From Graphs to Probability Distributions: A Hands-On Introduction to ERGMs

Tom Talpir

From molecules and neural networks in the brain to social networks, graphs provide a natural way to represent interactions between components. Understanding the structural properties of these graphs, and the design principles that govern their connectivity, is therefore crucial for understanding the phenomena they describe.

For example, suppose we are given a *C. elegans* connectome and are interested in studying the rules and patterns underlying its connectivity. In this tutorial we will learn to build a model that assigns a probability to every possible graph as a function of a set of structural features. Such a model can allow us to find which structural features are truly important for explaining the observed connectivity. Exponential Random Graph Models (ERGMs) are a class of statistical models designed for exactly this purpose. An ERGM defines a probability distribution over graphs, where the likelihood of a graph depends on a set of chosen structural features (such as the number of reciprocal connections), weighted by model parameters that quantify their importance [1].

The goal of this tutorial is to:

- Introduce working with graphs in Python and modeling probability distributions over graphs.
- Demonstrate how to fit a statistical model using Maximum Likelihood Estimation (MLE) and gradient descent, implemented from scratch in Python.
- Introduce Markov Chain Monte Carlo (MCMC) simulations as a practical tool for approximating intractable expectations.

We will use a publicly available *C. elegans* connectome as input data and construct a probability distribution over graphs to study the structural properties of the original network. The tutorial is intended for graduate students with basic programming experience and familiarity with the notion of probability distributions. The current version of the tutorial is designed to take approximately one day to complete, while an extended version involving additional experimentation and hyperparameter tuning could take up to a week.

Additional links

- Tutorial notebook (with solutions) - `https://github.com/tomtalp/AI_hub_tutorial`
- Production-level ERGM implementation developed in our lab -`https://github.com/tomtalp/pyERGM`

[1] A. Haber, A. A. Wanner, R. W. Friedrich, and E. Schneidman, "The structure and function of neural connectomes are shaped by a small number of design principles," bioRxiv, Mar. 2023, doi:10.1101/2023.03.15.532611