Todo list

Degree distribution in GCC

Benoît Richard Guiyuan Shi

September 3, 2018

Abstract

Write abstract

1 Introduction

Studying the fundamental properties of networks require to be able to abstract from the particular examples found in nature. This is usually done [?] by using a random model for the network generation and averaging the properties of interest over the set of possible networks. One common model is the configuration model [?] that allows to uniformly sample the space of all networks with a given degree distribution [?]. However, many real examples of networks are connected, as for example the World Wide Web or railroad networks, but no model known to us allows to uniformly sample the space of all connected networks of a given degree distribution.

Missing ref

Missing ref

Missing ref

Find other examples

A way to still study connected networks is to only consider the Giant Connected Components (GCC) of networks generated using the configuration model [?]. We study here how this method implies bias on the degree distribution of the GCC and propose an algorithm based on this knowledge to generate connected networks of arbitrary degree distribution.

Missing ref

2 Degree distribution in GCC

Per Bayes theorem we have for two random events A and B

$$P(A|B) = P(B|A)\frac{P(A)}{P(B)}. \quad (1)$$

{Bayes theorem}

We can apply it to compute the probability r_k that a vertex in the GCC has degree k

$$\begin{aligned} r_k &= P\left(deg(v) = k | v \in GCC\right) && (2)\\ &= P(v \in GCC | deg(v) = k)\frac{P(deg(v) = k)}{P(v \in GCC)} && (3)\\ &= (1 - P(v \notin GCC | deg(v) = k))\frac{p_k}{S} && (4)\\ &= \frac{p_k}{S}(1 - u^k). && (5)\end{aligned}$$

{Degree distribution in GCC}

where S is the probability that a random node is part of the GCC, p_k is the probability that a node has degree k and u is the probability that a node reached by following an edge of the network is not part of the GCC.

Justification of the equality $P(v \notin GCC|deg(v) = k) = u^k$ can be found inf Ref. [1].

In the context of the configuration model we choose the probabilities p_k which determine u and S through the following equations [1]

$$u = \frac{\sum_{k=1}^{\infty} k p_k u^{k-1}}{\sum_{k=1}^{\infty} k p_k} \tag{6}$$

{Expression for u}

$$S = 1 - \sum_{k=1}^{\infty} p_k u^k, \tag{7}$$

{Expression for S}

thus eliminating all unknown in eq. (5).

In conclusion we see that considering a vertex in the GCC biases the probability that it has degree k by a factor $(1 - u^k)/S$ as compared to choosing a vertex uniformly in the network. Since both u and S are smaller than 1, the net effect is to lower the proportion of low degree vertices in the GCC and thus to increase the proportion of high degree vertices.

3 Generating connected networks

3.1 Algorithm

The knowledge of the degree distribution in the GCC can be used generate a connected component of a given degree distribution r_k as follow: we first determine a degree distribution p_k fulfilling eq. (5) for some target degree distribution r_k. Then we generate a network with degree distribution p_k using the configuration model. Finally we take its GCC as our connected network. By construction the vertices in the GCC will have degree distribution r_k. Determining the factors p_k is not immediate however since u is an unknown which is itself a function of p_k. We propose an algorithm to determine it numerically.

First we isolate p_k from eq. (5) to get

$$p_k = S\pi_k(u), \quad \text{with} \quad \pi_k(z) = \frac{r_k}{1 - z^k} \tag{8}$$

Inserting this in the expression(6) for u, we get

$$u = \frac{\sum_{k=1}^{\infty} k\pi_k(u) u^{k-1}}{\sum_{k=1}^{\infty} k\pi_k(u)}. \tag{9}$$

{Fixpoint equation for u}

Therefore u is a fixpoint of the function

$$\mu(z) = \frac{\sum_{k=1}^{\infty} k\pi_k(z) z^{k-1}}{\sum_{k=1}^{\infty} k\pi_k(z)}, \tag{10}$$

{Defition of mu}

which is fully determined by the GCC degree distribution r_k. Note that for $r_1 = 0$, we have the fixpoint $u = 0$ and $p_k = r_k$ for all k. This is consistent with the fact that small component of a network produced with the configuration model have a probability 0 to have loop [1]. Indeed if $p_1 = 0$ all components must have loops, therefore the probability to have small components is 0 as well.

On the other hand $r_1 > 0$ implies $u > 0$. To approximate its value we define the sequence $u_{j+1} = \mu(u_j)$, with $u_0 = r_1$. This sequence will converge toward u for large j. A proof of this statement is given in Appendix A.

In practice we can not deal evaluate infinite sums numerically, thus we need to choose a cutoff index K for the sums such that

$$\sum_{k=K+1}^{\infty} k\pi_k(u) \ll 1. \tag{11}$$

For scale-free network with exponent between ?? and ??, this sum always diverges and thus this method is not applicable.

Find correct exponent

Once u is approximated, we can compute the first K probabilities p_k, which is sufficient to sample random numbers between 1 and K with relative probability p_k. If K is chosen such that $r_k << 1$ for $k > K$, the degree distribution in the GCC closely approximate the distribution r_k.

3.2 Erdos-Renyi reconstruction

In order to test the algorithm presented, we choose the target connected degree distribution r_k to be the degree distribution of the GCC of an Erdos-Renyi network. It is then expected that the reconstructed p_k closely approximate a Poisson degree distribution.

The probability to have degree k in an Erdos-Renyi network is

$$p_k = \frac{c^k}{k!} e^{-c}, \tag{12}$$

where c is the mean degree in the network. Using eq. (6) and (7) to find u and S yield everything we need to be able to determine the GCC degree distribution r_k from eq. (5). We can therefore use the algorithm on these r_k.

When computing S for the original Poisson distribution, we should however be cautious, as the reconstructed p_0 will always be 0. The expected result, correctly normalized, is therefore

$$p_k = \frac{c^k}{k!} \frac{1}{e^c - 1}. \tag{13}$$

The expected bias ratio r_k/p_k is shown for various mean degree c and a cutoff constant $K = 10000$ in fig. 1 together with the same value computed from the algorithm presented above. As it can be seen, the agreement is very good. During the computations it has been observed that the closer the mean degree is to the critical value $c = 1$, the slower the fixpoint iteration converges.

4 Discussion

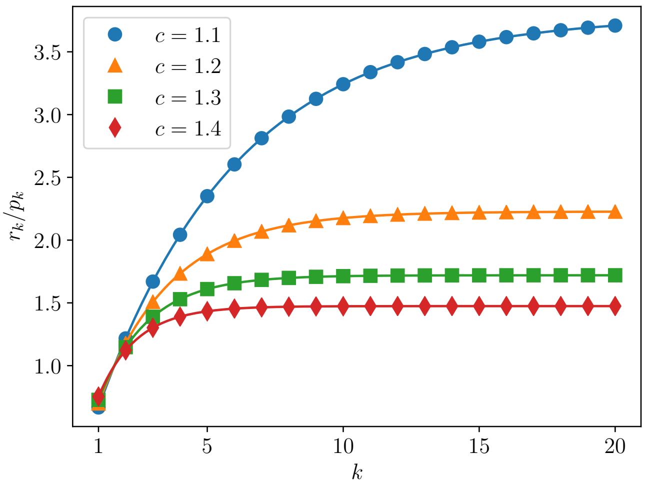
Write discussion

A Convergence of the fixpoint iteration

{Appendix: Fixpoint convergence}

First notice that the case $r_1 = 0$ is trivial, as described in the main text. We will therefore assume in this Appendix that $r_1 > 0$, immediately giving $\mu(0) = r_1 > 0$. Second, not that (10) tells us that $z < 1$ implies $\mu(z) < 1$. From there we separate two cases:

If $u = 1$ is the unique solution of eq. (9) then $\mu(z)$ must be continuous for $z \in [0, 1]$ and $\mu'(1) < 1$, making $u = 1$ and attractive fixpoint.



{Figure: Erdos-Renyi reconstruction}

Figure 1: Bias factor r_k/p_k for r_k being the degree distribution of the GCC of an Erdos-Renyi network with various mean degree c and cutoff constant $K = 10000$. The p_k have been determined using the algorithm presented in the text. Solid line is the expected value $(1 - u^k)/S$ for the bias factor.

On the other hand if there is another solution u^* to eq. (9), it is the unique solution with $0 \leq u^* < 1$ since $\mu(z)$ is an increasing function of z, as it is demonstrated in Appendix B. Moreover, since $\mu(0) > 0$ we have $\mu'(u^*) < 1$, which makes it an attracting fixpoint and makes $u = 1$ a repulsive one.

We can thus conclude that the fixpoint iteration proposed always converges and converges to the degenerate case $u = 1$ only if it is the unique possibility.

B Monotonicity of $\mu(z)$

{Appendix: Monotonicity}

To prove that $\mu(z)$ is an increasing function, we compute its derivative with respect to z, which yields

$$\mu'(z) = \left[\sum_{k=1}^{\infty} k\pi_k(z)\right]^{-2} (s_1(z) + s_2(z)) \tag{14}$$

$$s_1(z) = \sum_{j,k} kj\pi'_k(z)\pi_j(z)\left(z^{k-1} - z^{j-1}\right) \tag{15}$$

$$s_2(z) = \sum_{j,k} k(k-1)j\pi_k(z)\pi_j(z)z^{k-2}. \tag{16}$$

The sum $s_1(z)$ can be rewritten as

$$s_1(z) = \sum_{j>k} kj \left(\pi_k'(z)\pi_j(z) - \pi_j'(z)\pi_k(z)\right) \left(z^{k-1} - z^{j-1}\right) \tag{17}$$

$$= \sum_{j>k} \frac{kr_k}{1-z^k} \frac{jr_j}{1-z^j} \frac{z^k - z^j}{z^2} \left(\frac{k}{z^{-k}-1} - \frac{j}{z^{-j}-1}\right) \tag{18}$$

$$= \sum_{j>k} kj\pi_k(z)\pi_j(z) \frac{z^k - z^j}{z^2} \left(\frac{k}{z^{-k}-1} - \frac{j}{z^{-j}-1}\right). \tag{19}$$

Using the fact that the function

$$f_z(\lambda) = \frac{\lambda}{z^{-\lambda} - 1} \tag{20}$$

is a decreasing function of λ we can see that for $z \in [0, 1)$ and $j > k$ we have

$$z^k - z^j \geq 0 \tag{21}$$

$$\frac{k}{z^{-k}-1} - \frac{j}{z^{-j}-1} \geq 0, \tag{22}$$

and thus $s_1(z) \geq 0$. Moreover each terms in $s_2(z)$ is non-negative, so we have $s_2(z) \geq 0$. We can therefore conclude that $\mu'(z) \geq 0$ and thus that $\mu(z)$ is an increasing function of z.

References

[1] M. Newman. *Networks: an introduction.* 2010.