Fast Random Approximation of Closeness Centrality Using a Threshold

DANIEL CHAPARRO and JOHN FULGONI
Columbia University in the City of New York

1. PROBLEM STATEMENT

1.1 Introduction

Closeness Centrality is the measure of how important a node is within a network. We can use a node's distance to all other nodes as a tool to determine how valuable that node is amongst others. The definition described in [Boldi and Vigna 2013; Cohen et al. 2014] measures the closeness centrality of a node as the average distance from it to every node. This can be computed by using a single-source shortest path algorithm, such as Dijkstra's Algorithm, in order to find distances between all pairs of nodes. It is important to note that while the closeness centrality of a particular node may be useful, it is much better to see how the closeness of one node compares relatively to that of all other nodes in the data set. Since running Dijkstra's Algorithm for every node would result in a running time of $O(n^3)$, researchers have been trying to make algorithms that get an approximation of the closeness centrality. It has been shown in [Eppstein and Wang 2000] that random sampling can provide a linear time approximation of the closeness centrality while also having a minimal error component.

Centrality is a fundamental tool in the study of social networks [Boldi and Vigna 2013]. We can use centrality in order to find the most important nodes with respect to other nodes in the set. Since centrality is the idea of a node's importance, it is also a matter of opinion of what a "better" solution to closeness and importance might be. There have been many bases for the worth of a node in a network, and many metrics have been designed in order to display importance.

1.2 Choices and Assumptions

We are given a graph *G(V,E)* with n vertices and m edges. We are particularly interested in closeness centrality in order to measure the worth of each node in the graph. In order to improve the speed of our algorithm, we decided to use a model proposed previously by Eppstein [Eppstein and Wang 2000]:

$$c_v = \frac{n-1}{\sum_{u \in V} d(u,v)} \tag{1}$$

Where d(u,v) is the Dijkstra's shortest path distance from node u to node v in Graph G. Like in [Eppstein and Wang 2000], if G is not connected, then c = 0. So we will test to make sure that our graph is connected.

To compare values of closeness, we used the theorem that was proven in Eppstein [Eppstein and Wang 2000]:

$$E\left[\frac{1}{\hat{c}_u}\right] = \frac{1}{c_u} \tag{2}$$

Boldi's intuition behind closeness is that the nodes that are more central have smaller distances, and a small denominator in equation 1 will result in a larger centrality [Boldi and Vigna 2013].

For our tests, we decided to use a relatively small data set from Facebook, which consisted of 4039 nodes in the graph[1]. While trying to compute the classic centrality, as described in [Cohen et al. 2014; Eppstein and Wang 2000] as the shortest path distance to all other nodes, we found that we did not have the capability to compute all of the closeness values in a timely fashion. It is understandable that this will take a lot of time, since the algorithm runs in $O(n^3)$. We decided to cut the given graph down to the first 2000 nodes in order to make testing more viable. Even though the number of nodes of the graph was cut in half, we found that the calculated error was minimal compared to the time saved running the algorithms. When cutting the data provided by Facebook, we tested to make sure that our graph was connected, otherwise the algorithm will fail.

Finally, since we are using a small sample to estimate the centrality of each node, we made the assumption that a random set of nodes with more connections (edges) will better represent the graph as a whole than just a random set of nodes.

1.3 Problem Formulation

In this paper, we show that a random sampling will provide a fast approximation of the closeness centrality, and that computing closeness only from nodes within a threshold will provide a more accurate approximation. In order to do this, we compute the value for closeness using equation 1 as well as the theorem for expected closeness 2 used in Eppstein [Eppstein and Wang 2000]. While computing closeness by choosing a random sample of connected nodes has shown to be reasonably accurate at a much more efficient speed, we will add a threshold to the random sampling process.

The idea from a threshold comes from the accepted notion that the smaller the sum of distances, the larger the centrality [Boldi and Vigna 2013]. We want to test the theory that if the random sampling is adjusted to only include nodes with a certain level of degree of edges. If the random sampling happens to choose a node with very few connections, it is unlikely in a large graph that the node is close to the node we are testing. By creating a threshold where a node will only be selected if the degree of the node is above the threshold, then we have a higher chance of picking a node that is closer to the node being tested.

We base the threshold off of the number of edges each node has, and we test two different methods. For the threshold, we do two separate experiments: one where the threshold is based on the median number of edges of all the nodes, and one where the threshold is based on the average number of edges of all the nodes. In both experiments, while choosing nodes at random, we only add a node to the random sampling set if it has a number of edges greater than the threshold.

[1] Data obtained from the Stanford Network Analysis Project http://snap.stanford.edu/data/egonets-Facebook.html

2. RELATED WORK

The problem of determining the importance of a node through closeness centrality have been discussed in numerous papers.

Boldi and Vigna [Boldi and Vigna 2013] have done extensive research on the properties of centrality and related definitions. They include definitions in their work for closeness, centrality, page rank, and more. They also discuss the different approaches to calculate centrality and the worth of nodes. For instance, different methods in the past have included the degree of a node, the node that is closest to other nodes, and the node through which most shortest paths pass. Boldi and Vigna discuss the notion of classic closeness, which for undirected, connected networks as the reciprocal of the sum of distances from a given node. Boldi and Vigna also reason that the smaller the sum of the distances from a given node, the more close that node is to other nodes, which is logically sound.

In *Computing Classic Closeness Centrality, at Scale*[Cohen et al. 2014], written by Edith Cohen, Daniel Delling, Thomas Pajor, and Renator F. Werneck, the four writers discuss the notion of sampling nodes and pivoting to compute closeness. In their experiments, they take two approaches to computing closeness. After computing the exact closeness, they take a uniform sample of nodes and calculate the shortest distance from each with respect to one node in order to calculate its closeness. Cohen et. al also propose the notion of pivoting, where they define the pivot of a node as the node in the sample which is closest to the original node. They then use the pivot's centrality in order to estimate the centrality of the node they are testing. This can result in high errors when not bounded, so they define a threshold in order to minimize the errors. They base their threshold on the distance of non-sampled nodes from the node they are testing. In the end, they found that using a hybrid of sampling and pivoting gave the best results.

Eppstein and Wang [Eppstein and Wang 2000] discuss the idea of creating a fast approximation of centrality by using random sampling. For their tests, they have a unweighted, connected graph G(V,E) with n nodes and m edges. Their algorithm consists of picking a vertex v, and then running a single-source shortest path algorithm from v to a random set of nodes. They then prove that the estimated value of centrality is equal to the true calculated value of centrality. We show this equation above in 2. The algorithm that they used, and that we modeled our algorithm after, went as follows:

(1) Let k be the number of iterations needed to obtain the desired error bound.
(2) In iteration i, pick vertex v uniformly at random from G and solve the SSSP problem with v as the source.
(3) Calculate the average centrality for vertex v, using equation 1

Eppstein and Wang state that since the single-source shortest path problem can be solved in O(nlogn + m) time, their algorithm takes O(k * m) time, with k being the number of iterations in order to bound error. Eppstein and Wang's research is primarily what made us think of the problem we face in this paper. If a random sampling of nodes can be proved to be enough to calculate a low error estimation of centrality, then there must be a more accurate way to compute centrality.

3. RESULTS

3.1 Our Algorithm and Testing

The algorithm we used is related to the algorithm proposed in [Eppstein and Wang 2000], but we altered the sampling process in order to get a lower error percentage. We did two sets of tests, based on different ideas for setting the threshold. Our first idea was to make the threshold the median degree of all the nodes in graph G. While setting the threshold to the median proved to be better than the sample, we found that setting the threshold to the average degree of all nodes in graph G resulted in a slightly lower error for centrality. From the results above, we can see that our method has a lower average error than the algorithm used by Eppstein and Wang.

Table I. Average Error for each Method using 2000 Nodes from Facebook Data

Test	Sampling	Average	Median
1	0.074255945	0.023672966	0.035658377
2	0.228441681	0.114593497	0.105286963
3	0.127333319	0.034793092	0.161474742
4	0.213221681	0.062091681	0.029957141
5	0.022076748	0.141191103	0.073128854
Average	0.133065875	0.075268468	0.081101216

(1) Let the number of sample nodes be 5% of the total number of nodes in the set.
(2) Populate the set of samples by randomly selecting nodes from graph G, as long as they have a higher degree than the threshold
(3) For each node v in graph G, calculate the SSSP distance to all nodes in the sample set.
(4) Calculate the average centrality for vertex v, using equation 1

For comparison, we decided to run four different algorithms on our set of data. The first (not shown in table) is the true, classic closeness from [Cohen et al. 2014] which runs in $O(n^3)$ time. The second algorithm, labeled as Sampling, is the random sampling algorithm used in Eppstein [Eppstein and Wang 2000] is the average error after five runs of the algorithm. The third and fourth algorithms are the modified random sampling using a threshold set either to the average number of edges, or the median number of edges. Since the algorithm used in [Eppstein and Wang 2000] was proven to be O(k*m) time, we can say that our algorithm runs in the same time, since we only change the the way we pick the nodes used in the sampling, not the process to calculate the centrality.

3.2 Claim 1: Median Based Threshold Provides a Fast Approximation

In table I we can see that by adding a threshold to the random sampling, we are able to produce results in linear time with little error. By running several trials, we show that the average error of our algorithm with the median threshold is 0.081101216, which is considerably better than the random sampling algorithm used in [Eppstein and Wang 2000], which had an average error of 0.133065875.

3.3 Claim 2: Average Based Threshold Provides a Fast, More Accurate Approximation

Table I shows that setting a threshold to the median number of edges will provide a fast approximation of closeness, and also that setting the threshold to the average number of edges is a slightly more accurate algorithm. After five runs, the average error of the average threshold was 0.075268468, which is significantly better than the original random sampling algorithm in [Eppstein and Wang 2000]. It is important to note that while the average based threshold has a lesser error than the median based threshold on average, there were some tests where the median would perform just as good or better than the average based threshold.

4. CONCLUSION

We believe that this problem provides the basis for further research. If the random sampling can be adjusted further so that better choices are made, than it might be possible to get a fast approximation of closeness that might have negligible errors. While the idea of closeness is up to opinion, we believe that we have created a method of fast approximation for classic closeness that can be studied further with even less error.

REFERENCES

Paolo Boldi and Sebastiano Vigna. 2013. Axioms for Centrality. *CoRR* abs/1308.2140 (2013). http://arxiv.org/abs/1308.2140

Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Computing Classic Closeness Centrality, at Scale. *CoRR* abs/1409.0035 (2014). http://arxiv.org/abs/1409.0035

David Eppstein and Joseph Wang. 2000. Fast Approximation of Centrality. *CoRR* cs.DS/0009005 (2000). http://arxiv.org/abs/cs.DS/0009005