Using Spanning Trees as a Metric of Graph Connectivity

Victor Alvarez

12 August 2021

Abstract

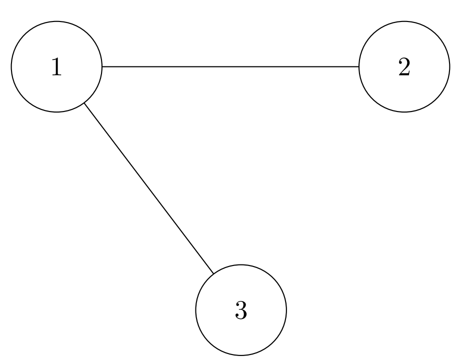
There exists many algorithms for finding minimum weight spanning trees of an undirected graph with weighted edges. There are also path finding algorithms, such as Dijkstra's algorithm, which will create spanning trees in the process (not necessarily minimum weight). This leads to the question: What can the number of spanning trees tell us about a graph? In this paper we explore how the number of spanning trees of a graph can be used as a measure of connectivity.

1 Definitions

Firstly, we will define some concepts that are important to the paper.

Definition 1.1 (Graph)**.** *A **graph** is a set of vertices, V, (also called nodes) and a set of edges E defining connections between the elements of V.*

For example, consider the following graph:



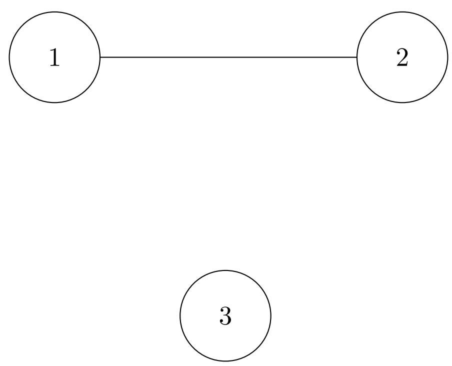
This graph is defined as $G = (V, E)$ where $V = \{1, 2, 3\}$ and $E = \{(1, 2), (1, 3)\}$. This paper will only deal with *undirected* graphs, meaning that edges do not have a direction. In other words the edges defined by $(1, 2)$ and $(2, 1)$ would refer to the same edge. In *directed* graphs this would not be the case, and $(1, 2) \neq (2, 1)$.

Definition 1.2 (Connected Graph)**.** *A graph G is said to be connected if there exists a path between any two vertices of G.*

The graph G defined above is not connected, however it would be connected if we simply added the edge $(2, 3)$.

Definition 1.3 (Subgraph)**.** *Given a graph $G = (V, E)$, a **subgraph** $H = (V', E')$ is a graph where $V' \subset V$ and $E' \subset E$, such that all edges in E' have endpoints that are in V'.*

For the earlier defined graph G, we could create a subgraph $H = (V', E')$ where $V = \{1, 2, 3\}$ and $E' = \{(1, 2)\}$



Definition 1.4 (Tree)**.** *A **tree** is a graph $T = (V, E)$ such that any two vertices are connected by exactly 1 path.*

Definition 1.5 (Spanning Tree)**.** *A subgraph T of a graph G that is a tree containing all of the vertices of G.*

Now, we will define a few different matrices that can be defined on graphs.

Definition 1.6 (Degree Matrix)**.** *For a graph $G = (V, E)$, where $|V| = n$, the degree matrix is a $n \times n$ diagonal matrix D defined as:*

$$D_{ij} = \begin{cases} deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Definition 1.7 (Adjacency Matrix)**.** *For a graph $G = (V, E)$ where $|V| = n$, the adjacency matrix is an $n \times n$ matrix defined as:*

$$D_{ij} = \begin{cases} 1 & \text{if an edge exists between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases}$$

So, for the graph G defined earlier we have the degree matrix

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Definition 1.8 (Laplacian Matrix)**.** *For a graph $G = (V, E)$ where $|V| = n$, the graph Laplacian is a symmetric $n \times n$ matrix defined as*

$$L = D - A$$

where D and A are the degree matrix and adjacency matrix of G, respectively.

Therefore, the Laplacian matrix is a matrix representation of a graph. The Laplacian matrix of the previously defined graph would be:

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Now, we have the necessary tools to define the theorem that is crucial to this paper.

Theorem 1.1 (Kirchoff's Theorem)**.** *For a connected graph $G = (V, E)$ where $|V| = n$, then the number of spanning trees of G is:*

$$t(G) = \frac{1}{n}\lambda_1\lambda_2 \ldots \lambda_{n-1}$$

where $\lambda_1, \ldots, \lambda_{n-1}$ are the eigenvalues of the Laplacian matrix representing G.

This value $t(G)$ turns out to be the same as *any* cofactor of the Laplacian of G.

In addition to Kirchoff's Theorem, we will be making use of another formula regarding graphs.

Theorem 1.2 (Cayley's Formula)**.** *For any natural number n, the value n^{n-2} gives the number of trees that can be constructed with n labeled vertices.*

Equivalently, this value n^{n-2} is the number of spanning trees that can be constructed given the *complete* graph of n vertices[1].
We will be calculating this value for any graph G by simply using $n = |V|$. We will refer to this value as the *Cayley Value*.

2 Process

The goal is to construct a measure of graph connectivity using the number of spanning trees of a graph. The most obvious measurement was to compare the number of spanning trees that exist in a graph to the total number of spanning trees that could exist given the number of vertices in the graph (i.e. the Cayley Value).

The networkx python library provided several functions that faciliated this process. Firstly, we use a library function that implements the Erdos-Renyi graph generator. Using this function we construct hundreds of connected graphs.

Then we implement Kirchoff's Theorem using the numpy python library to calcualte the determinant of the laplacian matrix of a graph. Using our implemented function we calcualte the numbe of spanning trees for each of our generated graphs, as well as their Cayley Values and store this information.

However, it becomes apparent very quickly that, as the number of vertices increases, the Cayley Values grow significantly faster than the number of spanning trees of the randomly generated connected graphs. This means that the ratio

$$\frac{\text{number of spanning trees}}{\text{Cayley Value}} \tag{1}$$

goes to zero very quickly.

Thus, we take the logarithm of each of both of these values base n, where n is the number of vertices in the graph. We do this to place the resulting ratios

[1]the **complete** graph of n vertices, denoted K_n, is a graph in which there exists an edge between all pairs of vertices.

on a much more reasonable scale. Therefore, the metric we will be using is

$$\frac{\log(\text{number of spanning trees})}{\log(\text{Cayley Value})} \tag{2}$$

Now, we also needed some existing metrics of graph connectivity to compare this metric to. We will be using graph density and edge connectivity.

Definition 2.1 (Graph Density)**.** *The graph density is the ratio of edges in the graph to the number of edges that could exist given the number of vertices in the graph. Given an undirected graph G, the density is given by*

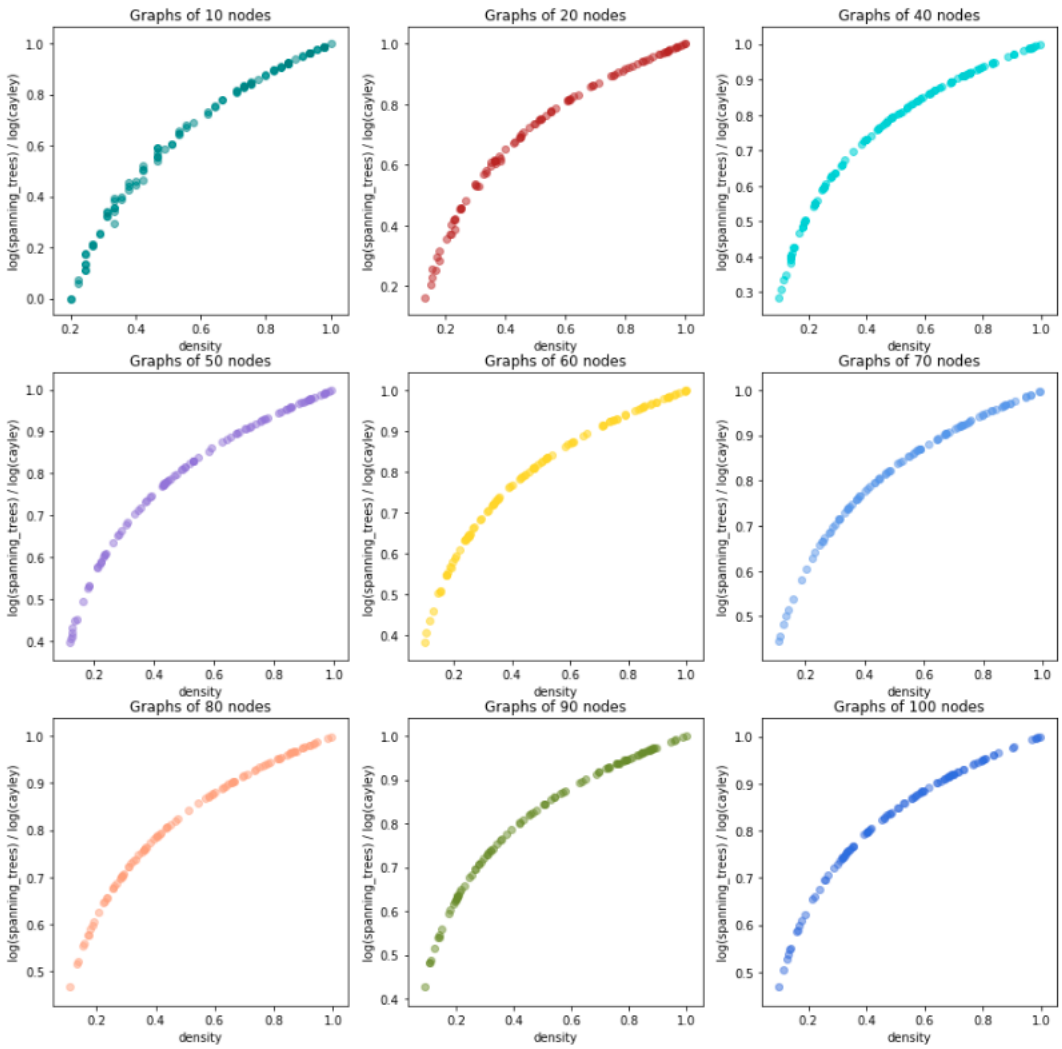
$$\frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)} \tag{3}$$

where m is the number of edges in G and n is the number of vertices in G.

Definition 2.2 (Edge Connectivity)**.** *Given a connected, undirected graph G the* ***edge connectivity*** *is the minimum number edges that must be removed for the graph to be disconnected.*

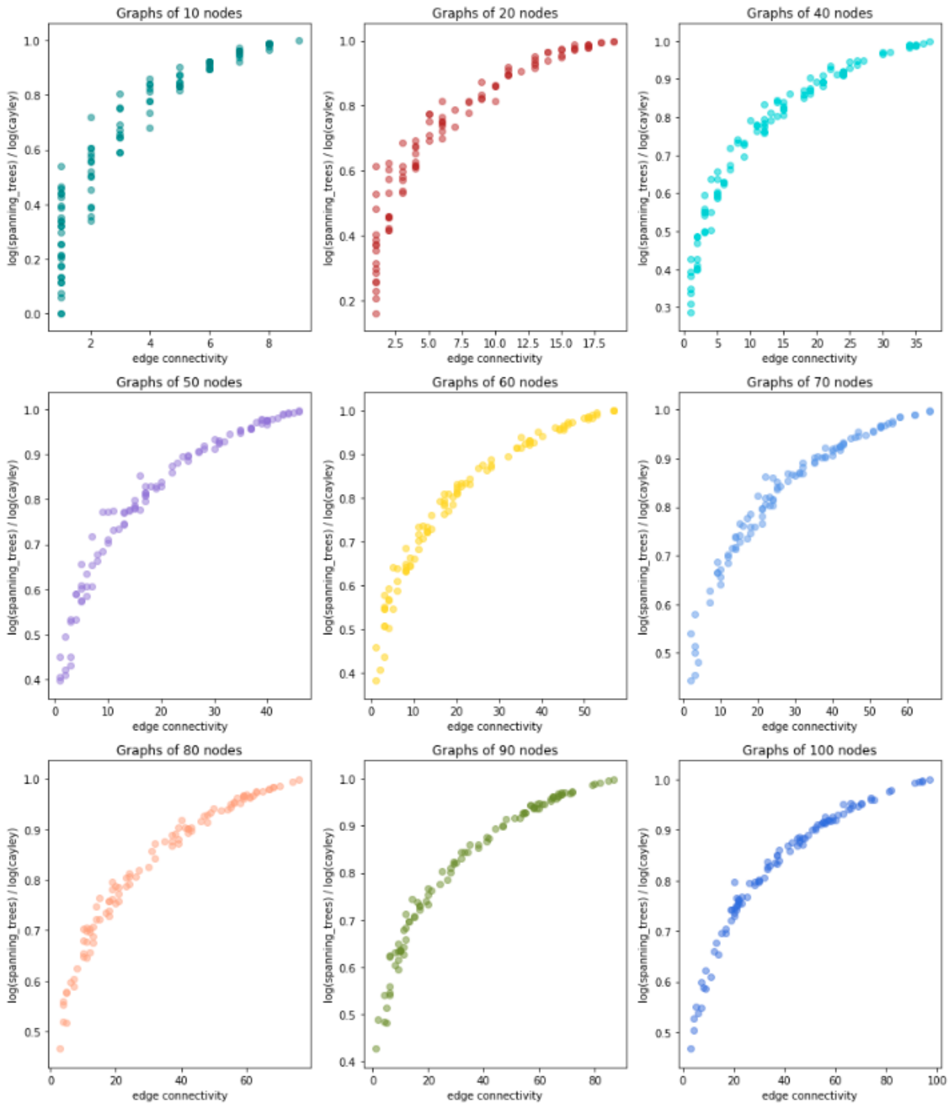
Now that we have all of the measurements of interest, we compare their values by plotting them against our new measurement.

3 Results



These are the plots of our new measurement (y-axis) against the graph densities (x-axis) separated by the number of vertices in the graphs. There are 100 of each graphs for each number of vertices. The graphs show an obvious trend, with the steepest section of the plots being when the graphs were of lower density. This would suggest that this new metric may be best used on sparse graphs, as this is where it is most sensitive.

The following graphs compare our new metric to the edge connectivity of graphs, again separated by the number of vertices in the graphs.



These plots show that there is significantly less variation, and a more obvious trend, as the number of vertices increases. This makes sense since the lower the number of vertices, the lower the probability of having strongly connected clusters within the graph.

Once again, for graphs with a higher number of vertices, where the trend if clearer, the plot is steepest when the edge connectivity is smaller. Using the Erdos-Renyi graph generator this relationship would make sense. Larger graphs with a low edge connectivity are likely to be very sparse graphs when generated using Erod-Renyi.

4 Conclusion

These plots demonstrate interesting relationships between this new measurement these two existing measurements of graph connectivity, for graphs generated from Erdos-Renyi. However, these graphs are not likely to have bridges or smaller clusters with greatly varied connectivity. With more computing power and perhaps a different graph generator, it would be interesting to consider graphs that are even more random in their structure than those used in this experiment.

All of the code for this project can be found here.

References

[1] Wikipedia contributors. Cayley's formula — Wikipedia, the free encyclopedia, 2021. [Online; accessed 12-August-2021].

[2] Wikipedia contributors. Connectivity (graph theory) — Wikipedia, the free encyclopedia, 2021. [Online; accessed 12-August-2021].

[3] Wikipedia contributors. Kirchhoff's theorem — Wikipedia, the free encyclopedia, 2021. [Online; accessed 12-August-2021].

[1] [2] [3]