Network Method of Moments

Yuan Zhang

Joint work with Dong Xia (HKUST)

MATLAB code: `github.com/yzhanghf`

Introduction and motivation

Parametric network analysis:

- Parametric model $\rightarrow$ point estimation $\stackrel{?}{\rightarrow}$ inference

Challenges:

- Inference may be difficult to derive
- Method is model-specific

Non-parametric methods:

- **Less ambitious goal:** not learning every detail of network model, just numerical features
- **More flexibility:** model-free/applicable to many models; weak model assumptions
- **Computation efficiency:** easier/faster than fitting some models

Introduction and motivation

Network method of moments

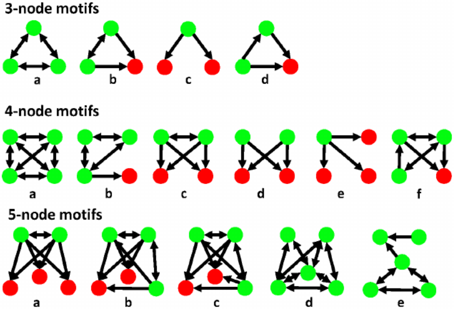
- Network moments extension of the classical *moments* in i.i.d. setting
- Fast and straightforward computation
- Model-free
- Universal and principled inference

Question: How to define *network moments*?

Introduction and motivation

Network moments *(Bickel et al. 2011)*

- Network moments are indexed by motifs
- Example: edge: $\text{mean}_{ij}(A_{ij})$
- Example: triangles: $\binom{n}{3}^{-1}\sum_{1\le i<j<k\le n} A_{ij}A_{jk}A_{ki}$
- More examples (directed networks, *Jayavelu et al. (2014)*):



Introduction and motivation

Descriptive power of network moments

- Network comparison:
  Different network moments $\Rightarrow$ different network models
- Knowing all moments determines exchangeable network model?
  "Nearly yes" ("Yes" for practitioners) *(Borgs et al, 2010)*
- May service some parametric models: ERGM:

$$\text{likelihood of } A \propto \exp\left\{\sum_k \text{Motif}_k(A)\right\}$$

- Related topics:
  - One sample inference *(This paper), (Shao, Xia & Z., 2022+)*
  - Two sample inference (network comparison) *(Ghoshdastidar et al, 2017), (Shao et al, 2022+)*

Introduction and motivation

Major challenge: distribution of network moments?

To better illustrate, we first describe the **base model**

Problem formulation

Data:

- Adjacency matrix: $A \in \mathbb{R}^{n \times n}$

$$A_{ij} = A_{ji} = \begin{cases} 1 & i \leftrightarrow j \\ 0 & \text{otherwise} \end{cases}$$

- Symmetric edge probability matrix: $W \in \mathbb{R}^{n \times n}$:

$$A_{ij} | W \overset{\text{independent}}{\sim} \text{Bernoulli}(W_{ij})$$

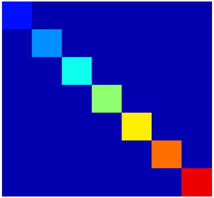
Problem formulation

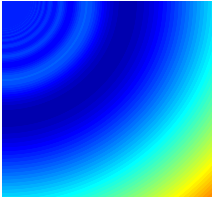
Exchangeable networks (Aldous-Hoover representation):

- Latent graphon function $f : [0,1]^2 \to [0,1]$
- Latent node position $X_i \sim \text{Uniform}[0,1]$:

$$W_{ij} = \rho_n \cdot f(X_i, X_j)$$

  ρ_n: sparsity multiplier
- f encodes structures; X_i encodes node's role; both inestimable





Problem formulation

Formulation of network motif:

- **Motif** R: r nodes and s edges
- Corresponding sample moment is the count statistic

$$\widehat{U}_n := \binom{n}{r}^{-1} \sum_{1 \le i_1 < \ldots < i_r \le n} h(A_{i_1,\ldots,i_r}),$$

where

$$h(A_{i_1,\ldots,i_r}) := \mathbb{1}_{[A_{i_1,\ldots,i_r} \sqsupseteq R]}$$

Problem formulation

Question: How to characterize $\widehat{U}_n$?

- Design a proper variance estimator $\widehat{S}_n^2$, and studentize:

$$\widehat{T}_n := \frac{\widehat{U}_n - \mathbb{E}[\widehat{U}_n]}{\widehat{S}_n}$$

What's next?

1. How to design $\widehat{S}_n = ?$
2. Distribution of $\widehat{T}_n$?

Before introducing our method, a quick literature review...

Distribution approximation

Existing literature

- Asymptotic normality *(Bickel et al, 2011)*
- Network bootstraps:
  - Node sub-sampling: *(Bhattacharyya & Bickel, 2015)*
  - Node re-sampling: *(Green & Shalizi, 2017)*
  - Low-rank approximation then bootstrap estimated low-rank structure: *(Levin & Levina, 2019)*
- Limitations:
  - No finite sample accuracy guarantee, only consistency

    $$\widehat{T}_n \to N(0,1)$$

    in $\xrightarrow{d}$, $\xrightarrow{p}$, etc
  - Slow computation (bootstrap methods)

Our method

Our paper:

- Analytical, higher-order accurate approximation to $F_{\widehat{T}_n}$
- Fast computation (eliminates bootstrap)
- Model-free & Versatility: applicable to non-smooth graphons *(Choi, 2017)*
- New theoretical insights
- Rate-optimal inference power + higher-order accurate risk control

Key intuition

Example: $R =$ Edge:

$$\widehat{U}_n = \binom{n}{2}^{-1} \sum_{1\le i<j\le n} W_{ij} + \sum_{1\le i<j\le n} (A_{ij} - W_{ij})$$
$$=: \binom{n}{2}^{-1} \sum_{1\le i<j\le n} f(X_i, X_j) + \binom{n}{2}^{-1} \sum_{1\le i<j\le n} \eta_{ij}$$

Key intuition

Decomposition of randomness

$$\widehat{U}_n = U_n + (\widehat{U}_n - U_n)$$

Here:

- $U_n = U_n(X_1, \ldots, X_n)$: variations of W, due to nodes' roles
- $\widehat{U}_n - U_n$: observational errors in $A|W$
- Under mild conditions, $\mathrm{Var}(U_n) \gg \mathrm{Var}(\widehat{U}_n - U_n)$, so we use U_n to design variance estimator

Key intuition

Example: $R =$ Triangle:

$$
\begin{aligned}
\widehat{U}_n &= \binom{n}{3}^{-1} \sum_{1\le i<j<k\le n} A_{ij}A_{jk}A_{ki} \\
&= \binom{n}{3}^{-1} \sum_{1\le i<j<k\le n} W_{ij}W_{jk}W_{ki} \\
&\quad + \binom{n}{3}^{-1} \sum_{1\le i<j<k\le n} \left\{ \eta_{ij}W_{jk}W_{ki} + W_{ij}\eta_{jk}W_{ki} + W_{ij}W_{jk}\eta_{ki} \right\} \\
&\quad + \text{(Product } \eta \text{ terms, remainder)}
\end{aligned}
$$

Noiseless version of the problem

- $U_n := \mathbb{E}[\widehat{U}_n | W]$ is a noiseless U-statistic and admits a **Hoeffding's ANOVA decomposition**

$$U_n - \mathbb{E}[U_n] = \frac{r}{n}\sum_{i=1}^{n} g_1(X_i) + \frac{r(r-1)}{n(n-1)} \sum_{1\le i<j\le n} g_2(X_i, X_j) + \cdots$$

where the uncorrelated g_k's are:

$$\begin{aligned} g_1(X_1) &:= \mathbb{E}[h(X_1,\ldots,X_r)|X_1] - \mathbb{E}[h] \\ g_2(X_1,X_2) &:= \mathbb{E}[h(X_1,\ldots,X_r)|X_1,X_2] - g_1(X_1) - g_1(X_2) - \mathbb{E}[h] \\ &\cdots\cdots \end{aligned}$$

Design of variance estimator

Design

$$\widehat{S}_n^2 = \frac{r^2}{n^2} \cdot \sum_{i=1}^{n} \Bigg\{ \underbrace{\frac{1}{\binom{n-1}{r-1}} \sum_{\substack{1 \le i_1 < \ldots < i_{r-1} \le n \\ i_1, \ldots, i_{r-1} \neq i}} h(A_{i, i_1, \ldots, i_{r-1}}) - \widehat{U}_n}_{\text{Estimates } g_1(X_1)} \Bigg\}^2$$

- **Theorem 3.3** ***Z. & Xia, (2022)*** $\widehat{S}_n^2$ is equivalent to network jackknife *(Maesono, 1997)*, but computes faster and more convenient for analysis

Next: Distribution of $\widehat{T}_n$?

Distribution of $\widehat{T}_n$?

If we simplify it (we can't, but just for this moment)...

$$U_n - \mathbb{E}[U_n] = \frac{r}{n}\sum_{i=1}^{n} g_1(X_i) + \cancel{\frac{r(r-1)}{n(n-1)}\sum_{1\le i<j\le n} g_2(X_i, X_j) + \cdots}$$

- Edgeworth expansion for i.i.d. data (need Cramer's condition!):

$$\mathrm{CDF}\left(\frac{\bar{g}_1(X)-\mu}{\sigma_{g_1(X_1)}/\sqrt{n}}; u\right) = \Phi(u) - \varphi(u)\frac{\mathbb{E}[g_1(X_1)^3](u^2-1)}{6\sqrt{n}\cdot\sigma^3_{g_1(X_1)}} + O(n^{-1})$$

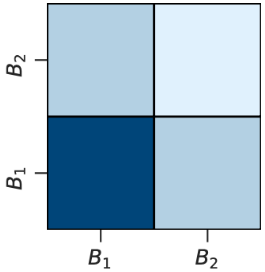
Edgeworth expansion for noiseless U-statistics

Cramer's condition:

- For noiseless U-statistic, $g_1(X_1)$ satisfies:

$$\limsup_{t\to\infty} \left| \mathbb{E}[e^{\mathbb{i} t g_1(X_1)}] \right| < 1$$

- Cramer's condition $\approx g_1(X_1)$ is **continuous**
- **Violation:** block model (think $R =$ Edge, then $n(g_1(X_i) + \mu) =$ expected degree of node i)



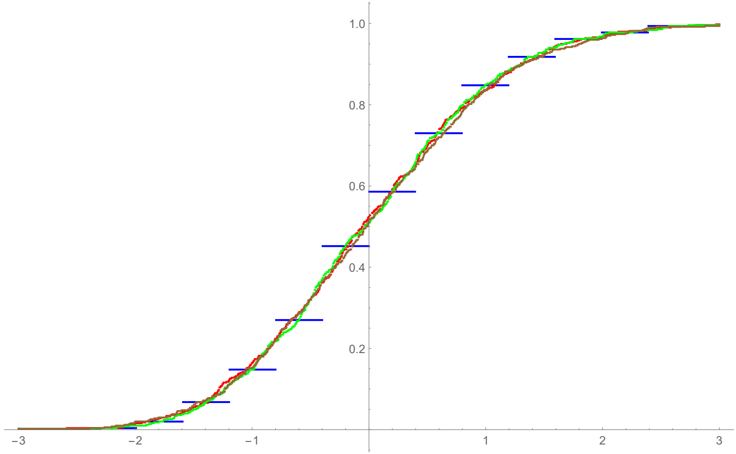
Key insight

$$\widehat{U}_n = \underbrace{U_n}_{\text{Randomness from } X_1,\ldots,X_n} + \underbrace{(\widehat{U}_n - U_n)}_{\text{Randomness from } A|W}$$

Key findings: $\widehat{U}_n - U_n$ provides a self-smoothing effect

- Observational error: behaves like a Gaussian smoother
- Sparsity: amplifies the smoothing effect to eliminate potential discontinuity in U_n

Observational noise smooths CDF



IID case illustration: data $X_1, \ldots, X_{10}$ are Bernoulli. Blue: original $\bar{X}$, Red: plus uniform noise, bandwidth $n^{-1/2}$, Green: mixed with normal noise, bandwidth $(\log n/n)^{1/2}$, Brown, normal noise bandwidth $n^{-1/2}$

Expansion expansion

Network Edgeworth expansion:

$$G_n(x) := \Phi(x) - \frac{\varphi(x)}{\sqrt{n} \cdot \xi_1^3} \cdot \left\{ \frac{2x^2+1}{6} \cdot \mathbb{E}[g_1^3(X_1)] + \frac{r-1}{2} \cdot (x^2+1)\mathbb{E}[g_1(X_1)g_1(X_2)g_2(X_1,X_2)] \right\}$$

where $\xi_1^2 := \mathbb{E}[g_1^2(X_1)]$, $\Phi(x)$ and φ: $N(0,1)$ CDF and PDF

- $\widehat{G}_n(x) :=$ empirical version

Main theorems

- **Theorems 3.1 & 3.2** ***(Z. & Xia, (2022))*** Assume
  1. $\rho_n^{-2s} \cdot \mathrm{Var}(g_1(X_1)) \geq \mathrm{Const} > 0$
  2. **Dense regime:** Either R is acyclic and $\rho_n = \omega(n^{-1/2})$, or R is cyclic and $\rho_n = \omega(n^{-1/r})$
  3. Either $\rho_n = O((\log n)^{-1})$ or Cramer's condition holds

  We have

$$\sup_{u \in \mathbb{R}} \left| F_{\widehat{T}_n}(u) - G_n(u) \right| = O(\mathscr{M}(\rho_n, n; R)) \ll n^{-1/2}$$

  where

$$\mathscr{M}(\rho_n, n; R) := \begin{cases} (\rho_n \cdot n)^{-1} \log^{1/2} n + n^{-1} \log^{3/2} n & \text{for acyclic } R \\ \rho_n^{-r/2} \cdot n^{-1} \log^{1/2} n + n^{-1} \log^{3/2} n & \text{for cyclic } R \end{cases}$$

  Result also holds: $G_n(u)$ replaced by $\widehat{G}_n(u)$ (O replaced by O_p with n^{-1} tail probability)

Main theorems (continued)

- **Theorem 3.4** ***(Z. & Xia, (2022))*** Assume
  1. (same as before)
  2. **Sparse regime:** Either R is acyclic and $n^{-1} \prec \rho_n \preceq n^{-1/2}$, or R is cyclic and $n^{-2/r} \prec \rho_n \preceq n^{-1/r}$
  3. (same as before)

  We have

$$\sup_{u\in\mathbb{R}} \left| F_{\widehat{T}_n}(u) - G_n(u) \right| \asymp \sup_{u\in\mathbb{R}} \left| F_{\widehat{T}_n}(u) - \Phi(u) \right| = O(\mathscr{M}(\rho_n, n; R)) \bigwedge o_p(1) \gg n^{-1/2}$$

- **Sparse regime:** Berry-Esseen bound dominates $n^{-1/2}$; using $N(0,1)$ approximation is good enough

Applications

One-sample inference:

- **Hypothesis testing:**
  - Type I error = $\alpha + O(\mathscr{M}(\rho_n, n; R))$:

$$\text{Estimated p-value} = 2\min\left\{\widehat{G}_n(t^{(\text{obs})}), 1 - \widehat{G}_n(t^{(\text{obs})})\right\}$$

  - Optimal separation condition under H_a.
- Length-optimal CI; nominal level = $1 - \alpha + O(\mathscr{M}(\rho_n, n; R))$:

$$\left(\widehat{U}_n \pm \widehat{q}_{\widehat{T}_{n;\alpha/2}} \cdot \widehat{S}_n\right)$$

  where (Cornish-Fisher expansion):

$$\widehat{q}_{\widehat{T}_{n;\alpha}} := z_\alpha - \frac{1}{\sqrt{n}\cdot \widehat{\xi}_1^3}\left\{\frac{2z_\alpha^2+1}{6}\cdot \widehat{E}[g_1^3(X_1)] + \frac{r-1}{2}(z_\alpha^2+1)\widehat{\mathbb{E}}[g_1(X_1)g_1(X_2)g_2(X_1,X_2)]\right\}$$

- Computes much faster than bootstrap iteration *(Beran, 1987, 1988)*

Applications

Understanding the accuracy of network bootstraps:

- Old theory *(Bhattacharya & Bickel, 2015), (Green & Shalizi, 2017), (Levin & Levina, 2019)*:
  $o(1)$ (only consistency, no finite-sample error rate)
- Our theory implies: $o(n^{-1/2})$ (when $\rho_n \gg n^{-1/2}$)

Our method vs network bootstraps:

- Our error bound $\ll$ network bootstraps
- Our computation time $\ll$ network bootstraps

Simulations

Set up:

- Graphons: SBM, smooth graphon, "non-smooth graphon"
- Motifs: edge, triangle, V-shape

Benchmarks:

- $N(0,1)$ approximation
- Node re-sampling *(Green & Shalizi, 2017), Alg. 1*
- Node sub-sampling *(Bhattacharya & Bickel, 2015), Alg. 1*

Performance measured by:

- $\|\widehat{F}_{\widehat{T}_n} - F_{\widehat{T}_n}\|_\infty$ on $(-2, 2)$
- 500 experiments, within each iteration: 10^5 Monte Carlo repeats to evaluate true CDF

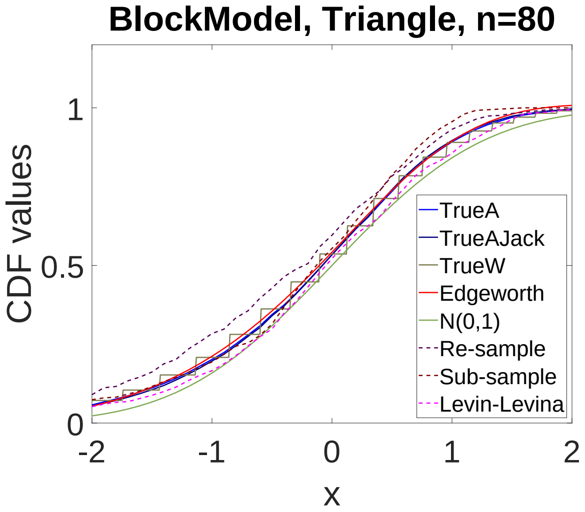


Figure: CDF curves, $n = 80$, SBM, triangle, bootstrap sample: 500. `TrueA` is $F_{\widehat{T}_n}$; `TrueAJack` is $F_{\widehat{T}_{n;\text{jackknife}}}$; `TrueW` is F_{T_n}; `Edgeworth` is our EEE; `Re-sample` is *Green & Shalizi, (2017)*; `Sub-sample` is *Bhattacharyya & Bickel, (2015)*; `Levin-Levina` is *Levin & Levina, (2019)*.

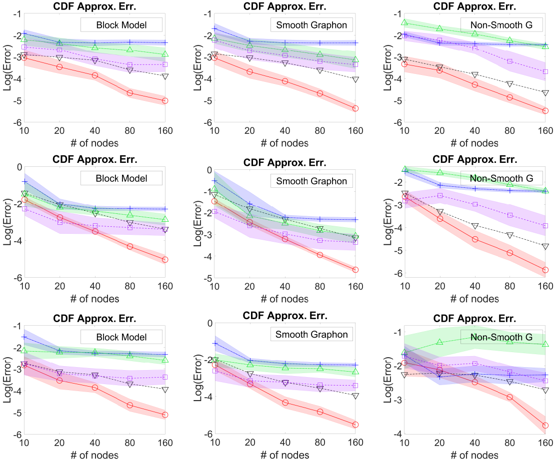


Figure: **Motifs:** row 1: `Edge`; row 2: `Triangle`; row 3: `Vshape`. CDF approximation errors. Our method, $N(0,1)$, Green & Shalizi (2017), Bhattacharrya & Bickel (2015), Levina & Levina (2019)

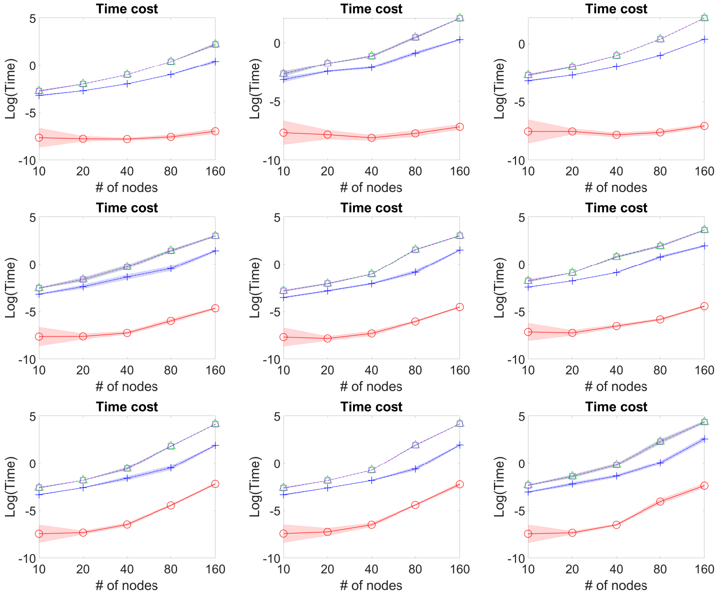


Figure: Log-time, Our method, Green & Shalizi (2017), Bhattacharyya & Bickel (2015), Levina & Levina (2019)

Table: Performance measures of 95% confidence intervals
$n = 80$, $\rho_n \asymp 1$, graphon: block model

Method	Edge	Triangle	V-shape	Three star
Our method	Coverage $= 0.957(0.202)$	$0.953(0.211)$	$0.956(0.205)$	$0.952(0.213)$
	Length $= 0.097(0.010)$	$0.040(0.008)$	$0.200(0.033)$	$0.145(0.033)$
	LogTime $= -8.448(0.110)$	$-7.214(0.083)$	$-7.165(0.082)$	$-7.180(0.353)$
Norm. Approx.	$0.950(0.218)$	$0.934(0.248)$	$0.942(0.235)$	$0.932(0.251)$
	$0.097(0.010)$	$0.040(0.008)$	$0.200(0.033)$	$0.145(0.033)$
	No time cost*	No time cost	No time cost	No time cost
Bhattacharrya & Bickel (2015)	$0.842(0.365)$	$0.870(0.337)$	$0.852(0.355)$	$0.852(0.355)$
	$0.068(0.009)$	$0.031(0.007)$	$0.147(0.026)$	$0.113(0.025)$
	$-2.591(0.008)$	$-2.160(0.026)$	$-2.127(0.024)$	$-0.992(0.006)$
Green & Shalizi (2017)	$0.938(0.241)$	$0.944(0.230)$	$0.934(0.249)$	$0.938(0.241)$
	$0.096(0.013)$	$0.044(0.010)$	$0.204(0.038)$	$0.150(0.037)$
	$-1.198(0.007)$	$0.499(0.032)$	$0.142(0.035)$	$0.383(0.010)$
Levina & Levina (2019)	$0.942(0.234)$	$0.942(0.234)$	$0.942(0.234)$	$0.942(0.234)$
	$0.099(0.013)$	$0.043(0.010)$	$0.209(0.039)$	$0.155(0.038)$
	$-1.188(0.004)$	$0.507(0.028)$	$0.142(0.027)$	$0.489(0.004)$

* $N(0,1)$ costs the same time as ours in evaluating the studentization

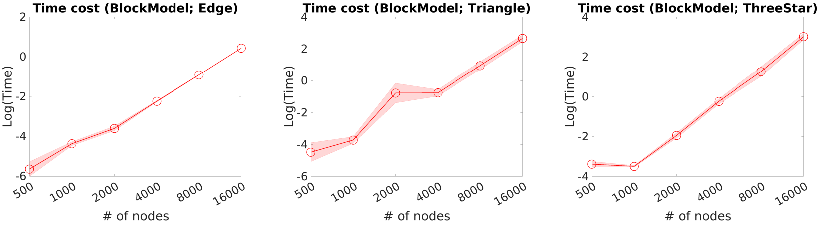


Figure: Scalability of our method on large networks.

Thank you!

Edgeworth expansions for network moments
Z. and Xia, *Annals of Statistics (2022)*

Thank you! Any questions?