Studying the self-similarity of complex networks

Julia Wei

December 18, 2018

Introduction

Motivation

How might researchers predict the severity of the annual flu season and the speed at which viral vectors infect a population? What details about human relationships might Facebook glean from an individual's "friends," groups, pages, and browsing habits in order to achieve a more invasive, personalized user experience? What insights into the complicated interactions among different regions of the human brain might neuroscientists gain by performing non-invasive functional imaging, and how might they control and shape those interactions?

Each of these questions belongs to a distinct field of study — epidemiology, sociology, and neuroscience, respectively. There is something appealing about identifying common methods and approaches to model seemingly disparate processes, and one such approach arises from the study of complex networks. Whether it is used to analyze viral, social, or neural networks, a theory of complex networks may be able to uncover similarities between diversely sourced networks.

Graph theorists have studied large complex networks since the 1950s, a textbook example being the Erdös-Rényi model for random graphs [1]. On the other hand, applications and experimental studies of complex networks gained popularity in the 1990s due to the confluence of large, available datasets and a sufficient increase in computational power and resources [2]. Then the field of complex networks exhibits at once elegant mathematical theories [3] and the crucial need to develop rigorous statistical methods that detect biases and uncertainties in models of noise-ridden data [4].

Broadly, the motivation for complex networks-based analyses is two-fold. First, how well do complex networks models represent structured or relational data, and how well do they capture useful statistical properties and characteristics of this data? Second, can complex networks models constitute a theory, demonstrating general network properties or mechanisms in datasets of diverse origin?

Keeping these two motivations in mind, we will study one property of complex networks — their self-similarity or lack thereof, given by the behavior of the statistical properties of a network under some coarse-graining transformation — from both theoretical and applied perspectives. Studying whether networks are self-similar can reveal insights about their structure that previous metrics and analyses, such as the clustering coefficient and degree distribution, do not uncover. In particular, self-similarity can be a tool to understand the

shared attributes of real-world networks. First, we will try to reproduce the main results of Radicchi et al., who studied the renormalization flows of a coarse-graining transformation called the "greedy coloring algorithm" (GCA) for different theoretical networks, including Erdös-Rényi, Barabási-Albert, and fractal model networks. They also study whether the fixed points of these flows are stable or unstable under slight perturbations to the networks. Afterward, we extend their work by looking at the renormalization flows of several real-world networks, including collaboration networks on the Arxiv and airport networks. We are also interested in understanding self-similarity with respect to the scale-free or small-world properties of certain networks, both in developing an analogous methodology to classify self-similar networks, and in understanding how self-similarity may better inform the other two properties.

Graph theory definitions

We will borrow several definitions from graph theory, although we will stick with "network" terminology for consistency (hence the use of "nodes" instead of "vertices") [5].

We can capture a network G as $G = (V, E)$, where V is a set of nodes and E is a set of edges. Two nodes a, b are adjacent if edge (a, b) is in E. For a node v in V, we define its degree $\deg(v)$ to be the number of nodes that are adjacent to v. The network can be directed, in which $(a, b) \neq (b, a)$ for $a, b \in V$. Then the ordering of nodes in an edge matters, perhaps indicating a flow from a to b. Alternatively, the network can be undirected, so that the ordering of nodes in an edge does not matter. In the undirected case, an edge may indicate a relation between two nodes. For either directed or undirected networks, we may have a weight function $w : E \to \mathbb{R}_{\geq 0}$, where weights are typically non-negative and associate each edge with some value, perhaps indicating the strength of an interaction or the amount of flow. Finally, we will make use of the distance function $d : V \times V \to \mathbb{R}_{\geq 0}$, where the distance between two nodes is the sum of the edge weights along the shortest path between them.

We will primarily study undirected networks with uniformly weighted edges, even though network datasets can be directed and weighted. We can either attempt to extend methods for undirected, uniformly weighted networks to the general case, or ignore the additional information (and deal with the consequences).

Two hypotheses about real-world networks

In this section, we will talk about the small-world and scale-free properties often mentioned in the studies of network datasets. There are two reasons to discuss these properties. First, what can we learn from the ways in which network datasets are analyzed and assigned these properties, in order to generalize these methods to the study of self-similarity? Second, how can we understand the relation between self-similarity and the small-world and scale-free properties?

The small-world property

In order to describe the small-world property, we first need to define two network properties, the characteristic path length and the clustering coefficient for some network $G = (V, E)$ [6]. We have that the characteristic path length of G is $L = \langle d(a, b) \rangle$, an average that is taken over all pairs $(a, b) \in V$. To evaluate the clustering coefficient C, we first let S_v be the set of v's neighbors, and let $E_v = \{(a, b) | a, b \in S_v\} \subseteq E$. Then $C_v = \frac{|E_v|}{\binom{|S_v|}{2}}$ for some $v \in V$, and $C = \langle C_v \rangle$, the average taken over all $v \in V$. Then a small-world network is characterized by a high C and a low L — it is highly clustered, yet on average any two nodes are close together.

To test for the small-world property, Watts et al. examine three network datasets based on the acting records of film actors, the organization of power grids, and the neural connectivity of *C. elegans*. They compare the L and C values of these network datasets to the L and C values of randomly generated graphs that have the same number of nodes and same average degree. They find that while $L \sim L_{random}$, C is several orders of magnitude larger than C_{random} for the datasets used. Watts et al. suggest that the high degree of connectivity in a small-world network can play to its advantage, such as enabling the rapid transmission of information through the network. In an appeal to intuition, Watts et al. also suggest that the small-world property is a common feature of real-world networks, whether designed by humans or the product of biological evolution, because it appears well-adapted to information transmission.

While intuitive, this hypothesis needs to be further tested by applying the small-world analysis to more network datasets. It is insufficient to note, as the authors do, that the datasets chosen were not cherry-picked. For this larger test, we also need a stricter definition of a small-world network — how many deviations from C_{random} should C be in order for a network to be small-world? — and such a definition is not provided by the article.

The scale-free property

In Albert et al., a scale-free network is defined by having a degree distribution that exhibits a power law above some degree threshold [7]. In this article, the scale-free property is attributed to four large network datasets: the World Wide Web, in which the nodes are webpages and edges are weblinks; the Internet, in which the nodes are routers and the edges are their adjacencies and interfaces; power grids, in which the nodes are transformers, generators, and substations and the edges are the transmission lines between them; and citation networks, in which the nodes are authors and the edges are co-authored papers.

Similar to Watts et al., the authors make a claim that the scale-free property is "common." They also make a claim about why networks are scale-free in a review article, proposing the Barabási-Albert model [2]. This is a hypothesis that the scale-free property arises due to the mechanism driving the growth of networks. In particular, the proposed mechanism is known as *preferential attachment*, in which new nodes are added to an existing network, and the new node's probability of sharing an edge with an existing node is proportional to the existing node's degree. However, while artificial networks generated by preferential attachment exhibit the scale-free property, this finding alone does not validate the hypothesis. The analysis of temporal data is needed to understand how real-world networks grow over

time, which is necessary to confirm growth mechanisms.

Because the scale-free property is demonstrated on four networks, we run into the same critique for the small-world property. How general is this result?

Broido et al. claim that few networks are scale-free [8]. To justify this claim, they analyzed 927 publicly available network datasets, using a goodness-of-fit test generated by fitting a power law model to the degree distribution and a likelihood test that fits the same degree data to alternative distributions like the log-normal. They also defined five different sets of statistical requirements for "scale-free" in order of stringency. The authors found that 52% of the datasets satisfied the "super-weak" criteria (in which the alternative distributions are not favored), while only 4% satisfied the strongest (in which the power-law has an exponent restricted to values between 2 and 3, satisfies the goodness-of-fit test, and also satisfies the likelihood test). Their results appear to contradict the belief that scale-free networks are common.

Furthermore, the authors cast doubt on the complex networks field's focus on the scale-free property and the associated preferential attachment mechanism. Because their datasets included biological, social, and information networks, they propose that there is no universal mechanism for generating such heterogenous networks.

Self-similarity in network datasets

Finally, we will discuss an experimental study of the self-similar property used to describe complex networks.

Song et al. report on the self-similarity for four examples of real-world networks, such as the Internet (circa 2005), two protein-interaction networks, and one cellular network [9]. The paper is inspired in part by an assumption that many real-world networks have the small-world property, so they have an exponential relationship between the number of nodes and the diameter of the network. Because this is not a power-law relation, it would seem that small-world networks are not scale-free. Song et al. look for power-law relationships that emerge under two different renormalization procedures, and claim that this power-law relationship reveals that while small-world networks aren't scale-free, they can be self-similar.

Two different renormalization procedures are used. The first, "box-counting," relies on covering the network with N boxes of length l (so that the nodes in a given box are connected a distance smaller than l. The power-law relationship between N and l is measured. In the second method, "cluster-growing," one seed node is chosen at random, and all the nodes a distance of at most l away from that node are clustered together to get some mass M. The power-law relationship between $\langle M \rangle$ and l is measured (where $\langle M \rangle$ is found by averaging over the M obtained from different seed nodes).

In heterogeneous networks, these two methods will give rise to different results. In particular, the relationship between $\langle M \rangle$ and l is exponential, because the hubs in the heterogeneous network are oversampled in comparison to the "box-counting" method.

This article raises questions about the relations between the small-world, scale-free, and self-similar properties, and the conditions, if any, under which one property implies another. Furthermore, in the same way that mechanisms have been proposed for small-world and scale-free networks alike, can a mechanism or explanation be given for self-similar networks?

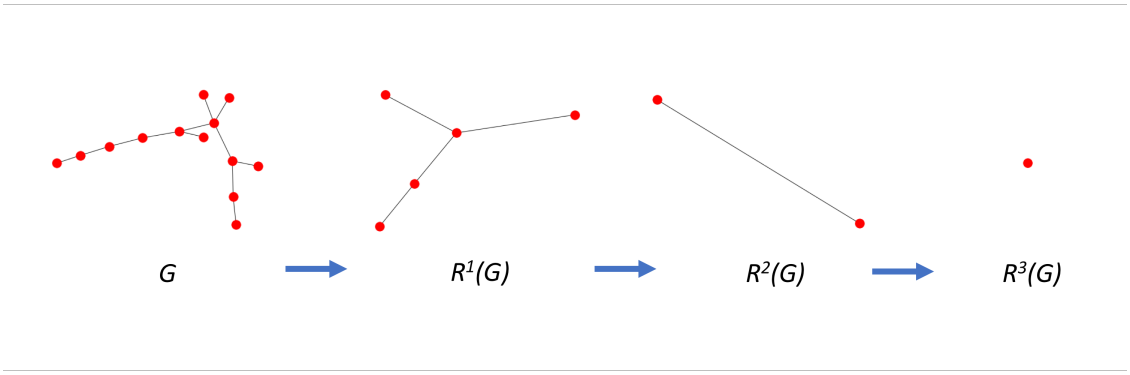

Figure 1: An example of the renormalization process in which GCA is applied three times to the initial graph G.

Theory

We first describe the greedy coloring algorithm, the renormalization method on which analyses of self-similarity in networks are based. Subsequently, we describe a growth mechanism for self-similar networks and the critical exponents arising from power-law relations between various quantities that characterize such networks.

The Greedy Coloring Algorithm

We describe one particular renormalization process for graphs called the greedy coloring algorithm (GCA), whose implementation is explained in [10].

In GCA, we begin with a graph $G_t = (V_t, E_t)$ at time t and l_B, the time-independent box size. Now, suppose we group the nodes into boxes, such that any two nodes a and b in any given box satisfy $d(a, b) \leq 2r_b$, where $l_B = 2r_B + 1$. We successfully cover (or "tile") the graph when every node belongs inside one box. We represent the boxes as nodes in the renormalized graph G_{t+1}. There is an edge between two nodes in G_{t+1} if two nodes in G_t that are now inside each of those boxes are adjacent. The goal of GCA is find a graph G_{t+1} with N_B nodes, the minimum number of boxes needed to cover G_t. This completes one iteration of the renormalization procedure.For a graph $G = G_0$, we repeatedly apply GCA until we obtain a graph G_t for some t such that G_t does not have edges. At this point, the renormalization terminates.

Because finding N_B is NP-Hard, there is no known method to efficiently compute N_B in the general case. Thus, we implement an approximate method that operates on a transformed version of $G_t = (V_t, E_t)$, which we will call $G' = (V', E')$. To construct G', we set $V' = V_t$

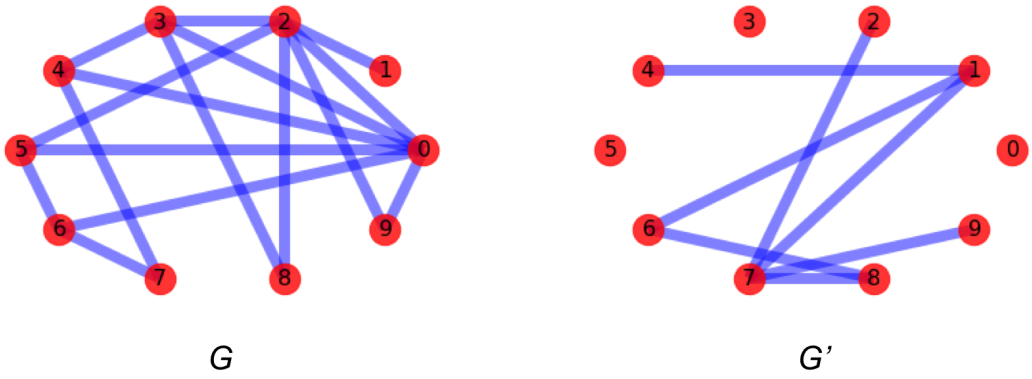

Figure 2: G (left) is a BA graph in which $n = 10$ and $m = 2$. For $l_B = 3$, we observe that the edges in G' correspond to nodes that are a distance of at least l_B apart in G.

and $E' = \{(v_i, v_j) | d(v_i, v_j) \geq l_B \text{ for } v_i, v_j \in V_t\}$. Here, $d(v_i, v_j)$ gives us the length of the shortest path between v_i and v_j. If no such path exists, we set $d(v_i, v_j) = \texttt{inf}$.

Next, as in the vertex coloring problem, we assign colors (or numerical labels) to the vertices in G' given by the coloring function $c : V' \to \mathbb{Z}_{\geq 0}$. For a coloring to be valid, the vertices along each edge are distinctly colored. Our goal is to find a color assignment that approximately uses the minimal number of colors. Once we obtain this assignment, we can transform G' and its valid coloring into G_{t+1}.

To obtain c, we first construct G' and order its vertices from highest to lowest degree.

0. Construct $G' = (V', E')$
1. Order the vertices V' from highest to lowest degree, obtaining v_1, ..., $v_{|V'|}$
2. Assign $c(v_1) = 0$
3. For i in (2:$|V'|$)
    a. Let S be an empty list
    b. For j < i

        Add c(v_j) to S if v_j is adjacent to v_i in G'
    c. Find the smallest $n \in \mathbb{Z}_{\geq 0}$ that is not in S
    d. Assign $c(v_i) = n$

Because of the way that G' is constructed, we know that any two vertices that have the same color are less than a distance l_B apart, so they can be grouped into the same box. Thus, the number of unique colors in the range of c corresponds to N_B, and vertices with the same color in G' correspond to vertices in the same box in G_{t+1}. Lastly, boxes v_i and v_j have an edge between them in G_{t+1} if there exist nodes $a \in v_i$ and $b \in v_j$ such that (a, b) is an edge in G_t.

A proposed growth mechanism for self-similar networks

A proposal for the growth mechanism of self-similar networks is as follows [11].

We begin with a star graph at time $t = 0$, which is a graph whose central node is adjacent to all other nodes. There are no additional edges. At each time step, $mk(t)$ new nodes are added to the graph for each node with degree $k(t)$. Here, m is an input parameter. As a result, the number of nodes at $t+1$ is given by $\tilde{N}(t+1) = \tilde{N}(t)+2m\tilde{K}(t)$, where $\tilde{K}(t)$ is the total number of edges in the graph at time t. Because the graph is connected and acyclic, we have that $\tilde{N}(t) = \tilde{K}(t)$, so we have that $\tilde{N}(t+1) = (2m+1)\tilde{N}(t)$. Let $n = 2m+1$.

Consider two different connectivity modes for the addition of edges at each time t. In Mode I, an edge is added between each of the new node and the node that it was added for, so the number of edges is $\tilde{k}(t+1) = m\tilde{k}(t)+\tilde{k}(t)$. Because one extra node is generated at each of the two nodes defining the diameter $\tilde{L}(t)$, we have that the diameter of the network grows linearly with t, or $\tilde{L}(t+1) = \tilde{L}(t)+2$. In Mode I, $\frac{N_B}{N} \sim \exp\left(-\frac{\ln n}{2} l_B\right)$ and $\frac{k(l_B)}{k_{hub}} \sim \exp\left(-\frac{\ln s}{2} l_B\right)$ for $s = m+1$, where k_{hub} is the highest degree in the graph and $k(l_B)$ is the highest degree of any box after performing a GCA covering. In particular, Mode I gives us a small-world network that doesn't have a fractal structure (the fractal dimensions would go to infinity because we have exponential laws).

In Mode II, the old edges are replaced with edges between the newly added nodes. If we consider the old diameter $\tilde{L}(t)$ of the graph, every edge along the path yielding this diameter has been replaced by three edges, so $\tilde{L}(t+1) = 3\tilde{L}(t)$. The degree increases as $\tilde{k}(t+1) = m\tilde{k}(t)$. Then we have $N_B(l_B) \sim l_B^{-d_B}$ where d_B is finite, and $k(l_B) \sim l_B^{-d_k}$ where d_k is also finite. This gives a fractal structure where d_B and d_k are fractal dimensions; however, because the diameter grows multiplicatively, the graph is no longer small-world.

The fractal model (FM), which results in self-similar networks, is then a stochastic combination of Modes I and II, in which edges from each of the two possible modes are chosen with probability e at each time step. This yields both finite fractal exponents and the small-world effect. Furthermore, the fractal model is characterized by anti-correlation, in which nodes with a large degree tend to be adjacent to nodes of low degree; the repulsion between "hubs" (nodes of high degree) at all length scales is necessary for the self-similarity property to emerge.

Critical exponents for the fractal model based on renormalization flows

Consider the fractal model (FM) under GCA, in which the time steps now represent the number of iterations of GCA. Then we have that

$$N_{t-1} = nN_t$$
$$k_{t-1} = sk_t$$
$$\beta = 1 + \frac{\log n}{\log s}$$

where n and s are the time-independent constants from the previous section, and β is the degree distribution exponent of the network [12]. Here, N_t is the number of nodes and k_t is the largest degree at step t of the renormalization process.

Then we have that the relative largest degree, κ_t at time step t can be given in terms of the relative network size $x_t = \frac{N_t}{N_0}$:

$$\begin{aligned}\kappa_t \sim \tfrac{K_t}{N_t} &= \frac{K_0}{N_0}(\frac{s}{n})^{-t} \\ &= \frac{K_0}{N_0}(\frac{N_t}{N_0})^{-\frac{(\beta-2)}{(\beta-1)}} \\ &= \frac{K_0}{N_0}x_t^{-\frac{(\beta-2)}{(\beta-1)}} \\ &\sim (N_0 x_t)^{-\frac{(\beta-2)}{(\beta-1)}}.\end{aligned}$$

Considering κ_t to be a scaling law as a function of $x_t N_0^{1/\nu}$, we see that $\nu = 1$ for any value of β. Radicchi et al. study renormalization flows based on the relative system volume x_t, rather than as a function of l_B (or an "order parameter" of the system). The authors show that this method is consistent with the extraction of critical exponents based on functions of "order parameters" by applying the renormalization-flow process to percolation and the two-dimensional Ising model. They recover the familiar critical exponents of each from examining relative percolation strength and magnetization, respectively, as functions of $x_t L_0$ (L_0 being the lattice size for each model).

Procedures

All code was written in `python`. The GCA and FM model generation algorithms were implemented from scratch. Other algorithms, such as the ones used to generate Erdös-Renyí, Barabasí-Albert, and Watts-Strogatz graphs, were available through the `networkx` package. The Grace computing cluster was used to generate all graph data, while data analysis was performed locally in Jupyter notebooks.

We now go through in detail the definitions and generation procedures of each of the four types of graphs that will be analyzed under GCA renormalization flows.

Erdös-Renyí graph generation

The parameters of an Erdös-Renyí (ER) graph are n, the number of nodes, and p, the probability that each possible edge is chosen. (In particular, these are the parameters accepted by the `networkx` Python package, which we use throughout our data generation and analysis procedures.)

Equivalently, we can characterize an ER graph $G = G(V, E)$ by n and $\langle k \rangle$, the average degree of each node. This characterization is done in Radicchi et al. (in which $\langle k \rangle = 2$). We need to convert between p and k. Because $\sum_{v \in V} \deg(v) = n\langle k \rangle = 2|E|$, we have that $|E| = \frac{n\langle k \rangle}{2}$. Because there are $\binom{n}{2}$ edges in the complete graph with n nodes, and $|E|$ of these possible edges were chosen for G, we have that $p = \frac{\langle k \rangle}{n-1}$, and that $\langle k \rangle = p(n-1)$. To generate an ER graph, we iterate through all the possible edges and choose each with probability p.

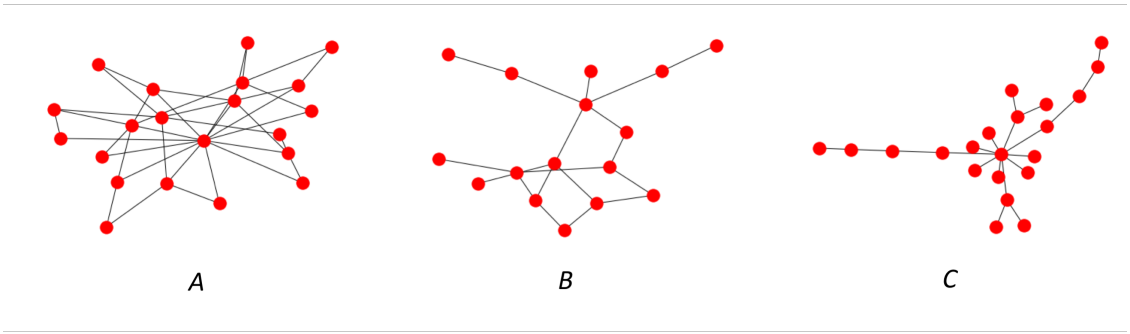

Figure 3: Examples of connected graphs that are produced by different generation processes. A. Barabasí-Albert graph with $n = 21$ and $m = 2$. B. Erdös-Renyí graph with $n = 16$ and $p = \frac{1}{8}$. C. Fractal model graph with $n = 21$, $m = 2$, and $e = 0.5$.

Fractal model graph generation

Having previously described the fractal model's growth mechanism, we now formalize FM graph generation. Here, e is the probability of selecting an edge from Mode I, and m controls the number of edges that are added at each iteration. The maximum node number is N.

0. Start with $G = (V = 0, E = \emptyset)$, N, m, and e
1. Add 4 vertices $\{i\}_{i=1}^{4}$ to V and edges $\{(i, 0)\}_{i=1}^{4}$ to E, so that G is a star graph
2. While $|V| + 2m|E| < N$
    a. Let $E_{old} = E$, the set of edges in G
    b. For v in V

        Add $m\text{deg}(v)$ new nodes to V, which we call the set $A = \{a_{vi}\}_{i=1}^{m\text{deg}(v)}$
        For a in A
            Add (a, v) to E
    c. For (a, b) in E_{old}

        With probability $(1 - e)$, remove (a, b) from E and add an edge between a randomly drawn vertex from $X = \{i|(a, i) \in E \text{ and } (a, i) \notin E_{old}\}$ and a randomly drawn vertex from $Y = \{i|(b, i) \in E \text{ and } (b, i) \notin E_{old}\}$ to E

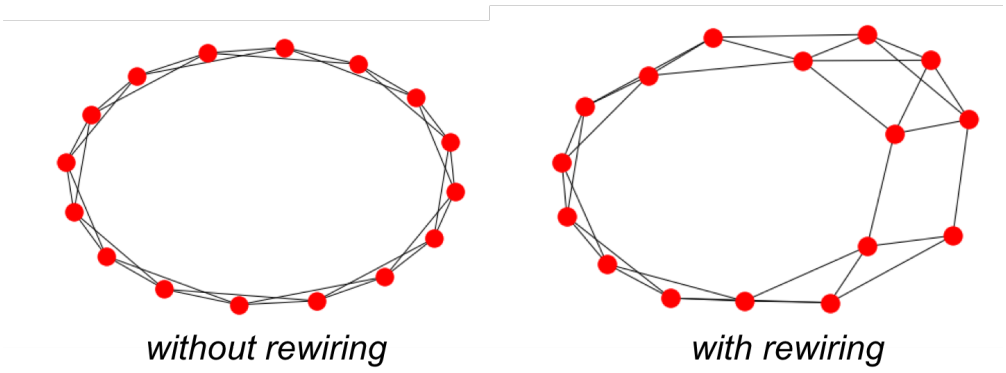

Figure 4: Two Watts-Strogatz graphs, one without rewiring (left) and one with rewiring (right). In both, $n = 16$ and $k = 4$. The rewiring probability for the graph on the right is 0.05.

Barabasí-Albert graph generation

The parameters for a Barabasí-Albert (BA) graph are n, the number of nodes, and m, the number of edges to attach from a new node to existing nodes.

To generate a BA graph, we begin with a single node. The method for adding nodes and edges via preferential attachment is as follows:

0. Initialize $G = (V = \{0\}, E = \emptyset)$
1. For i in (1:n)
    a. Choose m elements without replacement from $S = \{1, ..., n-1\}$ where the probability of selecting $j \in S$ is given by $\frac{\deg(j)}{\sum_{k \in S} \deg(k)}$
    b. For each element a of the m chosen elements, add edge (i, a) to E

Watts-Strogatz graph generation

The parameters for a Watts-Strogatz (WS) graph are n, k, the number of nearest neighbors that each node is initially connected to, and p, the probability of "rewiring" each edge. For simplicity, we assume k is even. To generate a WS graph, we first need to create the ring graph on n nodes with k nearest neighbors, which is done in step 2. below. Then, we need to do the rewirings, as shown in step 1.

0. Initialize $G = (V = \{1, ..., n\}, E = \emptyset)$
1. Consider $G_{cycle} = (V = \{1, ..., n\}, E = \{(i, i+1)\}_{i=1}^{n-1} \cup \{(n, 1)\})$, a cyclic graph
2. For each i in V, find its k nearest neighbors N_i in G_{cycle}. Set $E = E \bigcup_{j \in N_i} (i, j)$.
3. For i in (1:n)

a. Let A be the set of edges (i, a) in G_{cycle}, and B be the set of edges (i, b) in G. Consider $E_i = A \cap B$.

b. For each e in E_i, remove e from E with probability p, and add an edge (i, c) to E, where $c \in V \backslash \{v | (i, v) \in E\}$ is selected uniformly randomly

Analysis

For a graph of a given type with N_0 nodes, we perform the renormalization procedure and calculate the relative maximum degree at each time step, $\kappa_t = \frac{\max \deg(G_t)}{N_t - 1}$, as well as the relative graph size $x_t = \frac{N_t}{N_0}$.

For each step t, initial number of nodes N_0, and graph type, we repeat the generation of graphs and the subsequent renormalization procedure 10 times. We can average across the set S of 10 graphs and report the mean κ_t and mean x_t for those graphs. Then the error of the κ_t is given by $\sigma_{\kappa_t}^2 = \sqrt{\langle \kappa_t^2 \rangle - \langle \kappa_t \rangle^2}$. We also find the susceptibility $\chi_t = N_0(\langle \kappa_t^2 \rangle - \langle \kappa_t \rangle^2)$ by finding the variance of the κ_t for each $i \in S$. Lastly, we have that the variance of χ_t, $\mathrm{Var}(\chi_t)$, is given by

$$\begin{aligned} &\mathrm{Var}(\chi_t) \\ &\quad = \mathrm{Var}(N_0 \mathrm{Var}(\kappa_t)) \\ &\quad = N_0^2 \mathrm{Var}(\mathrm{Var}(\kappa_t)) \\ &\quad \approx \frac{N_0^2}{|S|}(\mu_4 - \frac{|S| - 3}{|S| - 1}\mu_2^2) \end{aligned}$$

where $|S| = 10$. Here, μ_i is the ith moment of κ_t, so $\mu_i = \frac{1}{|S|}\sum_j (\kappa_t)_j^i$ where j indexes the elements of S [13]. Then the error of the χ_t is given by $\sigma_{\chi_t} = \sqrt{\mathrm{Var}(\chi_t)}$.

We plot k_t and χ_t as functions of x_t for each type of graph. We also plot k_t as a function of $x_t N_0^{1/\nu}$ and $x_t N_0^{1/\gamma}$, in order to study the scaling exponents, ν and γ. Although these exponents differ for different types of graphs, they can be found regardless of whether or not the graph is self-similar [12].

We begin with ER graphs. For all ER graphs analyzed, $\langle k \rangle = 2$, which corresponds to a p of $\frac{2}{n-1}$. Then, p exceeds the critical probability $p_c = \frac{1}{n}$ for $n > 1$, so with high probability the ER graph will have a giant connected component. However, of the various types of graphs that we are considering, only the ER model generates an initial graph that isn't necessarily connected – there can be two nodes in the graph such that no path between them exists. Moreover, the initial graph can have isolated nodes that don't transform under the renormalization procedure; the number of these isolated nodes will only increase with the number of renormalization steps. When we initially analyzed ER graphs, we included the isolated nodes in the analysis; however, this resulted in a spike downward in κ_t's value when x_t is small, due to the accumulation of isolated nodes. As the plots in Radicchi et al. do not face this issue, we assume moving forward that only the giant connected component is considered in the computation of κ_t. This isn't an issue for the other graphs, because they are by construction connected from the start. Then after any number of steps, the renormalization procedure will continue to yield connected graphs.

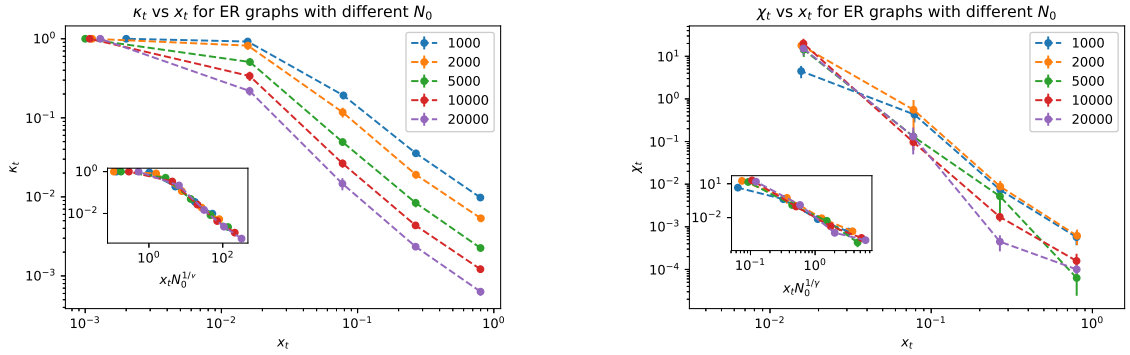

Figure 5: GCA algorithm applied to ER graphs of various initial sizes, given in the legend. We have that $\langle k \rangle = 2$ and $l_B = 3$. On the left, we have the plot of κ_t as a function of x_t. On the right, we have the plot of χ_t as a function of x_t. The insets are plots of κ_t and χ_t as functions of $x_t N_0^{1/\nu}$ and $x_t N_0^{1/\gamma}$, respectively. We get that $\nu = 1.65 \pm 0.04$ and $\gamma = 4.9 \pm 0.1$.

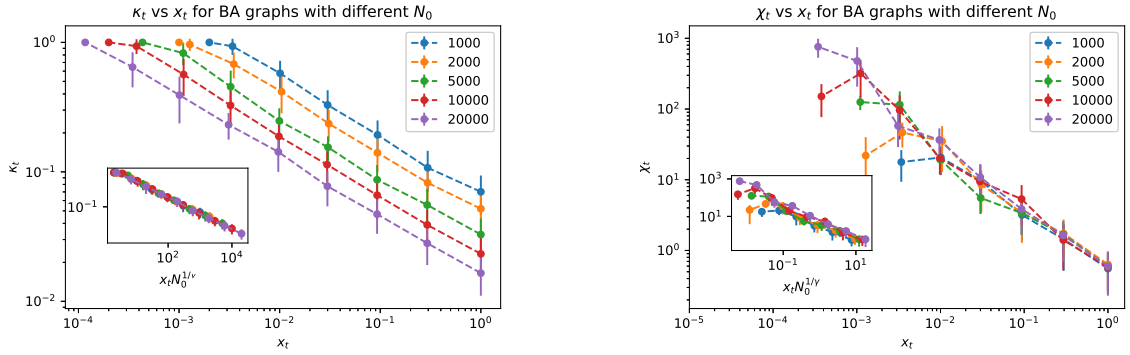

Figure 6: GCA algorithm applied to BA graphs of various initial sizes. We have that $m = 1$ and $l_B = 3$. On the left, we have the plot of κ_t as a function of x_t, and $\nu = 1.00 \pm 0.02$. On the right, we have the plot of χ_t as a function of x_t, and $\gamma = 3.4 \pm 0.1$.

In Figure 5, we see in the insets the "data collapse" observed in Radicchi et al., which indicate that κ_t and χ_t are scaling functions of $x_t N_0^{1/\nu}$ and $x_t N_0^{1/\gamma}$, respectively. As the graph of κ_t against x_t is nonlinear, we conclude that the ER graph is not self-similar under GCA transformations.

The authors make various claims based on the empirical findings for ν and γ – they observe that $\nu = \gamma$ (within errors), that $\nu = 2$ for graphs that aren't self-similar and that ν varies for self-similar graphs. However, they also add the caveat that results may vary based on the specific transformation that is adopted. From our ER results, we get a somewhat different value for ν, but ν does not equal γ. It could be the case that we didn't average over sufficiently many instances of ER graphs.

For BA graphs, we have that $\nu = 1.00 \pm 0.02$, which is consistent with the value derived for the fractal network. That the BA graph is self-similar is also suggested by the linearity of the κ_t plots over several orders of magnitude of x_t – in particular, for $N_0 = 20,000$, we see a linear plot over a range of x_t of $[10^{-4}, 1]$. Because the BA graph is scale-free, this suggests that some scale-free graphs also satisfy self-similarity. On the other hand, we have

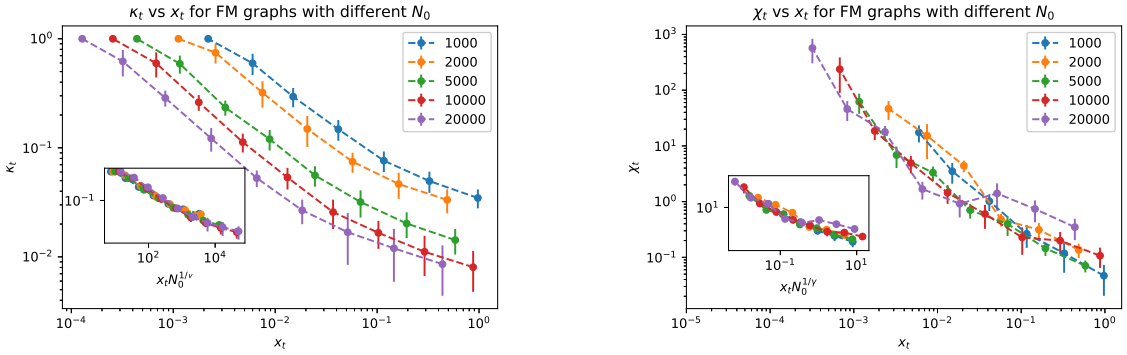

Figure 7: GCA algorithm applied to FM graphs of various initial sizes. We have that $l_B = 3$, $m = 1$, and $e = 0.5$. On the left, we have the plot of κ_t as a function of x_t. On the right, we have the plot of χ_t as a function of x_t. In the insets, we have that $\nu = 0.85 \pm 0.02$ on the left and $\gamma = 3.3 \pm 0.1$ on the right.

that $\gamma = 3.4 \pm 0.1$, which is again not in agreement with the authors' findings that $\nu = \gamma$.

For the fractal model, we observe that $\nu = 0.85 \pm 0.02$ for the κ_t scaling function in the fractal model; while this doesn't equal the predicted value of 1 with errors, it's not that far off. Furthermore, while the fractal model theory predicts a linear relationship in the log-log plots of Figure 7, we see that the κ_t plot appears to have a curvature, rendering the relationship nonlinear. This discrepancy could instead be an artifact of our implementation of the fractal model, which produces a larger variance in x_t than the other models. Then it could be the case that individual plots look linear, but the averaged plots for 10 graphs is instead nonlinear. For example, we show in Figure 8 examples of κ_t curves for individual graphs of different initial sizes. As the curvature is less clear, a followup step would be to compare the errors of linear fits to individual plots, with the errors of linear fits to the averaged plot.

Overall, we do get that the ν values of the BA and FM graphs are closer to 1 while that of the non-self-similar ER graph is closer to 2, which is consistent with the author's findings. However, we do not find that $\nu = \gamma$ for any of the graphs analyzed. Identifying a theoretical basis to explain this empirical relation would be a good starting point. Alternatively, we could try increasing the size of S, or varying the transformation method for a fixed graph and reporting the resulting variance in the results.

Perturbations

We study the change in exponent values when graphs that are self-similar under the GCA transformation are slightly perturbed. Because self-similar graphs are the fixed points of the renormalization transformation, perturbing them is one way to characterize the stability of these fixed points.

First, we look at WS graphs. We consider WS graphs without rewirings to be non-perturbed; as a result, there is only one non-perturbed WS graph for each pair of parameter values (n, k) because $p = 0$ and no rewirings are performed.

As seen on the right in Figure 9, the rewiring perturbation results in a nonlinear relation-

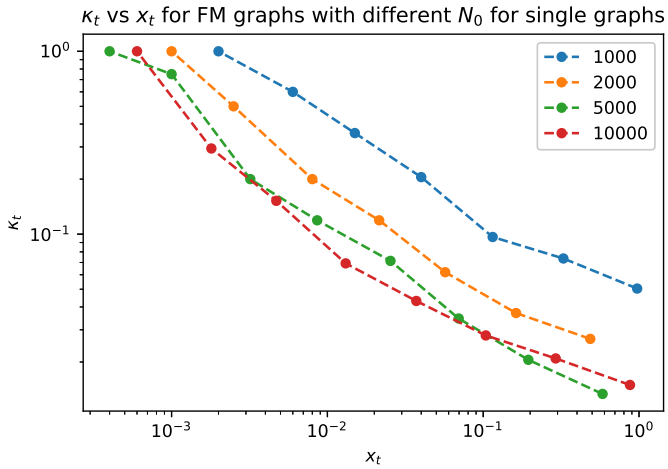

Figure 8: Results from applying the GCA algorithm to single FM graphs of various initial sizes.

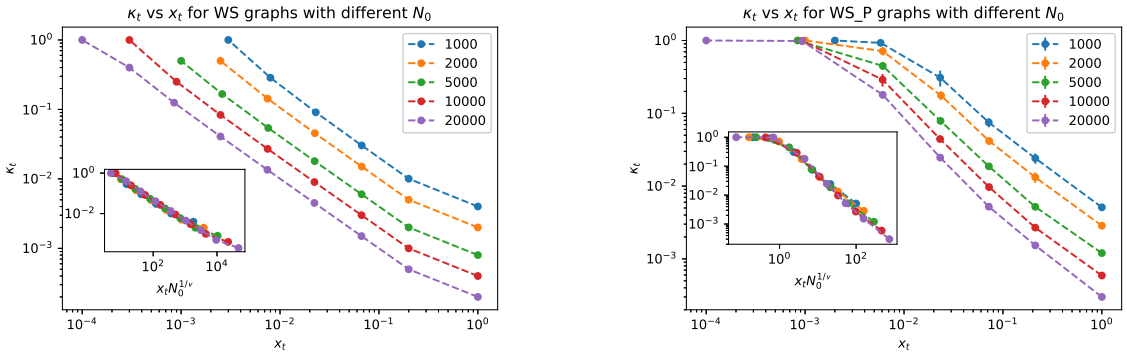

Figure 9: GCA algorithm applied to WS graphs of various initial sizes, with and without rewiring. On the left, we have the plot of κ_t as a function of x_t for WS graphs with $k = 4$ and no rewirings. On the right, we have the plot of κ_t as a function of x_t for WS graphs with $\langle k \rangle = 4$ and a rewiring probability $p = 0.01$. In the insets, we have that $\nu = 0.92 \pm 0.02$ on the left and $\nu = 1.40 \pm 0.05$ on the right.

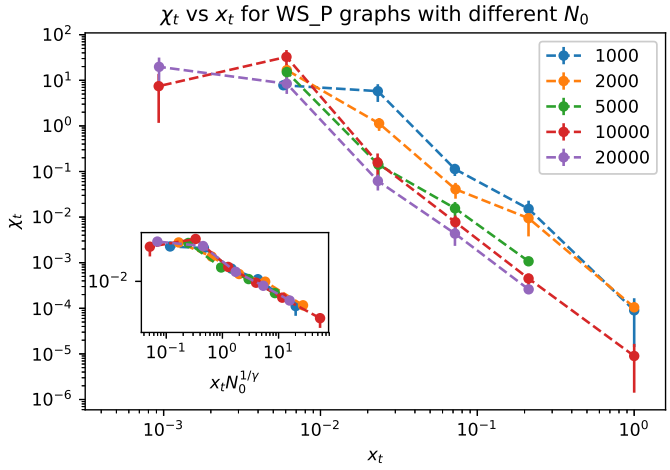

Figure 10: Because the WS graphs without rewirings are deterministically generated, the χ_t value is not meaningful (the variance is 0) and doesn't serve as a basis for comparison. Then, we plot χ_t as a function of x_t for the perturbed WS graphs, with the same parameters as in 9. We obtain $\gamma = 2.3 \pm 0.1$.

ship between κ_t and x_t. The exponent ν also changes from 0.92 ± 0.02, close to the values of other self-similar graphs, to 1.40 ± 0.05, which is closer to the values for non-self-similar graphs. This indicates that the fixed point given by a WS graph without rewiring is unstable, replicating the findings of the authors.

We cannot compare γ from the χ_t plots of perturbed WS graphs to a corresponding plot for nonperturbed WS graphs, the relatively low value of $\gamma = 2.3 \pm 0.1$ is also hard to compare to our empirical γ values for both self-similar and non-self-similar graphs. Also, in general, the uncertainties on γ are several times higher than for ν, given the apparent fluctuations and increase in noise when we go from κ_t to χ_t. Then from χ_t plots alone, it is difficult to draw conclusions about the perturbations' impact on WS networks.

Next, we look at perturbed BA graphs in Figure 11. To perturb a BA graph, we first generate the BA graph $G = (V, E)$ as usual. Then we add $0.05|E|$ edges to the graph, choosing uniformly and without replacement from all possible (non-looping) edges. We see that adding additional edges disturbs the linearity of the $kappa_t$ plots, so these graphs are no longer self-similar under the GCA transformation. Likewise, we have that $\nu = 1.4 \pm 0.1$, so the exponent significantly increased from its prior, unperturbed value of $\nu = 1.00 \pm 0.02$. As a result, this fixed point is unstable as well.

Finally, we look at perturbed FM graphs in Figure 12. To perturb the FM model, we first generate an FM graph as usual, and then add $0.05|E|$ edges to the graph in the same manner that we did for the perturbed BA graph. Thus, we observe an increase in ν from 0.85 ± 0.02 to 1.35 ± 0.05, and a less clear, but probable increase in γ from 3.3 ± 0.1 to 3.5 ± 0.2. While the increase that we see is not as large as the authors' observations, in which they recover $\nu = 2$, the empirical exponent for non-self-similar graphs, it is still substantial, and supports the idea that the FM graph is also an unstable fixed point in the GCA transformation. Thus, all the graphs that we perturb are unstable fixed points.

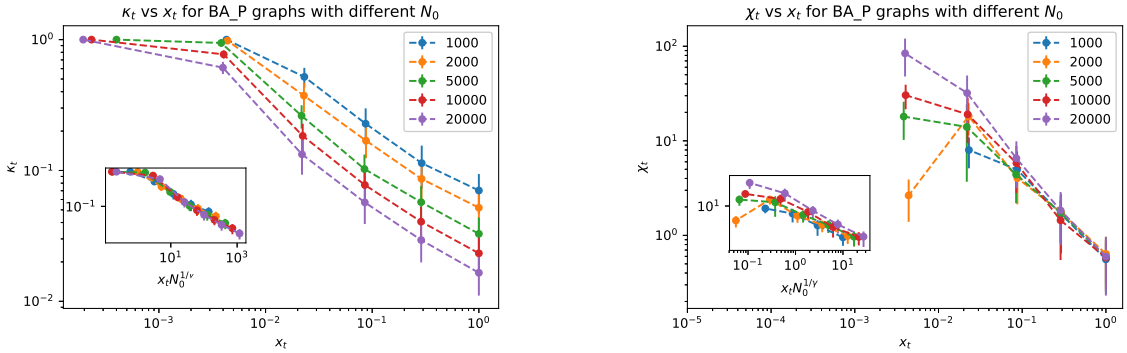

Figure 11: GCA algorithm applied to perturbed BA graphs of various initial sizes. We have that $l_B = 3$ and $m = 1$. On the left, we have the plot of κ_t as a function of x_t. On the right, we have the plot of χ_t as a function of x_t. In the insets, we have that $\nu = 1.4 \pm 0.1$ on the left and $\gamma = 3 \pm 0.2$ on the right.

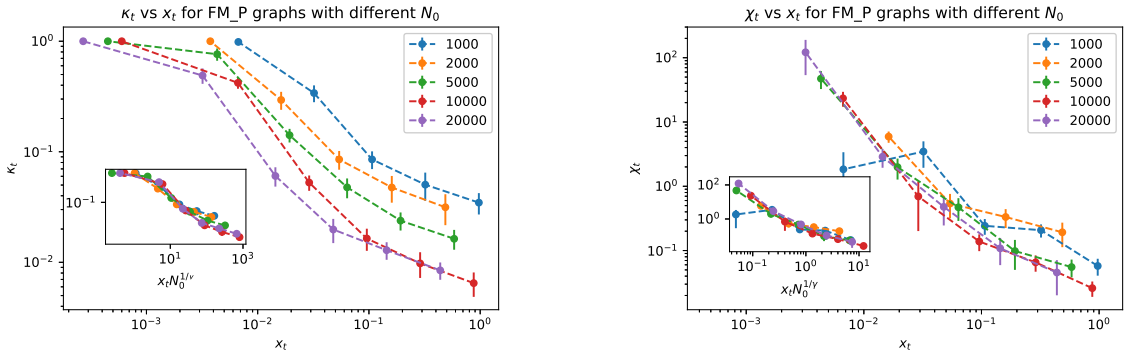

Figure 12: GCA algorithm applied to perturbed FM graphs of various initial sizes. We have that $l_B = 3$, $m = 1$, and $e = 0.5$. On the left, we have the plot of κ_t as a function of x_t. On the right, we have the plot of χ_t as a function of x_t. In the insets, we have that $\nu = 1.35 \pm 0.05$ on the left and $\gamma = 3.5 \pm 0.2$ on the right.

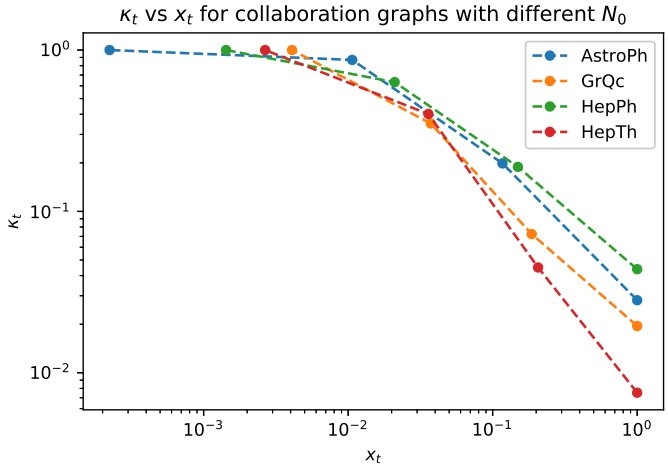

Figure 13: We plot the κ_t as a function of x_t for collaboration networks of different Arxiv sections.

Real-world networks

To extend the work of Radicchi et al., which is concerned with the renormalization flows of different theoretical graphs, we now consider datasets of real-world networks. In particular, in order to select datasets, we looked for datasets that capture similar kinds of data yet vary in graph size, similar to the simulations for different graph sizes that were performed above.

The first sets of data that we use come from the Stanford Network Analysis Project. The network data represent collaboration between authors on different physics sections of the Arxiv between January 1993 and April 2003 [14]. For each physics section, the authors are nodes in the graph, and an edge is established between two authors if both authors are listed on a paper in that Arxiv section during that time period.

The physics sections that we look at are "Astro" Physics (AstroPh), "General Relativity" (GrQc), "High Energy Physics Phenomenology" (HepPh), and "High Energy Physics Theory" (HepTh). We summarize their graph sizes and number of edges in Table 1 below (as in the case of the Erdos-Renyi graphs, we take the largest connected component to analyze, which consists of nearly all of the nodes in the original graph in all cases.)

Collaboration networks		
Arxiv section	Nodes	Edges
AstroPh	17,903	197,031
GrQc	4,158	13,428
HepPh	11,204	117,649
HepTh	8,638	24,827

Table 1: The sizes and number of edges of collaboration networks representing different Arxiv sections.

The κ_t plots in Figure 13 are not linear, so these collaboration networks are not self-similar under GCA. Also, we have that the networks aren't particularly similar to each other

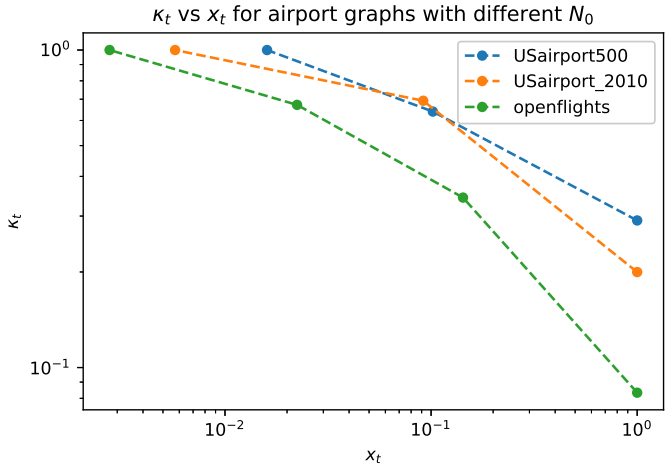

Figure 14: We plot the κ_t as a function of x_t for airport networks.

either, which is seen through the way the flows intersect – in contrast, the theoretical graphs of varying sizes that we looked at have flows that don't intersect prior to the last iteration of GCA. As a result, it's unclear what structure or model these collaboration networks can be classified as, although they appear to be mutually distinct.

The second type is airport data from [15], in which airports are nodes and an edge exists between two nodes if there is a flight between them in a certain year. In particular, the datasets are of flights between the 500 busiest commercial airports in the U.S. in 2003 ("USairport500"), flights between all U.S. airports in 2010 ("USairport_2010"), and flights between all airports in the world as crowd-sourced from Openflight ("openflights") [16, 17]. Their graph sizes are summarized in Table 2 below.

Airport networks		
Airport dataset	Nodes	Edges
USairport500	500	2,980
USairport_2010	1,574	17,215
openflights	2,939	15,677

Table 2: The sizes and number of edges of different airport networks.

Similar to our results for the collaboration networks, we observe that the κ_t plots in Figure 14 are not linear, so airport networks are also not self-similar under GCA. Also, we again have that the networks aren't particularly similar to each other either, although we wouldn't expect the U.S. to be representative of the flights around the world. If anything, the three iterations of the U.S. 500 flights look the most linear of the three results. A self-similar network is one in which there is an even distribution of "hubs" at every scale; one interpretation for the concavity of the "openflights" plot is that there are many airports of relatively low degree, and after one application of GCA these all contract into a new, even more highly connected hub. Then, given that the graph becomes more and more densely connected, few GCA applications are needed before the algorithm terminates.

Discussion

Relating self-similarity to the small-world and scale-free properties

In this project we analyzed scale-free BA networks, finding them to be self-similar under the GCA transformation. However, these two properties are not, in general, the same: a power-law distribution over the degrees of the nodes in a network, indicating the scale-free property, does not imply length-scale invariance, indicating the self-similar property. One counterexample is the Internet, which has a scale-free graphical representation, yet fails to have a fractal topology [11].

Second, we see while the unperturbed WS network is self-similar, it is not small-world because the characteristic path length is large. Once we perturb the network by adding "shortcuts" that shrink the characteristic path length, we find that the WS small-world network is not self-similar. At the same time, we have that the fractal model is self-similar and small-world. This tells us that self-similarity appears to be independent of small-world; one property can hold but not the other.

Thus, characterizing the self-similarity and the renormalization flows of various graphs presents a distinct view and analysis methodology of networks that isn't captured by the small-world and scale-free properties.

The significance of unstable fixed points

Of the self-similar networks that we identified – ring graphs (WS graphs without perturbation), the FM model graphs, and BA graphs, we saw that a small perturbation resulted in graphs that were no longer self-similar. Unlike the authors, the ν values for the perturbed graphs weren't 2, but fell within the range of 1 to 2 – possibly indicating an intermediate state between self-similar and non-self-similar networks. The γ values were more sensitive, because they captured the fluctuations in κ_t across a set of graphs. This suggests that the self-similar property is somewhat fragile. In particular, as defined by renormalization flows, self-similarity is unlikely to be found in networks that are noisy, such as real networks.

This raises the question of relaxing or revising the definition of self-similarity in order to capture an approximately self-similar network. For example, if a network only differs in 5% of its edges (the perturbation that we applied) from a self-similar network, could it still be recognized as approximately self-similar? Developing a definition of self-similar that is robust to small perturbations would be helpful.

Applicability to real-world networks

The examples of real-world collaboration and airport networks that we gave were not self-similar under the GCA transformation. They were also small and dense, whereas we would hazard a guess that GCA renormalization works best for sparse, large networks.

For example, a larger network is more likely to take more iterations of GCA to renormalize, giving us more data and a better informed understanding of the renormalization flows. Considering connected networks in which the average degree of the network exceeds 1, the maximum iterations we could possibly yield comes from the case of the self-similar

line graph. Renormalizing a line graph takes $\log_2(n)$ iterations, so we expect fewer than 10 iterations for graphs with fewer than 1000 nodes. Because a line graph is not at all like a real-world network, it isn't a close upper bound for small networks – a small real-world network could take far fewer iterations to renormalize, leaving us with little k_t data. With larger networks, this upper bound increases, bettering the chances of an increased number of iterations; however, the bound is also dependent on the sparsity of the network – networks that are densely connected will necessarily contract and become at least as dense as they were prior to the GCA application.

Then one extension of the project is to apply GCA to large, sparse, real-world networks, such as genetic networks. Given computational memory constraints, we were unable to run network analyses for graphs exceeding 20,000 nodes – whereas genetic networks can contain millions of nodes. Rewriting the program extensively and in a more memory-efficient way could enable this analysis, yielding informative results. Exploring the possibility of self-similarity in real-world networks is interesting in its own right. It reveals the fractal structure of a network, and the exponents extracted from the renormalization flows constitute potential universality classes by which networks can be classified.

Outlook

We studied the renormalization flows of κ_t and χ_t on several types of theoretical graphs, providing the BA graph analyses as an addition to the ER and FM graphs that were presented in the original article. In particular, our observations that self-similar graphs have ν close to 1, whereas graphs that aren't self-similar have ν between 1.3 and 2, are consistent with the authors' findings. We also studied the effects of perturbations on self-similar graphs, finding that WS, BA, and FM graphs all constitute unstable fixed points under the GCA transformation. Lastly, we extended the work of Radicchi et al. by applying the GCA transformation to examples of real-world networks, such as collaboration and airport networks, finding that none of these networks are self-similar.

A lot more can be done to build on these results. Because it appears that self-similarity could provides a different way to classify networks, there are various worthwhile directions that we could take next. In terms of theory, we need to find an explanation for the empirical values of ν and γ of theoretical graphs; besides the exponents for the FM model, the other values are purely empirical currently. In terms of numerical experiments, the parameters that we chose are the ones that are reported in the paper; however, we can vary l_B and vary the specific parameters for each graph to examine the robustness of the GCA transformation under a variety of circumstances. In terms of real-world networks, we could apply GCA transformations to large, sparse networks to identify self-similar structures.

We could also extend GCA itself. One potentially fruitful question would be to apply GCA to weighted networks, because we can't currently capture spatial information – just the relational information between any two nodes. Because our implementation begins with computing shortest distances between any two nodes in a given graph, this automatically extends to the case of a weighted network. From there, determining the coloring is straightforward. By defining, for instance, the distances between any two new nodes in the renormalized network as the shortest distance between any two nodes (one in each of the

two new nodes) in the old network, we can complete the implementation of the weighted GCA and apply it to weighted networks. Another extension of GCA is to the study of hypergraphs, or graphs whose edges can have more than 2 nodes. Because a hypergraph is capable of capturing more than pair-wise relations, we could look for self-similarity in more complicated, higher-dimensional networks.

References

[1] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] Eugene P. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. In *Mathematics and Science*, pages 291–306. World Scientific, 1990.

[4] Mark K. Transtrum, Benjamin B. Machta, Kevin S. Brown, Bryan C. Daniels, Christopher R. Myers, and James P. Sethna. Perspective: Sloppiness and emergent theories in physics, biology, and beyond. *The Journal of Chemical Physics*, 143(1):010901, 2015.

[5] Jiri Matousek and Jaroslav Nesetril. *Invitation to discrete mathematics*. Oxford University Press, 2009.

[6] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[7] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[8] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *arXiv preprint arXiv:1801.03400*, 2018.

[9] Chaoming Song, Shlomo Havlin, and Hernan A Makse. Self-similarity of complex networks. *Nature*, 433(7024):392, 2005.

[10] Chaoming Song, Lazaros K Gallos, Shlomo Havlin, and Hernán A Makse. How to calculate the fractal dimension of a complex network: the box covering algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(03):P03006, 2007.

[11] Chaoming Song, Shlomo Havlin, and Hernán A Makse. Supplementary information for origins of fractality in the growth of complex networks. *Nature Physics*, 2(4):275, 2006.

[12] Filippo Radicchi, José J Ramasco, Alain Barrat, and Santo Fortunato. Complex networks renormalization: Flows and fixed points. *Physical review letters*, 101(14):148701, 2008.

[13] Wikipedia contributors. Variance — Wikipedia, the free encyclopedia, 2016. [Online; accessed 15-December-2018].

[14] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[15] Tore Opsahl. Datasets. https://toreopsahl.com/datasets/#usairports. [Online; accessed 17-December-2018].

[16] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3(4):276, 2007.

[17] Tore Opsahl. Why anchorage is not (that) important: Binary ties and sample selection. https://toreopsahl.com/2011/08/12/why-anchorage-is-not-that-important-binary-ties-and-sample-selection/. [Online; accessed 17-December-2018].