STA6800 - Statistical Analysis of Network

Parameter Estimation of ERGM

Ick Hoon Jin

Yonsei University, Department of Statistics and Data Science

Current Methods for ERGM

- Approximation-based Algorithm: Maximize the likelihood function with MCMC samples.
  - Maximum Pseudo-likelihood Estimation (MPLE).
  - Markov Chain Monte Carlo Maximum Likelihood Estimation (MCMCMLE).
  - Markov Chain Monte Carlo Stochastic Approximation (MCMCSA).
  - Varying Truncation Stochastic Approximation MCMC Method with Trajectory Averaging (VTSAMCMC).

Current Methods for ERGM

- Auxiliary Variable Markov Chain Monte Carlo (MCMC) Algorithm: Introduce auxiliary variables to cancel the normalizing constant ratio $\kappa(\theta)/\kappa(\theta')$ or to approximate the normalizing constant. Used for the Bayesian Inference.
  - The Exchange Algorithm.
  - Auxiliary Variable Metropolis-Hasting Algorithm (AVMH).
  - Adaptive Exchange Monte Carlo Algorithm (AEXMC).

Maximum Pseudo Likelihood Estimation

- Approximate the likelihood function by a product of series of conditional likelihood functions by ignoring dependence within components of X.
- The conditional and pseudo likelihood of the ERGMs is

$$\text{logit}\left\{P_\theta(X_{ij}=1|X_{ij}^c=x_{ij}^c)\right\}=\theta^t g(x_{ij}^c),$$
$$\log PL(\theta,x)=\sum_{ij}\theta^t g(x_{ij}^c)x_{ij}-\sum_{ij}\log\left\{1+\theta^t g(x_{ij}^c)\right\}.$$

- This method is the simplest one.
- Since this method totally ignores certain dependent structures within data, the performance is not generally satisfactory.

MCMCMLE

- Approximate $\kappa(\theta)$ using Monte Carlo samples generated from a distribution $f(x|\theta^{(0)})$, where $\theta^{(0)}$ is an initial estimate of θ.
- Draw $x_1, \cdots, x_m$ denote random samples drawn from $f(x_{\text{obs}}|\theta^{(0)})$ via MCMC simulations, the log-ratio of likelihood can be approximated by

$$l(\theta) - l(\theta^{(0)}) = (\theta - \theta^{(0)})^t g(x_{\text{obs}}) - \log \left[\frac{1}{m} \sum_{i=1}^{m} \exp \left\{ (\theta - \theta^{(0)})^t g(x_i) \right\} \right],$$

  as $m \to \infty$. Maximization of $l(\theta) - l(\theta^{(0)})$ via θ will approximate $\hat{\theta}$ (MLE).
- The performance of MCMLE depends on the choice of $\theta^{(0)}$. If $\theta^{(0)}$ does not lie in the attraction region of true MLE, the method may converge to a suboptimal solution or fail to converge.

MCMC Stochastic Approximation

- The theory of exponential family implies that maximizing the ERGMs is equivalent to solving the system of equations

$$E_{\hat{\theta}}\{g(x)\} = g(x_{obs})$$

that is satisfied if and only if $\hat{\theta}$ is the maximum likelihood estimator of θ and it requires

1. Independent network generation
2. Parameter estimation update with stochastic approximation

- This method is inefficient in generating independent network samples. The number of iteration steps for generating each sample x_{k+1} is in the order of $100n^2$ where n is the number of nodes.

Exchange Algorithm

Augment the distribution $f(x|\theta)$ by an auxiliary variable such that the normalizing constant ratio $\kappa(\theta)/\kappa(\theta')$ can be canceled in simulations.

1. Propose a candidate point θ' from a proposal distribution denoted by $q(\theta'|\theta, x)$.
2. Generate an auxiliary variable $y \sim f(y|\theta')$ using a perfect sampler.
3. Accept θ' with probability $\min\{1, r(\theta, \theta'|x)\}$, where

$$r(\theta, \theta'|x) = \frac{\pi(\theta')}{\pi(\theta)} \cdot \frac{f(x|\theta')}{f(x|\theta)} \cdot \frac{f(y|\theta)}{f(y|\theta')} \cdot \frac{q(\theta|\theta', x)}{q(\theta'|\theta, x)}.$$

Exchange Algorithm

- Although the exchange algorithms work well for some discrete models, such as the Ising and autologistic models, they cannot be applied to many other models for which perfect sampling is not available.
- We can generate auxiliary variable y via MCMC samples, but there exists theoretical flaws on convergence issues. If the mixing of MCMC sample is very slow, which is a general feature of ERGMs, θ' tends to accept with a high probability.

Monte Carlo MH Algorithm

- At each iteration, the MCMH algorithms replaces the unknown normalizing constant ratio $\kappa(\theta)/\kappa(\theta')$ by a Monte Carlo estimates and importance sampling approach.
- A Monte Carlo version of the Metropolis-Hastings algorithm.
- Unlike the exchange algorithms, the MCMH algorithm avoids the requirement for perfect sampling, and thus can be applied to many statistical models for which perfect sampling is not possible.
- This algorithm also suffers from a theoretical flaw on convergence, i.e. the importance sampling estimator might fail to converge to the true ratio $\kappa(\theta)/\kappa(\theta')$ with a finite number of samples.

Goodness-of-fit Plots for ERGMs

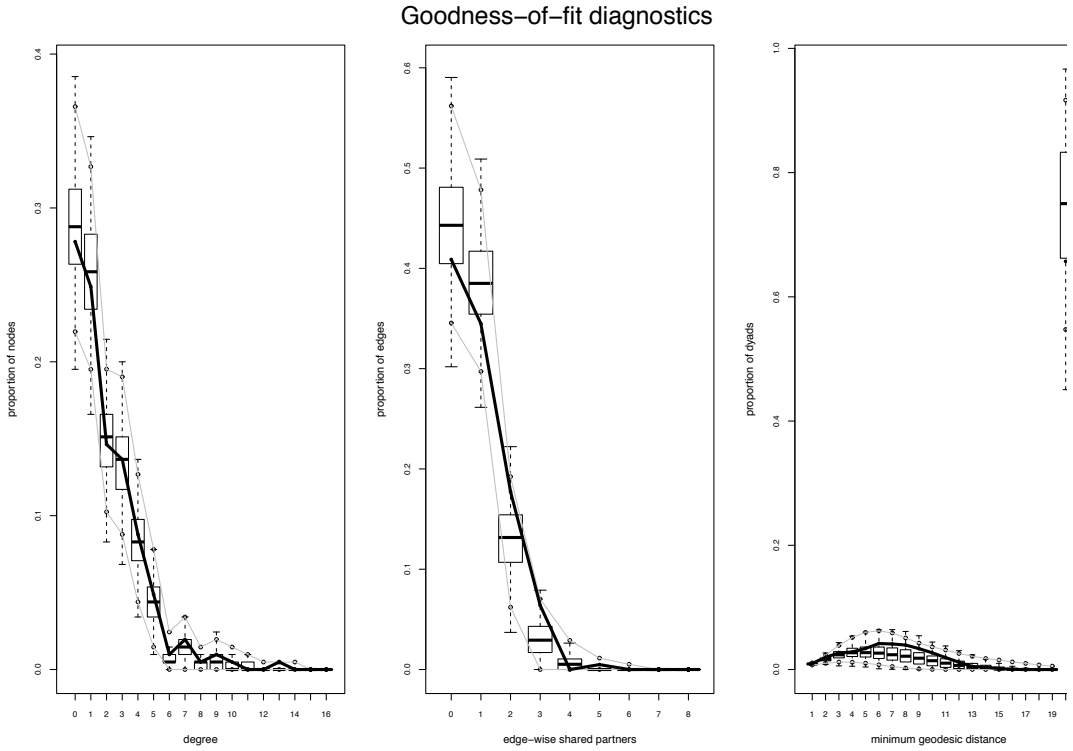

Figure: GOF plots for the high school student friendship network

Motivation

- The likelihood function of the ERGM is

$$f(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{\kappa(\boldsymbol{\theta})} \exp\left(\boldsymbol{\theta}^T S(\mathbf{y})\right)$$

  where

$$\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d)^T, \quad S(\mathbf{y}) = (S_1(\mathbf{y}), \cdots, S_d(\mathbf{y}))^T$$

- Due to 2 issues, intractability of $\kappa(\boldsymbol{\theta})$ and model degeneracy, it is difficult to estimate $\boldsymbol{\theta}$ well.
- These problem can be solved by using the varying truncation stochastic approximation MCMC.

Intractability of $\kappa(\theta)$

- Normalizing constant

$$\kappa(\boldsymbol{\theta}) = \sum_{\text{all possible } \mathbf{y}} \exp\left(\boldsymbol{\theta}^T S(\mathbf{y})\right)$$

- MPLE
  - Assume dyadic independence
  - Highly dependent on the observed network
- MCMLE
  - Depend on the choice of $\boldsymbol{\theta}^{(0)}$
  - Converge to a local optimal solution or fail to converge due to model degeneracy.

Stochastic Approximation

- Solve a system of equation of the form

$$h(\theta) = 0.$$

- The classical SA algorithm is of the form

$$\begin{aligned} \theta_{k+1} &= \theta_k + a_k Y_k \\ &= \theta_k + a_k \{h(\theta_k) + \omega_k\}, \quad k \geq 0. \end{aligned}$$

$Y = h(\theta) + \omega$ is noisy estimate. ω is mean-zero noise.

- Under appropriate conditions (on a_k, h, ω_k), the algorithm indeed can be shown to converge to a solution.

MCMC Stochastic Approximation

Finding MLE of θ is equivalent to solving $E_\theta(\boldsymbol{S}(\boldsymbol{Y})) = \boldsymbol{S}(\boldsymbol{y}_{obs})$ in exponential families.

- (Step 1) Independence network generation: Sample $\boldsymbol{y}^{(k+1)}$ from the $f(\boldsymbol{y}|\boldsymbol{\theta}^{(k)})$
  - Start with a random graph in which each arc variable Y_{ij} is determined independently, with a probability 0.5 for the values 0 and 1
  - Update the random graph using the Gibbs sampler or the MH algorithm

MCMC Stochastic Approximation

- (Step 2) Parameter estimate update with SA

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - a_k D^{-1}(U(\boldsymbol{y}_{k+1}, \overline{\boldsymbol{y}}_{k+1}) - \boldsymbol{S}(\boldsymbol{y}_{obs}))$$

where

$$U(\boldsymbol{y}_{k+1}, \overline{\boldsymbol{y}}_{k+1}) = P(\overline{\boldsymbol{y}}_{k+1}|\boldsymbol{y}_{k+1})\boldsymbol{S}(\overline{\boldsymbol{y}}_{k+1}) + (1 - P(\overline{\boldsymbol{y}}_{k+1}|\boldsymbol{y}_{k+1}))\boldsymbol{S}(\boldsymbol{y}_{k+1})$$
$$\overline{\boldsymbol{y}}_{k+1} = 1 - \boldsymbol{y}_{k+1}$$

- Inefficiency in generating independent network samples: Order of $100n^2$

Model Degeneracy

- For some configurations of θ, the model lumps all or most of its probability mass on just one or a few possible graphs.
  - Complete(fully connected) or empty(entirely unconnected) networks
- A solution to avoid this problem is to specify a model whose parameter space contains no or less degeneracy regions
  - But, this is often more difficult than usual.

Setup

(C_1) Set

$$a_k = C_a \left(\frac{k_0}{\max(k_0, k)} \right)^{\eta}, \quad b_k = C_b \left(\frac{k_0}{\max(k_0, k)} \right)^{\xi},$$

for some constants $k_0 > 1, \eta \in (1/2, 1), \xi \in (1/2, \eta), C_a > 0, C_b > 0$.

(C_2)

$$\bigcup_{s \geq 0} \mathcal{K}_s = \Theta, \quad \mathcal{K}_s \subset \text{int}(\mathcal{K}_{s+1})$$

Setup

- $\mathcal{X}$: a space of social networks.
- $\mathcal{T} : \mathcal{X} \times \Theta \to \mathcal{X}_0 \times \mathcal{K}_0$ (reinitialization mechanism)
- σ_k: the number of reinitialization performed until iteration k. ($\sigma_0 = 0$)

Varying truncation SAMCMC algorithm for ERGMs

1. Draw an auxiliary network y_{k+1} from the distribution $f(y|\theta^{(k)})$ using the Gibbs sampler iterating for m sweeps

2. Set $\theta^{(k+\frac{1}{2})} = \theta^{(k)} + a_k \left(S(y_{k+1}) - S(y_{obs})\right)$

3. If $||\theta^{(k+\frac{1}{2})} - \theta^{(k)}|| \leq b_k$ and $\theta^{(k+\frac{1}{2})} \in \mathcal{K}_{\sigma_k}$, then set $\sigma_{k+1} = \sigma_k$ and $(y_{k+1}, \theta^{(k+1)}) = (y_{k+1}, \theta^{(k+\frac{1}{2})})$;

   Otherwise,set $\sigma_{k+1} = \sigma_k + 1$ and $(y_{k+1}, \theta^{(k+1)}) = \mathcal{T}(y_k, \theta^{(k)})$.

Trajectory averaging estimator

θ can be estimated by the trajectory averaging estimator

$$\bar{\theta}_n = \frac{1}{n}\sum_{k=1}^{n}\theta^{(k)}.$$

In practice, to reduce the variation of the estimate, we often use

$$\bar{\theta}(n_0, n) = \frac{1}{n-n_0}\sum_{k=n_0+1}^{n}\theta^{(k)},$$

to estimate θ, where n_0 denotes the number of burn-in iterations.

Free parameters

- Free parameters: $\{a_k\}, \{b_k\}, \{\mathcal{K}_s, s \geq 0\}, m$
- $k_0 = 100, \eta = 0.65, \xi = (0.5 + \eta)/2$
- C_a ,C_b : adjusted for different examples
- Choose $\mathcal{K}_0$ to be around MPLE.
- In this article, we set
  $\mathcal{K}_{s,1} = [-4(s+1), 4(s+1)], \mathcal{K}_{s,2} = \cdots = \mathcal{K}_{s,d} = [-2(s+1), 2(s+1)]$
- m=1

Numerical Examples

- Data: Florentine Business Network(around 1430, 16 powerful families in Florence), AddHealth network(1994-1995 National Study of Adolescent Health)

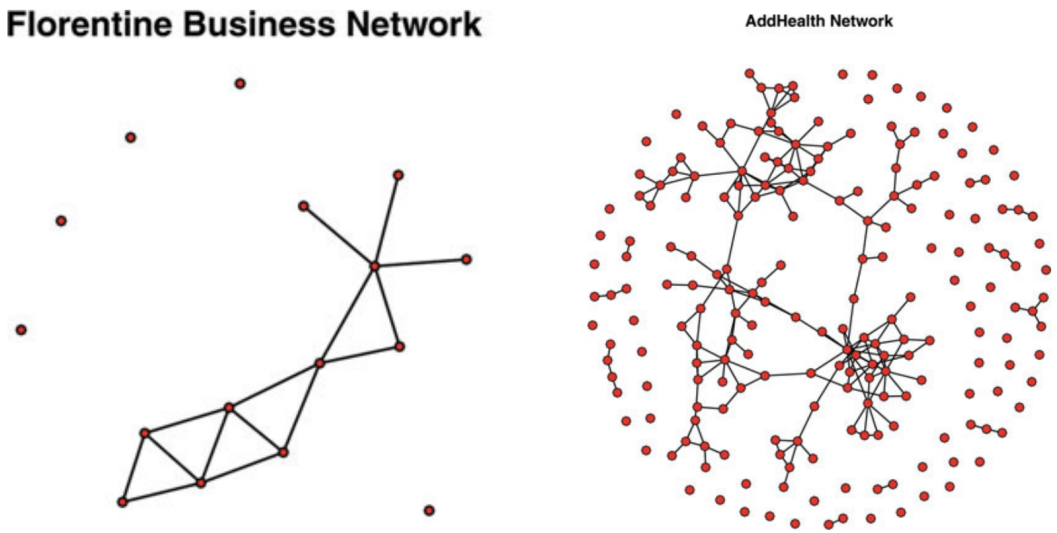

Figure: Visualizations of example networks

Numerical Examples

- Methods: MCMLE, SAA (Stochastic Approximation Algorithm), Varying truncation SAMCMC
- SAMCMC: independent 5 runs(each of 200,000 iterations) (m=1, $C_a = 0.01$, $C_b = 1000$, η, ξ, κ_s are defalut)

Example 1: Florentine Business Network

- Node: 16 powerful families in Florence around 1430s
- Edge: business ties, such as loans, credits, and joint partership
- Model:

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{\kappa(\boldsymbol{\theta})} \exp\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 S_2(\boldsymbol{y})\}$$

where $S_1(y)$ is the edge count and $S_2(y)$ is the 2-star count

Table 1. Parameter estimates for the Florentine business network, which are calculated by averaging over five independent runs with the standard deviations (Monte Carlo error) given in the parentheses

	Edge count (θ_1)	k_2-star (θ_2)	CPU time
SAMCMC	-2.733 (4.2×10^{-4})	0.198 (9.0×10^{-5})	3.2 sec
MCMLE	-3.191 (2.6×10^{-1})	0.412 (1.2×10^{-1})	78.1 sec
SAA	-2.842 (7.7×10^{-3})	0.283 (3.9×10^{-2})	370.4 sec

Example 1: Results

- Goodness-of-fit diagnostics & Model degeneracy region

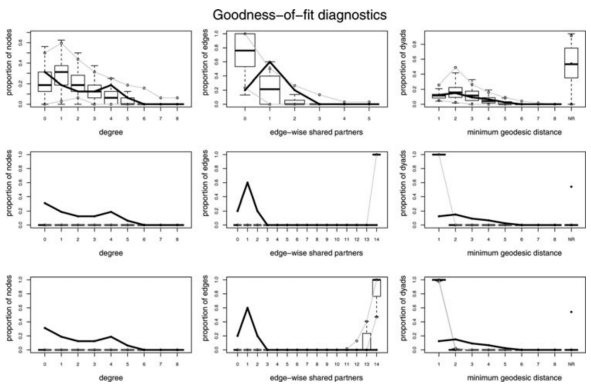

Figure 2. GOF plots for the Florentine business network: row 1: SAMCMC, row 2: MCMLE, and row 3: SAA. The solid line shows the observed network statistics, and the boxplots represent the distributions of simulated network statistics.

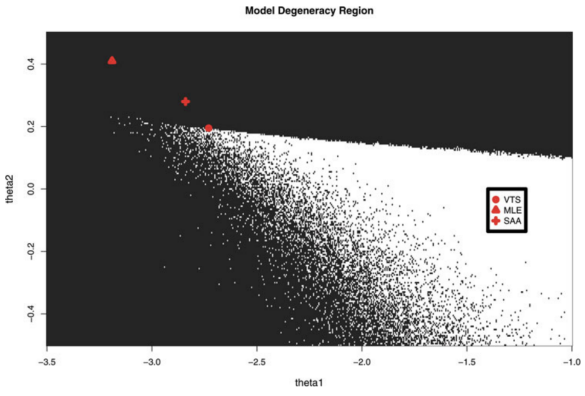

Figure 3. Visualization of the degeneracy (black) and nondegeneracy (white) regions of model (20). The circle (o), plus (+), and triangle (Δ) indicate the estimates produced by varying truncation SAMCMC, SAA, and MCMLE, respectively. The online version of this figure is in color.

- Degeneracy : With running Gibbs sampler 5 times consisted of 10000 iterations at each grid point of 250 x 100 lattice on [-3.5,-1.0]x[-0.5,0.5], if any of 5x10000 networks have an edge count less than 5 of greater than 10, we declare this point as a degeneracy point.

Example 1: Results

- For different m(the number of sweeps in Gibbs) = 1, 5, 10, $\theta^{(n)}$ converges to the same value.

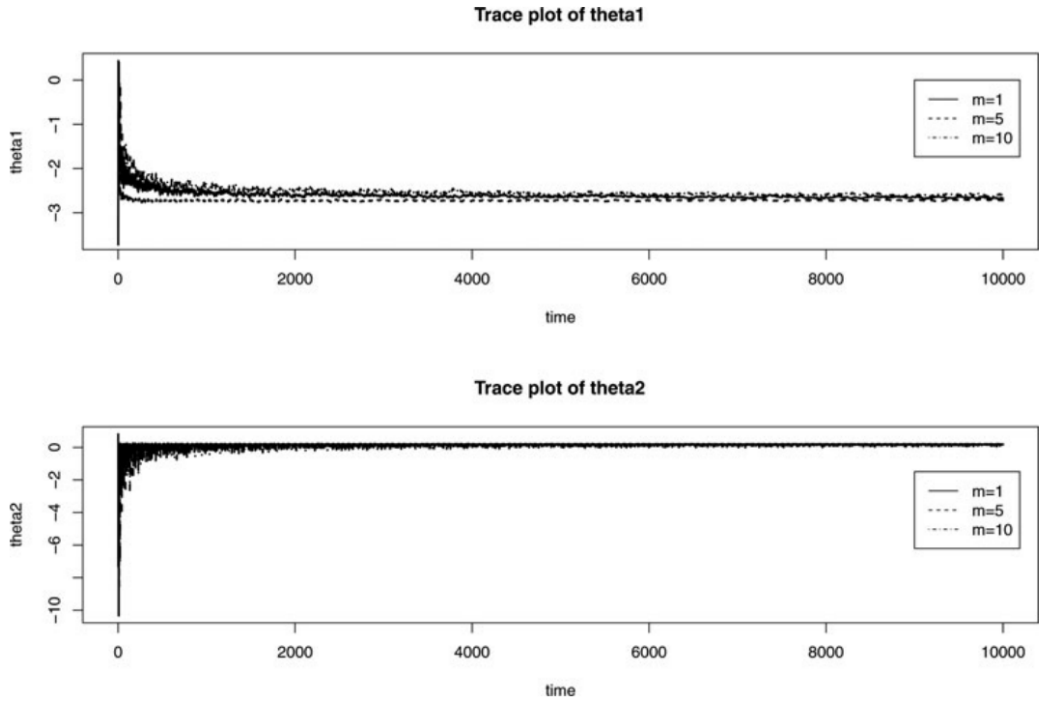

Figure 4. Trajectories of θ produced by SAMCMC for the Florentine business network with different values of m: $m = 1$ (red), $m = 5$ (black), and $m = 10$ (blue).

Example 2: The Student Friendship Network

- Node: 205 students grades 7 through 12
- Edge: mutual friendship(originally directed favor)
- Model 1 & 2

$$f(y|\theta) = \frac{1}{\kappa(\theta)} \exp\{\theta_1 S_1(y) + \theta_2 u(y|\tau) + \theta_3 w(y|\tau) + \theta_4 v(y|\tau) + \sum_{k=1}^{22} \theta_{k+4} \sum\sum_{i<j} y_{ij} h_k(X_i, X_j)\}$$

$$f(y|\theta) = \frac{1}{\kappa(\theta)} \exp\{\sum_{k=1}^{3} \theta_k S_k(y) + \theta_4 T(y) + \sum_{k=1}^{22} \theta_{k+4} \sum\sum_{i<j} y_{ij} h_k(X_i, X_j)\}$$

Example 2: The Student Friendship Network

- Model 3 & 4

$$f(y|\theta) = \frac{1}{\kappa(\theta)} \exp\{\theta_1 S_1(y) + \sum_{k=2}^{8} \theta_k D_k(y) + \theta_9 u(y|\tau) + \sum_{k=1}^{22} \theta_{k+9} \sum\sum_{i<j} y_{ij} h_k(X_i, X_j)\}$$

$$f(y|\theta) = \frac{1}{\kappa(\theta)} \exp\{\sum_{k=1}^{3} \theta_k S_k(y) + \theta_4 T(y) + \sum_{k=2}^{8} \theta_{k+3} D_k(y) + \theta_{12} u(y|\tau) + \theta_{13} w(y|\tau) + \theta_{14} v(y|\tau) + \sum_{k=1}^{22} \theta_{k+14} \sum\sum_{i<j} y_{ij} h_k(X_i, X_j)\}$$

Example 2: Results

- Goodness-of fit & Estimates

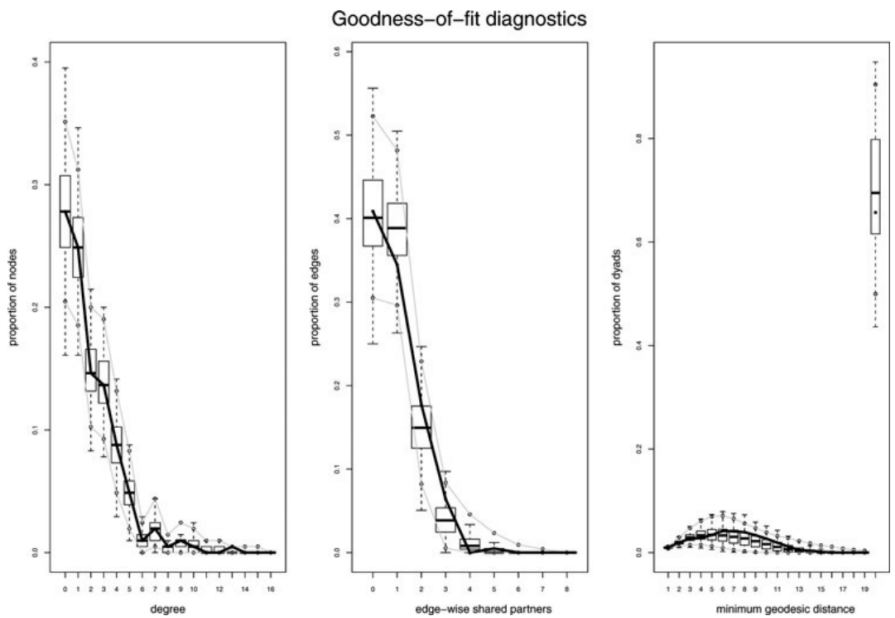

Figure 10. GOF plots for the high school student friendship network with fitted estimates using Model 4.

Table 6. Estimates produced by SAMCMC for the high school student friendship network. The estimates are calculated by averaging over five independent runs with their standard deviations (Monte Carlo error) given in the parentheses

Coefficient	Model 1	Model 2	Model 3	Model 4
Edge counts	$-10.512\ (3.2\times10^{-3})$	$-10.400\ (4.6\times10^{-2})$	$-10.052\ (1.3\times10^{-3})$	$-10.876\ (7.8\times10^{-2})$
k_2-star	–	$0.268\ (2.5\times10^{-4})$	–	$0.066\ (9.6\times10^{-3})$
k_3-star	–	$-0.068\ (2.8\times10^{-5})$	–	$-0.025\ (1.1\times10^{-3})$
Triangle	–	$0.939\ (5.4\times10^{-4})$	–	$0.853\ (6.4\times10^{-4})$
Degree (2)	–	–	$-0.721\ (1.3\times10^{-4})$	$-0.071\ (1.7\times10^{-2})$
Degree (3)	–	–	$-0.834\ (4.5\times10^{-4})$	$0.373\ (3.1\times10^{-2})$
Degree (4)	–	–	$-1.116\ (4.0\times10^{-4})$	$0.401\ (3.9\times10^{-2})$
Degree (5)	–	–	$-1.366\ (4.8\times10^{-4})$	$0.228\ (4.1\times10^{-2})$
Degree (6)	–	–	$-2.482\ (1.0\times10^{-3})$	$-1.012\ (4.1\times10^{-2})$
Degree (7)	–	–	$-1.166\ (6.3\times10^{-4})$	$0.056\ (3.4\times10^{-2})$
Degree (8)	–	–	$-1.812\ (1.3\times10^{-3})$	$-0.981\ (2.8\times10^{-2})$
GWD ($\tau=0.25$)	$0.006\ (3.9\times10^{-4})$	–	$-1.256\ (2.0\times10^{-4})$	$0.505\ (3.3\times10^{-2})$
GWDSP ($\tau=0.25$)	$0.007\ (5.2\times10^{-5})$	–	–	$0.103\ (1.4\times10^{-4})$
GWESP ($\tau=0.25$)	$1.376\ (1.6\times10^{-4})$	–	–	$1.034\ (5.8\times10^{-4})$
NF (grade 8)	$1.440\ (1.6\times10^{-3})$	$1.079\ (3.1\times10^{-2})$	$1.708\ (9.8\times10^{-4})$	$1.218\ (2.3\times10^{-2})$
NF (grade 9)	$2.184\ (1.4\times10^{-3})$	$1.972\ (4.0\times10^{-2})$	$2.691\ (1.0\times10^{-3})$	$2.000\ (3.1\times10^{-2})$
NF (grade 10)	$2.514\ (1.7\times10^{-3})$	$2.369\ (4.2\times10^{-2})$	$3.130\ (8.5\times10^{-4})$	$2.335\ (3.3\times10^{-2})$
NF (grade 11)	$2.295\ (1.2\times10^{-3})$	$2.121\ (3.7\times10^{-2})$	$2.760\ (1.2\times10^{-3})$	$2.111\ (2.9\times10^{-2})$
NF (grade 12)	$2.895\ (1.4\times10^{-3})$	$2.809\ (2.9\times10^{-2})$	$3.524\ (9.6\times10^{-4})$	$2.698\ (2.2\times10^{-2})$
NF (race: black)	$0.627\ (3.6\times10^{-4})$	$0.734\ (5.6\times10^{-4})$	$0.610\ (3.5\times10^{-4})$	$0.603\ (3.9\times10^{-4})$
NF (race: Hispanic)	$-0.385\ (2.5\times10^{-4})$	$-0.400\ (1.3\times10^{-3})$	$-0.455\ (3.1\times10^{-4})$	$-0.378\ (4.0\times10^{-4})$
NF (race: NatAm)	$-0.335\ (3.3\times10^{-4})$	$-0.387\ (1.0\times10^{-3})$	$-0.408\ (2.6\times10^{-4})$	$-0.337\ (4.5\times10^{-4})$
NF (sex: female)	$0.141\ (1.1\times10^{-4})$	$0.145\ (7.7\times10^{-5})$	$0.157\ (1.7\times10^{-4})$	$0.132\ (8.0\times10^{-5})$
AD (grade 1)	$-0.121\ (9.8\times10^{-4})$	$-0.161\ (4.3\times10^{-2})$	$-0.010\ (9.1\times10^{-4})$	$-0.211\ (3.1\times10^{-2})$
AD (grade 2)	$0.130\ (1.1\times10^{-3})$	$0.102\ (3.9\times10^{-2})$	$0.295\ (8.8\times10^{-4})$	$0.034\ (2.8\times10^{-2})$
AD (grade 3)	$-0.102\ (9.5\times10^{-4})$	$-0.202\ (3.3\times10^{-2})$	$-0.027\ (1.2\times10^{-3})$	$-0.195\ (2.4\times10^{-2})$
DHF (grade 7)	$6.007\ (2.7\times10^{-3})$	$5.891\ (4.3\times10^{-2})$	$7.538\ (1.6\times10^{-3})$	$5.532\ (2.8\times10^{-2})$
DHF (grade 8)	$3.251\ (7.9\times10^{-4})$	$3.862\ (2.6\times10^{-2})$	$4.402\ (1.3\times10^{-3})$	$3.217\ (1.7\times10^{-2})$
DHF (grade 9)	$1.595\ (1.0\times10^{-3})$	$1.886\ (4.9\times10^{-2})$	$2.145\ (1.4\times10^{-3})$	$1.499\ (3.3\times10^{-2})$
DHF (grade 10)	$1.078\ (1.4\times10^{-3})$	$1.203\ (5.4\times10^{-2})$	$1.428\ (6.9\times10^{-4})$	$0.969\ (3.8\times10^{-2})$
DHF (grade 11)	$1.868\ (1.3\times10^{-3})$	$2.115\ (4.1\times10^{-2})$	$2.665\ (1.1\times10^{-3})$	$1.751\ (3.1\times10^{-2})$
DHF (grade 12)	$1.034\ (7.4\times10^{-4})$	$1.112\ (2.2\times10^{-2})$	$1.476\ (5.6\times10^{-4})$	$0.946\ (1.7\times10^{-2})$
DHF (race: white)	$0.682\ (7.3\times10^{-4})$	$0.698\ (3.1\times10^{-3})$	$0.714\ (7.1\times10^{-4})$	$0.677\ (1.1\times10^{-3})$
DHF (race: Hispanic)	$0.566\ (5.5\times10^{-4})$	$0.544\ (5.0\times10^{-4})$	$0.683\ (4.7\times10^{-4})$	$0.559\ (3.6\times10^{-4})$
DHF (race: NatAm)	$1.052\ (3.1\times10^{-4})$	$1.084\ (3.4\times10^{-4})$	$1.265\ (2.2\times10^{-4})$	$1.035\ (3.9\times10^{-4})$
UHF (sex)	$0.544\ (2.3\times10^{-4})$	$0.577\ (1.5\times10^{-4})$	$0.630\ (2.0\times10^{-4})$	$0.538\ (1.8\times10^{-4})$
Time	12.8 hr	6.4 hr	3.5 hr	41.6 hr

NOTE: NF, node factor effect; DHF, different homophily effect; UHF, uniform homophily effect; AD, absolute different effect; NatAm, native American. Models 2 and 4 took 2.5 times more iterations than Models 1 and 3.

Conclusion

- The Varying truncation SAMCMC algorithm can be applied to ERGM.
- It prevents the estimater from going to degeneracy region by reinitialization.
- The estimator converges to MLE of the ERGM and is asymtotically normally distributed and asymtotically efficient.
- It is a solution for the model degeneracy problem despite of the normalizing constant.