Parameter Estimation in ERGMs: Fundamentals and computational challenges

Dave Hunter

Penn State Dept. of Statistics
Joint with Mark and Carter and many others

MURI networks grant meeting, November 18, 2008

Reminder: The Model Class

Exponential-Family Random Graph Model (ERGM)

$$P_\theta(Y = y) \propto \exp\{\theta^t g(y)\} \quad \text{for all } y \in \mathcal{Y}.$$

or

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)},$$

where

- Y is a random network on n nodes (a matrix of 0's and 1's)
- $\theta \in \mathbb{R}^p$ is a vector of parameters
- $g : \mathcal{Y} \to \mathbb{R}^p$ is given: $g(y)$ are the graph statistics
- $\kappa(\theta)$ makes all the probabilities sum to 1
- $\mathcal{Y}$ is fairly restrictive for now (e.g., node set is fixed)

The goal of estimation

Exponential-family Random Graph Model (ERGM)

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)} \quad \text{for all } y \in \mathcal{Y}.$$

If θ is not known, the above equation defines a model *class*, not a model.

Goal:

Use observed data (a network y^{obs}) to determine the "best" model from the model class.

In other words, find the "best" θ.

The likelihood function

The model class:

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)}$$

The likelihood function

The model class:

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)}$$

- It follows that $\kappa(\theta)$ is a normalizing “constant”:

$$\kappa(\theta) = \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\theta^t g(z)\}.$$

The likelihood function

The model class:

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)}$$

- It follows that $\kappa(\theta)$ is a normalizing "constant":

$$\kappa(\theta) = \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\theta^t g(z)\}.$$

- Let y^{obs} denote the observed graph, i.e., the data.

The likelihood function

The model class:

$$P_\theta(Y = y) = \frac{\exp\{\theta^t g(y)\}}{\kappa(\theta)}$$

- It follows that $\kappa(\theta)$ is a normalizing "constant":

$$\kappa(\theta) = \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\theta^t g(z)\}.$$

- Let y^{obs} denote the observed graph, i.e., the data.
- Likelihood function: View $P_\theta(Y = y^{\mathrm{obs}})$ as function of θ
- Goal: Find $\hat{\theta}$ that maximizes log of likelihood

$$\ell(\theta) = \theta^t g(y^{\mathrm{obs}}) - \log \kappa(\theta).$$

- Result: The maximum likelihood estimate.

More on MLE: Challenges

More on MLE: Challenges

The fact that $P_{\hat{\theta}}(Y = y^{\text{obs}})$ is as large as possible in this model class does NOT mean that y^{obs} is particularly likely relative to other networks!
(The model class itself might be inappropriate. We call this *degeneracy*.)

More on MLE: Challenges

The fact that $P_{\hat{\theta}}(Y = y^{\text{obs}})$ is as large as possible in this model class does NOT mean that y^{obs} is particularly likely relative to other networks!
(The model class itself might be inappropriate. We call this *degeneracy*.)

$\ell(\theta) = \theta^t g(y^{\text{obs}})$ is in general incredibly difficult to evaluate, let alone maximize:
Evaluating $\kappa(\theta)$ directly involves $2^{\binom{n}{2}}$ summands.

A nifty fact regarding the MLE $\hat{\theta}$

- Because we're dealing with an exponential family of models,

$$\mathrm{E}_{\hat{\theta}}\, g(Y) = g(y^{\mathrm{obs}})$$

  and no other value of θ has this property.
- In words:

  The MLE gives the unique model in the model class under which the mean value of the vector of statistics equals its observed value.

- This fact may even be exploited to approximate $\hat{\theta}$. (See Snijders 2002, *J. of Social Structure*. Idea is to use a Robbins-Monro-like algorithm.)

Different approach: Approximate log-likelihood ratio

- Suppose we fix θ_0. A bit of algebra shows that

$$\ell(\theta) - \ell(\theta_0) = (\theta - \theta_0)^t g(y^{\text{obs}}) - \log \mathrm{E}_{\theta_0}\left[\exp\left\{(\theta - \theta_0)^t g(Y)\right\}\right].$$

Different approach: Approximate log-likelihood ratio

- Suppose we fix θ_0. A bit of algebra shows that

$$\ell(\theta) - \ell(\theta_0) = (\theta - \theta_0)^t g(y^{\text{obs}}) - \log \mathrm{E}_{\theta_0}\left[\exp\left\{(\theta - \theta_0)^t g(Y)\right\}\right].$$

- Thus, $\ell(\theta) - \ell(\theta_0)$ involves a mean, which may be approximated by a sample mean:

$$\ell(\theta) - \ell(\theta_0) \approx (\theta - \theta_0)^t g(y^{\text{obs}}) - \log \frac{1}{m} \sum_{i=1}^{m} \exp\left\{(\theta - \theta_0)^t g(Y_i)\right\},$$

where $Y_1, Y_2, \ldots, Y_m$ is a random sample of networks from the distribution defined by the ERGM with parameter θ_0.

Different approach: Approximate log-likelihood ratio

- Suppose we fix θ_0. A bit of algebra shows that

$$\ell(\theta) - \ell(\theta_0) = (\theta - \theta_0)^t g(y^{\text{obs}}) - \log \mathrm{E}_{\theta_0}\left[\exp\left\{(\theta - \theta_0)^t g(Y)\right\}\right].$$

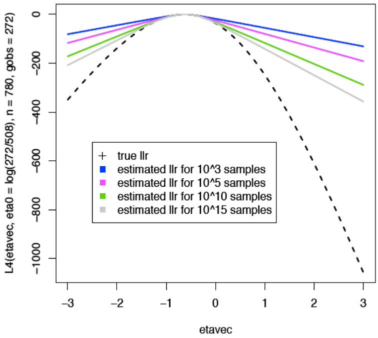
- Thus, $\ell(\theta) - \ell(\theta_0)$ involves a mean, which may be approximated by a sample mean:

$$\ell(\theta) - \ell(\theta_0) \approx (\theta - \theta_0)^t g(y^{\text{obs}}) - \log \frac{1}{m} \sum_{i=1}^{m} \exp\left\{(\theta - \theta_0)^t g(Y_i)\right\},$$

  where $Y_1, Y_2, \ldots, Y_m$ is a random sample of networks from the distribution defined by the ERGM with parameter θ_0.

- So simulating random networks enables approximate MLE.

Challenge: Approximation of LLR is very hard



← *from working paper of Ruth Hummel, PSU student supported by MURI grant this semester.*

- Naive approximation of LLR not good far from θ_0, even for gigantic samples
- Possible remedies: Smarter approximation; keeping close to θ_0; exploiting other existing techniques for ratios of normalizing constants

How should θ_0 be chosen?

- Theoretically, the estimated value of $\ell(\theta) - \ell(\theta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of θ_0.
- (Challenge: Building better MCMC routines — and parallelization — will always help.)

How should θ_0 be chosen?

- Theoretically, the estimated value of $\ell(\theta) - \ell(\theta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of θ_0.
- (Challenge: Building better MCMC routines — and parallelization — will always help.)
- However, in practice this convergence can be agonizingly slow, especially if θ_0 is not chosen close to the maximizer of the likelihood.
- A choice that sometimes works is the MPLE (maximum pseudolikelihood estimate).

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

Conditional on $Y_{ij}^c = y_{ij}^c$, Y has only two possible states, depending on whether $Y_{ij} = 0$ or $Y_{ij} = 1$.

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

Conditional on $Y_{ij}^c = y_{ij}^c$, Y has only two possible states, depending on whether $Y_{ij} = 0$ or $Y_{ij} = 1$.
Let's calculate the ratio of the two respective probabilities.

[We'll use $P_\theta(Y = y) = \exp\{\theta^t g(y)\}/\kappa(\theta)$.]

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

$$\frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \frac{\exp\{\theta^t g(y_{ij}^+)\}}{\exp\{\theta^t g(y_{ij}^-)\}}$$

A lot of cancellation happened on the right hand side!

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

$$\frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \exp\{\theta^t [g(y_{ij}^+) - g(y_{ij}^-)]\}$$

A lot of cancellation happened on the right hand side!

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in y other than (i, j)
- y_{ij}^+ denotes the same network as y but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as y but with $y_{ij} = 0$

$$\log \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \theta^t [g(y_{ij}^+) - g(y_{ij}^-)]$$

MPLE background: Conditional log-odds of edge

Notation: For a network y and a pair (i, j) of nodes,

- $\Delta g(y)_{ij}$ denotes the vector of change statistics,

$$\Delta g(y)_{ij} = g(y_{ij}^{+}) - g(y_{ij}^{-}).$$

So $\Delta g(y)_{ij}$ is the conditional log-odds of edge (i, j).

$$\log \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} \quad = \quad \theta^t \Delta g(y)_{ij}$$

NB: The change statistics $\Delta g(y)_{ij}$ are integral to both MCMC and MPLE.

MPLE: Intuition

- Assume that there is no dependence among the Y_{ij}.
- In other words, assume the marginal $P(Y_{ij} = 1)$ and the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$ coincide.

MPLE: Intuition

- Assume that there is no dependence among the Y_{ij}.
- In other words, assume the marginal $P(Y_{ij} = 1)$ and the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$ coincide.
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \theta^t \Delta g(y^{\mathrm{obs}})_{ij},$$

so we obtain an estimate of θ using straightforward logistic regression.

MPLE: Intuition

- Assume that there is no dependence among the Y_{ij}.
- In other words, assume the marginal $P(Y_{ij} = 1)$ and the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$ coincide.
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \theta^t \Delta g(y^{\mathrm{obs}})_{ij},$$

  so we obtain an estimate of θ using straightforward logistic regression.
- Result: The maximum pseudolikelihood estimate.

MPLE warnings & challenges

Unfortunately, little is known about the quality of MPL estimates in general, but they can be very bad (cf. van Duijn et al, 2008).

- If the model is bad, you'll get MPLE results quite easily (unlike MLE results), masking the problem.
- If the model is good, in many cases the MPLE looks "close" to the MLE; however, "close" can be deceiving, since small changes in θ can sometimes lead to large differences in the behavior of randomly generated networks.

MPLE warnings & challenges

Unfortunately, little is known about the quality of MPL estimates in general, but they can be very bad (cf. van Duijn et al, 2008).

- If the model is bad, you'll get MPLE results quite easily (unlike MLE results), masking the problem.
- If the model is good, in many cases the MPLE looks "close" to the MLE; however, "close" can be deceiving, since small changes in θ can sometimes lead to large differences in the behavior of randomly generated networks.

Nevertheless, if MPLE must be found...

- For large networks, MPLE can be computationally burdensome: There are $\binom{n}{2}$ "observations" in a linear regression model.

MPLE warnings & challenges

Unfortunately, little is known about the quality of MPL estimates in general, but they can be very bad (cf. van Duijn et al, 2008).

- If the model is bad, you'll get MPLE results quite easily (unlike MLE results), masking the problem.
- If the model is good, in many cases the MPLE looks "close" to the MLE; however, "close" can be deceiving, since small changes in θ can sometimes lead to large differences in the behavior of randomly generated networks.

Nevertheless, if MPLE must be found...

- For large networks, MPLE can be computationally burdensome: There are $\binom{n}{2}$ "observations" in a linear regression model.
- MPLE via change statistics requires a network y^{obs}; yet the model depends on y only through $g(y)$ so what if we have only $g(y^{\mathrm{obs}})$? One answer: Find a network whose statistics are equal to $g(y\mathrm{obs})$.

Curved Exponential Families

- Degree distributions get a lot of attention. For a network on (say) $n = 100$ nodes, denoted by Y, we posit an ERGM in which

$$P_\eta(Y = y) \propto \eta_0 E(y) + \eta_1 D_1(y) + \cdots + \eta_{99} D_{99}(y),$$

  where $D_i(y) = \#$ nodes of degree i.
- Death by parameter!

Curved Exponential Families

- Degree distributions get a lot of attention. For a network on (say) $n = 100$ nodes, denoted by Y, we posit an ERGM in which

$$P_\eta(Y = y) \propto \eta_0 E(y) + \eta_1 D_1(y) + \cdots + \eta_{99} D_{99}(y),$$

  where $D_i(y) = \#$ nodes of degree i.
- Death by parameter!
- More parsimonious model: For $1 \leq i \leq 99$, let

$$\eta_i(\theta, \alpha) = \theta e^{\alpha} \left[1 - (1 - e^{-\alpha})^i \right].$$

Curved Exponential Families

- Degree distributions get a lot of attention. For a network on (say) $n = 100$ nodes, denoted by Y, we posit an ERGM in which

$$P_\eta(Y = y) \propto \eta_0 E(y) + \eta_1 D_1(y) + \cdots + \eta_{99} D_{99}(y),$$

  where $D_i(y) = \#$ nodes of degree i.
- Death by parameter!
- More parsimonious model: For $1 \leq i \leq 99$, let

$$\eta_i(\theta, \alpha) = \theta e^{\alpha} \left[1 - (1 - e^{-\alpha})^i \right].$$

- η_i is nonlinear in α (hence *curved* EF model)
- Challenge: Maximizing MLE is even harder here and requires a lot of storage.