Exponential-Family Random Graph Models with Time Varying Network Parameters

Abstract

Dynamic networks are a general language for describing time-evolving complex systems, and have long been an interesting research area. It is a fundamental research question to model time varying network parameters. However, due to difficulties in modeling functional network parameters, there is little progress in the current literature to effectively model time varying network parameters. In this work, we consider the situation in which network parameters are univariate nonparametric functions instead of constants. Using a kernel regression techniques, we introduce a novel unified procedure to effectively estimate those functional network parameters in the exponential-family random graph models. Moreover, by adopting the finite mixture models, we extend our model to mixture of exponential-family random graph models with functional network parameters which simultaneously allows both modeling and detecting communities for the dynamic networks. To choose optimal number of communities and kernel bandwidth, we propose conditional likelihood BIC and choose bandwidth by adopting the idea of network cross validation. Furthermore, we design an efficient variational expectation-maximization algorithm to find approximate maximum local likelihood estimates of network parameters and global estimates of mixing proportions. The power of our method is demonstrated in depth simulation studies and real-world applications to dynamic arm trade networks.

Keywords. Dynamic networks, Exponential-family random graph model (ERGM), Functional network parameter, Variational EM algorithm, Minorization-maximization, Model selection, Arm trade networks.
1 Introduction

Dynamic networks are a general language for describing time-evolving complex systems, and have long been an interesting research area. Recently, due to advances in data collection technologies, larger-scale and longer time span dynamic network analysis has been increasingly important in various research fields such as sociology, business, finance, bioinformatics, genomics, physics, artificial intelligence, and many others. In the current literature, to handle networks that change over time, many works are done under Markov assumption. The idea of dynamic Markovian models of network starts from modeling social network dynamics. Holland & Leinhardt (1977) proposed continuous-time Markov process and Wasserman (1979, 1980) continued and provided estimators for various parametric models. Later, Leenders (1995) studied maximum likelihood estimation for continuous-time Markov process models based on the assumptions that dyad processes are independent and stationary. Snijders (2001) relaxed assumptions of previous models and introduced parameterizations of continuous-time Markov processes that allow dyad processes to be dependent and less restrictive stationarity assumption.

More recently, the exponential-family random graph modeling (ERGM) framework has become a popular approach for conducting inference on network structure, due to its generalizability (Frank & Strauss 1986, Strauss & Ikeda 1990, Wasserman & Pattison 1996, Robins & Pattison 2001, Snijders et al. 2006, Hunter et al. 2008). Exponential family random graph models allow researchers to incorporate interesting features of the network into statistical models. For many years, the application of the ERGM framework was limited to static network datasets. However, recent developments have extended ERGMs to the modeling of dynamic networks under discrete-time Markov models as alternatives to continuous-time Markov models. Hanneke et al. (2010) proposed a temporal ERGM (TERGM) to fit the model to a network series and Krivitsky & Handcock (2014) proposed a separable temporal ERGM (STERGM) that gives more flexibility in modeling dynamic networks.

Several works are also done in non-Markovian approach to model dynamic networks. Butts (2008) developed a relational event framework based on survival analysis theory. Hunter et al. (2011) continued this approach and used multivariate counting processes to model network dynamics. They focused on dynamic egocentric framework and used multivariate counting process to every node to model continuous-time network data. Similarly, Perry & Wolfe (2013) used multivariate counting process to every pair of nodes which is also known as dynamic relational framework.

In this work, we consider the situation in which network parameters are univariate nonparametric functions instead of constants. Using a kernel regression techniques, we introduce a novel unified procedure to effectively estimate those functional network parameters in the exponential-family random graph models. Moreover, by adopting the finite mixture models, we extend our model to mixture of exponential-family random graph models with functional network parameters which simultaneously allows both modeling and detecting communities for the dynamic networks.

In Section 2, we first present exponential-family random graph model with functional network parameters. We also extend our model to mixture of exponential-family random graph models with functional network parameters.
parameters by adopting the finite mixture models. Section 3 designs an efficient variational expectation-maximization algorithm to find approximate maximum local likelihood estimates of network parameters and global estimates of mixing proportions. Given these estimates, we can further infer membership labels and solve the problem of community detection for dynamic networks. In Section 4, we first choose optimal number of communities using conditional likelihood BIC and choose kernel bandwidth by adopting the idea of network cross validation. The power of our method is demonstrated by simulation studies in Section 5 and real-world application to dynamic arm trade networks in Section 6.

2 Exponential-family Random Graph Models with Time Varying Network Parameters: A Semiparametric Approach

In this section, we present the mixture of exponential-family random graph models with functional network parameters. First we introduce some necessary notation. Let \( Y_t = (Y_{t,ij})_{1 \leq i \neq j \leq n} \in Y \) represent the network at time \( t \) and denote by \( y_t = (y_{t,ij})_{1 \leq i \neq j \leq n} \) the corresponding observed network at time \( t \). Define \( Y \) as the set of all possible networks.

2.1 Exponential-family random graph model with functional network parameters

Let \( \theta(\cdot) \) be the functional network parameters of interest. Given some univariate covariate \( u_t \), exponential-family random graph model with functional network parameters is written as follows:

\[
P_{\theta(u_t)}(Y_t = y_t) = \exp\{\theta(u_t)'g(y_t) - \psi(\theta(u_t))\}.
\]

where \( \psi(\theta(u_t)) = \log \sum_{y_t^* \in Y} \exp [\theta(u_t)'g(y_t^*)] \).

Here \( g(y_t) \) is the \( p \)-dimensional sufficient statistic from network \( y_t \) and \( C(\theta(u_t)) = \exp\{\psi(\theta(u_t))\} \) is the normalization term. However, the model in (1) is not scalable for modeling large networks. To make exponential-family distributions scalable, dyadic independence is usually used in the specification of ERGMs.

Therefore, we focus on scalable exponential-family random graph models, which are characterized by following dyadic independence,

\[
P_{\theta(u_t)}(Y_t = y_t) = \prod_{i < j} P_{\theta(u_t)}(D_{t,ij} = d),
\]

where \( D_{t,ij} \) corresponds to \( Y_{t,ij} \) in the case of undirected edges and \( (Y_{t,ij}, Y_{t,ji}) \) in the case of directed edges. The subscribed \( i < j \) and superscripted \( n \) mean that the product in (2) should be taken over all pairs \( (i,j) \) with \( 1 \leq i < j \leq n \).

Dyadic independence has at least two advantages: (a) it facilitates estimation, because the computing time to evaluate the likelihood function scales linearly with \( n^2 \); (b) by design it bypasses the so-called model degeneracy problem: if \( n \) is large, some exponential family models without dyadic independence then to be ill-defined and impractical for modeling networks (Strauss 1986, Handcock et al. 2003, Schweinberger 2011, Vu et al. 2013).
In this paper, we focus on scalable exponential-family models by assuming the dyadic independence.

2.2 Mixture of exponential-family random graph models with functional network parameters

Most exponential families with dyadic independence are either simplistic or nonparsimonious (model with $O(n)$ parameters) (Vu et al. 2013). We therefore adopt the finite mixture model, which offers powerful statistical techniques to identify subpopulations with certain commonality within overall population.

Define $Z = (Z_1, \ldots, Z_n)$ as the membership vector, where $Z_1, \ldots, Z_n \in \{1, \ldots, K\}$. Denote by $K$ the number of communities. Assume that $Z_i$’s are independently drawn from a multinomial distribution with parameter $\pi = (\pi_1, \ldots, \pi_K)$, where $\pi_k > 0$ for all $k$. We therefore assume that the probability mass function has a K-component mixture form as follows:

$$P_{\pi, \theta(u)}(Y_t = y_t) = \sum_{z \in \{1, \ldots, K\}^n} P_{\theta(u)}(Y_t = y_t | z) P_{\pi}(Z = z)$$

Here we extend dyadic independence to conditional dyadic independence given the community structure of networks as follows:

$$P_{\theta(u)}(Y_t = y_t | Z = z) = \prod_{i<j} P_{\theta_{z_i z_j}(u)}(D_{t,ij} = d | z).$$

Given (4), the probability mass function of K-component mixture is written as follows:

$$P_{\pi, \theta(u)}(Y_t = y_t) = \sum_{z \in \{1, \ldots, K\}^n} \prod_{i<j} P_{\theta_{z_i z_j}(u)}(D_{t,ij} = d | z) P_{\pi}(Z = z)$$

Now, given the independent data $\{(y_t, u_t), 1 \leq t \leq T\}$ our goal is to estimate $\pi$’s and nonparametric function $\theta(\cdot)$’s. The log-likelihood function for the observed data is given by

$$\ell = \sum_{t=1}^{T} \log \left[ P_{\pi, \theta(u)}(Y_t = y_t) \right]$$

$$= \sum_{t=1}^{T} \log \left[ \sum_{z \in \{1, \ldots, K\}^n} P_{\theta(u)}(Y_t = y_t | z) P_{\pi}(Z = z) \right]$$

Since $\theta(\cdot)$ is nonparametric, we need nonparametric smoothing techniques for (6). Here we employ kernel regression techniques. In kernel regression, we first use local constants $\theta^u$ to approximate $\theta(u)$. Let $K_h(\cdot) = h^{-1}K(\cdot/h)$ be a rescaled kernel of a kernel function $K(\cdot)$ with a bandwidth $h$. Then the corresponding local log-likelihood function for observed data is

$$\ell_u = \sum_{t=1}^{T} \log \left[ \sum_{z \in \{1, \ldots, K\}^n} P_{\theta^u}(Y_t = y_t | z) P_{\pi}(Z = z) \right] K_h(u_t - u)$$
Thus our aim is first solve parameter $\pi$ via maximizing the log-likelihood $\ell$,

$$\hat{\pi} = \arg\max_{\pi} \ell(\pi, \theta(u))$$

and solve local network parameters $\theta^u$ via maximizing the local log-likelihood $\ell_u$, namely

$$\hat{\theta}^u = \arg\max_{\theta^u} \ell_u(\hat{\pi}, \theta^u)$$

Before proceeding we give specific examples of undirected and directed network models.

**Example 1: Undirected Network Model**

Here we introduce one parameter: (a) edge (density) parameter, $\theta^e$. There are two possible values 1 and 0. The probabilities for two possible values of each dyad are given as follows:

1. $P_{\theta^e(u_i)}(D_{t,ij} = 1|z_i = k, z_j = l) = \exp(\theta^e_k(u_i) + \theta^e_l(u_i) - \psi(\theta^e(u_i)))$
2. $P_{\theta^e(u_i)}(D_{t,ij} = 0|z_i = k, z_j = l) = \exp(-\psi(\theta^e(u_i)))$

where $\psi(\theta^e(u_i)) = \log(1 + \exp(\theta^e_k(u_i) + \theta^e_l(u_i)))$.

**Example 2: Directed Network Model with Outgoing-edge and Reciprocity Parameters**

Here, we introduce two parameters: (a) outgoing-edge parameter, $\theta^{oe}$ and (b) reciprocity parameter, $\theta^{re}$. There are four possible values, $(1, 1), (1, 0), (0, 1)$ and $(0, 0)$. The probabilities for all possible values of each dyad are given as follows:

1. $P_{\theta^{(u_i)}}(D_{t,ij} = (1, 1)|z_i = k, z_j = l) = \exp(\theta^{oe}_k(u_i) + \theta^{re}_l(u_i) - \psi(\theta(u_i)))$
2. $P_{\theta^{(u_i)}}(D_{t,ij} = (1, 0)|z_i = k, z_j = l) = \exp(\theta^{oe}_k(u_i) - \psi(\theta(u_i)))$
3. $P_{\theta^{(u_i)}}(D_{t,ij} = (0, 1)|z_i = k, z_j = l) = \exp(\theta^{re}_k(u_i) - \psi(\theta(u_i)))$
4. $P_{\theta^{(u_i)}}(D_{t,ij} = (0, 0)|z_i = k, z_j = l) = \exp(-\psi(\theta(u_i)))$

where $\psi(\theta(u_i)) = \log(1 + \exp(\theta^{oe}_k(u_i) + \theta^{re}_l(u_i)) + \exp(\theta^{oe}_k(u_i)) + \exp(\theta^{re}_l(u_i)))$.

### 2.3 Parameter identifiability

**Lemma 1.** (Allman et al. 2011, Theorem 14) The parameters of the random graph mixture model, with $\kappa$-state edge variables and $K \geq 2$ latent groups, are identifiable, up to label switching, from the distribution of $K\theta$, provided $\kappa \geq \frac{(K+1)K}{2}$ and the $\kappa$-entry vectors $\{P_{kl}\}_{1 \leq k \leq K}$ are linearly independent.

**Assumption 1.** We assume that for any $u$, the $K$ values $\{\theta^u_k, 1 \leq k \leq K\}$ are distinct.

**Theorem 1.** Under the condition of Lemma 1 and with fixed community membership, mixing proportion parameter $\{\pi_k : 1 \leq k \leq K\}$ is identifiable up to label switching and local network parameters $\{\theta^u : \forall u \in U\}$ are identifiable up to global label switching.
3 Effective Variational EM algorithm

For a given \( u \), one may maximize the local log-likelihood function (7) using a variational EM algorithm. However, in practice we typically want to evaluate the unknown functions at a set of grid points. This requires us to maximize the local likelihood function (7) at different grid points. This imposes some challenges because the labels in the variational EM algorithm may change at different grid points. Thus a naive implementation of the variational EM algorithm may fail to yield smooth estimated curves. The key idea is that given current estimates of parameters, \( \tilde{\pi} \) and \( \tilde{\theta}(\cdot) \), E-step estimates variational parameter at observed \( u_1, \ldots, u_T \) and M-step uses the obtained common variational parameter estimate to update all estimates of network parameters at each grid point. Hence, we effectively prevent the label switching issue at different grid points.

3.1 Variational E-step

Using Jensen’s inequality, log-likelihood function is bounded from the below as follows:

\[
\ell(\pi, \theta(u_t)) = \sum_{t=1}^{T} \log \left( \sum_{z \in Z} P_{\pi, \theta(u_t)}(Y_t = y, Z = z) \right) \\
\geq \sum_{t=1}^{T} \sum_{z \in Z} \log \left( \frac{P_{\pi, \theta(u_t)}(Y_t = y, Z = z)}{A(z)} \right) A(z) \\
= \sum_{t=1}^{T} \left[ E_A(\log P_{\pi, \theta(u_t)}(Y_t = y, Z = z)) - E_A(\log A(z)) \right]
\]

Let \( \Gamma = (\gamma_1, \ldots, \gamma_n)' \) be the variational parameter, which denotes the mixed membership of nodes where \( \Gamma \) is \( (n \times K) \) matrix and \( \{\gamma_i, i = 1, \ldots, n\} \) are \( (K \times 1) \) vectors. Let \( P_{\gamma_i}(Z_i = z_i) \) be Multinomial \( (1; \gamma_{i1}, \ldots, \gamma_{iK}) \) for \( i = 1, 2, \ldots, n \). A natural subset of tractable choices is given by setting,

\[
A(z) = P_{\gamma}(Z = z) = \prod_{i=1}^{n} P_{\gamma_i}(Z_i = z_i)
\]

Now we can write our effective lower bound as follows:

\[
\text{ELBO}(\pi, \theta(u_t); \Gamma) = \sum_{t=1}^{T} \left[ \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \gamma_{ik} \gamma_{jl} \log P_{\theta(u_t)}(D_{t,ij} = d|z) \right] \\
+ T \left[ \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} (\log \pi_k - \log \gamma_{ik}) \right]
\]

Let \( \hat{\Gamma}, \hat{\pi}, \hat{\theta}(\cdot) \) be the current estimate. In the E-step our goal is to find updates, \( \gamma_{ik}^{\text{new}} \) by solving the following univariate optimization problem:

\[
\gamma_{ik}^{\text{new}} \leftarrow \arg \max_{\gamma_{ik}} \{ \text{ELBO}(\gamma_{ik}; \gamma_{i'k'} = \hat{\gamma}_{i'k'}, (i', k') \neq (i, k), \hat{\pi}, \hat{\theta}(\cdot)) \}
\]
We can write our lower bound with respect to $\gamma_i$ for $i = 1, \ldots, n - 1$ given all other current estimates,

$$ELBO(\gamma_i; \gamma_j = \tilde{\gamma}_j, j > i, \tilde{\pi}, \tilde{\theta}(\cdot)) = \sum_{t=1}^{T} \left[ \sum_{j=i+1}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \gamma_{ik} \tilde{\gamma}_{jl} \log P_{\tilde{\theta}(u_t)}(D_{t,ij} = d|z) \right]$$

$$+ T \sum_{k=1}^{K} \gamma_{ik} (\log \tilde{\pi}_k - \log \gamma_{ik}) \right]$$

(8)

and for $\gamma_n$,

$$ELBO(\gamma_n; \tilde{\pi}) = T \sum_{k=1}^{K} \gamma_{nk} (\log \tilde{\pi}_k - \log \gamma_{nk}) \right]$$

(9)

However, to solve the above optimization problem we have to solve non-convex optimization problem. To make computationally attractive, we introduce $Q(\gamma_i; \tilde{\Gamma}, \tilde{\pi}, \tilde{\theta}(\cdot))$ which is a minorization function of the lower bound (8) and (9).

Here we consider following minorization function. For $i = 1, \ldots, n - 1$,

$$Q(\gamma_i; \tilde{\Gamma}, \tilde{\pi}, \tilde{\theta}(\cdot)) = \sum_{t=1}^{T} \left[ \sum_{j=i+1}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \left( \frac{\tilde{\gamma}_{jl}}{2\tilde{\gamma}_{ik}} + \frac{\tilde{\gamma}_{jl}}{2\tilde{\gamma}_{ik}} \right) \log P_{\tilde{\theta}(u_t)}(D_{t,ij} = d|z) \right]$$

$$+ T \sum_{k=1}^{K} \gamma_{ik} (\log \tilde{\pi}_k - \log \gamma_{ik} - \frac{\gamma_{ik}}{\tilde{\gamma}_{ik}} + 1) \right]$$

(10)

and for $i = n$,

$$Q(\gamma_n; \tilde{\Gamma}, \tilde{\pi}) = T \sum_{k=1}^{K} \gamma_{nk} (\log \tilde{\pi}_k - \log \tilde{\gamma}_{nk} - \frac{\gamma_{nk}}{\tilde{\gamma}_{nk}} + 1) \right]$$

Note that $Q(\gamma_i; \tilde{\Gamma}, \tilde{\pi}, \tilde{\theta}(\cdot))$ is a concave function, and we can maximize $Q(\gamma_i; \tilde{\Gamma}, \tilde{\pi}, \tilde{\theta}(\cdot))$ solving a constrained quadratic programming problems, under constraints $\gamma_{i1}, \ldots, \gamma_{iK} \geq 0$ and $\sum_{k=1}^{K} \gamma_{ik} = 1$ for $i = 1, \ldots, n$.

### 3.2 Variational M-step

In the variational M-step, maximization with respect to $\pi$ and $\theta^u$ may be accomplished separately. First, to derive the closed-form updates for $\pi$, we maximize following lower bound via introducing Lagrange multiplier with the constraint $\sum_{k=1}^{K} \pi_k = 1$.

$$ELBO(\pi; \theta^u, \tilde{\Gamma}) = \sum_{t=1}^{T} \left[ \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \tilde{\gamma}_{ik} \tilde{\gamma}_{jl} \log P_{\theta^u}(D_{t,ij} = d|z) \right]$$

$$+ T \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\gamma}_{ik} (\log \pi_k - \log \tilde{\gamma}_{ik}) \right]$$

Then it is easy to obtain the closed-form update for $\pi$, that is

$$\tilde{\pi}^{\text{new}}_k = \frac{1}{n} \sum_{i=1}^{n} \tilde{\gamma}_{ik}, \quad k = 1, \ldots, K.$$  

(11)

Next to update $\theta^u$, we maximize following local lower bound using Newton-Raphson method with the
gradient and Hessian of (12).

\[
\text{ELBO}(\theta^u, \hat{\pi}, \hat{\Gamma}) = \sum_{t=1}^{T} \left[ \sum_{i<j} \sum_{k=1}^{K} \tilde{\gamma}_{ik} \tilde{\gamma}_{jl} \log P_{\theta^u}(D_{t,ij} = d|z) \right] K_{h}(u_{t} - u) \\
+ \sum_{t=1}^{T} \left[ \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\gamma}_{ik} (\log \tilde{\pi}_k - \log \tilde{\gamma}_{ik}) \right] K_{h}(u_{t} - u).
\]

(12)

The successor point \( \hat{\theta}^{u(new)} \) is given by

\[
\hat{\theta}^{u(new)} = \hat{\theta}^{u} - H(\hat{\theta}^{u})^{-1} \nabla \text{ELBO}(\hat{\theta}^{u}, \hat{\pi}, \hat{\Gamma}).
\]

(13)

### 3.3 Ascent property of ELBO

We can show that our variational EM algorithm preserves ascent property of lower bound of the log-likelihood, which leads the best estimates of lower bound. In Variational E-step, we maximize our lower bound with respect to variational parameters through MM algorithm via introducing surrogate function to minorize the lower bound which satisfy following equations,

\[
Q(\gamma_i; \hat{\Gamma}, \hat{\pi}, \hat{\theta}^u) \leq \text{ELBO}(\gamma_i; \gamma_j = \tilde{\gamma}_j, j > i, \hat{\pi}, \hat{\theta}^u),
\]

\[
Q(\tilde{\gamma}_i; \hat{\Gamma}, \hat{\pi}, \hat{\theta}^u) = \text{ELBO}(\tilde{\gamma}_i; \gamma_j = \tilde{\gamma}_j, j > i, \hat{\pi}, \hat{\theta}^u),
\]

for all \( i = 1, \ldots, n \). Therefore using above equations we have,

\[
\text{ELBO}(\gamma_j; \gamma_j = \tilde{\gamma}_j, j > i, \hat{\pi}, \hat{\theta}^u) \leq Q(\tilde{\gamma}_i^{(new)}; \hat{\Gamma}, \hat{\pi}, \hat{\theta}^u) \leq \text{ELBO}(\tilde{\gamma}_i^{(new)}; \gamma_j = \tilde{\gamma}_j, j > i, \hat{\pi}, \hat{\theta}^u),
\]

for \( i = 1, \ldots, n \). In Variational M-step, we maximize our local lower bound with respect to network parameters which leads,

\[
\text{ELBO}(\hat{\pi}, \hat{\theta}^u; \tilde{\gamma}^{(new)}) \leq \text{ELBO}(\tilde{\pi}^{(new)}, \hat{\theta}^{u(new)}, \tilde{\gamma}^{(new)}).
\]

Therefore, we prove the ascent property of local lower bound of the log-likelihood,

### 4 Model Selection

In practice, it is important to effectively choose the number of communities. To determine number of communities \( K \), we consider the information criterion approach. Bayesian Information Criterion (BIC) has the general form of \(-2L + \delta \times df\), where \( L \) is the maximum log-likelihood, \( \delta = \log N \), and \( df \) is the degree of freedom to measure model complexity.

Here, we use the conditional likelihood of the network series, conditioning on an estimate of the membership vector, to construct an effective model selection criterion.
We obtain the conditional log-likelihood of the network series \( y_1, y_2, \ldots, y_T \) given estimated membership vector as

\[
cl(\theta(\cdot), \hat{z}) = \sum_{t=1}^{T} \log(P_{\theta(u_t)}(Y_t = y_t|\hat{z}))
\]

which can be written using conditional dyadic independence (4) in the form

\[
cl(\theta(\cdot), \hat{z}) = \sum_{t=1}^{T} \sum_{i<j} \log(P_{\theta_{i,j}(u_t)}(D_{t,ij} = d|\hat{z})).
\]

Next, to specify the degree of freedom in (3), we follow Fan et al. (2001) and Huang et al. (2013) to derive the degree of freedom. Denote by \( \text{df} = \tau_K h^{-1} |Z| \left( K(0) - \frac{1}{2} \int K^2(t) \, dt \right) \) the degree of freedom of a univariate nonparametric function, where \( Z \) is the support of the covariate \( Z \), and \( \tau_K = \frac{K(0) - \frac{1}{2} \int K^2(t) \, dt}{\int (K(t) - \frac{1}{2} K \ast K(t))^2 \, dt} \). Hence, for each pair of \((K, h)\), the conditional likelihood BIC is defined as

\[
\text{clBIC}(K, h) = -2cl(\hat{\theta}_{\text{mle}}(\cdot), \hat{z}) + \log(T(n(n-1)/2)) \cdot \text{df}(K, h),
\]

where \( \text{df}(K, h) = \text{tr}(H^{-1}V_K) \cdot \text{df} \) approximates the degrees of freedom based on \( H_K = E(-\nabla^2_{\theta(\cdot)} cl(\theta(\cdot), \hat{z})) \) and \( V_K = \text{Var}(\nabla_{\theta(\cdot)} cl(\theta(\cdot), \hat{z})) \). We first select optimal \( K \) by minimizing the clBIC score, and then choose \( h \) by network cross-validation (NCV). We choose \( K \) by minimizing the best available clBIC score for each choice of \( K \) over different choices of \( h \), Namely,

\[
\hat{K} = \arg \min_{(K, h)} \text{clBIC}(K, h)
\]

After fixing \( K = \hat{K} \), we use negative conditional log-likelihood to construct loss, and choose bandwidth \( h \) by NCV. We follow Chen and Lei (2016) to describe the \( V \)-fold dynamic network cross-validation procedure. We summarize the details in Algorithm 1. We follow Chen & Lei (to appear) to perform NCV multiple times with independent \( V \)-fold splits and output the most frequent \( \hat{h} \). We used \( V = 3 \) and twenty repetitions.

5 Simulation Studies

In this section, we conduct two simulation studies. Section 5.1 considers ERGM with functional network parameters, while Section 5.2 considers mixture of ERGMs with functional network parameters. Before proceeding, we first introduce the common simulation setting and two measurements to compare numerical performances. We assume the dynamic network data are observed at equally spaced 51 discrete time points, with node size 100. Here, we introduce several average metrics over 100 replications.

First, to assess the performance of the estimator of the network parameter function, we consider the square
Algorithm 1 $V$-fold dynamic networks cross-validation

**Input:** adjacency matrices, $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_T$, a set of candidate values for $h$, number of folds $V \geq 2$.

1. Block-wise node-pair splitting:
   Randomly split the nodes into $V$ equal-sized subsets $\{N_v : 1 \leq v \leq V\}$, and split the adjacency matrices corresponding into $V \times V$ equal sized blocks for $t = 1, \ldots, T$,
   \[
   \mathbf{Y}_t = (\mathbf{Y}_t^{(uv)} : 1 \leq u, v \leq V),
   \]
   where $\mathbf{Y}_t^{(uv)}$ is the submatrix of $\mathbf{Y}_t$ with rows in $N_u$ and columns in $N_v$.

2. For each $1 \leq v \leq V$, and each $h$
   - **Estimation:** Estimate functional network parameters $\theta(\cdot)$ and membership vector $\mathbf{z}$ using the submatrix, $\mathbf{Y}_t^{(v)}$ obtained by removing the $\mathbf{Y}_t^{(vv)}$ in subset $N_v$ for all $t = 1, \ldots, T$,
   - **Validation:** Calculate the predictive loss evaluated on $\mathbf{Y}_t^{(vv)}$. For the loss function $\ell$, we use negative conditional log-likelihood.

3. Let $\hat{L}(\mathbf{Y}_1, \ldots, \mathbf{Y}_T, h) = \sum_{v=1}^{V} \hat{L}^{(v)}(\mathbf{Y}_1, \ldots, \mathbf{Y}_T, h)$.

**Output:**

\[
\hat{h} = \arg\min_h \hat{L}(\mathbf{Y}_1, \ldots, \mathbf{Y}_T, h)
\]

root of the average squared error (RASE) of estimator of the function,

\[
\text{RASE}_\theta = \sqrt{\frac{1}{G} \sum_{u \in U} \sum_{k=1}^{K} \left( \hat{\theta}_k^u - \theta_k^u \right)^2},
\]

where $G$ is the number of grid points.

Next, to assess the clustering performance, we calculate the Rand Index (RI) (Rand 1971). The measure $\text{RI}(\mathbf{z}, \hat{\mathbf{z}})$ calculates the proportion of pairs whose estimated labels correspond to the true labels in terms of being assigned to the same or different groups:

\[
\text{RI}(\mathbf{z}, \hat{\mathbf{z}}) = \frac{1}{\binom{n}{2}} \sum_{i < j} (I\{z_i = z_j\}I\{\hat{z}_i = \hat{z}_j\} + I\{z_i \neq z_j\}I\{\hat{z}_i \neq \hat{z}_j\}).
\]

### 5.1 ERGM with functional network parameters

In this section, we conduct simulation for exponential-family random graph model with functional network parameter. Model 1 is directed network with functional out-going edge parameter and functional reciprocity parameter, $\theta_1^{oe}(u) = 1 - \sin(2\pi u)$ and $\theta_1^{rc}(u) = \cos(3\pi u) - 1$, i.e., Example 2.

To simulate directed dynamic networks, we generate all the dyads between two nodes at each time point from probabilities with specified functional network parameters in Model 1.
Now we examine the functional network parameter estimation performance using $\text{RASE}_\theta$. The results are reported in Table 1. Overall, our proposed method achieves a reasonably low $\text{RASE}_\theta$. Moreover, we plot true network parameter functions and estimated network parameter functions in Figure 1 using a typical sample from simulated dynamic networks which has median $\text{RASE}_\theta$. As you can see from Figure 1, estimated network parameter functions are very close to true network parameter functions.

Table 1: Mean and standard deviation of $\text{RASE}_{\theta^\text{oe}}$ and $\text{RASE}_{\theta^\text{re}}$ for Model 1.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$h$</th>
<th>$\text{RASE}_{\theta^\text{oe}}$</th>
<th>$\text{RASE}_{\theta^\text{re}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.05</td>
<td>0.0287 (0.0044)</td>
<td>0.0409 (0.0079)</td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td>0.0308 (0.0045)</td>
<td>0.0425 (0.0081)</td>
</tr>
<tr>
<td></td>
<td>0.07</td>
<td>0.0379 (0.0048)</td>
<td>0.0529 (0.0087)</td>
</tr>
<tr>
<td></td>
<td>0.08</td>
<td>0.0423 (0.0048)</td>
<td>0.0593 (0.0088)</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>0.0521 (0.0048)</td>
<td>0.0765 (0.0089)</td>
</tr>
</tbody>
</table>

Figure 1: Plots of true network parameter functions and estimated network parameter functions with a typical sample of simulated dynamic networks ($n = 100$).

5.2 Mixture of ERGMs with functional network parameters

Here, we conduct simulation for a two-component mixture of semiparametric exponential-family random graph models. Model 2 is mixture of undirected network with functional edge parameters, $\theta_1^e(u) = \frac{1}{2} \cos(3\pi u); \theta_2^e(u) = -\frac{1}{2} \cos(3\pi u) - 0.75$, i.e., two-component mixture of Example 1.

To simulate $K$-component mixture of semiparametric exponential-family random graph models, we specify each network structure by choosing randomly the memberships of the nodes according to the fixed mixing proportions. Next, we simulate all the dyads between two nodes at each time point based on the probabilities with specified functional network parameters and given membership. Here, we focus on the two-component mixture with mixing proportion, $\pi = (0.5, 0.5)$ and simulate undirected dynamic networks which has two clusters.
We first check the performance of algorithm at identifying the correct number of communities. For given bandwidth $h$, we count the frequencies of selected $K$’s over 100 repeats. For each simulated dynamic network data, we should select $K$ by minimizing the cIBIC score over the four $K$’s and five bandwidths. The frequencies of such selected $K$’s over 100 repeats are presented in the last column of Table 2. As shown in Table 2, our proposed cIBIC criterion based on min cIBIC has a convincing performance of choosing correct number of communities.

Table 2: Frequencies of selected K’s by cIBIC for Model 2.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$h = 0.05$</th>
<th>$h = 0.06$</th>
<th>$h = 0.07$</th>
<th>$h = 0.08$</th>
<th>$h = 0.09$</th>
<th>$\text{min}_h \text{cIBIC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>$K = 1$ 0</td>
<td>$K = 2$ 87</td>
<td>$K = 3$ 12</td>
<td>$K = 4$ 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$K = 2$ 87</td>
<td>$K = 2$ 85</td>
<td>$K = 3$ 14</td>
<td>$K = 4$ 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$K = 3$ 84</td>
<td>$K = 3$ 83</td>
<td>$K = 3$ 15</td>
<td>$K = 4$ 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$K = 4$ 83</td>
<td>$K = 4$ 83</td>
<td>$K = 4$ 16</td>
<td>$K = 4$ 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The average Rand Index results are reported in Table 3 and we achieve a high average Rand Index for the correct number of mixtures. When $K = 1$, all nodes are clustered into the same community and so the Rand Index only depends on the true labels. The results of Table 2 and 3 together tell us that our algorithm based on conditional likelihood BIC performs convincingly in choosing the correct number of communities and assigning nodes to communities.

Table 3: Mean and standard deviation of Rand Index for Model 2.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$h$</th>
<th>$K = 1$</th>
<th>$K = 2$</th>
<th>$K = 3$</th>
<th>$K = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.05</td>
<td>0.4993 (0.0067)</td>
<td>0.9791 (0.0130)</td>
<td>0.8870 (0.0380)</td>
<td>0.8299 (0.0466)</td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td>0.4993 (0.0067)</td>
<td>0.9793 (0.0128)</td>
<td>0.8870 (0.0379)</td>
<td>0.8299 (0.0465)</td>
</tr>
<tr>
<td></td>
<td>0.07</td>
<td>0.4993 (0.0067)</td>
<td>0.9791 (0.0130)</td>
<td>0.8870 (0.0379)</td>
<td>0.8298 (0.0467)</td>
</tr>
<tr>
<td></td>
<td>0.08</td>
<td>0.4993 (0.0067)</td>
<td>0.9791 (0.0130)</td>
<td>0.8870 (0.0379)</td>
<td>0.8296 (0.0466)</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>0.4993 (0.0067)</td>
<td>0.9789 (0.0128)</td>
<td>0.8867 (0.0378)</td>
<td>0.8294 (0.0463)</td>
</tr>
</tbody>
</table>

Next, we examine the functional network parameter estimation performance using $\text{RASE}_{\theta_e}$. The results are reported in Table 4. Our proposed method achieves a reasonably low $\text{RASE}_{\theta_e}$. In this simulation, both estimation and selection performances of our proposed method is not sensitive to bandwidths.

Table 4: Mean and standard deviation of $\text{RASE}_{\theta_e}$ for Model 2.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$h$</th>
<th>$\text{RASE}_{\theta_e}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.05</td>
<td>0.1851 (0.0185)</td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td>0.1862 (0.0185)</td>
</tr>
<tr>
<td></td>
<td>0.07</td>
<td>0.1907 (0.0183)</td>
</tr>
<tr>
<td></td>
<td>0.08</td>
<td>0.1930 (0.0182)</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>0.1988 (0.0180)</td>
</tr>
</tbody>
</table>
6 Arm Trade Data

In real application, we use a data from the Stockholm International Peace Research Institute (SIPRI) Arms Transfers Database which contains information on all transfers of major conventional weapons from 1950. Here we focus on yearly arm trade networks from 1994 to 2013. To define networks, we first define edges as follows: for any $t = 1994, \ldots, 2013$, $y_{t,ij} = 1$ if the volume of international transfers of arms, measured by trend indicator value (TIV) from country $i$ to country $j$ in year $t$ exceeds 1 million dollar in year $t$, and $y_{t,ij} = 0$ otherwise. Next, to define the nodes in the networks, we choose the countries that satisfy above edge criterion at least three times from 1994 to 2013. By using this criterion, among 198 existing countries, 145 countries are remaining in the networks. Now, we model the arm trade networks employing mixture of semiparametric ERGMs with out-going edge parameter and reciprocity parameter, i.e., $K$-component mixture of Example 2.

We use our proposed clBIC to determine the number of communities and $K = 4$ is chosen, which implies four communities of countries with different functional out-going edge parameter and reciprocity parameter. Next we use NCV to choose bandwidth and $h = 0.25$ is chosen.

In Figure 2, we plot the estimate out-going edge parameter and reciprocity parameter functions for each community and in Figure 3 we choose four specific years and plot arm trade networks with estimated communities. As you can see from Figure 2, there are overall decreasing trend in arm trades until early 2000, however after that there are overall increasing trends. Countries in community $K_3$ show decreasing trend after 2004, while countries in community $K_1$ and $K_2$ show slight decreasing trend after 2010. Countries in community $K_4$ have most arm trades among all communities, while countries in $K_3$ have least arm trades, which appear as isolated nodes in Figure 3. Countries in $K_1$ and $K_2$ show similar trends but in different scale. Countries in $K_1$ have slightly more arm trades than countries in $K_2$.

Figure 2: Estimated out-going edge parameter and reciprocity parameter functions for each community
Figure 3: Arm trade networks with estimated communities in four different years. Nodes assigned to $K_1$, $K_2$, $K_3$, and $K_4$ are colored orange, green, blue, and violet, respectively.

References


