



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Faculty and Researchers

Faculty and Researchers' Publications

---

2017-06-07

## A faster method to estimate closeness centrality ranking

Saxena, Akрати; Gera, Ralucca; Iyengar, S.R.S.

---

A. Saxena, R. Gera, S.R.S. Iyengar, "A faster method to estimate closeness centrality ranking," arXiv:1706.02083v1 [cs.SI] 7 Jun 2017.  
<http://hdl.handle.net/10945/56660>

---

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

# A Faster Method to Estimate Closeness Centrality Ranking

Akrati Saxena\*                      Raluca Gera\*\*  
 akrati.saxena@iitrpr.ac.in        rgera@nps.edu

S. R. S. Iyengar\*  
 sudarshan@iitrpr.ac.in

\*Department of Computer Science and Engineering,  
 Indian Institute of Technology Ropar, India

\*\*Department of Applied Mathematics  
 Naval Postgraduate School,  
 Monterey, CA 93943 USA

## Abstract

Closeness centrality is one way of measuring how central a node is in the given network. The closeness centrality measure assigns a centrality value to each node based on its accessibility to the whole network. In real life applications, we are mainly interested in ranking nodes based on their centrality values. The classical method to compute the rank of a node first computes the closeness centrality of all nodes and then compares them to get its rank. Its time complexity is  $O(n \cdot m + n)$ , where  $n$  represents total number of nodes, and  $m$  represents total number of edges in the network. In the present work, we propose a heuristic method to fast estimate the closeness rank of a node in  $O(\alpha \cdot m)$  time complexity, where  $\alpha = 3$ . We also propose an extended improved method using uniform sampling technique. This method better estimates the rank and it has the time complexity  $O(\alpha \cdot m)$ , where  $\alpha \approx 10 - 100$ . This is an excellent improvement over the classical centrality ranking method. The efficiency of the

proposed methods is verified on real world scale-free social networks using absolute and weighted error functions.

## 1 Introduction

In network science, various centrality measures have been defined to identify important nodes based on the application context. As an example, one can consider identifying the location within a city where to place a new public service, so that it is easily accessible for everyone. Similarly, identifying central people that have ideal social network location for the purpose of information dissemination or network influence. In such kind of applications, the nodes who can access the entire network faster need to be selected. To capture this specific property of the reachability of nodes to the entire network, researchers have defined closeness centrality.

The closeness centrality of a node  $u$  is defined as  $C(u) = \frac{n-1}{\sum_v d(u,v)}$ , where  $n$  is total number of nodes and  $d(u,v)$  is the shortest distance between two nodes  $u$  and  $v$  [1]. It represents the closeness of a node to all other nodes in the given network. The closeness centrality of a node can be computed by executing breadth first traversal (BFT) [2] from the respective node. The time complexity to compute the closeness centrality of a node is  $O(m)$ , where  $m$  represents total number of edges in the network. The proposed definition of closeness centrality [1] is only applicable for the connected networks. In the present work, we have considered real world scale-free connected networks.

The proposed closeness centrality measure assigns a centrality value to each node. But in real life applications, we are mainly interested in the relative importance of the node rather than its centrality value. This can be measured using the closeness centrality rank of the node based on the computed closeness centrality values of all nodes. The classical method to compute the closeness rank of a node has two steps: (1) calculate the closeness centrality values of all nodes, and (2) compare these values to determine the closeness rank of the node. The time complexity of the first step is  $O(n \cdot m)$  as it computes the closeness centrality of all nodes, and for the second step, it is  $O(n)$  as it compares the centrality value of the given node with all other nodes. So, the overall time complexity of this process is  $O(n \cdot m + n) = O(n \cdot m)$  that is very high, given that the entire network is required. This method is not feasible for large real world networks because of its high time complexity and dynamic characteristics of the networks.

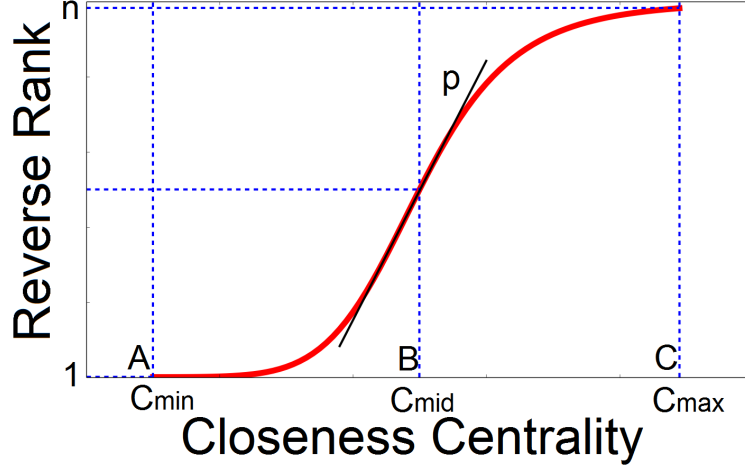


Figure 1: Reverse Rank versus Closeness Centrality

Real world complex networks such as WWW network, online social networks, collaboration networks, communications networks, are growing very fast with time. So, the computation of different centrality measures based on the global structure of the network is very high. Current literature focuses mainly on proposing approximation methods to compute closeness centrality measure. Even if we apply these approximation methods, we need to approximate closeness centrality values of all nodes to estimate the rank of a node, and its complexity is still high. Besides the size, there are other issues like real world networks are highly dynamic in nature. In these dynamic networks, the importance of different nodes keeps changing with time. So, to compute the rank of a node, we need the current complete snapshot of the network. Due to the large size of the network, it may not be feasible to download the entire network, store it, and process it.

In the present work, we propose heuristic and randomized heuristic methods to fast estimate the closeness centrality rank of a node. The complexity of the proposed methods is  $O(m)$  that is a great improvement over the classical ranking method that has the complexity  $O(nm)$ . The proposed methods exploit the structural properties of the network to reduce the time complexity of closeness ranking. For example, the nodes in the center of a network have high closeness values, while the ones in the periphery have the least closeness values. Closeness values of the middle layer nodes increase sharply

from periphery to center. Due to this unique behavior of closeness centrality in real world scale-free social networks, we observe that the reverse ranking versus closeness centrality follows a sigmoid curve as shown in Figure 1. In reverse ranking, the node having the lowest closeness value has the highest rank 1 and the node having the highest closeness value has the least rank. This sigmoid curve helps to estimate the closeness rank of a node without computing the closeness values of all nodes.

The proposed methods are simulated on different types of real world networks, such as online social networks, collaboration network, communication network, and so on. The accuracy of the proposed methods is measured using absolute and weighted error functions. Results show that the proposed method can be used efficiently for large size dynamic networks.

The main contributions of the paper are as follows:

- We study the characteristics of closeness centrality and their dependency on the structural properties of the real world scale-free social networks.
- We propose methods to estimate the closeness centrality rank of a node in  $O(m)$  time.
- The proposed methods are simulated on real world networks and their efficiency is verified using absolute and weighted error functions.

As per the best of our knowledge, this is the first work in this direction. The rest of this paper is organized as follows. In the next section, we discuss the brief literature on closeness centrality. Section 3 and 4 explain datasets and notation used in the paper respectively. In section 5, we study the behavior of closeness centrality. These observations help to construct the closeness ranking estimation method. In section 6 and 7, we propose methods to estimate the closeness rank and discuss their complexity analysis. In section 8, the simulation results are discussed. The paper is concluded in section 9. The proposed work further opens up various future directions that are discussed in the last section.

## 2 Related Work

Closeness centrality denotes reachability of a node to the given network. In undirected and unweighted networks, the reachability of two nodes only con-

siders the minimum number of hops to reach from one node to another. But in other types of networks like weighted networks, directed networks, the link weights and directions also affects the distance between two nodes. So, the closeness centrality has been extended to these networks like weighted networks [3], directed networks [4], disconnected networks [5], multilayer networks [6], overlapped community structure [7], and so on.

The complexity to compute the closeness centrality in large scale networks is very high. It has attracted the researchers towards the following problems related to measuring the closeness centrality of a node:

1. Update closeness centrality in dynamic networks,
2. Approximation algorithms for closeness centrality,
3. Identify top- $k$  nodes,
4. Other works like the computation of closeness centrality in parallel or distributed environment, its correlation with other centrality measures, and so on.

Real world networks are highly dynamic and the computation of closeness centrality of all nodes for each change in the networks will be a cumbersome task. In dynamic networks, for each update, the closeness centrality of some nodes may remain unaffected. Kas et al. proposed a method to update closeness centrality in dynamic networks [8]. The proposed method uses the set of affected nodes to update the closeness centrality whenever there is any addition, removal, or modification of nodes or edges. Yen also proposed an algorithm called CENDY (Closeness centrality and avErage path leNght in DYnamic networks) to update closeness centrality whenever an edge is updated [9]. Sariyuce et al. proposed a method to update closeness centrality using the level difference information of breadth first traversal [10].

The classical method to compute closeness centrality of a node requires the entire network and it is costly for big networks. There are some works that provide approximation methods to compute it fast. Cohen et al. proposed a sampling based method to approximate closeness centrality in directed and undirected networks [11]. Eppstein et al. proposed a randomized approximation algorithm with time complexity  $O(\frac{\log n}{\epsilon^2} \cdot m)$  to approximate the closeness centrality within an additive error of  $\epsilon \cdot \Delta$  with high probability, where  $\Delta$  is the diameter of the network [12]. Rattigan used the concept of network

structure index (NSI) to approximate the values of different centrality measures that are based on the shortest paths in the given network [13]. Some other approximation methods for closeness centrality include [14–16].

Most of the real life applications focus on identifying few top nodes having the highest closeness centrality. Okamoto et al. proposed a method to rank  $k$  highest closeness centrality nodes using a hybrid of approximate and exact algorithms [17]. Ufimtsev proposed an algorithm to identify high closeness centrality nodes using group testing [18]. Olsen et al. presented an efficient technique to find  $k$  most central nodes based on closeness centrality [19]. They used intermediate results of centrality computation to minimize the computation time. Bergamini et al. proposed a faster method to identify top- $k$  nodes in undirected networks by approximating the upper bound on closeness centrality using BFT [20].

Wehmuth et al. studied the correlation of closeness centrality with the local neighborhood volume of the node [21]. The ranking based on local neighborhood volume is named as DACCER (Distributed Assessment of the Closeness CEnterality Ranking) and is highly correlated with closeness centrality ranking in both real world and synthetic networks.

Bader et al. proposed parallel algorithm to compute closeness centrality, where it executes a breadth first traversal (BFT) from each vertex as a root [22]. Lehmann and Kaufmann proposed a method for decentralized computation of closeness centrality [23]. Wang et al. proposed a distributed algorithm that estimates closeness centrality with 91% accuracy in terms of ordering on random geometric, Erdős-Rényi, and Barabási-Albert graphs [24].

Closeness centrality has been applied to study various research topics like collaboration networks [25,26], brain network [27], community detection [28], identification of the community of a node by using the community information of other nodes [29], closeness preferential attachment (CPN) model to generate synthetic networks [30], and so on.

Currently, there is no work to estimate the rank of a node using its closeness centrality. In our previous works, we proposed methods to estimate the degree rank of a node using its local information. First, we proposed a method based on the power law degree distribution of scale-free networks and it computes the degree rank of a node in  $O(1)$  time [31,32]. We further compute the variance in the rank estimation using power law degree distribution [33]. Next, we propose sampling based methods to estimate the degree rank in scale-free and random networks [34]. These works focus on directly estimating the rank of a node based on the centrality measure. In

the present work, we rank the nodes based on closeness centrality that itself is a global centrality measure.

### 3 Datasets

We have studied structural properties of the closeness centrality using real world network datasets that are briefly explained in Table 1.

Table 1: Datasets

Network	Type	#Nodes	#Edges	Ref
Brightkite	Social Network	56739	212945	[35]
DBLP	Co-authorship Network	317080	1049866	[36]
Digg	Social Network	261489	1536577	[37]
Enron	Communication Network	84384	295889	[38]
Epinion	Social Network	75877	405739	[39]
Facebook	Social Network	63392	816831	[40]
Gowalla	Social Network	196591	950327	[35]
Google+	Social Network	107614	12238285	[41]
Slashdot	Social Network	82168	504230	[42]
Twitter	Social Network	81306	1342296	[41]

### 4 Terminologies

Table 2 explains the terminologies used in the paper.  $G(V, E)$  represents a network where  $V$  is the set of nodes and  $E$  is the set of edges.

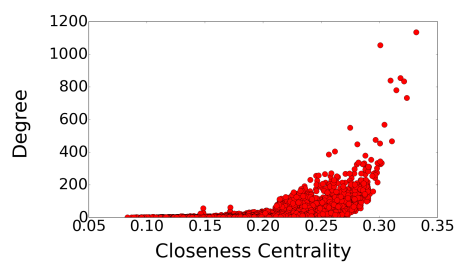
### 5 Closeness Centrality Behavior

In this section, we study the behavioral characteristics of closeness centrality and their correlation with the network structure.

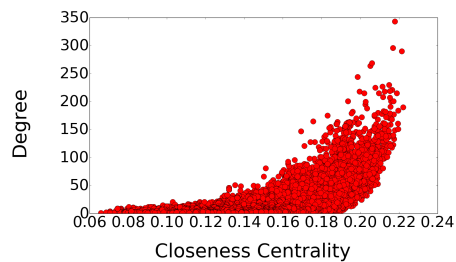
#### 5.1 Closeness Centrality vs. Degree Centrality

First, we study the correlation between closeness centrality and degree of the nodes. Results do not show that the closeness centrality is correlated with

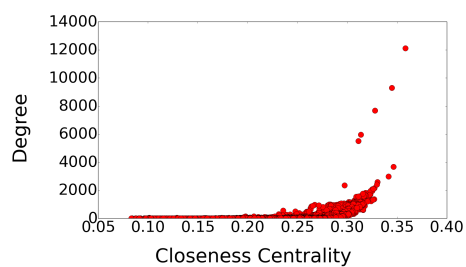




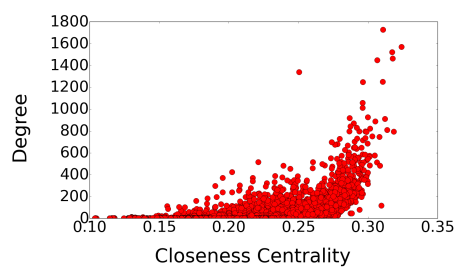
(a) Brightkite



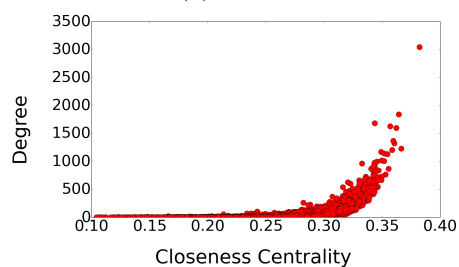
(b) DBLP



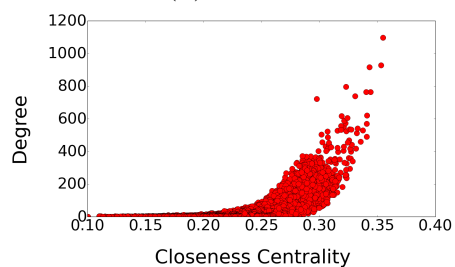
(c) Digg



(d) Enron



(e) Epinion



(f) Facebook

Figure 2: Degree versus Closeness Centrality

Table 2: Terminologies for the paper

Notation	Description
$n$	Total number of nodes in the network
$m$	Total number of edges in the network
$u, v, w$	Nodes in the network
$deg(u)$	Degree of node $u$
$C(u)$	Closeness centrality of node $u$
$c_{max}$	Maximum closeness centrality in the network
$c_{min}$	Minimum closeness centrality in the network
$c_{mid}$	Closeness centrality of middle ranked node in the network
$c'_{max}$	Estimated maximum closeness centrality in the network
$c'_{min}$	Estimated minimum closeness centrality in the network
$c'_{mid}$	Estimated closeness centrality of middle ranked node
$R_{rev}(u)$	Reverse rank of node $u$ in the network
$R_{act}(u)$	Actual rank of node $u$ in the network
$R_{est}(u)$	Estimated rank of node $u$ in the network

the degree, but they show that the node having the highest degree either has the highest closeness centrality or it is very close to the highest closeness centrality as shown in Figure 2. This is desirable since the degree of a node is a local characteristic and it can be computed in  $O(1)$  time. This information can be used to identify the node having highest closeness centrality. Further details are explained in Section 6.

## 5.2 Closeness Centrality Pattern from Center to Periphery

In scale-free networks, central nodes have the highest closeness centrality and the extreme periphery nodes have minimum closeness centrality. As we move from the center to the periphery closeness centrality of the nodes decreases. To analyze this pattern in depth, we execute breadth first traversal from the central node (having the highest closeness centrality) until all nodes are traversed. The plots of closeness centrality of nodes versus their distance from the center node are shown in Figure 3. Results show that the nodes farthest from the central nodes have minimum closeness centrality, and the

outermost level of the BFT (also referred as the outermost periphery) is very sparse. This behavior of the closeness centrality can be used to find the minimum closeness centrality nodes in the network, by looking at the diametrically opposed nodes from the central ones.

### 5.3 Closeness Centrality vs. Reverse Ranking

Real world networks have a dense central region that contains a very few number of nodes having the highest closeness centrality. These central nodes are highly connected with each other and also with rest of the network. The nodes belonging to the outermost peripheral layers have the least closeness values. Closeness centrality of all other nodes lies between this range and increases sharply as we move from the periphery to the center.

Due to this behavior, the reverse ranking versus closeness centrality of nodes follows a sigmoid curve as shown in Figure 1. In reverse ranking, a node having highest closeness value will have the smallest rank  $n$  (where  $n$  is the total number of nodes) and the node having the lowest closeness value will have the rank 1. We plot reverse rank versus closeness centrality for more than 20 real world social networks and find that they follow the sigmoid curve and it is symmetric in most of these networks. The results are displayed in Figure 4 for some of the datasets, where the  $x$ -axis represents closeness centrality and the  $y$ -axis represents the reverse ranking of the nodes.

We study this curve in depth and find that the 4-parameter logistic equation can better fit the curve. The equation of the 4-parameter logistic is defined as,

$$R_{rev}(u) = n + \frac{1 - n}{1 + \left(\frac{C(u)}{c_{mid}}\right)^p}, \quad (1)$$

where,  $c_{mid}$  represents closeness centrality of the middle ranked node in the network,  $n$  represents total number of nodes, and  $p$  denotes slope of the logistic curve at the middle point (also called hill's slope). All parameters are displayed in Figure 1. We will use this logistic equation to estimate the closeness rank of a node as explained in the next section.

## 6 The Heuristic Method for Closeness Ranking

In this section, we propose a method to estimate the closeness rank of a node. The node of interest whose rank we compute is mentioned as *interested node*. The proposed method exploits the structural characteristics of the closeness centrality to efficiently estimate the ranking. As we discussed, the reverse rank versus closeness centrality follows a sigmoid curve, we use this characteristic to estimate the rank of a node. Once we estimate both parameters of the logistic Equation 1, the closeness rank of a node can be estimated in  $O(1)$  time.

We will now discuss methods to estimate both of these parameters: (1) closeness centrality of the middle-ranked node ( $c_{mid}$ ) and (2) slope of the logistic curve ( $p$ ). Next, we will discuss the method to estimate closeness rank of a node and its complexity.

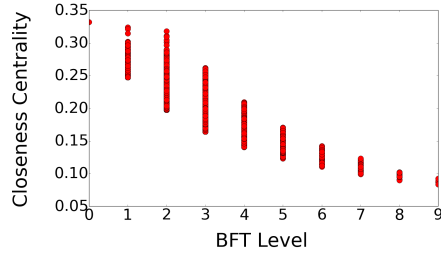
### 6.1 Estimate closeness centrality of middle ranked node ( $c_{mid}$ )

We observed that in most of the real world networks, reverse rank versus closeness centrality follows a symmetric sigmoid pattern. The plots for some of these networks are shown in Figure 4, where the plots (a) – (d) are symmetric. We use this information to compute the value of  $c_{mid}$ , using the network.

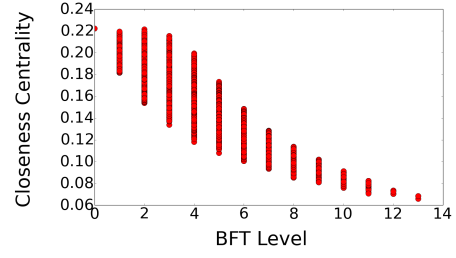
Using the property 5.1, the maximum closeness centrality can be estimated as,  $c'_{max} = C(u)$  where  $deg(u) \geq deg(v), \forall v \in V$ . While estimating the closeness centrality of the *interested node*, keep track of the node having the highest degree. Once the highest degree node is known, we can compute its closeness centrality using the standard computation method. We thus have the following observation.

**Observation 1.** *The maximum closeness centrality can be estimated as,  $c'_{max} = C(u)$  where  $deg(u) \geq deg(v), \forall v \in V(G)$ .*

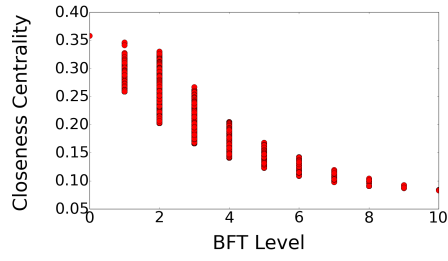
In a network, the nodes having the maximum distance from the central node have minimum closeness centrality as observed in figure 3. So, we can keep track of the nodes falling on the outermost level of BFT while computing the maximum closeness centrality. Let  $w$  be a node in the network chosen



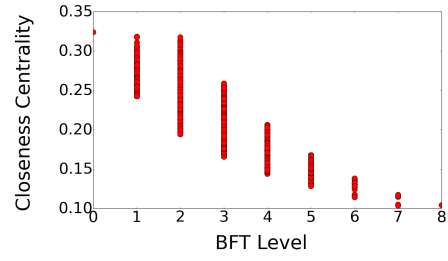
(a) Brightkite



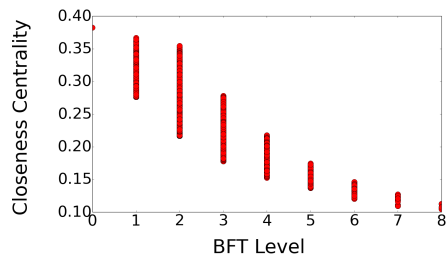
(b) DBLP



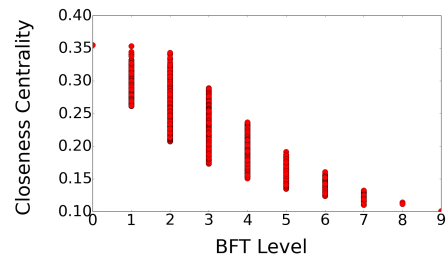
(c) Digg



(d) Enron



(e) Epinion



(f) Facebook

Figure 3: Closeness Centrality versus BFT Level from the Central Node

uniformly at random from all the nodes farthest away from  $u$  (i.e.  $d(u, w)$  is maximum) for  $u$  identified as a central node in Observation 1. Using the property 5.2, the minimum closeness centrality can be estimated by the closeness centrality of  $w$ . We thus have the following observation.

**Observation 2.** *The minimum closeness centrality can be estimated as,  $c'_{min} = C(w)$ ,  $\exists w$  where  $d(w, u)$  is max, for  $u$  identified in Observation 1.*

We now use Observations 1 and 2 to estimate the closeness centrality of the middle ranked node.

**Proposition 1.** *In the symmetric sigmoid curve of reverse rank versus closeness centrality, the closeness centrality of the middle-ranked node, ( $c_{mid}$ ), can be computed as  $c_{mid} = (c_{max} + c_{min})/2$ .*

*Proof.* If the sigmoid curve is symmetric, then using Figure 1 we note that:  
The distance from  $C$  to  $A = c_{max} - c_{min}$   
The distance from  $A$  to  $B = (c_{max} - c_{min})/2$   
The distance of  $B$  from the origin point can be computed as,

$$c_{mid} = c_{min} + (c_{max} - c_{min})/2 = (c_{min} + c_{max})/2, \quad (2)$$

as desired.  $\square$

## 6.2 Estimate Slope of the Sigmoid Curve ( $p$ )

We measured the slope of the sigmoid curve for 20 real world networks using scaled levenberg-marquardt algorithm [43] with 1000 iterations and 0.0001 tolerance. We observed that the slope ranges from 10-15. The slope for the discussed datasets is shown in Table 3. The average of these values is used as the value for  $p$  in the simulation. We empirically observed that the slight variation in the estimation of  $p$  does not cause more error in the ranking.

## 6.3 Estimate Closeness Rank

After estimating all the needed parameters for the sigmoid curve, the closeness rank of the *interested node*  $u$  can be estimated using Proposition 2.

**Proposition 2.** *In a network  $G$ , the closeness rank of a node  $u$  can be computed as,  $R_{act}(u) = 1 + \frac{n-1}{1 + \left(\frac{C_G(u)}{c_{mid}}\right)^p}$ .*

Table 3: Networks versus their p values

Network	p value	Network	p value
Brightkit	12.18	Gowalla	10.79
DBLP	14.11	Google+	15.95
Digg	14.23	Facebook	12.74
Enron	11.47	Twitter	14.47
Epinion	12.99	Slashdot	14.89
		<b>Average</b>	13.38

*Proof.* Using Equation 1, the reverse rank of a node  $u$  can be computed as,

$$R_{rev}(u) = n + \frac{1-n}{1+\left(\frac{C(u)}{c_{mid}}\right)^p}.$$

The actual rank of a node can be estimated by subtracting its reverse rank from the total number of nodes plus 1. We thus have that,

$$\begin{aligned} R_{act}(u) &= n - R_{rev}(u) + 1 \\ R_{act}(u) &= n - n - \frac{1-n}{1+\left(\frac{C(u)}{c_{mid}}\right)^p} + 1, \end{aligned}$$

$$R_{act}(u) = 1 + \frac{n-1}{1+\left(\frac{C(u)}{c_{mid}}\right)^p}, \quad (3)$$

as desired.  $\square$

Thus, we can now estimate the rank of a nodes in a network  $G$  using Corollary 1.

**Corollary 1.** *In a network  $G$ , the closeness rank of a node  $u$  can be estimated as,  $R_{est}(u) = 1 + \frac{n-1}{1+\left(\frac{C(u)}{c'_{mid}}\right)^{p'}}$ , where  $c'_{mid}$  and  $p'$  are the estimated values of closeness centrality of middle ranked node and slope of the sigmoidal closeness centrality curve respectively.*

The combined method to estimate the closeness rank of a node is explained in Algorithm 1. Here,  $closeness\_centrality(G, u)$  method returns

the closeness centrality of node  $u$ . *closeness centrality1*( $G, u$ ) method returns the closeness centrality  $C(u)$  of node  $u$ , the node  $w$  having the highest degree in the network, and the network size  $n$ . *closeness centrality2*( $G, w$ ) method returns the closeness centrality of node  $w$ , and list of the nodes having maximum distance from the node  $w$ . *random\_choice*(*cmin\_list*) function returns a value uniformly at random from the given list *cmin\_list*. *closeness centrality*( $G, u$ ), *closeness centrality1*( $G, u$ ), and *closeness centrality2*( $G, u$ ) methods can be implemented by modifying the BFT algorithm as they just need to keep track of few variables.

---

**Algorithm 1:** *EstimateClosenessRank*( $G, u, p$ )

---

- 1  $(C(u), w, n) = \text{closeness\_centrality1}(G, u);$
  - 2  $(c'_{max}, \text{cmin\_list}) = \text{closeness\_centrality2}(G, w);$
  - 3  $c'_{min} = \text{closeness\_centrality}(G, \text{random\_choice}(\text{cmin\_list}));$
  - 4  $c'_{mid} = (c'_{max} + c'_{min})/2;$
  - 5 Estimate closeness rank of the node using equation 3 as,  

$$R_{est}(u) = 1 + \frac{n-1}{1 + \left(\frac{C(u)}{c'_{mid}}\right)^p};$$
  - 6 Return  $R_{est}(u);$
- 

## 6.4 Complexity Analysis

In this section, we will discuss the time complexity of the proposed heuristic method that is explained in Algorithm 1. The complexity of step 1 is  $O(m)$  as it executes one BFT and keeps track of the highest degree node while executing the BFT. The time complexity of step 2 is  $O(m)$  as it executes BFT from the node  $w$  and returns the list of nodes that are traversed during the last level of BFT. The time complexity of step 3 is  $O(m)$ , as we assume that *random\_choice*(*cmin\_list*) function returns a value in constant time as the size of the list is very small. Step 4 and 5 take  $O(1)$  time. So, the overall complexity of the proposed method is  $O(m) + O(m) + O(m) + 2 \cdot O(1) = O(m)$ . This is a great improvement over the classical ranking method that takes  $O(n \cdot m)$  time.



## 7 The Randomized Heuristic Method

We observed that the accuracy of the estimated rank highly depends on the accuracy of the  $c_{mid}$  estimator. So, we propose an improved  $c_{mid}$  estimator to increase the accuracy of the proposed method.

In the heuristic method, we assumed that the sigmoid curve is symmetric. But in some real world networks, the sigmoid curve might not be symmetric, but very close as seen in Figure 4 (e), (f), and the heuristic method will give a huge error for such cases. So, we propose an improved randomized method that uses uniformly random samples to estimate the value of  $c_{mid}$ . The improved method picks  $k$  nodes uniformly at random and computes their closeness centrality values. The average of these  $k$  closeness values is used as the estimated value of  $c_{mid}$ . Results show that the estimated value of  $c_{mid}$  is very close to its actual value. Further details are explained in the Results section. The complete algorithm is explained in Algorithm 2.

---

**Algorithm 2:** *RandomizedClosenessRank*( $G, u, p, k$ )

---

```

1  $(C(u), n) = \text{closeness\_centrality3}(G, u);$ 
2 Take a list  $L$ ;
3 for  $i \leftarrow 1$  to  $k$  do
4   | Select a node  $w$  uniformly at random;
5   | Add  $\text{closeness\_centrality}(G, w)$  in list  $L$ ;
6  $c'_{mid} = \text{average of all values of } L;$ 
7 Estimate closeness rank of the node using equation 3 as,
   
$$R_{est}(u) = 1 + \frac{n-1}{1 + \left(\frac{C(u)}{c'_{mid}}\right)^p};$$

8 Return  $R_{est}(u);$ 
```

---

### 7.1 Complexity Analysis

In *RandomizedClosenessRank*( $G, u, p, k$ ) algorithm, the complexity of *closeness\\_centrality3*( $G, u$ ) method is  $O(m)$  as it returns the closeness centrality of node  $u$  and the total number of nodes. In the *for loop*,  $k$  nodes are chosen uniformly at random and their closeness centrality is computed. So, the complexity of the for loop is  $O(k \cdot m)$ . Complexity of step 6 and 7 is  $O(1)$ . Thus,

the overall complexity of the proposed method is  $O(m) + O(k \cdot m) + O(1) = O((k + 1)m)$ . As  $k \ll n$ , the complexity of the proposed method is  $O(m)$ .

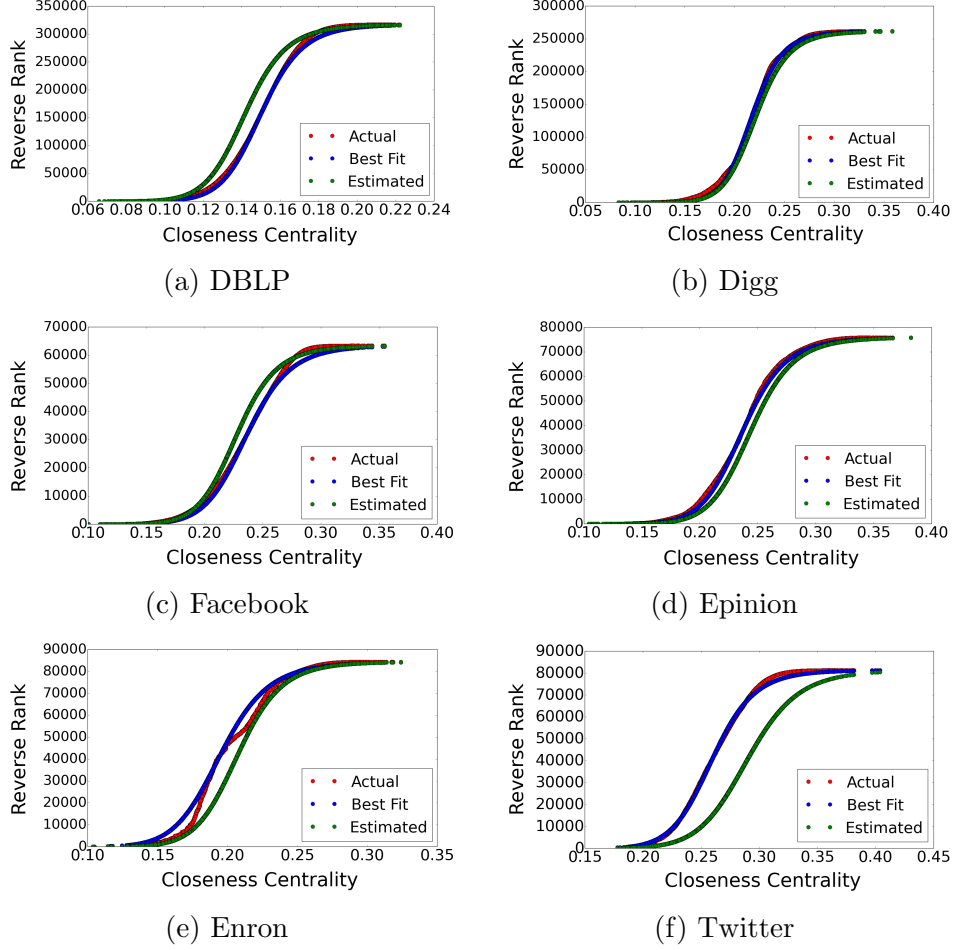


Figure 4: Reverse Rank versus Closeness Centrality

## 8 Simulation Results

In this section, we discuss error functions and simulation results on real world datasets.

## 8.1 Error Functions

The accuracy of the proposed methods is computed using absolute and weighted error functions that are discussed below:

1. **Absolute Error:** Absolute error for a node  $u$  is computed as,

$$Err_{abs}(u) = |R_{est}(u) - R_{act}(u)|.$$

The **percentage average absolute error** can be computed as

$$Err_{paae} = \frac{\text{average absolute error}}{\text{network size}} \cdot 100\%.$$

where the *average absolute error* is computed by taking the average of absolute error for each node.

2. **Weighted Error:** In real life applications, importance of the rank depends on where does a node stand and how many total nodes are there. If you get 100 rank, it is admirable if there are 100,000 people but it is not good enough if there are only 500 people. So, the same rank value can be more important in a larger network than in a smaller one. So, the impact of the error depends on percentile of the node as well as on the network size. The proposed weighted error function considers both of these parameters and it is defined as,

$$Err_{wtd}(u) = \frac{Err_{abs}(u)}{n} \times percentile(u)\%.$$

The percentile of a node  $u$  can be calculated as,  $percentile(u) = \frac{n - R_{act}(u) + 1}{n} \times 100$ . The weighted error increases linearly with the percentile of the node, and decreases with the network size.

## 8.2 Discussion

The proposed methods are simulated on all datasets discussed in Table 1. To measure the accuracy of the proposed methods, the absolute and weighted errors are computed for each node. Then it is averaged over all nodes to compute the overall error in the proposed methods. The errors of the proposed methods are shown in Table 4.

Table 4: Error in the Estimated Ranking

Network	Best Fit Error		Heuristic Method Error		Rand. Heuristic Method Error (k=50)	
	Paae	Wtd	Paae	Wtd	Paae	Wtd
Brightkite	1.48	0.79	7.96	3.89	2.96	1.32
DBLP	1.29	0.68	9.47	4.38	3.29	1.91
Digg	1.24	0.49	4.50	2.48	3.60	1.94
Enron	4.14	1.81	8.56	3.92	5.64	2.94
Epinion	1.42	0.70	7.72	3.96	2.16	1.00
Facebook	1.72	0.94	6.68	3.54	3.64	1.92
Google+	1.44	0.64	20.57	12.22	3.70	1.77
Gowalla	1.52	0.81	17.96	9.30	3.97	1.94
Slashdot	0.78	0.29	6.29	2.69	2.91	1.47
Twitter	0.98	0.59	24.64	14.70	3.41	1.96

The best fit error is computed by using the best-fit logistic curve on the reverse rank versus closeness centrality curve. The parameters of the best fit curve are computed using scaled levenberg-marquardt method with 1000 iterations and 0.0001 tolerance [43]. Once the parameters of the best fit curve are computed, the Equation 3 is used to compute the rank of a node. Results show that the error computed using best-fit parameters is very low, and the sigmoid closeness pattern can be efficiently used to estimate the rank of the nodes.

The error for heuristic and randomized heuristic methods is shown in Table 4 for  $p = 13.38$ . The error varies with the  $p$  value. The reverse rank versus closeness centrality plots for the best fit and approximated parameters are shown in Figure 4. The heuristic method gives a high error on some of the real world networks due to the error in the estimation of the parameters of the logistic curve or if the curve is not smooth. The complexity to estimate the closeness rank of a node is same as computing its closeness centrality; this is a great improvement over the classical ranking method.

Next, we show that the improved randomized heuristic method gives a great improvement over the heuristic method. To compute the error, each experiment is repeated 40 times for  $k = 50$ , and the average of the errors is shown in Table 4.

Thus, the results show that the sigmoid behavior of closeness centrality

can be used to fast estimate the closeness rank of a node. The proposed methods can be efficiently applied on real world online networks as the proposed methods do not need to store the network. Their APIs can be used to run the BFT and compute the closeness centrality.

## 9 Conclusion

In the present work, we studied behavior of the closeness centrality and its correlation with the structural properties of the network on real world scale-free social networks. We observed that the reverse ranking versus closeness centrality follows a sigmoid pattern. We further analyze how does the closeness centrality of nodes change as we move from the central region to the periphery. These unique characteristics of closeness centrality are used to propose heuristic method for closeness ranking of a node. The complexity of the proposed method is  $O(m)$  that is a great improvement over the classical ranking method that takes  $O(n \cdot m)$  time.

The proposed method is further improved using uniformly random samples where the closeness centrality values of  $k$  sampled nodes are used to estimate a parameter ( $c_{mid}$ ) of the sigmoid curve. The complexity of the improved method is  $O(k \cdot m) \approx O(m)$  as ( $k \ll n$ ). The accuracy of the proposed methods is verified using absolute and weighted error functions. Results show that the proposed methods can be efficiently used to estimate the closeness rank of a node.

## 10 Future Directions

In the proposed methods, we have estimated the slope of the sigmoid curve as an average of the slopes observed in real world networks. The slope of the curve denotes how sharply the closeness centrality increases for all middle layered nodes from periphery to central region. It depends on the density of the network and how the density changes from periphery to the center.

We study the correlation of the slope of the sigmoid curve ( $p$ ) with the density of the network. In figure 5, we plot slope versus density for BA networks [44] having 40,000 nodes. We observe that the slope increases with the density, reaches its maximum value, and then it further decreases with the increase in the density. This correlation can be used to propose an estimator

for  $p$  value.

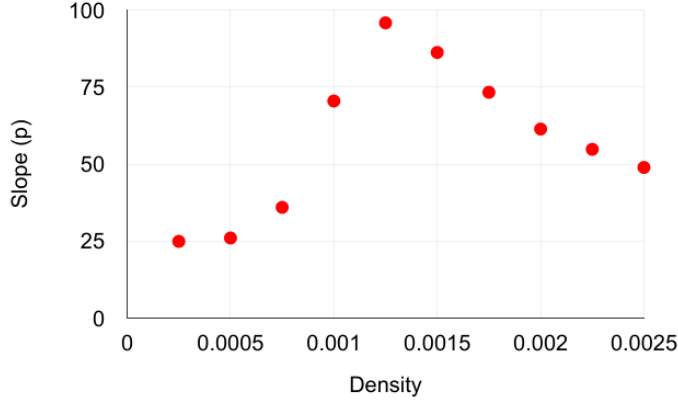


Figure 5: Slope vs. Density on BA networks.

The proposed methods can be further improved by estimating the closeness centrality of a node using its local information without having the entire network. These methods will be highly required due to the fast growth of the networks. This will improve the time complexity of the proposed methods many folds. The proposed methods can also be extended to other types of networks like weighted networks, directed networks, hypergraphs, and so on.

The rank estimation of a node based on other centrality measures like betweenness centrality, coreness, PageRank, is still an open problem.

## References

- [1] L. C. Freeman, Centrality in social networks conceptual clarification, *Social networks* 1 (3) (1978) 215–239.
- [2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to algorithms*, Vol. 6, MIT press Cambridge, 2001.
- [3] N. Ruslan, S. Sharif, Improved closeness centrality using arithmetic mean approach, in: *INNOVATION AND ANALYTICS CONFERENCE AND EXHIBITION (IACE 2015): Proceedings of the 2nd Innovation*

and Analytics Conference & Exhibition, Vol. 1691, AIP Publishing, 2015, p. 050022.

- [4] Y. Du, C. Gao, X. Chen, Y. Hu, R. Sadiq, Y. Deng, A new closeness centrality measure via effective distance in complex networks, *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25 (3) (2015) 033112.
- [5] Y. Rochat, Closeness centrality extended to unconnected graphs: The harmonic centrality index, in: *ASNA*, no. EPFL-CONF-200525, 2009.
- [6] F. Barzinpour, B. H. Ali-Ahmadi, S. Alizadeh, S. G. Jalali Naini, Clustering networks heterogeneous data in defining a comprehensive closeness centrality index, *Mathematical Problems in Engineering* 2014.
- [7] M. K. Tarkowski, P. Szczepański, T. Rahwan, T. P. Michalak, M. Wooldridge, Closeness Centrality for Networks with Overlapping Community Structure, in: *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [8] M. Kas, M. Wachs, K. M. Carley, L. R. Carley, Incremental algorithm for updating betweenness centrality in dynamically growing networks, in: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ACM, 2013, pp. 33–40.
- [9] C.-C. Yen, M.-Y. Yeh, M.-S. Chen, An efficient approach to updating closeness centrality and average path length in dynamic networks, in: *Data Mining (ICDM)*, 2013 IEEE 13th International Conference on, IEEE, 2013, pp. 867–876.
- [10] A. E. Sariyuce, K. Kaya, E. Saule, U. V. Catalyurek, Incremental algorithms for network management and analysis based on closeness centrality, *arXiv preprint arXiv:1303.0422*.
- [11] D. Delling, R. Werneck, Computing Classic Closeness Centrality, at *Scale*.
- [12] D. Eppstein, J. Wang, Fast approximation of centrality., *J. Graph Algorithms Appl.* 8 (2004) 39–45.

- [13] M. J. Rattigan, M. Maier, D. Jensen, Using structure indices for efficient approximation of network properties, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 357–366.
- [14] S. Y. Chan, I. X. Y. Leung, P. Liò, Fast centrality approximation in modular networks, in: Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management, ACM, 2009, pp. 31–38.
- [15] U. Brandes, C. Pich, Centrality estimation in large networks, *International Journal of Bifurcation and Chaos* 17 (07) (2007) 2303–2318.
- [16] J. Pfeffer, K. M. Carley, k-centralities: local approximations of global measures based on shortest paths, in: Proceedings of the 21st international conference companion on World Wide Web, ACM, 2012, pp. 1043–1050.
- [17] K. Okamoto, W. Chen, X.-Y. Li, Ranking of closeness centrality for large-scale social networks, in: *Frontiers in Algorithmics*, Springer, 2008, pp. 186–195.
- [18] V. Ufimtsev, S. Bhowmick, An extremely fast algorithm for identifying high closeness centrality vertices in large-scale networks, in: Proceedings of the Fourth Workshop on Irregular Applications: Architectures and Algorithms, IEEE Press, 2014, pp. 53–56.
- [19] P. W. Olsen, A. G. Labouseur, J.-H. Hwang, Efficient top-k closeness centrality search, in: Data Engineering (ICDE), 2014 IEEE 30th International Conference on, IEEE, 2014, pp. 196–207.
- [20] E. Bergamini, M. Borassi, P. Crescenzi, A. Marino, H. Meyerhenke, Computing Top-k Closeness Centrality Faster in Unweighted Graphs.
- [21] K. Wehmuth, A. Ziviani, Distributed assessment of the closeness centrality ranking in complex networks, in: Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners, ACM, 2012, pp. 43–48.



- [22] D. A. Bader, K. Madduri, Parallel algorithms for evaluating centrality indices in real-world networks, in: *Parallel Processing, 2006. ICPP 2006. International Conference on*, IEEE, 2006, pp. 539–550.
- [23] K. A. Lehmann, M. Kaufmann, Decentralized algorithms for evaluating centrality in complex networks.
- [24] W. Wang, C. Y. Tang, Distributed estimation of closeness centrality, in: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, 2015, pp. 4860–4865.
- [25] M. E. Newman, Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality, *Physical review E* 64 (1) (2001) 016132.
- [26] E. Yan, Y. Ding, Applying centrality measures to impact analysis: A coauthorship network analysis, *Journal of the American Society for Information Science and Technology* 60 (10) (2009) 2107–2118.
- [27] O. Sporns, C. J. Honey, R. Kötter, Identification and classification of hubs in brain networks, *PloS one* 2 (10) (2007) e1049.
- [28] S. Jarukasemratana, T. Murata, X. Liu, Community detection algorithm based on centrality and node closeness in scale-free networks, 29 (2) (2014) 234–244.
- [29] J. Zhang, X. Ma, W. Liu, Y. Bai, Inferring community members in social networks by closeness centrality examination, in: *Web Information Systems and Applications Conference (WISA), 2012 Ninth*, IEEE, 2012, pp. 131–134.
- [30] K. Ko, K. J. Lee, C. Park, Rethinking preferential attachment scheme: degree centrality versus closeness centrality, *Connections* 28 (1) (2008) 4–15.
- [31] A. Saxena, V. Malik, S. Iyengar, Rank me thou shalln’t compare me, *arXiv preprint arXiv:1511.09050*.
- [32] A. Saxena, V. Malik, S. Iyengar, Estimating the degree centrality ranking, in: *2016 8th International Conference on Communication Systems and Networks (COMSNETS), IEEE, 2016*, pp. 1–2.

- [33] A. Saxena, V. Malik, S. Iyengar, Estimating the degree centrality ranking of a node, arXiv preprint arXiv:1511.05732.
- [34] A. Saxena, R. Gera, S. Iyengar, Degree ranking, arXiv preprint arXiv:1706.01205.
- [35] E. Cho, S. A. Myers, J. Leskovec, Friendship and mobility: user movement in location-based social networks, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 1082–1090.
- [36] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowledge and Information Systems 42 (1) (2015) 181–213.
- [37] T. Hogg, K. Lerman, Social dynamics of digg, EPJ Data Science 1 (1) (2012) 1–26.
- [38] B. Klimt, Y. Yang, The enron corpus: A new dataset for email classification research, in: Machine learning: ECML 2004, Springer, 2004, pp. 217–226.
- [39] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: International semantic Web conference, Springer, 2003, pp. 351–368.
- [40] B. Viswanath, A. Mislove, M. Cha, K. P. Gummadi, On the evolution of user interaction in facebook, in: Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 37–42.
- [41] J. J. McAuley, J. Leskovec, Learning to discover social circles in ego networks., in: NIPS, Vol. 2012, 2012, pp. 548–56.
- [42] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, 2010, pp. 1361–1370.
- [43] J. J. Moré, The levenberg-marquardt algorithm: implementation and theory, in: Numerical analysis, Springer, 1978, pp. 105–116.
- [44] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, science 286 (5439) (1999) 509–512.