Network Topology Inference

Gonzalo Mateos
Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://www.hajim.rochester.edu/ece/sites/gmateos/

March 28, 2023
Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Network topology inference

- So far dealt with modeling and inference of observed network graphs
  - **Q**: If a portion of $G$ is unobserved, can we infer it from data?
- Discussed construction of representations $G(V, E)$ for network mapping
  - Largely informal methodology, lacking an element of validation
- **Formulate instead as statistical inference task**, i.e. given
  - Measurements $x_i$ of attributes at some or all vertices $i \in V$
  - Indicators $y_{ij}$ of edge status for some vertex pairs $\{i, j\} \in V^{(2)}$
  - A collection $\mathcal{G}$ of candidate graphs $G$

**Goal**: infer the topology of the network graph $G(V, E)$

- Three canonical **network topology inference** problems
  1. Link prediction
  2. Association network inference
  3. Tomographic network topology inference
Suppose we observe vertex attributes $\mathbf{x} = [x_1, \ldots, x_{N_v}]^T$; and

Edge status is only observed for some subset of pairs $V_{obs}^{(2)} \subset V^{(2)}$

**Goal:** predict edge status for all other pairs, i.e., $V_{miss}^{(2)} = V^{(2)} \setminus V_{obs}^{(2)}$
Association network inference

- Suppose we only observe vertex attributes $\mathbf{x} = [x_1, \ldots, x_N]^\top$; and
- Assume $(i, j)$ defined by nontrivial ‘level of association’ among $x_i, x_j$
- **Goal:** predict edge status for all vertex pairs $V^{(2)}$
Suppose we only observe $x_i$ for vertices $i \subset V$ in the ‘perimeter’ of $G$.

**Goal:** predict edge and vertex status in the ‘interior’ of $G$. 
Link prediction

Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Link prediction

Let $G(V, E)$ be a random graph, with adjacency matrix $Y \in \{0, 1\}^{N_v \times N_v}$

$\Rightarrow$ $Y^{obs}$ and $Y^{miss}$ denote entries in $V^{(2)}_{obs}$ and $V^{(2)}_{miss}$

Link prediction

Predict entries in $Y^{miss}$, given observations $Y^{obs} = y^{obs}$ and possibly various vertex attributes $X = x \in \mathbb{R}^{N_v}$

Edge status information may be missing due to:

$\Rightarrow$ Difficulty in observation, issues of sampling

$\Rightarrow$ Edge is not yet present, wish to predict future status

Given a model for $X$ and $(Y^{obs}, Y^{miss})$, jointly predict $Y^{miss}$ based on

$$P(Y^{miss} | Y^{obs} = y^{obs}, X = x)$$

$\Rightarrow$ More manageable to predict the variables $Y^{miss}_{ij}$ individually
Informal scoring methods

- **Idea**: compute score \( s(i, j) \) for missing ‘potential edges’ \( \{i, j\} \in V_{miss}^{(2)} \)
  - Predicted edges returned by retaining the top \( n^* \) scores

- **Scores designed to assess certain local structural properties of \( G^{obs} \)**
  - Distance-based, inspired by the small-world principle
    \[
    s(i, j) = -\text{dist}_{G^{obs}}(i, j)
    \]
  - Neighborhood-based, e.g., the number of common neighbors
    \[
    s(i, j) = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}| \quad \text{or} \quad s(i, j) = \frac{|\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}|}{|\mathcal{N}_i^{obs} \cup \mathcal{N}_j^{obs}|}
    \]
  - Favor loosely-connected common neighbors [Adamic-Adar’03]
    \[
    s(i, j) = \sum_{k \in \mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}} \frac{1}{\log |\mathcal{N}_k^{obs}|}
    \]
Tests on co-authorship networks

- Results from a link prediction study in [Liben-Nowell-Kleinberg’03]
Classification methods

- **Idea:** use training data $\mathbf{y}^{obs}$ and $\mathbf{x}$ to build a binary classifier
  - Classifier is in turn used to predict the entries in $\mathbf{Y}^{miss}$
- **Logistic regression classifiers** most popular, based on the model

$$\log \left( \frac{P_\beta(\mathbf{Y}_{ij} = 1 \mid \mathbf{Z}_{ij} = \mathbf{z})}{P_\beta(\mathbf{Y}_{ij} = 0 \mid \mathbf{Z}_{ij} = \mathbf{z})} \right) = \mathbf{\beta}^\top \mathbf{z}, \quad \text{where}$$

(i) $\mathbf{\beta} \in \mathbb{R}^K$ is a vector of regression coefficients; and
(ii) $\mathbf{Z}_{ij}$ is a vector of explanatory variables indexed by $\{i, j\}$

$$\mathbf{Z}_{ij} = [g_1(\mathbf{Y}_{(ij)}^{obs}, \mathbf{X}), \ldots, g_K(\mathbf{Y}_{(ij)}^{obs}, \mathbf{X})]^\top$$

- Functions $g_k(\cdot)$ encode useful predictive information in $\mathbf{y}_{(ij)}^{obs}$ and $\mathbf{x}$
- **Ex:** vertex attributes, score functions, network statistics in ERGMs
Logistic regression classifier

- **Train:** Obtain MLE $\hat{\beta}$ via iteratively-reweighted LS
- **Test:** Potential edges $(i, j)$ declared present based on probabilities

\[
P_{\hat{\beta}}(Y_{ij} = 1 \mid Z_{ij} = z) = \frac{\exp(\hat{\beta}^\top z)}{1 + \exp(\hat{\beta}^\top z)}
\]

- Logistic regression assumes $Y_{ij}$ conditionally independent given $z$
  - Seldom the case with relational network data
- Underlying mechanism of data missingness is important
  - Classification for link prediction reminiscent of cross-validation
  - Assumption that data are missing at random is fundamental
Latent variable models

- In addition to a linear predictor $\beta^\top z$, latent models describe $Y_{ij}$ as a function of vertex-specific latent variables $u_i$ and $u_j$.

- Latent models are flexible to capture underlying social mechanisms. 
  Ex: homophily (transitivity) and stochastic equivalence (groups).
Latent class and distance models

- **Latent distance model:** node $i$ has unobserved position $\mathbf{U}_i \in \mathbb{R}^d$
  - Positions $\mathbf{U}_i$ in latent space assumed i.i.d. e.g., Gaussian distributed
  - Model cond. probability of edge $Y_{ij}$ as function of $\beta^\top \mathbf{z} - \|\mathbf{u}_i - \mathbf{u}_j\|_2$
  - **Homophily:** Nearby nodes in latent space more likely to link

- **Latent class model:** node $i$ belongs to unobserved class $\mathbf{U}_i \in \{1, \ldots, k\}$
  - Classes $\mathbf{U}_i$ assumed i.i.d. e.g., multinomial distributed
  - Model cond. probability of edge $Y_{ij}$ as function of $\beta^\top \mathbf{z} - \theta_{u_i, u_j}$
  - **Stochastic equivalence:** Nodes in same class equally likely to link

Let $M \in \mathbb{R}^{N_v \times N_v}$ be an unknown, random, and symmetric matrix

$$M = U^\top \Lambda U + E,$$

where

(i) $U = [u_1, \ldots, u_{N_v}]$ is a random orthonormal matrix of latent variables;
(ii) $\Lambda$ is a random diagonal matrix; and
(iii) $E$ is a symmetric matrix of i.i.d. noise entries $\epsilon_{ij}$

Latent eigenmodel subsumes the class and distance variants [Hoff'08]

⇒ Notice that $M_{ij} = u_i^\top \Lambda u_j + \epsilon_{ij}$

The logistic regression model with latent variables is

$$\log \left[ \frac{P_\beta(Y_{ij} = 1 \mid Z_{ij} = z, M_{ij} = m)}{P_\beta(Y_{ij} = 0 \mid Z_{ij} = z, M_{ij} = m)} \right] = \beta^\top z + m$$

$Y_{ij}$ still assumed conditionally independent given $Z_{ij}$ and $M_{ij}$

⇒ But they are conditionally dependent given only $Z_{ij}$
Bayesian link prediction

- Specify distributions for $\mathbf{U}, \mathbf{\Lambda}, \mathbf{E}$ to make statistical link predictions
  - Bayesian inference natural $\Rightarrow$ Specify a prior for $\beta$ as well
- To predict those entries in $\mathbf{Y}^{miss}$, threshold the posterior mean

$$
\mathbb{E} \left[ \frac{\exp \left( \beta^\top \mathbf{Z}_{ij} + M_{ij} \right)}{1 + \exp \left( \beta^\top \mathbf{Z}_{ij} + M_{ij} \right)} \middle| \mathbf{Y}^{obs} = \mathbf{y}^{obs}, \mathbf{Z}_{ij} = \mathbf{z} \right]
$$

- Use MCMC algorithms to approximate the posterior distribution
  - Gaussian distributions attractive for their conjugacy properties
- Higher complexity than MLE for standard logistic regression
  - Need to generate draws for $N_v^2$ unobserved variables $\{\mathbf{U}_{ij}\}$
  - Major cost reduction with reduced rank($\mathbf{U}$) $= k \ll N_v$ models
Case study

Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Lawyer collaboration network

- Network $G^{obs}$ of working relationships among lawyers [Lazega’01]
  - Nodes are $N_v = 36$ partners, edges indicate partners worked together

- Data includes various node-level attributes:
  - Seniority (node labels indicate rank ordering)
  - Office location (triangle, square or pentagon)
  - Type of practice, i.e., litigation (red) and corporate (cyan)
  - Gender (three partners are female labeled 27, 29 and 34)

- Goal: predict cooperation among social actors in an organization
Define the following set of explanatory variables:

\[ Z_{ij}^{(1)} = \text{seniority}_i + \text{seniority}_j, \quad Z_{ij}^{(2)} = \text{practice}_i + \text{practice}_j \]
\[ Z_{ij}^{(3)} = \mathbb{1}\{\text{practice}_i = \text{practice}_j\}, \quad Z_{ij}^{(4)} = \mathbb{1}\{\text{gender}_i = \text{gender}_j\} \]
\[ Z_{ij}^{(5)} = \mathbb{1}\{\text{office}_i = \text{office}_j\}, \quad Z_{ij}^{(6)} = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}| \]

**Method 1:** standard logistic regression with \( Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)} \)

**Method 2:** standard logistic regression with \( Z_{ij}^{(1)}, \ldots, Z_{ij}^{(6)} \)

**Method 3** informal scoring method with \( s(i, j) = Z_{ij}^{(6)} \)

**Method 4:** logistic regression with \( Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)} \) and latent eigenmodel

Five-fold cross-validation over the set of \( 36(36 - 1)/2 = 630 \) vertex pairs

\( \Rightarrow \) For each fold, \( 630/5 = 126 \) pairs in \( \mathcal{Y}^{miss} \) and the rest in \( \mathcal{Y}^{obs} \)
Receiver operating characteristic

Receiver operating characteristic curves show predictive performance

- Method 1 performs worst ⇒ Agnostic to network structure
- Informal Method 3 yields slightly worst performance than 2 and 4
Inference of association networks

Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Association network inference

- Given a collection of $N_v$ elements represented as vertices $v \in V$
  - Let $x_i \in \mathbb{R}^m$ be a vector of observed vertex attributes, for all $i \in V$
- User-defined similarity $\text{sim}(i, j) = f(x_i, x_j)$ specifies edges $(i, j) \in E$
  - Q: What if $\text{sim}$ values themselves (i.e., edge status) not observable?

**Association network inference**

Infer non-trivial $\text{sim}$ values from vertex observations $\{x_1, \ldots, x_{N_v}\}$

- Various choices to be made, hence multiple possible approaches
  - **Choice of $\text{sim}$**: correlation, partial correlation, mutual information
  - **Choice of inference**: hypothesis testing, regression, ad hoc
  - **Choice of parameters**: testing thresholds, tuning regularization
Correlation networks

Let $X_i \in \mathbb{R}$ be an RV of interest corresponding to $i \in V$

Pearson product-moment correlation as $\text{sim}$ between vertex pairs

$$\text{sim}(i,j) := \rho_{ij} = \frac{\text{cov}[X_i, X_j]}{\sqrt{\text{var}[X_i] \text{var}[X_j]}}, \ i, j \in V$$

Def: the correlation network graph $G(V, E)$ has edge set

$$E = \left\{ (i, j) \in V^{(2)} : \rho_{ij} \neq 0 \right\}$$

Association network inference $\Leftrightarrow$ Inference of non-zero correlations

Inference of $E$ typically approached as a testing problem

$$H_0 : \rho_{ij} = 0 \quad \text{versus} \quad H_1 : \rho_{ij} \neq 0$$
Let $x_{i1}, \ldots, x_{in}$ be observations of zero-mean $X_i$, for each $i \in V$

⇒ Common choice of test statistic are empirical correlations

\[
\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii} \hat{\sigma}_{jj}}}, \quad \text{where } \hat{\Sigma} = [\hat{\sigma}_{ij}] = \frac{X^\top X}{n - 1}
\]

⇒ Convenient alternative statistic is Fisher’s transformation

\[
z_{ij} = \frac{1}{2} \log \left( \frac{1 + \hat{\rho}_{ij}}{1 - \hat{\rho}_{ij}} \right), \quad i, j \in V
\]

⇒ Under $H_0$, $z_{ij} \sim \mathcal{N}(0, \frac{1}{n-3})$ ⇒ Simple to assess significance

⇒ Reject $H_0$ at significance level $\alpha$, i.e., assign edge $(i, j)$ if $|Z_{ij}| > \frac{z_{\alpha/2}}{\sqrt{n-3}}$

Error rate control: $P_{H_0}$ (false edge) $= P_{H_0} \left( |Z_{ij}| > \frac{z_{\alpha/2}}{\sqrt{n-3}} \right) = \alpha$
Interesting testing challenges emerge with large-scale networks

⇒ Suppose we test all \( \binom{N_v}{2} \) vertex pairs, each at level \( \alpha \)

⇒ Even if the true \( G \) is the empty graph, i.e., \( E = \emptyset \)

⇒ We expect to declare \( \binom{N_v}{2} \alpha \) spurious edges just by chance!

⇒ For a large graph, this number can be considerable

⇒ Ex: For \( G \) of order \( N_v = 100 \) and individual tests at level \( \alpha = 0.05 \)

⇒ Expected number of spurious edges is \( 4950 \times 0.05 \approx 250 \)

⇒ This predicament known as the multiple testing problem in statistics
Correction for multiple testing

- **Idea:** Control errors at the level of collection of tests, not individually
- **False discovery rate (FDR) control,** i.e., for given level $\gamma$ ensure

$$
FDR = \mathbb{E} \left[ \frac{R_{false}}{R} \mid R > 0 \right] P( R > 0 ) \leq \gamma
$$

- $R$ is the total number of edges detected; and
- $R_{false}$ is the total number of false edges detected

- **Method of FDR control at level $\gamma$** [Benjamini-Hochberg'94]

**Step 1:** Sort $p$-values for all $N = \binom{N_v}{2}$ tests, yields $p(1) \leq \ldots \leq p(N)$

**Step 2:** Reject $H_0$, i.e., declare all those edges for which

$$
p(k) \leq \left( \frac{k}{N} \right) \gamma
$$
Use correlations carefully: ‘correlation does not imply causation’

- Vertices \( i, j \in V \) may have high \( \rho_{ij} \) because they influence each other

- But \( \rho_{ij} \) could be high if both \( i, j \) influenced by a third vertex \( k \in V \)
  \[ \Rightarrow \text{Correlation networks may declare edges due to latent variables} \]

Partial correlations better capture direct influence among vertices

- For \( i, j \in V \) consider latent vertices \( S_m = \{ k_1, \ldots, k_m \} \subset V \setminus \{ i, j \} \)

- Partial correlation of \( X_i \) and \( X_j \), adjusting for \( X_{S_m} = [X_{k_1}, \ldots, X_{k_m}]^\top \) is

\[
\rho_{ij|S_m} = \frac{\text{cov}[X_i, X_j | X_{S_m}]}{\sqrt{\text{var}[X_i | X_{S_m}] \cdot \text{var}[X_j | X_{S_m}]}} , \quad i, j \in V
\]

Q: How do we obtain these partial correlations?
Computing partial correlations

- Given $X_{S_m} = [X_{k_1}, \ldots, X_{k_m}]^\top$, the partial correlation of $X_i$ and $X_j$ is

$$\rho_{ij|S_m} = \frac{\text{cov}[X_i, X_j \mid X_{S_m}]}{\sqrt{\text{var}[X_i \mid X_{S_m}] \text{var}[X_j \mid X_{S_m}]}} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m} \sigma_{jj|S_m}}}$$

- Here $\sigma_{ii|S_m}, \sigma_{jj|S_m}$ and $\sigma_{ij|S_m}$ are diagonal and off-diagonal elements of

$$\Sigma_{11|2} := \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \in \mathbb{R}^{2 \times 2}$$

- Matrices $\Sigma_{11}, \Sigma_{22}$ and $\Sigma_{21} = \Sigma_{12}^\top$ are blocks of the covariance matrix

$$\text{cov} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \text{ where } W_1 = [X_i, X_j]^\top \text{ and } W_2 = X_{S_m}$$
Various ways to use partial correlations to define edges in $G$

\textbf{Ex:} $X_i, X_j$ correlated regardless of what $m$ vertices we condition upon

\[
E = \left\{ (i, j) \in V^{(2)} : \rho_{ij|S_m} \neq 0, \text{ for all } S_m \in V^{(m)}_{\{i,j\}} \right\}
\]

Inference of potential edge $(i, j)$ as a testing problem

\[
H_0 : \rho_{ij|S_m} = 0 \text{ for some } S_m \in V^{(m)}_{\{i,j\}}
\]
\[
H_1 : \rho_{ij|S_m} \neq 0 \text{ for all } S_m \in V^{(m)}_{\{i,j\}}
\]

Again, given measurements $x_{i1}, \ldots, x_{in}$ for each $i \in V$ need to:

\begin{itemize}
  \item Select a test statistic
  \item Construct an appropriate null distribution
  \item Adjust for multiple testing
\end{itemize}
Testing partial correlations

- Often consider a collection (over $S_m$) of smaller testing sub-problems

$$H'_0 : \rho_{ij|S_m} = 0 \text{ versus } H'_1 : \rho_{ij|S_m} \neq 0$$

- **Statistic**: empirical partial correlations $\hat{\rho}_{ij|S_m}$, or Fisher’s $z$-scores

$$z_{ij|S_m} = \frac{1}{2} \log \left( \frac{1 + \hat{\rho}_{ij|S_m}}{1 - \hat{\rho}_{ij|S_m}} \right)$$

$\Rightarrow$ From asymptotic theory, under $H'_0$ then $z_{ij|S_m} \sim \mathcal{N}(0, \frac{1}{n-m-3})$

- Multiple tests for each $\{i, j\} \in V^{(2)}$. How do we combine $p$-values?

  - If $p_{ij|S_m}$ is the $p$-value for testing $H'_0$ versus $H'_1$ for $\{i, j\}$, use

$$p_{ij}^{\max} = \max \left\{ p_{ij|S_m} : S_m \in V^{(m)}_{\setminus\{i,j\}} \right\}$$

  - FDR control possible from collection $\{p_{ij}^{\max}\}_{i,j}$ [Wille-Bühlmann’06]
Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Case study: Gene-regulatory interactions

- Genes are segments of DNA encoding information about cell functions
- Such information used in the expression of genes
  - Creation of biochemical products, i.e., RNA or proteins
- Regulation of a gene refers to the control of its expression
  - Ex: regulation exerted during transcription, copy of DNA to RNA
    - Controlling genes are transcription factors (TFs)
    - Controlled genes are termed targets
    - Regulation type: activation or repression
- Regulatory interactions among genes basic to the workings of organisms
  - Inference of interactions → Finding TF/target gene pairs
- Such relational information summarized in gene-regulatory networks
Microarray data

- Relative levels of gene expression in the cell can be measured
  - Genome-wide scale data obtained using microarray technologies

- For each gene $i \in V$, measure an expression profile $x_i \in \mathbb{R}^n$
  - Vector $x_i$ has gene expression levels under $n$ different conditions
  - Ex: change in pH, heat level, oxygen concentrations

- Microarray data commonly used to infer gene regulatory interactions
Regulatory interactions in E. coli

- Use microarray data and correlation methods to infer TF/target pairs

![Image representation of 445 microarray expression profiles collected for E. coli, under various conditions, for the 153 genes that are listed as known transcription factors in the RegulonDB database. Larger negative values are indicated with darker shades of blue, and larger positive values, in yellow to orange. Shades of green indicate values comparatively close to zero.]

- **Dataset**: relative log expression RNA levels, for genes in E. coli
  - 4,345 genes measured under 445 different experimental conditions

- **Ground truth**: 153 TFs, and TF/target pairs from database RegulonDB
Three correlation based methods to infer TF/target gene pairs

⇒ Interactions declared if suitable $p$-values fall below a threshold

**Method 1:** Pearson correlation between TF and potential target gene

**Method 2:** Partial correlation, controlling for shared effects of one ($m = 1$) other TF, across all 152 other TFs

**Method 3:** Full partial correlation, simultaneously controlling for shared effects of all ($m = 152$) other TFs

⇒ In all cases applied Fisher transformation to obtain $z$-scores

⇒ Asymptotic Gaussian distributions for $p$-values, with $n = 445$

⇒ Compared inferred graphs to ground-truth network from RegulonDB
Performance comparisons

- ROC and Precision/Recall curves for Methods 1, 2, and 3
  - Precision: fraction of predicted links that are true
  - Recall: fraction of true links that are correctly predicted

- Method 1 performs worst, but none is stellar
  - Correlation not strong indicator of regulation in this data

- All methods share a region of high precision, but a very small recall
  - Limitations in number/diversity of profiles [Faith et al'07]
Predicting new TF/target gene pairs

- In biology, often interest is in predicting **new interactions**

- 11 interactions found for TF \( lrp \), 10 experimentally confirmed (dotted)
  - 5 interacting target genes were new (**magenta**, **red**, **cyan**)
  - 4 present in RegulonDB (**magenta**, **cyan**), but not as \( lrp \) targets
Gaussian graphical model networks

- Suppose variables \( \{X_i\}_{i \in V} \) have multivariate Gaussian distribution
  - Consider \( \rho_{ij} | V \setminus \{i,j\} \) conditioning on all other vertices \( (m = N_v - 2) \)

Theorem

**Under the Gaussian assumption, vertices** \( i, j \in V \) **have partial correlation**

\[
\rho_{ij} | V \setminus \{i,j\} = 0
\]

**if and only if** \( X_i \) **and** \( X_j \) **are conditionally independent given** \( \{X_k\}_{k \in V \setminus \{i,j\}} \)

- **Def:** the **conditional independence graph** \( G(V, E) \) **has edge set**

\[
E = \left\{ (i, j) \in V^{(2)} : \rho_{ij} | V \setminus \{i,j\} \neq 0 \right\}
\]

- **⇒** A special and popular case of partial correlation networks
- **⇒** Also known as **Gaussian Markov random field (GMRF)**
Let $\Sigma$ be the covariance matrix of $X = [X_1, \ldots, X_{N_v}]^T$.

**Def:** The concentration matrix is $\Omega = \Sigma^{-1}$ with entries $\omega_{ij}$.

**Key result:** For GGMs, the partial correlations can be expressed as

$$
\rho_{ij|V\setminus\{i,j\}} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}
$$

$\Rightarrow$ Non-zero entries in $\Omega \iff$ Edges in the graph $G$.

**Inferring $G$ from data in this context known as covariance selection.**

$\Rightarrow$ Classical methods are ‘network-agnostic,’ and effectively test

$$
H_0 : \rho_{ij|V\setminus\{i,j\}} = 0 \quad \text{versus} \quad H_1 : \rho_{ij|V\setminus\{i,j\}} \neq 0
$$

$\Rightarrow$ Often not scalable, and $n \ll N_v$ so estimation of $\hat{\Sigma}$ challenging.

Graphical Lasso

- Sparsity-regularized maximum-likelihood estimator of $\Omega$ [Yuan-Lin’07]

$$\hat{\Omega} \in \arg\max_{\Omega \succeq 0} \left\{ \log \det \Omega - \text{trace}(\hat{\Sigma}\Omega) - \lambda \|\Omega\|_1 \right\}$$

$\Rightarrow$ Effective when $n \ll N_v$, encourages interpretable models

$\Rightarrow$ Scalable solvers using coordinate-descent [Friedman et al’08]

- Performance guarantee: Graphical lasso with $\lambda = 2\sqrt{\frac{\log N_v}{n}}$ satisfies

$$\|\hat{\Omega} - \Omega_0\|_2 \leq \sqrt{\frac{d_{\text{max}}^2 \log N_v}{n}} \quad \text{w.h.p.}$$

$\Rightarrow$ Ground-truth $\Omega_0$, maximum nodal degree $d_{\text{max}}$

- Support consistency for $n = O(d_{\text{max}}^2 \log N_v)$ [Ravikumar et al’11]
Covariance selection meets linear regression

**Idea:** separately estimate neighborhoods \( \mathcal{N}_i := \{j : (i, j) \in E\}, \ i \in \mathcal{V} \)

- Conditional mean of \( X_i \) given \( X_{(-i)} = [X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{N_v}]^\top \) is

\[
\mathbb{E} \left[ X_i \mid X_{(-i)} = x_{(-i)} \right] = \beta_{(-i)}^\top x_{(-i)}
\]

- Entries of \( \beta_{(-i)} \) expressible in terms of those in \( \Omega = \Sigma^{-1} \), namely

\[
\beta_{(-i),j} = -\frac{\omega_{ij}}{\omega_{ii}}
\]

\( \Rightarrow \) Non-zero \( \beta_{(-i),j} \Leftrightarrow \) Non-zero \( \omega_{ij} \) in \( \Omega \Leftrightarrow \) Edge \((i, j)\) in \( G \)

\( \Rightarrow \) In other words, \( \text{supp}(\beta_{(-i)}) := \{j : \beta_{(-i),j} \neq 0\} \equiv \mathcal{N}_i \)

- Suggests inference of \( G \) via least-squares (LS) regression, to estimate

\[
\beta_{(-i)} = \arg \min_{\theta} \mathbb{E} \left[ (X_i - \theta^\top X_{(-i)})^2 \right]
\]
Sparsity and the $\ell_1$ norm

Consider minimizing a quadratic function of $\theta$ as in LS or ridge

Q: What is the effect of an $\ell_1$-norm constraint, i.e., $\|\theta\|_1 = \sum_i |\theta_i| \leq \tau$?

$\Rightarrow$ Level sets touch constrain set in a kink $\rightarrow$ Sparse solution

Lasso estimator enables estimation and variable selection [Tibshirani'94]

$$\hat{\theta}_{\text{Lasso}} = \arg \min_{\theta} \sum_{i=1}^{n} (y_i - x_i^T \theta)^2, \text{ s. to } \|\theta\|_1 \leq \tau$$
Cycle over vertices $i \in V$ and estimate $\hat{N}_i = \text{supp}(\hat{\beta}_{(-i)})$, where

$$\hat{\beta}_{(-i)} \in \arg \min_{\beta \in \mathbb{R}^{N_v - 1}} \left\{ \sum_{p=1}^{n} (x_{pi} - x_p^\top \beta) I_{i \in \beta} + \lambda \|\beta\|_1 \right\}$$

$\Rightarrow$ Separable lasso problems per vertex

No guarantee that $\hat{\beta}_{(-i),j} \neq 0$ implies $\hat{\beta}_{(-j),i} \neq 0$ and vice versa

- Combine information in $\hat{N}_i$ and $\hat{N}_j$ to enforce symmetry
- OR rule: $(i, j) \in E$ if $\beta_{(-i),j} \neq 0$ or $\beta_{(-j),i} \neq 0$. Likewise, AND rule

Support consistency for either rule [Meinshausen-Bühlmann’06]

- Suitable choice of $\lambda$, sparsity of $\Omega_0$, and sample complexity $n \ll N_v$
Inference of GGMs with edges $E = \{(i, j) \in V^{(2)} : \rho_{ij|V\setminus\{i,j\}} \neq 0\}$

**Association network inference:**

Find pairs $\{i, j\}$ for which $\rho_{ij|V\setminus\{i,j\}} \neq 0$

**Covariance selection:**

$\rho_{ij|V\setminus\{i,j\}} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$

Find non-zero entries $\omega_{ij} \neq 0$ in the concentration matrix $\Omega = \Sigma^{-1}$

**Variable selection in linear regression:**

$\beta_{(-i),j} = -\frac{\omega_{ij}}{\omega_{ii}}$

Find non-zero regression coefficients in

$\beta_{(-i)} = \arg\min_{\theta} \mathbb{E}\left[(X_i - \theta^\top X_{(-i)})^2\right]$
Comparative summary

- **Parallelizable** neighborhood-based regression (NBR)
  - Conditional likelihood per vertex $i \in V$, disregards $\Omega \succeq 0$
  - Tends to be computationally faster

- Graphical Lasso minimizes a (regularized) global likelihood
  \[ L(\Omega) = \log \det \Omega - \text{trace}(\hat{\Sigma} \Omega) \]
  - Tends to be (statistically) more efficient

- NBR method tractable even for discrete or mixed graphical models
  - Ising-model selection for $X \in \{-1, +1\}^{N_v}$

Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
In imaging, tomography refers to imaging by sections (e.g., MRI)
Reconstruction algorithms relate ‘external data’ to internal structure

**Goal:** create images of internal aspects of the human body

Predict edge and vertex status in the ‘interior’ of $G$, given only observations $x_i$ for vertices $i \in V$ in the ‘exterior’ of $G$

Most difficult case of topology inference. An ill-posed inverse problem

⇒ **Inverse problem:** invert mapping from ‘internal’ to ‘external’

⇒ **Ill-posed:** the mapping is many-to-one

Most work has dealt with inference of **tree topologies**

Ex: computer network topologies, phylogenetic tree, media cascades
Def: an undirected tree $T = (V_T, E_T)$ is a connected acyclic graph

Nomenclature:
- Rooted tree: tree with a single vertex $r \in V_T$ singled out
- Leaves: subset of vertices $L \subset V_T$ of degree one
- Internal vertices: those vertices in $V_T \setminus \{\{r\} \cup L\}$
- Binary tree: root and internal vertices have at most two children
Given $n$ i.i.d. measurements of RVs $\{X_1, \ldots, X_{N_L}\}$ on $N_L$ vertices.

Consider the family $\mathcal{T}_{N_L}$ of binary trees with $N_L$ labeled leaves.

If we know $r$ then all trees in $\mathcal{T}_{N_L}$ will be rooted at $r$.

Tomographic tree topology inference

Find a tree $\hat{T} \in \mathcal{T}_{N_L}$ that ‘best’ explains the data $\{x_1, \ldots, x_{N_L}\}$.

Often of interest to infer a set of branch weights as well.
Multicast probes: measurements

- **Ex:** Consider inference of computer network topologies, e.g., Internet
- **Multicast packets** sent from a node \((r)\) to multiple destinations \((L)\)
  \(\Rightarrow\) Probes forwarded at routing devices, could be lost en route

For leaves \(\ell \in L\), consider the indicator \(X_\ell = \mathbb{1}\{\ell \text{ received the probe}\}\)

\(\Rightarrow\) Send \(n\) multicast probes to yield data \(\{x_\ell \in \{0, 1\}^n\}_{\ell \in L}\)
Multicast probes: structure

- Think of leaf RVs \( \{X_1, \ldots, X_{N_L}\} \) as samples of a process \( \{X_j\}_{j \in V_T} \)

- Useful notation to describe process’ structure
  - **Def:** closest common ancestor \( a(U) \) to a set of leaves \( U \subseteq L \)
  - **Def:** set \( d(j) \) of all immediate descendants of internal vertex \( j \)

- **Multicast tree enforces hereditary constraints**
  
  \[ X_{a(U)} = 0 \] implies \( X_j = 0 \) for all \( j \in U \)
  
  \[ \text{If } X_j = 1 \text{ for at least one } j \in d(k), \text{ then } X_k = 1 \]
Hierarchical clustering-based methods

- Hierarchical clustering groups $N_L$ objects based on (dis)similarity
  - Entire hierarchy of nested partitions obtained → dendrogram

- Natural tool for tomographic inference of tree topologies
  - $N_L$ leaves as ‘objects’, dendrogram as the inferred tree $\hat{T}$

- Tailor a (dis)similarity to the tomographic inference problem at hand
Multicast probes: dissimilarity

- Shared packet loss rate indicative of close leaves in a multicast tree
- Two types of shared loss between a pair of leaves \( j, k \in L \)
  - **True:** loss of packets in the path common to vertices \( j \) and \( k \)
  - **False:** losses on paths after the closest common ancestor \( a(\{j, k\}) \)
- Net shared loss rate includes both effects ⇒ misleading similarity
  ⇒ Can obtain true shared loss rates via simple packet-loss model
Recall the cascade process \( \{X_j\}_{j \in V_T} \) induced by multicast probing

Specify a Markov model down the tree

- **Root** \( r \): set \( X_r = 1 \)
- **Internal vertex** \( k \): if \( X_k = 0 \), then \( X_j = 0 \) for all \( j \in d(k) \). Otherwise,

\[
P(X_j = 1 \mid X_k = 1) = 1 - P(X_j = 0 \mid X_k = 1) = \alpha_j, \ j \in d(k)
\]

\( \Rightarrow \) Probes successfully transmitted through link \((k, j)\) w.p. \( \alpha_j \)

- **Probe successfully transmitted from** \( r \) **to** \( k \) **w.p.**

\[
P(X_k = 1 \mid X_r = 1) := A(k) = \prod_{j \succ k} \alpha_j
\]

\( \Rightarrow j \succ k \) denotes ancestral vertices of \( k \) in path from \( r \)

- **True shared loss rate** for two leaf vertices \( j, k \in L \) is \( 1 - A(a(\{j, k\})) \)
Estimating shared loss rates

Let $L(k)$ be the set of leaves that are descendants of $k$

- Probability that at least one descendant leaf of $k$ received a packet

$$\gamma(k) = P \left( \bigcup_{j \in L(k)} \{X_j = 1\} \right)$$

- Key: Using probabilistic arguments, can establish the relation

$$1 - \frac{\gamma(k)}{A(k)} = \prod_{j \in d(k)} \left[1 - \frac{\gamma(j)}{A(k)}\right]$$

⇒ Given values $\{\gamma(k)\}_{k \in V_T}$, can solve for the $\{A(k)\}_{k \in V_T}$

- But $\{\gamma(k)\}_{k \in V_T}$ unknown! Use leaf measurements to form estimates

$$\hat{\gamma}(k) = \frac{1}{n} \sum_{i=1}^{n} \max_{j \in L(k)} (x_{ji})$$
Agglomerative hierarchical clustering algorithm

- Greedy, agglomerative algorithm based on shared loss similarities

**S1:** Estimate packet losses \( \hat{\gamma}(j) \) at the leaves \( j \in L \)

**S2:** Estimate shared loss \( 1 - \hat{A}(a(\{j, k\})) \) for all pairs \( j, k \in L \)

Estimate: \( \hat{\gamma}(a(\{j, k\})) = \frac{1}{n} \sum_{i=1}^{n} \max_{s \in \{j, k\}} (x_{si}), \ j, k \in L \)

Solve: \( 1 - \frac{\hat{\gamma}(a(\{j, k\}))}{\hat{A}(a(\{j, k\}))} = \prod_{i \in \{j, k\}} \left[ 1 - \frac{\hat{\gamma}(i)}{\hat{A}(a(\{j, k\}))} \right] \)

**S3:** Merge pair \( \{j^*, k^*\} = \arg \max_{j, k} [1 - \hat{A}(a(\{j, k\}))] \)

**S4:** Exchange \( \{j^*, k^*\} \) for \( a(\{j^*, k^*\}) \) in \( L \) and go back to S2

- Can establish theoretical consistency guarantees for recovering \( T \)
Likelihood-based methods

- Probability models of leaf RVs $\{X_\ell\}_{\ell \in L}$ used for defining (dis)similarities
  - But having such models $f(x \mid T)$ also enables ML inference
- If the $n$ observations $\{x_i\}_{i=1}^n$ are independent, the likelihood is
  $$\mathcal{L}_n(T) = \prod_{i=1}^n f(x_i \mid T)$$
- Models often include other parameters $\theta$ (e.g., the $\alpha_j$) beyond $T$
  - In this case $\mathcal{L}_n(T)$ is an integrated likelihood, namely
    $$\mathcal{L}_n(T) = \prod_{i=1}^n \int_{\theta \in \Theta} f(x_i \mid T, \theta) f(\theta \mid T) d\theta$$
- Integrals may be computationally challenging. The ML estimate is
  $$\hat{T}_{ML} = \arg \max_{T \in \mathcal{T}_{NL}} \mathcal{L}_n(T)$$
Network topology inference problems

Link prediction

Case study: Predicting lawyer collaboration

Inference of association networks

Case study: Inferring genetic regulatory interactions

Tomographic network topology inference

Case study: Computer network topology identification
Sandwich probing

- Consider network tree topology inference via end-to-end probing
  - Packet drops rare (i.e., drop rate < 2%) ⇒ Shared loss rates ineffective
- Alternative measuring time-delay differences: sandwich probes
  - Send small probe to \(i\), then large probe to \(j\), other small probe to \(i\) last
  - Measure time-delay difference (TDD) between small packets

1: Send to MSU1
2: Send to MSU2
3: Send to MSU1

1: Send to MSU1
2: Send to Berkeley
3: Send to MSU1

- If paths overlap, large probe induces high delay in the second small one
  ⇒ Large TDD values indicative of close leaves in the tree topology
Sent sandwich probes every 50 ms to random pairs $j, k \in L$

⇒ Total of 9,567 measured delay differences over 8 minutes

For each pair $j, k \in L$, let $x_{jk}$ be the average TDD

⇒ The Central Limit Theorem suggests $x_{jk} \sim \mathcal{N}(\mu_{jk}, \sigma^2_{jk})$

⇒ Independence of the $x_{jk}$ reasonable by experimental setup
Agglomerative likelihood tree (ALT) algorithm

- Hierarchical clustering with likelihood-based similarity measure
- Let $\ell_{ij}(\mu) = \log f(x_{ij}|\mu)$ be the Gaussian log-likelihood ($\sigma_{ij}^2$ known)
- Initialize a set of vertices $S$ with the leaves, i.e., $S = L$

**Def:** similarity among leaves is estimated mean TDD

$$\hat{\mu}_{ij} = \hat{\mu}_{ji} = \arg \max_\mu [\ell_{ij}(\mu) + \ell_{ji}(\mu)], \ i, j \in L$$

- Merge $\{i^*, j^*\} = \arg \max_{i,j} \hat{\mu}_{ij}$. Exchange $\{i^*, j^*\}$ for $a(\{i^*, j^*\})$ in $S$
- Algorithm then iterates until $|S| = 1$, by merging after calculating

$$\hat{\mu}_{kl} = \hat{\mu}_{lk} = \arg \max_\mu \sum_{m \in L(k)} \sum_{p \in L(l)} [\ell_{mp}(\mu) + \ell_{pm}(\mu)], \ k, l \in S$$

⇒ Recall $L(k)$ is the set of leaves descended by $k$
Inferred topology

- Ground-truth topology obtained via traceroute probing
  - traceroute replies often ‘turned-off’ for security
  - Tomographic topology inference approaches relevant!

- ALT-inferred topology binary by construction ⇒ introduces artifacts

Glossary

- Topology inference
- Link prediction
- Scoring methods
- Logistic regression
- Missing data
- Latent variable models
- Latent eigenmodel
- Association networks
- Correlation networks
- Pearson correlation
- Fisher's transformation
- Multiple testing

- False discovery rate
- Gene-regulatory networks
- Microarray data
- Partial correlation
- Gaussian graphical models
- Concentration matrix
- Variable selection
- Network tomography
- Muticat probing
- Shared packet loss
- Sandwich probing
- Time-delay difference