RARE EVENTS IN COMPLEX NETWORKS AND THEIR EFFICIENT ESTIMATION

Konstantin M. Zuev, James L. Beck, Steven Wu

SIAM Workshop on Network Science 2014
July 6-7 · Chicago

Summary

The dependence of our society on complex networks constantly grows with the worldwide increase in urbanization and globalization. As a result there is an increasing demand for understanding the behaviour of these networks. In particular, the probabilities of extreme events (rare-occurrence, high-impact) must be accurately estimated. In this work, a new Markov chain Monte Carlo method for efficient estimation of small probabilities of rare events in complex networks is presented.

Problem formulation

A network topology is represented as a graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$ are sets of n nodes and m links, respectively. A *network state* is defined as an m-tuple $s = (s_1, \dots, s_m)$ with $s_i \in [0, 1]$, where $s_i = 1$ if link e_i is fully operational (or "up") and $s_i = 0$ if link e_i is completely failed (or "down"). If $s_i \in (0, 1)$, then link e_i is partially operational. The set of all network states $\mathcal{S}$ is then an m-dimensional hypercube,

$$\mathcal{S} = \{(s_1, \dots, s_m) \mid s_i \in [0, 1]\} = [0, 1]^m \tag{1}$$

Let $\pi(s)$ be a probability distribution on the network state space $\mathcal{S}$ which provides a probability model for the occurrence of different network states, $s \sim \pi(s)$. Furthermore, we define a *performance function* $\mu : \mathcal{S} \to \mathbb{R}$ that quantifies the degree to which the network provides the required service. In the context of networks, μ is typically interpreted as a utility function, i.e. higher values of μ correspond to better network performance. Let us define the *failure domain* $\mathcal{F} \subset \mathcal{S}$ as follows:

$$\mathcal{F} = \{s \in \mathcal{S} \mid \mu(s) < \mu^*\}, \tag{2}$$

where μ^* is the critical threshold.

The *network reliability problem* is to compute the probability of failure $p_{\mathcal{F}}$, that is given by the following integral:

$$p_{\mathcal{F}} = \mathbb{P}(s \in \mathcal{F}) = \int_{\mathcal{S}} \pi(s) I_{\mathcal{F}}(s) ds = \mathbb{E}_\pi[I_{\mathcal{F}}]. \tag{3}$$

Several classical reliability problems [2, 6, 3] are special cases of the above general formulation, e.g. Source-to-Terminal Connectedness, Network Connectedness, Traffic to Central Site, to name but a few.

We make the following real-life assumptions:

(i) *The computational effort for evaluating the network performance function $\mu(s)$ for each state $s \in \mathcal{S}$ is significant*, thereby making the indicator function $I_{\mathcal{F}}(s)$ expensive to compute. Therefore, it is essential to minimize the number of such function evaluations;

(ii) *The number of edges m is large, i.e. $m \gg 1$.* Many actual networks have millions (e.g. road networks), or even billions, of edges (e.g. the Internet);

(iii) *The probability of failure $p_{\mathcal{F}}$ is very small, i.e. $p_{\mathcal{F}} \ll 1$.* Real-life networks are reliable to some extent (otherwise they would not be in use), and their failures are usually *rare events.*

These assumptions make the network reliability problem computationally very challenging.

Subset Simulation method

Subset Simulation was originally developed in [1] for estimation of small failure probabilities of complex civil engineering structures such as tall buildings and bridges at risk from earthquakes. The main idea of the method is to represent a small failure probability $p_{\mathcal{F}}$ as a product $p_{\mathcal{F}} = \prod_{j=1}^{L} p_j$ of larger probabilities $p_j > p_{\mathcal{F}}$, where the factors p_j are estimated sequentially, $p_j \approx \hat{p}_j$ to obtain an estimate $\hat{p}_{\mathcal{F}}$ for $p_{\mathcal{F}}$ as $\hat{p}_{\mathcal{F}} = \prod_{j=1}^{L} \hat{p}_j$. To achieve this goal, let us consider a sequence of nested subsets of the network state space $\mathcal{S}$, starting from the entire space and shrinking to the failure domain:

$$\mathcal{S} = \mathcal{F}_0 \supset \mathcal{F}_1 \supset \ldots \supset \mathcal{F}_L = \mathcal{F} \tag{4}$$

Subsets $\mathcal{F}_0, \dots, \mathcal{F}_{L-1}$ are called *intermediate failure domains.* The failure probability can be written then as a product of conditional probabilities:

$$p_{\mathcal{F}} = \prod_{j=1}^{L} \mathbb{P}(\mathcal{F}_j | \mathcal{F}_{j-1}) = \prod_{j=1}^{L} p_j, \tag{5}$$

where $p_j = \mathbb{P}(\mathcal{F}_j | \mathcal{F}_{j-1})$ is the conditional probability at the $(j-1)^{\text{th}}$ conditional level. Clearly, by choosing the intermediate failure domains $\mathcal{F}_1, \dots, \mathcal{F}_{L-1}$ appropriately, all conditional probabilities $p_1, \dots, p_L$ can be made sufficiently large. The original network reliability problem (estimation of the small failure probability $p_{\mathcal{F}}$) is thus replaced by a sequence of L intermediate problems: estimation of the larger failure probabilities p_j, $j = 1, \dots, L$.

The first probability $p_1 = \mathbb{P}(\mathcal{F}_1|\mathcal{S}) = \mathbb{P}(\mathcal{F}_1)$ can be simply estimated by the Monte Carlo simulation (MCS):

$$\begin{gathered} p_1 \approx \hat{p}_1 = \frac{1}{N} \sum_{i=1}^{N} I_{\mathcal{F}_1}(s_0^{(i)}), \\ s_0^{(i)} \overset{i.i.d.}{\sim} \pi(s|\mathcal{F}_0) \equiv \pi(s) \end{gathered} \tag{6}$$

We assume here that $\mathcal{F}_1$ is chosen in such a way that p_1 is relatively large, so that the MCS estimate (6) is accurate for a moderate sample size N. In actual implementation of the algorithm, the intermediate failure domains $\mathcal{F}_j$ are chosen adaptively.

For $j \geq 2$, to estimate p_j using MSC one needs to simulate i.i.d. samples from conditional distribution $\pi(s|\mathcal{F}_{j-1})$, which, for general $\pi(s)$ and $\mathcal{F}_{j-1}$, is not a trivial task. For example, it would be inefficient to use MCS for this purpose (i.e. to sample from $\pi(s)$ and accept only those samples that belong to $\mathcal{F}_{j-1}$), especially at higher levels. Sampling from $\pi(s|\mathcal{F}_{j-1})$ for $j \geq 2$ can be done by a specifically tailored Markov chain Monte Carlo (MCMC) technique at the expense of generating dependent samples.

The Metropolis-Hastings (MH) algorithm [5], perhaps the most popular MCMC algorithm, suffers from the curse of dimensionality. Namely, it is not efficient in high-dimensional conditional probability spaces, because it produces a Markov chain with very highly correlated states. Therefore, if the total number of network links m is large, then the MH algorithm will be inefficient for sampling from $\pi(s|\mathcal{F}_{j-1})$, where $\mathcal{F}_{j-1} \subset \mathcal{S} = [0, 1]^m$. In Subset Simulation, the Modified Metropolis algorithm (MMA) [1] is used instead for sampling from the conditional distributions $\pi(s|\mathcal{F}_{j-1})$. MMA differs from the MH algorithm in the way the candidate state $\xi = (\xi_1, \dots, \xi_m)$ is generated. Instead of using an m-dimensional proposal PDF on $\mathcal{S}$ to directly obtain the candidate state, in MMA a sequence of univariate proposal PDFs is used. Namely, each coordinate ξ_k of the candidate state is generated separately using a univariate proposal distribution $q_k(s_k|s_{j-1,k}^{(i)})$ dependent on the k^{th} coordinate $s_{j-1,k}^{(i)}$ of the current state. Then a check is made whether the m-variate candidate $\xi \in \mathcal{S}$ generated in such a way belongs to the subset $\mathcal{F}_{j-1}$ in which case it is accepted as the next Markov chain state; otherwise it is rejected and the current MCMC sample is repeated. For details on MMA, we refer the reader to the original paper [1] and to [7] where the algorithm is discussed in depth.

Let us assume now that we are given $k < N$ seeds $s_{j-1}^{(1)}, \dots, s_{j-1}^{(k)} \sim \pi(s|\mathcal{F}_{j-1})$, where $j = 2, \dots, L$. Then, using MMA, we can generate k Markov chains with the total number of N states starting from these seeds and construct an estimate for p_j similar to (6), where MCS samples are replaced by MCMC samples:

$$\begin{gathered} p_j \approx \hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} I_{\mathcal{F}_j}(s_{j-1}^{(i)}), \\ s_{j-1}^{(i)} \overset{MMA}{\sim} \pi(s|\mathcal{F}_{j-1}) \end{gathered} \tag{7}$$

Note that all samples $s_{j-1}^{(1)}, \dots, s_{j-1}^{(N)}$ in (7) are identically distributed in the stationary state of the Markov chain, but are not independent. Nevertheless, these MCMC samples can be used for statistical averaging as if they were i.i.d., although with some reduction in efficiency [4].

Finally, Subset Simulation uses the estimates (6) for p_1 and (7) for p_j, $j \geq 2$, to obtain the estimate for the failure probability:

$$p_{\mathcal{F}} \approx \hat{p}_{\mathcal{F}} = \prod_{j=1}^{L} \hat{p}_j \tag{8}$$

The efficiency of the method will be demonstrated with illustrative examples where small-world network models are compared in terms of reliability of networks they produce.

References

[1] S. Au and J. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Prob Eng Mech*, 16:263–277, 2001.

[2] M. Ball, C. Colbourn, and J. Provan. Network reliability. *Handbooks in OR and MS*, 7:673–762, 1995.

[3] C. Colbourn. *The combinatorics of network reliability.* Oxford University Press, New York, USA, 1987.

[4] J. Doob. *Stochastic processes.* Wiley, New York, 1953.

[5] W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[6] A. Rosenthal. Computing the reliability of complex networks. *SIAM Journal on Applied Mathematics*, 32:384–393, 1977.

[7] K. Zuev, J. Beck, S. Au, and L. Katafygiotis. Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers and Structures*, 92–93:283–296, 2012.