CS 224W PROJECT FINAL REPORT

MARK HOLMSTROM

1. Abstract

Many networks are worked with and viewed as static objects while this often isn't the case. Oftentimes, they change over time as new nodes are added and connections are made. One such class of networks that evolve over time are citation networks, networks that show how different authors or papers reference each other. As new papers are published and added to the network, the structure and importance of each node shifts. In this paper, we will study network evolution on citation networks and how the relative importance (represented by centrality) of a paper develops over time. The goal of the project is to be able to see how the centrality of a paper in a citation network develops after it is published and make an attempt at being able to predict this change over time in centrality.

2. Introduction

If we think about how important an academic work is, one way to measure this importance is to consider how many other works cite this one as source. In terms of a citation network, this is essentially the node's centrality since that measures how connected it is to other nodes. If we view a work's importance as equivalent to its centrality in a citation network, then the goal of this project is to create and assess a method for predicting the future centrality of a paper, given some information about its history in the citation network. This is an interesting problem in that it hasn't been looked at too much before.

Part of what makes this interesting is that as far as I can tell, there hasn't been too much work done in this field before. There has been some research done on network evolution, but most of that had to do with generating random graphs of increasing size or studying sentiment and behavior propogation through social networks. This project is different in that it is studying how a real-world network grows through the addition of new nodes and it is specifically focusing on looking at how node centrality in such a network develops.

3. Previous Work

As mentioned above, there has been some previous work done around this idea of graph growth, but it is limited. Most work about graph growth such as Erdos and Renyi's "On the Evolution of Random Graphs" discuss random graphs and how they evolve by either generating random graphs of increasing size or generating a graph randomly using certain rules. While this idea falls along similar lines, the purpose of this project is a bit different. Instead of discussing graph evolution in general, we look at an example of graph evolution,

build a model based on that example, and then apply our model to another network of very similar nature.
There has been work along this example-based approach before, but as far as I can tell them focused on very different aspects of the network than node centrality. Leskovec, Kleinberg and Faloutsos' "Graph Evolution: Densification and Shrinking Diameters" discusses evolution in a number of different real-world graphs looking specifically at the graph diameter and then proposes various models for how this evolution may be occurring. Kossinets and Watts' "Empirical Analysis of An Evolving Social Network" look at different high-level properties of a social graph and how they develop over time. That's really the key difference between previous works and this project. Those works focus on a big-picture model for how the graph as a whole is changing while this project focuses on how in specific cases, the centrality of nodes in a citation graph develop and looks for overarching trends in these developments.

4. Model/Algorithm Details

I developed a structure for discussing a node's evolving centrality over time and then used that structure to formulate and refine a process for estimating the node's growing centrality.

4.1. **Dataset.** I used publicly available data from the SNAP website. In particular I chose to use physics phenomenology and theory citation networks http://snap.stanford.edu/data/cit-HepTh.html and http://snap.stanford.edu/data/cit-HepPh.html . The reason why I chose to use these data sets is since they are citation networks for two very similar fields, they should have similar growth patterns and structure. Secondly, they come with information about when each article was published. This is very important for establishing the centrality over time structure that I analyzed. With these characteristics, they are perfect for the task at hand. I used the theory data set as training data and the phenomenology data as testing data.

4.2. **Data Structure.** The end goal of the project was to be able to talk about the role of one node in a citation network and how that role changes as the node is added to the network and the network grows around it. In this sense, the data that was collected for each node revolved around looking at the centrality of that node and how that centrality ended up changing over time. The way this worked was that I started with the day that the node was added to the network (the day the paper was published) and created the subgraph representing the citation network at that point in time. From there, we can then calculate the centrality of the node at that point in time. From then onward, we revisit the citation network a step forward in time and reevaluate the node's centrality at this new point in time. We can keep doing this until we get to present day (or the final state of the citation network). We now have a function representation of how the node's centrality develops. Due to the time interval that these citation graphs span (about 10 years or so), I decided that the best time step for this update would be monthly since we end up with around 50 or so measurements per node.

4.3. **Centrality Type.** There are different types of node centrality and how it is measured is an important aspect of the project. My goal in picking centrality measurements was to be as true to the original goal of measuring citation as possible. Since we are focusing on how much a paper is cited, I used a slightly modified version of centrality, let's call it In Centrality: $inDegree/(N-1)$ where N is the number of nodes in the graph. This is the standard definition of centrality modified to ignore the out degree of a node. The reason why we do this is that we don't care about how many other papers this one cites, we only care about how many papers cite it. Another notion of centrality that I thought would be relevant is the closeness centrality since that considers how closely connected this paper is to all of the papers in the network via citations. In theory, this sounds like a good idea since it is a more indirect way of measuring how influential a paper is. This could be a more useful metric than In Centrality in the case where a paper itself isn't cited many times but those papers that do cite it build off of some important foundation and become widely cited themselves. In this case, the In Centrality of the node may be low even though the paper is influential. The idea was that closeness centrality would pick up on this indirect importance since the node would be quite closely connected to all those descendants of the important works the paper influences.

4.4. **Prediction Problem.** As mentioned at the beginning, the goal of this project was to develop a method for looking at the initial growth of a node's centrality and then estimate its future centrality based on this initial growth. If we look at the function representation from before, the idea is that we start with the first x months of a node's centrality in its citation graph and then we try to predict the node's centrality over the next y months using these initial data points.

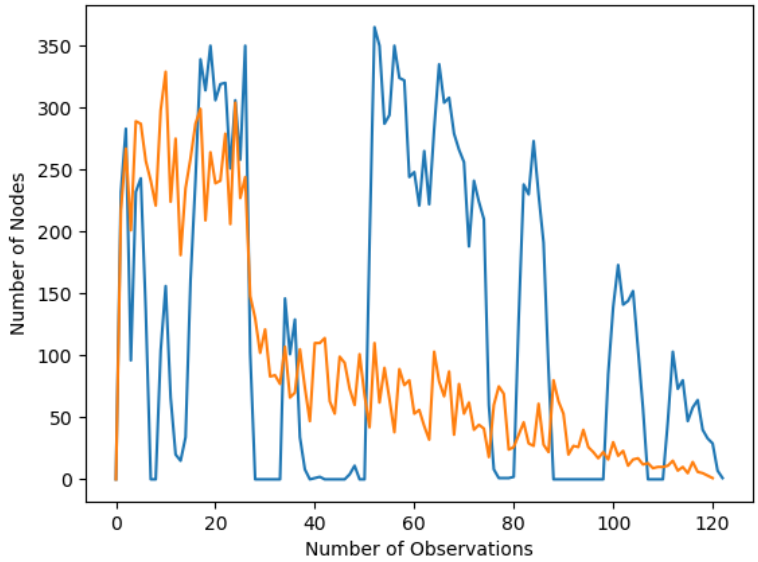
4.5. **Prediction Algorithm Design.** Predicting a whole function is a fairly difficult problem. In this particular case, I decided to make my algorithm be a twist off of a problem I worked on in CS229 last year. The basic intuition behind the approach is that we know that the function we are predicting could take on one of many different shapes. Rather than try to make an equation to model the shape of the function, we can instead leverage our training data and linearly combine functions in our training data to make a prediction. Therefore, it makes the most sense to use a k nearest neighbors approach: we pick the k "closest" training examples to our test case over the first x months. Then, we take an average of those k examples as an estimate for the future of the test case. There are many specifics here that are very important. First off, how do we measure which training examples are closest? Well there are a few different metrics that which can be used to measure the distance between two functions. Rather than commit to one ahead of time, it seemed like better experimental design to explore the possible options first. I will go into more details on these metrics soon. In addition to using a metric to determine closeness, I thought it only made sense that we also take a weighted average of k nearest neighbors rather than just taking them all to have the same weight. The weights are of the form $exp(\alpha dist(f,g))$ where α is some nonpositive constant and dist is the metric we are using to estimate the distance between the functions. This weighting is very small for functions that are far from the one that we are trying to

estimate and larger for functions that are close. A note: in order to properly test this algorithm, we need to have $x + y$ months worth of data for the centrality curve we are trying to predict. In order to choose proper values for x and y, I first had to examine the lifespans of different papers in the graphs. See more below.

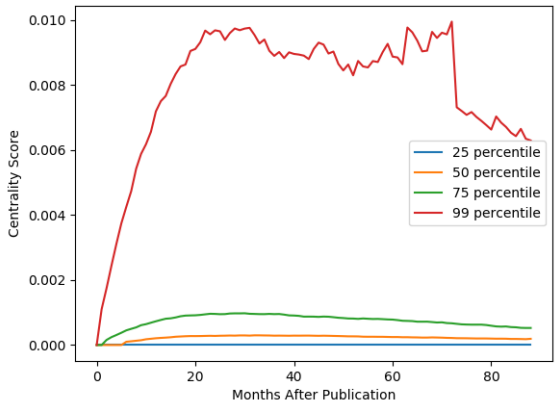
4.6. **Possible Metrics.** There are many possible metrics for determining distance between two centrality functions. I choose 4 candidates: the clear choices are the $L2$ metric: $\sum_{month=0}^{x}(f(month)-g(month))^2$ and the $L1$ metric: $\sum_{month=0}^{x} abs(f(month)-g(month))$. Realizing that these functions grow over time and it is likely that the behavior of the function near month x is most important, I created two more metrics: the relative scale metric: $\sum_{month=0}^{x} abs((f(month)-g(month))/(f(month)+g(month))$ and the boundary-focused metric: $\sum_{month=0}^{x} abs(f(month)-g(month)/(x-month)$. Each of these push for a function that fits the first x months of the testing in a more focused way. The boundary-focused metric weighs more heavily toward being close to the test function near time x. The scaled metric focuses on percentage difference between two points in the functions rather than the actual difference.

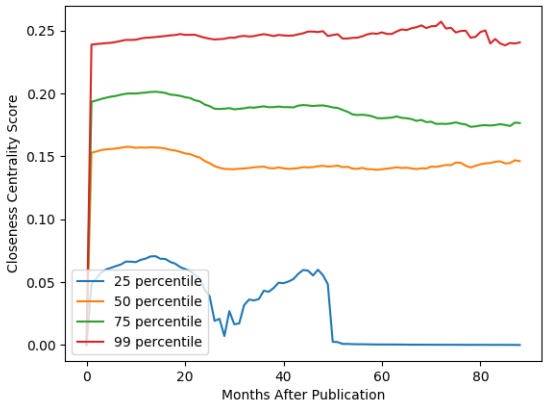
5. Results and Analysis

5.1. **Data Set Summary Statistics.** To start things off, I wanted to present some summary statistics on certain aspects of the centrality over time functions. These are important since they informed the rest of my experimenting and data analysis. First we have the distribution curve for the lifetime of the papers in the theory and phenomenology networks.



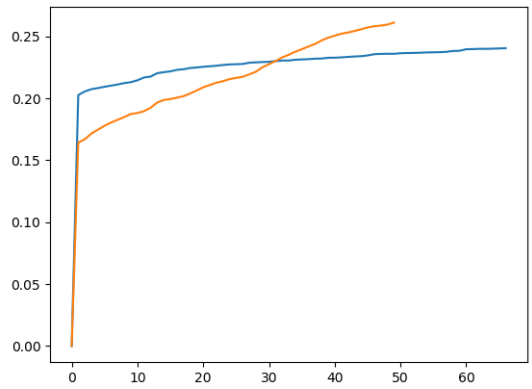
Note that the blue curve is phenomenology and the orange curve is theory. Based on this data, I choose the lifespan of all predictions (the value $x + y$) to be 50 since setting it any higher than that would exclude too large of a fraction of the training data and setting it any lower would make the analysis harder since we want to make x as large as reasonably possible. The reason why I wanted to mention these things first is that they influenced how my values for x and y pan out which definitely effects the results of the experiment. Given this data, I started $x = 25$ and $y = 25$, but later I tried out decreasing x and increasing y to see if the prediction can be as effective with less information. The additional summary statistics I wanted to offer are the percentile distributions for In Centrality and Closeness Centrality over the theory network.

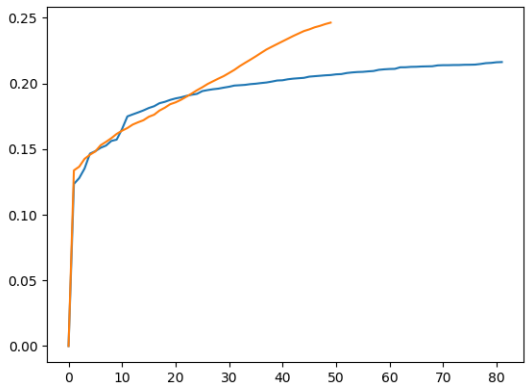




The graph for In Centrality shows a pattern of early growth followed by later decay. This

should be expected since we would expect for the In Centrality of a node to grow as others start citing the paper. Eventually, the information in the paper will become less relevant for current research and the paper would not be cited often even though new papers are being published. This would be the decrease in centrality that we are seeing. The graph for Closeness Centrality is less promising. The main concern here is that there is not much change in the values for Closeness Centrality over time. This tells us that Closeness Centrality is likely not a good measurement of how the importance of a paper changes over time since it doesn't change much for most nodes. With further investigation, I found that this was the case a decent fraction of the time, so I ended up shifting my analysis to focus on In Centrality exclusively. The reason why I think this is an issue is that closeness centrality could easily be measuring which important papers this particular one cites rather than how influential this one is on future works. Here are two examples in blue and my estimator's attempt to fit them in orange. Most of them closeness centrality curves look the left one where the closeness centrality starts high and stays there. A couple closeness centralities grow over time such as the right but these are definitely in the minority.





If I could redo this part of the project, I would likely modify the formula for closeness centrality as well so it only considers the incoming edges to the node only. As it stands now, loading centrality data takes about half of a day and I can't tie up my computer for that long again.

5.2. **Best Metric For Predicting Future Centrality.** In the Algorithm section, I proposed 4 potential metrics for picking the k-nearest neighbors and their weightings. I will now go over my method for determining which of these is best. For the sake of consistency, it is necessary to choose one consistent measure of how well these processes are performing. The most traditional method of looking at the distance between two functions is the integral of the difference, which essentially gives us the total area between the estimate and the actual function. In terms of the 4 pre-defined metrics, the closest one to this integral would be the $L1$ metric since that is essentially a Riemann sum used to estimate the integral. In this case, we do want to punish straying far from the actual values a little bit harder, so I chose

to use the $L2$ metric instead. Formally, if we have our function f and an estimate to it g, we calculate how close f is to g using the $L2$ of their difference: $\sum(f(t) - g(t))^2$. This is quite unfair if we are summing over the matching data (the first x time points) since this would give the $L2$ norm an inherent advantage as a neighbor selection process since it is focused on minimizing the $L2$ norm of the neighbors on the matching. In order to eliminate this bias, we instead focus on looking only at how accurate our prediction is on the future (data after x time points). As mentioned previously, we are limiting our view to looking at most $x + y$ months into the future in order to having enough eligible training examples for matching. This gives us that we will measure success using the $L2$ norm on the estimate function and the actual function on the time steps between x and $x + y$: $\sum_{t=x}^{x+y-1} (f(t) - g(t))^2$. We will evaluate each algorithm's overall performance by looking at the average of these $L2$ norms over all of the different functions in the phenomenology set being approximated by the k-nearest neighbors in the theory set. Here is a table of some of the results using differing numbers of neighbors:

x	y	α	k	metric type	mean square error
25	25	-5	5	L2	$4.05 * 10^{-6}$
25	25	-5	5	L1	$4.56 * 10^{-6}$
25	25	-5	5	Boundary	$28.5 * 10^{-6}$
25	25	-5	5	Scaled	$9.9 * 10^{-6}$
25	25	-5	3	L2	$4.74 * 10^{-6}$
25	25	-5	3	L1	$4.58 * 10^{-6}$
25	25	-5	3	Boundary	$28.5 * 10^{-6}$
25	25	-5	3	Scaled	$10.2 * 10^{-6}$
25	25	-5	7	L2	$3.83 * 10^{-6}$
25	25	-5	7	L1	$4.12 * 10^{-6}$
25	25	-5	7	Boundary	$414 * 10^{-6}$
25	25	-5	7	Scaled	$9.82 * 10^{-6}$

Given everything here, it is very clear that the $L2$ and $L1$ norms definitely preform the best (and in fact very similarly to each other). This is a very significant finding. It tells us that the other two metrics are not as good at capturing the essence of the functions as $L1$ and $L2$. The failure of the boundary metric tells us that the behavior of the nodes' centrality some significant amount of time before the present does gives us information about how the centrality will change in the future. The failure of the Scaled metric tells us that trying to scale the centrality difference to how large the centrality currently is worse than not scaling it all. This tells us that the amount of centrality a node has is relevant, and growth is not simply proportional to its magnitude.

5.3. **Optimizing Parameters.** There are a number of parameters that effect how the k-nearest neighbor matching works and how the neighbors are combined to form an estimate. α, k, x, and y play the important roles here. If we look at the different variables, we see

that α and k don't mean anything significant in terms of what the output means, so we can change these around in whatever way will best optimize our results. If we look at x and y, it is clear these do say something about our results and what they mean since they are how long of a time interval we are given to make our prediction and how far into the future of a prediction are we expected to make. For that reason they will not be messed around with here since we are currently discussing optimizing the parameters of the model. I played around a little bit with the parameters to see if I could find optimal ones for the $L2$ norm and for the $L1$ norm. Here is another table of results.

x	y	α	k	metric type	mean square error
25	25	-5	5	L2	$4.05 * 10^{-6}$
25	25	-5	5	L1	$4.56 * 10^{-6}$
25	25	-5	10	L2	$3.63 * 10^{-6}$
25	25	-5	10	L1	$3.87 * 10^{-6}$
25	25	-5	15	L2	$3.65 * 10^{-6}$
25	25	-5	15	L1	$3.87 * 10^{-6}$
25	25	-3	5	L2	$4.05 * 10^{-6}$
25	25	-3	5	L1	$4.58 * 10^{-6}$
25	25	-3	10	L2	$3.63 * 10^{-6}$
25	25	-3	10	L1	$3.86 * 10^{-6}$
25	25	-3	15	L2	$3.65 * 10^{-6}$
25	25	-3	15	L1	$3.82 * 10^{-6}$
25	25	0	5	L2	$4.05 * 10^{-6}$
25	25	0	5	L1	$4.63 * 10^{-6}$
25	25	0	3	L2	$4.74 * 10^{-6}$
25	25	0	3	L1	$4.56 * 10^{-6}$
25	25	-1000	5	L2	$4.11 * 10^{-6}$
25	25	-1000	5	L1	$7.31 * 10^{-6}$
25	25	-10000	15	L2	$4.84 * 10^{-6}$
25	25	-10000	15	L1	$8.32 * 10^{-6}$

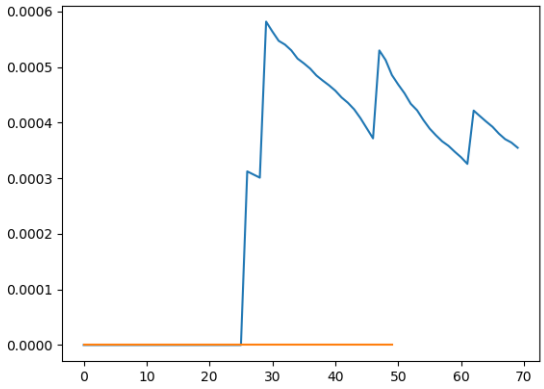
As you can see, the end result is that the $L2$ norm works best with $k = 10$ and $\alpha = -3$ or -5 and the $L1$ norm works best with $k = 15$ and $\alpha = -3$. The most interesting part about this probably is that the effectiveness of k-nearest neighbors method doesn't change much as we take different values of k and α. Errors being similar across α's is likely a side-effect of the fact that for α values that are as small as the ones we are looking at, is likely due to the fact that the given α values barely weight the average at all. It is not quite as clear why the k values are giving similar results across the different options. My main guess is that since we have so many different training examples, it is likely that many of them are quite similar to the testing example we are looking at. If this is the case, then it doesn't matter too much how many of them we choose, as long as we choose a reasonable number of them.

5.4. **Pushing The Limits of Prior Information.** Ideally, we want to be able to make as large of a future estimate as possible (large y) with the smallest prior information (x) possible. Here we will discuss the tradeoffs of pushing this idea forward. As mentioned previously, in order to keep the pool of eligible training functions large, I picked the total amount of time we are considering $x+y$ to be constant at $x+y=50$. For all of the previous work, we stuck with $x=25$ and $y=25$, predicting exactly half of the centrality. Now, that we have a good idea of the optimal metric, k, and α, we can use the best parameters for $L1$ and $L2$ to take a look at decreasing x. Here are the results:

x	y	α	k	metric type	mean square error
25	25	-3	10	L2	$3.63 * 10^{-6}$
25	25	-3	10	L1	$3.86 * 10^{-6}$
15	35	-3	10	L2	$11.6 * 10^{-6}$
15	35	-3	10	L1	$11.9 * 10^{-6}$
8	42	-3	10	L2	$29.2 * 10^{-6}$
8	42	-3	10	L1	$28.2 * 10^{-6}$

As we can see here, reducing the amount of time we have to observe the function does significant increase the error of our estimate. On one hand, we do expect the error to go up since we are summing over a larger y, but if the error was around the same at each time step, it would increase much less than we see here. Additionally, this growth in error is expected since we are making predictions farther into the future using less information about the node's current centrality.

5.5. **Issue: Late Bloomers.** One inevitable issue with this problem and trying to get a good estimate of a function is the idea of a late bloomer. Imagine a situation where a paper is published and it stays quietly unknown for a while. One day a few years later, some researcher picks it up and realizes its significance in their work and cites it. This causes the paper's popularity to begin to bloom and become much more cited long after it was originally published. Consider this blue actual function from the phenomenology data set and this orange estimation.



We see that if x is not more than 25, then all we see that the the first x parts of this function are 0. That means we will get a constant 0 function for the representative of this node in the matching algorithm. This means that the matching algorithm will pick the orange function seen here as the closest approximation. Clearly this approximation is not accurate. The issue here is that there does not appear to be any way to predict if a function will be a late bloomer or not since there is no way to differentiate it from a function that stays constantly at 0 within the first x time steps.

5.6. **Estimation Successful?** The last thing to look at is to try to determine how successful our estimation really is. This is a bit tricky. If we look at our MSE averages from above, with the best parameters, we are off by about 0.00035 per timestep on average. This is an overestimate since we are using the $L2$ norm. If we look at the overall scale of the In Centralities, we see that they often are on a similar scale to this with a median peak centrality of about 0.0004 or so. At first glance this looks terrible, but the estimations are actually not that bad. If we consider the fact that In Centralities on the high end get quite large (to 0.01 and higher), this average error would be much worse if the estimates were truly terrible. From the examples I looked at (I would include some, but no space and weird LaTex graph spacing), the estimates are still quite rough, but they do the job and offer a prediction that is usually not extremely far off (within 30% or so) from the actual values.

6. Works Cited

(1) P. Erdos, A. Renyi. On the evolution of random graphs. Magyar Tud. Akad. Mat. Kutato Int. Koezl., 1960.
(2) CS229 Problem Set 1, Problem 4 http://cs229.stanford.edu/ps/ps1/ps1.pdf (I know this is from this year, but the problem is the same as before).
(3) J. Leskovec, J. Kleinberg, C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM TKDD, 2007.
(4) G. Kossinets, D.J. Watts. Empirical Analysis of an Evolving Social Network. Science, 2006.