Computing top-k Closeness Centrality Faster in Unweighted Graphs

Social Network Analysis for Computer Scientists — Course paper

Antonis Mouratis
a.mouratis@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Tushar Pal
t.pal@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

Centrality measures give us a better understanding of the flow of information inside the network, by measuring the reachability or accessibility of the nodes [10]. Closeness centrality measures how short the shortest paths are from node i to all nodes. As one may assume the problem of closeness centrality is directly connected to the APSP(All Pairs Shortest Path). Since, the best method to solve it is BFS(Breadth first search) with complexity $O(m \cdot n)$, given a graph $G = (V, E)$ with $|V| = n, |E| = m$, and because social networks grow exponentially, there is a need to approach the problem from a different angle(e.g. approximation). Here, we verify the method described in [1] to calculate top-k nodes by closeness centrality as being both faster and having similiar results as the full BFS version. We then compare the new approach to other related centrality measures to see the potential for substitution and reduction in compute time. We also see the application of top-k closeness centrality to solve real world problem statements.

KEYWORDS

Centrality, closeness, farness, social network analysis, network science

ACM Reference Format:
Antonis Mouratis and Tushar Pal. 2022. Computing top-k Closeness Centrality Faster in Unweighted Graphs: Social Network Analysis for Computer Scientists — Course paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2022 (SNACS '22).* ACM, New York, NY, USA, 4 pages.

1 INTRODUCTION

In this day and age, social media platforms have become the center of attention. People worldwide use them to the extent where they have become an integral part of our daily lives and are thus a good indicator of who we are. Organisations now want to use these social networks to identify patterns and find the appropriate targets for tasks like ad campaigns and fraud detection. Network analysis helps utilise resources more efficiently and profitably towards the required demographic.

To furnish this type of work, identifying the central or most influential node is a fundamental task, and the metric for that task is the work of centrality measures. Centrality indicates how crucial a node of the network is, giving intuition as to how it affects the rest of the nodes and the flow of information through it. The simplest approach is **Degree** centrality, which indicates how much direct influence a node has on its neighbors. As noted in [6] and [7], there are various means of calculating centrality in a social network, that are used depending on the target outcome desired from the operation.

A common issue in practical networks however, is that they can have a large number of data points leading to high time complexity when identifying the important nodes. Factoring in the multiple structures that a network can be and the vast applications of them, and we see that choosing a suitable centrality measure depends on a lot of factors. Our work thus is centered around the closeness centrality in unweighted networks approach proposed in [1]. This is a faster optimised algorithm that returns the top-k nodes ordered in descending order of closeness centrality. Our primary focus is centered around comparing two methods of that approach by trying different values of k and verifying the intuition that nodes with high degree centrality, tend to have also high closeness centrality. As a secondary contribution to original paper [1], we are going to compare the closeness centrality to other measures like pagerank.

We begin by verifying that the closeness centrality top-k ranking process of [Bergamini et all 2017] provides comparable results with standard approaches that use full breadth-first search (BFS). As noted by [Béres and Pálovics 2018][3], comparing rankings across centrality measures is difficult. By treating the rankings from the standard BFS based approach as the ground truth, we use Normalized Discounted Cumulative Gain (nDCG) as the comparison metric to ascertain how accurate the optimised algorithm performs. The time complexity in practice of top-k centrality is less than $O(m \cdot n)$ as it only processes for k nodes, while the standard approach first computes centrality for all nodes in $O(m \cdot n)$ then ranks them in $O(n)$ time. We also compare the variants of closeness centrality with degree and pagerank centrality using Kendall ranked correlation [9], to see how correlated the faster Top k closeness centrality is to other degree dependant centrality measures. We use two real world datasets to derive quantifiable usecases using closeness centrality. The first is the **Reddit interaction networks** [5] which contains monthly user interaction networks from the year 2014 for 2046 subreddit communities from reddit.com. Our intent is to identify top communities for targeted ad campaigns. The second dataset is the **NYC Taxi trips data** that contains trip details in New York City for Green, Yellow and ride sharing cabs. As referenced by [Deri et all 2016][4], computing a network of traffic movement for mapping using naive Dijkstra's algorithm is very compute and time

SNACS '22, Master CS, Fall 2022, Leiden, the Netherlands

intensive, requiring smarter optimised methods. Here our intent is to identify taxi pickup zones from where maximum locations can be reached the fastest, as a mitigation for demand surges in time. We use closeness centrality as taxi trips are mostly point to point, and the influence of trips from other zones to the target zone can be ignored.

As already mentioned, there has been research towards the top-k closeness centrality problem, mainly focused on the BFS algorithm. The goal is to build algorithms based on BFS that are faster in practice by proposing techniques like setting lower bound on the farness of the nodes [1], in order to set a limit at our computation budget, or categorize the networks(e.g. Dynamic networks [2] or Temporal networks [8]) and approach them differently based on their characteristics.

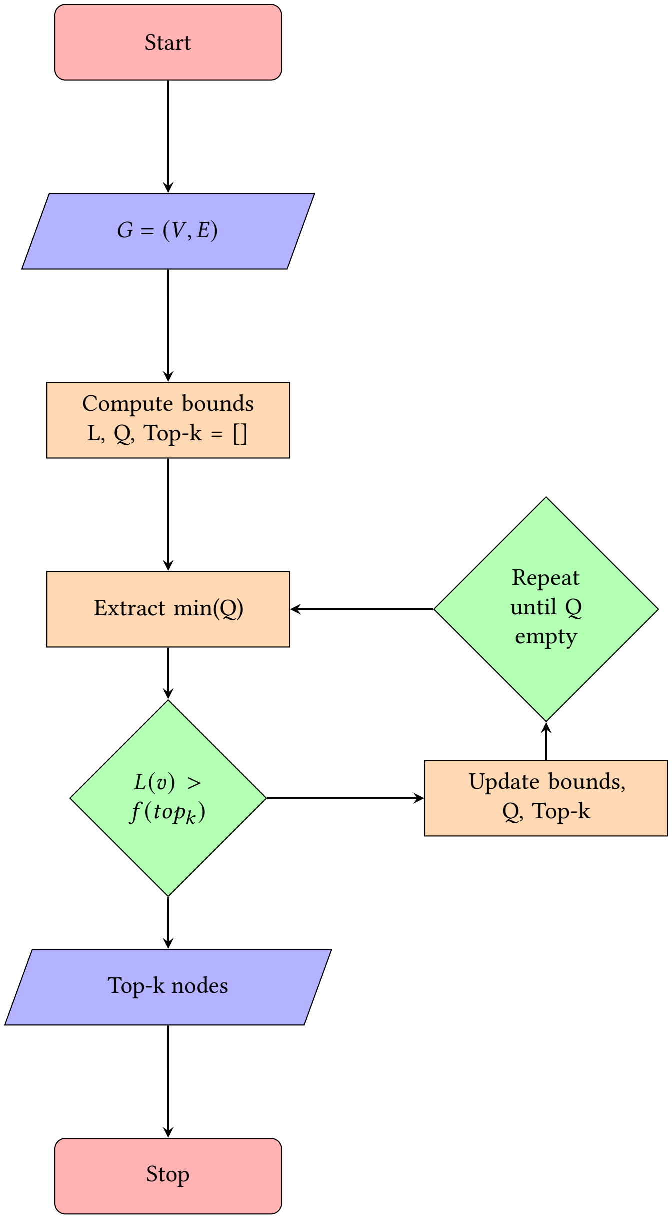
In the first section we are going to get in more detail about the problem by strictly defining necessary notation and giving a theoretical background. We will explain the main goal of this paper and give an overview of the approach of the algorithms that are going to be implementing. The next section will be about related work on the matter followed by the suggested algorithm and the two different approaches that are going to be implemented. The next section will be about the datasets that we will work on, to test our intuitions and suggestions and then propose and discuss the results from the experiments.

2 PROBLEM STATEMENT

In this section we will strictly define the problem of top-k closeness centrality. Our focus is on unweighted undirected graphs, as proposed in [1]. Given a graph $G = (V, E)$, $|V| = n$, $|E| = m$, we define R(v) as the set of reachachable nodes from v and $r(v) = |R(v)|$. Also, for the sake of simplicity we are going to use the definition of closeness and farness as described in the same source paper and it is (1):

$$f(v) = \frac{\sum_{w \in R(v)} d(w,v) \cdot (n-1)}{(r(v)-1)^2}, \quad c(v) = \frac{1}{f(v)} \tag{1}$$

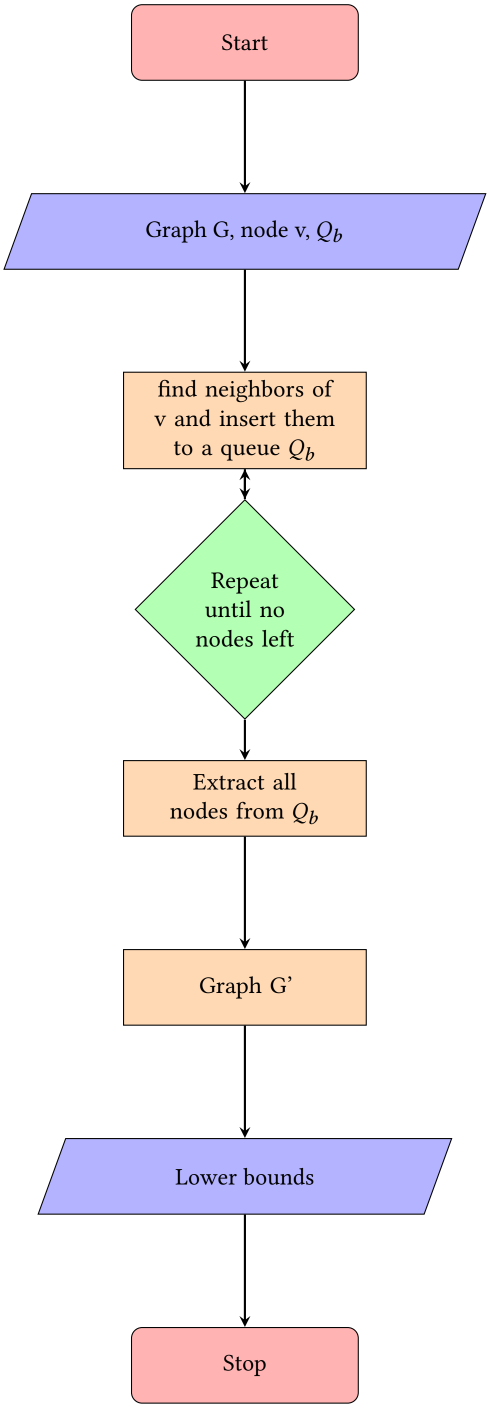
Overview of the algorithm:
Describe basic idea of the algorithm



Strategy for computing the bounds is Degree centrality

Two strategies for updating the bounds

The first strategy for updating the bounds is a pruning form of the BFS algorithm, BFScut, meaning that we keep track of lower bound $L(v)$ of node v of farness and exclude the nodes that cannot belong to the top-k ones based on their lower bound at each iteration. The BFS algorithm is described here (2):


Start
Graph G, node v, Q_b
find neighbors of v and insert them to a queue Q_b
Repeat until no nodes left
Extract all nodes from Q_b
Graph G'
Lower bounds
Stop

Flowchart for BFScut

Pseudocodes

3 RELATED WORK

Centrality measures are a group of metrics that aim to quantify a certain characteristic about the nodes in a network, and the nature of the influence between these nodes and their neighbors. However, the very nature of centrality calculation involves visiting all the nodes and parsing the related edges, which is traditionally not a scalable operation [Kang et al. 2011, Chen et al. 2012].

3.1 Degree Centrality

Degree centrality calculates the number of neighbors at distance 1 to a node in a network. Mathematically, degree centrality $C_D(v)$ is defined as $C_D(x) = d(v)$ where d(v) is the degree of a node v. This can be normalised again as $C'_D(v) = \frac{d(v)}{n-1}$ where n is the size of the network. The time complexity is O(n^2). It is the most intuitively understood measure, but also one that can be easily misleading.

3.2 Closeness Centrality

The idea behind closeness centrality is measuring how efficiently a node can spread information to other connected nodes. For a node v, is defined as $C_C(v) = \frac{n-1}{\sum_{w \in V} d(v,w)}$. However in our approach we define it in line with [Lin 1976; Wasserman and Faust 1994; Boldi and Vigna 2013; 2014; Olsen et al. 2014] as $C'_C(v) = \frac{(r(v)-1)^2}{\sum_{w \in V} d(v,w) \cdot (n-1)}$, where r(v) is the number of nodes reachable from v. The time complexity is O(m.n), where m is the number of edges in the network. In [Bergamini et all 2017] it is noted however that their implementation practically has complexity much lower than O(m.n).

3.3 Software Libraries

Even with all the research into optimised centrality algorithms, most implementations use the unoptimised full BFS textbook version. Boost Graph Library [Hagberg et al. 2008], igraph [Stein and Joyner 2005] and NetworkX [Siek et al. 2001] are notable mentions in this respect. This could be because efficient available exact algorithms for top-k closeness centrality, like [Olsen et al. 2014], are relatively recent and utilise several other non-trivial routines. We will use the NetworKit [Staudt et al. 2014] framework, which implements optimised Top-k centrality from [1] as well as the other centrality measures. Networkit is an open source C++ framework for network graph analysis.

4 ALGORITHM AND APPROACHES

5 DATA

6 EXPERIMENTS

7 CONCLUSION

ACKNOWLEDGMENTS

REFERENCES

[1] Elisabetta Bergamini, Michele Borassi, Pierluigi Crescenzi, Andrea Marino, and Henning Meyerhenke. 2019. Computing top-k closeness centrality faster in unweighted graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 5 (2019), 1–40.

[2] Patrick Bisenius, Elisabetta Bergamin, Eugenio Angriman, and Henning Meyerhenke. 2018. fully-dynamic graphs. In *2018 Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 21–35.

[3] Ferenc Béres, Róbert Pálovics, Anna Oláh, and András Benczúr. 2018. Temporal walk based centrality metric for graph streams. *Applied Network Science* 3 (08 2018). https://doi.org/10.1007/s41109-018-0080-5

[4] Joya A. Deri, Franz Franchetti, and José M. F. Moura. 2016. Big data computation of taxi movement in New York City. In *2016 IEEE International Conference on Big Data (Big Data)*. 2616–2625. https://doi.org/10.1109/BigData.2016.7840904

[5] William L. Hamilton, Justine Zhang, Cristian Danescu-Niculescu-Mizil, Dan Jurafsky, and Jure Leskovec. 2017. Loyalty in Online Communities. *CoRR* abs/1703.03386 (2017). arXiv:1703.03386 http://arxiv.org/abs/1703.03386

[6] Madhumangal Pal Kousik Das, Sovan Samanta. 2018. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining* 8, 13 (2018). https://doi.org/10.1007/s13278-018-0493-2

[7] Siti Nurulain Mohd Rum, Razali Yaakob, and Lilly Affendey. 2018. Detecting Influencers in Social Media Using Social Network Analysis (SNA). *International Journal of Engineering Technology* 7 (12 2018), 950. https://doi.org/10.14419/ijet.v7i4.38.27615

[8] Lutz Oettershagen and Petra Mutzel. 2022. temporal networks. *Knowledge and Information Systems* 64, 2 (2022), 507–535.

[9] Lutz Oettershagen, Petra Mutzel, and Nils M. Kriege. 2022. Temporal Walk Centrality: Ranking Nodes in Evolving Networks. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 1640–1650. https://doi.org/10.1145/3485447.3512210

[10] Rahul Saxena and Mahipal Jadeja. 2022. Network centrality measures: Role and importance in social networks. In *Principles of Social Networking*. Springer, 29–54.