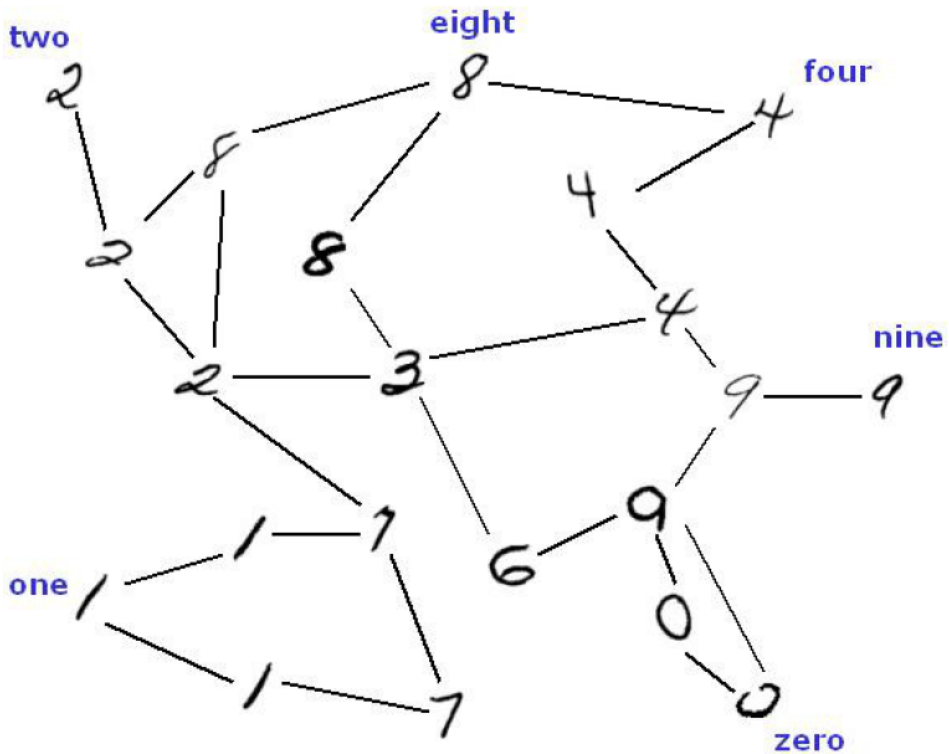
Outline

Classification using unlabelled data



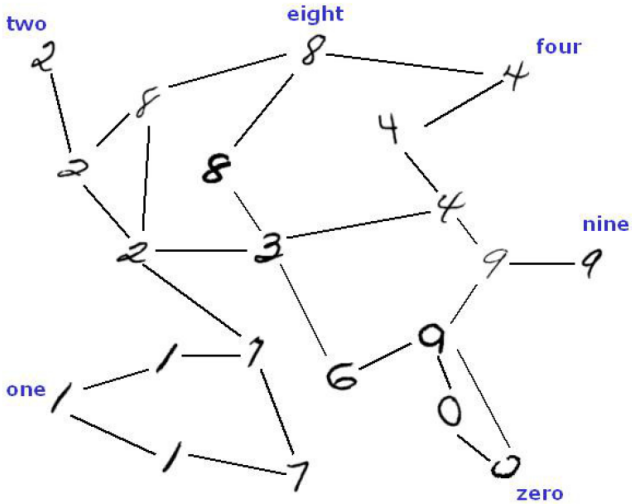
Assumption: there is information in the data distribution, unlabelled data is also informative.

Graph-based Learning: Labeled and Unlabeled Data as a Graph

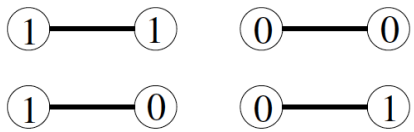


- **Idea**: Construct a graph connecting similar data points
- Let the hidden/observed labels be random variables on the nodes of this graph (i.e. the graph is an MRF)
- **Intuition**: **Similar** data points have similar **labels**
- Information **propagates** from labelled data points Graph encodes intuition

A graph of hand written digits data



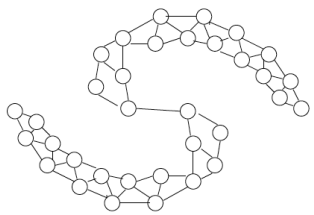
- **nodes**: instances in $L \cup U$. Binary labels $\mathbf{y} \in \{0,1\}^N$
- **edges**: weight matrix $\mathbf{W}$ assumed given.
- Energy $\mathbf{E} = \sum_{i,j=1}^{N} \mathbf{W}_{ij}(y_i - y_j)^2$



happy, low energy

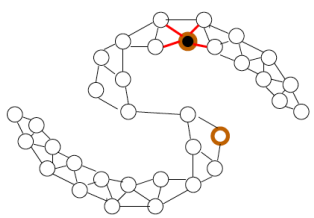
unhappy, high energy

With no labelled data, then $\mathbf{y} = \mathbf{1}$ or $\mathbf{y} = \mathbf{0}$ is a min energy configuration:

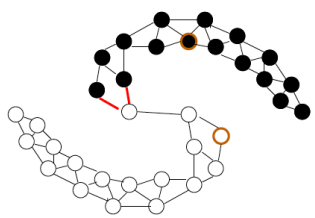


energy=0

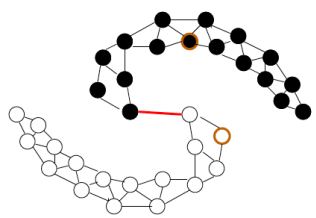
Conditioned on labeled data:



energy=4



energy=2



energy=1

Gaussian Random Fields: Undirected Graph

$$
\begin{aligned}
p(\mathbf{y}) &\propto \exp(-\mathbf{E}(\mathbf{y}))_{|Y_L=L} \\
&= \exp\left(-\sum_{i,j=1}^{N} \mathbf{W}_{ij}(y_i - y_j)^2\right)_{|Y_L=L} \\
&= \exp\left(-\mathbf{y}^\top \mathbf{L}\mathbf{y}\right)_{|Y_L=L}
\end{aligned} \tag{1}
$$

$\mathbf{L} = \mathbf{D} - \mathbf{W}$, The distribution of $\mathbf{y}_U$ given $\mathbf{y}_L$ is Gaussian, $\mathbf{y}_U \sim \mathcal{N}(f_U, \frac{1}{2}(\mathbf{L}_{UU})^{-1})$, where $f_U = (\mathbf{L}_{UU})^{-1}\mathbf{L}_{UL}\mathbf{y}_L$.
It simply leading us to the way, where we can infer about the unlabelled data by aggregating information from the graph.

$$
f_i = \frac{\sum_{j\sim i} \mathbf{W}_{ij} f_j}{\sum_{j\sim i} \mathbf{W}_{j\sim i}} \quad i \in U \tag{2}
$$

Graph Regularization

A simple example, graph regularized problem could be posed as

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}, \text{data}) + \lambda(\mathbf{x}^\top h(\mathbf{W})\mathbf{x}) \tag{3}$$

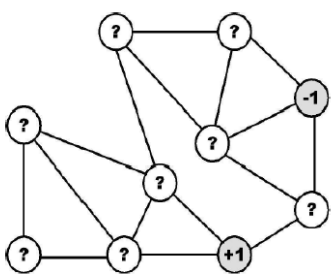
Most popularly used: $h(\mathbf{W}) = \mathbf{L}$.

Problem: Given noisy graph signal $f = y_0 + \eta$ recover y_0.
$y^\star = (I + \gamma L)^{-1} f$

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

Data fitting term

"Smoothness" assumptio



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

$$\underset{\mathbf{x}}{\text{minimize}} \quad (y - \mathbf{x})^2 + \lambda(\mathbf{x}^\top \mathbf{L}\mathbf{x}) \tag{4}$$

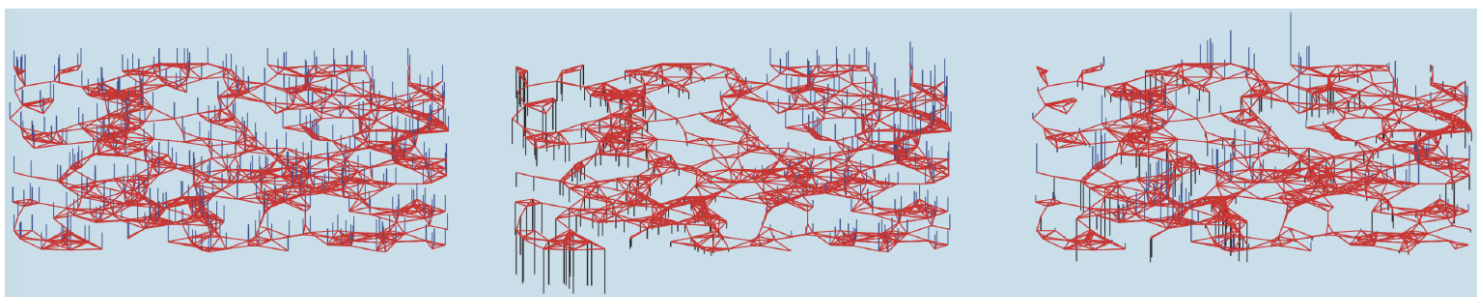
Eigenvectors and Graph Fourier Transform

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

$$L = \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix} \begin{bmatrix} — & \chi_0 & — \\ & \cdots & \\ — & \chi_{N-1} & — \end{bmatrix}$$

$$\chi \qquad\qquad \Lambda \qquad\qquad \chi^T$$

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



χ_0 χ_1 χ_{50}

Low frequency → High frequency

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

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

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

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

Graph Signal Processing Workflow

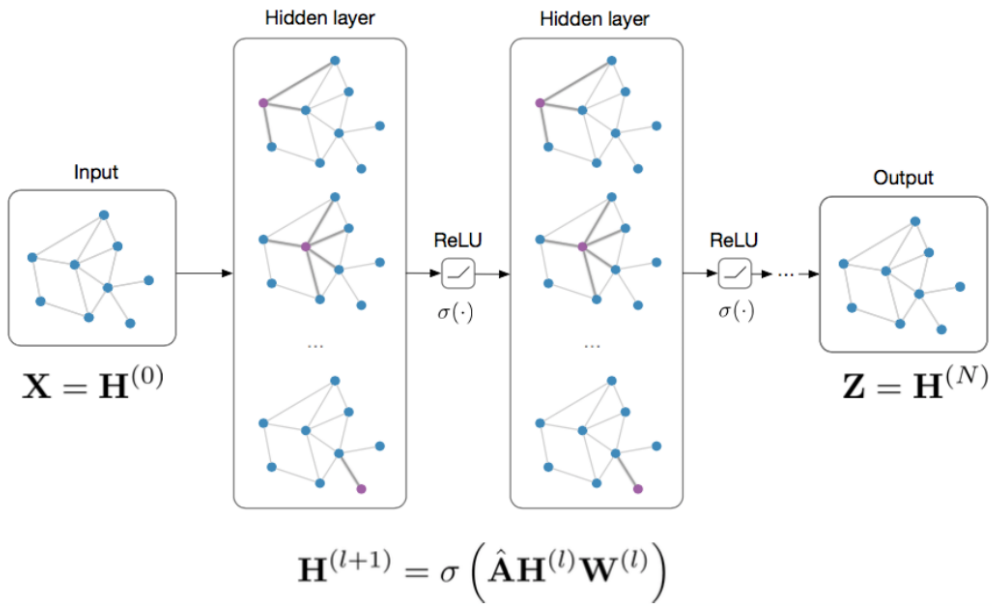
- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $-\nabla^2$	graph Laplacian: L
eigenfunctions: $e^{j\omega x}$	eigenvectors: χ_ℓ
	$f : V \to \mathbb{R}^N$
Classical FT: $\hat{f}(\omega) \models \int (e^{j\omega x})^* f(x) dx$	Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i)$
$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$	$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

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

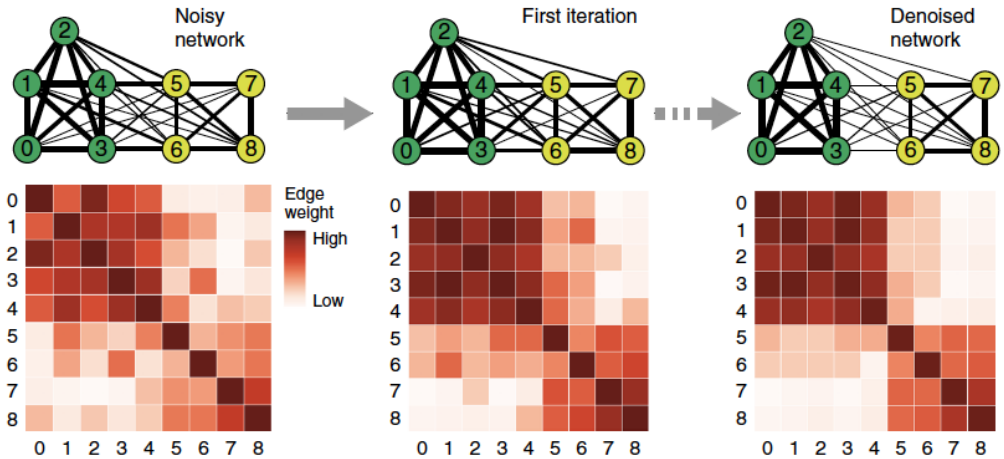
Graph Convolution Neural Networks

- Forward pass: $\hat{g}_{\theta^{(k+1)}}(L)\Big(\mathrm{ReLU}(\hat{g}_{\theta^{(k)}}(L)f)\Big)$

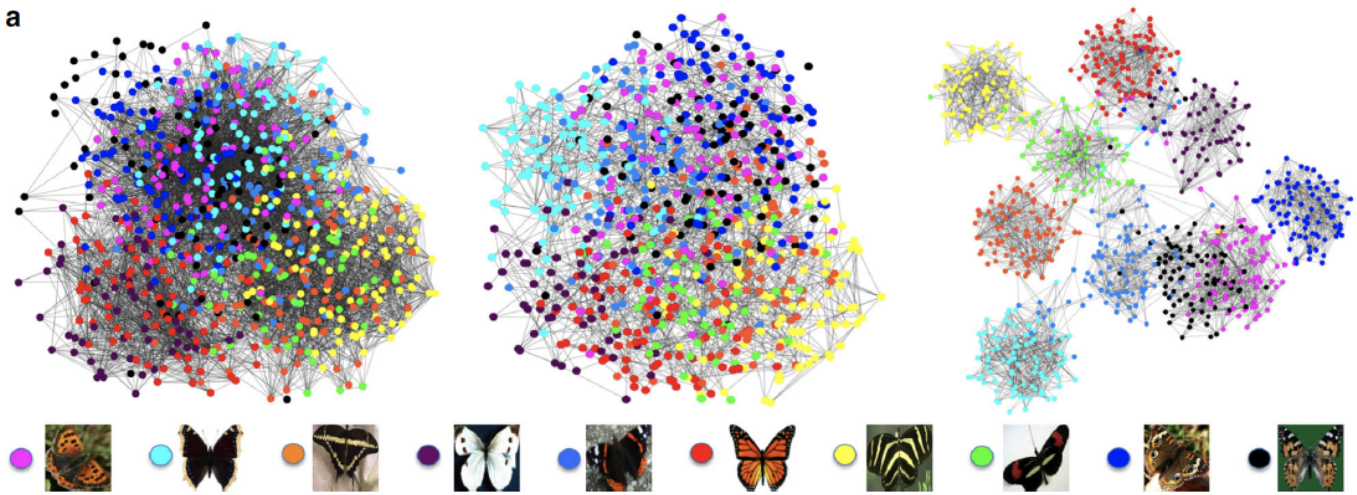


Kipf and Welling, "Semi-supervised classification with graph convolutional networks," ICLR, 2017.

Network Enhancement via Denoising



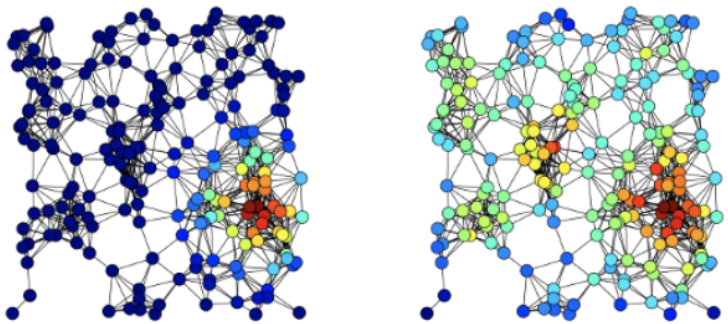
Network Denoising



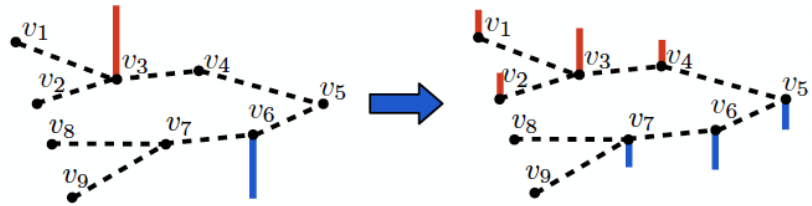
Classification with fine grained data

Applications

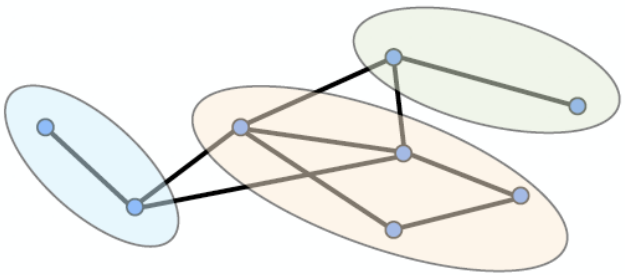
centrality, diffused information, class membership, node labels (and node-level features in general), structured inference **ALL** be understood or solved with the help of graphs.



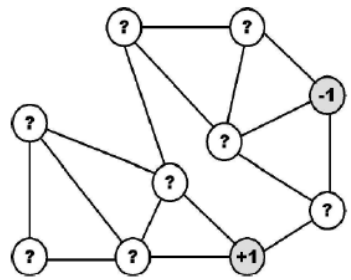
network science



network diffusion

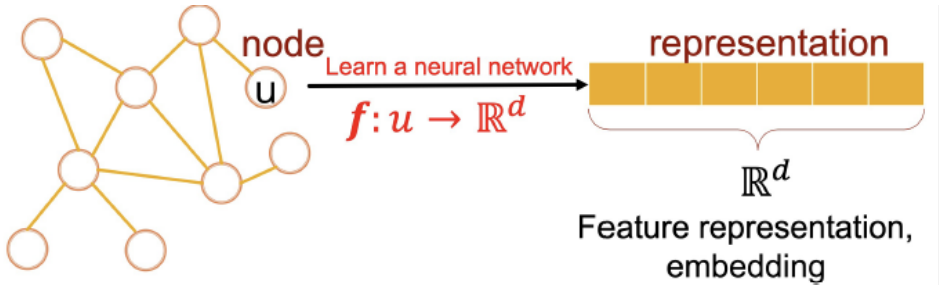


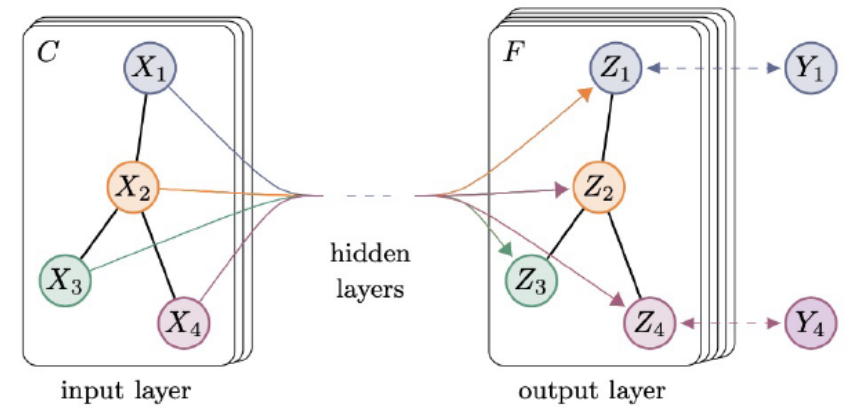
unsupervised learning (dimensionality reduction, clustering)



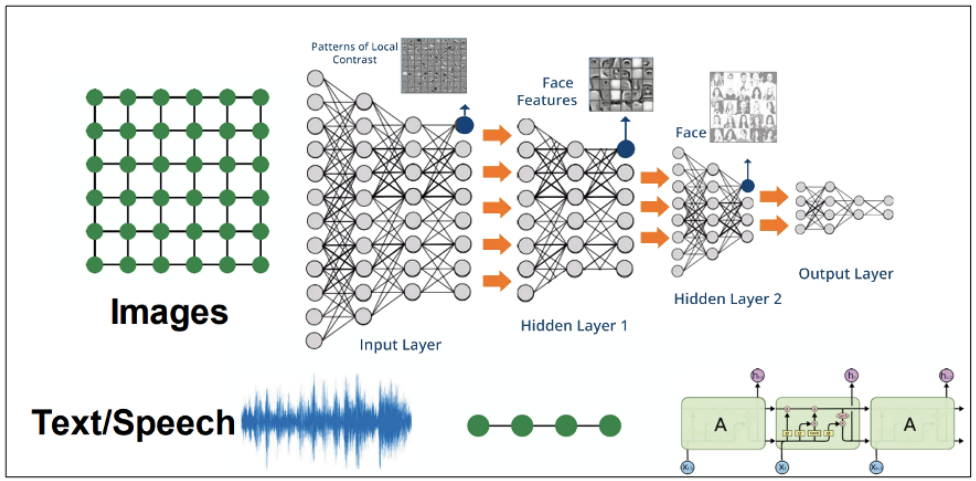
semi-supervised learning

Machinne Learning with Graphs



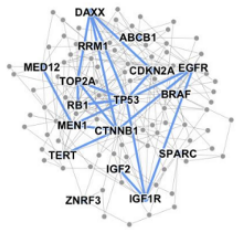


Deep Learning with Graphs



Traditional Deep Learning

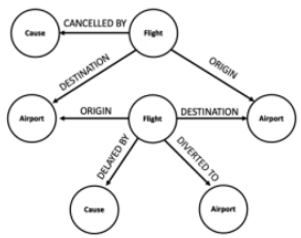
Why Graphs



Disease Pathways

Image credit: Medium

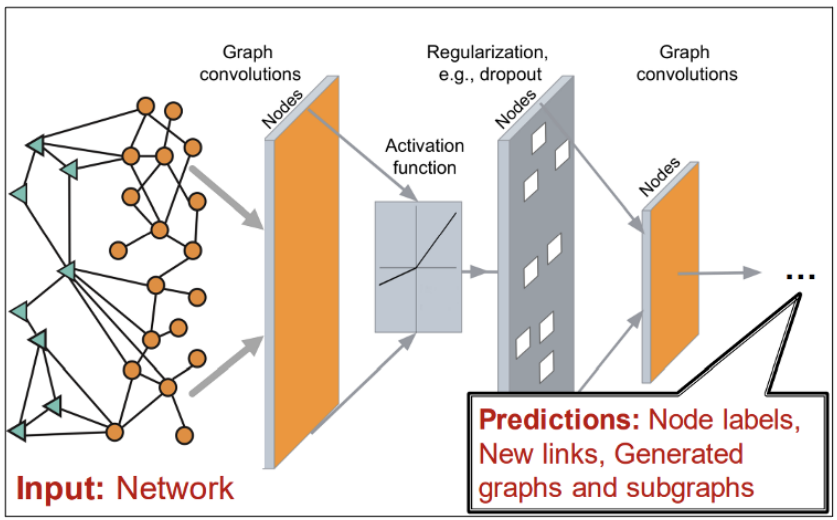
Social Networks

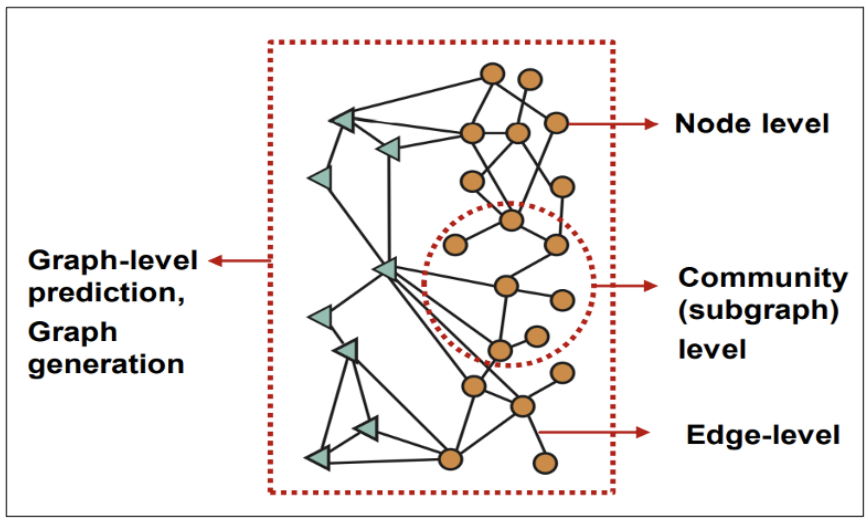


Event Graphs

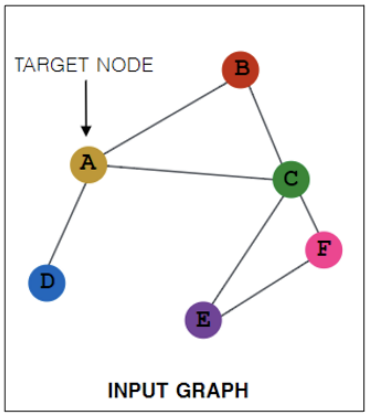
Citation Networks

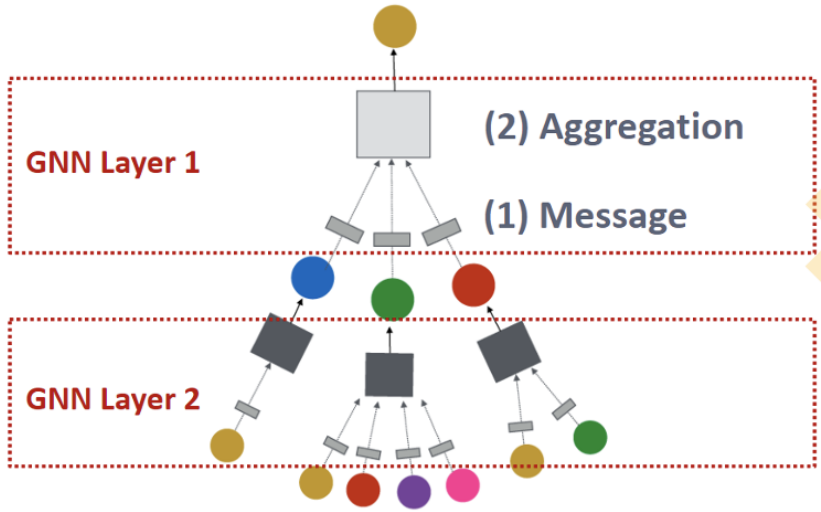
Graph Neural Networks



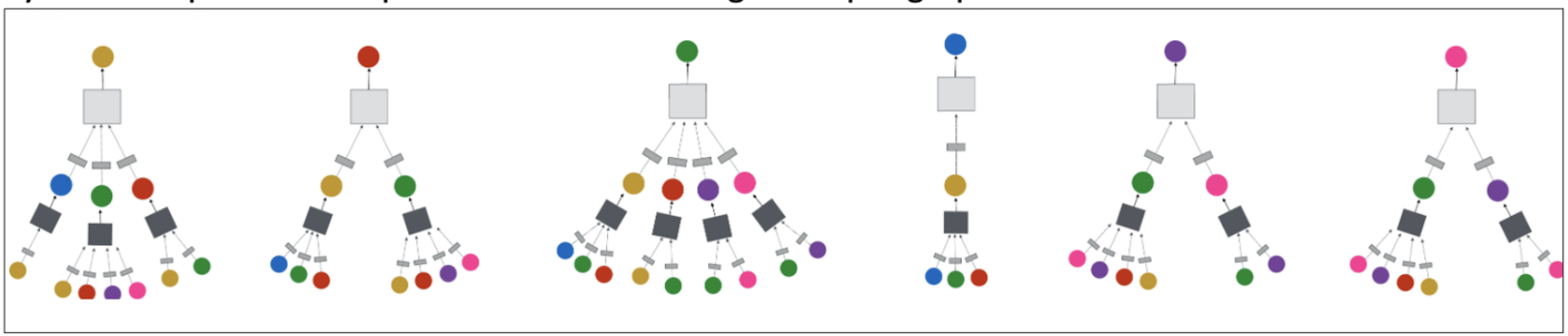


GNN Layers





2 Layered Computation Graph for all the nodes of given input graph:



GNN: Message Passing and Aggregation

- Message computation for the neighboring nodes:

$$\mathbf{m}_u^{(l)} = \mathrm{MSG}^{(l)}\left(\mathbf{h}_u^{(l-1)}\right), u \in \{N(v) \cup v\}$$

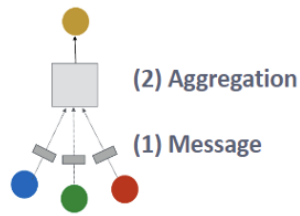
- Aggregation: $\mathbf{h}_v^{(l)} = \mathrm{AGG}^{(l)}\left(\left\{\mathbf{m}_u^{(l)}, u \in N(v)\right\}, \mathbf{m}_v^{(l)}\right)$

- Nonlinearity (activation) can be added to message or aggregation step to increase expressiveness!

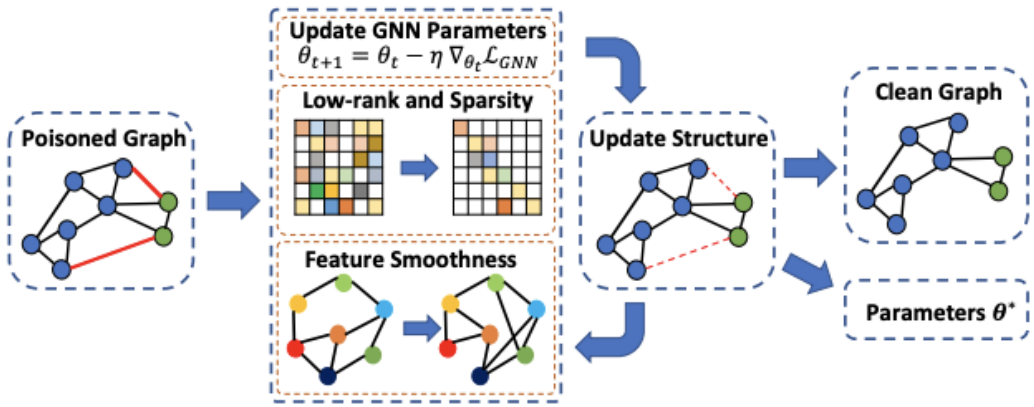
Message + Aggregation Operation in GCN: $\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \mathbf{W}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}\right)$

Message

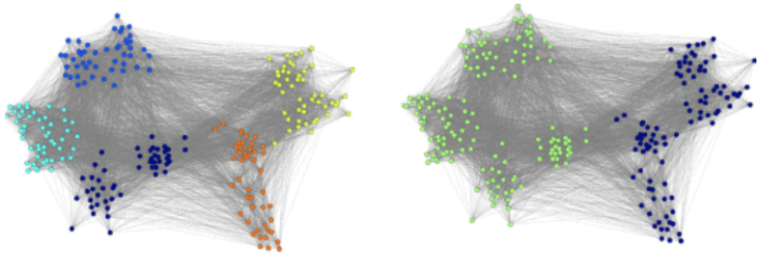
Aggregation



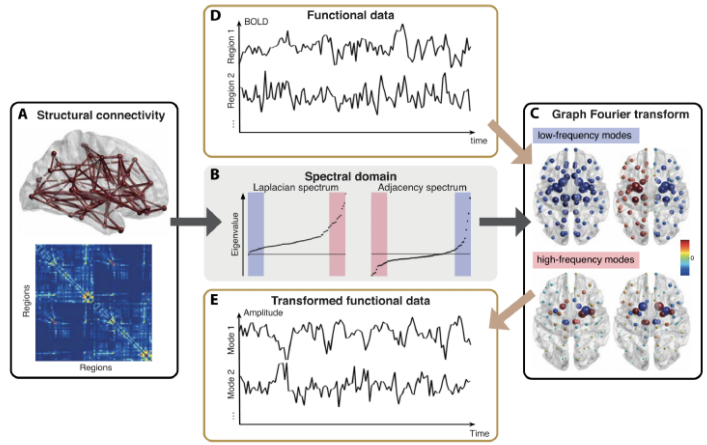
Importance of Topology



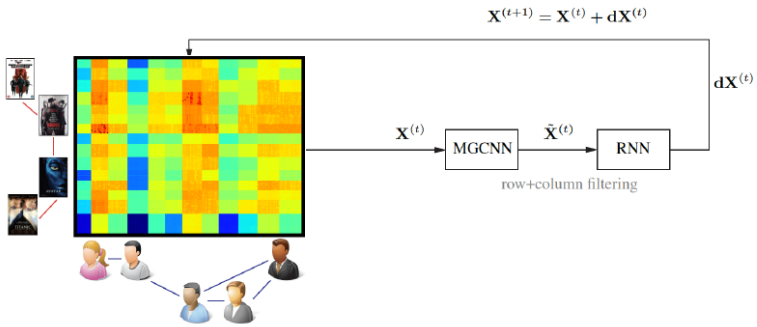
Applications



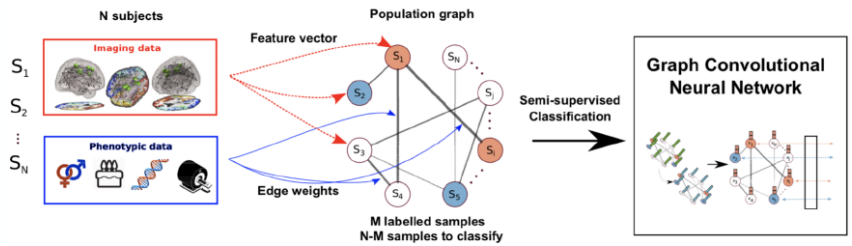
community detection



functional brain imaging



recommender system



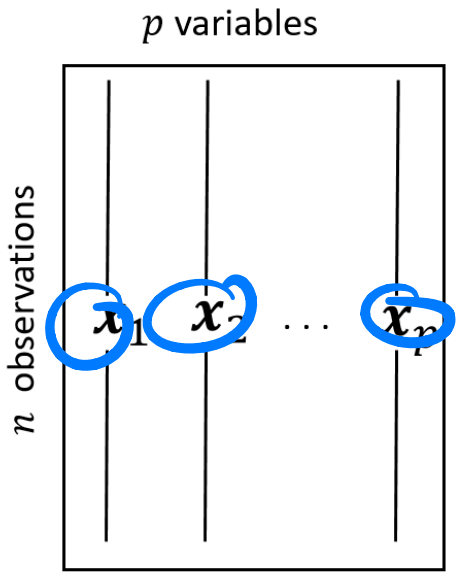
disease classification

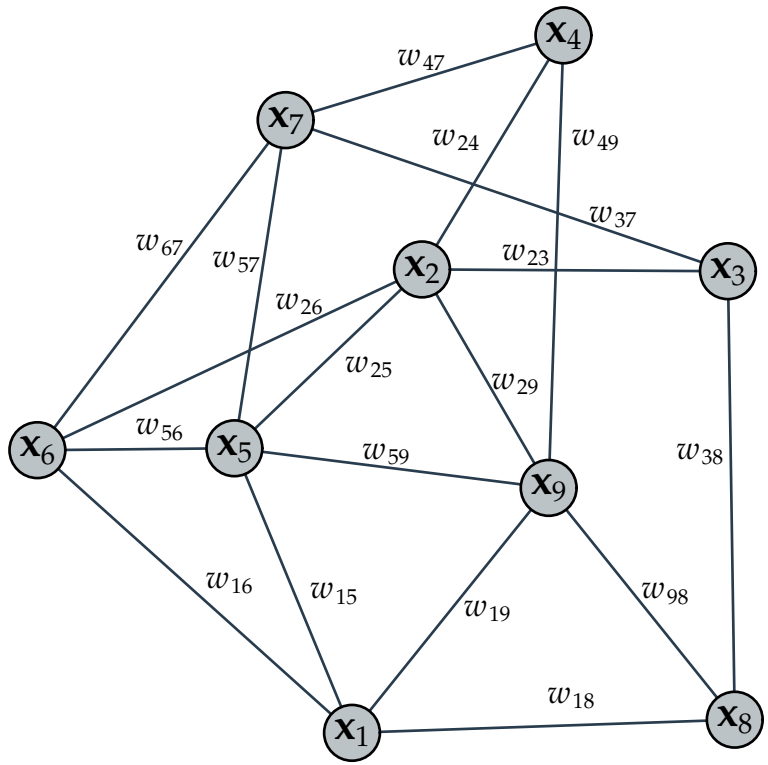
Tremblay and Borgnat 2014, Monti et al. 2017, Huang et al. 2018, Parisot et al. 2018

Outline

Scheme of Graph Learning

- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ be an undirecetd graph with p nodes, and each node is associated with a r.v x_i, such $\mathbf{x} = (x_1, x_2, \ldots, x_p) \in \mathcal{X}^p$.
- The goal is to obtain a graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ from the data $\mathbf{x}$.





Graph Learning from Data.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1**-regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \left\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i \neq j} \mathbf{\Theta}_{ij}\right\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j\right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1-**regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \Big\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i\neq j} \mathbf{\Theta}_{ij}\Big\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1-**regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \Big\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i\neq j} \mathbf{\Theta}_{ij}\Big\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1-**regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \left\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i \neq j} \mathbf{\Theta}_{ij}\right\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1-**regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \left\{\Theta \in \mathbb{S}_{++}^p | \Theta_{ij} \leq 0, \Theta = \Theta^\top\right\}$.
- $\mathcal{S}_\Theta = \left\{\Theta \in \mathbb{S}_{+}^p | \Theta_{ij} \leq 0, \Theta_{ii} = -\sum_{i \neq j} \Theta_{ij}\right\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1-**regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \Big\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i \neq j} \mathbf{\Theta}_{ij}\Big\}$ positivity along with Laplacian constraint.

Undirected Graph Learning: GMRF

Given $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^{-1})\}_{i=1}^{n}$, estimate the graph matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$.

- **Covariance selection** [Dempster, 1972]: Non zero elements of $\mathbf{S}^{-1}$ lends you a graph, where $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$. Not suitable for high-dimensional data.

- **Neighborhood regression** [Meinshausen and Bühlmann, 2006]:

$$\underset{\{\theta_{ij}\}:\theta_{ii}=0}{\text{minimize}} \left(x_i - \sum_{j=1, j\neq i}^{p} \theta_{ij} x_j \right)^2 + \alpha \sum_{j \neq i} |\theta_{ij}| \;\forall\; i = 1, 2, \ldots, p$$

- ℓ_1**-regularized MLE** [Banerjee et al., 2008]:

$$\underset{\mathbf{\Theta} \succ \mathbf{0}}{\text{maximize}} \;\; \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Computational efficient GLasso algorithm [Friedman et al., 2008].
- Ising model: $\log\det$ relaxation [Banerjee et al., 2008] and logistic regression [Ravikumar et al., 2010].
- $\mathcal{S}_\Theta = \big\{\mathbf{\Theta} \in \mathbb{S}_{++}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta} = \mathbf{\Theta}^\top\big\}$.
- $\mathcal{S}_\Theta = \Big\{\mathbf{\Theta} \in \mathbb{S}_{+}^p | \mathbf{\Theta}_{ij} \leq 0, \mathbf{\Theta}_{ii} = -\sum_{i\neq j} \mathbf{\Theta}_{ij}\Big\}$ **positivity along with Laplacian constraint**.

Graph learning from smooth signals

Quantifying smoothness:

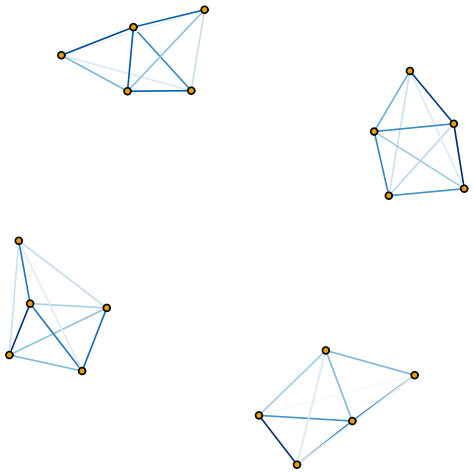
$$\text{tr}\left(\mathbf{X}\mathbf{L}\mathbf{X}^\top\right) = \sum_{i,j} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$$

A smaller $\text{tr}\left(\mathbf{X}^\top\mathbf{L}\mathbf{X}\right)$ indicating a smoother signal $\mathbf{X}$ over the graph $\mathcal{G}$, where $\mathbf{L}$ is its Laplacian matrix.
When a graph $\mathcal{G}$ is not already available, we can learn it directly from the data $\mathbf{X}$ by finding the graph weights that minimize the smoothness term combined with some regularization term:

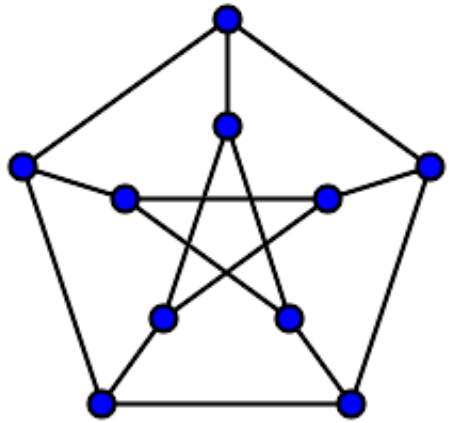
$$\hat{\mathbf{L}} := \arg\min_{\mathbf{L}} \text{tr}\left(\mathbf{X}\mathbf{L}\mathbf{X}^\top\right) + \lambda h(\mathbf{L}),$$

- Smaller distance $\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$ between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ will force to learn a graph with larger affinity value w_{ij}, and vice versa.
- Higher value of weight w_{ij} will imply the features $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, and hence, strongly connected.
- $h(\mathbf{L})$ is a regularization function (e.g., $\left\|\mathbf{L}\right\|_1$, $\left\|\mathbf{L}\right\|_F^2$, $\log\det(\mathbf{L})$)

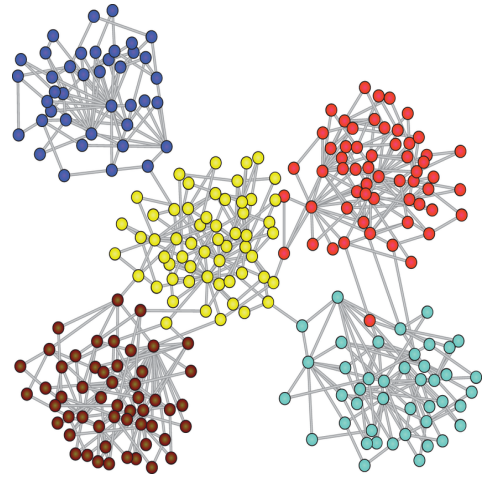
Structured Graphs



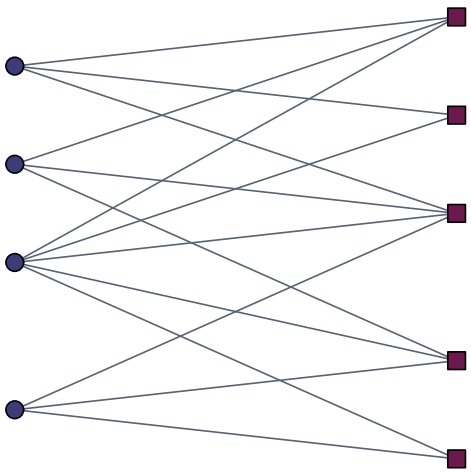
(iii) Multi-component graph



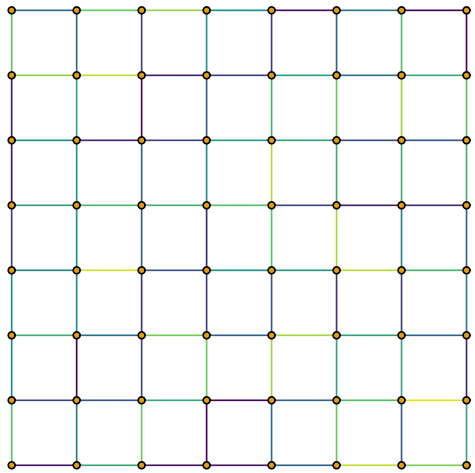
(iv) Regular graph



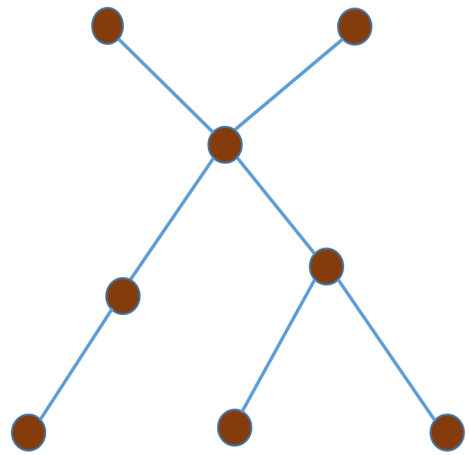
(v) Modular graph



(vi) Bipartite graph



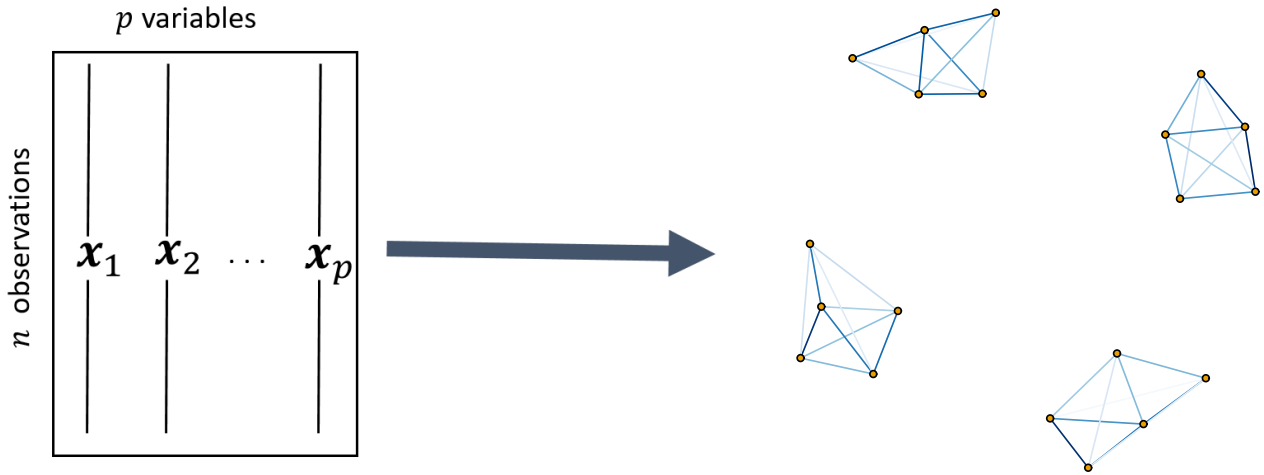
(vii) Grid graph



(viii) Tree graph

Figure 2: Useful graph structures

Structured Graph Learning is NP-Hard



Structured graph learning from data

- involves both the estimation of structure (**graph connectivity**) and parameters (**graph weights**),
- parameter estimation is well explored (e.g., maximum likelihood).
- But **structure** is a **combinatiorial** constraint and NP-hard to impose,
- thus, in general learning a **structured** graph becomes **NP-hard**, and challenging [Bogdanov et al., 2008].

State-Of-The-Art and Proposed Direction

State-of-the-art:

- **Maximum weight spanning tree** for tree structure [Chow and Liu, 1968].
- **Local-separation** and **walk summability** for Erdos-Renyi graphs, power-law graphs, and small-world graphs [Anandkumar et al., 2012].
- **Scale free** [Liu and Ihler, 2011], **Degree distribution** [Huang and Jebara, 2008], **Overlapping structured sparsity** [Tarzanagh and Michailidis, 2017], and **Multiple graphs** [Hao et al., 2017].
- Existing methods are restricted to some particular structures and it is difficult to extend them to learn other useful structures, e.g., multi-component, bipartite, etc.

Proposed Direction Graph (**structure**) $\Longleftrightarrow$ Graph matrix (**spectrum**)

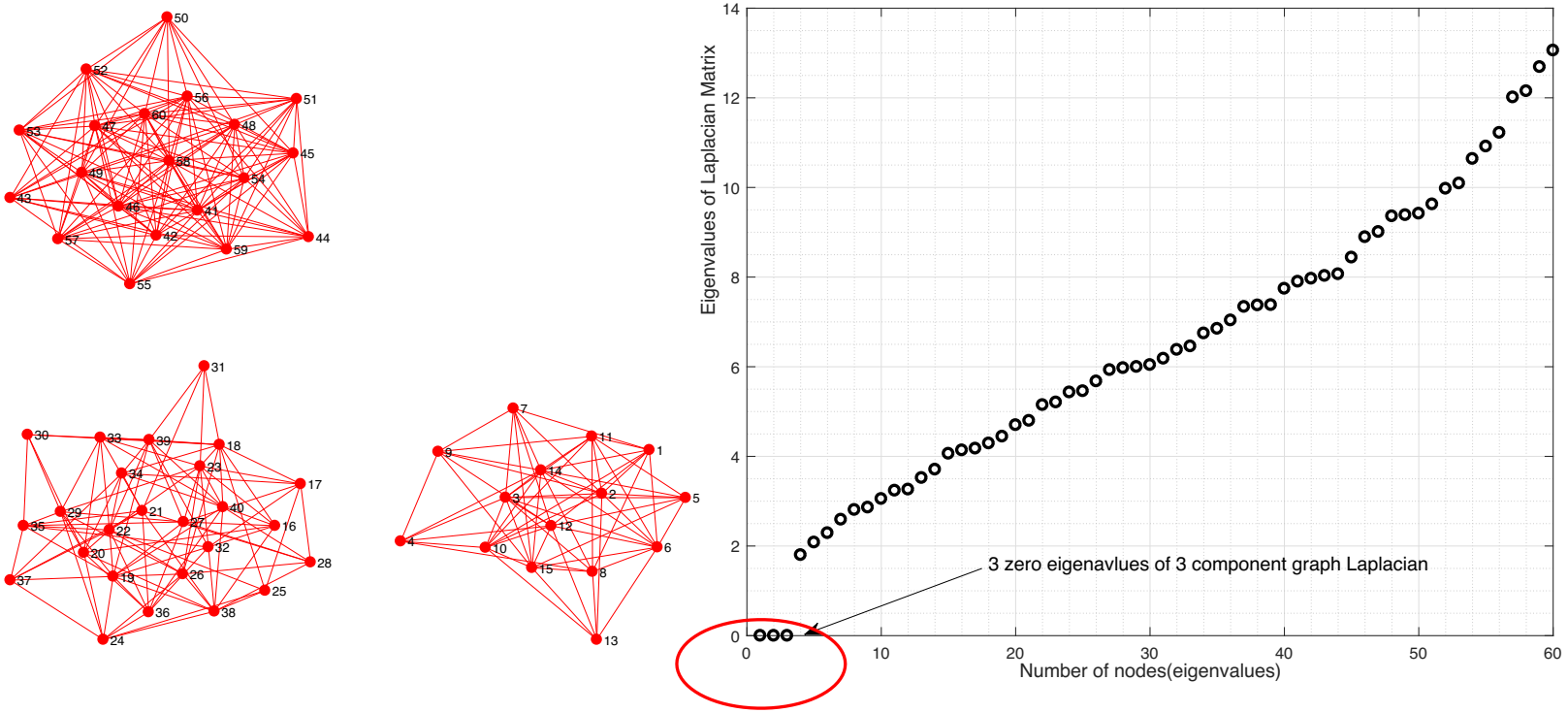
- **Structural properties** of many important graph families are **exactly encoded** in the **spectral properties** of their matrices [Chung, 1997, Van Mieghem, 2010].
- Utilizing spectral properties will be the the enabling factor for learning structured graphs.

Motivation 1: Structure via Laplacian Eigenvalues

$$\mathbf{\Theta} = \mathbf{U}\text{Diag}(\boldsymbol{\lambda})\mathbf{U}^T$$

For a multi-component graph the first k eigenvalues of its Laplacian matrix are zero:

$$\mathcal{S}_\lambda = \{\{\lambda_j = 0\}_{j=1}^{k},\ c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$$



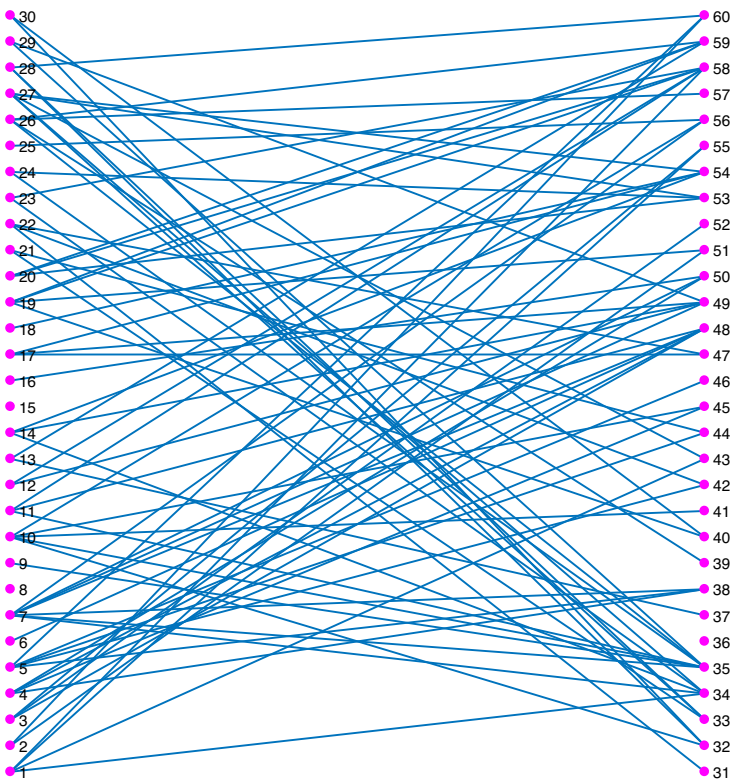
Motivation 2: Structure via Adjacency Eigenvalues

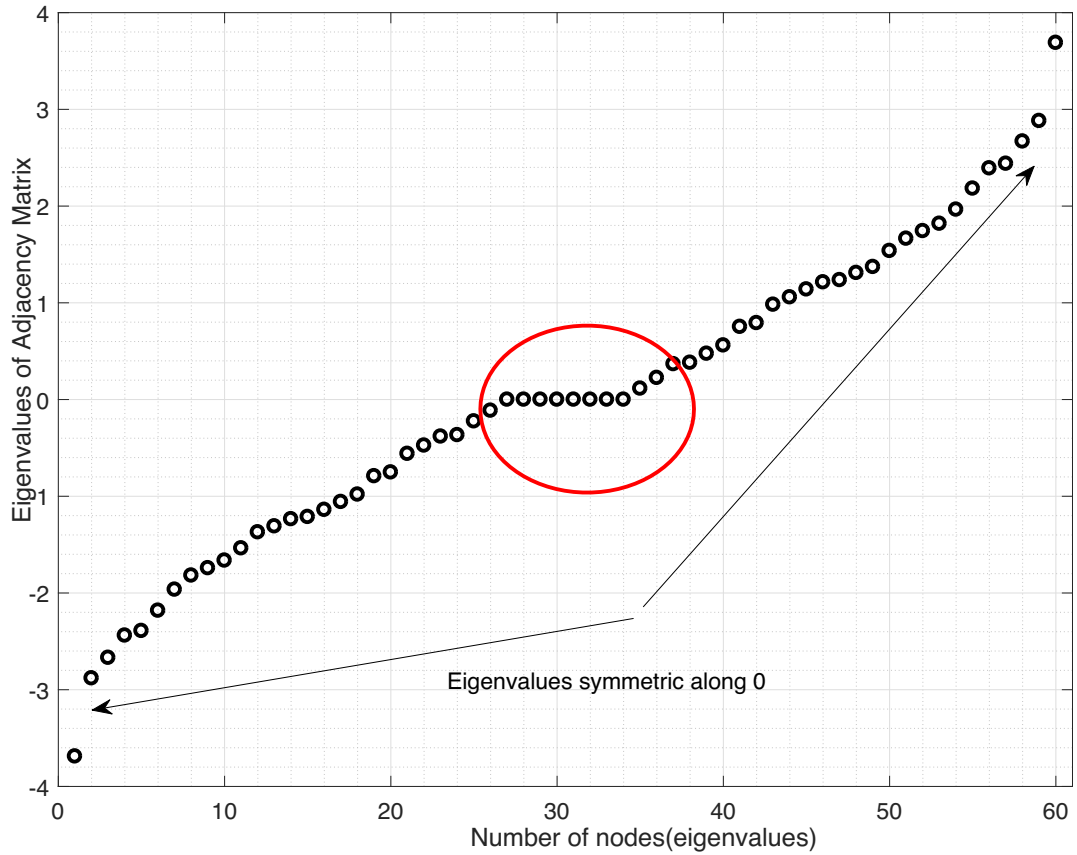
Adjacency matrix $\mathcal{A}(\mathbf{\Theta})$: $\mathcal{A}(\mathbf{\Theta}) = \mathsf{Diag}(\mathsf{diag}(\mathbf{\Theta})) - \mathbf{\Theta}$.

$$\mathcal{A}(\mathbf{\Theta}) = \mathbf{V}\mathsf{Diag}(\boldsymbol{\psi})\mathbf{V}^T$$

For a bipartite graph the eigenvalues are symmetric about the origin:

$$\mathcal{S}_\psi = \{\psi_i = -\psi_{p-i+1},\ \forall i\ = 1, \ldots, p\}.$$





Setting our Goal

- **Given**: Observations $\mathbf{X} = \{\mathbf{x}^{(i)} \in \mathbb{R}^p\}_{i=1}^n$
- Compute sample covariance matrix: $\mathbf{S} = \frac{1}{n}\sum_{i=1}^n (\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^\top$.
- **Goal**: Estimate $\mathbf{\Theta}$, where $\mathbf{\Theta}$ satisfies the Laplacian constraints

$$\mathcal{S}_\Theta = \left\{ \mathbf{\Theta} \in \mathbb{S}_+^p | \mathbf{\Theta}_{ij} \leq 0, \text{for } i \neq j, \mathbf{\Theta}_{ii} = -\sum_{i \neq j} \mathbf{\Theta}_{ij} \;\; (\mathbf{\Theta} \cdot \mathbf{1} = \mathbf{0}) \right\}$$

,

- and eigenvalues of $\boldsymbol{\lambda}(\mathbf{\Theta})$ satisfy some application depended spectral constraints $\mathcal{S}_\lambda$.

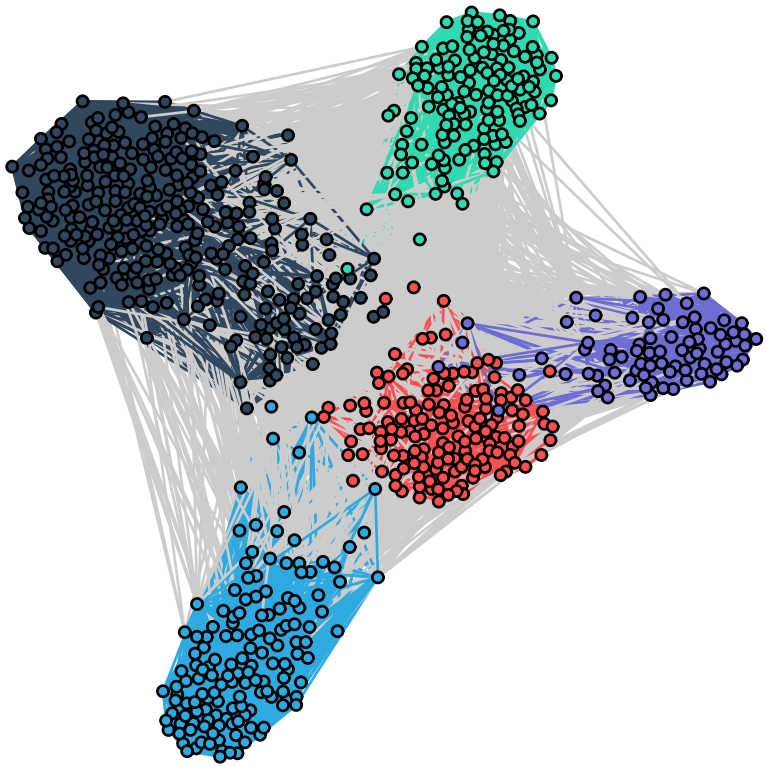
Proposed Unified Framework for Structured Graph Learning (SGL)

$$\begin{aligned} \underset{\boldsymbol{\Theta}}{\text{maximize}} \quad & \log \text{ gdet}(\boldsymbol{\Theta}) - \text{tr}\big(\boldsymbol{\Theta}\mathbf{S}\big) - \alpha h(\boldsymbol{\Theta}), \\ \text{subject to} \quad & \boldsymbol{\Theta} \in \mathcal{S}_{\boldsymbol{\Theta}}, \ \boldsymbol{\lambda}(\mathcal{T}(\boldsymbol{\Theta})) \in \mathcal{S}_{\mathcal{T}} \end{aligned}$$

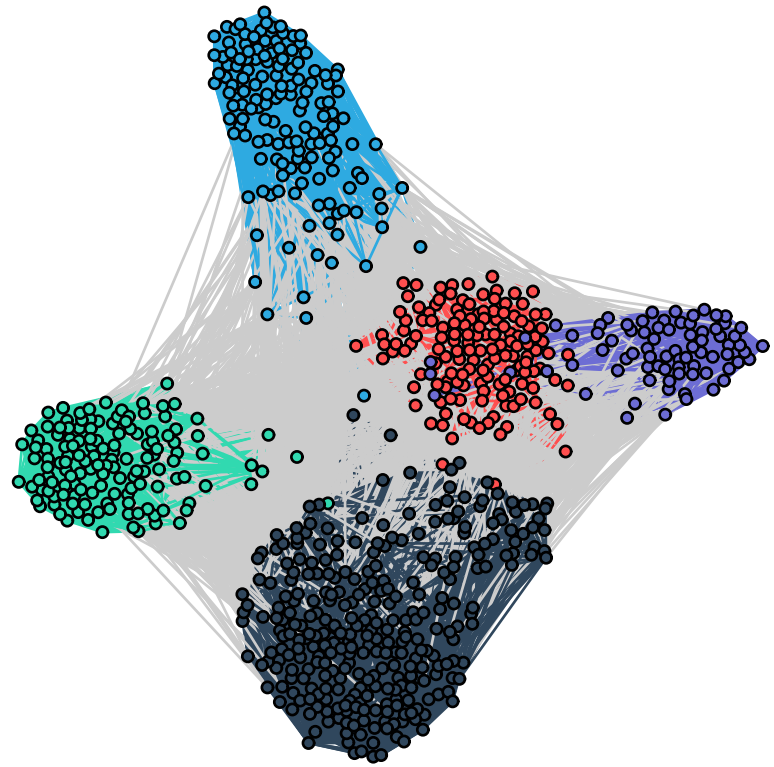
- $\text{gdet}(\cdot)$ is the generalized determinant defined as the non-zero eigenvalues product,
- $\mathcal{S}_{\boldsymbol{\Theta}}$ encodes the typical constraints of a **Laplacian matrix**,
- $\boldsymbol{\lambda}(\mathcal{T}(\boldsymbol{\Theta}))$ is the vector containing the **eigenvalues** of matrix $\mathcal{T}(\boldsymbol{\Theta})$,
- $\mathcal{T}(\cdot)$ is the **transformation matrix** to consider the eigenvalues of **different graph matrices**, and
- $\mathcal{S}_{\mathcal{T}}$ allows to include **spectral constraints** in the eigenvalues.
- Precisely $\mathcal{S}_{\mathcal{T}}$ will facilitate the process of incorporating the spectral properties required for enforcing structure.
- If $\mathbf{x}^{(i)} \sim \mathcal{N}(0, \boldsymbol{\Sigma} = \boldsymbol{\Theta}^{\dagger})$ then its an **approximate likelihood estimation**, IGMRF.
- For **arbitrary distributed data**, it is a signed constrained log-determinant divergence minimization of a positive semi definite $M-$matrix.

The proposed formulation has converted the **combinatorial** structural constraints into **analytical** spectral constraints.

Real Data: Cancer Dataset [Weinstein et al., 2013]

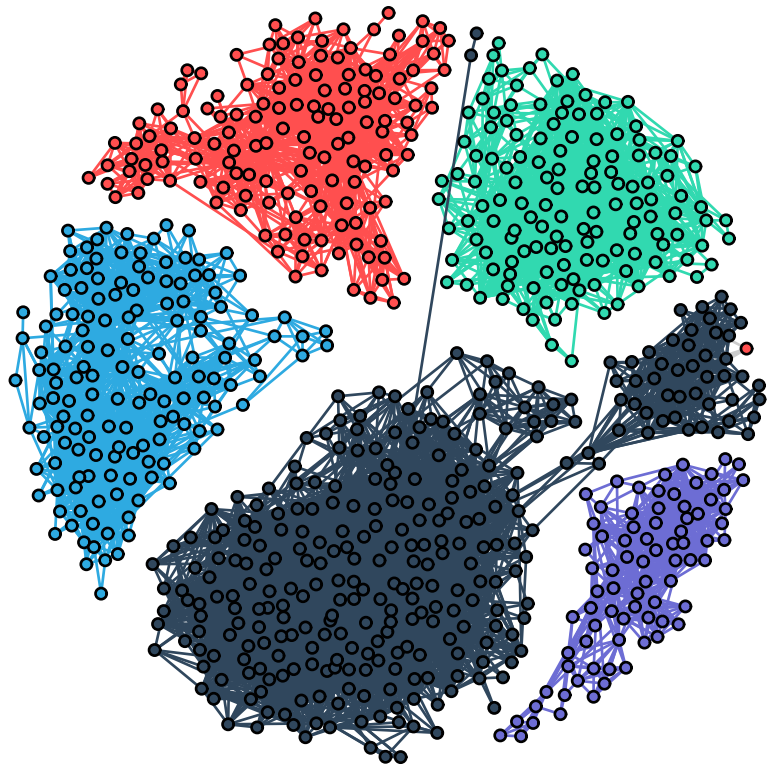


(iii) GLasso [Friedman et al., 2008]



(iv) GGL [Egilmez et al., 2017]

Real Data: Cancer Dataset [Weinstein et al., 2013]



(v) SGL with $k = 5$

Learning an enhanced graph

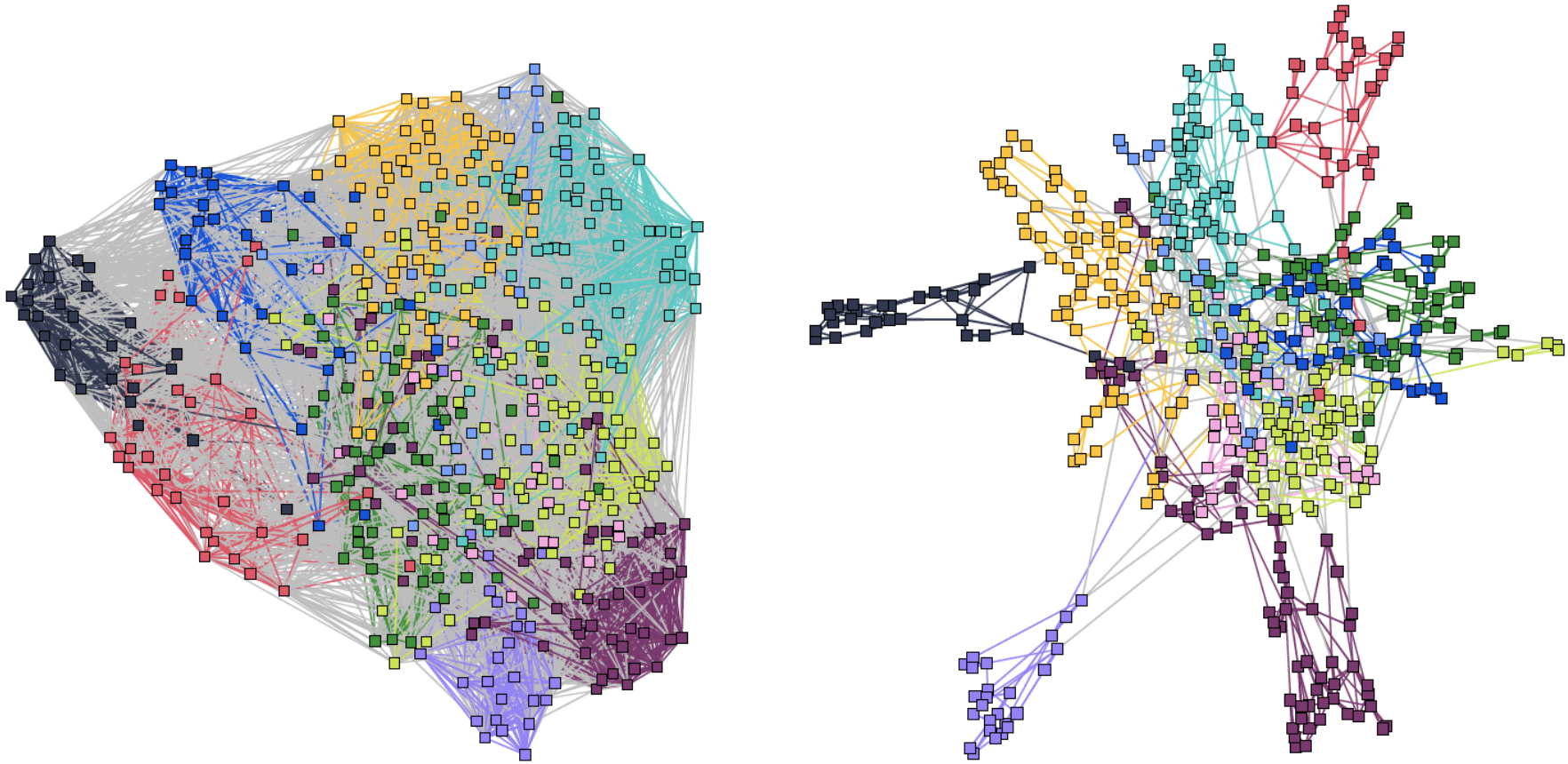


Figure 3: Learned graph networks of stocks from all sectors (each color represents a sector) of the S&P500 using the (a) state-of-the-art Gaussian model (b) and heavy-tailed model

- Kumar, Sandeep, et al., & Palomar, D. (2019). Structured graph learning via Laplacian spectral constraints. In Advances in Neural Information Processing Systems (pp. 11651-11663).
- Kumar, Sandeep, et al. "A Unified Framework for Structured Graph Learning via Spectral Constraints." Journal of Machine Learning Research 21.22 (2020): 1-60.

Acknowledgement

Some parts of the slides are taken from the slides of

- Prof. Zoubin Ghahramani, University of Cambridge, http://mlg.eng.cam.ac.uk/zoubin/
- Prof. Xiaowen Dong, Oxford University, https://web.media.mit.edu/~xdong/
- Prof. Jure Leskovec, Stanford University, https://cs.stanford.edu/people/jure/

Contact us:

- **Email**: ksandeep@iitd.ac.in
- **Wesbite**: https://sites.google.com/view/sandeepkr/home

An R based software package for structured graph learning.
https://cran.r-project.org/web/packages/spectralGraphTopology/index.html

Thank You.

References

Anandkumar, A., Tan, V. Y., Huang, F., and Willsky, A. S. (2012).

High-dimensional gaussian graphical model selection: Walk summability and local separation criterion.

Journal of Machine Learning Research, 13(Aug):2293–2337.

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008).

Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data.

Journal of Machine Learning Research, 9(Mar):485–516.

Barabási, A.-L. et al. (2016).

Network science.

Cambridge university press.

Bogdanov, A., Mossel, E., and Vadhan, S. (2008).

The complexity of distinguishing markov random fields.

In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 331–342. Springer.

Chow, C. and Liu, C. (1968).

Approximating discrete probability distributions with dependence trees.

IEEE transactions on Information Theory, 14(3):462–467.

References

Chung, F. R. (1997).

Spectral graph theory.

Number 92. American Mathematical Soc.

Dempster, A. P. (1972).

Covariance selection.

Biometrics, pages 157–175.

Egilmez, H. E., Pavez, E., and Ortega, A. (2017).

Graph learning from data under laplacian and structural constraints.

IEEE Journal of Selected Topics in Signal Processing, 11(6):825–841.

Friedman, J., Hastie, T., and Tibshirani, R. (2008).

Sparse inverse covariance estimation with the graphical lasso.

Biostatistics, 9(3):432–441.

Hao, B., Sun, W. W., Liu, Y., and Cheng, G. (2017).

Simultaneous clustering and estimation of heterogeneous graphical models.

The Journal of Machine Learning Research, 18(1):7981–8038.

Huang, B. and Jebara, T. (2008).

Maximum likelihood graph structure estimation with degree distributions.

In *Analyzing Graphs: Theory and Applications, NIPS Workshop*, volume 14.

References

Liu, Q. and Ihler, A. (2011).

Learning scale free networks by reweighted l1 regularization.

In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 40–48.

Meinshausen, N. and Bühlmann, P. (2006).

High-dimensional graphs and variable selection with the lasso.

The annals of statistics, 34(3):1436–1462.

Ravikumar, P., Wainwright, M. J., Lafferty, J. D., et al. (2010).

High-dimensional ising model selection using ℓ_1-regularized logistic regression.

The Annals of Statistics, 38(3):1287–1319.

Tarzanagh, D. A. and Michailidis, G. (2017).

Estimation of graphical models through structured norm minimization.

The Journal of Machine Learning Research, 18(1):7692–7739.

Van Mieghem, P. (2010).

Graph spectra for complex networks.

Cambridge University Press.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., Network, C. G. A. R., et al. (2013).

The cancer genome atlas pan-cancer analysis project.

Nature Genetics, 45(10):1113.