A Perfect Sampling Method for Exponential Random Graph Models

Carter T. Butts
Department of Sociology and
Institute for Mathematical Behavioral Sciences
University of California, Irvine
buttsc@uci.edu

This work was supported by ONR award N00014-08-1-1015.
The Basic Issue

- ERG-parameterized models represent a major advance in the study of social (and other) networks...
  - Fully generic representation for models on finite graph sets
  - (Relatively) well-developed inferential theory
  - Increasingly well-developed theory of model parameterization (though much more is needed!)

- But no general way to perform exact simulation
  - “Easy” special cases exist (e.g., $N, p$), but direct methods exponentially hard in general
  - So far, exclusive reliance on approximate simulation using MCMC; can work well, but quality hard to ensure

- Since almost all ERG applications involve simulation, this is a major issue!
Assume $G = (V, E)$ to be the graph formed by edge set $E$ on vertex set $V$.

- Often, will take $|V| = n$ to be fixed, and assume elements of $V$ to be uniquely identified.
- $E$ may be random, in which case $G = (V, E)$ is a random graph.
- Adjacency matrix $Y \in \{0, 1\}^{N \times N}$ (may also be random); for $G$ random, will use notation $y$ for adjacency matrix of realization $g$ of $G$.
- Graph/adjacency matrix sets denoted by $\mathcal{G}, \mathcal{Y}$; set of all graphs/adjacency matrices of order $n$ denoted $\mathcal{G}_n, \mathcal{Y}_n$.

Additional matrix notation:
- $y_{ij}^+, y_{ij}^-$ denote matrix $y$ with $i, j$ cell set to 1 or 0 (respectively).
- $y_{ij}^c$ denotes all cells of matrix $y$ other than $y_{ij}$.
- Can be applied to random matrices, as well.
Reminder: Exponential Families for Random Graphs

Let $G$ be a random graph w/countable support $\mathcal{G}$, represented through its random adjacency matrix $Y$ on corresponding support $\mathcal{Y}$. The pmf of $Y$ is then given in ERG form by

$$
\Pr(Y = y | t, \theta) = \frac{\exp(\theta^T t(y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta^T t(y'))} I_{\mathcal{Y}}(y) \tag{1}
$$

- $\theta^T t$: linear predictor
  - $t : \mathcal{Y} \to \mathbb{R}^m$: vector of sufficient statistics
  - $\theta \in \mathbb{R}^m$: vector of parameters
  - $\sum_{y' \in \mathcal{Y}} \exp(\theta^T t(y'))$: normalizing factor (aka partition function, $Z$)

- Intuition: ERG places more/less weight on structures with certain features, as determined by $t$ and $\theta$
  - Model is complete for pmfs on $\mathcal{G}$, few constraints on $t$
Approximate ERG Simulation via the Gibbs Sampler

- Direct simulation is infeasible due to incomputable normalizing factor
- Approximate solution: single update Gibbs sampler (Snijders, 2002)
  - Define $\Delta_{ij}(y) = t(y_{ij}^+) - t(y_{ij}^-)$; it follows that
    \[
    \Pr(Y_{ij} = 1 \mid y_{ij}^c, t, \theta) = \frac{1}{1 + \exp(-\theta^T \Delta_{ij}(y))} \quad (2)
    \]
    \[
    = \logit^{-1}(\theta^T \Delta_{ij}(y)) \quad (3)
    \]
  - Let sequence $Y^{(1)}, Y^{(2)}, \ldots$ be formed by identifying a vertex pair $\{i, j\}$ (directed case: $(i, j)$) at each step, and letting $Y^{(i)} = (Y^{(i-1)})^+_{ij}$ with probability given by Equation 3 and $Y^{(i)} = (Y^{(i-1)})^-_{ij}$ otherwise
  - Under mild regularity conditions, $Y^{(1)}, Y^{(2)}, \ldots$ forms an ergodic Markov chain with equilibrium pmf $\text{ERG}(\theta, t, \mathcal{V})$
- Better MCMC algorithms exist, but most are similar – this one will be of use to us later
Avoiding Approximation: “Exact” Sampling Schemes

► General goal: obtaining draws which are “exactly” iid with a given pmf/pdf
  ▶ Obviously, this only works up to the limits of one’s numerical capabilities (and often approximate uniform RNG); thus some call this “perfect” rather than “exact” sampling

► Many standard methods for simple problems (e.g., inverse CDF, rejection), but performance unacceptable on most complex problems

► Ingenious scheme from Propp and Wilson (1996) called “Coupling From The Past” (CFTP)
  ▶ Builds on MCMC in a general way
  ▶ Applicable to complex, high-dimensional problems
Coupling from the Past

The scheme, in a nutshell:

- Start with a Markov chain $Y$ on support $S$ w/equilibrium distribution $f$
- Designate some (arbitrary) point as iteration 0 (w/state $Y^{(0)}$)
- Consider some (also arbitrary) iteration $-i < 0$, and define the function $X_0(y)$ to be the (random) state of $Y^{(0)}$ in the evolution of $Y^{(-i)}, Y^{(-i+1)}, \ldots, Y^{(0)}$, with initial condition $Y^{(-i)} = y$
- If the above evolution has common $X_0(y) = y^{(0)}$ for all $y \in S$ (holding constant the “random component,” aka *coupling*), then $y^{(0)}$ would result from any (infinite) history of $Y$ prior to $-i$
- Since 0 was chosen independently of $Y$, $y^{(0)}$ is a random draw from an infinite realization of $Y$, and hence from $f$
- If this fails, we can go further into the past and try again (keeping the same coupling as before); if $Y$ is ergodic, this will work a.s. (eventually)
Coalescence Detection

Sounds too good to be true! What’s the catch?

The problem is coalescence detection: how do we know when $X_0(y)$ would have converged over all $y \in S$?

- Could run forward from all elements in $S$, but this is worse than brute force!
- Need a clever way to detect coalescence while simulating only a small number of chains

Conventional solution: try to find a monotone chain

- Let $\leq$ be a partial order on $S$, and let $s_h, s_l \in S$ be unique maximum, minimum elements
- Define a Markov chain, $Y$, on $S$ w/transition function $\phi$ based on random variable $U$ such that $s \leq s'$ implies $\phi(s|U = u) \leq \phi(s'|U = u)$; then $Y$ is said to be a monotone chain on $S$

If $Y$ is monotone, then we need only check that $X_0(s_h) = X_0(s_l)$, since any other state will be “sandwiched” between the respective chains

- Remember that we are holding $U$ constant here!
Back to ERGs

This is lovely, but of little direct use to us

- Typical ERG chains aren't monotone, and none have been found which are usable
  - I came up with one (the "digit value sampler"), but it's worse than brute force....

Alternate idea: create two “bounding chains” which stochastically dominate/are dominated by a “target chain” on $\mathcal{Y}$ (with respect to some partial order)

- Target chain is an MCMC with desired equilibrium
- “Upper” chain dominates target, “lower” chain is dominated by target (to which both are coupled)
- Upper and lower chains started on maximum/minimum elements of $\mathcal{Y}$; if they meet, then they necessarily “sandwich” all past histories of the target (and hence the target has coalesced)
  - Similar to dominated CFTP (Kendall, 1997; Kendall and Møller, 2000) (aka “Coupling Into and From The Past”), but we don’t use the bounding chains for coupling in the same way

Of course, we now need a partial order, and a bounding process....
The Subgraph Relation

- Given graphs $G, H$, $G$ is a subgraph of $H$ (denoted $G \subseteq H$) if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$

  - If $y$ and $y'$ are the adjacency matrices of $G$ and $H$, $G \subseteq H$ implies $y_{ij} \leq y'_{ij}$ for all $i, j$
  - We use $y \subseteq y'$ to denote this condition

- $\subseteq$ forms a partial order on any $\mathcal{Y}$

  - For $\mathcal{Y}_n$, we also have unique maximum element $K_n$ (complete graph) and minimum element $N_n$ (null graph)
Bounding Processes

► Let $Y$ be a single-update Gibbs sampler w/equilibrium distribution $\text{ERG}(\theta, t, \mathcal{Y}_n)$; we want processes $(L, U)$ such that $L^{(i)} \subseteq Y^{(i)} \subseteq U^{(i)}$ for all $i \geq 0$ and for all realizations of $Y$

▷ Define change score functions $\Delta^L$ and $\Delta^U$ on $\theta$ and graph set $\mathcal{A}$ as follows:

\[
\Delta^L_{ijk} (\mathcal{A}, \theta) = \begin{cases} 
\max_{y \in \mathcal{A}} \Delta_{ijk}(y) & \theta_k \leq 0 \\
\min_{y \in \mathcal{A}} \Delta_{ijk}(y) & \theta_k > 0 
\end{cases} 
\]  

(4)

\[
\Delta^U_{ijk} (\mathcal{A}, \theta) = \begin{cases} 
\min_{y \in \mathcal{A}} \Delta_{ijk}(y) & \theta_k \leq 0 \\
\max_{y \in \mathcal{A}} \Delta_{ijk}(y) & \theta_k > 0 
\end{cases} 
\]  

(5)

◊ Intuition: $\Delta^L_{ij}$ biased towards “downward” transitions, $\Delta^U_{ij}$ biased towards “upward” transitions
Bounding Processes, Cont.

Assume that, for some given \( i \), \( L^{(i)} \subseteq Y^{(i)} \subseteq U^{(i)} \), and let

\[
\mathcal{B}^{(i)} = \{ y \in \mathcal{Y}_n : L^{(i)} \subseteq y \subseteq U^{(i)} \}
\]

be the set of adjacency matrices bounded by \( U \) and \( L \) at \( i \).

Assume that edge states determined by \( u^{(0)}, u^{(1)}, \ldots, w/u^{(i)} \) iid uniform on \([0, 1]\).

Bounding processes then evolve by (for some choice of \( j, k \) to update)

\[
L^{(i+1)} = \begin{cases} 
(L^{(i)})^+_j & u^{(i)} \leq \text{logit}^{-1} \left( \theta^T \Delta_L^{jk} \left( \mathcal{B}^{(i)}, \theta \right) \right) \\
(L^{(i)})^-_j & u^{(i)} > \text{logit}^{-1} \left( \theta^T \Delta_L^{jk} \left( \mathcal{B}^{(i)}, \theta \right) \right)
\end{cases}
\]

(6)

\[
U^{(i+1)} = \begin{cases} 
(U^{(i)})^+_j & u^{(i)} \leq \text{logit}^{-1} \left( \theta^T \Delta_U^{jk} \left( \mathcal{B}^{(i)}, \theta \right) \right) \\
(U^{(i)})^-_j & u^{(i)} > \text{logit}^{-1} \left( \theta^T \Delta_U^{jk} \left( \mathcal{B}^{(i)}, \theta \right) \right)
\end{cases}
\]

(7)

Intuition: \( \Pr \left( U^{(i+1)}_{jk} = 1 \right) \geq \Pr \left( Y^{(i+1)}_{jk} = 1 \right) \geq \Pr \left( L^{(i+1)}_{jk} = 1 \right) \), by construction of \( \Delta_U, \Delta_L \)
Bounding Processes, Cont.

- We can now put the pieces together:
  - If, at iteration $i$, $L^{(i)} \subseteq Y^{(i)} \subseteq U^{(i)}$, then $L^{(i+1)} \subseteq Y^{(i+1)} \subseteq U^{(i+1)}$
    - True because, for any choice of edge to update (across all three processes), an edge is added to $Y$ only if it is also added to $U$, and an edge is removed from $Y$ only if it is also removed from $L$
    - By construction of $\Delta U, \Delta L$, this holds regardless of the current state of $Y$
  - Since $N_n \subseteq Y^{(i)} \subseteq K_n$, we can guarantee the above for some fixed iteration 0 by setting $L^{(0)} = N_n, U^{(0)} = K_n$; then, by induction, the condition holds for all $i \geq 0$
  - Let us assume that, at some iteration $i > 0$, $L^{(i)} = U^{(i)}$. Then clearly $L^{(i)} = Y^{(i)} = U^{(i)}$, regardless of $Y^{(0)}$; this implies that $Y$ has coalesced
    - Moreover, this will occur in finite expected time if $\theta^T \Delta$ (and hence $\theta^T \Delta U, \theta^T \Delta L$) is finite
Perfect Sampling for ERGs

Given the bounding processes, our approach is now straightforward:

1. Choose iteration $-i$, set $L^{-i} = N_n, U^{(i)} = K_n$
2. Evolve $U, L$ forward until coalescence detected, or 0 reached
3. If 0 reached, let $i := -2i$ (or the like), and start over (keeping the same values of $u$ and edge update choices for iterations $-i, \ldots, 0$)
4. Otherwise, set $Y^{(-j)} := L^{(-j)}$ (for coalescence point $-j$) and simulate $Y$ forward until iteration 0
5. Return $Y^{(0)}$, which is distributed ERG($\theta, t, \mathcal{Y}_n$)

"Geometric backing-off" based on binary search argument (Propp and Wilson, 1996)

Convergence time no faster than mixing speed of $Y$ ( alas), and can be slower

- Takes at least $N^2$ steps, but this is better than $2^{N^2}$...
Wait a minute – what about computation for $\Delta^U$ and $\Delta^L$?

They depend upon $B^{(i)} = \{y \in Y_n : L^{(i)} \subseteq y \subseteq U^{(i)}\}$, which is equal to $Y_n$ for at least one iteration.

If direct computation were feasible, we wouldn’t need this algorithm in the first place!

More bounding arguments:

Good: assume $t$ such that $t_i(Y) \leq t_i(Y')$ for all $Y \subseteq Y'$ (i.e., the elements of $t$ are weakly monotone increasing in edge addition). Then $\max_{y \in B^{(i)}} \Delta_{jk}(y) \leq t\left(U^+_{jk}\right) - t\left(L^-_{jk}\right)$, and $\min_{y \in B^{(i)}} \Delta_{jk}(y) \geq 0$.

Better: assume $t$ such that $\delta$ is weakly monotone increasing in edge addition. Then $\max_{y \in B^{(i)}} \Delta_{jk}(y) \leq t\left(U^+_{jk}\right) - t\left(U^-_{jk}\right)$ and $\min_{y \in B^{(i)}} \Delta_{jk}(y) \geq t\left(L^+_{jk}\right) - t\left(L^-_{jk}\right)$.

This is true for all subgraph census statistics, so e.g. everything arising from Hammersley-Clifford (Besag, 1974) (including curved families defined thereon) is covered...
Aside: Subgraph Census Bounds

Why do these bounds work for all subgraph census statistics?

- Let $t$ count copies of $H$, and let $\mathcal{H}_{ij}$ be the set of “edge-missing preconditions” for $H$ (i.e., $\{H' : \{H' \cup (i, j)\} \simeq H\}$).

- Clearly, $\Delta_{ij}(y) = |\{\mathcal{H}_{ij} \subseteq G\}|$, for $G$ having adjacency matrix $y_{ij}$.

- Since adding non-$ij$ edges to $y$ cannot decrease $|\mathcal{H}_{ij}|$, it follows that $\Delta_{ij}(y) \leq \Delta_{ij}(y')$ for all $y \subseteq y'$. 

\[ \Delta = 1 \]
\[ \Delta = 3 \]
\[ \Delta = 4 \]
Numerical Example: Two-star and Triangle Models

Pr(Y ∈ {K, N}|θ), Two-Star Model

Log Mean Coalescence Time, Two-Star Model

Pr(Y ∈ {K, N}|θ), Triangle Model

Log Mean Coalescence Time, Triangle Model
Numerical Example, Cont.
Numerical Example, Cont.

Mean $K_{1,2}$ Count, Triangle Model

StdDev $K_{1,2}$ Count, Triangle Model

Mean $K_3$ Triangle Model

StdDev $K_3$ Count, Triangle Model
Summary

- Exact/perfect sampling for ERGs is feasible in at least some cases

- Basic approach: modified CFTP
  - Start with single-edge update Gibbs sampler
  - Detect coalescence via coupled bounding processes that “sandwich” Gibbs states
  - Changescores for bounding processes can be themselves bounded using subgraph relations

- Algorithm can be slow, but does work
  - Has trouble when bounds are loose, or when underlying sampler mixes poorly
  - On bright side, you know when it’s not working (unlike MCMC)
Open Problems

► Tighter linear predictor bounds
  ▶ Per-element bounds are best possible (for subgraph census case, at least), but bounds on the linear predictor can be much tighter (big problem for curved models)
  ▶ Have gotten better results with pre-computation for degree, but very expensive (one-time $O(N^4)$ cost)

► Escape from the single-update Gibbs
  ▶ Not clear that one can do much else, but worth further thought
  ▶ Can something akin to TNT be done by looking at edge states which unequivocally present or absent (using the bounding chains)?

► More exotic algorithms
  ▶ Is there another way of doing this? I don’t know of anything substantially faster than CFTP, but that doesn’t mean it’s not out there....
1 References


