Homework 2: Small World Models

Ben Zuckier

Network Science II

1 Losing Nodes

1. Getting disconnected:

   For a node to be disconnected in the "original" Watts-Strogatz model we need a node to not gain any shortcuts and to lose all of its original connections. Let p be the probability of having a connection replaced with a shortcut. Let c be the number of neighbor nodes that each node is connected to (and therefore the starting degree of every node).

   We said that the chance of gaining m shortcuts is $p_m = \frac{(pc)^m}{m!}\exp(-pc)$ so the chance of gaining 0 shortcuts is

   $$p_m = \frac{(pc)^m}{m!}\exp(-pc) \Rightarrow p_{m=0} = \frac{(pc)^0}{0!}\exp(-pc) = \exp(-pc)$$

   A node can lose a connection and not gain a shortcut if one of its neighbors has their connecting edge "switched". Now let p_r be the chance of a node losing r edges. From the basic product rule of probability (each time with probability p) we see that

   $$p_r = \underbrace{p \times p \times \cdots \times p}_{r \text{ times}} = p^r$$

   So the probability of losing all c starting edges would be

   $$p_{r=c} = p^c$$

   Now we assume that losing edges and gaining edges is independent and we use the product rule again to find that the probability of a node ending up with 0 edges is

   $$p_{k=0} = p_{r=c} \times p_{m=0} = p^c e^{-pc} = \left(pe^{-p}\right)^c$$

   □

2. Likelihood with six neighbors and one million nodes:

   Considering a network with one million nodes, each connected to their 6 neighbors ($c = 6$). We want to find the maximum likely number of nodes that could get disconnected in the worst case scenario.

   Let $n = 1 \times 10^6$ be the number of nodes in the network.
   Let N be the number of nodes that get disconnected.
   We determined that the probability of a node getting disconnected is $p_{k=0} = (pe^{-p})^6$.

$$\Rightarrow N = np_{k=0} = 1 \times 10^6 \left(pe^{-p}\right)^6$$

   Now we find probability p s.t. we maximize N for $0 \leq p \leq 1$.

$$\Rightarrow p = 1,\ N_{\max} = 2478.752$$

   This is about 0.25% of the network ($\frac{1}{4}$ of one percent).

3. Realistic disconnection?

   In a real network, p is much smaller. Let $p = 0.01$. Now with $c = 6$ how big does the network have to be before we expect a node to get disconnected?

   We set 1 equal to np_k and solve for n.

$$1 = np_k = n\left(pe^{-p}\right)^c = n\left(0.01e^{-0.01}\right)^6 \Rightarrow\ n = 1.062 \times 10^{12}$$

   So we need over a trillion nodes before this becomes a problem. We also see directly from the probability that this has a 9.42×10^{-13} chance of happening. Very unlikely.

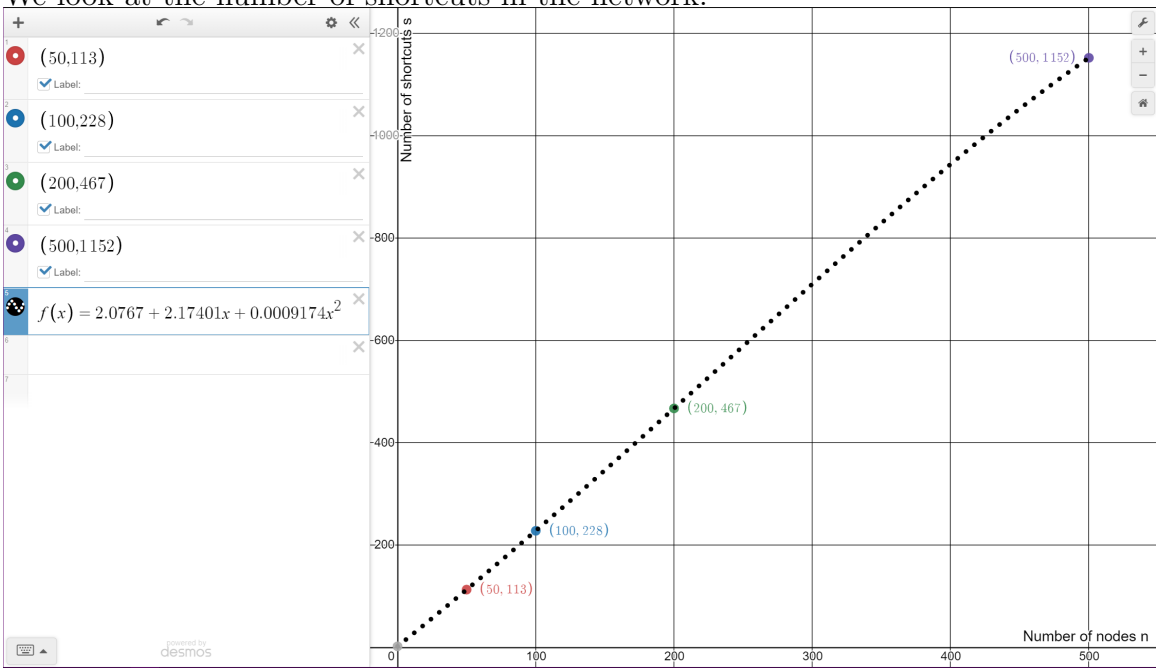
2 Watts-Strogatz Simulation

I worked with Benjie to program the simulation.

1. Varying n:

   We start with a network with $c = 6$, $p = 0.3$, and vary $n \in \{50, 100, 200, 500\}$.

   We look at the number of shortcuts in the network.

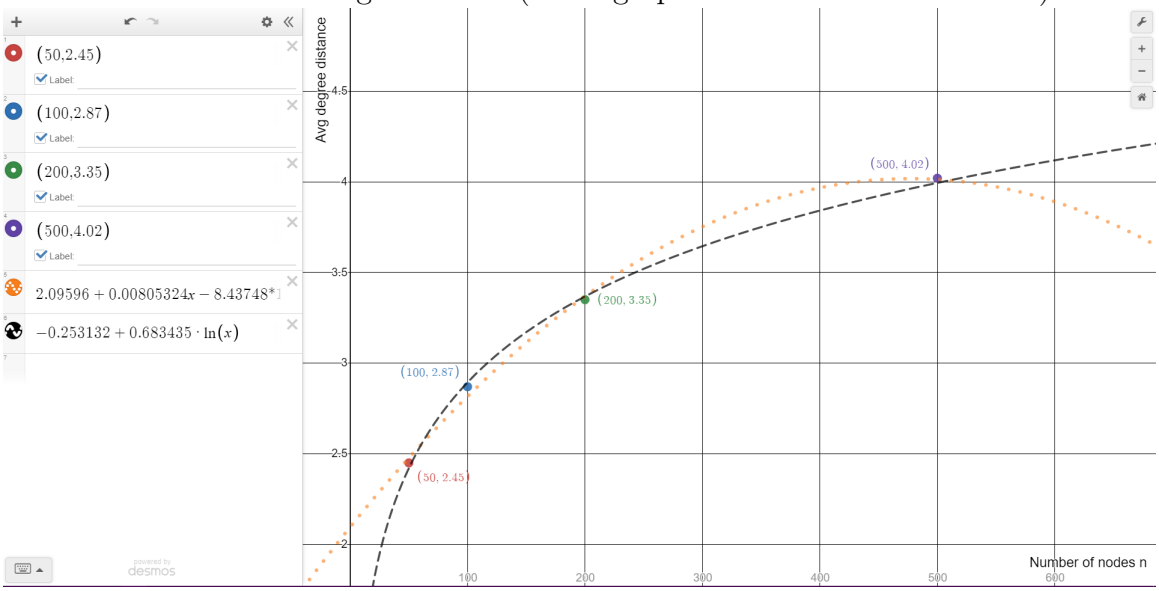


   We can see that the number of shortcuts s increases linearly in the overall number of nodes n. Our average of $\frac{s}{n}$ for these four examples is about 2.3. This is exactly what I expected. We can calculate this directly using:

$$s = \frac{1}{2}ncp \Rightarrow \frac{n}{s} = \frac{2}{cp} = \frac{2}{0.3 \times 3} = 2.22$$

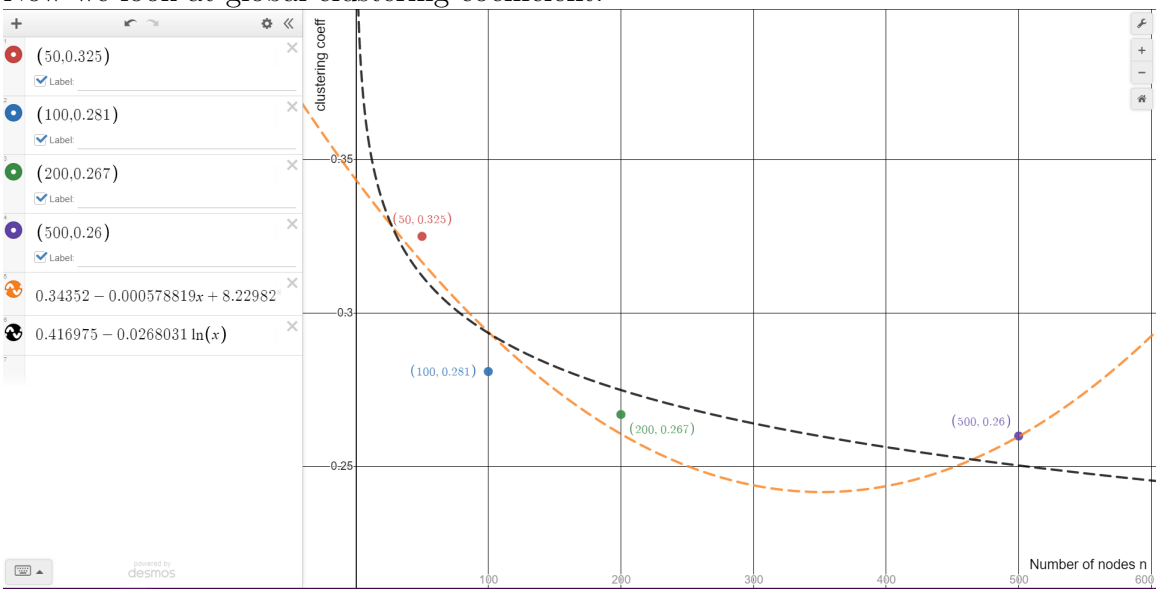
   (I just noticed that something must be wrong here because this is $\frac{n}{s}$ not $\frac{s}{n}$ and I used c as Mathematica's k when it should really be $k/2$. Still strange that I got the right answer.... Regardless, the ratio of shortcuts to nodes should be constant.)

Next we look at the average distance (mean graph distance in Mathematica)



We see here that a really good fit for the graph is not linear or polynomial, (especially the "bad" orange graph after $n = 500$) but actually logarithmic. This makes sense as we saw from lecture that the small world network average distance grows proportional to $\log(n)$ (or $\log(\log(n))$ in super small world). Intuitively, there is an increase in the number of shortcuts as n increases but it still takes some amount of work proportional to $\log(n)$ to traverse the shortcuts to get from one node to another.

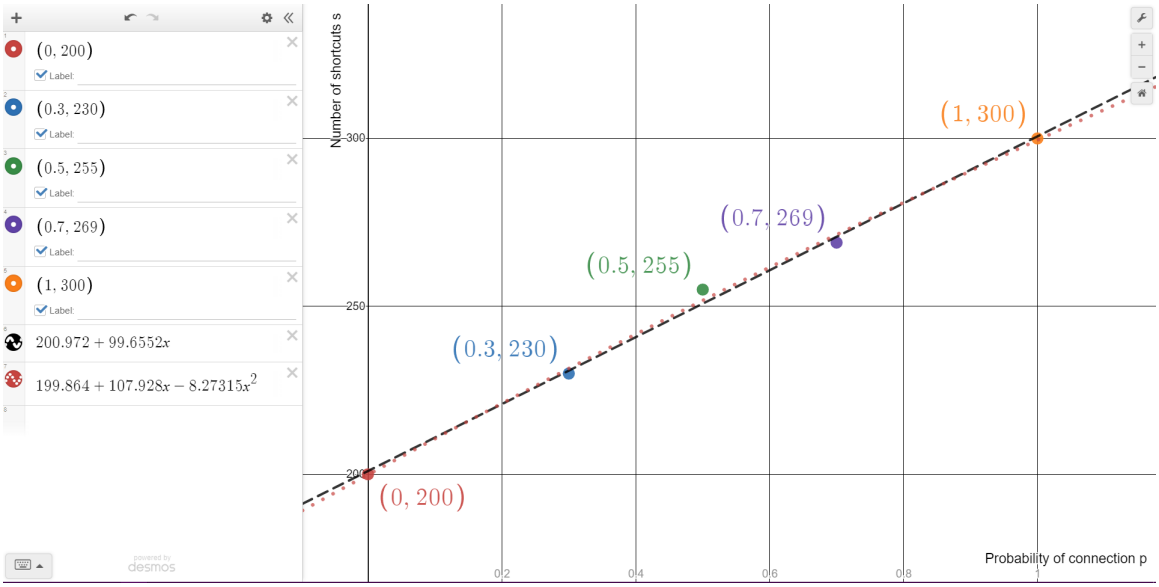
Now we look at global clustering coefficient.



Again, we see a change proportional to $\log(x)$, but this time a decrease (and fairly small). From lecture I would think that it should have no effect but maybe it has something to do with the inherent clustering of the initial condition of being connected to c neighbors (but that is dwarfed by larger networks which is why the value stabilizes).

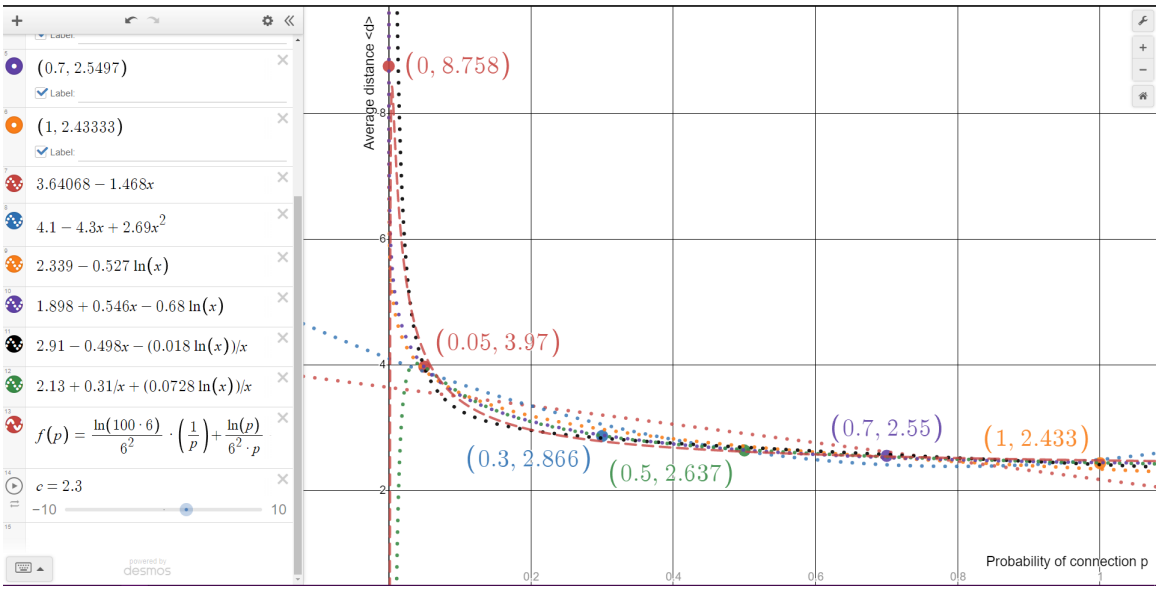
2. Now $n = 100$, $c = 6$, and vary $p \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$:

   Number of shortcuts:



We see linear increase here which makes sense. We know from above that $s = \frac{1}{2}ncp$ so as p increases, s increases.

I started looking at the average distance in the network and quickly realized that with $p = 0$ we wouldn't have any small world effects. We explained in lecture that $< d > \approx \dfrac{\ln(ncp)}{c^2 p}$ and that small world effects only kick in with $p \gg \dfrac{1}{nc}$ which for us means that $p \gg 0.0017$. Certainly this would not be satisfied for $p = 0$. As such, I added results for $p = 0.05$ and didn't include $p = 0$ in my fit analysis (since $\ln(0)$ is undefined).

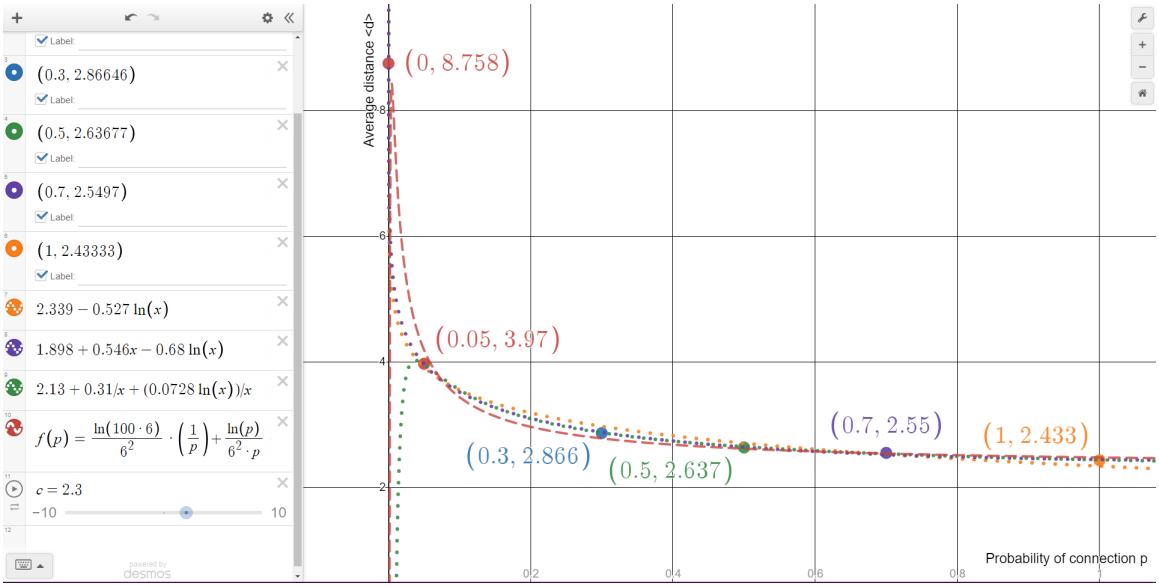


There is a lot going on here. I have a number of best fit plots shown for different functions of p. The adjusted R-Squared value for these as reported by Mathematica are: 0.646, 0.919, 0.977, 0.998, 0.994, 0.9995 for the red, blue, orange, purple, black, and green fit functions respectively (not the red $f(x)$, that's explained in a moment).

I was expecting the model to be fit best by some function of $\left\{\ln(p), \frac{1}{p}\right\}$ or $\left\{\frac{\ln(p)}{p}, \frac{1}{p}\right\}$ since the most we can separate p in the above equation is as

$$< d >= \left(\frac{\ln(p)}{p}\right)\frac{1}{c^2} + \left(\frac{1}{p}\right)\frac{\ln(nc)}{c^2}$$

This last equation is displayed as $f(x)$ in the graph (plus some constant because I the simulation is off I think because of both randomness and since mathematica doesn't assign nodes shortcuts back to themselves).
Now let's look at just the functions that have some $\ln(p)$ factor:



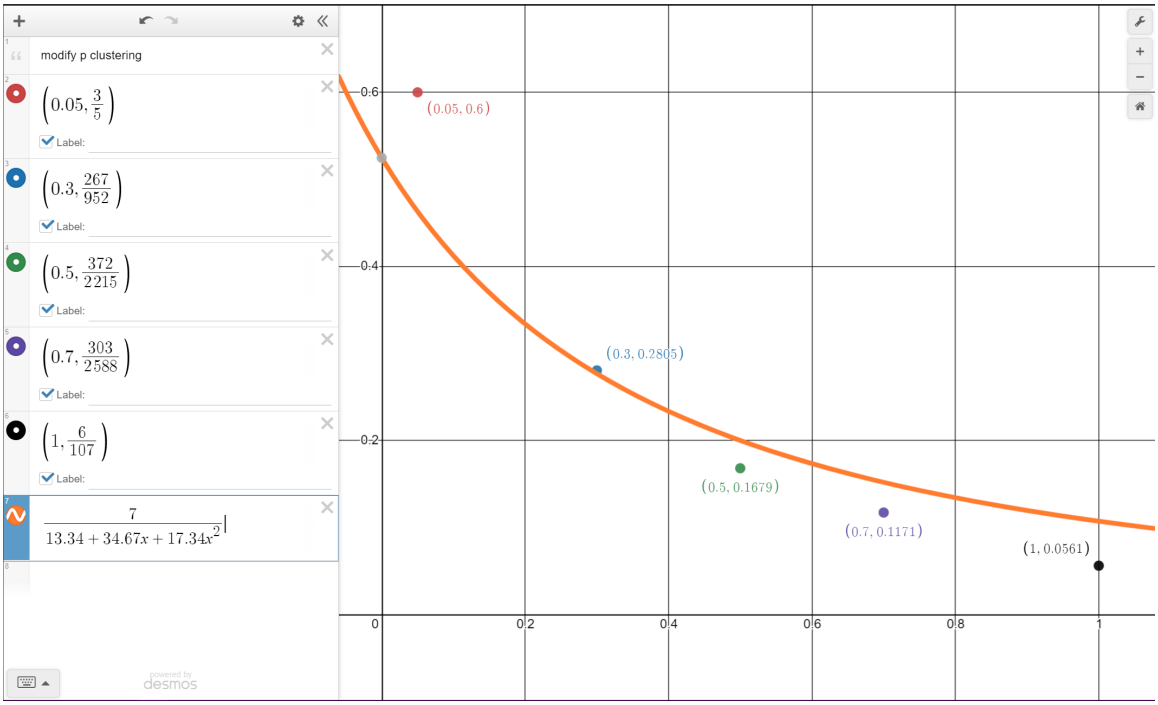
We see that the green function that has the adjusted R-Squared of 0.9995 is almost exactly the same as the theoretical best $f(p)$ so this graph is good.

The green graph falls apart for values p smaller than 0.05 and doesn't extend to the $p = 0$ since we don't have data from the model about that because the small world effects do not apply unless $p \gg \frac{1}{nc}$. But notice that the theoretical best fit $f(p)$ also starts from $-\infty$ and has the peak near the first value, like the green graph (our best "best fit").

Now for global clustering I was already careful going in due to our last adventure with p average distance. I remember for lecture that there was a nonlinearly separable formula for clustering, given by

$$C = \frac{3(c-2)}{4(c-1)+8cp+4cp^2}$$

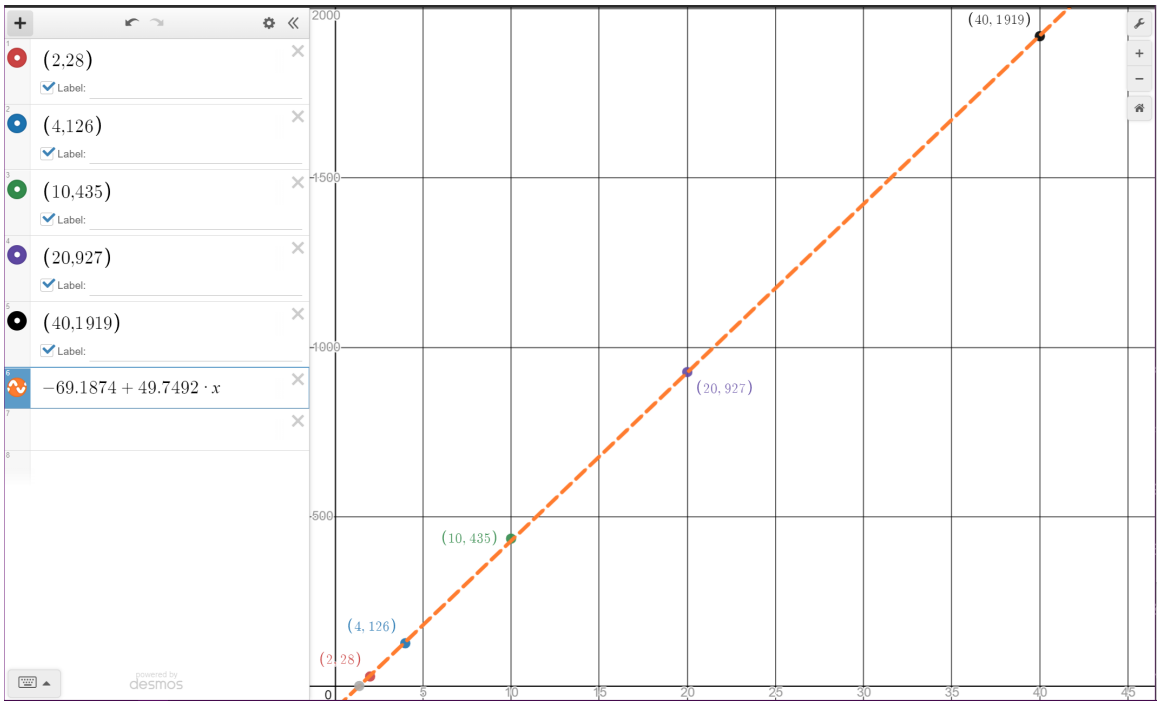
So I used Mathematica's NonlinearModelFit to find best fit here.



The R-Squared for the orange best fit is 0.981 which seems good without overfitting too much (plus it seems like if not for the first value it would do even better).

3. Now $n = 100$, $p = 0.3$, and vary $c \in \{2, 4, 10, 20, 40\}$ or what Mathematica would call $k = c/2 \in \{1, 2, 5, 10, 20\}$:

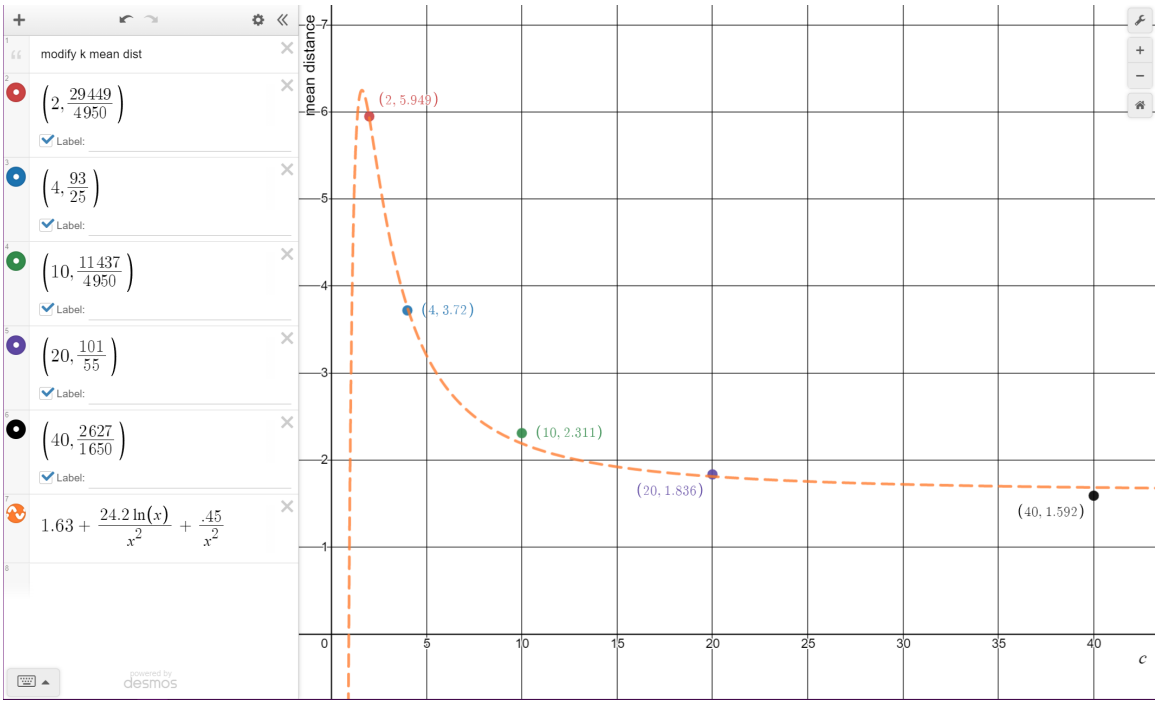
   Number of shortcuts:



   Again the same linear increase we've been seeing. We know $s = \frac{1}{2}ncp$ so as c increases, s increases.

Using the same logic as part 2 above for average distance, we are smarter and are expecting more complicated best-fit behavior due to our theoretical formula $< d > \approx \frac{\ln(ncp)}{c^2 p}$. Separating c as much as possible this time we get

$$< d >= \left(\frac{\ln(c)}{c^2}\right)\frac{1}{p} + \left(\frac{1}{c^2}\right)\frac{\ln(np)}{p}$$

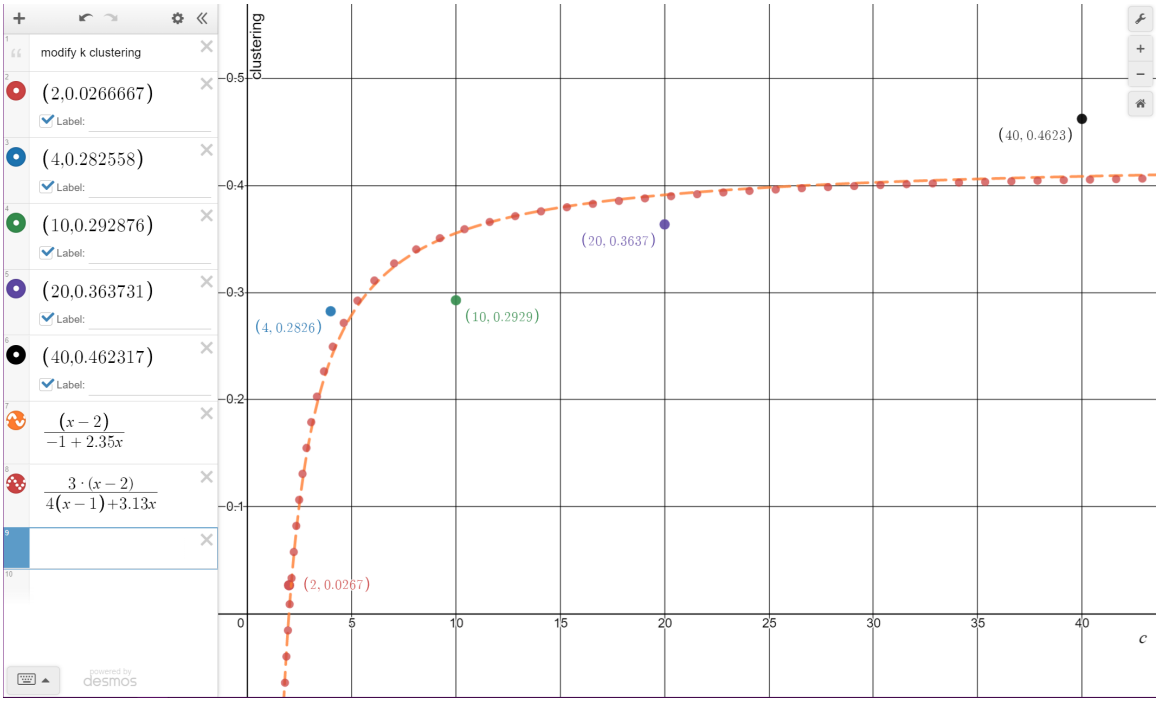


My prediction, the orange line, is incredibly good. The adjusted R-Squared is 0.996 (actually might be a bit overfit). Knowing the "form" of the function ahead of time is very helpful.

Now again being cautious, we have learned our lesson from the clustering example from 2. We know that

$$C = \frac{3(c-2)}{4(c-1)+8cp+4cp^2} = \frac{3(c-2)}{4(c-1)+c(8p+4p^2)}$$

where we separate c as much as possible here.



We still see a lot of the same bouncy artifacts but overall we have a surprisingly good fit. The orange line of fit isn't including the constant multipliers and the red one is. In the orange case we have an adjusted R-Squared of 0.97496 and the red is 0.975. Incredibly close. And both respectable.

3 Ultra-Ultra Small Network

Now we consider a modification of the Watts-Strogatz model in which two connected "hubs" are added to the center of the network. When a shortcut is added, it is added to one of the hubs. In this model, the average distance between nodes is independent of the network size n for any p and c.

The intuition for this is that instead of needing to traverse multiple shortcuts from the start node before getting to the destination node like in regular Watts-Strogatz, once we get to a shortcut (from both ends, the start and destination) we meet at one of the hubs and are "done".

Slightly more formally, we said in lecture that the intuitive understanding for the Watts-Strogatz was that we have n^* steps of $\ln\left(\frac{n}{n^*}\right)$ work each time. As just explained we don't have this "extra" $\ln\left(\frac{n}{n^*}\right)$ work in this case. So what is n^*? That is nothing but the average distance between ends of shortcuts. In that model we had $2s$ ends of shortcuts where $s = \frac{1}{2}ncp$ is the number of shortcuts. However, in our model we only have s ends of shortcuts since one end always maps to a hub. So the typical distance between ends of shortcuts for us is

$$\xi = n^* = \frac{n}{s} = \frac{2\cancel{n}}{\cancel{n}cp} = \frac{2}{cp}$$

As we can see, this value is independent of n, the size of the network, and only is a function of c and p.
The average length in the network is some function of this: $l = F(\xi) = \mathcal{F}(c, p)$.

At each end of our "journey" we need to get to a shortcut. That is to say, that from the source we need to a shortcut to a hub, and from the destination we need to get to a shortcut to get to the hub to meet up. We can expect to go $\frac{\xi}{2}$ at each end, or ξ in total. But we also can take strides of up to length $\frac{c}{2}$. So far we have $l = F \circ \left(\frac{4}{c^2 p}\right)$.

Once we get to a shortcut on each end, we need to take the shortcut to the hub, adding a distance of 2 total. Then, we are either at the same hub, or need to move to the other hub to "meet up". This adds 0.5 on average. Therefore our final estimation for path length (average distance between nodes) is:

$$< d >= l \approx \frac{4}{c^2 p} + 2.5$$

Once again, we see this is not dependent on the size of the network n, only on c and p. □

Note: I was thinking that in reality it would be somewhat connected to the size of the network because we could get from source to destination "directly" not using shortcuts if it was within $\frac{c}{2}$ nodes away and the proportion of nodes that are within this distance is dependent on n.