The Journal of Open Source Software

PANINIpy: Package of Algorithms for Nonparametric Inference with Networks In Python

Alec Kirkley [1,2,3] **and Baiyue He** [1]

1 Institute of Data Science, University of Hong Kong, Hong Kong **2** Department of Urban Planning and Design, University of Hong Kong, Hong Kong **3** Urban Systems Institute, University of Hong Kong, Hong Kong

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Complex networks provide a highly flexible representation of the relational structure within a variety of real-world systems, from city streets to the Internet [Barabási 2016]. The topology and dynamics of real network data are often too complex to summarize or visualize using traditional data analysis methods, which has triggered a substantial research movement within multiple fields—including physics, computer science, sociology, mathematics, and economics among others—to develop new tools for statistical inference and machine learning specifically suited for networks.

Research on complex network inference has the goal of learning meaningful structural and dynamical regularities in network data in a manner that is often independent of the particular application of interest but relies on fundamental principles that govern a wide range of networked systems, such as transitivity, degree heterogeneity, and assortativity (Newman, 2018). A substantial amount of research within complex network inference over the last two decades has focused on the construction and evaluation of algorithms for *community detection*—the task of inferring groups of nodes that exhibit particularly strong connectivity or that have shared roles or features (Fortunato, 2010). Another major area of interest is *network reconstruction* (Peel et al., 2022), which aims to infer statistically significant functional connections from time series or other activity patterns as well as identify spurious correlations and missing edges in partially observed noisy network data. A third notable focus area within complex network inference is the clustering of network populations or multilayer networks arising in longitudinal and cross-sectional studies (Young et al., 2022).

Although community detection, network reconstruction, and network population clustering are some of the most widely researched areas in complex network inference, there are a broad range of tasks for which there is active development of new methods. For example, there is a large new body of work aimed at inferring statistically significant structure in higher order networks (Battiston et al., 2021) and networks with different types of metadata on the nodes and/or edges (Fajardo-Fontiveros et al., 2022).

Statement of Need

Due to their discrete, relational, and heterogeneous nature, complex networks present new obstacles for statistical inference. Many inference objectives on networks are intrinsically combinatorial and produce complex summaries in the form of sets or partitions. These factors make scalability and interpretability of critical importance for practical algorithms, which are not often easily accommodated within learning frameworks that focus on continuous ordered data. There are also a number of ways uncertainty can be introduced in the collection of a network dataset, whether through measurement error, sampling bias, or fluctuations across

experimental settings in longitudinal or cross-sectional studies. These factors underscore the importance of developing new principled and flexible methods for extracting structural and dynamical regularities in networks that do not rely on ad hoc parameter choices or heuristics, allowing them to be robust in the presence of noise.

PANINIpy is a flexible and easy-to-use collection of nonparametric statistical inference methods for unsupervised learning with network data. These methods are unified in their motivation from fundamental principles–currently, the Minimum Description Length (MDL) principle underlies all the methods in PANINIpy—and their lack of dependence on arbitrary parameter choices that can impose unwanted biases in inference results. PANINIpy is highly accessible for practitioners as its methods do not require the user to manually tune any input parameters and everything is written from scratch in pure Python to be optimized for each task of interest without reliance on existing packages. PANINIpy therefore provides an important complement to existing network analysis packages in Python such as NetworkX (Hagberg et al., 2008) that focus primarily on network metrics, network visualization, and community detection.

Related Software Packages

There are number of existing Python packages containing individual methods that perform nonparametric inference with networks, but none that are unified under this scope with the ease-to-use pure Python implementation of PANINIpy. The Graph-Tool (Peixoto, 2014) package includes a number of flexible, principled Bayesian methods for complex network inference, many of which are nonparametric. As its core functionalities are implemented in C++, Graph-tool is highly efficient given the computational demand of the inference problems it considers. Graph-tool relies on unique data structures and Markov chain Monte Carlo methods for greater speed and flexibility, but these features are often challenging for new users to navigate. Other popular packages such as NetworkX (Hagberg et al., 2008) and iGraph (Csardi & Nepusz, 2006) also have methods for complex network inference—largely for the task of community detection—but are much broader in scope, being used primarily for network summary statistics and visualization. PANINIpy fills an important gap in the software space for network inference methods with very simple dependencies in pure Python.

Current Modules

Modules can be flexibly added to the package as needed. All modules take as input a standard representation of a network (either as an edgelist or an adjacency list in Python). The existing modules at the time of this publication are:

- **hypergraph_binning**: Methods for identifying MDL-optimal temporally contiguous partitions of event data between distinct node sets (e.g. users and products). Utilizes method of (A. Kirkley, 2024b).
- **population_clustering**: Methods for performing clustering of observed network populations, multilayer network layers, or temporal networks. Utilizes method of (A. Kirkley et al., 2023). Also includes method for generating synthetic network populations using the method of (Young et al., 2022).
- **distributional_regionalization**: Methods for performing MDL-based regionalization on distributional (e.g. census) data over space. Utilizes method of (A. Kirkley, 2022).
- **hub_identification**: Methods for inferring hub nodes in a network using different information theoretic criteria. Utilizes method of (A. Kirkley, 2024a).
- **community_regionalization**: Perform contiguous regionalization of spatial network data, applicable to a wide class of community detection objectives. Utilizes method of (Morel-Balbi & Kirkley, 2024).
- **network_backbones**: Perform global and local network backboning for a weighted network. Utilizes method of (Alec Kirkley, 2024).

Please refer to the documentation for details on the methodology, implementation, and usage for each module.

Acknowledgments

This work was supported by an HKU Urban Systems Institute Fellowship Grant and the Hong Kong Research Grants Council under ECS–27302523 and GRF-17301024.

References

Battiston, F., Amico, E., Barrat, A., Bianconi, G., Ferraz de Arruda, G., Franceschiello, B., Iacopini, I., Kéfi, S., Latora, V., Moreno, Y., & others. (2021). The physics of higher-order interactions in complex systems. *Nature Physics*, *17*(10), 1093–1098.

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Fajardo-Fontiveros, O., Guimerà, R., & Sales-Pardo, M. (2022). Node metadata can produce predictability crossovers in network inference problems. *Physical Review X*, *12*(1), 011010.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*(3-5), 75–174.

Hagberg, A., Swart, P. J., & Schult, D. A. (2008). *Exploring network structure, dynamics, and function using NetworkX*. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Kirkley, A. (2022). Spatial regionalization based on optimal information compression. *Communications Physics*, *5*, 249.

Kirkley, Alec. (2024). *Fast nonparametric inference of network backbones for graph sparsification*. https://arxiv.org/abs/2409.06417

Kirkley, A. (2024a). Identifying hubs in directed networks. *Physical Review E*, *109*, 034310.

Kirkley, A. (2024b). Inference of dynamic hypergraph representations in temporal interaction data. *Physical Review E*, *109*, 054306.

Kirkley, A., Rojas, A., Rosvall, M., & Young, J.-G. (2023). Compressing network populations with modal networks reveals structural diversity. *Communications Physics*, *6*, 148.

Morel-Balbi, S., & Kirkley, A. (2024). Bayesian regionalization of urban mobility networks. *Physical Review Research*, *6*, 033307.

Newman, M. (2018). *Networks*. Oxford University Press.

Peel, L., Peixoto, T. P., & De Domenico, M. (2022). Statistical inference links data and theory in network science. *Nature Communications*, *13*(1), 6794.

Peixoto, T. P. (2014). The graph-tool python library. *Figshare*. https://doi.org/10.6084/m9.figshare.1164194

Young, J.-G., Kirkley, A., & Newman, M. E. (2022). Clustering of heterogeneous populations of networks. *Physical Review E*, *105*(1), 014312.