Random Networks via Monte Carlo Simulations on the Raspberry Pi Zero W

A Computational Physics II project by Jeremy Kazimer (jdkazime@buffalo.edu)

Introduction

Whether it's obvious or not, networks are everywhere. Well, to be more precise, the information that they encode is everywhere. What I mean by this is that networks on their highest level are composed of two properties: nodes, N, and edges, M. More will be elaborated upon in the theory section, but essentially each node represents some object in a pre-defined space. This could be a person, a neuron, or... well, anything, really (Bianconi 2015). Anything that takes on an identity and can be classified by said identity. As for the other property, the edges, these encode the interactions between any set of nodes. If, for example, two people follow each other on Twitter, then they have an edge between them. These are the surface level properties of a network. There are of course more, but these are the essentials.

Thus, taking a step back, it is more obvious that networks are everywhere. To elaborate, with the right data as defined by information theory (Anand and Bianconi 2009), a network can be constructed such that the nodes encode any set of objects and the edges the subsequent interactions. To continue the Twitter analogy, take any N twitter users. These are the nodes, the objects, of our system. Each of theses users, these nodes, have unique properties, such as their username, their profile picture, etc. This is what it means for a node to encode information. They don't have to encode information, but they inherently have the structure to do so. Regardless, we can then define the existence of an edge on whether a user follows another user. If they do and follow each other, then they share an edge. If one person follows another, but not the other way around, then there exists only one edge from that user to the followed user. These are directional edges. There are also weighted edges such that perhaps an interaction between two nodes is more important than the other nodes. Either way, if they do not follow each other, then there is no edge between that node.

Really, what an edge represents here is the ability for information to spread; in a closed system, that is to say that there is no possibility of two completely separated nodes of communicating, information travels from node to node by their edges. As a physical analogy, think of this like a set of interacting particles (Barré et al. 2018); if there is no collision, that is to say particles are not interacting with each other, then energy is not spread around the system, assuming the lack of physical entropy. However, if they are interacting through collision, then this energy, the information, is spread via this transfer.

Ultimately, with the above assessment, the goal of this project is to characterize these networks on a purely random level. That is to say, for a network of size N it is generated randomly with edge probability p, the probability of an edge between two nodes, so that both the configuration and the number of edges M is random each time, within the probability constraints (Li, Mucha, and Taylor 2017). With that, we can create an analogy between this and that of actual particle interaction via information-theoretic entropic analysis, among other things. In order to make definitive statements about these random networks, the goal of this assignment, I will employ the use of Monte Carlo methods so that we can validate known theory and also make observations of our own. As a further constraint, this will also be created such that it can be reasonably ran on a Pi Zero W, which introduces its own margin of error.

Network Theory

Graph Structure

In order to generate networks, we must first define all of its properties such that the properties are understood. First, speaking of nodes we define

$$\mathcal{V} = \{1, 2, \ldots, n\} \tag{1}$$

for $n \in \mathbb{N}$ such that $\mathcal{V}$ is the set of all nodes (Li, Mucha, and Taylor 2017) and 1 and 2 represent the first and second nodes, for example. The number of nodes N is then just the cardinality of $\mathcal{V}$ such that $N = |\mathcal{V}|$. Then,

for a single-layer graph Erdős–Rényi G_{NP} , (Li, Mucha, and Taylor 2017) that is to say a two-dimensional graph with no coupling constant D_x (Kobayashi and Onaga 2021), this is defined as the adjacency matrix A such that

$$G_{NP}(\mathcal{V}, E) = A = \left[A_{ij}\right] \tag{2}$$

for $0 \leq i, j < N$. Then, the number of elements in the adjacency would then be

$$|G_{NP}| = |A| = |N| \times |N| \tag{3}$$

Note that this is not the number of edges, but rather the number of all entries for this adjacency matrix or, in computer science terms, the size of the array. From this point onward, we'll only refer to G_{NP} as A. We can then define the values of A_{ij} such that

$$A_{ij} = \begin{cases} 1 : (i,j) \in E \\ 0 : (i,j) \notin E \end{cases} \tag{4}$$

where E is the set of all edges (Li, Mucha, and Taylor 2017) such that

$$E = \{(i,j) : p_{ij} \leq p\} \tag{5}$$

for some particular edge probability p_{ij} and a total edge probability p. Basically, each entry of A, A_{ij}, has an associated probability p_{ij} such that if it's less than the fixed parameter probability p then there exists an edge. We can then define the total number of edges as

$$M = |E| \tag{6}$$

For an unweighted and undirected matrix, this is the same as saying $2M = \sum_{ij} A_{ij}$, due to the symmetry of the adjacency matrix (Li, Mucha, and Taylor 2017). The expected number of edges then, for an A graph, would be

$$\bar{M} = \frac{N(N-1)}{2} \cdot p \tag{7}$$

because, without self edges, that is we disallow a node from being connected to itself, there are N nodes. However, we remove the locations where a node can connect itself such that there remains $N-1$ spots remaining. Then, we divide by 2, due once again to the symmetry of an undirected and unweighted adjacency matrix (n.d.). This leaves us with $\frac{N(N-1)}{2}$ edges. Since this is a purely probabilistic process, we multiply this term by p since generally an expectation function takes on the form

$$\bar{x} = x \cdot p \tag{8}$$

We can then relate this to the number of potential edges P, that is to say areas where edges don't exist such that they could exist if under some stochastic process, can be represented as

$$P = \frac{N(N-1)}{2} - M \tag{9}$$

since $|A|$ counts edges twice. We can then define the degree matrix, that is the number of edges connected to each node, as

$$D = \text{diag}[d_1, d_2, \ldots d_N] \tag{10}$$

where $d_i = \sum_i A_{ij}$ for the i-th node (Li, Mucha, and Taylor 2017). The unnormalized Laplacian matrix L is then given by the equation

$$L = D - A \tag{11}$$

This is particularly useful to us in that this is the foundation of spectral analysis. That is to say, we use the Laplacian in order to extract its eigenvalues λ and eigenvectors $\vec{v}$ for a variety of uses. This also appears in fields such as machine learning and computer vision, but for our uses this is simply the basis of spectral analysis.

Stochastic Block Model

Rewiring

Von Neumann Entropy

Spectral Analysis

VNE

Namely, von Neumann Entropy (VNE) ... *THIS IS FOR THE NEXT PART! Manlio...* (Domenico and Biamonte 2016)

Edge Ranking

Using Monte Carlo as an Explorative Tool

...

Potential Applications to Real Physics

Results

Comparison to other Methods

Although there aren't many other analytical methods that are similar to network theory, spectral theory (Domenico and Biamonte 2016) in particular, the one that stands out the most is principal component analysis (PCA) (Jolliffe and Cadima 2016). This family of algorithms in particular examines the eigenvalues and eigenvectors so that a fuller picture of the data is formed. Typically, this is used to reduce down data to a scalable form such that loss of information is minimized.

Network theory and, by extension, spectral theory is a bit better than PCA in some cases because it allows for a more complex relationship to be developed by the data. That is to say, complex networks are able to encode very specific interactions between nodes, especially when multiple layers, a multilayer or multiplex network, are introduced. Since PCA reduces these features down to only the eigenvectors and eigenvalues typically, many of these rich relationships are lost in translation.

Spectral theory is particularly important, especially with the introduction of Shannon entropy via von Neumann Entropy (VNE) (Li, Mucha, and Taylor 2017). This could be perhaps be treated as an extension of both PCA and network theory, since entropy takes on the form of its eigenvalues. Regardless, VNE is far more common a tool of analysis in network theory than it is in PCA. Really, it in itself is reducing data down to a single value, akin to PCA, but this is used rarely to more so describe the entire system, as opposed to its

individual components. Perturbation theory (Li, Mucha, and Taylor 2017) is also useful in studying how the network changes under slight modifications to the network, something that PCA doesn't exactly cover.

Nevertheless, it is the amount of information that can be gleaned from network theory that sets it above PCA. The complex systems can also be captured, but at the cost of runtime complexity. It is true that in general PCA is faster than network theory, since its main bound is calculating eigenvalues. Whereas, with network theory, there are many methods from modularity, to rewiring, to edge ranking, which all have significant runtime complexities. However, their availability as an analysis tool is more important than runtime, in a larger scale. Of course, this may not be the case for the Raspberry Pi Zero W, but initial conditions can be set such that runtime complexity isn't a major factor.

Conclusion

....

References

Anand, Kartik, and Ginestra Bianconi. 2009. "Entropy Measures for Networks: Toward an Information Theory of Complex Topologies." *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics* 80 (October): 045102. https://doi.org/10.1103/PhysRevE.80.045102.

Barré, Julien, J. Carrillo, Pierre Degond, Diane Peurichard, and Ewelina Zatorska. 2018. "Particle Interactions Mediated by Dynamical Networks: Assessment of Macroscopic Descriptions." *Journal of Nonlinear Science* 28 (February). https://doi.org/10.1007/s00332-017-9408-z.

Bianconi, Ginestra. 2015. "Interdisciplinary and Physics Challenges of Network Theory." *EPL (Europhysics Letters)* 111 (September). https://doi.org/10.1209/0295-5075/111/56001.

Domenico, Manlio, and Jacob Biamonte. 2016. "Spectral Entropies as Information-Theoretic Tools for Complex Network Comparison." *Physical Review X* 6 (September). https://doi.org/10.1103/PhysRevX.6.041062.

Jolliffe, Ian, and Jorge Cadima. 2016. "Principal Component Analysis: A Review and Recent Developments." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374 (April): 20150202. https://doi.org/10.1098/rsta.2015.0202.

Kobayashi, Teruyoshi, and Tomokatsu Onaga. 2021. "Dynamics of Diffusion on Monoplex and Multiplex Networks: A Message-Passing Approach," March.

Li, Zichao, Peter Mucha, and Dane Taylor. 2017. "Network-Ensemble Comparisons with Stochastic Rewiring and von Neumann Entropy." *SIAM Journal on Applied Mathematics* 78 (April). https://doi.org/10.1137/17M1124218.

n.d. *Avrim Blum: CS 598 (CRN 62819) Topics in Algorithms (2015)*. https://www.cs.cmu.edu/~avrim/598/.