Feature Based Network Sampling
Discussion paper

Christian Franssen & Bernd Heidergott & Ines Linder

October 18, 2020

1 Introduction

Sampling of networks can be of interest for a variety of reasons. If only limited data is available, sampling of networks helps constructing networks using this data. It also allows us to identify other networks with similar characteristics. Sampling of networks is considered a difficult task, as most network measures yield a large and often seemingly unsolvable system of interdependencies, and the area of network sampling is an active area of research. Main research streams in this area are Exponential Random Graphs (ERG), where Maximum Likelihood is applied to fit a probabilistic model to the observed network statistics, and the encoder-decoder approach from machine learning, where a given network is represented by a low-dimensional vector (encoding) and then from this information a network is constructed (decoding). In this research we introduce a new method for sampling networks: Feature Based Network Sampling (FBNS). FBNS constructs random networks satisfying pre-specified network measures, also called structural features. These networks are directed and weighted by construction, but can be transformed into undirected and unweighted networks using an adjusted loss function. In network theory, networks model entities and the connections between these entities. We distinguish undirected and directed networks, depending on the symmetric or asymmetric nature of connections between entities. Furthermore, we differentiate unweighted and weighted networks. In contrast to unweighted networks, weighted networks make a distinction in the importance of connections. Simulation of networks finds its origin in studies by Erdös and R´enyi and Gilbert. Erdös and R´enyi introduced a model where all networks on a fixed entity set with a fixed number of connections are equally likely. Gilbert proposed a similar random network model, where each connection has a fixed probability for its presence. Simulation of networks by these models is straightforward as the presence of a connection is independent of the presence of other connections. Other network measures such as the degree distribution and assortativity come with strong dependencies between connections. Therefore, simulation of networks satisfying these network measures is more complicated.

2 What we have done so far

We define a network $G(V, E)$, where $V = \{1, 2, ..., n\}$ is the set of nodes representing entities and $(i, j) \in E$ if there is an edge representing a connection between nodes i and j. The size of the network is determined by the number of nodes. We call a pair of nodes neighbours if there is an edge between them. The degree of a node is defined by the number of its neighbours. For directed networks, we denote the in-degree as the total incoming neighbours and the out-degree for the total outgoing neighbours. We call a network simple when there is at most one (directed) edge between each pair of nodes and no edge between a node and itself (loops). Furthermore, we call a network complete when all possible connections are present. In the case of a weighted network, we have a function: $A : E \mapsto \mathbb{R}$ which maps every edge to a positive value , and we call matrix A the *weight matrix.* A mapping $h : A \mapsto \mathbb{R}$ is called a *graph statistic*, and examples are the number of triangles in the graph, or the out degree of a particular node.

Consider a given graph $\mathbf{Y}$, i.e., weight matrix $A_{\mathbf{Y}}$, together with the ensuing graph statistics $H(\mathbf{Y}) = (h_1(\mathbf{Y}), \ldots, h_k(\mathbf{Y}))$, for some k. The Feature Based Network Sampling problem is to find a graph $\mathbf{X}$, i.e., weight matrix $A_{\mathbf{X}}$ such that $\mathbf{X}$ is a miniseries of

$$L = (X, \mathbf{Y}) := \sum_{i=1}^{k} (h_i(A_X) - h_i(A_{\mathbf{Y}})^2.$$

Note that the graph statistics typically do not uniquely define a graph, and the set

$$S(\mathbf{Y}) = \{X : L(X, \mathbf{Y}) = \min_{Z} L(Z, \mathbf{Y})\}$$

contains a whole class of graphs. For finding samples of minimizers for the above loss function, i.e., elements in $S(\mathbf{Y})$, we use a steepest descent algorithm. The basic setup is to start initially with a random graph U, and apply a steepest descent algorithm for solving

$$\min_{X} L(X, \mathbf{Y}).$$

We developed a numerical efficient algorithm for finding minimizers. [This is not straightforward and uses some twists and tweaks.] Starting with initial random graphs yields samples in $S(\mathbf{Y})$. Mathematically speaking, we compute the projection of the initial random graph on the class of graphs that have the same graph characteristics as $\mathbf{Y}$.

3 Joint Research Possibilities

We believe that the application of fitting via a steepest descent algorithm has numerical and theoretical value. First, it provides an alternative to ERGs which usually have to refer to MCMC for estimating the normalizing constant. Secondly, it is a rather simple algorithm for producing graphs that are indistinguishable with respect to H. So, showing $X \in S(\mathbf{Y})$ rather than $\mathbf{Y}$ addresses the privacy issue or anonymization issue in publishing graphs. Thirdly, this algorithm allows for sampling graphs with pre-specified connectivity (e.g., Kemeny constant) and cluster structure.

From a mathematical perspective, a key challenge is to understand the statistical properties of $S(\mathbf{Y})$. Is it possible to enhance the algorithm so that we can guarantee that the samples are uniform on $S(\mathbf{Y})$? Maybe by somehow having the negative entropy has a loss? Another question is how to extend this approach to sampling of weighted graphs with co-variates. We believe that the machinery we develop may also be of use in the detection of anomalies. Lastly, given graphs measured at different points in time. Can we use FBNS to fit a Q matrix that best explains the observed networks by means of a continuous time Markov chain?