# Introduction to statistical network analysis

Yuan Zhang

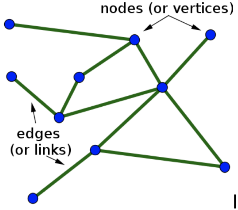
Department of Statistics, Ohio State University

15 May 2025

Download this tutorial:

# Basic concepts

# Basic concepts

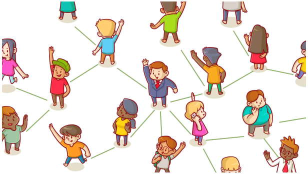



- Nodes (Vertices): participants in the network
  Network size: number of nodes
- Edges (Links): connections or relationships between nodes
- **Examples [nodes (edges)]:**
  - people (friendship)
  - webpages (hyperlinks)
  - paper (citation)
  - genes (regulatory actions)
  - brain regions (oxygen level correlation)
- Adjacency matrix: $A_{i,j}$ is the edge between nodes $(i, j)$

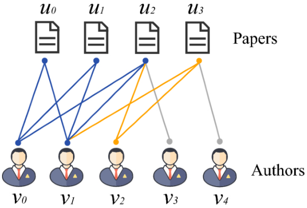
# Basic concepts

Classification of network data: (1) partite

- Unipartite network: any node may connect to anyone else



- Bipartite network: nodes partitioned into two groups, only between-group edges are possible



Classification of network data: (2) edge direction

- Undirected (Symmetric) network: $A_{i,j} = A_{j,i}$
- Directed (Asymmetric) network: $A_{i,j}$ and $A_{j,i}$ might not equal

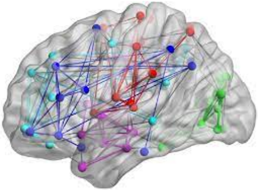



Figure: Left: undirected network; right: directed network

# Basic concepts

Classification of network data: (2) edge weight

- Binary: $A_{i,j} \in \{0,1\}$
  **Example:** friend (1) or non-friend (0)
- Signed: $A_{i,j} \in \{0,1,-1\}$
  **Example:** friend (1) or foe ($-1$) or no interaction (0)
- Weighted:
  **Example:** trade surplus/deficit between countries

# Research topics

Statistical research aims:

- modeling network formation
- finding roles of individual nodes
  **Examples:** community detection, node embedding
- stochastic behavior of network features
  **Examples:** network moments
- link prediction

# Research topics

Network data vs conventional data:

- Networks: **no individual observation**, only relational data
- Deriving "**network analogues** of classical techniques"
  **Examples:**
  - one-/two-sample test
  - cross-validation
  - method-of-moments
  - goodness-of-fit test
  - re/subsampling

# Popular network models

# Overview

Now we survey Network models: Erdos-Renyi model

- Simple $\rightarrow$ complex
- For simplicity, focus on undirected, binary networks
- Model probability matrix:

$$P_{i,j} := \mathbb{P}(A_{i,j} = 1)$$

# Network models: Erdos-Renyi

**Erdos-Renyi (ER) model** (Newman 2018)

- Model formula:

$$P_{i,j} := \mathbb{P}(A_{i,j} = 1) \equiv p$$

- **Definition:** network density is the (order of) average edge probability:

$$\rho_n \asymp \bar{P} := \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} P_{i,j}$$

  In ER model, $\rho_n \asymp p$

- Density is usually a crucial measure for assessing the **problem difficulty** of estimation/testing

# Erdos-Renyi model

$$P_{i,j} := \mathbb{P}(A_{i,j} = 1) \equiv p \asymp \rho_n$$

Model estimation:

- **Moment estimator:**

$$\widehat{p} := \binom{n}{2}^{-1} \sum_{1 \le i < j \le n} A_{i,j}$$

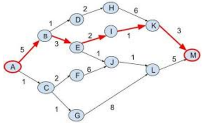
- **Consistency** (Chen et al. 2021): When $\binom{n}{2} p \to \infty$, $\widehat{p}/p \xrightarrow{p} 1$
  Roughly speaking

$$\frac{|\widehat{p} - p|}{p} \underset{\frown}{\overset{\text{approx.}}{}} \frac{1}{\sqrt{n^2 p}}$$

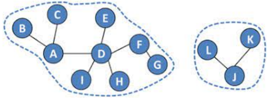
# Erdos-Renyi model

ER model:

- Too simple to fit real-world data well
- But a good null model ("no structure/pattern") for testing (Gao & Lafferty 2017*a*)
- Neat for studying some network properties
  - **Example:** average shortest path (Katzav et al. 2018)




  - **Example:** is the network connected? (Erdos & Rényi 1960)



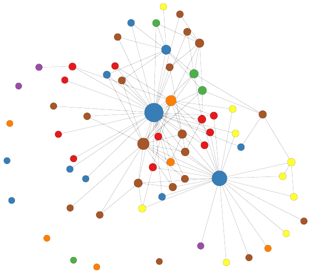

Answers to these questions are the same for: (i) ER model; and (ii) general model with all $P_{i,j} \asymp \rho_n$

# $\beta$-model

- **Definition:** degree is the total number of edges of a node

$$d_i := \sum_{j:j\neq i} A_{i,j}$$

- Average degree: $\bar{d} := \frac{1}{n}\sum_{i=1}^{n} d_i \asymp \rho_n n$
- Degree describes the node's **popularity** (hub, leaf)



# $\beta$-model

Degree-driven network model:

- Network structure entirely determined by degrees
- Network $\rightarrow$ degrees: easy to compute
- Degree sequence $\stackrel{?}{\rightarrow}$ network model

**Configuration model** (Chung & Lu 2002):

- Not a probabilistic model for "network population"
- Given a degree sequence $d_1, \ldots, d_n$, generate network
  1. Set $S := \{1, \ldots, 1, 2, \ldots, 2, \ldots, n, \ldots, n\}$, each $i$ repeats $d_i$ times
  2. Randomly select two entries from $S$, make an edge, delete them from $S$
  3. Repeat step 2 until $S$ is exhausted

# $\beta$-model

$\beta$**-model** (Chatterjee et al. 2011):

- **Formula:**

$$P_{i,j} = \frac{e^{\beta_i+\beta_j}}{1+e^{\beta_i+\beta_j}}$$

  Parameters $\beta_i \in \mathbb{R}$

- **Estimation: MLE**, negative log-likelihood:

$$L(\beta) := \sum_{1\le i<j\le n} \log\left(1+e^{\beta_i+\beta_j}\right) - \sum_{i=1}^{n} \beta_i d_i$$

  Then

$$\widehat{\beta} := \arg\min_{\beta} L(\beta)$$

- **Consistency:** if true $\beta_i$'s are $O(1)$, then $\|\widehat{\beta}-\beta\|_\infty \xrightarrow{p} 0$

# $\beta$-model

$\beta$-model:

$$P_{i,j} = \frac{e^{\beta_i+\beta_j}}{1+e^{\beta_i+\beta_j}}$$

$$L(\beta) := \sum_{1\le i<j\le n} \log\left(1+e^{\beta_i+\beta_j}\right) - \sum_{i=1}^{n} \beta_i d_i$$

Pros and cons:

- (+) Scalable to very large and sparse networks (Shao et al. 2023)
- (+) Privacy protection (only publicize degrees) (Karwa & Slavković 2016)
- ($-$) Limited expressivity

# Stochastic block model (SBM)

- Nodes are partitioned into $K$ communities (blocks, groups)
- **Formula** (Holland et al. 1983)

$$P_{i,j} := \rho_n P_{c_i, c_j}$$

$c_i$: which community node $i$ belongs to

$\rho_n$: network sparsity rescaler

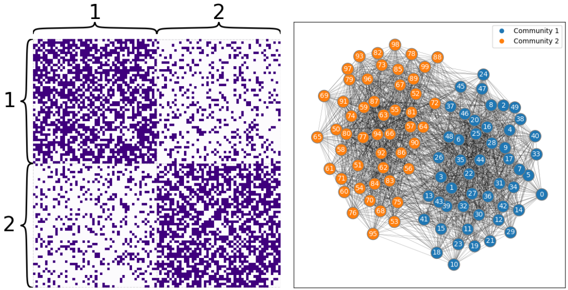



Figure: Left: heatmap of adjacency $A$; right: network plot (nodes have been sorted)

# Stochastic block model (SBM)

SBM:

$$P_{i,j} := \rho_n B_{c_i,c_j}$$

where $B$: community-level edge probability matrix

- **Estimation:** MLE:

$$L(c,B) := \sum_{1\le i<j\le n} \Big\{ A_{i,j} \log B_{c_i,c_j} + (1-A_{i,j}) \log(1-B_{c_i,c_j}) \Big\}$$

$$(\widehat{c},\widehat{B}) := \arg\max_{c_i\in[1:K],B\in[0,1]^{K\times K}} L(c,B)$$

- Estimating $c_{\text{true}}$ is community detection (Abbe 2018)
- If $\rho_n \gg n^{-1}\log n$, then MLE is consistent: as $n\to\infty$,
  - $\widehat{c} = c_{\text{true}}$ a.s. (Bickel & Chen 2009)
  - $\widehat{B} \xrightarrow{p} B_{\text{true}}$

**Question:** How to understand this requirement on $\rho_n$?

# Stochastic block model (SBM)

$$L(c,B) := \sum_{1\le i<j\le n} \left\{ A_{i,j}\log B_{c_i,c_j} + (1-A_{i,j})\log(1-B_{c_i,c_j}) \right\}$$

- Exact MLE infeasible (combinatorial optimization for $c$)
- In practice: Tabu search (Zhao et al. 2012)
  1. Initialize $c$ and $B$
  2. Iterate until convergence:
     - For $i \in [1:n]$: update $c_i$
     - Update $B$

# Stochastic block model (SBM)

SBM:

$$P_{i,j} := \rho_n P_{c_i,c_j}$$

MLE is too slow, can we do better?

- **Alternative formula:**

$$P = \rho_n Z B Z^T$$

  where membership matrix $Z$ may look like:

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

  corresponding to $c = (1, 1, 4, \ldots, 2)$, i.e., each row of $Z$ is a one-hot vector indicating membership

- Verify: $P_{i,j} = \rho_n Z_{i,\cdot} B Z_{j,\cdot}^T = \rho_n B_{c_i,c_j}$

# Stochastic block model (SBM)

SBM (matrix form):

$$P_{n\times n} = \rho_n Z_{n\times K} B_{K\times K} (Z^T)_{K\times n}$$

Therefore:

- $P$ is rank-$K$
- **SVD** of $P$:

  $$P = USU^T$$

  where $S : K \times K$ diagonal; $U : n \times K$ orthonormal: $U^T U = I$
- $U$ only has $K$ **different rows**, one for each community (Qin & Rohe 2013)
- cluster rows of $U \Rightarrow$ true community labels

# Stochastic block model (SBM)

SBM (matrix form):

$$P = \rho_n Z B Z^T$$

In practice, we only observe $A$ ...

- Suppose we know $K$
- Leading-$K$ SVD of $A$:

  $$A \approx \widehat{U}\widehat{S}\widehat{U}^T$$

  (network sparsity absorbed into $\widehat{S}$)
- Cluster the rows of $\widehat{U}$ $\Rightarrow$ community detection
- The above method is called spectral clustering (Lei & Rinaldo 2015)
- **Consistency:** $\rho_n \gg n^{-1}\log n \Rightarrow$ misclassification rate $\xrightarrow{p} 0$

# Stochastic block model (SBM)

SBM:

$$P_{i,j} := \rho_n P_{c_i,c_j}$$

- We only explained “block model”
- What about “stochastic”?
- **Two-stage** generation (Bickel & Chen 2009):
  1. “Stochastic”: each node chooses its community:

     $$c_1,\ldots,c_n \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(q_1,\ldots,q_K)$$

  2. “Block model”:

     $$P_{i,j} := \rho_n P_{c_i,c_j}$$

- Stochastic or not doesn’t matter so much in practice

# Stochastic block model (SBM)

Arguably the “most famous” network model

- Simple, yet expressive
- Communities are very interpretable
  **Examples:**
  - social circles (friendship network) (Yang et al. 2013)
  - functional regions (brain image network) (Wu et al. 2022)
  - research areas (citation network) (Ji et al. 2022)
- Towards general models: SBM with growing $K$

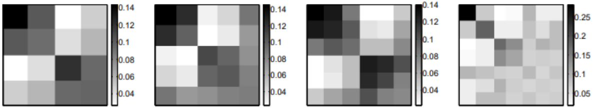
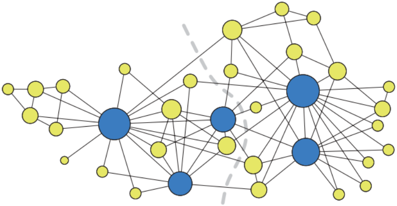



Figure: Choi et al. (2012): as $K$ increases, SBM captures more structural details

# Degree-corrected SBM



- What may go wrong with SBM?
- Why? (Recall spectral clustering ...)
- In community detection, we want to **adjust for degree heterogeneity**

# Degree-corrected SBM

- **Formula:**

$$P_{i,j} = \rho_n \theta_i \theta_j B_{c_i,c_j}$$

where $\theta_i$: degree correction

- **Matrix form:**

$$P = \rho_n \Theta Z B Z^T \Theta$$

where $\Theta = \mathrm{diag}(\theta_1, \ldots, \theta_n)$

- **Identifiability:** require $\sum_{i:c_i=k} \theta_i = 1$ for each community $k$

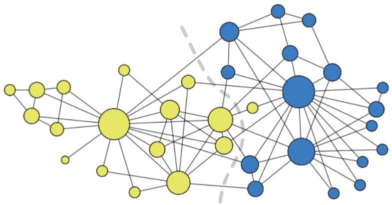


Figure: Karrer & Newman (2011): DCSBM adjusts for degrees

# Degree-corrected SBM

- **Spectral clustering** (Von Luxburg 2007):
  1. Compute Laplacian:

$$L := D^{-1/2} A D^{-1/2}$$

     where $D = \mathrm{diag}(d_1, \ldots, d_n)$
     **Interpretation:** reweighted edges

$$L_{i,j} = \frac{A_{i,j}}{\sqrt{d_i d_j}}$$

     (how much attention do we pay to our relationship?)
  2. Leading-$K$ SVD of $L$:

$$L \approx \widehat{U}\widehat{S}\widehat{U}^T$$

  3. Cluster rows of $\widehat{U}$

# Degree-corrected SBM

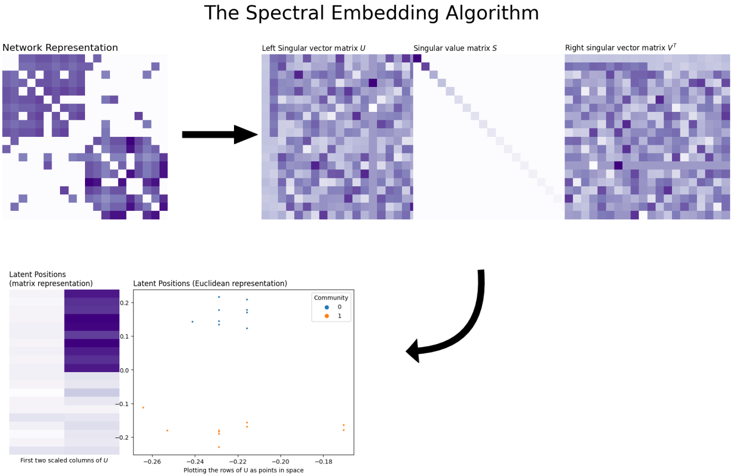



Figure: Illustration of spectral community detection for SBM (Bridgeford et al. 2022)

# Quick recap

Which methods have we seen so far?

- Moment estimator for ER model
- MLE for $\beta$-model and SBM
- Spectral method for SBM

# More about SBM/DCSBM

Overlapping communities:

- Airoldi et al. (2008) **Mixed-Membership Stochastic Blockmodel (MMSB):**
  1. Prior step:
  $$\vec{\pi}_i \overset{\text{i.i.d.}}{\sim} \text{Dirichlet}$$
  2. For each $(i, j)$, draw $c_i \sim$ Multinom.$(\vec{\pi}_i)$, do the same for $j$. Then generate $P_{i,j}$:
  $$P_{i,j} = B_{c_i, c_j}$$

  * **Parameter estimation:** EM algorithm
- Zhang et al. (2020) **Overlapping Continuous Community Assignment Model (OCCAM):**
  - **Matrix form:**
  $$P = \rho_n \Theta Z B Z^T \Theta$$

    but now $Z$ may have continuous rows, satisfying $\|Z_{i,\cdot}\|_2 = 1$ Similarly, (Jin et al. 2024): $\|Z_{i,\cdot}\|_1 = 1$
  - **Parameter estimation:** spectral clustering

# More about SBM/DCSBM

Other estimation methods:

- Modularity (Girvan & Newman 2002, Newman 2006)

$$\frac{1}{2m}\sum_{i<j}\left(A_{i,j}-\frac{d_i d_j}{2m}\right)\mathbb{1}_{[c_i=c_j]}$$

  $m$: total number of edges

- Analytical approximate EM algorithm (Ball et al. 2011)
- Semi-definite programming (SDP)

$$\arg\max_X \langle A, X\rangle, \quad \text{s.t.}, X_{ii}\equiv 1, X\succeq 0$$

  Then spectral cluster $X$ (Cai & Li 2015, Amini & Levina 2018)

- "SCORE" (Jin 2015)
- Spectral clustering + majority vote refinement (Gao et al. 2018)

# More about SBM/DCSBM

Extension: popularity-adjusted SBM (Sengupta & Chen 2018):

- **Motivation:** overall sparse but locally dense networks
  DCSBM cannot describe this scenario:
  - $i, j \in$ same community $k$
  - $i$ is globally more popular than $j$
  - ... but not among members of some community $\ell$
- **Formulation:** for $i \in$ community $k$, $j \in$ community $\ell$:

$$P_{i,j} = \Lambda_i^{(\ell)} \Lambda_j^{(k)}$$

  where $\Lambda$ encodes:
  - community-level edge probabilities
  - node's popularity for each $(k, \ell)$

  You can think: $\Lambda_i^{(\ell)} = \theta_i^{(\ell)} \sqrt{\rho_n B_{k,\ell}}$ (recall $c_i = k$)
- **Reduce to DCSBM:**

$$\Lambda_i^{(\ell)} = \theta_i \sqrt{\rho_n B_{k,\ell}} \quad \Rightarrow \quad P_{i,j} = \rho_n \theta_i \theta_j B_{k,\ell}$$

- MLE for $\Lambda$'s $\Rightarrow$ community detection

# More about SBM/DCSBM

popularity-adjusted SBM (Cont'd)

$$P_{i,j} = \Lambda_i^{(\ell)} \Lambda_j^{(k)}$$

This is subtle: MLE for $\Lambda$'s $\Rightarrow$ community detection

- MLE gives you all $\Lambda_i^{(\ell)}$'s ..
- ... but that's insufficient for determining $P_{i,j}$
- For each $(i, j)$:

$$P_{i,j} = \Lambda_i^{(?)} \Lambda_j^{(?)}$$

# Random dot-product model (RDPG)

- Generalization of SBM/DCSBM (Young & Scheinerman 2007)
- **Formula:** recall DCSBM (no $\rho_n$):

$$P = \Theta Z B Z^T \Theta$$

Suppose $B \succ 0$ (diagonal all positive), then

$$P = (\Theta Z B^{1/2}) \cdot (\Theta Z B^{1/2}) =: X X^T$$

where $X \in \mathbb{R}^{n \times K}$: latent space matrix

- In other words:
  - Each node has an embedding $X_{i,\cdot}$
  - Edge probability: $P_{i,j} = \langle X_{i,\cdot}, X_{j,\cdot} \rangle$
- **Estimation:** can estimate $X$ up to a rotation/reflection $O$: Under mild conditions, $\widehat{X}$ by SVD: $\|\widehat{X} - XO\|_{2\to\infty} \xrightarrow{p} 0$ (Cape et al. 2019); CLT (Athreya et al. 2022)

# Random dot-product model (RDPG)

Approximately low-rank structure is ubiquitous

- Udell & Townsend (2019): for any $A$

$$\inf_{\mathrm{rank}(\widehat{A}) \le C \log n / \varepsilon^2} \|A - \widehat{A}\|_{\max} \le \varepsilon \|A\|$$

- Xu (2018): Holder smoothness in graphon (will explain later) $\Rightarrow$ polynomial eigenvalue decay

$$\sum_{i>r} \lambda_i^2 = O(r^{-\beta} + n^{-1})$$

# Random dot-product model (RDPG)

$$P = XX^T, \quad A \approx \widehat{X}\widehat{X}^T, \quad \widehat{X} \approx XO$$

Assortativity constraint?

- Generalized RDPG (Rubin-Delanchy et al. 2022)

$$P = X \begin{pmatrix} I_p & 0 \\ 0 & -I_q \end{pmatrix} X^T$$

**Estimation:** still SVD $A$:

$$A = \widehat{U}\widehat{S}\widehat{V}^T$$

compare the signs of columns: $\widehat{U}_{\cdot,i}$ vs $\widehat{V}_{\cdot,i}$ to know sign

# Random dot-product model (RDPG)

$$P = XX^T, \quad A \approx \widehat{X}\widehat{X}^T, \quad \widehat{X} \approx XO$$

Interpretability

- Rows $i$ of $\widehat{X}$: embedding of node $i$

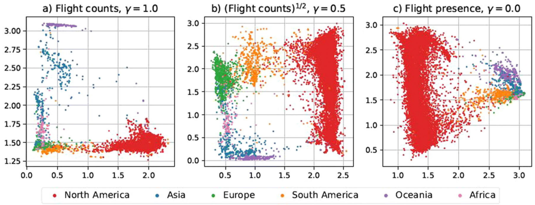



Figure: Gallagher et al. (2024): embedding of airports in flight network

- However, this embedding is limited to that particular network, due to the unknown $O$
  Cannot be directly used for comparing nodes' roles across networks

# Latent space models

Distance model (Hoff et al. 2002)

- **Formulation:**

$$\text{Logit}(P_{i,j}) := \alpha + \beta x_{i,j} + \|z_i - z_j\|$$

  where
  - $x_{i,j}$: observable edge covariate
    (two node covariates $\rightarrow$ edge covariate)
  - $z_i$: latent space position
- **Identifiability:** $\sum_{i=1}^n z_i = \vec{0}$
- **Estimation:** Variational EM

Multiplicative $Z$ term (Ma et al. 2020)

- **Formulation:**

$$\text{Logit}(P_{i,j}) := \alpha + \beta x_{i,j} + \langle z_i, z_j \rangle$$

  Identifiability condition omitted here
- **Estimation:** (regularized) MLE by (projected) gradient

# ERGM

Exponential random graph models (ERGM) (Hunter et al. 2008)

- Directly model the distributions of subgraphs (motifs)
- Dependent edge generation
- Not a convenient generative model (model $\stackrel{?}{\to}$ data)

**Formula:**

- Set of motifs (subgraphs) $\mathscr{R}_1, \ldots, \mathscr{R}_m$
  **Examples:** `edge`, `triangle`, `three-star`
- Given an adjacency matrix $a$, let $R(a)$ count the frequency of $\mathscr{R}$ in $a$
- Network likelihood (Hunter & Handcock 2006):

$$\mathbb{P}(A = a) \propto \exp\Big\{ \sum_{i=1}^{m} \theta_i R_i(a) \Big\}$$

To fit an ERGM to data:

- Choose the set of motifs, compute $R_i(a)$'s
- Estimate $\theta_i$'s

# ERGM

$$\mathbb{P}(A = a) \propto \exp\Big\{\sum_{i=1}^{m} \theta_i R_i(a)\Big\}$$

Given $\theta$, how to generate data?

- Start with an initial adjacency
- Each time, decide whether to flip an edge $(i, j)$, resulting $A$ (before flip) $\to A'$ (after flip)
- **Metropolis-Hastings** (Liu & Liu 2001): accept the flip w.p.

$$\min\Big(1, \exp\Big[\sum_i \theta_i\{R_i(A') - R_i(A)\}\Big]\Big)$$

- Iterate many rounds (with burn-in and thinning)
  - Burn-in: discard the first few rounds (e.g., first 1000 rounds)
  - Thinning: take data from every few rounds, as if they were i.i.d.

# ERGM

$$\mathbb{P}(A = a) \propto \exp\Big\{\sum_{i=1}^{m} \theta_i R_i(a)\Big\}$$

Given $A = a$, how to estimate $\theta$?

- MLE is hard, don't know normalizing constant
- Score equation:

  $$\nabla \ell(\theta) = 0,$$

  that is, for each $i$,

  $$R_i(a) = \mathbb{E}[R_i(A)]$$

- Robbins–Monro update (Snijders 2002):

  $$\theta_i^{(n+1)} = \theta_i^{(n)} + \alpha_n \Big\{ R_i(a) - \widehat{\mathbb{E}}^{\theta^{(n)}}[R_i(A)] \Big\},$$

  where: $\widehat{\mathbb{E}}[R_i(A)]$ evaluated by MH-MC
  step size $\alpha_i$: $\sum_i \alpha_i = \infty$, $\sum_i \alpha_i^2 < \infty$

# Graphon model

Overview:

- A very general framework
- Contains the aforementioned models as special cases

**Two-stage network generation:**

1. Latent node positions: $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim}$ Uniform$[0,1]$
2. Latent graphon function:

$$P_{i,j} = f(X_i, X_j) \tag{1}$$

Understandings:

- $X_i$ describes “what kind of node”
- $f$ captures all structural information in the network

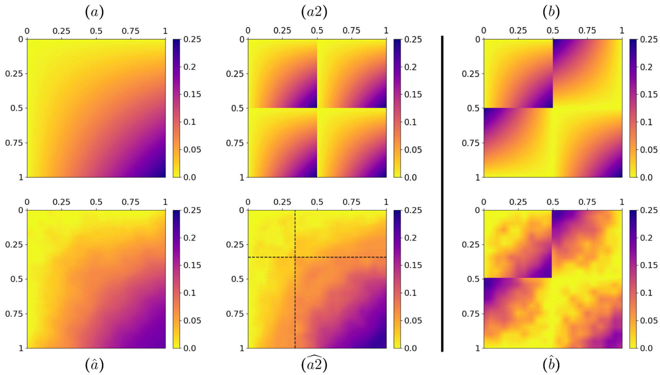



Figure: Illustration of some graphons (Sischka & Kauermann 2025)

# Graphon model

**Examples:**

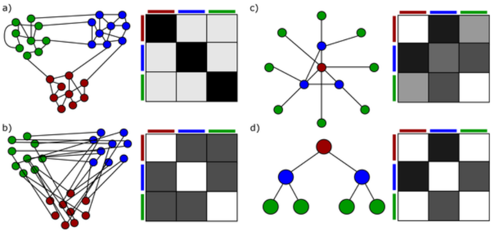
- ER model: $f(x,y) \equiv p$
- $\beta$-model:

$$\beta_i = g(X_i)$$
$$f(x,y) = \text{Logit}^{-1}(g(X_i) + g(X_j))$$

for proper $g(\cdot)$

**Examples:** SBM

- $K$ communities $\Leftrightarrow$ partition $[0,1]$ into $K$ sub-intervals
- $X_i \in k$th interval $\Leftrightarrow i \in$ community $k$
- Graphon $f$: piece-wise constant



# Graphon model

**Aldous-Hoover theorem** (Zhao 2023)

- Exchangeable network: for any permutation $\pi : \mathbb{N} \to \mathbb{N}$,

$$(A_{i,j}) \stackrel{d}{=} (A_{\pi(i),\pi(j)})$$

- Aldous-Hoover: Exchangeable network has the representation

$$A_{i,j} \sim \text{ Bernoulli}(f(X_i, X_j)),$$

  for $X_i \stackrel{\text{i.i.d.}}{\sim}$ Uniform$[0,1]$
- Exchangeability $\Rightarrow$ graphon model
- All nodes i.i.d. choose their roles $\Rightarrow$ exchangeabililty

# Graphon model

**Examples:**

- RDPG with i.i.d. latent positions $\Rightarrow$ graphon model
- Latent space model with i.i.d. latent positions $\Rightarrow$ graphon model
- Recall ERGM:

$$\mathbb{P}(A = a) \propto \exp\left\{ \sum_{i=1}^{m} \theta_i R_i(a) \right\}$$

where each $R_i(a)$: some motif count
Chatterjee & Diaconis (2013): ERGM $\approx$ graphon model (as $n \to \infty$)

# Graphon model

$$X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0,1]$$
$$P_{i,j} = f(X_i, X_j)$$

Graphon estimation:

- Cannot estimate $f$ or $X_i$'s (not identifiable)
- May estimate $P_{i,j}$'s
- Smoothness assumption on $f$ [in equiv. class]
- How to estimate $P_{i,j}$'s?

# Graphon model

$$X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0,1]$$
$$P_{i,j} = f(X_i, X_j)$$

Graphon estimation:

- **Method 1:** universal singular value thresholding (USVT) (Chatterjee 2015)
  1. SVD $A$:
  $$A = \widehat{U}\widehat{S}\widehat{V}^T, \quad (S \text{ diag. sorted})$$
  2. Keep singular values above $C\sqrt{n}$ for $C > 2$
     Say, keep the first $K \Rightarrow (\widetilde{U}, \widetilde{S}, \widetilde{V}) = (\widehat{U}_{\cdot,1:K}, \widehat{S}_{1:K,1:K}, \widehat{V}_{\cdot,1:K})$
  3. Estimator:
  $$\widehat{P} := \widetilde{U}\widetilde{S}\widetilde{V}^T$$

Achieves SOTA MSE (Xu 2018, Luo & Gao 2024)

# Graphon model

$$X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0,1]$$
$$P_{i,j} = f(X_i, X_j)$$

Graphon estimation:

- **Method 2:** Neighborhood smoothing (NS) (Zhang et al. 2017)
  1. Dissimilarity measure of graphon slices $f(\cdot, X_i)$ and $f(\cdot, X_j)$:
  $$d(i,j) := \max_{i' \neq i,j} |\langle A_{i,\cdot} - A_{j,\cdot}, A_{i',\cdot} \rangle| / n$$
  2. For each node $i$, set neighborhood $\mathcal{N}_i := \{j : \text{smallest } h = \log n / n \text{ proportion of } d(i,j)\text{'s}\}$
  3. Estimator:
  $$\widehat{P}_{i,j} := \frac{1}{|\mathcal{N}_i|} \sum_{i' \in \mathcal{N}_i} A_{i',j}$$

Achieves state-of-the-art "$2 \to \infty$ error"; explicit formula

## Selected research topics

# Overview

We will cover these selected topics

- Method-of-moments
- Goodness-of-fit tests
- Cross-validation
- Graph matching
- Extensions to weighted, directed, bipartite graphs
- Other topics (briefly mention)

# Method-of-moments

Network moments

- **Definition:** a network moment indexed by a motif $R$ is

$$\widehat{U}_n := \binom{n}{r}^{-1} \sum_{i_1 < \cdots < i_r} \mathbb{1}_{[A_{i_1,\ldots,i_r} \cong R]}$$

- **Examples:** triangle count

$$\widehat{U}_n = \binom{n}{3}^{-1} \sum_{i<j<k} A_{i,j} A_{j,k} A_{k,i}$$

- **CLT:**

$$\frac{\widehat{U}_n - \mathbb{E}[U_n]}{\widehat{S}_n} \xrightarrow{p} N(0,1),$$

where $S_n$ is some variance estimator (Bickel et al. 2011, Zhang & Xia 2022)

- Moments can be used to identify network models
Application: network two-sample test

# Goodness-of-fit tests

Is the network "structureless"? (Gao & Lafferty 2017*b*)

- $H_0 : P_{i,j} \equiv p$ (ER as null model)
- $H_a$ : non-constant graphon model
- **Method-of-moment test:** null distribution:

$$n^{3/2} g(p) \cdot (T_2, T_3)^T \xrightarrow{d} N(0, I_2), \tag{2}$$

where $T_2 = E^3 - T$ and $T_2 = 3E^2(1-E) - V$, here:

- $E$ counts edges
- $T$ counts triangles
- $V$ counts V-shapes

$g(\cdot)$ is known function; easy to estimate $p$

# Goodness-of-fit tests

GoF test for SBM (Lei 2016)

- $H_0 : K = K_0$ vs $H_a : K > K_0$
- Based on observed $A$, fit SBM with $K = K_0$, get $\widehat{P}$
- **Idea:** is the residual network "pure noise"?
- Test statistic:

$$\widetilde{A}_{i,j} := \frac{A_{i,j} - \widehat{P}_{i,j}}{\sqrt{(n-1)\widehat{P}_{i,j}(1-\widehat{P}_{i,j})}}, \quad i \neq j$$

  Then under $H_0$,

$$n^{2/3} \cdot \left\{ \pm \lambda_1(\widetilde{A}) - 2 \right\} \xrightarrow{d} \text{Tracy-Widom}(1),$$

  where $\lambda_1 \geq \cdots \geq \lambda_n$ are sorted eigenvalues
- **Application: decide** $K$

# Goodness-of-fit tests

Key to build valid GoF method:

- Number of continuous parameters is small under $H_0$
- **Example:** ER model has 1 parameter
- **Example:** SBM has $O(n+K^2)$, but the $n$ memberships can be perfectly recovered for not-very-sparse networks, so we just deal with $O(K^2)$ parameters
- **Example:** DCSBM (Zhang & Amini 2023, Jin et al. 2025)
  $H_0$ : model is an SBM (no degree corrections)

# Goodness-of-fit tests

**No known method** to test GoF for:

- $\beta$-model
- Latent space models, including RDPG
- General graphon model

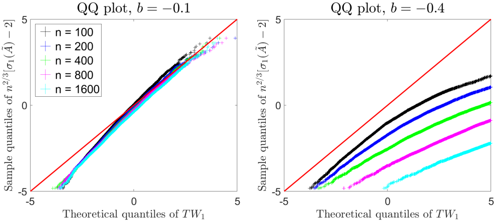



Figure: Shao et al. (2023): exactly mimicking Lei (2016) for $\beta$-model won't work (each curve)

# Cross-validation

CV for network data?

- **Motivation:** select tuning parameter(s)
  **Example:** number of communities $K$ in SBM
  **Example:** bandwidth $h$ in neighborhood smoothing
- **Challenge:** we may observe only one $A$, how to split data?

**Method 1 (node-split CV)** (Chen & Lei 2018)

1. Write
$$A = \begin{pmatrix} A^{(11)} & A^{(12)} \\ A^{(21)} & A^{(22)} \end{pmatrix}$$
2. Use $[A^{(11)}, A^{(12)}]$ to train model (say, SBM)
3. Validation: look at how well fitted model predicts $A^{(22)}$
   (beware $A^{(21)} = \{A^{(12)}\}^T$ already used for training)

# Cross-validation

**Method 2 (edge-splitting CV):** (Li et al. 2020)

1. Randomly sample $M_{i,j} \overset{\text{i.i.d.}}{\sim}$ Bernoulli$(q)$; set $M_{j,i} = M_{i,j}$
   Training data: $\{(i,j) : M_{i,j} = 1\}$
   Validation data: $\{(i,j) : M_{i,j} = 0\}$
2. Fit a model to training data using, say, low-rank method
   May fill in zeros to validation data spots
   This outputs $\widetilde{A}$, then $\widehat{A} = \widetilde{A}/q$ is link predictor
3. See how well $\widehat{A}$ predicts $A$ entries on validation data

# Graph matching

- **Data:** two adjacency matrices $A^{(1)}, A^{(2)} \in \{0,1\}^{n\times n}$
- **Problem:** estimate a permutation matrix $\Pi$ minimizing

$$\|A^{(1)} - \Pi A^{(2)} \Pi^T\|_F^2 \tag{3}$$

- **Applications** (Conte et al. 2004, Lyzinski et al. 2015):
  - Account matching in different social networks
  - Connectome alignment in brain imaging
  - Structurally similar chemical particles
  - Network two-sample test
- **Complexity:** (3) is NP (not known to be P)
- Statistical question: if $A$'s are structured, would the problem be more solvable?

# Graph matching

**Approach 1: low-rank approximation**

- Assume model is low-rank: $A \sim P = XX^T$
- Leading-$k$ SVD:

$$A \approx \widehat{X}\widehat{X}^T$$

- **Davis-Kahan theorem:**

$$\widehat{X} \approx XO$$

for (inestimable) orthonormal $O : O^T O = I$

- Minimize:

$$\|\widehat{X}^{(1)} - \Pi\widehat{X}^{(2)}O\|_F$$

over $\Pi$ and $O$

# Graph matching

$$\|\widehat{X}^{(1)} - \Pi\widehat{X}^{(2)}O\|_F \tag{4}$$

**Approach 1: low-rank approximation (cont'd)**

- joint optimization of (4) is still difficult
- Zhang (2018): first estimate $O$, then $\Pi$ (no iteration!)
- **Key idea:** match row distributions, get $\widehat{O}$; then easy to estimate $\Pi$
- Alternatively, Tang et al. (2014) uses a rotation-invariant kernel to run MMD (Gretton et al. 2012) on $X$-rows

# Graph matching

**Approach 2: local network moments**

- **Correlated ER model:**

$$\mathrm{Cor}\left(A^{(1)}_{i,j}, A^{(2)}_{\pi^*(i),\pi^*(j)}\right) = \rho$$

  for some true permutation $\pi^*$

- **Method 1: degree profile** (Ding et al. 2021):
  1. For each node $i$, compute degree distribution of its neighbors: $F_i$
  2. Similarity matrix:

$$S_{i,j} := \mathrm{distance}(F_i, F_j)$$

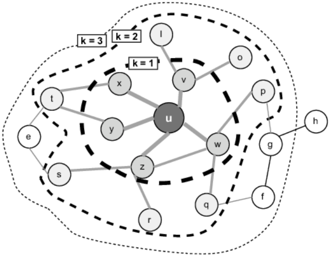
  3. Estimate $\pi$ by Hungarian algorithm

**Approach 2: local network moments (cont'd)**

- **Method 2: large neighborhood statistics** (Mossel & Xu 2020):
  1. Seed $I_0$: a small set of correctly matched nodes
  2. For each $k$ and each $j_1 \in$ net 1, $j_2 \in$ net 2, compute

     $$w_{k;j_1,j_2} := \{i : i \in I_0, i \in \{k\text{-hop neighborhoods of both } i \text{ and } j\}\}$$

     $k$ : tuning parameter; if $|w_{k;j_1,j_2}|$ is large, match $(j_1, j_2)$
- Many recent (strong) results (Ding & Du 2023, Mao et al. 2024) and more ...



# Extensions to weighted, directed, bipartite graphs

**Weighted networks:**

- $A_{i,j} \sim$ Bernoulli $\Rightarrow A_{i,j} \sim \mathscr{F}_\theta$
- **Example: Poisson $\beta$-model** (Yan et al. 2016)

$$\mathbb{P}(A_{i,j} = x) = \frac{e^{(\beta_i+\beta_j)x}}{x!} e^{-e^{\beta_i+\beta_j}}$$

- **Example: weighted SBM** (Xu et al. 2020)

$$A_{i,j} \sim \mathscr{F}_{c_i,c_j}$$

  There are $\binom{K}{2}$ different $\mathscr{F}$'s
- Typically the same methods (MLE, spectral, ...) work for the weighted version

# Extensions to weighted, directed, bipartite graphs

**Directed networks:**

- Each node may have two roles: sender and receiver
- **Example: directed SBM** (Wang & Wong 1987)

$$\mathbb{P}(A_{i,j} = 1) = P_{c_i, d_j},$$

  where
  - $c_i$: community label as sender
  - $d_j$: community label as receiver
- Spectral clustering:

$$A \approx \widehat{U} S \widehat{V}^T,$$

  then:
  - Cluster rows of $\widehat{U}$ to estimate $\vec{c}$
  - Cluster rows of $\widehat{V}$ to estimate $\vec{d}$

# Extensions to weighted, directed, bipartite graphs

**Bipartite networks:**

- Two node sets of distinct meanings
  **Example:** recommender system: buyers $\leftrightarrow$ goods
  **Example:** drug-target network
- $A$ is $m \times n$
- Method and theory: mostly similar to directed networks
  Some differences:
  - Unlike directed networks, no self-loops
  - Asymptotics: $m, n$ may grow at different rates

# Other topics

# Resampling networks

Problem description:

- **Observe:** one $A$
- **Goal:** sampling distribution of some model functional (e.g., triangle count)

Recall i.i.d. case: observe $X_1, \ldots, X_n$, goal: sample mean

- **Parametric bootstrap:** use $X$ to fit a model $F_{\widehat{\theta}}$, resample from $F$, repeatedly get $X_1^*, \ldots, X_n^*$ and bootstrapped $\bar{X}^*$
- **Non-parametric bootstrap:** resample from $X$, repeatedly get $X_1^*, \ldots, X_n^*$ and bootstrapped $\bar{X}^*$
- These two methods produce similar $\bar{X}^*$ distributions

# Resampling networks

Network? ... observe $A$, goal: triangle count

- **Method 1:** use $A$ to fit $\widehat{P}$, then repeatedly generate $A^*$ from $\widehat{P}$ and compute triangle count
- **Method 2** (Green & Shalizi 2017): repeatedly resample nodes, each time induce $A^*$, count triangles
  Specifically, $k_1^*, \ldots, k_n^*$ resampled from $[n]$, then

$$A^*_{i,j} := A_{k_i,k_j}$$

- Methods 1 and 2 may give you very different triangle count sampling distributions!

What happened?

- **Method 1.5:** use $A$ to fit $\widehat{P}$, then still resample nodes, each time induce $\widehat{P}^*$, then sample $A^*$ from $\widehat{P}^*$, count triangles
- Method 1.5 $\approx$ Method 2 $\neq$ Method 1
- Reading: Zhang & Xia (2022)

# Dynamic networks

What do data look like?

- **Scenario 1: network snapshots:** $\{A^{(t)}\}_{t=1}^{T}$ (Sewell & Chen 2015, Pensky 2019)
  **Examples:** friendship, international trade
- **Scenario 2: stream of edges:** $(i, j, t)$, where $t$ : time stamp (Less studied) (Perry & Wolfe 2013, Fang et al. 2024)
  **Examples:** emails, contact (epidemiology)

Network snapshots:

- Build model for each layer
- Parameters evolve over time / across layers
- Different layers may have similar/shared parameters
- Tensor analysis (Dr. Xia's tutorial tomorrow)
- Some works: convert edge stream to snapshots

# Nodal covariates

Sometimes you observe more than a network ...

- **Data:** $A \in \mathbb{R}^{n \times n}$ and $X \in \mathbb{R}^{n \times p}$
- **Examples:** individual covariates (gender, age, job, ...)
- Structure of $X$ may relate to structure of $A$

Use $X$ to help modeling $A$
**Example:** covariate-assisted community detection (Binkiewicz et al. 2017)

- **Method:**

$$\text{Apply spectral clustering to:} \quad A + \lambda X X^T$$

  $\lambda$ : tuning parameter

- Edge covariates can be utilized similarly

# Nodal covariates

Use $A$ to model/predict $X$

**Example:** network regression (Le & Li 2022)

- **Data:** on each node, observe $(X_i, Y_i)$
- **Goal:** use $X$ and $A$ to predict $Y$
- **Model:** $C_X := \text{ColumnSpace}(X)$, $C_P := \text{ColumnSpace}(P)$,

$$\mathbb{E}[Y|X] = \underbrace{X\beta}_{\in C_X} + \underbrace{X\theta}_{\in C_X \cap C_P} + \underbrace{\alpha}_{\in C_P}$$

  and $(X\beta) \perp (X\theta)$ and $(X\theta) \perp \alpha$
- **Estimation strategy:** first estimate $\theta$

# Nodal covariates

Some understandings

- Beware of differences in $X$ and $A$ structures (considering $X$ might or might not help $A$'s model fitting) (Zhang et al. 2016, Deshpande et al. 2018)
- Be careful when building joint models
  **Example:** shall we worry about this model for joint community detection?

$$P_{i,j} \propto \exp\left\{B_{c_i,c_j} + C_{X_i,X_j}\right\}$$

  where suppose each $X_i \in [K]$, here
  - $c_i$: network community label
  - $X_i$: covariate community label

# Sampling network

What if we have to explore the network?

- We might not be able to sample nodes i.i.d. from $\{$all kinds of nodes$\}$
- **Example (snowball):** start from an initial set of people, ask them to refer friends to our study, and friends ask their further friends, ...
- Study strategy:
  1. Model the true (complete) network
  2. Specify a sampling scheme
  3. Analyze what the sampling scheme will produce

# Sampling network

**Example:** random walk (Athreya & Röllin 2016)

- Mechanism. Start from $i_0$; randomly pick a neighbor, walk to $i_1$; randomly pick a neighbor of $i_1$, walk to $i_2$, ... Obtain the induced network observation
- **Theorem.** Network $\sim$ graphon $f(u,v)$, then the network surveyed by random walk is approximately generated by

  $$\text{degree-weighted graphon:} \quad f(X_i, X_j),$$

  but now $X_i$ is **<u>not</u>** Uniform$[0,1]$, instead, has pdf:

  $$p_X(u) := \frac{\int_0^1 f(u,v)\mathrm{d}v}{\int_0^1\int_0^1 f(u',v)\mathrm{d}u'\mathrm{d}v} \propto \mathbb{E}[\text{degree}] \text{ at } u$$

- Many more other sampling schemes (non-backtracking, snowball, ...) (Rohe 2019)

# Core periphery structure

- Aforementioned models attempt to model **all** nodes
- But often there are seclusive nodes with few connections (Rombach et al. 2014)
- **Core:** dense and structured subnetwork
- **Periphery:** the rest of network

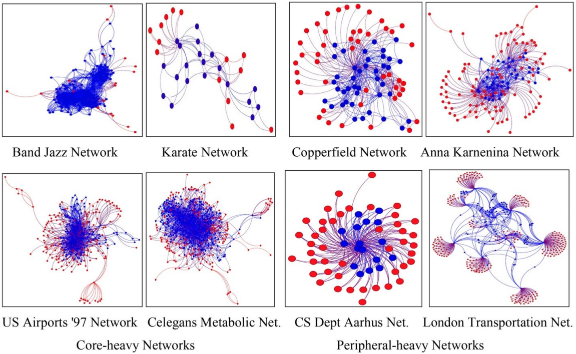



Figure: Blue: core; red: periphery (Meghanathan 2024)

# Core periphery structure

Miao & Li (2023)

- Informative model for core part
- Non/Less-informative model for periphery part

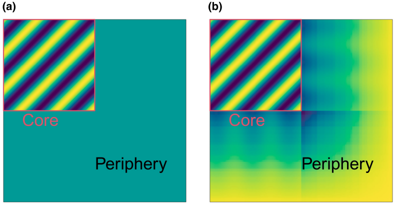



Figure: Two periphery models: left: ER; right: configuration (Miao & Li 2023)

# References

Abbe, E. (2018), ‘Community detection and stochastic block models: recent developments’, *Journal of Machine Learning Research* **18**(177), 1–86.

Airoldi, E. M., Blei, D., Fienberg, S. & Xing, E. (2008), ‘Mixed membership stochastic blockmodels’, *Advances in neural information processing systems* **21**.

Amini, A. A. & Levina, E. (2018), ‘On semidefinite relaxations for the block model’, *The Annals of Statistics* **46**(1), 149–179.

Athreya, A., Cape, J. & Tang, M. (2022), ‘Eigenvalues of stochastic blockmodel graphs and random graphs with low-rank edge probability matrices’, *Sankhya A* pp. 1–28.

Athreya, S. & Röllin, A. (2016), ‘Dense graph limits under respondent-driven sampling’.

Ball, B., Karrer, B. & Newman, M. E. (2011), ‘Efficient and principled method for detecting communities in networks’, *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* **84**(3), 036103.

Bickel, P. J. & Chen, A. (2009), ‘A nonparametric view of network models and newman–girvan and other modularities’, *Proceedings of the National Academy of Sciences* **106**(50), 21068–21073.

Bickel, P. J., Chen, A. & Levina, E. (2011), ‘The method of moments and degree distributions for network models’.

Binkiewicz, N., Vogelstein, J. T. & Rohe, K. (2017), ‘Covariate-assisted spectral clustering’, *Biometrika* **104**(2), 361–377.

Bridgeford, E. W., Loftus, A. & Vogelstein, J. T. (2022), 'Hands-on network machine learning with scikit-learn and graspologic'.
**URL:** *https://docs.neurodata.io/graph-stats-book/coverpage.html*

Cai, T. T. & Li, X. (2015), 'Robust and computationally feasible community detection in the presence of arbitrary outlier nodes', *The Annals of Statistics* **43**(3), 1027–1059.

Cape, J., Tang, M. & Priebe, C. E. (2019), 'The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics', *The Annals of Statistics* **47**(5).

Chatterjee, S. (2015), 'Matrix estimation by universal singular value thresholding'.

Chatterjee, S. & Diaconis, P. (2013), 'Estimating and understanding exponential random graph models'.

Chatterjee, S., Diaconis, P. & Sly, A. (2011), 'Random graphs with a given degree sequence', *The Annals of Applied Probability* **21**(4), 1400–1435.

Chen, K. & Lei, J. (2018), 'Network cross-validation for determining the number of communities in network data', *Journal of the American Statistical Association* **113**(521), 241–251.

Chen, M., Kato, K. & Leng, C. (2021), 'Analysis of networks via the sparse $\beta$-model', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **83**(5), 887–910.

Choi, D. S., Wolfe, P. J. & Airoldi, E. M. (2012), 'Stochastic blockmodels with a growing number of classes', *Biometrika* **99**(2), 273–284.

Chung, F. & Lu, L. (2002), 'The average distances in random graphs with given expected degrees', *Proceedings of the National Academy of Sciences* **99**(25), 15879–15882.

Conte, D., Foggia, P., Sansone, C. & Vento, M. (2004), 'Thirty years of graph matching in pattern recognition', *International journal of pattern recognition and artificial intelligence* **18**(03), 265–298.

Deshpande, Y., Sen, S., Montanari, A. & Mossel, E. (2018), 'Contextual stochastic block models', *Advances in Neural Information Processing Systems* **31**.

Ding, J. & Du, H. (2023), 'Matching recovery threshold for correlated random graphs', *The Annals of Statistics* **51**(4), 1718–1743.

Ding, J., Ma, Z., Wu, Y. & Xu, J. (2021), 'Efficient random graph matching via degree profiles', *Probability Theory and Related Fields* **179**, 29–115.

Erdos, P. & Rényi, A. (1960), 'On the evolution of random graphs', *Publ. Math. Inst. Hungar. Acad. Sci* **5**, 17–61.

Fang, G., Xu, G., Xu, H., Zhu, X. & Guan, Y. (2024), 'Group network hawkes process', *Journal of the American Statistical Association* **119**(547), 2328–2344.

Gallagher, I., Jones, A., Bertiger, A., Priebe, C. E. & Rubin-Delanchy, P. (2024), 'Spectral embedding of weighted graphs', *Journal of the American Statistical Association* **119**(547), 1923–1932.

Gao, C. & Lafferty, J. (2017*a*), 'Testing for global network structure using small subgraph statistics', *arXiv preprint arXiv:1710.00862* .

Gao, C. & Lafferty, J. (2017*b*), 'Testing network structure using relations between small subgraph probabilities', *arXiv preprint arXiv:1704.06742* .

Gao, C., Ma, Z., Zhang, A. Y. & Zhou, H. H. (2018), 'Community detection in degree-corrected block models', *The Annals of Statistics* **46**(5).

Girvan, M. & Newman, M. E. (2002), 'Community structure in social and biological networks', *Proceedings of the national academy of sciences* **99**(12), 7821–7826.

Green, A. & Shalizi, C. R. (2017), 'Bootstrapping exchangeable random graphs', *Electronic Journal of Statistics* .
**URL:** *https://api.semanticscholar.org/CorpusID:88523244*

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. (2012), 'A kernel two-sample test', *The Journal of Machine Learning Research* **13**(1), 723–773.

Hoff, P. D., Raftery, A. E. & Handcock, M. S. (2002), 'Latent space approaches to social network analysis', *Journal of the American Statistical Association* **97**(460), 1090–1098.

Holland, P. W., Laskey, K. B. & Leinhardt, S. (1983), 'Stochastic blockmodels: First steps', *Social networks* **5**(2), 109–137.

Hunter, D. R. & Handcock, M. S. (2006), 'Inference in curved exponential family models for networks', *Journal of computational and graphical statistics* **15**(3), 565–583.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. & Morris, M. (2008), 'ergm: A package to fit, simulate and diagnose exponential-family models for networks', *Journal of statistical software* **24**, 1–29.

Ji, P., Jin, J., Ke, Z. T. & Li, W. (2022), 'Co-citation and co-authorship networks of statisticians', *Journal of Business & Economic Statistics* **40**(2), 469–485.

Jin, J. (2015), 'Fast community detection by score'.

Jin, J., Ke, Z. T. & Luo, S. (2024), 'Mixed membership estimation for social networks', *Journal of Econometrics* **239**(2), 105369.

Jin, J., Ke, Z. T., Tang, J. & Wang, J. (2025), 'Network goodness-of-fit for the block-model family', *Journal of the American Statistical Association* (just-accepted), 1–27.

Karrer, B. & Newman, M. E. (2011), 'Stochastic blockmodels and community structure in networks', *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* **83**(1), 016107.

Karwa, V. & Slavković, A. (2016), 'Inference using noisy degrees: Differentially private $\beta$-model and synthetic graphs', *The Annals of Statistics* **44**(1), 87–112.

Katzav, E., Biham, O. & Hartmann, A. K. (2018), 'Distribution of shortest path lengths in subcritical erdős-rényi networks', *Physical Review E* **98**(1), 012301.

Le, C. M. & Li, T. (2022), 'Linear regression and its inference on noisy network-linked data', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **84**(5), 1851–1885.

Lei, J. (2016), 'A goodness-of-fit test for stochastic block models', *The Annals of Statistics* **44**(1), 401.

Lei, J. & Rinaldo, A. (2015), ‘Consistency of spectral clustering in stochastic block models’, *The Annals of Statistics* pp. 215–237.

Li, T., Levina, E. & Zhu, J. (2020), ‘Network cross-validation by edge sampling’, *Biometrika* **107**(2), 257–276.

Liu, J. S. & Liu, J. S. (2001), *Monte Carlo strategies in scientific computing*, Vol. 10, Springer.

Luo, Y. & Gao, C. (2024), ‘Computational lower bounds for graphon estimation via low-degree polynomials’, *The Annals of Statistics* **52**(5), 2318–2348.

Lyzinski, V., Fishkind, D. E., Fiori, M., Vogelstein, J. T., Priebe, C. E. & Sapiro, G. (2015), ‘Graph matching: Relax at your own risk’, *IEEE transactions on pattern analysis and machine intelligence* **38**(1), 60–73.

Ma, Z., Ma, Z. & Yuan, H. (2020), ‘Universal latent space model fitting for large networks with edge covariates’, *J. Mach. Learn. Res.* **21**, 4:1–4:67.

Mao, C., Wu, Y., Xu, J. & Yu, S. H. (2024), ‘Testing network correlation efficiently via counting trees’, *The Annals of Statistics* **52**(6), 2483–2505.

Meghanathan, N. (2024), ‘Local clustering coefficient-based iterative peeling strategy to extract the core and peripheral layers of a network’, *Applied Network Science* **9**(1), 49.

Miao, R. & Li, T. (2023), ‘Informative core identification in complex networks’, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **85**(1), 108–126.

Mossel, E. & Xu, J. (2020), 'Seeded graph matching via large neighborhood statistics', *Random Structures & Algorithms* **57**(3), 570–611.

Newman, M. (2018), *Networks*, Oxford university press.

Newman, M. E. (2006), 'Modularity and community structure in networks', *Proceedings of the national academy of sciences* **103**(23), 8577–8582.

Pensky, M. (2019), 'Dynamic network models and graphon estimation', *The Annals of Statistics* **47**(4).

Perry, P. O. & Wolfe, P. J. (2013), 'Point process modelling for directed interaction networks', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **75**(5), 821–849.

Qin, T. & Rohe, K. (2013), 'Regularized spectral clustering under the degree-corrected stochastic blockmodel', *Advances in neural information processing systems* **26**.

Rohe, K. (2019), 'A critical threshold for design effects in network sampling'.

Rombach, M. P., Porter, M. A., Fowler, J. H. & Mucha, P. J. (2014), 'Core-periphery structure in networks', *SIAM Journal on Applied mathematics* **74**(1), 167–190.

Rubin-Delanchy, P., Cape, J., Tang, M. & Priebe, C. E. (2022), 'A statistical interpretation of spectral embedding: the generalised random dot product graph', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **84**(4), 1446–1473.

Sengupta, S. & Chen, Y. (2018), 'A block model for node popularity in networks with community structure', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **80**(2), 365–386.

Sewell, D. K. & Chen, Y. (2015), 'Latent space models for dynamic networks', *Journal of the american statistical association* **110**(512), 1646–1657.

Shao, M., Zhang, Y., Wang, Q., Zhang, Y., Luo, J. & Yan, T. (2023), 'L-2 regularized maximum likelihood for $\beta$-model in large and sparse networks', *arXiv preprint arXiv:2110.11856* .

Sischka, B. & Kauermann, G. (2025), 'Stochastic block smooth graphon model', *Journal of Computational and Graphical Statistics* **34**(1), 140–154.

Snijders, T. A. (2002), 'Markov chain monte carlo estimation of exponential random graph models', *Journal of Social Structure* **3**(2), 1–40.

Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V. & Priebe, C. E. (2014), 'A nonparametric two-sample hypothesis testing problem for random dot product graphs', *arXiv preprint arXiv:1409.2344* .

Udell, M. & Townsend, A. (2019), 'Why are big data matrices approximately low rank?', *SIAM Journal on Mathematics of Data Science* **1**(1), 144–160.

Von Luxburg, U. (2007), 'A tutorial on spectral clustering', *Statistics and computing* **17**, 395–416.

Wang, Y. J. & Wong, G. Y. (1987), 'Stochastic blockmodels for directed graphs', *Journal of the American Statistical Association* **82**(397), 8–19.

Wu, Q., Huang, X., Culbreth, A. J., Waltz, J. A., Hong, L. E. & Chen, S. (2022), 'Extracting brain disease-related connectome subgraphs by adaptive dense subgraph discovery', *Biometrics* **78**(4), 1566–1578.

Xu, J. (2018), Rates of convergence of spectral methods for graphon estimation, *in* 'International Conference on Machine Learning', PMLR, pp. 5433–5442.

Xu, M., Jog, V. & Loh, P.-L. (2020), 'Optimal rates for community estimation in the weighted stochastic block model', *The Annals of Statistics* **48**(1), 183–204.

Yan, T., Qin, H. & Wang, H. (2016), 'Asymptotics in undirected random graph models parameterized by the strengths of vertices', *Statistica Sinica* pp. 273–293.

Yang, J., McAuley, J. & Leskovec, J. (2013), Community detection in networks with node attributes, *in* '2013 IEEE 13th international conference on data mining', IEEE, pp. 1151–1156.

Young, S. J. & Scheinerman, E. R. (2007), Random dot product graph models for social networks, *in* 'International Workshop on Algorithms and Models for the Web-Graph', Springer, pp. 138–149.

Zhang, L. & Amini, A. A. (2023), 'Adjusted chi-square test for degree-corrected block models', *The Annals of Statistics* **51**(6), 2366–2385.

Zhang, Y. (2018), 'Unseeded low-rank graph matching by transform-based unsupervised point registration', *arXiv preprint arXiv:1807.04680* .

Zhang, Y., Levina, E. & Zhu, J. (2016), 'Community detection in networks with node features', *Electronic Journal of Statistics* **10**(2), 3153.

Zhang, Y., Levina, E. & Zhu, J. (2017), ‘Estimating network edge probabilities by neighbourhood smoothing’, *Biometrika* **104**(4), 771–783.

Zhang, Y., Levina, E. & Zhu, J. (2020), ‘Detecting overlapping communities in networks using spectral methods’, *SIAM Journal on Mathematics of Data Science* **2**(2), 265–283.

Zhang, Y. & Xia, D. (2022), ‘Edgeworth expansions for network moments’, *The Annals of Statistics* **50**(2), 726–753.

Zhao, Y. (2023), *Graph Theory and Additive Combinatorics: Exploring Structure and Randomness*, Cambridge University Press.

Zhao, Y., Levina, E. & Zhu, J. (2012), ‘Consistency of community detection in networks under degree-corrected stochastic block models’, *The Annals of Statistics* **40**(4), 2266–2292.