A model petting zoo for interacting with network structure

Leto Peel based on peer reviews by 2 anonymous reviewers

If you work, study or play in network science then chances are you have generated a network. Whether or not you have a real-world system to analyse, synthetic networks play an important role in network science. Generating networks of a chosen size can provide a null model for a statistical test, a test bed for new algorithms or the basis for studying the interplay between structure and dynamics in complex systems. Consequently network science literature contains a wide array of network models: some designed as processes to replicate observed properties and others for the purposes of statistical inference. However, these models have different parameters and constraints associated with their generative models, may or may not have the ability to control for random noise and do not always have readily available software implementations, thus making them unavailable to network science practitioners.

The article of Cazabet et al. (2023) introduces a software "zoo," called Structify-Net, that contains a range of models that the authors have captured from the wild. The authors have focused on developing a framework that enables the generation networks of a chosen size, according to number of nodes and edges, and provides the means to control for randomness, by interpolating between the specified structure and a random graph. The article also discusses an interesting use case to examine the interplay between network structure and node attributes, which might compliment methods based on permutation tests (Bianconi et al. 2009, Ehrhardt and Wolfe 2019).

Structify-Net presents some interesting future opportunities. For instance, the independence that Structify-Net imposes on edge ranking (defined by the model) and the expected number of edges (defined by the user) might offer a route towards exploring network growth or evolution. Like any zoo Structify-Net is not complete in that there are many more exotic "species" that the authors, or perhaps others in the network science community, may later collect. Collecting more model implementations to align with reviews of network models (Goldenberg et al. 2010) together with methods of statistical inference has the potential to lay

the foundations for the ever important bridge between theory and practice in network science (Peel et al. 2022).

References:

Bianconi, Ginestra, Paolo Pin, and Matteo Marsili (2009) Assessing the Relevance of Node Features for Network Structure. Proceedings of the National Academy of Sciences 106, 28: 11433–38. https://doi.org/10.1073/pnas.0811511106

Cazabet, Remy, Salvatore Citraro, and Giulio Rossetti (2023) Structify-Net: Random Graph Generation with Controlled Size and Customized Structure. arXiv, ver. 2 peer-reviewed and recommended by Peer Community in Network Science. https://doi.org/10.48550/arXiv.2306.05274

Ehrhardt, Beate, and Patrick J. Wolfe (2019) Network Modularity in the Presence of Covariates'. SIAM Review 61, 2: 261–76. https://doi.org/10.1137/17M1111528

Goldenberg, Anna, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi (2010) A Survey of Statistical Network Models. Foundations and Trends in Machine Learning 2, 2: 129–233. https://doi.org/10.1561/2200000005

Peel, Leto, Tiago P. Peixoto, and Manlio De Domenico (2022) Statistical Inference Links Data and Theory in Network Science. Nature Communications 13, 1: 6794. https://doi.org/10.1038/s41467-022-34267-9

Reviews

Evaluation round #2

Reviewed by anonymous reviewer 1, 06 September 2023

I think the authors have thoughtfully addressed my comments, and I don't have any other revisions I would recommend before publication.

Reviewed by anonymous reviewer 2, 06 September 2023

I thank the authors for their careful consideration of my previous comments and their work on improving the manuscript. I find the purpose of the paper much clearer and the link to other works better explained. I believe the paper could be published as it is, although I still have two minor suggestions.

1. I would personnally prefer putting the description of related works before the introduction of the new framework. As a reader, I always find it easier to know what has been done before to better understand the value of something new.

2. The authors now explain well how their package is different from other netwrok generator packages. Maybe it would be good to also mention why the package is useful for researchers who think of making their own code directly (how would the functions built in the package make their life easier? Are there benefits to having everyone using the same software? etc.)

I wish the authors good luck with this work!

Evaluation round #1

Authors' reply, 06 September 2023

Download author's reply

Decision by Leto Peel, posted 29 June 2023, validated 30 June 2023

Revision recommended

I think this preprint shows potential and would like to recommend a revised version of it. The current version falls a bit short. The reviewers make some very nice suggestions to this end. In particular, I'd like to emphasize the point about connecting better with previous literature. It seems there are a number of models that could be considered as part of the current version (e.g., see Peter Hoff's work on latent space models), so not pointing this out seems like a missed opportunity. Providing a summary of features/benefits over other packages that involve network generation would also help promote the proposed software.

Reviewed by anonymous reviewer 1, 16 June 2023

In this manuscript the authors develop a method for generating networks with a given number of nodes (n) and number of edges (m), which have planted structure according to a ranking of the node pairs and a non-decreasing probability function mapping these ranks to edge existence probabilities. The authors construct a parameterization for their probability function which allows one to interpolate between the extremes of a completely random graph (equal probability for all edge pairs, epsilon = 1) and a completely deterministic graph given the rankings (probability 1 for the m highest ranking node pairs and 0 elsewhere, epsilon = 0). They then construct a collection ("structure zoo") of different ranking matrices that can be used along with their tunable probability function to generate networks with a wide variety of planted structures and any desired level of noise. They conduct experiments to examine the effect of changing epsilon on the small-worldness of the generated graphs, using different rank matrices from the structure zoo, in a manner reminiscent of the original small-world experiment of Watts and Strogatz. The authors package their method and structure zoo into an easy-to-use Python package and make their code for the article available as well.

For the most part, the method and experiments are clearly presented and easy to follow. However, I do think the authors could beef up their literature review and motivational examples in order to better place their method in context with previous work. I also think it would be nice if the authors could clarify how their method can be used in conjunction with real data given its limitations for statistical inference.

A few more specific points along these lines:

(1) The proposed method has many similarities to a graphon model, and I think the paper could be improved with some discussion of the graphon literature to demonstrate why the proposed method should be preferred. Perhaps one advantage of the proposed method is that it has a more easily controlled noise level?

(2) On a similar note, the graphs being generated are specific instances of inhomogeneous random graphs, so it could be worth discussing that literature further to identify any similar previous work and how the proposed method improves upon it.

(3) I think I generally understand the authors' use of the rational Bezier curve as a means of interpolating between the two extremes of complete equality and inequality in the distribution of the probabilities over the edge pairs while fixing m. But how did they decide on this family of curves for interpolation? Is it the most "natural" in some sense? Or does its derivative have a particularly nice form? I'm curious to know because I hadn't heard of these functions before reading the paper.

(4) In the motivation section, the authors highlight diffusion as a particular process for which understanding the role of network structure is important, but there are many other example applications they could mention (e.g. synchronization, percolation). By singling out diffusion it appears like the proposed method has a more limited set of applications than it does, and I think the motivation would be strengthened by adding some additional examples.

(5) The authors do a good job of pointing out the limitations of their method at the end of the paper, in particular the scalability and the independence assumption they make. I would also add here the challenges resulting from the poor compatibility of this method with parameter estimation techniques, necessitating the comparison of ad hoc summary statistics as discussed in the last paragraph.

(6) I think it would be interesting to see how the detectability of community structure varies with epsilon for block-structured rank matrices. Does this have a clean phase transition like in the standard detectability setting? Not a suggestion for publication, just a thought for future studies.

(7) A very minor point that I found confusing was that the authors said epsilon "controls how strongly the random graph is driven by the community structure." I think they should change the wording to something like "planted structure" or "rank structure", since their method can represent much more than community structure.

(8) Another very minor point for presentation is that I found the in-text citation format a bit confusing. There should be brackets or parentheses or something around the in-text citations to separate them from the other text.

I enjoyed reading this paper, and I hope the authors find my suggestions helpful.

Reviewed by anonymous reviewer 2, 20 June 2023

This article proposes a general framework for the generation of random networks along with the corresponding Python package. The main contribution of the paper is to spell out a definition of a random graph that can be further specified to reproduce well-known models (SBMs, small-world networks, etc.). While the idea of building an umbrella framework for the bestiary of random graphs we know seems good, there are several problems that should be adressed before publication.

Major comments:

- The theoretical contribution of the paper is not clear enough. The paper does not explain the intuition behind the general generative process that it proposes (the two steps on page 3), and most importantly what the rank function may represent or may be interpreted. Right now, it is not clear why one would use this framework instead of directly using specific and already-known models.

- Linked to this, it is not clear what the Structify-net package adds to the existing software tools we already have. In particular, the graph generator functions in NetworkX seem to already do a fine job in helping users to generate the graphs they are interested in. The authors should show what their tool compares to already existing solutions and why it is a useful addition.

- The application on replicating the Watts-Strogatz experiment does not seem to be the best way to demonstrate the value of the framework, because it is very specific. A better way could be to compare the results of the graph generator to several classic graph generators (either theoretically, looking at graph characteristics that are produced, such as clustering, diameter, etc., or comparing software capabilities, such as computing time).

Minor comments:

- Page 2, the structure of the section on "Motivational examples" is very short and discusses somewhat random problems. The authors may want to develop better this section to explain better cases where random graphs are useful, such that they can use these cases later in the paper (the examples in the "Structure Zoo"

section go in all directions and it might be useful for the reader to have a bit of background before hand).

- Page 2, there are many fields where the Erdös-Rényi model is definitely not the most commonly used. Maybe the most known, or elementary?

    - Page 4, "P(r) = p, p in [0,1]" reads as: the probability function is a constant p.

    - Page 4, what is the rationale behind using Bézier curves (accent missing)? What do the endpoints and control point represent?

    - The sub-sections on block structures page 6 are straightforward and could be shortened.

    - In comparison, the sections on star structure and core-preiphery page 7 could be better developed. Are the formulations for the rank functions proposed the only ones possible? How were they derived?

    - Why are fractal models useful? (see first minor comment)

    - Page 8, "hierarchical structure" in networks has been defined in many ways and by many other papers, and hierarchy in networks is a vibrant topic in social network analysis...

    - Check typos: page 2,"heterogeneityBarthélémy", page 4, "Section describes"...