Watts–Strogatz Networks with Node Degrees Drawn from a Normal Distribution

Oscar Wohlfahrt (ojw2015@nyu.edu)

Abstract

Many real-world networks have small-world properties, with high clustering combined with short path length, but do not have uniform node degree. In this project, we modify the standard Watts–Strogatz model by assigning the number of edges per node from a normal distribution instead of giving all nodes the same degree. We then rewire edges with a probability p and measure the characteristic path length $L(p)$ and clustering coefficient $C(p)$. Our results show that this approach still produces networks with small-world characteristics for a range of p-values. We also apply our method to the neural network of *C. Elegans*, and find a rewiring probability that minimizes the difference between the actual network and the model.

1 Introduction

Graphs can be used to model everything from social networks, the spread of a disease, power grids , neural networks of species or mapping out paths in cities. Ordinarily, networks have been assumed to be either completely regular, where each node in a graph is connected to a certain amount of its closest neighbors, or completely random, where connections between nodes are decided at random. Many real life networks lie somewhere in-between these extremes. It is common to have both a short characteristic path length, which completely random graphs will have, but also to be highly clustered, which regular graphs will be. This type of network is commonly called a "Small World Network". and is commonly known through the "six degrees of separation" phenomenon.

In the paper *Collective dynamics of 'small-World' networks* Watts and Strogatz (1998) D. Watts and S. Strogatz consider a procedure to rewire regular networks. Starting from a ring lattice with n vertices and k edges per vertex, each edge is rewired at random with a probability p. In this way a graph can be tuned between being completely regular, when $p = 0$, and completely random, when $p = 1$. The graphs structure is quantified by their characteristic path length $L(p)$ and clustering coefficient $C(p)$, which will be defined in the next section.

In real networks it is often the case that each vertex has the different amounts of edges. For instance, in a social network, there are typically some people who are connected to a large amount of their close neighbors, whilst others have few connections. To mimic this behavior we propose to draw the amount of edges each vertex has from a normal distribution, centered at k with standard deviation σ. After creating the graph it is rewired through the same procedure as described above.

In this report we aim to investigate wether these graphs will have the same structural properties as is typical for small-world networks, with a low $L(p)$ ans high $C(p)$, as well as model some real world networks and find suiting values of p for these.

2 Problem setup

To investigate this problem we use a Python script which simulates the creation and rewiring of the graphs. We then used the same measures, namely $L(p)$ and $C(p)$, as Watts and Strogatz (1998) to investigate the graphs properties.

The information about which edges are connected is stored in an adjacency matrix. If an edge exists between vertex i and vertex j, a 1 is added to the graph both at position (i, j) and (j, i), since the graph should be undirected. When rewiring, duplicate edges is not allowed, and an edge is not allowed to connect the vertex with itself. The amount of edges each vertex has is drawn from a normal distribution centered at k. We have chosen to use a standard deviation of 1 for all experiments, but this is chosen quite arbitrarily. To get the random number the Pythons module *random* is used.

$L(p)$ is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices. If there doesn't exist a path between a pair of vertices, they are not considered in this average. To find the shortest path between two vertices we use Dijkstra's algorithm.

$C(p)$ is defined as the fraction of the edges between a vertices k nearest neighbors that exists. If a vertex v has k neighbors, then at most $\frac{k(k-1)}{2}$ edges can exist between them. This happens when every neighbor of v is connected to every other neighbor of v. $C(p)$ is the fraction of these edges that actually exist, averaged over all vertices. To find $C(p)$ we create a Python function which iterates through the graph and check wether a edge between neighbors exists or not.

To investigate wether the graphs with a normally distributed number of edges have the typical small-world properties, we calculate $L(p)$ and $C(p)$ for values of p from 10^{-4} to 1. $L(p)$ and $C(p)$ is calculated 10 times for each value of p and averaged. These averaged values of L and C are then normalized with $L(0)$ and $C(0)$.

In addition, we modeled the neural network of the nematode *C. Elegans*, which at the time when the paper Watts and Strogatz (1998) was written, was the only completely mapped neural network. This neural network has 282 nodes and an average of 14 edges per vertex. The graph of the neural network of *C. Elegans*, has a characteristic path length of 2.65, and a clustering coefficient of 0.28. We define the difference d between two graphs as the sum of the difference between the actual values of L and C and the the graphs L and C squared. As in the equation below,

$$d = (L_{actual} - L_{graph})^2 + (C_{actual} - C_{graph})^2 \tag{1}$$

our goal is to approximate a value of p that minimizes this difference, which we do through plotting p against d.

3 Results

As said before, a small-world network typically has a low characteristic path length, but is highly clustered. For our graph to be small-world behavior we therefore want a relatively low value of $L(p)$, but a high value of $C(p)$. In Figure 1 the normalized values of L and C have been plotted against different values of p.

We can see that we get small-world traits for quite a large range of p-values. It should be noted that the x-axis is logarithmic, as the characteristic path length drops very quickly as p increases. When comparing this plot to the one presented in Watts and Strogatz (1998), we can see that the characteristic path length seems to drop more quickly. One reason for this could be that some nodes in this graph will have a very large amount of neighbors, and thus already be connected to other nodes that are relatively far away.

When modeling the neural network of *C. Elegans* we found through some initial trials that the optimal p-value would be somewhere between 0.1 and 0.2. Figure 2 shows the difference between the actual graph and the modeled graph for p-values between 0.1 and 0.2. From this plot we can see that the optimal p-value will be a little below 0.14.

4 Summary

To summarize, we investigated a variation of Watts-Strogatz networks, where the amount of edges per vertex was drawn from a normal distribution rather than being fixed. We found that these graphs still behave like small world networks for a large range of p. We also used these graphs to model the neural network of *C. Elegans*, and found that the value of p that creates the best fit, is slightly below 0.14. Overall, the results show that these graphs will still have small-world behavior when we let p vary, and that they could work well for modeling real world phenomenon. Moving forward, I believe it would be interesting to use these networks to model how diseases spread through a population, and on networks with larger values of n and k.

References

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393.

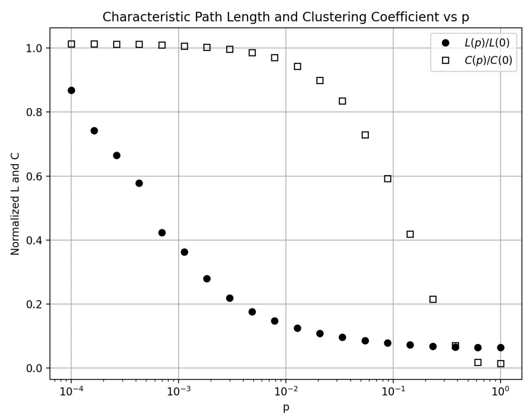

Figure 1: Characteristic path length and clustering coefficient plotted against p. Using n = 1000 and k = 10. L(p) and C(p) are normalized with L(0) and C(0)

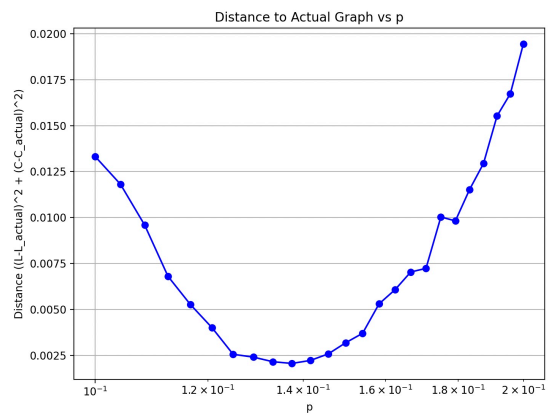

Figure 2: Difference between L and C of *C. Elegans* and modeled graphs for different values of p. Using n = 282 and k = 14