Factors affecting Top-k Closeness Centrality and its relationship with degree centrality in unweighted undirected networks

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

Closeness centrality ranking helps us understand the flow of information inside a network, by measuring the reachability or accessibility of the nodes [23]. Closeness centrality measures how short the shortest paths are from node v 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 due to the exponential growth of edges for a given growth of nodes in a network, 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 delivering the same results as the full BFS version. We see that for BFSCUT variant of top-k centrality, degree and average clustering coefficient have a positive correlation with the computation time, while for LB variant the diameter has a highly negative correlation. We establish that diameter has an inverse effect on the correlation between degree and closeness centrality of a network, and density and average clustering coefficient have a positive effect. We also determine that in general, degree and closeness centrality of nodes are not correlated.

KEYWORDS

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

ACM Reference Format:
Antonis Mouratis and Tushar Pal. 2022. Factors affecting Top-k Closeness Centrality and its relationship with degree centrality in unweighted undirected networks: 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, 9 pages.

1 INTRODUCTION

In modern times, we work with multiple networks in various areas of application. Social media networks connect their users across time and space, road networks allow for efficient transportation, while collaboration networks connect like minds towards a common goal. Analysing these networks allows us to concentrate resources on nodes that best suit our goals, whether it is to identify influencers of fake news, or upcoming transport hubs in need of better facilities, or even major leaders in a corporate office setup. To furnish this type of work, identifying the central or most influential node is a fundamental task, performed through the use 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. Degree centrality calculates the number of neighbors at distance 1 to a node in a network. For a network with V vertices and E edges, 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 $C'_D(v) = \frac{d(v)}{|V|-1}$. The time complexity is O($|V|^2$) for a dense network, and O($|E|$) for a sparse network. It is the most intuitively understood measure, but also one that can be easily misleading. As noted in [11] and [17], there are various means of calculating centrality in a social network, that are used depending on the target outcome desired from the operation.

The metric we are interested in this paper is **closeness** centrality, which shows us how close a node in a graph is to the other nodes. From closeness centrality we understand which nodes are most important for the flow of information within the network. Measuring the influence of a user in a social network, or the importance of a node as a hub in a transport infrastructure are some of the applications of this centrality measure. A common issue in practical networks however, is the high correlation between network and time complexity. Thus, work has been done on developing more optimised algorithms for computing top-k nodes by closeness centrality [1]. Our work in this paper is centered around this source paper, on unweighted undirected networks. We will analyse the network metrics that affect the computation time of top-k closeness centrality with respect to vanilla. We will also determine the factors that influence a high correlation between degree and closeness centrality in a given network.

We first compare the time taken for the **BFSCUT** and **LB** variants of top-k and vanilla closeness centrality for varying number of nodes (k), identifying the network characteristics that influence computation time. We then verify that the top-k closeness centrality algorithm of Bergamini et all provides identical results as the standard approaches that use full breadth-first search (BFS). However comparing rankings across centrality measures is not straightforward [3]. To compare rankings, we use **Normalized Discounted Cumulative Gain** (nDCG) as the comparison metric to ascertain

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

how similiar two rankings are from different centrality measuring techniques. We will also use **KendallTau** rank correlation to compare pairwise positional similiarity of node rankings between algorithms. We use nDCG score and KendallTau to compare degree centrality ranking to vanilla and top-k closeness rankings [20].

In the first section we will define the problem statement in more detail by 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 implemented in the source paper. 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. Then we describe the datasets used in our experiments, followed by the experiments and results. We finally conclude with our inferences, as well as suggest future work.

2 PROBLEM STATEMENT

Closeness centrality gives shows us how close a node is to all other nodes in the network. That means it actually informs us how fast the information flows within the network and what are most the central nodes. The problem of finding such top-k nodes, given a graph $G = (V, E), |V| = n, |E| = m$ is not new and it is proven to be unsolvable in time $O(|E|^{2-\epsilon})$, in terms of complexity for $\epsilon > 0$. Its reduction is actually on the APSP problem(All Pairs Shortest Paths) of which the solution is based on the Breadth First Search(BFS) algorithm. In terms of complexity there is another way of solving the problem by using fast matrix multiplication, in time $O(n^{2.373} logn)$ [Zwick 2002; Williams 2012]. In the case of BFS, we use the algorithm for each vertex $v \in V$, in time $O(m \cdot n)$. While the BFS approach is preferred(real-world networks are usually sparse that is, m is not much bigger than n), it is too time-consuming if the input graph is very big. Thus, the aim is to find possible improvements to the algorithms by using approximation or present modifications of the BFS algorithm. It is proven in the source paper that at its worst case the algorithm cannot be further improved and the algorithm proposed performs much better on real-world networks.

In this section we will strictly define the problem of top-k closeness centrality. Our focus is on unweighted graphs, as proposed in [1]. We define R(v) as the set of reachable 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):

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

The rest of the important notation that is going to be used later for the algorithms proposed is given in the Notation table:

Symbol	Definition
$G = (V, E)$	Graph with node set V and edge set E
n, m	$\|V\|, \|E\|$
outdeg(v)	Number of nodes v goes to
deg(v)	Degree of node v
$d(v, w)$	Number of edges in a shortest path from v to w
$R(v)$	Set of nodes reachable from v
$r(v)$	$\|R(v)\|$
$\Gamma_d(v)$	Set of nodes at distance d from v
$\Gamma(v)$	Set of neighbors of v
$\gamma_d(v)$	$\|\Gamma_d(v)\|$
$\bar{\gamma}(v)$	Upper bound of γ
$S(v)$	Total distance of node v
$S_d^{CUT}(v, r)$	Lower bound on $S(v), r(v) = r$, for BFSCut
$S_d^{LB}(v, r)$	Lower bound on $S(v), r(v) = r$, for LB
$f(v)$	$f(v) = \frac{\sum_{w \in R(v)} d(w,v) \cdot (n-1)}{(r(v)-1)^2}$
$c(v)$	$c(v) = \frac{1}{f(v)}$
$L_d^{CUT}(v, r)$	Lower bound on $f(v)$, if $r(v) = r$

Table 1: Notation

As it is already mentioned, the approach proposed is based on the BFS algorithm, which is briefly described here (2.2), where $Q_1$is a queue with all the nodes of the graph andQ_2 is an empty queue. Its main idea is that we initialize a queue Q_1 with all the nodes and an empty Q_2. For each node v we find its neighbors v_i and then the neighbors of these nodes and extract them until Q_1 is empty and they are all inserted to Q_2. After the process comes to an end, we extract every node from Q_2, so that we have a tree, where the root is the node v, and we can easily find the all the paths to all the other nodes. By repeating this for every node in the graph we get all the possible shortest paths for each pair of nodes. That is the problem of All Pairs Shortest Paths(APSP). The time complexity of the BFS algorithm is $O(n + m)$, since we iterate over all nodes and their edges, and it is $O(n \cdot m)$, when we apply it for every node in the graph. An example is given in Figure 2.2, where i values represent the levels of the tree.

2.1 Top-k Closeness Centrality

It is proved in [1] that the complexity of the BFS-based approach cannot be further improved by constructing a reduction from the problem of computing the most central vertex (the case k = 1) to the Orthogonal Vector problem. So, the focus is shifted on dealing with real life networks and improving the performance on a practical level. Regarding the problem's hardness, it is proved in [24] that finding the least closeness-central vertex is not subquadratic-time solvable. In the same paper it is shown that in densly weighted graphs, the complexity of centrality measures is linked to the one in computing the APSP. In the same direction, it is proved in [1] that finding the most central node is not possible in $O(m^{2-\epsilon}$ time.

Thus, in order to deal with such hardness results, there has been research focused on proposing approximation algorithms, with the simplest approach being, sampling distance between a node v and l other nodes w, and returning the average of all values $d(v, w)$, with time complexity $O(l \cdot m)$. The most recent result of this is by

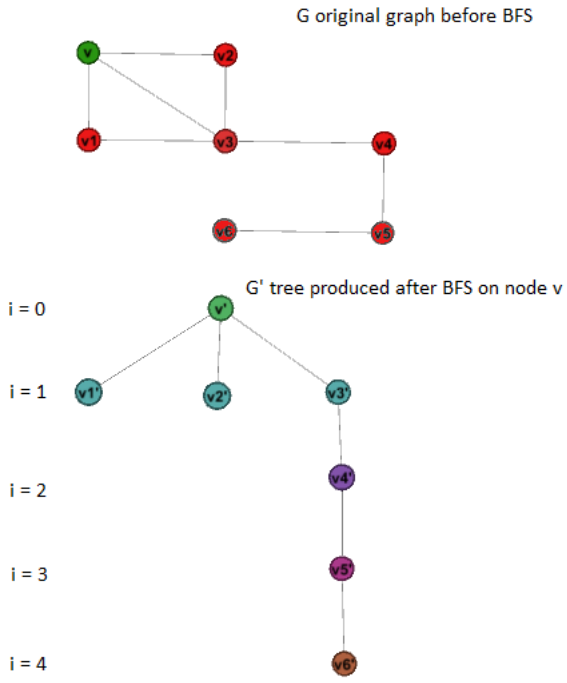


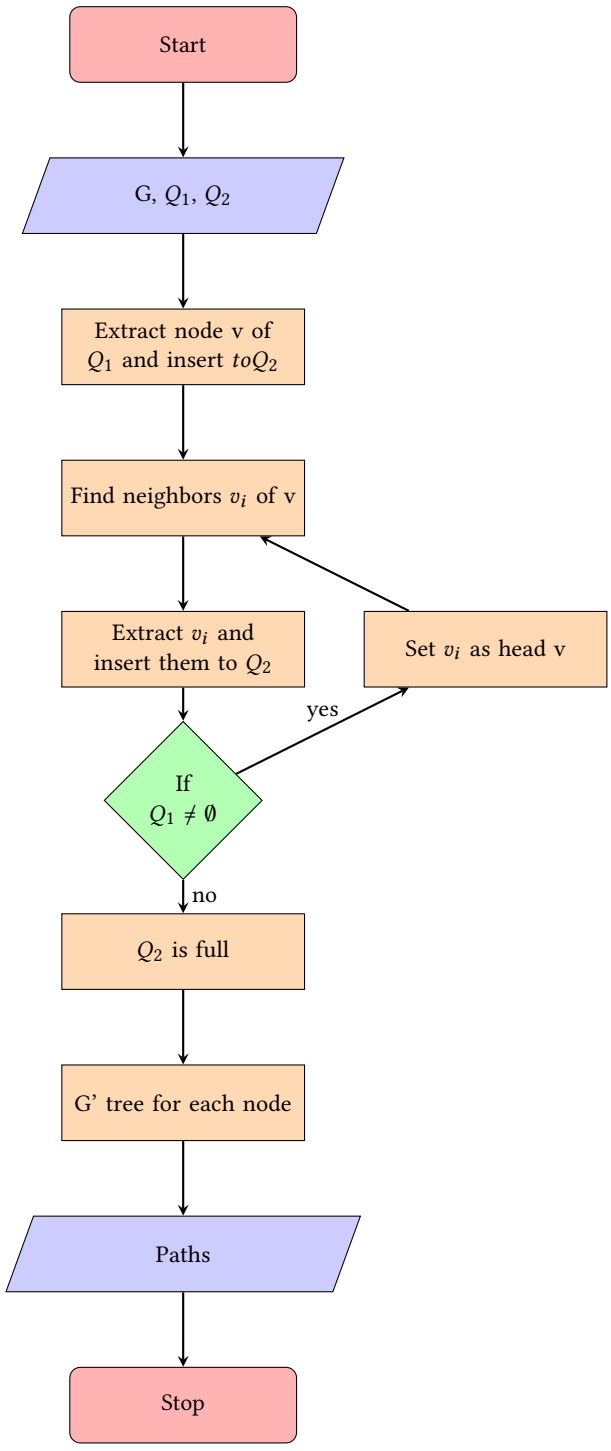
Figure 1: Example of BFS for node v before and after the algorithm

[4], where the algorithm proposed is an approximation of closeness centrality with a coefficient of variation ϵ using $O(\epsilon^{-2})$ single Source Shortest Paths(SSSP) computations. However, these techniques have been shown to not be a good fit for many real-world problems, where we work on low diameter graphs.

2.2 Problem statement definition

We now want to establish the conditions under which the hypothesis that high degree centrality nodes tend to have high closeness centrality holds true. Through this, we can use the degree centrality as a cheaper substitute function in places where time is a constraint and an approximation is all that is required. For example in temporal network data streams to identify emerging patterns of significance. Thus we compare degree centrality rankings to both vanilla closeness centrality, as well as top-k closeness centrality variants for the nodes of a network. We also compare vanilla and top-k closeness centrality to each other to verify that all algorithms return the exact same ranking of nodes for a given network.

We also want to establish the conditions under which computing top-k closeness centrality is favorable compared to vanilla closeness, based on the computation time. It is possible that for certain networks it is better to compute closeness centrality for all the nodes, instead of encountering the overhead of preprocessing for all nodes and computing only for top-k.



3 RELATED WORK

3.1 Closeness centrality approaches

As it is mentioned in [5], closeness centrality is a good metric to determine the most "influential" nodes within a network, but such a measure is not applicable in large-scale networks. That is why the focus is shifted on the relation between closeness and

other centrality metrics or approximation algorithms, with the simplest approach being to sample distances from a node v to l other nodes w as mentioned earlier and then return the average. The time complexity of this problem is $O(l \cdot m)$. More approximation algorithms are proposed in [6], [7] based on the concept of All-Distance Sketch, where the main idea is centered around computing the distances from a node v and other nodes, chosen in such way that they provide good estimates for the node's closeness centrality. This problem's time complexity is $O(logn)$.

Other approaches have mainly focused on real-world networks analyses and one of them is proposed in [16], where the authors develop heuristics to determine the top-k nodes in a varying environment. Another approach [21] is focused on exploiting the properties of real-world networks in order to develop algorithms that can work in practice much better than $O(m \cdot n)$. Last but not least, there are some that try to approach the problem by categorizing the networks. One of them is [2] which is focused on dynamic networks, and [19], [8] which are focused on temporal networks.

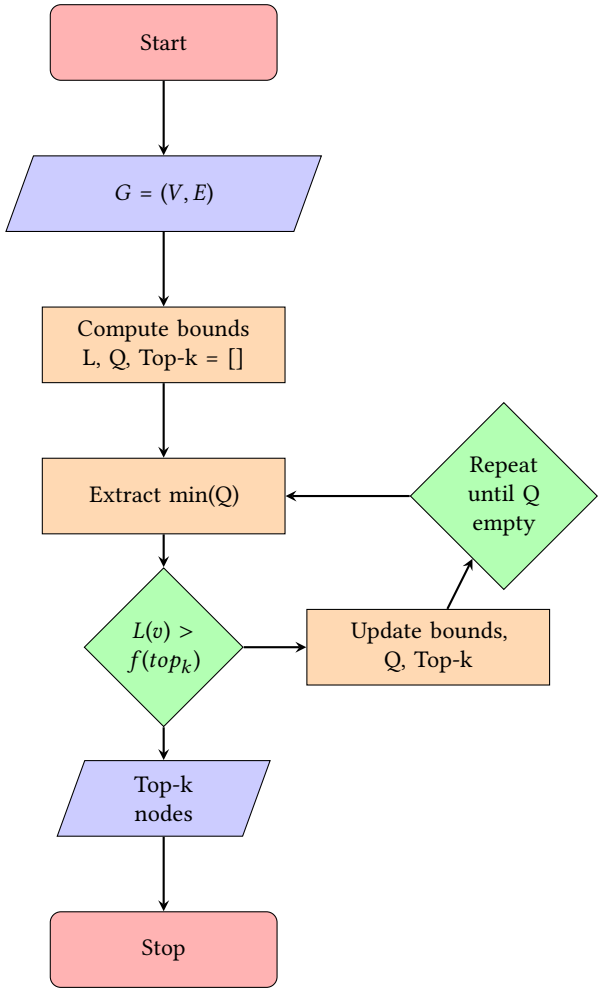
3.2 Software libraries used

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., are relatively recent and utilise several other non-trivial routines. We will use the NetworKit [Staudt et al. 2014] framework, version 10.0, which implements optimised top-k closeness centrality from [1] as well as the other centrality measures. Networkit is an open source C++ framework for network graph analysis, with python bindings, making our analysis streamlined.

4 ALGORITHM AND APPROACHES

Overview of the algorithm:

The main idea of the algorithm is centered around computing the farness for a batch a nodes $v_1, \ldots, v_k$ and inserting them to a list called "Top-k" in an increasing order. Then, compute lower bounds of farness $L(v_i)$ for the rest of the nodes and insert the nodes to a queue Q in an increasing order of lower bound farness as well. At each iteration we extract the head v of the Q, compare its value to the k-th element in the list and if $f(v) < L(v)$ then the algorithm stops because we can no longer find nodes with lower farness value. Otherwise, we update the lower bounds based on two methods. The source paper presents two strategies for the computation of the weights and two for their update. Our approach for the computation is going to be focused on the intuition that degree value of a node gives a good insight about its closeness as well. More specifically, nodes with higher degree centrality are most likely to be the ones that are the top-k ones in terms of closeness as well. By experimenting on different datasets, we will try to provide validation on that strong assumption as well. The algorithm is described in the flowchart(4):



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. An overview of the BFSCut algorithm is proposed in Figure 2.

The second approach is again based on BFS and its main idea is that during the process of applying the algorithm for a node s and creating the tree, we can derive information about the lower bounds of the rest of the nodes. If we consider the node s to be the initial node, we can see that the distance $d(s, v_i)$ is i if the node v_i is reachable from v and in the level i. So, the distance $d(v_i, v_j)$ between two nodes v_i, v_j with i<j must be at least j - i. For instance, looking at Figure 2.2, the nodes v'_2 and v'_6 have distance 3(i = 1, j = 4). What follows is the equation given in the equation below (2):

$$\sum_{w \in R(s)} |d(s, w) - d(s, v)| \leq S(v), \forall v \in R(v) \tag{2}$$

$x \leftarrow$ Farn(Top[k]); // Farn and Top are global variables
Create queue Q;
Q.enqueue(v);
Mark v as visited;
$d \leftarrow 0$; $S \leftarrow 0$; $\tilde{\gamma} \leftarrow$ outdeg(v); $nd \leftarrow 1$;
while Q *is not empty* **do**
  $u \leftarrow Q$.dequeue();
  if $d(v,u) > d$ **then**
    $d \leftarrow d+1$;
    $L_d^{\mathrm{CUT}}(v, r(v)) \leftarrow \frac{(n-1)(S-\tilde{\gamma}+(d+2)(r(v)-nd))}{(r(v)-1)^2}$;
    if $L_d^{CUT}(v, r(v)) \geq x$ **then return** $+\infty$;
    $\tilde{\gamma} \leftarrow 0$
  for w *in adjacency list of* u **do**
    if w *is not visited* **then**
      $S \leftarrow S + d(v,w)$;
      $\tilde{\gamma} \leftarrow \tilde{\gamma} +$ outdeg(w);
      $nd \leftarrow nd+1$;
      Q.enqueue(w);
      Mark w as visited
    else
      // we use Remark 6.3
      $L_d^{\mathrm{CUT}}(v, r(v)) \leftarrow L_d^{\mathrm{CUT}}(v, r(v)) + \frac{(n-1)}{(r(v)-1)^2}$;
      if $L_d^{CUT}(v, r(v)) \geq x$ **then return** x;
return $\frac{S(n-1)}{(r(v)-1)^2}$;

Figure 2: BFSCut approach for updating lower bound

Now by noticing that the number of nodes with distance 1 from v are its degree value since they are its neighbors, we can say that the nodes w from the equation (2) are in distance bigger than 2. Thus, that helps us define the lower bound of S(v) such that:

$$2(\#\{w \in R(s) : |d(s,w) - d(s,v)| \leq 1\} - deg(v) - 1) + deg(v) + \\ + \sum_{w \in R(v)} |d(s,w) - d(s,v)|, |d(s,w) - d(s,v)| > 1 \quad (3)$$

By multiplying the equation (3) by $\frac{(n-1)}{(r(v)-1)^2}$, we obtain a lower bound $f(v)$ of v, named $L_s^{LB}(v, r(v))$. The complete algorithm for undirected graphs is proposed below:

In the case of directed graphs the bound is changed to (4)

$$2(\#\{w \in R(s) : |d(s,w) - d(s,v)| \leq 1\}) + \\ + \sum_{w \in R(v)} (d(s,w) - d(s,v)) - deg(v) - 2 \quad (4)$$

5 DATASETS

In this section, we are going to discuss the datasets we have chosen for our experiments. The algorithm in the source paper operates on unweighted networks, which is reflected in our dataset corpus selection. This is because of their implementation of BFS to solve the APSP problem. We have chosen primarily undirected networks, but directed ones are also used with their directionality removed. The networks were chosen to vary with respect to their degree, diameter, and average clustering coefficient, to study the effect of these factors on the performance of the Top-K closeness centrality algorithm. The networks fall into three types, which are described below.

Input : A graph $G = (V, E)$, a source node s
Output: Lower bounds $L_s^{\mathrm{LB}}(v, r(v))$ of each node $v \in R(s)$
$d \leftarrow$ BFSfrom(s);
maxD $\leftarrow \max_{v \in V} d(s,v)$;
sum$\Gamma_{\leq 0} \leftarrow 0$; sum$\Gamma_{\leq -1} \leftarrow 0$; sum$\Gamma_{>\mathsf{maxD}+1} \leftarrow 0$;
for $i = 1, 2, ...,$ maxD **do**
  $\Gamma_i \leftarrow \{w \in V : d(s,w) = i\}$;
  $\gamma_i \leftarrow \#\Gamma_i$;
  sum$\Gamma_{\leq i} \leftarrow$ sum$\Gamma_{\leq i-1} + \gamma_i$;
  sum$\Gamma_{>i} \leftarrow |V| -$ sum$\Gamma_{\leq i}$;
L(1) $\leftarrow \gamma_1 + \gamma_2 +$ sum$\Gamma_{>2} - 2$;
for $i = 2, ...,$ maxD **do**
  L(i) $\leftarrow$ L($i-1$) + sum$\Gamma_{\leq i-3} -$ sum$\Gamma_{>i+1}$;
for $i = 1, ...,$ maxD **do**
  foreach $v \in \Gamma_i$ **do**
    $L_s^{\mathrm{LB}}(v, r(v)) \leftarrow (\mathsf{L}(i) - \deg(v)) \cdot \frac{(n-1)}{(r(v)-1)^2}$;
return $L_s^{LB}(v, r(v)) \quad \forall v \in V$

Figure 3: LB approach for updating lower bound

(1) **Road networks:** Road networks are typically representations of transport networks. The analysis of traffic flow patterns[26], potential improvement points for infrastructure development, and analysis and design of better transport systems are some of the tasks usually performed on road networks. Closeness centrality is a key component to identifying major points of interaction which could potentially turn into transport hubs.
We used the **us_roads** corpus, specifically the states of Alaska, Connecticut, New Hampshire and Vermont. We also included a larger network for the US state of California [15], with a significantly larger number of nodes and edges. All road networks have lower density and average clustering coefficient, and higher diameter.
(2) **Social networks:** The ability of closeness centrality to be an indicator of how influential a node is in a network, makes it highly applicable in the context of social networks. Typical analysis would include usecases like identifying top influencers in a social network [18], or analysing the behaviour of users across commmunities [9]. In our experiments, we have used the **rt-retweet**, **digg_reply** and **email_enron** datasets.
The twitter dataset is a network of users that have been retweeted by each other. Digg is a news website, and the data consists of users who replied to other users on the discussion threads of various articles. Finally the Enron email dataset [10] is from the fallout of the Enron scandal, where top executives were implicated using their email communications. The nodes in this dataset are the senders and receivers of these emails. Finally, all three datasets were originally directed, but have been used in an undirected manner.
(3) **Collaboration networks:** Collaboration networks are a special type of social network, where the average clustering coefficient is higher than normal. Nodes in these networks tend to interact highly with each other, which is verified by the nature of the actual datasets used here. We have used the Arvix collaboration networks [13] for Condense Matter,

Astro and High Energy Physics (ca-CondMat, ca-AstroPh, ca-HepPh).
The **TerroristRel** dataset is a network of the members of the terrorist group Aum Shinrikyo, specifically which members have known relations with which other members. Finally, in the **marvel_universe** dataset, each edge is between two characters who have interacted at some time in the comics. Closeness centrality helps us narrow down origin points of influential research, or leaders in organisations of interest.

All datasets are chosen from [12], [22], [14] and The Netzschleuder network catalogue and repository

6 EXPERIMENTS AND RESULTS

6.1 Network statistics

We first analysed the network characteristics of the datasets we are using, to undestand them better. From 4, we see the distribution of nodes vs edges, as well as how the diameter, degree and average clustering coefficient are related to each other. The networks appear to have a linear relationship between nodes and edges. We also see that the road networks have a much higher diameter, while for the other networks it is near single digit. The collaboration networks have the highest average clustering coefficient. Both road as well as collaboration networks have low density as well.

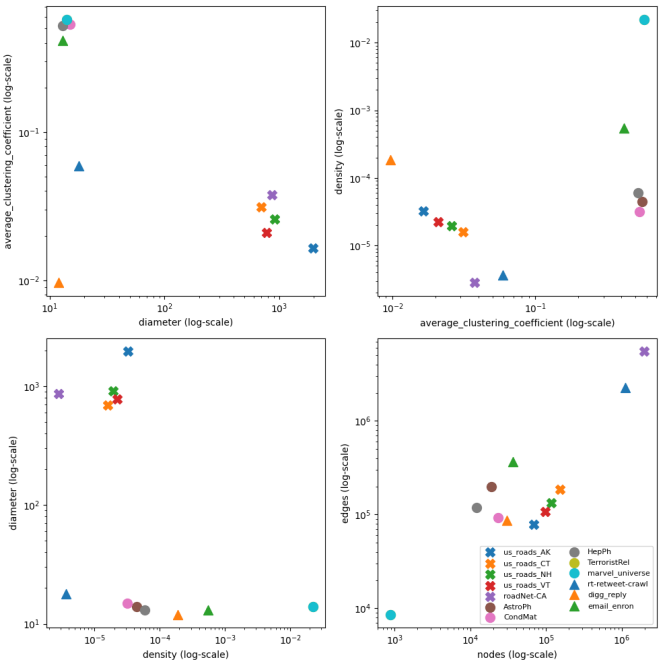


Figure 4: Dataset network inter statistics relationships

6.2 Time taken analysis

For our comparisions, we will utilize the degree initialization of the top-k closeness centrality calculation, with BFSCUT and LB variants of update process. We will also use the base closeness centrality measure, and compare all these variants to degree centrality. The

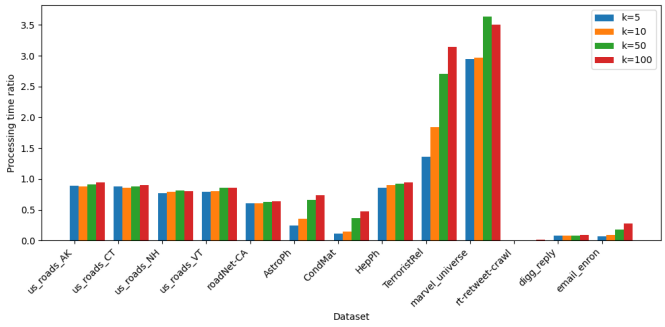


Figure 5: Ratio of top-k closeness BFSCUT to closeness centrality computation time for all datasets

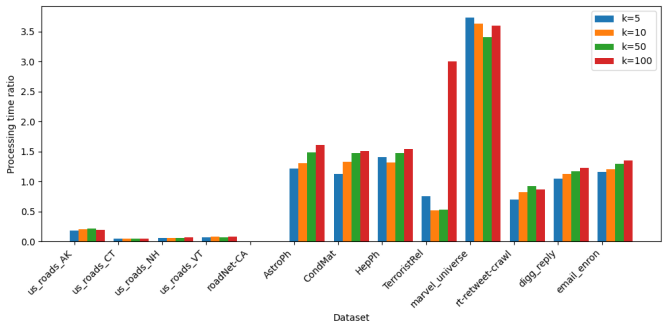
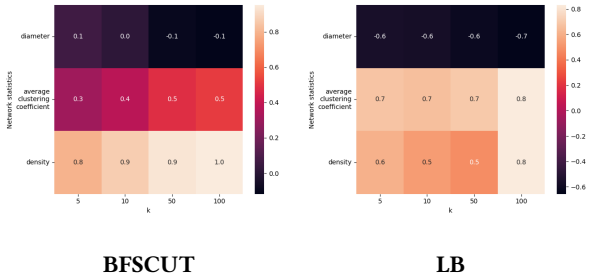


Figure 6: Ratio of top-k closeness LB to closeness centrality computation time for all datasets

first experiment we ran was to compare the time taken to compute closeness for all nodes, and top-k closeness for k nodes equal to 5, 10, 50 and 100 across all datasets (a more realistic and applicable scenario would be to test the datasets for mostly low values of k, but we wanted to see if there is a significant change in the performance by increasing the value of k a lot.) We then take the ratio of the top-k closeness variants time and the closeness centrality time, as a means of comparing how much faster or slower top-k is for a particular dataset while normalizing for varying numbers of nodes and edges.

From figure 5 and 6, we see that the LB variant is much faster for the road networks, compared to the others. We also see that the TerroristRel and marvel_universe datasets need a lot of time, compared to the others. This is attributed to their having the highest density and average clustering coefficient, requiring the BFS to work on far more neighbors per node. These observations are further confirmed from the Pearson's correlation of the time ratios to the diameter, density and average clustering coefficient across the values of k in figure 7. While BFSCUT variant shows a stronger positive correlation for computation time ratio and density, LB variant shows a strong positive correlation for both density and average clustering coefficient. The figures also show a strong negative correlation for diameter and computation time ratio, for LB variant. This tracks as the larger diameter road networks have much lower densities. While the collaboration networks have much higher densities.



BFSCUT **LB**

Figure 7: Pearson's correlation of Time ratios (BFSCUT and LB) and network statistics across k

6.3 nDCG score and KendallTau correlation

We then wanted to test our hypothesis that nodes with high degree centrality also tend to have high closeness centrality. For this, we used normalized discounted cumulative gain (nDCG) score and KendallTau rank correlation. nDCG works by comparing the ground truth ranking of items, to the rankings provided by a test system, while also weighing in the position of each item. While KendallTau rank correlation works by comparing the number of instances where the order of ranking is preserved according to the ground truth, vs when it is not. Both nDCG score and KendallTau's rank correlation are widely used for comparing node rankings [25], [27].

We first establish the similarity of node ranking between closeness centrality and the top-k variants in Figure 8. We see that the approximated BFS in top-k does not result in spurious rankings, as all three centrality measures have perfect alliance. This allows us to compare only degree and closeness centrality, simplifying the experiments. From the Pearson's correlation of nDCG scores across k and the density, diameter and average clustering coefficient of the networks, we again see a negative correlation between diameter and the nDCG score, and a positive correlation for the other two metrics. From this we surmise that the denser a network and more connected the nodes, the higher the chance that the degree and closeness centrality will be aligned. While an increased diameter works in the reverse by making a network more sparse, thus decreasing the chances of such an alignment between the two centrality measures. However the figures (8),(10) best show that such a relationship between degree and closeness centrality is highly unlikely as there are no networks that present a significant correlation.

In figure 11, we see the results of comparing the KendallTau rank correlation score of degree centrality vs closeness and BFSCUT and LB variants of top-k closeness centrality across all datasets for k=100. Figure 12 shows us the Pearson's correlation of the KendallTau rank correlations between degree and closeness centrality for each network and their density, diameter and average clustering coefficient across the various values of k. From these we see a confirmation of the highly inverse relationship between diameter and the correlation between degree and closeness centrality rankings. Average clustering coefficient too has the same positive correlation with the degree and closeness centrality rankings. These correlations increase as the value of k increases.

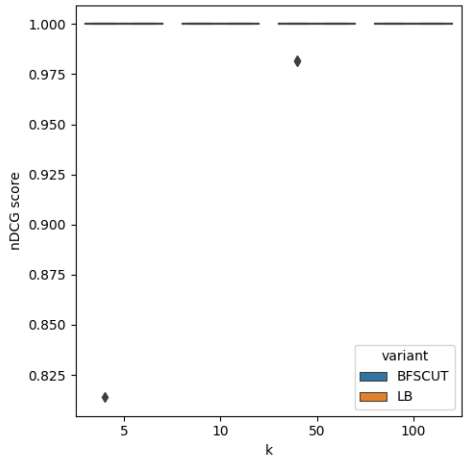


Figure 8: Comparing similarity of top-k and base closeness centrality rankings using nDCG score

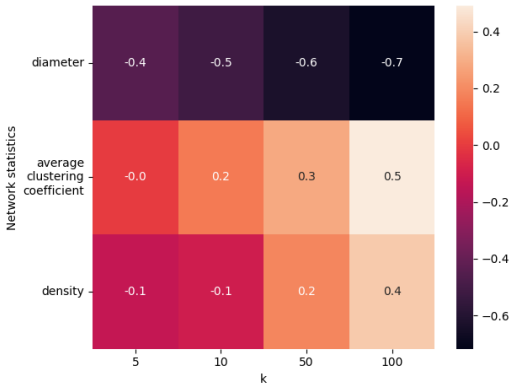


Figure 9: Pearson's correlation of nDCG scores and network statistics across k

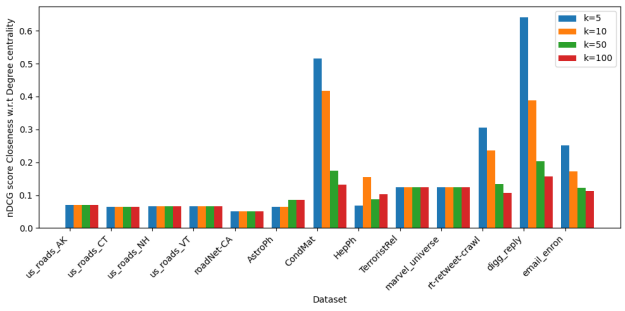


Figure 10: nDCG score of degree and closeness centrality across k and datasets

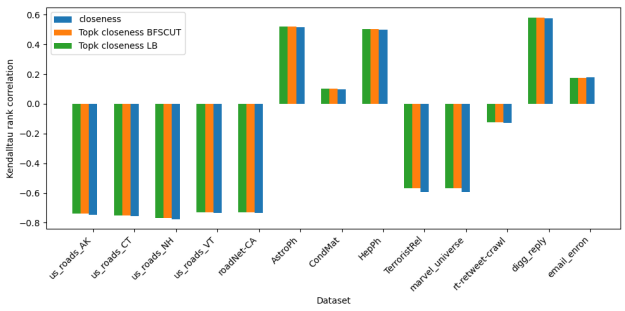


Figure 11: KendallTau rank comparison of closeness centrality variants across datasets for k=100

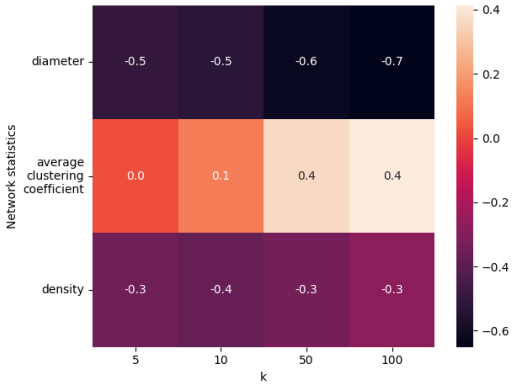


Figure 12: Pearson's correlation of KendallTau rank correlations and network statistics across k

7 CONCLUSION

In this paper we have studied the factors that affect the correlation between closeness centrality and degree centrality. We have also identified the types of networks where using the top-k closeness centrality BFSCUT and LB variants would be beneficial from a computation time perspective, compared to vanilla closeness centrality. We have shown that the computation time of BFSCUT variant of top-k closeness centrality is highly dependant on the density, and to an extent on the clustering coefficient of the network. In contrast, for the LB variant there is a highly inverse relationship between the diameter of the network and the computation time. Both density and average clustering coefficient are positively related to computation time as well. The nDCG scores and KendallTau correlation prove again that diameter has a highly inverse relationship with respect to the correlation between the degree and closeness centrality of nodes. However, our findings seem to suggest that correlation between degree and closeness centrality of nodes is non existant in regards to the datasets we experimented on, and maybe such correlation is strong in a lot larger networks.

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. [n. d.]. *Computing Top-<italic>k</italic> Closeness Centrality in Fully-dynamic Graphs.* 21–35. https://doi.org/10.1137/1.9781611975055.3 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611975055.3

[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] Shiri Chechik, Edith Cohen, and Haim Kaplan. 2015. Average distance queries through weighted samples in graphs and metric spaces: High scalability with tight statistical guarantees. *arXiv preprint arXiv:1503.08528* (2015).

[5] Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. 2012. Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications* 391, 4 (2012), 1777–1787.

[6] Edith Cohen. 2014. All-distances sketches, revisited: HIP estimators for massive graphs analysis. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 88–99.

[7] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F Werneck. 2014. Computing classic closeness centrality, at scale. In *Proceedings of the second ACM conference on Online social networks*. 37–50.

[8] Pierluigi Crescenzi, Clémence Magnien, and Andrea Marino. 2020. Finding top-k nodes for temporal closeness in large temporal graphs. *Algorithms* 13, 9 (2020), 211.

[9] 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

[10] Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *Machine Learning: ECML 2004*, Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 217–226.

[11] 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

[12] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13 Companion)*. Association for Computing Machinery, New York, NY, USA, 1343–1350. https://doi.org/10.1145/2487788.2488173

[13] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data* 1, 1 (mar 2007), 2–es. https://doi.org/10.1145/1217299.1217301

[14] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[15] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2008. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *CoRR* abs/0810.1355 (2008). arXiv:0810.1355 http://arxiv.org/abs/0810.1355

[16] Yeon-sup Lim, Daniel S Menasché, Bruno Ribeiro, Don Towsley, and Prithwish Basu. 2011. Online estimating the k central nodes of a network. In *2011 IEEE Network Science Workshop*. IEEE, 118–122.

[17] 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

[18] 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

[19] Lutz Oettershagen and Petra Mutzel. 2022. Computing top-k temporal closeness in temporal networks. *Knowledge and Information Systems* 64, 2 (2022), 507–535.

[20] 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

[21] Paul W Olsen, Alan G Labouseur, and Jeong-Hyon Hwang. 2014. Efficient top-k closeness centrality search. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 196–207.

[22] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. https://networkrepository.com

[23] Rahul Saxena and Mahipal Jadeja. 2022. Network centrality measures: Role and importance in social networks. In *Principles of Social Networking*. Springer, 29–54.
[24] Silvia M Velasquez, Martiniano M Ricardi, Christian Peter Poulsen, Ai Oikawa, Adiphol Dilokpimol, Adnan Halim, Silvina Mangano, Silvina Paola Denita Juarez, Eliana Marzol, Juan D Salgado Salter, et al. 2015. Complex regulation of prolyl-4-hydroxylases impacts root hair expansion. *Molecular plant* 8, 5 (2015), 734–746.
[25] Josh Ying, Wen-Ning Kuo, Vincent Tseng, and Hsueh-Chan Lu. 2014. Mining User Check-In Behavior with a Random Walk for Urban Point-of-Interest Recommendations. *ACM Transactions on Intelligent Systems and Technology* 5 (09 2014). https://doi.org/10.1145/2523068
[26] Yuanyuan Zhang, Xuesong Wang, Peng Zeng, and Xiaohong Chen. 2011. Centrality Characteristics of Road Network Patterns of Traffic Analysis Zones. *Transportation Research Record* 2256, 1 (2011), 16–24. https://doi.org/10.3141/2256-03 arXiv:https://doi.org/10.3141/2256-03
[27] Aybike Şimşek. 2022. Lexical sorting centrality to distinguish spreading abilities of nodes in complex networks under the Susceptible-Infectious-Recovered (SIR) model. *Journal of King Saud University - Computer and Information Sciences* 34, 8, Part A (2022), 4810–4820. https://doi.org/10.1016/j.jksuci.2021.06.010