Are we there yet? When to stop a Markov chain while generating random graphs*

Jaideep Ray, Ali Pinar, and C. Seshadhri**

Sandia National Laboratories, Livermore, CA 94550
{jairay,apinar,scomand}@sandia.gov

Abstract. Markov chains are convenient means of generating realizations of networks with a given (joint or otherwise) degree distribution, since they simply require a procedure for rewiring edges. The major challenge is to find the right number of steps to run such a chain, so that we generate truly independent samples. Theoretical bounds for mixing times of these Markov chains are too large to be practically useful. Practitioners have no useful guide for choosing the length, and tend to pick numbers fairly arbitrarily. We give a principled mathematical argument showing that it suffices for the length to be proportional to the number of desired number of edges. We also prescribe a method for choosing this proportionality constant. We run a series of experiments showing that the distributions of common graph properties converge in this time, providing empirical evidence for our claims.

Keywords: graph generation, Markov chain Monte Carlo, independent samples

1 Introduction

Degree distributions (DD) and joint degree distributions (JDD) are some of the most fundamental properties of real world networks. The degree distribution of an undirected graph G is a vector $\boldsymbol{f}$, where $f(d)$ is the number of vertices of degree d. The *joint degree distribution* is an $n \times n$ matrix $\mathbf{J}$, where the entry $J(i,j)$ is the number of edges between vertices of degree i and degree j. The landmark paper [1] observing heavy-tailed degree distributions in real networks forms the basis of much research on these graphs. Notions like *assortativity* [2], that are captured by the joint degree distribution, are an important metric used to understand these networks. To gain deeper understanding of these graph properties, we often perform experiments trying to understand how the degree

* This work was funded by the Applied Mathematics Program at the U.S. Department of Energy and performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

** This author was supported by an Early-Career award from the Laboratory Directed Research & Development (LDRD) program at Sandia National Laboratories.

distribution affects other graph properties. For example, is assortativity correlated with the clustering coefficient [3]? A key ingredient to performing these studies is generating uniform random graphs with a prescribed (joint) DD.

Markov chain Monte Carlo (MCMC) methods are a common means of doing this [4,5,6,7,8]. We start with a given graph with a specified DD (or JDD [8]); it is often a real graph whose properties we are studying. There is a simple and standard procedure that performs a random edge swap preserving the DD [4,9,10] (or JDD [8]). This gives a Markov chain on the space of graphs with the given DD (JDD), and we take many steps to generate a sample. But how many steps should we take to generate a uniform random sample?

If a bound on the mixing time of this chain is known, then that gives a convenient bound on the number of steps to take. For the DD and JDD Markov chains [1], theoretical bounds have been given [4]. These are of the form $O(n^6)$, where n is number of vertices of the graph. Even for a moderate size of $n \approx 1000$, this is quite useless in practice. Empirically, the number of steps is usually chosen quite arbitrarily. Since this sampling can often form the basis of experiments, this is quite dangerous. If a Markov chain has not mixed properly, samples generated may be highly correlated and conclusions drawn from them can be erroneous.

1.1 Results

The primary goal of this paper is to bridge this gap between theory and practice. Our results hold for both DD and JDD Markov chains. The results for JDD are more involved and interesting, so only they are presented in this paper. We give a mathematically principled argument showing that to generate a graph with $|E|$ edges, it suffices to run the Markov chain $O(|E|)$ steps. The constant hidden in the big-Oh depends on a desired accuracy. Our experiments show that $10|E| - 30|E|$ steps are enough for the purpose of ascertaining various graph properties.

1. *Theoretical results:* Mathematically, this range is achieved by approximating the behavior of the entire Markov chain by a set of coupled 2-state Markov chains, one for each pair of vertices. This is a heuristic approximation in case of JDD (but for DD, this is a provable equivalence.) The mixing time of these 2-state chains can be directly bounded by $O(|E|)$ (where the constant is a standard dependence on the desired accuracy). This means that in $O(|E|)$ steps, while we may not be able to assert total mixing, *each edge appears as if we are in the uniform distribution.* Observe that this is certainly a necessary condition for total mixing.

2. *Empirical results:* This is in two parts. First, we give empirical evidence that our predicted length works in practice. It is quite difficult to directly ascertain that a given sample is truly uniform random [11]. So, for a given length ℓ, we generate a number of sample graphs, each with a separate ℓ-length walk, and plot the distribution of a common graph parameter (say, clustering coefficient). We observe that for $\ell > 10|E|$, these distributions converge and do not change

[1] These bounds only hold when the graph generated is not necessarily simple.

further. On the other hand, when ℓ is only $|E|$, the distribution is very far from reaching convergence. Our predictions clearly match the experiments. Next, we justify the approximation of the Markov chain on the space of graphs as a set of coupled 2-state chains. We look at the behavior of an individual edge over a very long walk in the overall Markov chain, i.e., a long binary time-series with 0/1 indicating the absence/presence of an edge at each step of the Markov chain. If our approximations are correct, then thinning this series by a factor of $O(|E|)$ should lead to a sequence of practically independent samples. We run statistical tests to show that this really does happen.

Our idea is similar to Sokal's method [12] for deciding the "sufficiency" of samples obtained from MCMC based on *autocorrelation*. The idea of Sokal, as adopted by Stanton and Pinar [8], was to look at the individual edges as a binary time-series. They then compute the autocorrelation, at different lags, which can be thought of as a measure of how long it takes for the time-series to become uncorrelated. It is suggested to keep walking (in the Markov chain) until all auto-correlations are below a prescribed threshold. However, Sokal's method has two major practical drawbacks - (1) the autocorrelation analysis is performed for all the edges (n^2 in number for a graph with n vertices) that might appear in the MCMC chain and (2) one has to choose a autocorrelation threshold, for which there are no guidelines. In contrast, our method estimates the number of Markov chain steps with a closed-form expression.

2 Theoretical analysis

We first describe the Stanton and Pinar Markov chain for preserving the joint degree distribution [8]. This is analogous to the degree distribution preserving chain [4,7]. These methods are quite standard and come with schemes to generate a specific graph with a given DD or JDD.

Consider an undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$. As mentioned earlier, the *joint degree distribution* is an $n \times n$ matrix $\mathbf{J}$, where $J(i, j)$ is the number of edges between vertices of degree i and degree j. We will also use the degree distribution $\boldsymbol{f}$, where the coordinate $f(d)$ the number of vertices of degree d.

The process of generating a new graph, from an older one, by swapping edges, is called "rewiring". The rewiring is done as follows. We use d_v to denote the degree of v. The process is depicted in Fig. 1. This may lead to self-loops and parallel edges, and there are methods of dealing with this. We will not get into those details, and refer the reader to [8]. Note that every vertex maintains its degree, and the joint degree distribution is always preserved. We also maintain lists of nodes and edges indexed by their degree, so that for a specified degree d, a uniform random edge incident to a degree d vertex can be located. The steps are:

– Pick a uniform random *endpoint*. This is done by choosing a uniform random edge and choosing each endpoint with probability 1/2. Suppose we choose endpoint u_1 incident to edge (u_1, v). d_v is arbitrary. See Fig. 1.

– Choose a uniform random edge with an endpoint of degree d_{u_1}. Let this edge be (u_2, w). d_w is arbitrary.

– Swap edges (u_1, v) and (u_2, w). This adds edges (u_1, w) and (u_2, v) and removes (u_1, v) and (u_2, w).

Details of the rewiring scheme and a Markov chain driving it to generate (correlated) graph samples e.g., discussions of ergodicity etc., can be found in [8].

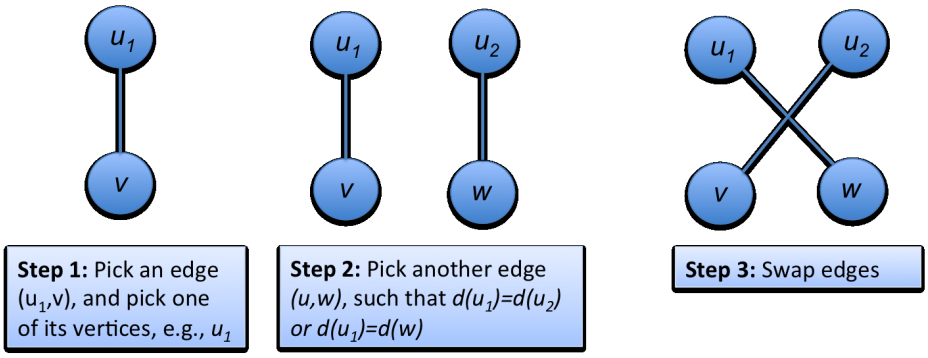

Fig. 1. The swapping operation for the Markov chain algorithm.

2.1 Approximation by many 2-state Markov Chains

Consider a fixed pair of labeled vertices (u, v). Let us try to understand the probability that this edge appears or disappears. Based on this, we can approximate the behavior of the pair (u, v) as a Markov chain. We start with a simple yet important claim.

Claim. Suppose at some stage in the Markov chain, the edge (u, v) is present. The probability that it is removed in the next step is

$$\frac{1}{m} + \frac{f(d_u)d_u + f(d_v)d_v - d_u - d_v}{2m^2}. \tag{1}$$

Proof. The swapping procedure picks two edges, which we shall refer to as e (the first edge) and e' (the second edge). If e is chosen to be (u, v), then (u, v) will definitely be swapped out. The probability of this is $1/m$. On the other hand, e may not be (u, v) but e' could be (u, v). If the random endpoint of e chosen has degree d_u (and is not u), then we might choose e' to be (u, v). The total number of edges incident to degree d_u vertices (but not u) is $(f(d_u) - 1)d_u$. Any of these edges is a potential candidate for e. Hence, the probability of choosing e with this property, and then $e' = (u, v)$ is

$$\frac{(f(d_u) - 1)d_u}{2m} \times \frac{1}{m} = \frac{(f(d_u) - 1)d_u}{2m^2}$$

We could also choose the random endpoint to have degree d_v. So the total probability of choosing $e' = (u, v)$ is

$$\frac{f(d_u)d_u + f(d_v)d_v - d_u - d_v}{2m^2}.$$

The total probability that (u, v) is swapped out is

$$\frac{1}{m} + \frac{f(d_u)d_u + f(d_v)d_v - d_u - d_v}{2m^2}.$$

□

While this claim may look fairly innocuous, it makes a very strong observation. When edge (u, v) is present, the probability that it is swapped out only depends on the values $d_u, d_v, f(d_u), f(d_v)$. These values are the same regardless of where we are in the Markov chain, because we always preserve the degree distribution! Hence, this satisfies the Markov property, and the probability is independent of the graph itself. But what about the probability that (u, v) becomes an edge?

This is unfortunately not truly Markovian, since it could depend on the remainder of the graph. Nonetheless, this dependence appears to be fairly weak. We can obtain a Markovian estimate for this probability with a simple heuristic. We guess the number of edges incident to vertex v that are also incident to a degree d vertex (for some d). Clearly, this number depends on the graph structure, but we can approximate it based on the JDD. The number of edges from degree d to degree d_v vertices is $\mathbf{J}(d, d_v)$. Of these, the average number of edges incident to a fixed vertex of degree d_v is $\mathbf{J}(d, d_v)/f(d_v)$. We shall approximate the number of edges incident to v with the other endpoint of degree d by this quantity.

Claim. Assume the heuristic approximation above. If at any stage of the Markov chain, the edge (u, v) is not present, the probability that edge (u, v) appears is given by

$$\frac{\mathbf{J}(d_u, d_v)}{2m^2}\left(\frac{d_u}{f(d_v)} + \frac{d_v}{f(d_u)}\right) \tag{2}$$

We omit the proof for this claim due space limitations; however, it is available in [13].

We now focus on the presence or absence of the edge (u, v) as we walk through the Markov chain. Based on the claims above, this can be thought of as a 2-state Markov chain (state 0 meaning no edge, and state 1 meaning presence of edge). The transition matrix $\mathbf{T}_{u,v}$ for this chain is

$$\mathbf{T}_{u,v} = \begin{pmatrix} 1 - \alpha_{u,v} & \alpha_{u,v} \\ \beta_{u,v} & 1 - \beta_{u,v} \end{pmatrix}, \tag{3}$$

where $\alpha_{u,v}$ (resp. $\beta_{u,v}$) is the probability that (u, v) appears (resp. disappears). These probabilities are given by Eq. 2 and Eq. 1 respectively. We will denote

this Markov chain by $\mathcal{M}_{u,v}$ and the stationary distribution of it by $\boldsymbol{\pi}_{u,v}$. The eigenvalues of this transition matrix are 1 and $1-(\alpha_{u,v}+\beta_{u,v})$. The important observation is that the second eigenvalue is at most $1-1/m$, by Eq. 1. The next claim follows from standard Markov chain arguments.

Claim. Set $N = m\ln(1/\epsilon)$. Let the final distribution after running $\mathcal{M}_{u,v}$ for N steps be $\mathbf{p}$. Then $\|\mathbf{p}-\boldsymbol{\pi}_{u,v}\| < \epsilon$.

Observe that $\boldsymbol{\pi}_{u,v}$ represents the probability of presence/absence edge (u,v) in the overall stationary distribution of the entire Markov chain. This claim implies that in $N = m\ln(1/\epsilon)$ steps, we are very close to the stationary distribution for each edge. This bound is independent of the edge. So each edge behaves like it should in the stationary distribution (as far as the overall graph is concerned, we cannot make a stronger claim).

Proof. Denote the unit eigenvectors of $\mathbf{T}$, corresponding to the eigenvalues 1 and $1-(\alpha_{u,v}+\beta_{u,v})$, as $\mathbf{e}_1$ and $\mathbf{e}_2$. Since these $\alpha_{u,v}+\beta_{u,v} > 0$, these form a basis. The initial state can be expressed as $\mathbf{v} = c_1\mathbf{e}_1 + c_2\mathbf{e}_2$. After N applications of the transition matrix we get

$$\mathbf{p} = \mathbf{T}^N\mathbf{v} = c_1\mathbf{T}^N\mathbf{e}_1 + c_2\mathbf{T}^N\mathbf{e}_2 = c_1\mathbf{e}_1 + c_2\left(1-(\alpha_{u,v}+\beta_{u,v})\right)^N\mathbf{e}_2.$$

Since $1-(\alpha_{u,v}+\beta_{u,v}) < 1$, the second term decays with N and $c_1\mathbf{e}_1$ is the stationary distribution $\boldsymbol{\pi}_{u,v}$. For convenience, set $\gamma = \alpha_{u,v}+\beta_{u,v}$. The key observation is that $\gamma \geq 1/m$, by Eq. 1. Hence,

$$N = m\ln(1/\epsilon) \geq \ln(1/\epsilon)/\gamma. \tag{4}$$

We can bound the norm of the difference $\mathbf{p}-\boldsymbol{\pi}_{u,v}$ as

$$\|\mathbf{p}-\boldsymbol{\pi}_{u,v}\| = \|(1-\gamma)^N c_2\mathbf{e}_2\|_2 \leq (1-\gamma)^{\ln(1/\epsilon)/\gamma} c_2\|\mathbf{e}_2\|_2 \leq \exp(-\ln(1/\epsilon)) = \epsilon \tag{5}$$

□

3 Verifying the edge-by-edge convergence

The expression for N, as derived in Section 2.1, is based on a heuristic and has to be verified. In addition, the expression is derived strictly applicable to an edge, and it is unlikely that after N steps, *all* edges will become decorrelated. The residual number of partially correlated edges and their effect on graphical metrics have to be quantified.

Below we construct a purely data-driven, non-parametric test for the independence of a edge, in a Markov chain of graphs. Any specified edge in a Markov chain of graphs traces a binary time-series $\{Z_t\}$, indicating the presence/absence of the edge at each step of the chain. Assume that the chain is very long, i.e., it takes $K \gg N$ steps. The time-series so formed will be auto-correlated, as observed by Stanton and Pinar [8]. However, if the time-series is thinned by a

factor k (i.e., we retain every k^{th} element to obtain $\{Z_t^k\}$, the $k-$thinned chain), the auto-correlation of $\{Z_t^k\}$ will decay and it will begin to resemble independent draws from a distribution. If Eq. 4 is correct, then $k = N$ should yield a time-series that resembles independent draws *more* than a first-order Markov process. Resemblance to either process is established by fitting an independent and first-order Markov process models to the thinned data and computing the log-likelihood. This forms the basis of our test. While this technique has been applied in other domains [14,15], this paper is the first application of this technique to graphs.

Consider the chain $\{Z_t^k\}$. We count the number, x_{ij}, of the $(i,j), i,j \in (0,1)$ transitions in it. x_{ij} are used to populate X, a 2×2 contingency table. Dividing each entry by the length of thinned chain $K/k-1$ provides us with the empirical probabilities p_{ij} of observing an (i,j) transition in $\{Z_t^k\}$. Let $\widehat{p_{ij}}$ and $\widehat{x_{ij}} = (K/k-1)\widehat{p_{ij}}$ be the predictions of the probabilities and expected values of the table entries provided by a model. In such a case, the goodness-of-fit of the model is provided by a likelihood ratio statistic (called the G^2-statistic; Chapter 4.2 in [16]) and a Bayesian Information Criterion (BIC) score

$$G^2 = -2\sum_{i=0}^{i=1}\sum_{i=0}^{i=1} x_{ij}\log\left(\frac{\widehat{x_{ij}}}{x_{ij}}\right), \qquad BIC = G^2 + q\log\left(\frac{K}{k} - 1,\right) \tag{6}$$

where q is the number of parameters in the model used to fit the table data. Typically log-linear models are used for the purpose (Chapter 2.2.3 in [16]); the log-linear models for table entries generated by independent sampling and a first-order Markov process are

$$\log(p_{ij}^{(I)}) = u^{(I)} + u_{1,(i)}^{(I)} + u_{2,(j)}^{(I)} \quad \text{and} \quad \log(p_{ij}^{(M)}) = u^{(M)} + u_{1,(i)}^{(M)} + u_{2,(j)}^{(M)} + u_{12,(ij)}^{(M)}, \tag{7}$$

where superscripts I, M indicate an independent and Markov process respectively. The maximum likelihood estimates (MLE) of the model parameters $(u_{b,(c)}^{(W)})$ are available in closed form (Chapters 2.2.3 and 3.1.2 in [16]; also [13]). We compare the fits of the two models thus: $\Delta BIC = BIC^{(I)} - BIC^{(M)}$. Large BIC values indicate a bad fit. A negative ΔBIC indicates that an independent model fits better than a Markov model.

This test is applied as follows. We construct a thinned binary time-series $\{Z_t^k\}$ for $k = N$ for each of the edges. The ΔBIC is computed and edges with negative ΔBIC are deemed to have become independent after N steps of the Markov chain.

4 Tests with real graphs

In this section, we estimate an ϵ for Eq. 4 within the context of a set of graphical metrics. We also verify that N steps of the Markov chain results in independent edge instances. All tests are done with four real networks - the neural network

Table 1. Characteristics of the graphs used in this paper. $(|V|, |E|)$ are the numbers of vertices and edges in the graph and G-R statistic is the Gelman-Rubin diagnostic.

Graph name	$(\|V\|, \|E\|)$	G-R diagnostic
C. Elegans	(297, 4296)	1.05
Netscience	(1461, 5484)	1.02
Power	(4941, 13188)	1.006
soc-Epinions1	(75879, 405740)	1.06

of *C. Elegans* [17] (referred to as "C. Elegans"), the power grid of the Western states of US [17] (called "Power"), co-authorship graph of network science researchers [18] (referred to as "Netscience") and a 75,000 vertex graph of the social network at Epinions.com [19] ("soc-Epinions1"). Their details are in Table 1. The first three were obtained from [20] while the fourth was downloaded from [21]. All the graphs were converted to undirected graphs by symmetrizing the edges.

In Fig. 2 we investigate the impact of ϵ in Eq. 4. We generate 1000 samples by running the Markov chain for $1|E|, 5|E|, 10|E|$ and $15|E|$ steps, corresponding to $\epsilon = 0.37, 6.7 \times 10^{-3}, 4.5 \times 10^{-5}$ and 3.06×10^{-7}. The Markov chain is started using the first three networks listed in Table 1. We calculate the global clustering coefficient, the graph diameter and the maximum eigenvalue for each graph and plot their distributions in Fig. 2. We find that for all three, $\epsilon < 5 \times 10^{-3}$ leads to distributions which are very close. We will proceed with $\epsilon = 4.5 \times 10^{-5}$ i.e., we will mix the Markov chain $10|E|$ times before extracting a sample.

We next calculate the fraction of edges that are deemed independent by the test described in Section 3. We run the Markov chain for $K = 1000N$ steps and construct the binary time-series $\{Z_t\}$ for all the edges. Thinned time-series $\{Z_t^k\}, k = N$ are constructed for $N = \{1, 5, 10, \ldots 30\}|E|$ and each time-series tested for independence. In Fig. 3, we plot the fraction of edges deemed independent as a function of $N/|E|$, for "C. Elegans", "Netscience" and "Power". We see that by $N = 10|E|$, more than 95% of the edges test independent, explaining the convergence of the distributions observed in Fig. 2.

The test of independence described in Section 3 can also be used to construct an ensemble of independent graphs, by running a very long Markov chain, and thinning by $k_* > N$, the thinning factor that renders *all* edges independent. Comparisions with graphs generated using $N = 10|E|$ are in [13], and the distributions are found to be very similar. Thus empirically, we find that a Markov chain, run for $10|E|$ steps generates independent, uniformly distributed graphs.

We now address a large graph (soc-Epinions1), where potentially $|V|^2$ distinct edges might be realized during a Markov chain. While $N = 10|E|$ might render a large fraction of edges independent, there may still be a significant number (*not* fraction) of edges that are still correlated with the starting graph. Since certain

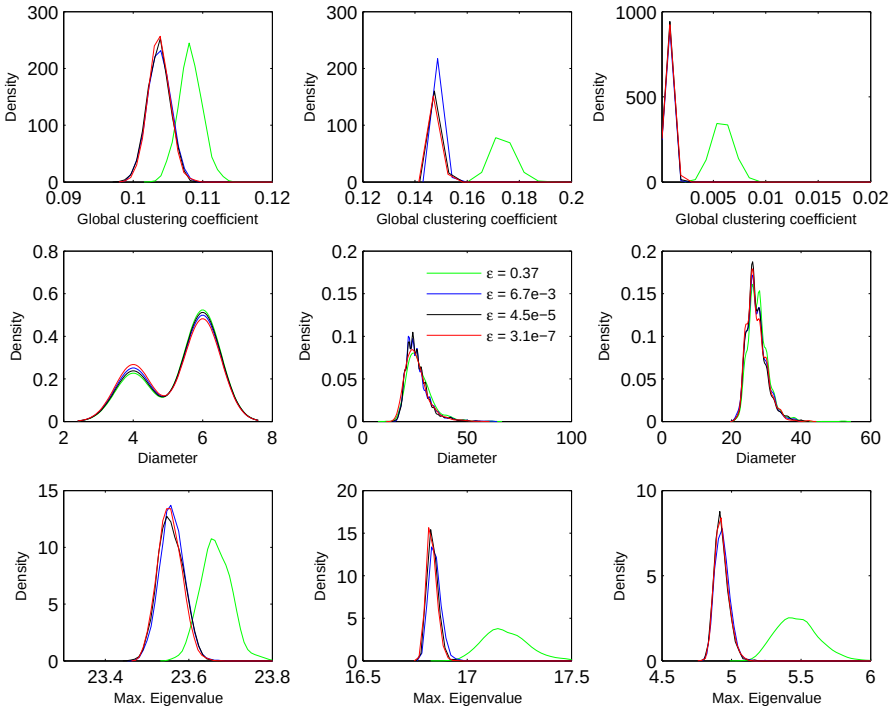

Fig. 2. Plots of the distributions of the global clustering coefficient, the graph diameter and the max eigenvalue of the graph Laplacian for "C. Elegans" (left), "Netscience" (middle) and "Power" (right), evaluated after $1|E|, 5|E|, 10|E|$ and $15|E|$ iterations of the Markov chain (green, blue, black and red lines respectively). The corresponding values of ϵ are in the legend. We see that the distributions converge at $\epsilon \sim 1.0^{-5}$.

graphical metrics, like diameter, can be quite sensitive to edges, we check whether a more stringent N is required for large graphs.

We generate an ensemble of 1000 graphs, starting from soc-Epinions1, using $N = 30|E|$. We also run a long Markov chain ($K = 210,000|E|$), and compute the thinning factor k required to render each of the edges independent. Due to the large number of edges realized during the Markov chain, this was calculated for only $0.1|E|$ (40,574) edges, chosen randomly from all the distinct edges that are realized by the Markov chain. In Fig. 4 (left) we plot the distribution of k obtained from the 40,574 sampled edges. We see that most of the k lie between $10|E|$ and $100|E|$; edges with thinning factors outside that range are about two orders of magnitude less abundant. The largest thinning factor identified was $k = 720|E|$. In Fig. 4 (right) we plot the distribution of diameter obtained using $N = 30|E|$, and compare against the distributions obtained from the long run using thinning factors $k = 5N, 9N$, and $13N$. We see small differences in distributions; for practical purposes, $N = 30|E|$ results in a converged distribution.

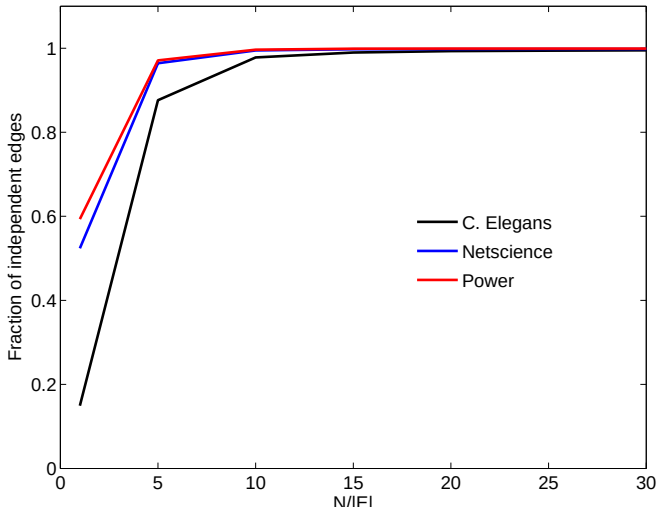

Fig. 3. Fraction of edges testing independent, for "C. Elegans", "Netscience" and "Power" for various values of N. We see that $N = 10|E|$ ensures that at least 95% of the edges become independent.

Finally, we address the question whether the results presented so far are independent of the starting graph. We generate two starting points by marching a Markov chain (initialized by a real network) for $N = 10,000|E|$ steps. We initialize 3 concurrent Markov chains with these graphs, and calculate the Gelman-Rubin (G-R) diagnostic [22] using the binary edge time-series. Values of the diagnostic between 1 and 1.1 indicate that the states of the concurrent Markov chain are not dependent on the starting location. We performed this test for all 4 graphs; the corresponding G-R diagnostics are tabulated in Table 1.

5 Conclusions

We have developed a method that allows one to generate a set of independent realizations of graphs with a prescribed joint degree distribution. The graphs are generated using a MCMC approach, employing the algorithm described in [8] as the "rewiring" mechanism. Our method involves running the Markov chain for N steps before extracting a graph realization; the Markov chain is run repeatedly to generate samples. We developed a model (and a closed-form expression) to estimate N that allows the 2-state Markov chain *of an edge* to converge to its stationary distribution. This is a necessary condition for how long a Markov chain *on the space of graphs* has to be run before an independent graph realization can be extracted from it. We find that $10|E| - 30|E|$ steps are sufficient to generate samples of graphs that provide converged distributions of graphical metrics like clustering coefficients, diameter and maximum eigenvalue of the graph Laplacian.

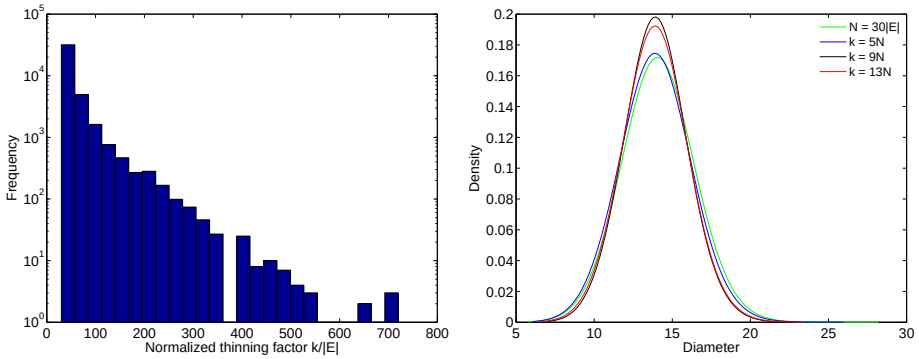

Fig. 4. Left: The normalized thinning factor $k/|E|$ for the soc-Epinions1 graph, as calculated for the 40,574 sampled edges. We see that the most thinning factors are lie in $(10|E|, 100|E|)$. Right: Plot of the graph diameter and distribution generated using $N = 30|E|$ as well as a long Markov chain with thinning factor k equal to various multiples of N. We see that the distributions are very similar.

We verified our model (for N) by constructing a non-parametric test for the independence of each edge. It is not dependent on any heuristics or graphical properties. It uses the time-series of the occurrence/non-occurrence of edges, thins them by N and fits a first-order Markov and an independent sampling model to the thinned time-series. Their BICs are used to perform model selection i.e., to decide whether the thinned chain resembles draws from an independent more than a first-order Markov process. The method is not new, but does not seem to have been used in the generation of independent graphs.

Finally, we repeated our tests with concurrent Markov chains, initialized with dispersed starting graphs. We employed the Gelman-Rubin diagnostic to verify that our tests were not being driven by the starting points of the Markov chain.

While this work enables the generation of independent graphs, including large ones, it has only been demonstrated on graphs where the JDD is preserved. Extending our method to the generation of independent graphs when some other graph property is held constant is currently under investigation.

Acknowledgments

We would like to thank Tamara G. Kolda for many helpful discussions.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5349) (1999) 509–512
2. Newman, M.E.J.: Assortative mixing in networks. Phys. Rev. Lett. **89** (Oct 2002) 208701

3. Holme, P., Zhao, J.: Exploring the assortativity-clustering space of a network's degree sequence. Phys. Rev. E **75** (Apr 2007) 046111
4. Kannan, R., Tetali, P., Vempala, S.: Simple markov-chain algorithms for generating bipartite graphs and tournaments. Random Struct. Algorithms **14**(4) (July 1999) 293–308
5. Jerrum, M., Sinclair, A.: Fast uniform generation of regular graphs. Theor. Comput. Sci. **73**(1) (1990) 91–100
6. Jerrum, M., Sinclair, A., Vigoda, E.: A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. J. ACM **51**(4) (2004) 671–697
7. Gkantsidis, C., Mihail, M., Zegura, E.W.: The Markov chain simulation method for generating connected power law random graphs. ALENEX (2003) 16–25
8. Stanton, I., Pinar, A.: Constructing and sampling graphs with a prescribed joint degree distribution using Markov chains. ACM Journal of Experimental Algorithmics to appear.
9. Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of escherichia coli. Nature Genetics **31** (2002) 64–68
10. Maslov, S., Sneppen, K.: Specificity and stability in topology of protein networks. Science **296**(5569) (2002) 910–913
11. Adams, S.: Dilbert: Random number generator (2001) Available at http://search.dilbert.com/comic/RandomNumberGenerator.
12. Sokal, A.: Monte Carlo methods in statistical mechanics: Foundations and new algorithms (1996)
13. Ray, J., Pinar, A., Seshadhri, C.: Are we there yet? when to stop a markov chain while generating random graphs. CoRR **abs/1202.3473** (2012)
14. Raftery, A., Lewis, S.M.: Implementing MCMC. In Gilks, W.R., Richardson, S., Spiegelhalter, D.J., eds.: Markov Chain Monte Carlo in Practice. Chapman and Hall (1996) 115–130
15. Raftery, A.E., Lewis, S.M.: How many iterations in the Gibbs sampler? In Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M., eds.: Bayesian Statistics. Volume 4., Oxford University Press (1992) 765–766
16. Bishop, Y.M., Fienberg, S.E., Holland, P.W.: Discrete multivariate analysis: Theory and practice. Springer-Verlag, New York, NY (2007)
17. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 440–442
18. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices, 036104. Phys. Rev. E **74** (2006)
19. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic Web. In Fensel, D., Sycara, K., Mylopoulos, J., eds.: The Semantic Web - ISWC 2003. Volume 2870 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2003) 351–368 10.1007/978-3-540-39718-2_23.
20. Newman, M.E.J.: Prof. M. E. J. Newman's collection of graphs at University of Michigan Available at http://www-personal.umich.edu/~mejn/netdata/.
21. Stanford Network Analysis Platform Collection of Graphs: The Epinions social network from the Stanford Network Analysis Platform collection http://snap.stanford.edu/data/soc-Epinions1.html.
22. Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences. Statistical Science **7** (1992) 457–472