

THE ERDŐS-RÉNYI MODEL, RANDOM GRAPHS, AND THRESHOLDS

MADDY RITTER & WILLIAM J. MCMILLIAN

ABSTRACT. This paper explores random graphs and the probabilities of fully connected components using the Erdős-Rényi model.

1. INTRODUCTION

We will begin by explaining some basic graph theory. A graph G is an ordered pair of two sets, V and E , where V is the set of vertices (also referred to herein as nodes), and E is a set of edges or lines that connect any two $v_i, v_j \in V$. We denote a graph as $G = (V, E)$. For the purposes of this paper, the graphs we are referring to will always be **simple** graphs. Every edge in a simple graph connects only two vertices v_i and v_j , and no two vertices are connected by more than one edge, and so an edge is defined as the set $\{v_i, v_j\}$. A **path** is a sequence of distinct edges and vertices that connect two vertices. We can walk along this path to get from one vertex to another without walking on the same edge or vertex twice.

Connected graphs are graphs such that there is a path between any two vertices in V . Figure 2 shows three connected graphs. In Figure 3, three graphs are shown. These three graphs have multiple disjoint parts, which we call **components**. There are edges that connect some nodes of the graphs, but there are not paths from all nodes to all others.

Complete graphs have every edge between any two vertices in V . An example of this is shown above on 5 vertices. A connected graph of size n , denoted K_n has $\frac{n(n-1)}{2}$ edges. The **degree** of a node is the number of edges that connect to it. In a complete graph, the degree of every node is $n - 1$. For n vertices this sums to $n(n - 1)$ edges. However, this counts all edges twice and therefore the proper number of edges is $\frac{n(n-1)}{2}$.

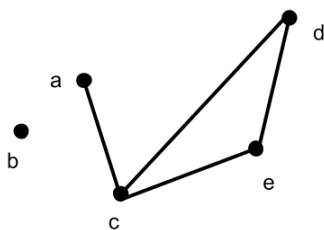


FIGURE 1. Above, we see a graph, $G = (V, E)$, where $V = \{a, b, c, d, e\}$ and $E = \{\{a, c\}, \{c, d\}, \{c, e\}, \{d, e\}\}$. There is an isolated vertex b and one cycle, $\{c, d, e, c\}$.

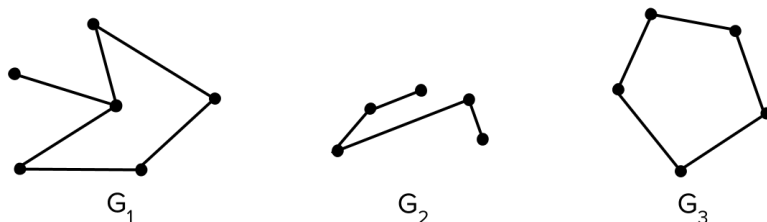


FIGURE 2. Some examples of simple connected graphs.

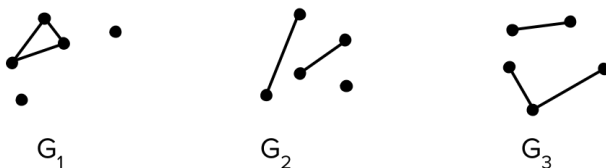


FIGURE 3. Some examples of disconnected graphs. Clearly, there are either isolated vertices or disjoint components, and thus are by definition disconnected.

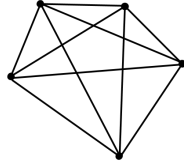


FIGURE 4. The complete, simple graph on 5 vertices. Every possible edge exists.

2. ERDŐS-RÉNYI

The Erdős-Rényi models refer to a pair of models for generating random graphs, $G_{n,m}$ and $G_{n,p}$, where edges are produced independently. The $G_{n,m}$ model produces a graph with n nodes and m distinct randomly selected edges. The $G_{n,p}$ model produces a graph with n nodes and gives all of the possible edges between the nodes a p chance of existing.

These models allow us to create graphs procedurally, edge by edge. For the $G_{n,m}$ model, we have m randomly selected edges. We can think of taking the ordered set of all possible edges, perfectly shuffling their order, and then picking the first m in the shuffled set to exist, leaving the rest. For the $G_{n,p}$ model, we can form a list of all possible edges, and with probability p , one at a time determine whether or not it will exist.

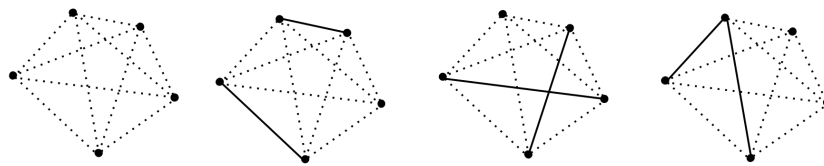


FIGURE 5. On the left, we see a graph on 5 vertices. Every possible edge is formed with a dotted line. The three right-hand images are various expected graphs produced by $G_{5,2}$, each with 5 vertices and 2 edges.

Theorem 2.1. *A random graph $G_{n,p}$ on average will have $\binom{n}{2}p$ edges.*

Proof. Suppose we have a graph on n vertices. If we observe the complete graph on n vertices, we can find the total number of edges possible by taking $\binom{n}{2}$. This is because $\binom{n}{2}$ counts every subset of 2 elements. Since each edge can be defined as a subset of 2 elements, $\binom{n}{2}$ is clearly equivalent to the number of possible edges. When we multiply this by probability p , we get the number of edges that are probable in $G_{n,p}$. \square

3. THRESHOLDS

To introduce the concept of thresholds, we will take a sampling of 10000 random graphs, shown below, and look at the number of vertices in the largest component of every graph. The p value for the graphs is slightly different in the two Figures, and we can see how the p value affects the graphs.

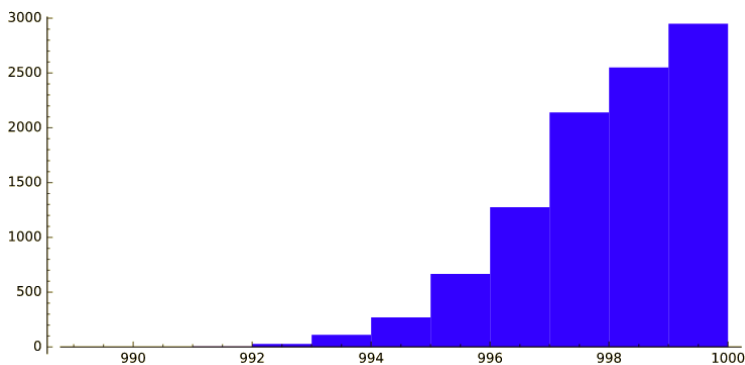


FIGURE 6. The x -axis is the number of vertices in the largest component for $G_{1000,0.006}$. The y -axis is the number of trials, out of a total of 10000.

We can observe in Figures 6 and 7 that the results for the largest connected component on $G_{1000,p}$. p in Figure 6 is .006, or .6 percent, and p in Figure 7 is .0069, or .69 percent. Though these values of p are very close, it is evidently much more likely that the graph will be connected, creating a component on all 1000 vertices, for $p = 0.0069$. This suggests that if there is a particular threshold for which it is much more likely that the graph will be connected than for just below this threshold.

We want to answer two questions:

Question 3.1. *Given n vertices, what is the smallest p such that we expect our random graph $G_{n,p}$ to be connected?*

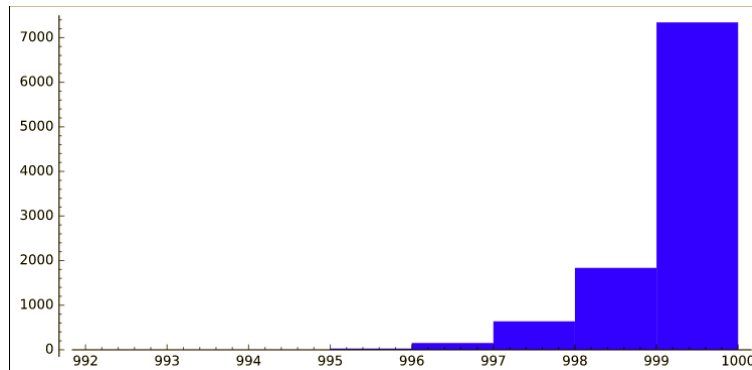


FIGURE 7. The x -axis is the number of vertices in the largest component for $G_{1000, .0069}$. The y -axis is the number of trials, out of a total of 10000.

Question 3.2. *Given n vertices, what is the smallest value of m such that we expect our random graph $G_{n,m}$ to be connected?*

The proof for Question 3.1 is fairly easy to find, but complex to understand, and so we can investigate it with computation. As randomness by definition ensures no absolute threshold, we can set conditions that makes us almost certain that the graph will be connected. The procedure of the algorithm used to compute the thresholds goes as follows:

Create variable \mathbf{n} for n to represent number of edges, and set it to 100.

$\$ \mathbf{n} = 100$

Begin a loop to test all n from 100 through 10,000.

$\$ \text{while } \mathbf{n} \leq 10000:$

Create a boolean variable called **trip**, to function like a *tripwire* when we have reached the threshold for n and need to move on the the next one.

$\$ \text{trip} = \text{False}$

For this particular n , we create a variable \mathbf{p} for p and set it to .001.

$\$ \mathbf{p} = .001$

Set a condition for while *tripwire* is false, we continue to test \mathbf{n} .

$\$ \text{while } \text{trip} == \text{False}:$

Make a loop to perform up to 500 trials.

$\$ \text{for } \mathbf{a} \text{ in range}(500):$

Make a graph \mathbf{G} using \mathbf{n} and \mathbf{p} .

```
$ G = graphs.RandomGNP(n, p)
```

If \mathbf{G} is not connected, exit the loop, restart the 500 trials, and increment \mathbf{p} by .0001.

```
$ if G.is_connected() == False:
$ break
```

If G is connected, check to see if we have made 500 graphs (we check the counter at 499 because computers begin at 0 instead of 1), then print \mathbf{p} . As the program runs, this will compile a list of our thresholds.

```
$ if a == 499:
$ print p
```

Trip the *tripwire* by setting **trip** to true so that the program moves onto the next \mathbf{n} , and stop running trails. \$ trip = True

```
$ break
$ if trip == True:
$ break
```

When a graph is produced that is not connected increment \mathbf{p} by .0001
\$ p = p + .0001

When the threshold is found, increment \mathbf{n} by 100 and begin trials.
\$ n = n + 100

In summary, we run a loop that produces $G_{n,p}$ graphs with $n = 100$ and $p = .001$. After the creation of each graph, we check if it is connected. If we create a graph that is not connected before reaching a count of 500, p is incremented by .0001 and the count is reset to 0. If we do reach 500 connected graphs, we consider that p to be the threshold for the given n . The algorithm then moves onto the next sized graph by incrementing n by 100. The results of the experiment are displayed in the figure below.

We decided that 500 was a reasonable choice for a trial stop limit given compiling constraints. The nature of randomness does not allow for any absolute threshold, so we will call our calculated thresholds, the *experimental threshold*. We can think of the threshold in giving us at least 99.8 certainty that a graph of size n with our experimental threshold will be connected.

It has been proven that connectivity has a sharp threshold, called the Friedgut threshold, with a critical probability of $\frac{\log(n)}{n}$ [3]. At this threshold we would more likely expect the graph to be connected than not. As n approaches infinity, values of p above the Friedgut threshold increase in their certainty of producing a connected graph, while the

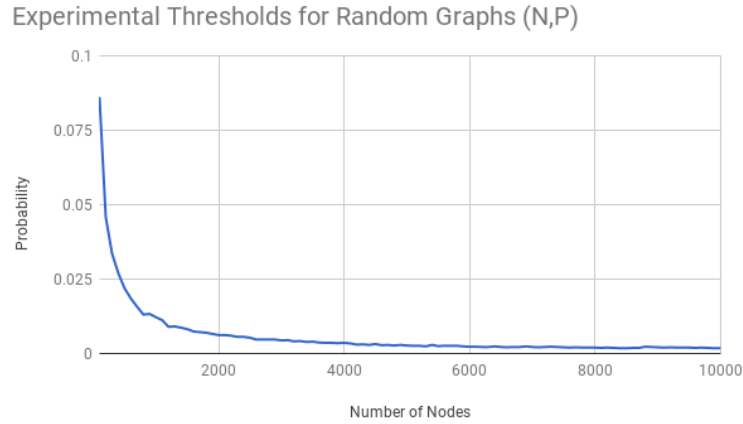


FIGURE 8. A graph of our calculated experimental thresholds for the Erdős-Rényi $G_{n,p}$ model.

certainty of p 's below it continually decrease. When we compare the experimental and Friedgut thresholds, it stands out that our experimental threshold resembles the function $\frac{\log(n)}{n}$ at scale.

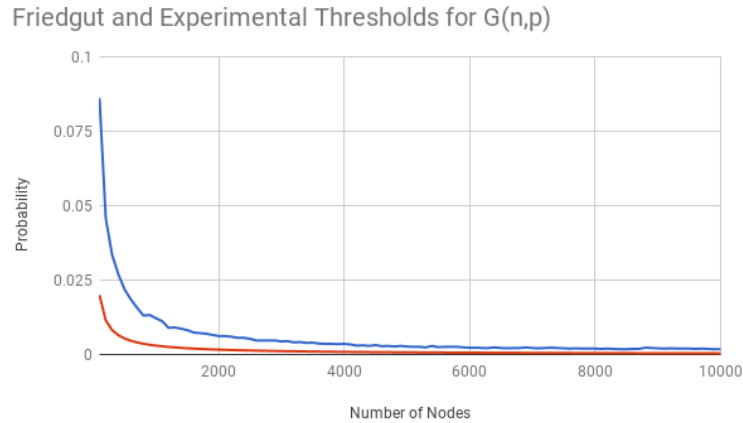


FIGURE 9. Above are the overlaid charts of the experimental and effective thresholds. The Friedgut threshold appears in red, and the experimental threshold in blue.

Our experimental threshold is on average 3.987 times larger than the effective threshold. This allows us to approximate our experimental threshold with the equation:

For $G_{n,p}$, the experimental threshold is approximately $\frac{3.987 \cdot \log(n)}{n}$

Unfortunately due to computing constraints, we were unable to do calculate a similar experimental threshold for the $G_{n;m}$ model.

4. APPLICATIONS OF THRESHOLDS TO REAL NETWORKS

The Erdős-Rényi model is a poor representation of real-world networks, as networks are generally constructed to model an environment impacted by resource constraints. Random graphs can be applied comparatively to models of real networks to highlight certain constraints or patterns that would otherwise not stand out. Erdős-Rényi graphs often highlight clustering or transitivity (when there is a high frequency of two neighbor vertices of a vertex to also be neighbors of each other). Thresholds for connectedness can be used in this same way to compare actual results to projections.

Take, for example, a social network mapping of a university where we represent individuals as nodes and connections or friendships as edges. If the university's administration had the goal of half of all students being directly connected, they could compare their actual results to a $G_{n,p}$ random graph with n equal to the number of students and p set to .5, so that all every friendship has a 50% chance of existing. If their goal was to have all students are at least indirectly connected, they could make a similar $G_{n,p}$ but make p our experimental threshold.

Knowing that people do not pick their friends at random, we would expect the administration's results to vary from the Erdős-Rényi model. Analysis of the differences in clustering could highlight relationship influences like affinity groups, athletic teams, course schedule, places of origin, or residential hall proximity. If certain goals were not met, the administration could utilize the analysis to better inform their following decisions about things like events on campus and student activities.

A more intricate, larger scale mapping can be done with larger networks like cities. Dave Troy is a mathematician who involved in this type of work. Troy retrieves data from social networks and analyzes activity on things like hashtags and commonly shared links. He then applies a forced-direct graph layout algorithm to the simple graph. This causes people with many relationships between them to be positioned into tight clusters in the graph, people with few relationships positioned on opposite sides of the graph, and people with many relationships on both sides positioned in the middle of the graph. With this, he can map out a social schematic of geographic regions. This can give us an idea of social dynamics among different groups and how they change based on current events.

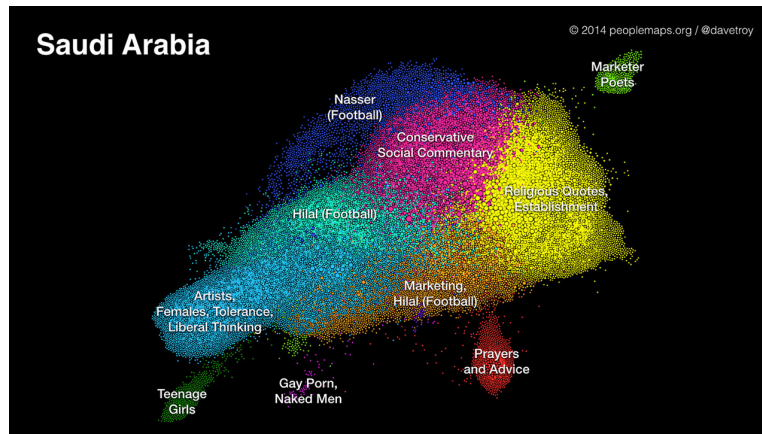


FIGURE 10. A map of people living in Saudi Arabia where each dot represents a node and each line represents a relationship. The graph is made without respect to geography, but nodes are organized by clusters. [2]

We can look at **scale-free** models to give a more accurate representation of real networks. Instead of having a static set of n nodes, we begin with a smaller number of nodes which will grow as the network continues to expand. Additionally, we will have **power-law degree distribution** where the probability of the degree of each node is dependent on the growth of the number of nodes.

5. CONCLUSION

In this paper we have learned about the Erdős-Rényi of random graphs and their thresholds for connectedness. The findings of our experimentation, though limited by computational constraints, have given us an idea of what a realistic threshold for connectedness would look like to ensure a high level of certainty. For further exploration, we might want to delve deeper into the implementation of other random graph models for the modeling of real-world networks. It would be of particular interest to us to understand how to simulate social networks, such as the ones mentioned in §4.

REFERENCES

- [1] Statistical Mechanics of Complex Networks, <https://barabasi.com/f/103.pdf>
- [2] People Maps by Dave Tory, <http://www.peplemaps.org>

- [3] Sharp Thresholds of Graphs Properties, and the k-SAT Problem,
https://www.math.ucsd.edu/~sbuss/CourseWeb/Math268.007WS/Friedgut_Treshold.pdf
E-mail address: ritte23m@mtholyoke.edu, wcmillian20@amherst.edu

DEPARTMENT OF MATHEMATICS AND STATISTICS, MOUNT HOLYOKE COLLEGE, SOUTH HADLEY, MA 01075