Homework 3: Generative Models

Ben Zuckier

Network Science II

1 Good to be Early

We will consider a Barabási-Albert model of a new field in physics with constant growth, after 5 years and 8,000 papers. Since we have linear growth, the number of papers is the same as the time t.

1. Now we compare the number of citations for first author in the field versus the author of paper number 20.

   Let $k_i(t)$ be the number of citations at time t for the paper published at time t_i. We know that $k_i(t) = m\sqrt{t/t_i}$ where m is the number of links that are added with each new node. We can compare this as a ratio:

   $$\frac{k_1(t)}{k_{20}(t)} = \frac{\cancel{m}\sqrt{\cancel{t}/1}}{\cancel{m}\sqrt{\cancel{t}/20}} = \frac{\sqrt{1}}{\sqrt{1/20}} = \sqrt{20} = 4.47214$$

2. Let $\mathfrak{t} = 8{,}000$, the time of the model "now".

   We will find a time t s.t. paper 20 will have as many citations as paper 1 had at time $\mathfrak{t} = 8{,}000$.

   Find t s.t. $k_{20}(t) = k_1(\mathfrak{t})$.

   $$m\sqrt{\frac{t}{20}} = m\sqrt{\frac{\mathfrak{t}}{1}} \Rightarrow \frac{t}{20} = \mathfrak{t} = 8{,}000 \quad \therefore \; t = 160{,}000$$

   At the time when there are 160,000 papers published, paper 20 will have the same number of citations as the first paper did at time $\mathfrak{t} = 8{,}000$.

   At the rate of $8{,}000/5 = 1{,}600$ papers per year, in $\mathfrak{t} - t = 160{,}000 - 8{,}000 = 152{,}000$ more papers, or 95 more years, paper 20 will have the same number of citations as paper 1 from the fifth year.

3. Let us find an expression (dependent on m, here the average bibliography length) to calculate the average number of citations per paper, for papers published between two times t_1 and t_2, where the times are represented as a fraction of the total network growth time $t_1 = \alpha t$ and $t2 = \beta t$.

   Assuming that $\alpha, \beta \mid t$. Let Δ be the number of time steps (papers) from $\alpha t \to \beta t$, inclusive of both ends, so $\Delta = \beta t - \alpha t + 1$. We need average k at time t for nodes created αt to βt. Average $k(t)$ for nodes in $\{v\}$ is $\sum_{i \in v} k_i(t)/[i \in v]$

(where $[\dots]$ is Iverson bracket notation, the denominator is the count of nodes in v) or average $k_{\alpha t}(t) \to k_{\beta t}(t)$.

$$\Rightarrow \overline{k_i}(t) \mid i \in [\alpha t, \beta t] = \frac{1}{\Delta} \sum_{i=\alpha t}^{\beta t} k_i(t)$$

and we know $k_i(t) = m\sqrt{t/t_i}$, so

$$\Rightarrow \frac{m}{\Delta} \sum_{i=\alpha t}^{\beta t} \sqrt{\frac{t}{i}} = \frac{m}{\beta t - \alpha t + 1} \sum_{i=\alpha t}^{\beta t} \sqrt{\frac{t}{i}}$$

4. Given $m = 2$ (and assuming $t = 8{,}000$ we can calculate the following averages using the above expression:

   - The average number of citations of the first 5% of the papers published in the field. This is the first paper through 5% of the total papers, so we have $\alpha t = 1$ so $\alpha = t^{-1} = 1/8{,}000$ and $\beta = 0.05$.

     $$\overline{k}_i(t) \mid i \in [1, 0.05t] = 172.47$$

   - The average number of citations of the papers in the 5% of papers published between the 50% and 55% papers published.

     $$\overline{k}_i(t) \mid i \in [0.5t, 0.55t] = 27.61$$

   - The average number of citations of the last 5% of the papers published in the field. So $\beta t = t$

     $$\overline{k}_i(t) \mid i \in [0.95t, t] = 20.26$$

5. This is because old papers can not cite new papers (we don't edit old papers, the size of the bibliography per paper is always m), and only new papers make connections (with a fixed probability to a given paper, based on that paper's degree). So all we have to know is how many new papers there have been in a given range and the probability that each paper has with linking.

2 Growth Without Preferential Attachment

We have a network that grows one node at a time in n with m new links assigned at *random* among the n existing nodes.

1. When we add a new node at a time step $t+1$, the probability a node gets a link is $1/n$ for each of the m new edges introduced by the new node, meaning that any node gets m/n new links on average upon adding a new node.

   The number of nodes with degree k at time t is $tp_k(t)$, and $t = n$ since we add one node at each time step, giving us $np_k(t)$ nodes of degree k at time t. Therefore the number of nodes that get a new edge at time t is $np_k(t) \times m/n = p_k(t) \times m$.

   These are the nodes that go from degree $k \to (k+1)$ at time t, but we also gain new nodes that go from degree $(k-1) \to k$. This is given by the number of nodes with degree $(k-1)$ times the probability of new links, or $np_{k-1}(t) \times m/n = np_{k-1}(t) \times m$.

   In total, the change in nodes of degree k is losing nodes that started with degree k and went to $k+1$, and gaining the nodes starting as $k-1$ that go to k. Or $p_{k-1}(t) \times m - p_k(t) \times m$.

   And we had $np_k(t)$ nodes of degree k to start so our number of nodes with degree k at time $t+1$ is

   $$(n+1)p_k(t+1) = np_k(t) + p_{k-1}(t) \times m - p_k(t) \times m$$

   But we need a special case for $k = m$ because nodes start with degree m and therefore we don't have nodes with degree $k < m$. So

   $$(n+1)p_m(t+1) = np_m(t) + 1 - p_m(t) \times m$$

   And we added one to the right side (in red) because of the new node we add that start with degree m.

   Now our full set of recursive equations is:

   $$\begin{cases} k = m, & (n+1)p_m(t+1) = np_m(t) - p_m(t) \times m + 1 \\ else, & (n+1)p_k(t+1) = np_k(t) - p_k(t) \times m + p_{k-1}(t) \times m \end{cases}$$

   In large t this is just

   $$\begin{cases} k = m, & (n+1)p_m = np_m - p_m m + 1 \\ else, & (n+1)p_k = np_k - p_k m + p_{k-1} m \end{cases}$$

   The first equation for the special case $k = m$, Eq1, is

   $$\text{Eq1: } p_m = 1 - p_m m \Rightarrow (1+m)p_m = 1 \Rightarrow p_m = \frac{1}{(1+m)}$$

And the second equation for the general case, Eq2, is

$$\text{Eq2: } p_k = p_{k-1}m - p_k m \Rightarrow (1+m)p_k = p_{k-1}m \Rightarrow p_k = p_{k-1}\frac{m}{(1+m)}$$

So our two simplified recursions are:

$$\begin{cases} k = m, & p_m = \frac{1}{(1+m)} \\ else, & p_k = p_{k-1}\frac{m}{(1+m)} \end{cases}$$

2. We can solve these starting from Eq2 in the case of $k|k = m+1$, where $k-1 = m$ meaning we can use Eq1. We repeat this using $m+2$ now that we learn $m+1$ and continue until we find the pattern for a closed form solution

$$\begin{aligned} p_{m+1} &= p_m \frac{m}{(1+m)} && = \frac{1}{(1+m)} \times \frac{m}{(1+m)} \\ p_{m+2} &= p_{m+1} \frac{m}{(1+m)} && = \frac{1}{(1+m)} \times \frac{m}{(1+m)} \times \frac{m}{(1+m)} \\ p_{m+3} &= p_{m+2} \frac{m}{(1+m)} && = \frac{1}{(1+m)} \times \frac{m}{(1+m)} \times \frac{m}{(1+m)} \times \frac{m}{(1+m)} \end{aligned}$$

We see a pattern emerge for some $k = m + c$:

$$p_{m+c} = \frac{1}{(1+m)}\left(\frac{m}{(1+m)}\right)^c = \frac{1}{(1+m)}\left(\frac{m}{(1+m)}\right)^{k-m}$$

Manipulating this to separate k:

$$p_{m+c} = \frac{1}{(1+m)}\left(\frac{m}{(1+m)}\right)^{-m}\left(\frac{m}{(1+m)}\right)^k = A\left(\frac{m}{(1+m)}\right)^k$$

Where A is some numerical constant dependent on m which is a fixed feature in the model. Ignoring this constant A, we conclude that the degree of a node is proportional to the following:

$$p_k \propto \left(\frac{m}{(1+m)}\right)^k = C^k$$

Where C is some constant smaller than 1, indicating clear exponential decay.

3. We can now express this in the way of traditional exponential decay, with $\lambda = \ln(C^{-1})$, giving us

$$p_k = \exp(-\lambda k)$$

We already showed that $C = m/(1+m)$, and $\log(1/(1+m)) \approx 1/m$, so:

$$p_k = \exp\left(-\frac{k}{m}\right)$$

Very similar to what was shown in lecture. As discussed over email with Professor, we could make the constant A from part 2 above look something like e/m, especially in the limit of large m.

3 Simulating the B-A Model

We will now empirically verify some properties of the B-A model with computer simulations of the model. We start with Mathematica's `BarabasiAlbertGraphDistribution`, and for each graph considered, we average across 10 simulations to get more stable results.

1. We know that the degree distribution of the B-A model is given by

$$p_k = \frac{2m(m+1)}{k^3}$$

We notice that the degree is independent of the size of the model n, so we simulate a large model with 300,000 nodes (each averaged across 10 simulations to get stable results).

We vary $m \in \{2, 3, 5, 10, 50\}$ to verify empirically the ratio between distributions. It is expected, for example, that for a given degree the model with $m = 3$ will have twice the number of nodes as the model with $m = 2$. Since

$$\left(\frac{2 \times 3(3+1)}{k^3}\right) / \left(\frac{2 \times 2(2+1)}{k^3}\right) = \frac{4}{2} = 2$$

We divide the number of nodes for each degree from each pair and take the mean across every degree. For example for $m = 2$ vs $m = 3$ we calculate the degree distribution for each. Then we divide the number of nodes for each degree from $m = 3$ by the number of nodes from $m = 2$ and take the average. More formally:

$$\sum_{i \in k} \left(\frac{1}{[i \in k]}\right) \frac{p_i^{(m_1)}}{p_i^{(m_2)}}$$

Where $[\dots]$ is Iverson bracket notation, $[i \in k]$ is the count of different degrees, making this the average. (In reality I did this only for degrees 1 to 1000 but only after confirming experimentally that it gives the same results.) And $p_i^{(m_1)}$ is the probability of degree $k = i$ from the network with $m = m_1$ (I did it with the count of nodes instead of the probability but since they have the same number of nodes $n = 300{,}000$ this is equivalent).

The following tables will show the expected ratio and empirical ratio of one value m vs the others, for each value m (total of 5 choose 2 = 10 pairs).

$m = 2$ vs	$m = 3$	$m = 5$	$m = 10$	$m = 50$
expected	2	5	18.33	425
empirical	1.56	2.62	5.30	27.17

$m = 3$ vs	$m = 5$	$m = 10$	$m = 50$
expected	2.5	9.17	212.5
empirical	1.74	3.51	17.97

$m = 5$ vs	$m = 10$	$m = 50$
expected	3.67	85
empirical	2.06	10.57

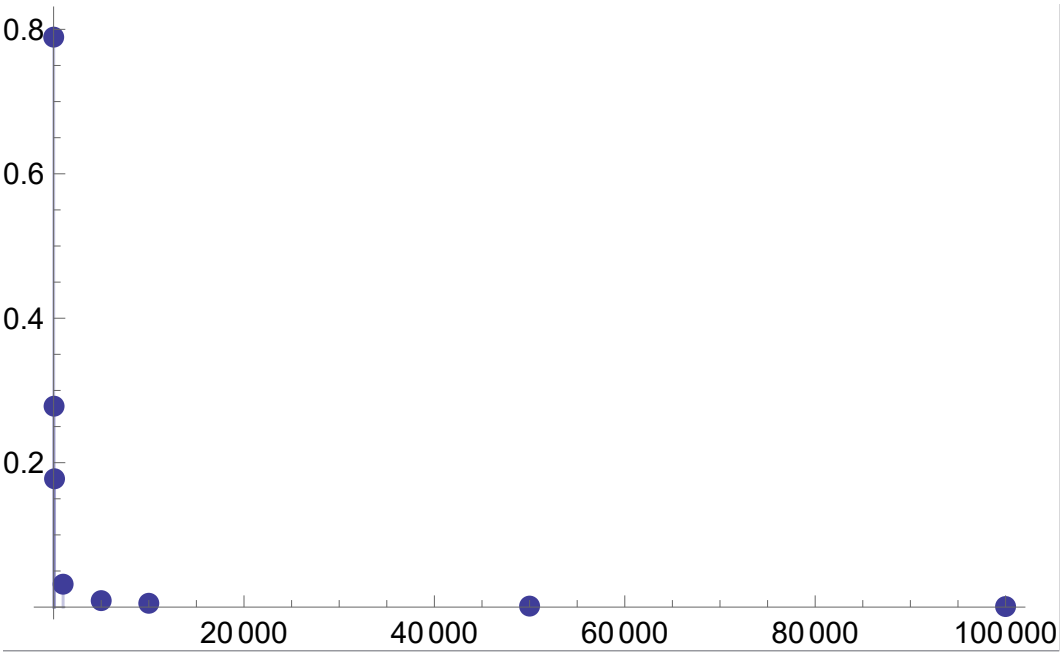
$m = 10$ vs	$m = 50$
expected	23.18
empirical	5.17

We see that the empirical ratio is always lower than that given by the theoretical formula. This is especially pronounced the larger the theoretical ratio should be, whereas the numbers are close for smaller ratios. I'm not exactly sure why this is the case – it could have something to do with how the model is simulated in Mathematica or it could be a

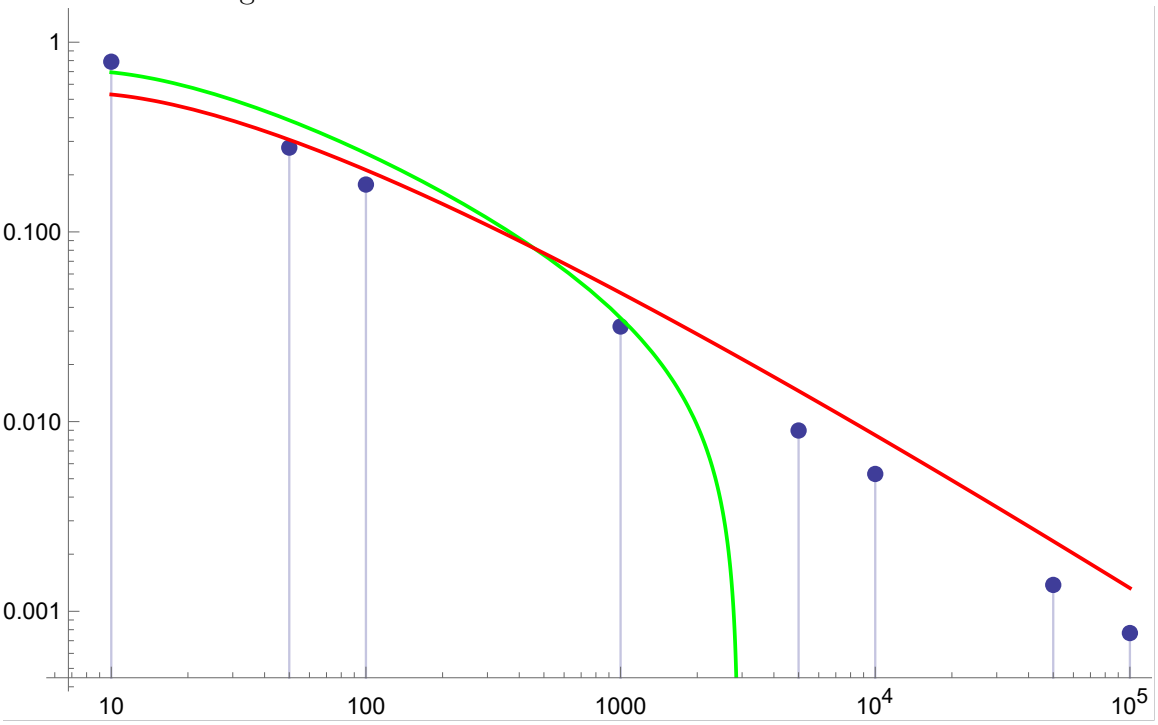
2. The clustering coefficient for the B-A model is given as $C = \ln(n)^2/n$. This is slower than the Erdős-Rényi model which goes to zero as $1/n$. We compare the clustering coefficient using Mathematica's `GlobalClusteringCoefficient` for models of $m = 5$ and various network sizes with $n \in \{10, 50, 100, 1{,}000, 5{,}000, 10{,}000, 50{,}000, 100{,}000\}$.

$\boldsymbol{n}$	10	50	100	1000	5000	10000	50000	100000
$\boldsymbol{C}$	0.79	0.28	0.18	0.03	0.009	0.0053	0.0014	0.0008

We can graph this:



Looks roughly logarithmic, but it's hard to tell. Let's graph the loglog plot instead and include the theoretical formula for C in red and just for fun the line of best fit from our data in green.



We see that the data is a straight line which means it's a log relationship, and the theoretical formula represented by the red line is a good fit. (Whereas the line of best fit is a little quirky after 10^3).

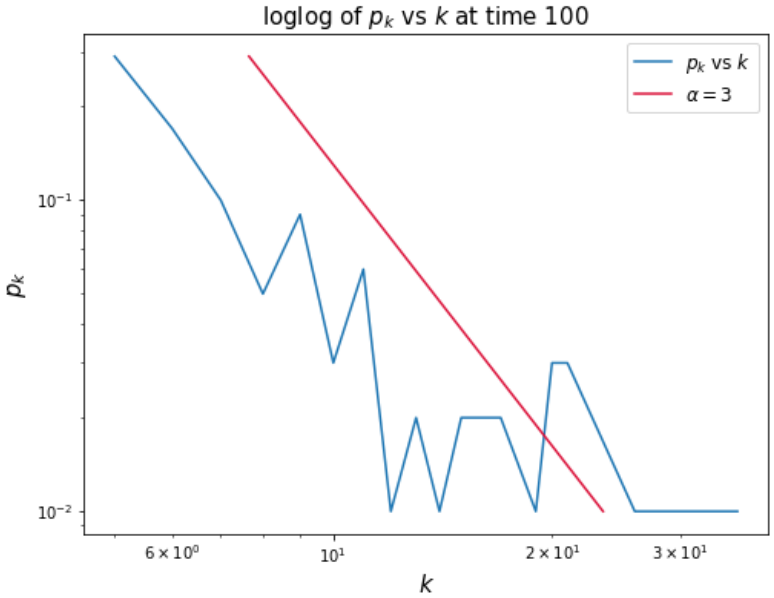
For the next 3 questions we need to manually simulate the network in order to track its growth and the degree dynamics of its nodes over time.

We begin with the discussion in lecture regarding simulating the Price model, but make a few key changes. To recap: we can efficiently simulate the degree distribution of the Price model by keeping an unordered list of every node that has been "targeted" (effectively a list to calculate the indegree q of every node). At each new timestep when we add a new node, we connect either to an element from the list with probability α or a node at random with probability $(1 - \alpha)$ and add it to the list. This works because now we are selecting nodes with some chance proportional to their indegree, and also giving a chance for new nodes (that had indegree of zero) to join the list.

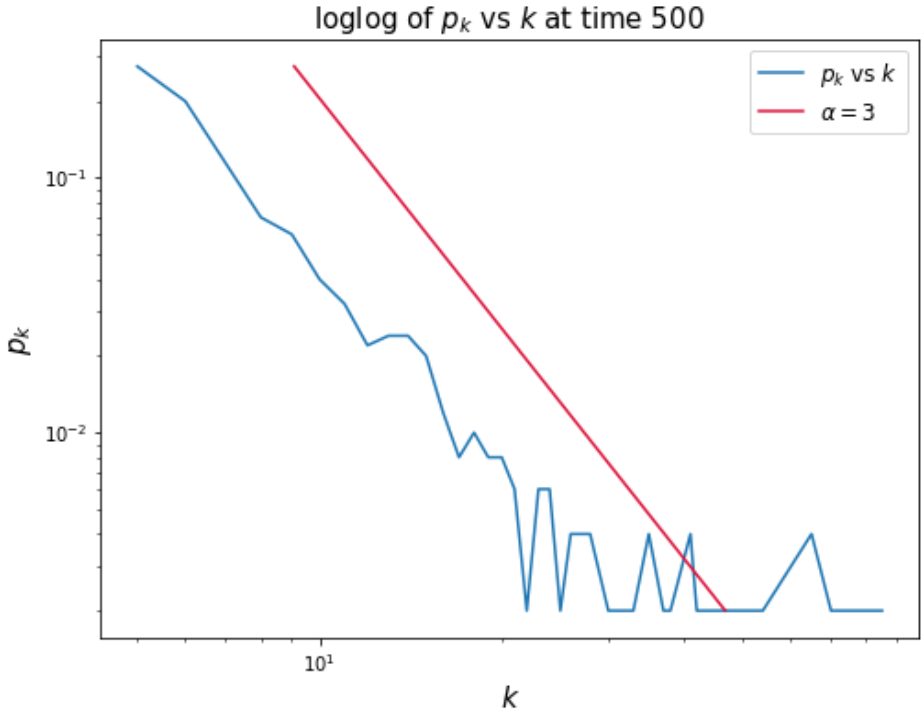
The difference between this and a B-A simulation is first of all that $\alpha = 1$ meaning we are always connecting to nodes based on their degree. The second difference is that we link based on the degree $(q+m)$ as opposed to just the indegree. The actual chance that we have to do for this simulation is surprisingly small. The only difference that makes this work is that when we add a new node at each timestep, we add it to the list m times in addition to adding our m "targets" that we link to. In this way the list given us k as opposed to q and everything can be counted from here. We start with a small arbitrary number of nodes connected to each other to seed the network, and add a node at each time step t, up to 10,000 nodes for our example.

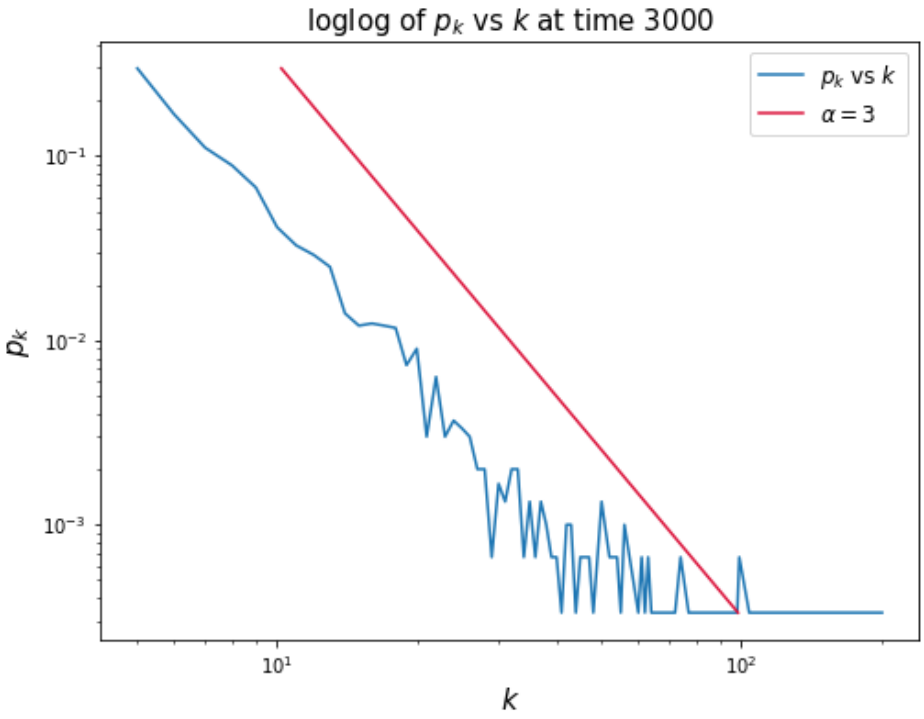
Code for this simulation can be found here[1].

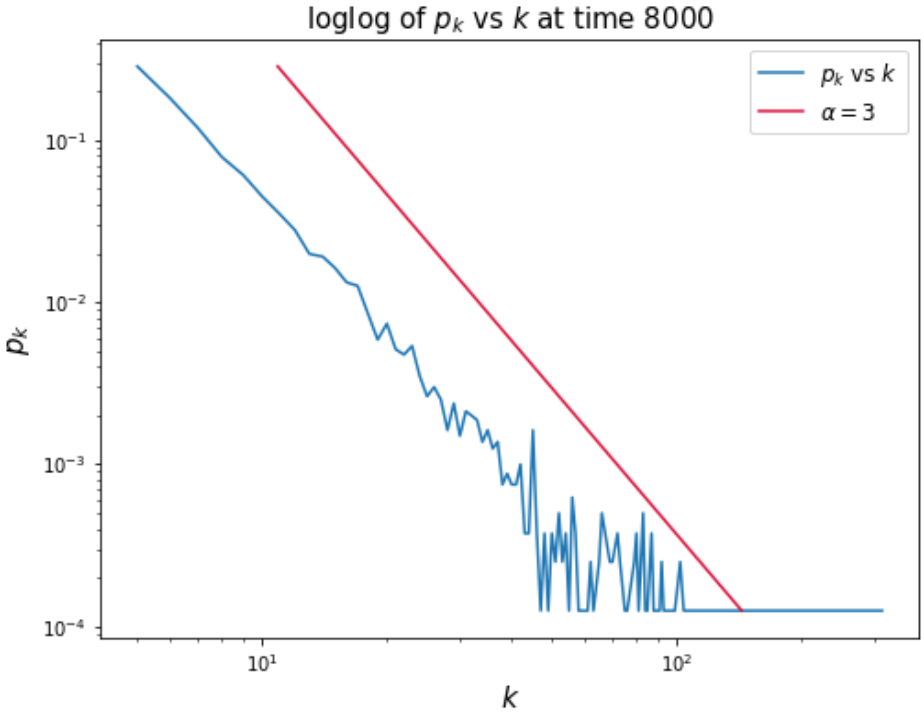
3. We want to see how our network's degree distribution scales over time. To do this we save the degree distributions for the simulation after $t \in \{100, 500, 3{,}000, 8{,}000, 10{,}000\}$ nodes added. We plot these counts p_k vs k as well as a line with slope $-\alpha = -3$ (roughly overtop the simulation data, but it was hard to get them to overlap well) in a loglog plot for each time t from above.

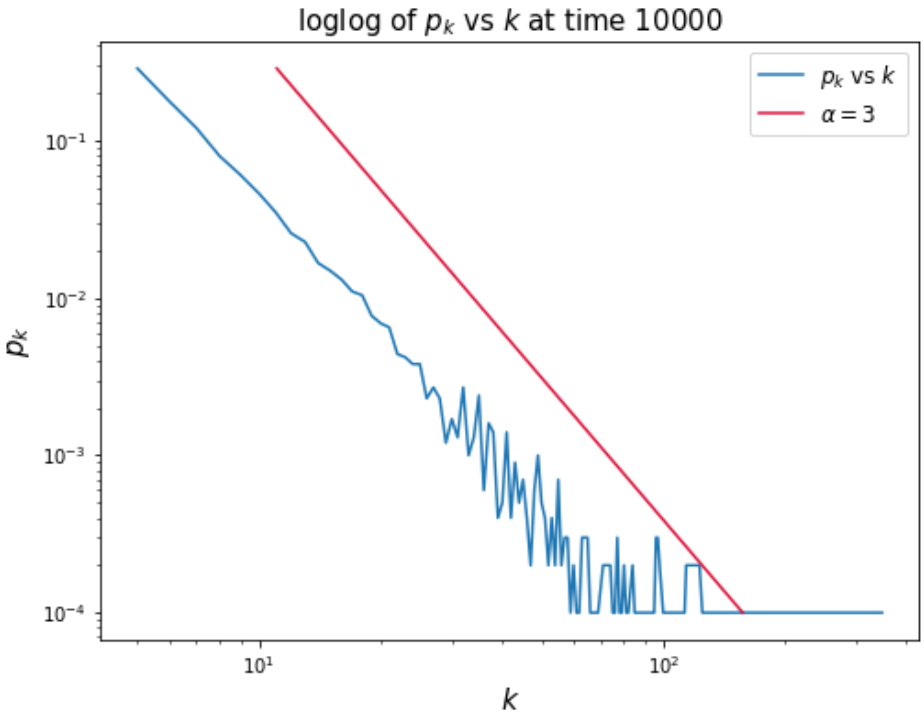


[1] https://colab.research.google.com/drive/1R7ZPil2-LNZgcZFK5WEvau7awWzxeamh?usp=sharing


loglog of p_k vs k at time 500
p_k vs k
$\alpha = 3$
10^{-1}
10^{-2}
p_k
10^{1}
k


loglog of p_k vs k at time 3000
p_k vs k
$\alpha = 3$
10^{-1}
10^{-2}
10^{-3}
p_k
10^{1}
10^{2}
k



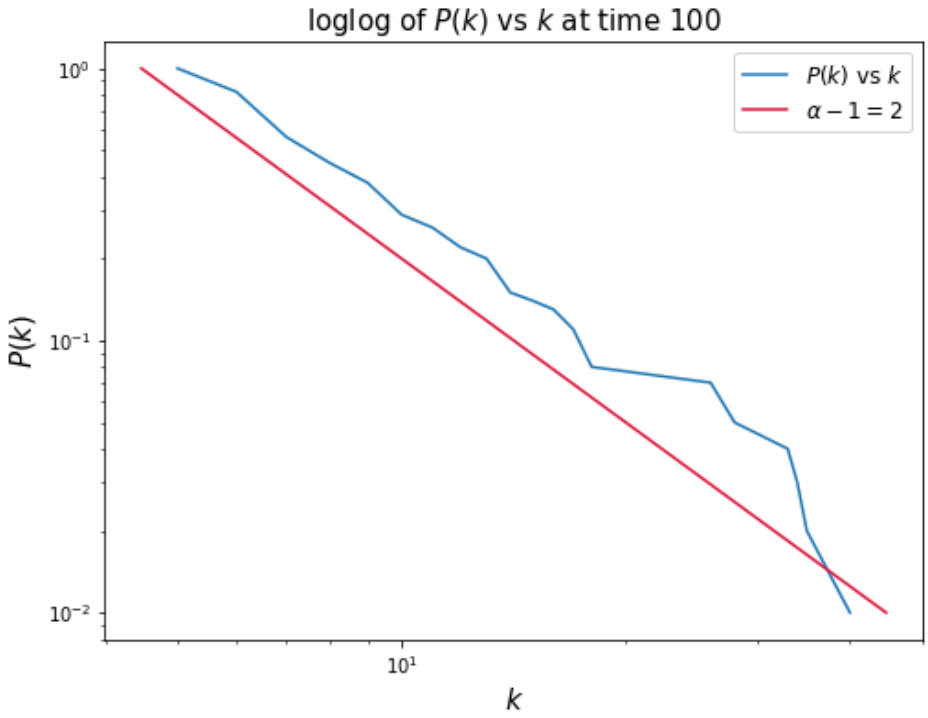


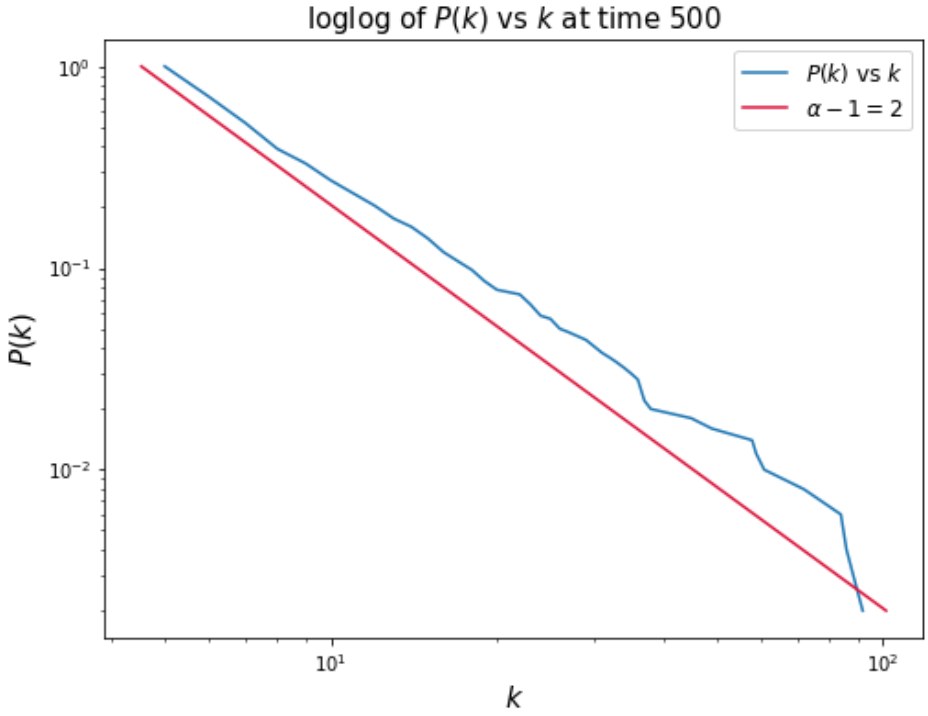
We see that the simulation in blue looks very close to the theoretical slope in red, despite it being very noisy (especially at high degrees).

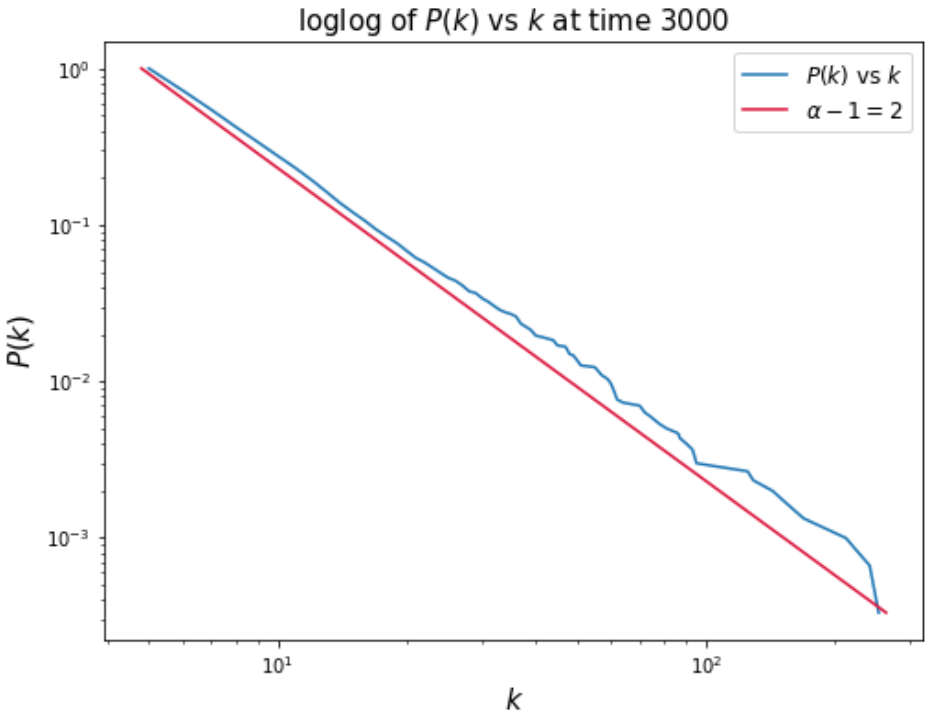
4. Now we do the same, except with the cumulative distribution $P(k)$ instead of p_k, where

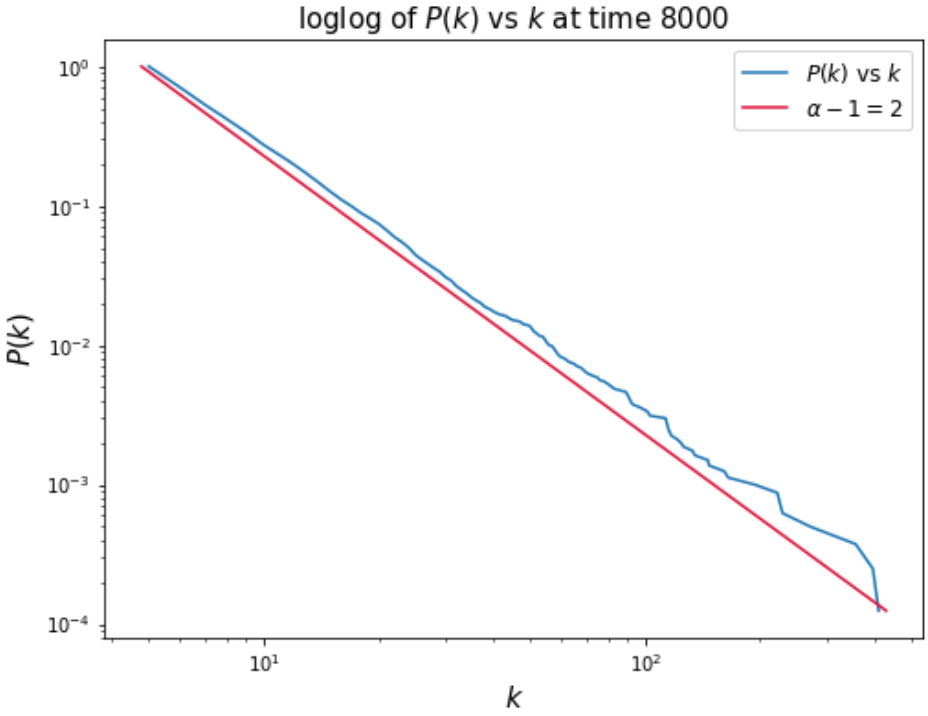
$$P(k) = \sum_{k'}^{\infty} p_{k'}$$

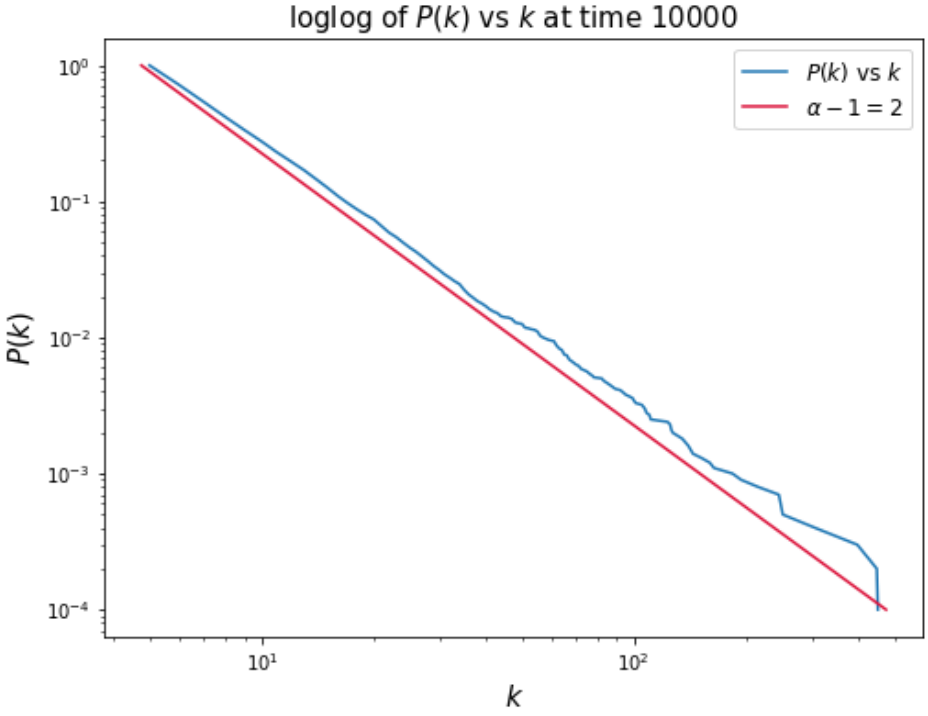
In our data, we take the count of each distribution and add the sum of all the greater distributions. We again plot this in a loglog plot and also plot $-(\alpha - 1) = -2$ for the same times t from above.




loglog of $P(k)$ vs k at time 500
$P(k)$ vs k
$\alpha - 1 = 2$
10^0
10^{-1}
10^{-2}
$P(k)$
10^1
10^2
k


loglog of $P(k)$ vs k at time 3000
$P(k)$ vs k
$\alpha - 1 = 2$
10^0
10^{-1}
10^{-2}
10^{-3}
$P(k)$
10^1
10^2
k





The theoretical line in red is a really good fit to the data in blue, especially as the network acquires more nodes.

5. Now, using the same algorithm model but averaging across 1000 simulations, we will follow some individual nodes that we introduced at times $t_i \in \{10, 100, 500, 1{,}000, 3{,}000\}$ and see how their degree changes at times $t \in \{100, 500, 3{,}000, 8{,}000, 10{,}000\}$. The theoretical behavior should be a square root as a function of time, or

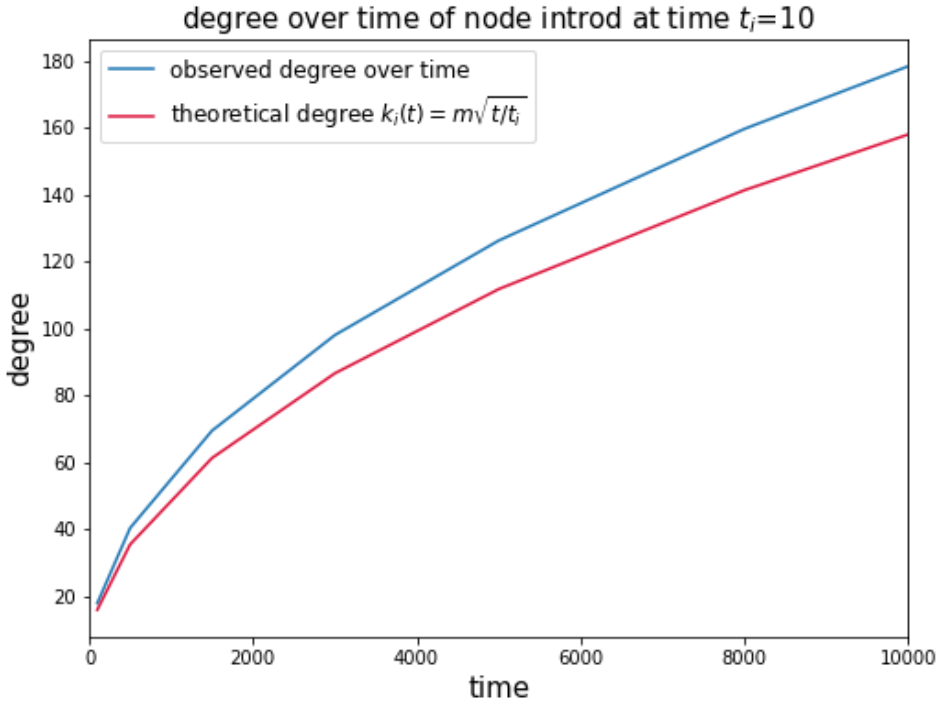
$$k_i(t) = m\sqrt{\frac{t}{t_i}}$$

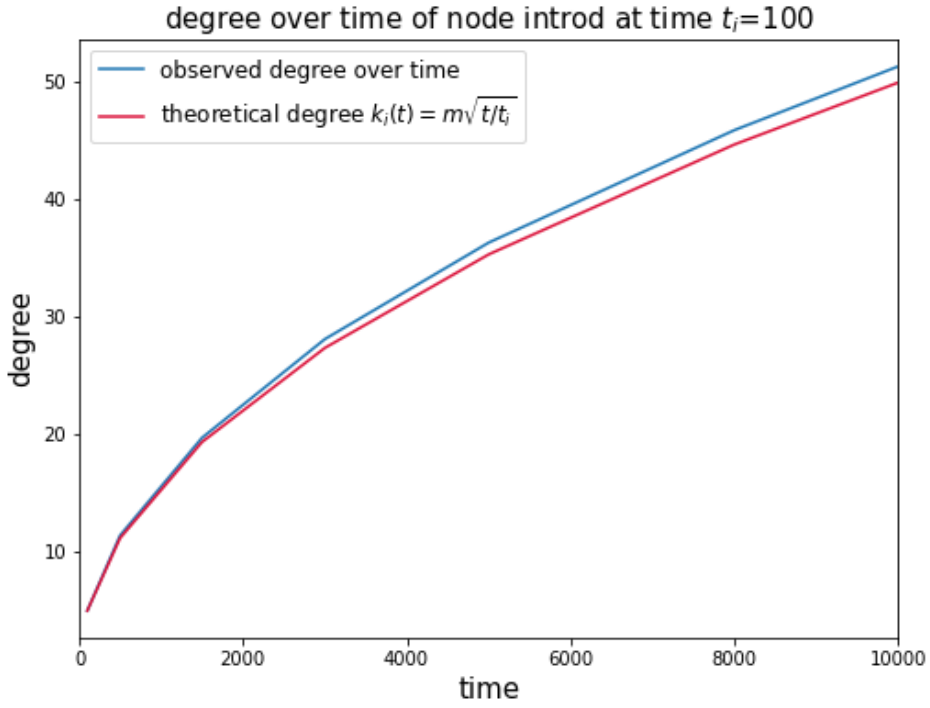
The full output and error for each of these nodes at each of these times can be found in the code. We took the absolute percent error between each node's degree distribution at each time and their theoretical prediction and then took the mean of that (or MAPE, Mean Absolute Percent Error) across each time in the network, and then took the mean of the MAPE across each node t_i to get the MMAPE.

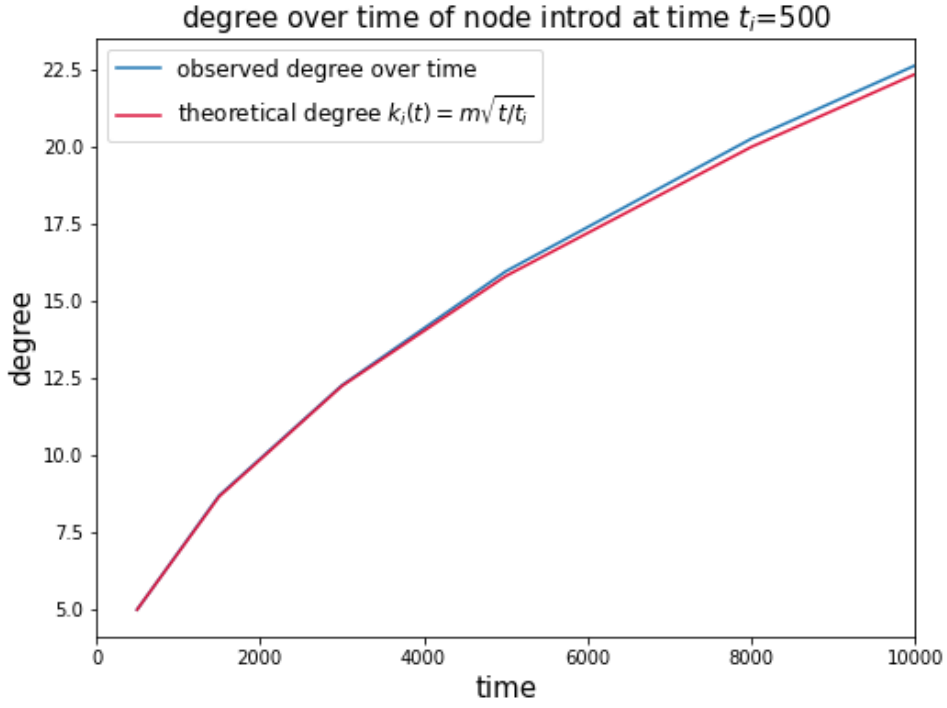
With our average across 1,000 simulations we get an MMAPE of 0.036 (about 4%), with the highest MAPE of 0.126 (about 13%) for node $t_i = 10$ and the smallest MAPE is 0.006 (less than 1%) for node $t_i = 3{,}000$. It had a median MAPE of 0.017 (about 2%) so it seems like node $t_i = 10$ is anomalously high.

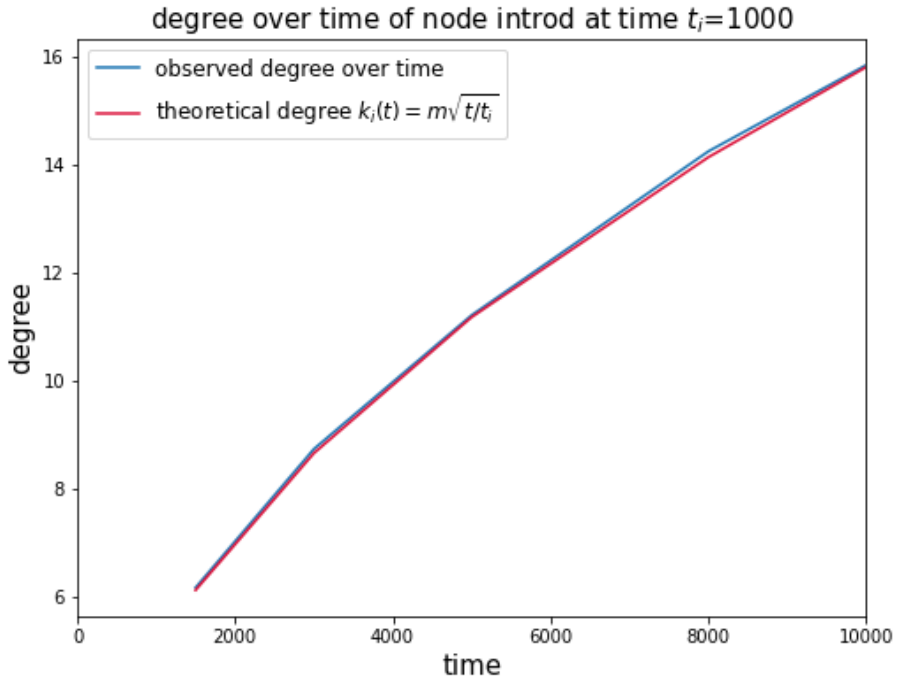
Out of curiosity, our non average simulation just using one run had a MMAPE of about 11%, a median MAPE of also about 11%, highest MAPE of about 17% for node $t_i = 10$, and a lowest MAPE of about 3% for node $t_i = 3{,}000$ (and node $t_i = 1{,}000$ was anomalously high at about 11% MAPE).

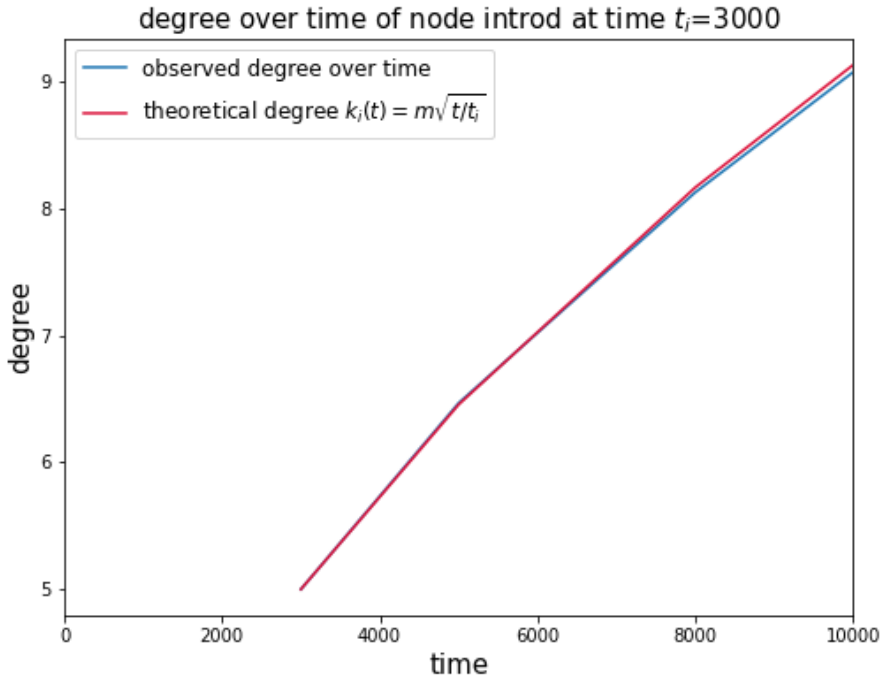
Now we plot the degree evolution over time for each node t_i and their theoretical degree over time function.




degree over time of node introd at time t_i=100
observed degree over time
theoretical degree $k_i(t) = m\sqrt{t/t_i}$
degree
time
10
20
30
40
50
0
2000
4000
6000
8000
10000

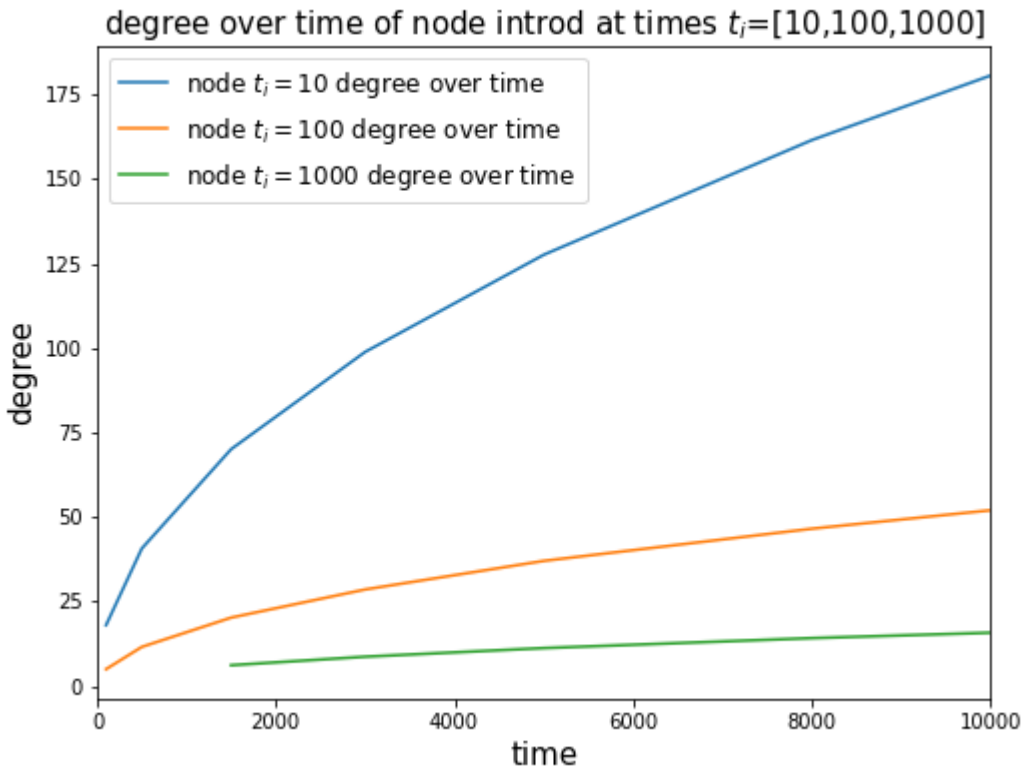

degree over time of node introd at time t_i=500
observed degree over time
theoretical degree $k_i(t) = m\sqrt{t/t_i}$
degree
time
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
0
2000
4000
6000
8000
10000





The observed empirical data from our 1,000 simulations in blue is incredibly close to the theoretical prediction in red. It is especially good the later the node was added. It seems like node $t_i = 10$ overestimates a lot, possibly because of some assumption in the simulation.

Now we will compare nodes from $t_i \in \{10, 100, 1{,}000\}$ and graph them.



We can clearly see the early mover advantage here. Even the difference between the node from $t_i = 10$ and $t_i = 100$ is huge, there's no way it will every come close, let alone the node from time $t_i = 1{,}000$. The early mover advantage has an incredibly strong effect.