Published in final edited form as:

J Math Sociol. 2015 ; 39(3): 174–202. doi:10.1080/0022250X.2015.1022279.

A Novel Simulation Method for Binary Discrete Exponential Families, with Application to Social Networks¹

Carter T. Butts²

²Departments of Sociology, Statistics, and EECS, and Institute for Mathematical Behavioral Sciences; University of California, Irvine; SSPA 2145; Irvine, CA 92697-5100; buttsc@uci.edu

Abstract

Stochastic models for finite binary vectors are widely used in sociology, with examples ranging from social influence models on dichotomous behaviors or attitudes to models for random graphs. Exact sampling for such models is difficult in the presence of dependence, leading to the use of Markov chain Monte Carlo (MCMC) as an approximation technique. While often effective, MCMC methods have variable execution time, and the quality of the resulting draws can be difficult to assess. Here, we present a novel alternative method for approximate sampling from binary discrete exponential families having fixed execution time and well-defined quality guarantees. We demonstrate the use of this sampling procedure in the context of random graph generation, with an application to the simulation of a large-scale social network using both geographical covariates and dyadic dependence mechanisms.

Keywords

statistical simulation; discrete exponential families; random graphs

1 Introduction

Stochastic models for finite binary vectors are widely used in sociology, with examples ranging from social influence models on dichotomous behaviors or attitudes (Coleman, 1964; Latané, 1996; Holyst et al., 2001; Robins et al., 2001) to models for random graphs (Holland and Leinhardt, 1981; Skvoretz et al., 2004; Robins et al., 2005; Wasserman and Robins, 2005). Exact sampling for such models is often quite difficult in the presence of dependence, generally leading to the use of Markov chain Monte Carlo (MCMC) as an approximation technique (Snijders, 2002; Hunter et al., 2008). While often effective, MCMC methods have variable execution time (i.e., time to convergence), can be slow to converge (Bhamidi et al., 2011), and produce draws whose quality can be difficult to assess (Gelman, 1996). Here, we present an alternative method for approximate sampling from binary discrete exponential families having fixed execution time and well-defined quality guarantees. In many cases, worst-case complexity of the sampling process is no worse than the worst-case complexity of an exact sampler for independent, inhomogeneous binary

¹This research was supported in part by ONR award N00014-08-1-1015, NIH award 1R01HD068395-01, and NSF awards BCS-0827027 and IIS-1251267.

variables (i.e., order N , where N is the number of variables). We demonstrate the use of this sampling procedure in the context of random graph generation, with application to simulation of exponential family random graph models (ERGMs) fit to network data, and to the exploratory simulation of large-scale social networks using both geographical covariates and dyadic dependence mechanisms.

1.1 Preliminaries

Let $Y = Y_1, \dots, Y_N$ be a finite vector of binary-valued random variables with joint support $\mathcal{Y}_N$. In a social network context, each Y_i represents an indicator for the presence of an edge between two vertices (i.e., Y is a vectorized adjacency matrix); in other settings, Y_i might indicate e.g., an individual's agreement or disagreement with an item on a survey instrument, the presence or absence of an individual or organization at an event, or an individual's decision to engage in an observable behavior at some point in time. Without loss of generality, we may write the joint distribution of Y in discrete exponential family form as

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

where $t: \mathcal{Y}_N \mapsto \mathbb{R}^p$ is a vector of sufficient statistics, $\theta \in \mathbb{R}^p$ is a parameter vector, and $I_{\mathcal{Y}_N}$ is an indicator function for membership in the support. Note that we may also write Y 's joint distribution in a very different, hierarchical form:

$$Pr(Y=y|\theta) = Pr(Y_1=y_1|\theta) Pr(Y_2=y_2|\theta, Y_1=y_1) \dots \times Pr(Y_N=y_N|\theta, Y_1=y_1, \dots, Y_{N-1}=y_{N-1}). \quad (2)$$

With the exception of relatively trivial cases, neither Equation 1 nor Equation 2 is of much direct use: direct calculation of the normalizing factor of the former is generally prohibitive, and the partially marginalized distributions of the latter (which involve similar normalizing factors) are generally unknown. Standard approaches to simulating Y in the general case typically employ Markov chain Monte Carlo methods, exploiting the fact that

$$\frac{Pr(Y=y'|\theta)}{Pr(Y=y|\theta)} = \exp[\theta^T [t(y') - t(y)]], \quad (3)$$

and hence that calculations depending only on probability ratios for alternative realizations of Y (as required, e.g., for Metropolis-Hastings algorithms) are often computationally inexpensive. (See, e.g. Morris et al. (2008) in the ERGM case.) Our approach in the present paper utilizes a different (but closely related) property of Equation 1, namely that

$$Pr(Y_i=1|\theta, Y_i^c=y_i^c) = [1 + \exp[\theta^T [t(y_i^-) - t(y_i^+)]]]^{-1}, \quad (4)$$

where Y_i^c refers to all elements of Y other than the i th, Y_i^+ refers to the vector Y such that the i th entry is set to 1, and Y_i^- refers to the same vector with the i th entry set to 0. (We have

here suppressed the support term for clarity of notation: $Pr(Y_i=1|\theta, Y_i^c=y_i^c)$ is obviously equal to 0 $I_{\mathcal{Y}_N}(y_i^+)=0$, and to 1 if $I_{\mathcal{Y}_N}(y_i^-)=0$.)

To see how we may leverage this property, let us consider Equation 2. While we cannot in general compute the marginal elements of this expression (the i th of which is a sum over a set of states that grows exponentially as 2^{N-i}), we can *bound* them using the full conditionals of Equation 4. Specifically, let $Y_{<i}^c$ refer to the elements $Y_1, \dots, Y_{i-1}$ of Y . By definition,

$$Pr(Y_i=1|\theta, Y_{<i}^c=y_{<i}^c) = \sum_{y' \in \mathcal{Y}_N: y'_{<i} = y_{<i}^c} Pr(Y_i=1|\theta, Y_i^c=y_i^c) Pr(Y_i^c=y_i^c|\theta, Y_{<i}^c=y_{<i}^c). \quad (5)$$

The marginal probability that Y_i will take state 1 (versus 0) given the states of the previous elements is thus a convex combination of the full conditionals for $Y_i = 1$ over all combinations of subsequent states. Let us define two vectors of extremal full conditionals, α and γ , such that $\alpha_i = \min_{y' \in \mathcal{Y}_N} Pr(Y_i=1|\theta, Y_i^c=y_i^c)$ and $\gamma_i = \max_{y' \in \mathcal{Y}_N} Pr(Y_i=1|\theta, Y_i^c=y_i^c)$. For simplicity in notation, let us similarly define $\beta_i \equiv Pr(Y_i=1|\theta, Y_{<i}^c=y_{<i}^c)$. From the convexity of the sum in Equation 5, it follows immediately that $\alpha_i \leq \beta_i \leq \gamma_i$, and that the vectors extrema thus bound the vector of marginal state 1 probabilities.¹

This last observation has been used in prior work by Butts (2011) to provide bounds on the behavior of random graph models. It can also be used, however, as the basis for a novel approximate sampling scheme for any finite binary vector Y that differs considerably from the standard MCMC approach. We now turn to a description of this algorithm.

2 Sampling Method

We begin by describing the approach in intuitive form, subsequently turning to the issue of efficient implementation. Let $R = R_1, \dots, R_N$ be a vector of iid uniform random variables on the unit interval. Equation 2 suggests an obvious (exact) sampling scheme for Y : starting with Y_1 and progressing in order $Y_2, Y_3, \dots, Y_i$, set $y_i = 1$ if $R_i < Pr(Y_i=1|\theta, Y_{<i}^c=y_{<i}^c) = \beta_i$, and 0 otherwise. This, of course, requires that we know the corresponding marginal distributions, which as noted are incomputable in the general case. In some circumstances, however, such knowledge may be unnecessary. Consider, for instance, the case in which (for some given realization r of R) $r_i < \alpha_i$. Since the corresponding marginal probability that $Y_i = 1$ is greater than or equal to α_i , it follows that y_i for this realization must be equal to 1. Likewise, if $r_i \geq \gamma_i$, y_i must be equal to 0 (since we know that the probability that $Y_i = 1$ is at most γ_i). Even where we cannot pin down β exactly, it may therefore be possible to “fix” many values of y in any given realization simply through the α, γ bounds. Furthermore, the fixation of this first collection of y values may – by allowing refinement of the bounds associated with the remaining variables – fix yet more values. In this way, the initial fixation

¹It should be borne in mind that β is a random vector, unlike the fixed vectors of α and γ . However, each β_i depends only on the outcomes preceding it, and is bounded by α_i and γ_i regardless of those outcomes.

process can spawn a “fixation cascade” that may result in many (or occasionally all) of the elements of y being exactly determinable. Of course, it may also be the case that this process terminates with some elements remaining unfixed; at this point, some intervention is required. The expedient we propose in this case is to approximate the first unfixed β_i by a uniform draw between the (updated) values of α_i and γ_i , fixing y_i to 1 or 0 if r_i is less than or greater than/equal to the approximation (respectively). Any fixation cascade created by this operation is then resolved, and the process is repeated until no variables remain unfixed. While the result of this process is not (in general) an exact draw from Y , it may prove to be a close approximation (especially if the initial fixation cascade covers most of the elements of Y). Unlike MCMC-based methods, the worst-case execution time for this algorithm is fixed and frequently linear in N , with additional performance enhancements available in some circumstances (as discussed below). Also unlike MCMC algorithms, this sampling method leads to a well-defined bound on the quality of the resulting draw. We return to this point in Section 3.

2.1 Algorithm

Our base algorithm is fairly straightforward: we advance through the elements of y , fixing each element as we proceed. (The order in which elements of y are visited is arbitrary, a fact that we exploit in section 2.2.1.) This process is shown in detail in Algorithm 1. Effectively, the operations performed when setting each y_i consist of three steps: *initialization*, *fixation*, and *perturbation*. In the initialization step (lines 2-4), we compute an updated pair of bounds

$(\alpha'_i \geq \alpha_i \text{ and } \gamma'_i \leq \gamma_i)$ for β_i , and draw the random input r_i . The bound pair (α'_i, γ'_i) is a refinement of the unconditional bounds (α_i, γ_i) , reflecting the fact that the elements $y_{<i}$ of y are known by the i th iteration. This is the mechanism by which the “fixation cascade” described above is implicitly employed: previously encountered elements of y fixed by their r values have the potential to narrow the interval between α' and γ' , thereby increasing the chance that the current element will itself be fixable. (Of course, fixation may also occur due to the impact of perturbed elements; we return to this issue in Section 3.)

After initialization, we proceed to the fixation step (lines 5-8). Here, we exploit the fact that $\alpha'_i \leq \beta_i \leq \gamma'_i$ to immediately recognize that $y_i | y_{<i}$ is necessarily equal to 1 if $r_i < \alpha'_i$ and equal to 0 if $r_i \geq \gamma'_i$, setting y_i accordingly. In some cases, of course, fixation fails: if $\alpha'_i \leq r_i < \gamma'_i$, we cannot say where β_i lies, and thus cannot set y_i . We then proceed to the perturbation step (lines 10-14), drawing an estimated threshold $\tau \sim U(\alpha'_i, \gamma'_i)$ and setting y_i accordingly. Having set the value of y_i (either immediately or following perturbation), we then proceed to the next element of y , repeating this process until all elements have been set.

As we visit each element of y once, the complexity of Algorithm 1 is clearly $\mathcal{O}(Nf(N))$, where $f(N)$ is the worst-case complexity of the initialization step (i.e., determining the (α'_i, γ'_i) bounds). In many cases (including network models based on degree, reciprocity, and mixing terms), this step can be performed in constant time, leading to a worst-case complexity that is linear in the number of variables. As this is also the worst-case complexity for simulation of N inhomogeneous Bernoulli trials, Algorithm 1 can clearly

achieve optimal performance in such settings. Also, unlike MCMC algorithms (which must be run for an unknown period to obtain convergence), the execution time for Algorithm 1 is fixed and independent of the distributional properties of Y . These are both attractive features for applications requiring user-unsupervised sampling (e.g., normalizing factor computation), wherein convergence assessment is difficult and predictable execution time is highly desirable.

2.2 Refinements

Various modifications and/or refinements of Algorithm 1 are possible, either to enhance speed or to improve the quality of the resulting draws. Here, we discuss several such refinements, before turning to the problem of quality assessment.

2.2.1 Fixation Probability Sorting—One observation that can prove useful when constructing refinements to the sampling process of Algorithm 1 is the fact that the order in which the elements of Y are given in Equation 2 (and hence the sampling algorithm) is arbitrary; we can thus re-order the sequence in which we visit the elements of Y within the sampling algorithm, so long as our choice does not depend on the realizations of R or previous elements of Y . A straightforward application of this idea can be used for quality improvement when the elements of Y are heterogeneous, with some close *ex ante* to fixation. Specifically, we modify Algorithm 1 by adding an *ordering step* prior to the main loop, in which we sort the elements of Y in ascending order by $\gamma - \alpha$. This results in an algorithm that begins by visiting the elements of Y most likely to be fixed, increasing the probability that the state of later elements will be decided by Y states assigned through a fixation cascade involving only R -fixed elements, rather than by a cascade “contaminated” by perturbation. In the general case, this results in a total complexity $\mathcal{O}(Nf(N) + N \log N)$ for the sampling algorithm; where $f(N)$ grows more slowly than $\log N$, the pre-sorting operation will dominate the worst-case complexity of the sampler. Where $f(N)$ grows more quickly, by contrast, the complexity of the pre-sort is on the same order as the complexity of the sampler itself, and cost is not a substantial consideration in its use.

2.2.2 Tacit Evaluation—Another family of refinements that can be employed when most elements of α are large (close to 1) or most elements of γ are small (close to 0) is to take the elements of y equal to 1 (respectively 0) by default, and explicitly evaluate only elements at high risk of taking the alternate state. (This is a technique used in the simulation of sparse Bernoulli graphs, where the expected number of Y_i equal to 1 is often $\mathcal{O}(\sqrt{N})$; see, e.g. Batagelj and Brandes (2005).) Consider, for example, the “sparse” case in which all $\gamma_i \ll \rho$ for some $\rho \ll 0.5$. We may then modify Algorithm 1 as follows. Rather than evaluating each y_i explicitly, we begin by drawing $k \sim \text{Geom}(\rho)$ (where Geom is the geometric distribution), setting $i = k + 1$, and treating all $y_{<i}$ as 0. We then explicitly draw $r_i \sim U(0, \rho)$ and evaluate y_i as usual. Once y_i has been set we draw a new k from the same distribution, set $i = i + k$, and update the new y_i (again treating all intervening elements of y as 0, and taking $r_i \sim U(0, \rho)$). This process is repeated until all elements of y have been set. Since updating y cannot increase any element of γ , it is not necessary to change ρ during execution; although probabilistically identical to Algorithm 1, this modified procedure will reduce the number of

element update executions by an expected factor of $1 - \rho$. (A directly analogous procedure can be used for the large α case.) While the efficiency gain by this modification is constant in ρ , this can be consequential where ρ itself varies with N . The above-cited case of sparse graphs provides one example of such a circumstance, but other cases with similar properties (e.g., grid-based point process approximations) occur in a number of fields.

2.2.3 Improved Threshold Simulation Techniques—As noted in Section 2, when fixation fails we must approximate the true probability that $Y_i = 1$, β_i , in order to carry out the perturbation process. Intuitively, setting τ (in the notation of Algorithm 1) equal to β_i will produce an exact draw from the sampler; although we do not know the values of β , we may still hope for a more intelligent approximation to the uniform draw of Algorithm 1. Here, we describe two approaches to this problem, both of which can improve performance in practice.

“Pseudo-Bayesian” Approximation: Our default method shown in Algorithm 1 is equivalent to treating β_i as entirely unknown, save for the fact that it lies within the interval (α'_i, γ'_i) . In practice, however, we are not entirely ignorant regarding β . For instance,

consider the case in which the “1-density” of Y , $\delta \equiv \mathbf{E} \sum_i Y_i / N$, is exactly or approximately known. Clearly, for a randomly selected element Y_i of Y , $\mathbf{E} \beta_i = \delta$; thus, we have in this case some “*a priori*” knowledge about β that could in principle be used to aid simulation.

To formalize this idea, let the pmf f represent our “prior” beliefs about an arbitrary β_i . (We will here assume that our “priors” are exchangeable; the term is used in quotation marks to emphasize that our use here is heuristic, and not assumed to satisfy the criteria for a proper Bayesian analysis.) Imagine that we subsequently learn that $\alpha'_i \leq \beta_i \leq \gamma'_i$. The corresponding “posterior” density for β_i is then

$$\begin{aligned} p(\beta_i = z | \alpha'_i, \gamma'_i) &\propto I_{\mathbb{R} \geq \alpha'_i}(z) I_{\mathbb{R} \leq \gamma'_i}(z) f(z) \\ &= \frac{f(z)}{\int_{\alpha'_i}^{\gamma'_i} f(x) dx} I_{x \in \mathbb{R}: \alpha'_i \leq x \leq \gamma'_i}(z) \\ &= \frac{f(z)}{F(\gamma'_i) - F(\alpha'_i)} I_{x \in \mathbb{R}: \alpha'_i \leq x \leq \gamma'_i}(z), \end{aligned}$$

where I is defined as before as a set indicator function, and F is the prior cdf of β_i . For random input r_i , then, the chance of setting $Y_i = 1$ corresponds to

$$Pr(r_i < \beta_i | \alpha'_i, \gamma'_i) = \begin{cases} 1 & \text{if } r_i < \alpha'_i \\ \frac{F(r_i) - F(\alpha'_i)}{F(\gamma'_i) - F(\alpha'_i)} & \text{if } \alpha'_i \leq r_i < \gamma'_i \\ 0 & \text{if } r_i \geq \gamma'_i \end{cases},$$

with the middle term being equal to $\int_{\alpha'_i}^{r_i} p(\beta_i = z | \alpha'_i, \gamma'_i) dz$, as derived above.

Taking f to be the uniform distribution on $(0, 1)$ gives the default perturbation method. However, there is no reason that we must employ this distribution. For instance, a natural alternative choice would be $\text{Beta}(\delta k, (1 - \delta)k)$, where δ is the “1-density” (or an approximation thereto) and k is a user-selected dispersion parameter. In a conventional Bayesian setting, $k = 2$ provides a prior weight equal to that of a uniform distribution, with $k = 1$ being the weight of the Jeffreys prior for a binomial likelihood. (Specifically, both of these distributions can be realized by the appropriate choice of k for $\delta = 0.5$.) Here, our selection is heuristic, and should be used to produce threshold distributions that have reasonable behavior when the bounding method fails. In the absence of additional information, small k (e.g., ≈ 1) with δ chosen as above leads to a distribution that is relatively flat in its middle range, but otherwise weighted heavily towards extreme thresholds (in the direction of δ). For models that combine low/high baseline probabilities of $Y_i = 1$ with (respectively) strong excitatory/inhibitory dependence effects, the resulting threshold probabilities are likely to be much better approximations of the true β elements than would be obtained by the default method.

As a final comment on this technique, it should be noted that additional information on the distribution of β can be gleaned in settings for which some observation from Y (or an approximation thereto) is available. A typical case is that in which $Y|\theta$ is chosen as a data model for observation y , which (to the extent that the model is adequate) can then be viewed as an approximate draw from the random process. In this latter case, Equation 4 gives a direct method for evaluating the full conditionals for a single Y element, given the rest of the realized process. Although the full conditionals for a single realization are not equal to the true β values for that realization, they nevertheless provide a rough approximation thereto. (This is, indeed, the basis for the pseudo-likelihood approximation of Besag (1974).) Inspection of the properties of the full conditional distribution may thus prove useful in choosing f , e.g., by suggesting good approximate δ values and/or choices for k in the Beta case, and/or alternative distributional forms. Since f is only used when the fixation process fails, its selection will have little impact for models requiring little perturbation. For less constrained models, however, intelligent approximation of β may greatly aid performance. Fortunately, one virtue of the sampling strategy employed here is that we may predict this degree of constraint on an *ex ante* basis. We turn to a discussion of this problem in Section 3.

Pseudo-Marginalization: As shown in Equation 5, the marginal probability

$Pr(Y_i = 1 | \theta, Y_{<i}^c = y_{<i}^c)$ is a convex combination of the full conditionals of Y_i , mixed over the possible realizations of $Y_{i+1}, \dots, Y_N$. Direct computation of this quantity is infeasible, but in some cases it may be possible to approximate it via a process of *pseudo-marginalization*.

The general principle is as follows. Imagine that, via an exact simulation, we were to obtain a set of draws $y^{(1)}, y^{(2)}, \dots, y^{(m)}$ from Y . Plainly, a consistent and unbiased estimator of

$Pr(Y_i = 1 | \theta, Y_{<i}^c = y_{<i}^c)$ is

$$\hat{\mu}_i(y_{<i}^c) = \frac{\sum_{j=1}^m \Pr(Y_i=1|\theta, Y_i^c=(y^{(j)})_i^c) I_{\{y' \in \mathcal{Y}_N: y'_{<i}^c=y_{<i}^c\}}(y^{(j)})}{\sum_{j=1}^m I_{\{y' \in \mathcal{Y}_N: y'_{<i}^c=y_{<i}^c\}}(y^{(j)})}, \quad (6)$$

with convergence occurring in the limit as $m \rightarrow \infty$. While we do not have an exact sample from Y , we may instead consider a set of draws from some random variable Y' whose distribution is close to Y ; we refer to the calculation of $\hat{\mu}$ under the resulting sample as “pseudo-marginalization.” Intuitively, if Y' is a reasonable approximation to Y (and/or if $\hat{\mu}_i(y_{<i}^c)$ is relatively insensitive to $y_{<i}^c$), then the pseudo-marginalized estimator of $\Pr(Y_i=1|\theta, Y_{<i}^c=y_{<i}^c)$ (hence β_i) may be fairly close to the true value. Using the estimated β_i values during perturbation can thus improve simulation performance, without losing quality guarantees.

The procedure for pseudo-marginalization is as follows. We begin by positing a distribution Y' , and drawing an initial *marginalization sample* $M = (y^{(1)}, \dots, y^{(m)})$, $y^{(i)} \sim Y'$. Ideally, Y' should be as close to Y as possible, while still being easily simulated. In practice, we suggest starting with an independent Bernoulli model, calibrated to preserve overall average features of Y (e.g., frequency of 1s, marginal covariate relationships, etc.). We then proceed with Algorithm 1, with the following simple modifications. Where fixation occurs (via the bounds), pseudo-marginalization is not employed. In the event that perturbation is necessary, the τ assignment of line 10 is replaced by $\tau := \hat{\mu}_i(y_{<i}^c)$ per Equation 6, with M being used as the set of sample draws. Finally, once an assignment is made to y_i (either by perturbation or fixation), we set $y^{(j)}_i = y_i$ for each $y^{(j)} \in M$. Thus, at the i th iteration, $y_{<i}^{(j)} = y_{<i}$, and the elements of M are identical to y at the conclusion of the algorithm. Otherwise, the sampling algorithm proceeds identically to Algorithm 1, and the quality assessment results of Section 3 are unaffected.

Although pseudo-marginalization does not improve the guaranteed quality of the sampler, it can in practice have a substantial effect. The total increase in computational and storage cost is proportional to m , and is thus a constant; however, this may still prove burdensome for large problems. In cases where dependence between elements is strong (especially if N is not too large), pseudo-marginalization can be a relatively “cheap” way of improving performance.

3 Quality Assessment

One useful property of the construction of Section 2 is that it provides a very direct means of assessing the quality of the approximate sample (both ex post and ex ante). To make this notion precise, define y^t to be the unique draw from Y corresponding to realization r from R (using the exact sampling method described in the introduction to Section 2). We can then consider the “quality” of a draw $y|r$ produced by our scheme in terms of its deviation from y^t — that is, we consider a draw to be of high quality when it conforms closely to what would have been obtained under the exact sampling mechanism, given the same random inputs. Let us define the quality, Q , of y by the normalized inverse Hamming distance between y and y^t

given r (i.e., $Q(y) = (N - D_H(y, y^t))/N$, where D_H is the Hamming distance). Q then varies from 0 in the worst case (i.e., all elements of y differ from the exact sample) to 1 in the best case (i.e., all elements of y are identical to the exact sample).

While we do not know $Q(y)$ exactly, we can provide a lower bound by exploiting the properties of the fixation cascade. Consider some element y_i of y , and let h_i be an indicator variable equal to 1 only if $y_i = y_i^t$ (else 0). Clearly, if $r_i < \alpha_i$ or $r_i \geq \gamma_i$, $y_i = y_i^t$; in this case, let $h_i = 1$. Moreover, let $y_{<i}^h$ denote all elements of y_j such that $j < i$ and $h_j = 1$. Define the bounds

$$\alpha_i'' = \min_{y' \in \mathcal{Y}_N: y_{<i}^h = y_{<i}^h} Pr(Y_i = 1 | \theta, Y_i^c = y_i'^c)$$

and

$$\gamma_i'' = \max_{y' \in \mathcal{Y}_N: y_{<i}^h = y_{<i}^h} Pr(Y_i = 1 | \theta, Y_i^c = y_i'^c)$$

in direct analogy to the (α_i', γ_i') pair of Algorithm 1. If $r_i < \alpha_i''$ or $r_i \geq \gamma_i''$, then we may fix the value of y_i by conditioning only on prior elements that are known to be equal to their corresponding elements in y^t . Since the set of Y realizations over which the extrema are

taken for (α_i'', γ_i'') is a superset of the set for which (α_i', γ_i') are taken (and a subset of the set for which (α_i, γ_i) are taken), it follows that we have $\alpha_i \leq \alpha_i'' \leq \alpha_i'$ and $\gamma_i \geq \gamma_i'' \geq \gamma_i'$ —thus,

in any case in which y_i would be fixed by (α_i'', γ_i'') , it will be fixed to the same value by the extrema of Algorithm 1. We may therefore set $h_i = 1$ where fixation under (α_i'', γ_i'') occurs.

By induction, this property holds for arbitrary i , and hence, for all i , $h_i = 1$ only if $y_i = y_i^t$.

Since $D_H(y, y^t) \leq N - \sum_{i=1}^N h_i$, this gives us the ex post quality bound

$$Q(y) \geq (N - \sum_{i=1}^N h_i) / N = \hat{Q}(y).$$

Although it should be noted that the fixation under (α_i'', γ_i'') required for h is stronger than the associated condition of Algorithm 1 (and hence one cannot simply set $h_i = 1$ where the latter occurs), computation of h can be performed by maintaining a separate set of bounds (i.e., α'' and γ'') and incrementing h_i when the stronger fixation condition is met. This is generally a straightforward modification of the base algorithm, and is not given here.

Ex post evaluation is useful, in that it provides us with a sense of how well the algorithm performed in a given instance. Ex ante evaluation, however, is also useful in determining whether the sampling algorithm is likely to prove successful. In this regard, note that we can place an immediate bound on the expected value of $Q(Y)$ by exploiting the initial fixation

condition: specifically, $EQ(Y) \geq \frac{1}{N} \sum_{i=1}^N (1 - \gamma_i + \alpha_i)$. This may, of course, understate the

quality of the approximation method in systems that are prone to long fixation cascades; typically, however, one would expect such systems to have fairly small $\gamma - \alpha$ intervals for most elements (since cascades are otherwise difficult to propagate), and thus to have reasonably large quality bounds. Regardless, a distribution with very high ex ante quality bounds is likely to be effectively approximated by the algorithm, and high $\mathbf{EQ}$ may thus be taken as diagnostic of situations in which use of the procedure is fairly safe.

While the above provide overall indicators of quality, it is apparent that we can do better: since we have, for each element, both the ex ante bound on its fixation ($1 - \gamma_i + \alpha_i$) and the ex post realization of its fixation state (both whether it was fixed prior to the first perturbation, and otherwise the state in which it was ultimately placed), we can construct more refined quality diagnostics that are tuned towards specific applications. For instance, we can place an upper bound on the number of elements of y mistakenly identified as 0 (or 1) by the number of i such that $y_i = 0$ (respectively $y_i = 1$) and $h_i = 0$. Similarly, the ex ante probability that a given Y_i will be *correctly* classified as a 1 is at least α_i , and the corresponding probability for correct classification as a 0 is at least $1 - \gamma_i$. More intricate diagnostics can be developed for applications such as random graphs, in which specific configurations of variable states may have particular significance.

As a final note, it must be emphasized that the ex post diagnostics should *not* be used to guide the sampling process itself (e.g., by repeatedly sampling until a high-quality draw is obtained). This is because $Q(y)$ is not in general independent of y^f , and the two may in fact be strongly related. (Consider, e.g., the case of simulating a random adjacency matrix with dependence induced by a triangle term. Extremely sparse realizations will have few potential two-paths, leading to long initial fixation cascades, and subsequently higher quality. Selecting for high quality draws will thus lead to a bias in favor of low-density, unclustered graphs – quite the opposite of what the model is intended to produce!) On the other hand, quality diagnostics can be safely used to place bounds on the possible states of y^f , to identify likely weaknesses in the approximation employed, etc. These considerations do not apply to ex ante diagnostics, which (being independent of R) are not subject to the same selection effects.

4 Evaluation and Comparison with MCMC in an Exactly Computable Case

Where the lower bound on quality, $\hat{Q}$, is high, it is evident that the bound sampler is close to exact; however, the bound sampler may still perform well even when $\hat{Q}$ is relatively low. In general, assessing performance in such cases requires comparison with high-quality MCMC runs (as shown in section 5.1). In a few cases, however, one can posit model families with edgewise dependence whose behavior can be solved exactly; in these cases, we may compare the bound sampler (and MCMC) directly to the behavior of an exact sampler.

Here we provide an example of such an analysis, using an exponential family form of the well-known $U|man$ model (Wasserman and Faust, 1994). The $U|man$ (standing for “uniform, given probabilities of mutual, asymmetric, and null dyads”) is a homogeneous random digraph family that incorporates biases in mean density and reciprocity. In exponential family form, this distribution may be written as

$$Pr(Y=y|\theta) = \frac{\exp(\theta_e t_e(y) + \theta_m t_m(y))}{[1 + 2 \exp(\theta_e) + \exp(2\theta_e + \theta_m)] \binom{n}{2}}, \quad (7)$$

where Y is a directed adjacency matrix of dimension $n \times n$, t_e is the count of edges within Y , t_m is the count of mutual dyads (i.e., node pairs i, j such that $Y_{ij} = Y_{ji} = 1$), and $\theta = (\theta_e, \theta_m)$ is a vector containing “edge” and “mutuality” parameters (respectively). Intuitively, the edge parameter corresponds to the inverse logit of the probability of an (i, j) edge conditional on there being no reciprocating (j, i) edge, while the mutuality parameter corresponds to the change in the inverse logit of this probability when a (j, i) edge is present. Although dyads are independent under $U|man$, it thus follows that *edges* still depend on one another, and this dependence can be extremely strong. Because all dependence here is governed by a single parameter (θ_m), and because the resulting model is tractable, the exponential family form of $U|man$ is a useful system for studying the impact of dependence on the bound sampler.

4.1 Fixation Rates

A first natural question to ask of the $U|man$ system is how (unperturbed) fixation probability varies with θ . Because this model is dyad independent, it suffices to consider the case of a representative i, j dyad. Per Algorithm 1, either (i, j) , (j, i) , or both may be initially fixed; if exactly one is initially fixed, then the other will necessarily be fixed by a cascade. (This last follows immediately from the fact that the probability of an (i, j) edge depends only on the state of the corresponding (j, i) edge.) It follows, then, that the probability that the (i, j) edge will be neither directly nor indirectly fixed is equal to the probability that neither (i, j) nor (j, i) is initially fixed. Let k be the position of the (i, j) edge variable in the Algorithm 1 sort order. By homogeneity, the non-fixation probability is then equal to

$$\begin{aligned} (\gamma_k - \alpha_k)^2 &= \left(\frac{1}{1 + \exp(-\theta_e - \theta_m)} - \frac{1}{1 + \exp(-\theta_e)} \right)^2 \\ &= 1 - \hat{Q} \end{aligned} \quad (8)$$

(where we have exploited the symmetry between α and γ with respect to the sign of θ_m and the square to write the two terms in the above difference in arbitrary order).

To provide an intuitive sense of how this varies in practice, Figure 1 shows $\hat{Q}$ (scaled to percentages for ease of display) as a function of θ_e and θ_m . As can be seen, fixation is near total except with respect to two “lobes” of high $|\theta_m|$, one occurring where the mutuality effect is strongly positive and the edge effect is somewhat negative and the other occurring where the mutuality effect is strongly negative and the edge effect is somewhat positive. These cases (with a strong mutuality effect that is largely balanced by the baseline density) are those that maximize the $\gamma - \alpha$ gap, and hence that minimize $\hat{Q}$.

4.2 Expected Edge, Mutual, and Triangle Counts

To get a better sense of how performance varies above and beyond what can be guaranteed by $\hat{Q}$, we here show a simple simulation study comparing mean edge, mutual, and triangle counts under the bound sampler to their $U|man$ expectations. By way of comparison to

standard methods, we also show the performance of a basic (single random edge update Metropolis) MCMC algorithm with various choices of burn-in time. For all simulation methods, we estimate the expected subgraph counts using the sample means from 1,000 independent draws; to ensure independence, each MCMC-based draw was taken from an independently run Markov chain seeded with an iid uniform graph and simulated forward for the indicated number of iterations.² Perturbation for the bound sampler was performed using the pseudo-Bayesian method of Section 2.2.3, with the pseudo-prior being a beta distribution with expectation equal to the expected density and a total concentration (i.e., parameter sum) of 6 (chosen to produce unimodality while still being diffuse). The simulation target in each case was a $U|man$ distribution for a digraph of order $n = 25$, with θ_e and θ_m each systematically varied in an 11-point sequence from -3 to 3 . The parameter range was selected to cover density levels from the nearly empty (mean degree less than 1) to the nearly complete (mean degree greater than 23), with a corresponding range in tendencies towards or away from mutuality. MCMC samples were taken from chains of length n (25), n^2 (625), and n^3 (15,625).

To assess simulation quality, we compare the mean counts for edges, mutuals, and triangles under each sampling scheme to the exact values from the corresponding $U|man$ distribution; we employ the mean absolute errors in these counts at each parameter value as our performance criterion. Figures 2-4 show the corresponding error surfaces (darker values indicate greater deviation) along with the mean absolute errors at each simulated value (rounded to the nearest integer). As a reference, the maximum numbers of edges, mutuals, and triangles for these digraphs are respectively 600, 300, and 2,300.

Figure 2 shows the error in the expected edge count (and hence density) under each method. As expected, the “gold standard” MCMC with a chain length that is long compared to the number of edges (bottom right) produces nearly flawless performance over the entire parameter space. On the other hand, the bound sampler (top left) also does quite well: it shows no or very minimal errors over much of the parameter space, with error rates that are still quite modest ($< 4\%$) in the regions of worst performance. Although error is obviously related to Q (errors cannot be large when Q is high), the sampler can still perform well when Q is lower: for instance, the region near $\theta_e \in (-2, -1)$, $\theta_m \approx 3$ (and its mirror image) has the worst quality bounds over the parameter range considered, but the simulated densities here are quite accurate. It is interesting to contrast this performance with that of Markov chains that are poorly mixed. When chain length is on the order of the number of nodes (top right panel), densities are generally far from their target values. This is not surprising (given that the chain cannot update many edges), but serves as a cautionary. Perhaps more significant is that chain lengths on the order of the number of edge variables (bottom left) are still quite insufficient to yield reasonable results. While intervals much greater than n^2 are easy to achieve for small graphs like those used here, typical default chain lengths for current ERGM MCMC tools (e.g., **ergm** v3.1) will prove inadequate for approximately $n = 100$.

Mean absolute errors for for mutuality (number of mutuals) are shown in Figure 3. The results here largely mirror those of Figure 2: very long-chain (i.e., $\gg n^2$) MCMC performs

²Simulation was performed using the **sna** (Butts, 2008) and **ergm** (Hunter et al., 2008) libraries.

flawlessly, the bound sampler works well (with modest errors over certain parts of the parameter space), and shorter-chain MCMC runs do very poorly.

Finally, we consider the propagation of simulation error into higher order structures, by way of the triangle count. As Figure 4 shows, the results in this case are not qualitatively different from those shown in the other figures: we continue to see strong performance from the bound sampler and long-chain MCMC simulations, and poor performance for shorter MCMC runs. In particular, we note that we do not see a tendency for biases in mutuality rate to propagate destructively into the triangle formation rate for the bound sampler, which continues to show error rates of 4% or less across the sampled parameter space. While the bound sampler is indeed approximate, we thus find that it here approaches the performance of high-quality MCMC simulations even for the parts of the parameter space for which is not provably near-exact.

As a final comment on these results, it should be noted that the edge/mutual model allows constant-time initialization, resulting in a worst-case time complexity of order N (in terms of graph size, n^2). The above experiments suggest that MCMC runs of substantially higher time complexity (e.g., as much as n^3) are necessary to obtain superior results for this system in the general case. Obviously, the implications of this for actual practice are not direct—in a practical setting, we could for instance enhance MCMC convergence by using a seed draw from a distribution close to the target, or employ an alternative proposal distribution, and we could likewise enhance the bound sampler by means of the techniques described elsewhere in this paper—and there is in any event little reason to choose either the bound sampler or MCMC over exact sampling for *U|man*! However, these results do demonstrate that there are settings in which the bound sampler can produce draws whose quality would require a much more time-complex MCMC run to match. In light of the fact that mixing times for current MCMC algorithms on ERGMs can be extremely long (Bhamidi et al., 2011), it is reasonable to suspect that more such cases will be found in practice.

5 Sample Applications

As noted above, the bound sampler may be employed in any setting involving a finite series of binary random variables whose joint distribution is specified in exponential family form. Currently, the most active area of such research in sociology centers on social networks, with exponential family models (ERGMs) serving as an increasingly widely used tool for specifying models on graphs. Here, we show two applications of the bound sampler to network simulation, both reflective of typical use cases in the social network field. First, we show how the bound sampler may be used to simulate draws from an ERGM fitted to observed network data (here, a communication network), comparing the results to what would be obtained from standard MCMC methods. Second, we provide an example of exploratory simulation, in which we use the bound sampler to investigate the effect of dependence (specifically, isolation avoidance) on a large-scale spatially embedded network. In this latter case, the network in question is sufficiently large to make MCMC simulation prohibitive, making the bound sampler the only effective means of exploring the impact of edgewise dependence on the structure of the resulting network.

5.1 Simulating Draws from Fitted ERGMs

To illustrate the use of the bound sampler as a tool to simulate networks from fitted ERGMs, we consider a simple model fit to an intraorganizational communication network collected by Butts et al. (2007). Specifically, we consider radio communication ties among all named communicants in the Port Authority Trans-Hudson (PATH) control desk network during the emergency phase of the 2001 World Trade Center disaster; each vertex represents an individual, and two individuals are tied if at least one radio call was made between them (in either direction). The resulting network (see Figure 5) contains 229 individuals and 269 edges (mean degree 2.35). As is evident from the figure, this network shows substantial degree heterogeneity, some but not all of which is associated with the presence of individuals occupying institutionalized coordinative roles (e.g., desk operator) within the organization (orange nodes). Although not obvious from visual inspection, this network is also relatively triangulated given its size and density (38 triangles, CUG $p < 0.001$), suggesting an underlying tendency towards triadic closure.

To capture these effects, we fit³ a simple ERGM to the PATH network using the following terms: differential mixing rates between and within nodes with institutionalized coordinative roles (ICRs); individual-specific fixed effects for degree (i.e., sociality terms); and a geometrically weighted edge-wise shared partner term (GWESP (Hunter, 2007)) to account for residual transitivity. The coefficients for the simulated model are summarized in Table 1. Note that the apparently negative effect of ICR status on interaction probability must be interpreted in light of the individual-level fixed effects: sociality is *positively* correlated with ICR status (Spearman's $\rho = 0.33$, $p < 0.0001$), and indeed the median sociality effect for those with ICRs is substantially higher than for those without (3.648 vs. 0.002). A more nuanced view is thus that those with ICRs are more likely to be highly active, but ICRs are less likely to mix with each other than might be expected given their overall activity level. Similarly, the negative GWESP effect indicates that triangulation is reduced, but this relative to the level of clustering that would be observed in a random graph with strong degree heterogeneity effects;⁴ although there are more triangles than would be expected due to size and density alone, there are fewer (and they are more dispersed through the graph) than would be expected given the strong variation in individual activity level.

To illustrate the behavior of the bound sampler on the PATH network model, we simulate 500 draws from the fitted ERGM; ex ante fixation probability sorting was used, and perturbations were made using a diffuse Beta prior distribution with expectation equal to the observed graph density. As a point of comparison, we also simulate draws from the “gold standard” MCMC method (Hunter et al., 2008), with a burn-in of 5×10^6 draws and a thinning interval of 5×10^5 (enough to revisit each edge variable an average of approximately 20 times between each draw; convergence was also verified using standard diagnostics (Goodreau et al., 2008)). Results from both simulations are shown in Figure 6. Our simulation quality index, $\hat{Q}$, suggests that the sampler works well here: on average,

³All MCMC-based simulation and inference was performed using **ergm** (Hunter et al., 2008) and the **statnet** software suite (Handcock et al., 2008).

⁴Ceteris paribus, enhanced degree variance leads to clustering, since there are fewer ways to avoid creating triangles when edges are concentrated on a smaller set of nodes than when edges are evenly dispersed through the graph.

approximately 99% of edge variables were subject to unperturbed fixation, with very consistent performance across simulation runs (panel (A)). Although the structure of a sparse graph can be sensitive to small numbers of edge variables, Figure 6 shows that the distributions of edges (panel (B)), triangles (panel (B)), degree scores (panel (C)), and ESP statistics (panel (D)) generated by the bound sampler draws fairly closely approximate the high-quality MCMC sample. Unlike the MCMC sample, however, draws from the bound sampler are guaranteed to be independent, and require visiting each edge variable at most once.

Simulated draws like those given here are clearly adequate to approximate model behavior in cases for which MCMC is too computationally expensive, or where independence of draws is a priority (e.g., quadrature problems). Alternately, simulated draws from the bound sampler can be used to produce fully independent “seed” draws for MCMC simulations, allowing shorter MCMC runs and ensuring that the results from each run are independent. This may particularly useful in settings such as goodness-of-fit assessment, in which seeding runs with the observed data is problematic (although widely performed).

5.2 Spatial Networks with Degree Effects

Although the bound sampler can be used as a complement to MCMC where both are available, its scalability also allows it to be used to generate networks that are too large to be effectively simulated by current MCMC methods. As an illustration of such an application, we here demonstrate its use to simulate draws from a social network model combining geographical effects and degree distribution bias. The network model we employ is an extension of the “social friendship” model of (Butts, 2002), an inhomogeneous Bernoulli graph model based on friendship data from a housing study by Festinger et al. (1950) used by Butts et al. (2012) to investigate the potential impact of geographical heterogeneity on network structure. Written in conditional ERG form, this model is a curved exponential family (Hunter and Handcock, 2006) specified by

$$Pr(Y_{ij}=1|\theta, t, \mathcal{Y}_N, X, Y_{ij}^c=y_{ij}^c) = \left[1 + \exp \left[-\log \left(\frac{(1+\alpha D_{ij})^\gamma}{p_b} - 1 \right) \right] \right]^{-1} \quad (9)$$

where D_{ij} is the geographical distance between i and j , $p_b \in (0, 1]$ is the baseline probability of an i, j edge (at distance 0), $\alpha \in (0, \infty)$ is a scaling parameter, and $\gamma \in (0, \infty)$ is a shape parameter governing the form of the decay of tie probabilities at long distances. (For an investigation of the consequences of this model for network structure, see Butts and Acton (2011) or Butts et al. (2012).) One factor uncaptured in this original model is the tendency for individuals to seek to maintain personal networks within a particular size range; intuitively, individuals are incentivized to avoid friendlessness (i.e., having degree 0), and the cost of maintaining friendship ties implies that few persons can sustain large numbers of friendships. A natural way to model such effects is to supplement Equation 9 by adding statistics that count the number of vertices having a particular degree. E.g.:

$$Pr(Y_{ij}=1|\theta, t, \mathcal{Y}_N, X, Y_{ij}^c=y_{ij}^c) = \left[1 + \exp \left[-\log \left(\frac{(1+\alpha D_{ij})^\gamma}{p_b} - 1 + \delta^T \left(t(y_{ij}^-) - t(y_{ij}^+) \right) \right) \right] \right]^{-1} \quad (10)$$

where $t_0, \dots, t_{N-1}$ is a vector of statistics such that $t_i(y)$ is the number of vertices of G having degree i , and $\delta \in \mathbb{R}^N$ is a parameter vector. Any particular choice of δ creates a specific “bias” in the degree distribution, relative to what would be obtained if the baseline effects of Equation 9 were the only factors influencing network structure.

Because vertex degrees are not independent of one another, the model of Equation 10 no longer has Bernoulli structure: the conditional probability of observing a particular edge depends upon other edges within the graph. (Specifically, this model exhibits Markov graph dependence; see Frank and Strauss (1986).) The traditional approach to simulation of such models is the use of MCMC (see, e.g. Snijders, 2002), which scales poorly for large networks (Bhamidi et al., 2011). Here, we employ instead our proposed sampling procedure, taking draws from the model of Equation 10 with fixed distance effects and various choices of δ . For the baseline parameters, we use the median parameter estimates from Butts (2002), specifically $p_b = 0.529$, $\alpha = 0.031423$, and $\gamma = 2.768$. Vertex positions for this simulation are based on block level data from the year 2000 US Census, for the Choctaw, MS metropolitan statistical area. In particular, each household within each census block is assigned a graphical location uniformly at random within the block area (with individuals not belonging to households being treated as households of size 1). After assigning households to positions, each household member is assigned a position from a circular jittering distribution centered on the household location, with a maximum radius of 5 meters. This placement process thus preserves within-block population densities, while also respecting local clustering associated with variable household size; while simple, this model has been shown to reasonably approximate the micro-distribution of household locations (Almqvist and Butts, 2012). The resulting population distribution is depicted in Figure 7 ($|V| = 9,758$).

For the δ effects, we here focus on the phenomenon of *isolation avoidance*: where ties are valuable, those who would otherwise be socially isolated may be motivated to seek them (hence leading to a reduced proportion of vertices with degree zero). In ERGM terms, this amounts to setting $\delta_i = 0$ for $i > 0$, and δ_0 to a negative number (reflecting the degree to which isolation is penalized), which is in turn equivalent to simply including an isolate term in the spatial model with parameter $\delta_0 < 0$.

To explore the potential impact of isolation avoidance in the Choctaw model, we take 100 draws using the bound sampler, with δ_0 set respectively to 0, -1 , -2 , and -4 (reflecting an increasingly sharp penalty for isolate formation). Mean values for selected network statistics are given in Table 2; standard errors (not shown) are less than 1% of the mean value in all cases, and all differences are significant at $p \ll 0.001$. The first column of Table 2 shows the behavior of the baseline model without dependence effects: this is an exact sample ($Q = 1$), in which approximately 32% of nodes are isolated on average (*Isolate Fraction*). The average mean degree in the baseline case is 1.28 (unsurprisingly low, given the large isolate fraction), but the graph is relatively clustered (transitivity of approximately 0.18). As a simple measure of embeddedness in cohesive subgroups, we employ the *mean core number*, which is equal to the mean of the number of the highest-order k -core (Wasserman and Faust, 1994) to which each vertex belongs. Vertices with a core number of 0 are isolates; those with a core number of 1 belong to pendant trees, those with a core number of 2 belong to bicomponents, etc. The mean core number under the baseline model is approximately 0.95,

reflecting the strong contribution from isolated and pendant vertices in the graph. Finally, we note that—as expected from the spatial model—there is a clear negative relationship between distance and the presence of a tie (expressed as a correlation between edge state and log distance).

How do these properties change when isolate avoidance is added to the model? For this, we look to the succeeding columns of Table 2, which show mean values for the above statistics at increasingly negative parameter values. First, we verify that, as δ_0 falls, the fraction of isolated vertices does indeed decline (dropping from a 32% baseline to just over 1% at $\delta_0 = -4$). Simulation quality (as evaluated via $\hat{Q}$) remains high throughout, although the fixation rate falls slightly as the dependence among edges is increased (from 1 in the baseline model to approximately 0.994 at $\delta_0 = -4$). With these indications that the sampler is behaving as expected, we proceed to consider the impact of increasing isolation avoidance on other aspects of network structure. For instance, a reduction in isolation could be achieved in practice via two processes: one could see a net increase in edges (with new edges forming from those who would otherwise be isolated to other vertices in the graph), or one could see a reallocation of edges (with an increase in edges to otherwise isolated nodes being compensated by a loss of edges by nodes that would otherwise have higher degree). As the table shows, the former mechanism is active here—mean degree increases as δ_0 falls, implying (given the fixed vertex set) a net increase in the number of edges in the graph. This increase is associated with an increase in the mean core number, reflecting the movement of vertices from the 0-shell (the set containing those in the 0-core but not the 1-core) into shells of higher order.

Does the movement of vertices from an isolated state to a better-connected one reflect embeddedness in dense subgroups, embeddedness in locally tree-like structures, or formation of lone dyads? We can tell from the declining transitivity score that the second of these possibilities is the dominant effect: given that isolates are being removed via edge addition, a declining transitivity score implies the formation of new incomplete 2-paths. Lone pairs do not produce such structures, and hence cannot be the source of the change; nor would the creation of dense, heavily triangulated subgroups. Several conclusions follow. First, since would-be isolates are not primarily pairing with each other, they are necessarily forming ties to vertices that would otherwise have higher degree; suppression of isolates thus has “spillover” effects for higher degree nodes. Second, since said potential isolates are generally embedding in locally tree-like structures, it seems plausible that isolate suppression increases connectivity without a corresponding increase in cohesion (i.e., formation of redundant paths). Intuitively, cohesion is driven heavily by the fraction of nodes having degree at least 2 (since any node of degree less than 2 cannot lie on a cycle), and hence social processes that act primarily to shift vertices from degree 0 to degree 1 will have a very different impact than those that act to increase degree more uniformly.

In addition to asking how the positional structure of the network changes as isolates are suppressed, we can also ask how isolate avoidance alters the spatial structure of the network. From Table 2, we can see that the negative relationship between tie probability and distance inherent in the spatial component of the model persists (with some mild fluctuations) as δ_0 becomes more negative; for a richer view of the relationship, it is useful to examine the

detailed distribution of ego-alter distances at each parameter value. Figure 8 shows the cumulative distribution functions (CDFs) for geographical distances among adjacent alters for representative draws from the bound sampler at each value of δ_0 (with the dotted black line showing CDF for distances among all dyads). Consistent with the negative spatial correlation, there is a strong tendency for adjacent alters to be proximate relative to what would be expected under random mixing (all model CDFs are above the baseline). Introducing an isolation penalty, however, greatly increases the number of long-range ties in the network: while 90% of ties are within 100m in the purely spatial model, this drops to 83% for $\delta_0 = -1$, 72% for $\delta_0 = -2$, and 54% for $\delta_0 = -4$. Moreover, the disparity is greatest for edges to alters who are relatively close (e.g., < 1km) while being outside of ego's immediate household area.

Taken together with the above results, this gives us a more complete understanding of what happens as isolation avoidance increases. Nodes that would otherwise be isolated form ties to others (not merely other isolates) who are farther away than would otherwise be likely, but still favoring those at short distances. These ties tend not to be embedded in locally cohesive structures, but are instead part of tree-like complexes (possibly converting previously isolated vertices into pendants). The result is a graph that is better connected, with slightly greater spatial mixing, but without an increase in overall clustering. (Indeed, the median core number holds constant at 1—and the average maximum at approximately 4—over all simulations, consonant with conclusion that most change involves movement from the 0-shell into the 1-shell.) This is a distinctive pattern of behavior that would be difficult or impossible to explore without the ability to efficiently simulate large networks, and in particular using conventional MCMC-based sampling schemes. The bound sampler gives us the ability to investigate much larger systems than have been accessible in the past.

6 Discussion

Like any sampling scheme, the bound sampler has inherent tradeoffs and complexities that may indicate—or contraindicate—its use in particular settings. Before concluding, we comment here on several issues related to the use of the bound sampler in practice, both in the context of network simulation and in other applications.

6.1 Anticipating Bound Sampler Quality

Section 4 analyzed bound sampler quality in the context of the exactly computable $U|man$ family; section 5 showed the behavior of the bound sampler in several more realistic applications. Beyond these special cases, what can be said regarding the quality of bound sampler draws in practice? Noting that a lower bound on Q can be obtained by the mean distance between and in the absence of any prior fixation, it follows that distributions admitting tight Bernoulli bounds (in the sense of Butts (2011)) will have provably high quality. Butts (2011) discusses various factors influencing bound tightness, the most important of which being the degree of inhomogeneity in baseline probabilities and the degree of coupling between variables. In general, the most loosely bounded distributions are those involving strong global coupling with a high degree of homogeneity; highly inhomogeneous distributions with weaker and/or more local coupling admit tighter bounds.

In the context of the bound sampler, the presence of fixation cascades can lead to better performance than a simple Bernoulli bound analysis would suggest, and this should thus be viewed as a sufficient but not necessary condition for high quality draws.

6.2 Guidance for Use Vis a Vis MCMC

As we have noted throughout, both the bound sampler and MCMC samplers are approximate; MCMC draws can in principle be made arbitrarily close to exact, but in practice this depends on mixing time (which can be quite long) and requires convergence assessment. Where circumstances permit the generation of high-quality MCMC samples (and neither computational costs nor convergence assessment are problematic), MCMC is recommended. This is particularly true when the $\hat{Q}$ statistic is low, and bound sampler quality cannot be guaranteed. On the other hand, the bound sampler is recommended as an alternative to MCMC in settings where the latter's mixing time is prohibitively long (due either to coupling time between variables or system size) and the quality of the former is acceptable. Pilot simulations may be required to assess this. The bound sampler is also suggested as a possible alternative to MCMC when independence of draws is a serious concern, and/or where unsupervised execution is a priority (again, assuming that the sample quality is adequate). The third alternative of relatively short MCMC runs seeded with independent bound sampler draws may be an even better option in such cases, and should be considered when neither approach alone appears to produce adequate results.

One exception to the above guidance is when the objective is to *bound* a statistic on Y (e.g., place an upper bound on the expected number of triangles formed within a network). In this case, the bound sampler allows provable bounds on e.g., the noncentral moments or quantiles of the statistic in question from an exact sample, a property that MCMC does not share. In these applications, the bound sampler may be preferred to MCMC even when the latter produces higher quality draws overall.

6.3 Non-ERGM Applications of the Bound Sampler

Although we have focused here on its use in the context of ERGM simulation, the bound sampler can be used in any setting in which one seeks to simulate binary vectors and can represent one's target distribution in exponential family form. One such setting is the modeling of influence on binary attitudes (approve/disapprove, agree/disagree) or behaviors within a fixed network or other social setting; the exponential family models employed by Holyst et al. (2001) to model social impact dynamics (Latané, 1981; 1996) are one example of the types of influence process to which the bound sampler could be applied. Other settings lending themselves naturally to representation as dependent binary processes are modeling of dichotomous voting systems (for/against) and attendance or participation by persons or organizations at events. Examples of such processes have been famously studied e.g. by Coleman (1964); Granovetter (1978), and exponential families for the latter have recently been used in dynamic network models (see, e.g. Almquist and Butts, 2014) but could easily be treated in their own right.

6.4 Generalizations to the Non-Binary Case

Finally, it should be noted that, while we have here been concerned exclusively with binary vectors, generalizations of the bound sampler to more general contexts are possible. For instance, consider the case in which we seek to simulate a vector, $Z = Z_1, \dots, Z_N$, such that Z_i has support $0, 1, \dots, M_i$ (with M known). Such random variables arise e.g. in the context of bounded count processes, as when an event has a fixed number of occasions to occur and the total number of occurrences is observed. By defining a binary vector, Y , such that (1) each Z_i is the sum of M_i distinct elements of Y and (2) the corresponding vector of sums has the same distribution as the original Z , we may reduce the problem of simulating a count vector to the problem of simulating its associated binary representation. As a general matter, wherever such a reduction can be carried out, the bound sampler can then be employed to sample the original variable.

7 Conclusion

In this paper, we have introduced a novel alternative sampling algorithm for binary vectors with fixed execution time and quality guarantees. This approach stands in contrast to MCMC methods, which have variable convergence time (taking “convergence” to be defined in pragmatic rather than ideal terms) and which cannot be deterministically related ex ante to an exact sampler. Our approach is based on sequential approximations to the partial marginals of the joint distribution, and exploits the fact that bounds on these marginals can often be used to exactly determine the “correct” state of a given random variable (vis a vis an exact sampler to which our sampler is stochastically coupled). Where the state of the corresponding exact sampler is uncertain, the elements of the simulated draw that are fixed (i.e., known to match the exact draw) can still be used to place bounds on the properties of the vector as a whole. (For instance, the total number of 1s in the exact draw must be at least the number of 1s in the fixed portion of the simulated draw, and no more than this number plus the number of unfixed elements.) These properties make the bound sampler a potentially useful alternative to MCMC in unsupervised settings, or when principled bounds on a random quantity are more important than exact estimates.

Although there are many problems that can be expressed in terms of random binary vectors, the most active in current sociological research is the simulation of random graphs (a fundamental task for both theory and empirical investigation in the study of social networks). To that end, we here illustrated the use of the bound sampler to simulate social networks from both empirically fitted ERGMs and from exploratory models. In the former case, we showed that the bound sampler can provide a reasonable proxy for MCMC-based draws, with the added benefits of fixed execution time and quality guarantees mentioned above. In the latter, we showed that the superior scalability of the bound sampler allowed it to be used to simulate very large networks, where current MCMC implementations are too slow to be useful. Given the increasing importance of large-scale networks within sociological research, there is a substantial need for principled techniques for statistical simulation of networks with tens of thousands (or more) vertices; the bound sampler would seem to have considerable virtue in such applications.

Finally, we note that there are many potential extensions and refinements of our basic algorithm (some of which were discussed in Section 2.2). While a drawback of our method is that, unlike the case of MCMC, one cannot improve simulation quality simply by running it longer, techniques such as pseudo-marginalization can be used to improve quality where necessary. Further research on when and where such methods can be used, and on further techniques to enhance simulation quality, would seem to be an important direction.

References

- Almquist ZW, Butts CT. Point process models for household distributions within small areal units. *Demographic Research*. 2012; 26(22):593–632.
- Almquist ZW, Butts CT. Logistic network regression for scalable analysis of networks with joint edge/vertex dynamics. *Sociological Methodology*. 2014; 44(1):273–321. [PubMed: 26120218]
- Batagelj V, Brandes U. Efficient generation of large random networks. *Physical Review E*. 2005; 71(3):036113–1.
- Besag J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*. 1974; 36(2):192–236.
- Bhamidi S, Bresler G, Sly A. Mixing time of exponential random graphs. *The Annals of Applied Probability*. 2011; 21(6):2146–2170.
- Butts, CT. Doctoral Dissertation. Carnegie Mellon University; 2002. *Spatial Models of Large-scale Interpersonal Networks*.
- Butts CT. Social network analysis with sna. *Journal of Statistical Software*. 2008; 24(6)
- Butts CT. Bernoulli graph bounds for general random graphs. *Sociological Methodology*. 2011; 41:299–345.
- Butts, CT.; Acton, RM. Spatial modeling of social networks.. In: Nyerges, T.; Couclelis, H.; McMaster, R., editors. *The SAGE Handbook of GIS and Society Research*. SAGE Publications; 2011. p. 222-250. chapter 12
- Butts CT, Acton RM, Hipp JR, Nagle NN. Geographical variability and network structure. *Social Networks*. 2012; 34:82–100.
- Butts CT, Petrescu-Prahova M, Cross BR. Responder communication networks in the World Trade Center Disaster: Implications for modeling of communication within emergency settings. *Journal of Mathematical Sociology*. 2007; 31(2):121–147.
- Coleman, JS. *Introduction to Mathematical Sociology*. Free Press; New York: 1964.
- Festinger, L.; Schachter, S.; Back, K. *Social Pressures in Informal Groups*. Stanford University Press; Stanford, California: 1950.
- Frank O, Strauss D. Markov graphs. *Journal of the American Statistical Association*. 1986; 81:832–842.
- Gelman, A. Inference and monitoring convergence.. In: Gilks, WR.; Richardson, S.; Spiegelhalter, DJ., editors. *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC; Boca Raton, FL.: 1996.
- Goodreau SM, Handcock MS, Hunter DR, Butts CT, Morris M. A statnet tutorial. *Journal of Statistical Software*. 2008; 24(9)
- Granovetter M. Threshold models of collective behavior. *American Journal of Sociology*. 1978; 83:1420–1443.
- Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M. statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*. 2008; 24(1):1–11. [PubMed: 18612375]
- Holland PW, Leinhardt S. An exponential family of probability distributions for directed graphs (with discussion). *Journal of the American Statistical Association*. 1981; 76(373):33–50.
- Holyst JA, Kacperski K, Schweitzer F. Social impact models of opinion dynamics. *Annual Reviews of Computational Physics*. 2001; 9:253–273.
- Hunter DR. Curved exponential family models for social networks. *Social Networks*. 2007; 29:216–230. [PubMed: 18311321]

- Hunter DR, Handcock MS. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*. 2006; 15:565–583.
- Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*. 2008; 24(3)
- Latané B. The psychology of social impact. *American Psychologist*. 1981; 36(4):343–356.
- Latané B. Dynamic social impact: The creation of culture by communication. *Journal of Communication*. 1996; 46(4):13–25.
- Morris M, Handcock MS, Hunter DR. Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*. 2008; 24(4):1–24. [PubMed: 18612375]
- Robins GL, Pattison PE, Elliott P. Network models for social influence processes. *Psychometrika*. 2001; 66:161–190.
- Robins GL, Pattison PE, Woolcock J. Small and other worlds: Network structures from local processes. *American Journal of Sociology*. 2005; 110(4):894–936.
- Skvoretz J, Fararo TJ, Agneessens F. Advances in biased net theory: Definitions, derivations, and estimations. *Social Networks*. 2004; 26:113–139.
- Snijders TAB. Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*. 2002; 3(2)
- Wasserman, S.; Faust, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press; Cambridge: 1994.
- Wasserman, S.; Robins, GL. An introduction to random graphs, dependence graphs, and p^* . In: Carrington, PJ.; Scott, J.; Wasserman, S., editors. *Models and Methods in Social Network Analysis*. Cambridge University Press; Cambridge: 2005. p. 192-214.chapter 10

Figure 1. Expected $\hat{Q}$ ($\times 100$) as a function of edge and mutuality parameters for the bound sampler (edge/mutual model). Shading interpolates linearly from white (lowest observed $\hat{Q}$) to black (maximum $\hat{Q}$).

Figure 2.

Mean absolute errors (rounded) in expected edge count for bound and MCMC samplers as a function of edge and mutuality parameters (edge/mutual model). Shading interpolates linearly from white (no error) to black (maximum observed error).

Figure 3.

Mean absolute errors (rounded) in expected mutual count for bound and MCMC samplers as a function of edge and mutuality parameters (edge/mutual model). Shading interpolates linearly from white (no error) to black (maximum observed error).

Figure 4.

Mean absolute errors (rounded) in expected triangle count for bound and MCMC samplers as a function of edge and mutuality parameters (edge/mutual model). Shading interpolates linearly from white (no error) to black (maximum observed error).

Figure 5.
PATH Control Desk communication network. Light colored nodes indicate individuals with institutionalized coordinative roles.

Figure 6.

Bound and MCMC sampler results for the PATH network model. (A) Distribution of ex post quality measure (Q) for the bound simulator; ex ante expected quality (here identical for all draws) shown via dotted vertical line. (B) Edge and triangle count distributions for the bound sampler (left) versus MCMC (right). (C)-(D) Degree distribution and edgewise shared partner distributions, respectively, for the bound sampler versus MCMC.

Population Distribution, Choctaw, MS

Figure 7. Population distribution for Choctaw, MS. Points are placed uniformly by census block ($|V| = 9,758$).

Figure 8. Cumulative distance distributions for Choctaw, MS. Dotted black line shows the baseline CDF over all dyads, colored lines show CDFs for distances between adjacent alters for each value of δ .

Table 1

Fitted ERG model for the PATH network simulation; sociality effects summarized for brevity.

Term	$\hat{\theta}$	Std. Err.	z Value	$Pr(> z)$	
Mixing (ICR/ICR)	-11.4759	0.4021	-28.541	0.0051	**
Mixing (ICR/Non-ICR)	-8.8792	1.2253	-7.247	< 0.0001	****
Mixing (Non-ICR/Non-ICR)	-7.2018	2.5701	-2.802	< 0.0001	****
GWESP($\alpha = -0.27$)	-0.4538	0.1519	-2.987	0.0028	**
Sociality (FE by actor)	Median $\hat{\theta}$: 0.0024, IQR $\hat{\theta}$: 1.1496				

Table 2

Mean graph statistics by isolate term parameter, Choctaw model (100 draws).

	$\delta_0 = 0$	$\delta_0 = -1$	$\delta_0 = -2$	$\delta_0 = -4$
Isolate Fraction	0.3249	0.1285	0.0607	0.0112
Mean Degree	1.2789	1.7707	2.0520	2.2940
Mean Core Number	0.9494	1.2833	1.4165	1.5043
Transitivity	0.1750	0.1590	0.1311	0.0913
$\rho(y_{ij}, \ln D_{ij})$	-0.1006	-0.1135	-0.1125	-0.0900
$\hat{Q}$	1.0000	0.9999	0.9994	0.9941

Algorithm 1

Approximate Sampling Algorithm

```

1: for  $i$  in  $1, \dots, N$  do
2:   Set  $\alpha_i' := \min_{y \in Y_N: y_{<i} = y_{<i}'} \Pr(Y_i = 1 \mid \theta, Y_i^c = y_i'^c)$ 
3:   Set  $\gamma_i' := \max_{y \in Y_N: y_{<i} = y_{<i}'} \Pr(Y_i = 1 \mid \theta, Y_i^c = y_i'^c)$ 
4:   Draw  $r_i$  from  $U(0, 1)$ 
5:   if  $r_i < \alpha_i'$  then
6:     Set  $y_i = 1$ 
7:   else if  $r_i < \gamma_i'$  then
8:     Set  $y_i = 0$ 
9:   else
10:    Draw  $\tau$  from  $U(\alpha_i', \gamma_i')$ 
11:    if  $r_i < \tau$  then
12:      Set  $y_i := 1$ 
13:    else
14:      Set  $y_i := 0$ 
15:    end if
16:  end if
17: end for
18: return  $y$ 

```