2017-06-10

Degree ranking using local information

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

A. Saxena, R. Gera, S.R.S., "Degree ranking using local information,"
arXiv:1706.01205v2 [cs.SI] 10 Jun 2017.
https://hdl.handle.net/10945/56661

Degree Ranking Using Local Information

Akrati Saxena*
akrati.saxena@iitrpr.ac.in

Ralucca Gera**
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

Most real world dynamic networks are evolved very fast with time. It is not feasible to collect the entire network at any given time to study its characteristics. This creates the need to propose local algorithms to study various properties of the network. In the present work, we estimate degree rank of a node without having the entire network. The proposed methods are based on the power law degree distribution characteristic or sampling techniques. The proposed methods are simulated on synthetic networks, as well as on real world social networks. The efficiency of the proposed methods is evaluated using absolute and weighted error functions. Results show that the degree rank of a node can be estimated with high accuracy using only 1% samples of the network size. The accuracy of the estimation decreases from high ranked to low ranked nodes. We further extend the proposed methods for random networks and validate their efficiency on synthetic random networks, that are generated using Erdős-Rényi model. Results show that the proposed methods can be efficiently used for random networks as well.

1 Introduction

In complex networks, all nodes have unique characteristics that can be captured using several of the centrality measures proposed in the literature like degree centrality [1], semi-local centrality [2], closeness centrality [3], betweenness centrality [4], eigenvector centrality [5], Katz centrality [6], PageRank [7], and so on. These centrality measures assign a value to each node based on its importance in the given context. But, in real life applications, we are mostly interested in the relative importance of the node with respect to the top ranked node. It can be measured using the rank of the node based on the given centrality measure. In the present work, we estimate degree rank of a node using local information. The degree of a node u is denoted by d_u, which represents the number of neighbors of the node. Degree rank of a node u is defined as, $R_{act}(u) = \sum_v X_{uv} + 1$, where $X_{uv} = 1$, if $d_v > d_u$, otherwise $X_{uv} = 0$. It has been referred as actual degree rank throughout the paper. A node having the highest degree is ranked 1. All nodes having the same degree will have the same rank.

The classical ranking method collects the degree of all nodes and compares them to compute the rank of a node. If the degree of a node can be computed in $O(1)$ time, then the time complexity of this method is $O(n)$. The space complexity is also high as it requires entire network for the processing. It is not feasible to collect the entire network and store and process it for the large scale networks. Due to this very reason, this method is not feasible for large-scale or distributed real world networks.

Real world networks are highly dynamic, so, the rank of a node keeps on changing with time. To estimate the latest rank of a node, the current snapshot of the entire network is required. Even if we collect this dataset, it might not be useful for further estimations. So, the complexity of preprocessing will be very high. There are some more constraints while studying these networks like online social networks can only be accessed using public interface calls or their API. The number of calls is constant due to API's restrictions. These networks can only be sampled using random walk or its variants like weighted random walk, metropolis-hastings random walk, and so on. It creates the need to propose efficient local algorithms to estimate various properties of the network using less amount of data.

In the present work, we propose four methods to estimate degree rank of a node without having the entire network. These methods use a small snapshot of the network that is collected using different sampling techniques. The pro-

posed methods require some network parameters like network size, maximum, minimum or average degree of the network, that are estimated using random walk samples. Once these pre-processing steps are done, the degree rank of a node can be estimated using the proposed methods. The first method uses power law degree distribution characteristic of real world scale-free networks [8] and estimates the degree rank of a node in $O(1)$ time. The next method uses the uniform sampling technique to collect the samples. It computes the local rank of the node in the collected samples, that is extrapolated to estimate its global rank. The last two methods use metropolis-hastings random walk and classical random walk to collect the samples for the rank estimation.

We further study the accuracy and efficiency of the proposed methods on synthetic as well as on real world networks. The accuracy is measured using absolute and weighted error functions. Results show that the proposed methods estimate rank of a node with high accuracy just by using 1% samples of the network size. So, these methods can be used efficiently for online social networks. The proposed methods are verified on 20 real world social networks and the detailed comparison of these methods is discussed in the Results section.

These ranking methods are further extended to rank nodes in random networks. The efficiency of the proposed methods is verified on random networks of 100,000-500,000 nodes. Results show that the degree rank of a node can be estimated efficiently using a small (1% nodes) sample size.

As per the best of our knowledge, this is the first extensive study of its kind. This work can be helpful to make progress in other domains like identification of influential nodes, comparison of the relative importance of nodes, etc. Identification of influential nodes has been the center of various other research problems like an epidemic [9], viral marketing [10], information diffusion [11,12], opinion formation [13], and so on. It has attracted researchers for quite a long time. The proposed methods can be used to rank influential nodes in different contexts. Degree centrality also has been combined with many other centrality measures to identify the influential nodes [2, 14, 15]. Fortunato et al. studied the correlation of in-degree with PageRank of the node [16]. They show that the PageRank is directly proportional to the in-degree, modulo an additive constant. Ghoshal and Barabasi studied the dependency of super stable nodes on their degrees [17]. All these applications show the importance of degree ranking in diverse domains of science. As the network size is increasing very fast with time, it is not feasible to implement

regular methods. So, local algorithms need to be used in such scenarios as they are practical for large-scale dynamic networks. In [18], authors propose fast heuristic methods to estimate the rank of the nodes based on the closeness centrality that itself is a global centrality measure.

The rest of this paper is organized as follows. Next, we discuss related work. In section 3, we discuss methods that are used to estimate the required network parameters. In section 4, all notation that will be used in the paper, are explained. Section 5 describes degree rank estimation methods for scale-free networks. Each of its subsection explains one method in depth. Section 6 explains datasets, error functions, and simulation results for all the proposed methods. Section 7 explains ranking methods for random networks and their validation on Erdos-Renyi networks. The paper is concluded in section 8. This project has various future directions that can be explored further. These are also discussed in the conclusion.

2 Related Work

Real world networks are highly dynamic. Their size is increasing very fast with time and in many cases, they are stored in a decentralized way. It is not feasible to store the entire network to study its characteristics like network size, average degree, clustering coefficient, and so on. A small snapshot of the dataset can be collected at any given time to study network characteristics. This has motivated researchers to use sampling techniques to study network parameters. While sampling, the main focus is that the collected dataset should be a good representative of the complete dataset.

The sampling techniques can be mainly categorized as node selection based sampling techniques [19], edge selection based sampling techniques [19], and graph traversal based sampling techniques. In node selection or edge selection methods, nodes or edges are sampled uniformly at random from the network respectively. Haralabopoulos and Anagnostopoulos proposed Enhanced Random Node Sampling method and compared its efficiency with already existing methods [20]. The paper contains the results of the estimation of network parameters like clustering coefficient, average degree, assortativity, and the number of components in real world networks.

The node or edge sampling methods are not feasible in real world networks as the structure of social networks is not known in advance. So, these networks can be sampled using graph traversal techniques like breadth

first search (BFS) [21], depth first search (DFS) [21], forest fire sampling (FFS) [19], snowball sampling [22], or random walk based methods like simple random walk (RW) [23], Metropolis Hastings random walk (MHRW) [24], reweighted random walk (RWRW) [25], respondent driven sampling (RDS) [26], supervised random walk [27], Modified TOpology Sampling (MTO) [28], walk-estimate [29], Frontier sampling (m-dimensional random walk) [30], Rank Degree sampling based on edge selection [31], preferential random walk [32], and so on.

Next, we discuss estimation methods for network parameters using sampling methods. Kurant et al. proposed a method called SafetyMargin that uses Induced Edges sampling techniques to estimate the network size [33]. The proposed method outperforms state of the art methods even using 10 times small sample size. In 2013, Hardiman and Katzir proposed a more efficient method to estimate the network size using random walk samples [34]. This method is discussed in more detail in Section 3.1. They also proposed methods to compute average clustering coefficient and global clustering coefficient of the network using random walk. There have been proposed some more methods to estimate network size like [35–39].

Sampling techniques also have been used to identify degree related properties like high degree nodes, average degree, or degree distribution of the network. Cooper et al. proposed a biased random walk method to identify high degree nodes in the scale-free networks [40]. Marchetti-Spaccamela proposed a method to estimate the degree of a node in directed network [41]. Dasgupta et al. proposed a method to estimate the average degree of the network using smooth random walk, that is discussed in Section 3.2 [42]. Eden et al. proposed an algorithm to estimate the average degree using $\tilde{O}(1)$ queries [43]. There have been proposed some more methods to estimate the average degree [35, 44]. Cem and Sarac proposed methods to estimate the size and the average degree of online social networks where only one random neighbor of the node can be accessed using API calls [45]. They further used ego-centric sampling and showed that the use of neighborhood information is not always beneficial to estimate network parameters like network size and average degree [46].

Ribeiro et al. studied the mean square error while computing the degree distribution of the network [47]. They further compute the normalized mean square error for estimating the out-degree and in-deg distribution of the directed networks [48]. The proposed method uses Directed Unbiased Random Walk (DURW) that takes a random jump with a fixed probability depending

on the degree of the node while taking the walk. The results show that the out-degree distribution can be estimated more efficiently and accuracy of the in-degree distribution is very less unless the graph is not symmetric.

Thus, we have seen that the sampling methods can be used to estimate various network parameters. In the present work, we use sampling techniques to estimate degree centrality rank of the nodes.

3 Estimation of Network Parameters

Different ranking methods require different network parameters that need to be estimated during the preprocessing steps. In this section, we discuss the methods that are used to estimate these parameters.

3.1 Estimate the Network Size

The network size is estimated using a method (HK method) that was proposed by Hardiman and Katzir [34]. The proposed estimator is based on the concept of collision to count total number of nodes, where samples are collected using the classical random walk. Authors use neighbors' information of the sampled nodes to detect the collision a step before it actually occurs. There is a high probability of collision on short distances due to the local traversal. So, a pair of nodes in random walk samples is considered to count the collision if their distance is long during the random walk. We have considered a pair if their distance is more than 2.5% of sample size. It is the same as taken by the authors.

3.2 Estimate the Average degree

The average degree of the network is used while estimating the rank of a node using power law degree distribution. It is estimated using a method (AD method) that was proposed by Dasgupta et al. [42]. The samples are collected using smoothed random walk with a distribution $D_{d,c}$, where the probability to sample a node u is directly proportional to $d_u + c$, and c is constant during the entire random walk. The samples generated using this distribution are equivalent to samples generated from the network, where $c/2$ self-loops are added to each node.

They proposed two estimators called: 1. Guess&Smooth, 2. Smooth. The optimal value of c is decided using Guess&-Smooth estimator. This smoothing parameter c is used by Smooth estimator to collect the samples. These two estimators are combined to propose an estimator that takes $O(logU \cdot loglogU)$ samples to estimate average degree with high accuracy, where U is an upper bound on the maximum degree of the network. Thus, this estimator takes very less number of samples to estimate the average degree.

The estimated average degree of the network is required while estimating the rank of a node using power law degree distribution method.

3.3 Estimate the Maximum Degree

In power law degree distribution, the frequency of highest degree node is almost 1. In the analysis, maximum degree is estimated as the available maximum degree in the samples, $d'_{max} = max\{d_u, \forall u \in S\}$, where S is the set of samples.

3.4 Estimate the Minimum Degree

In real world networks, we observe that the minimum degree is 1 or close to 1. We use the same value of minimum degree for the analysis, so $d'_{min} = 1$. In BA and ER networks, the minimum degree is estimated as the available minimum degree in the samples.

The minimum and maximum degree are estimated using the network size estimator samples.

4 Notation

$\mathcal{G}(f)$ represents a set of networks having n nodes, and all networks are generated using the same degree distribution f. Table 1 contains all notation used in the paper.

5 Estimate the Degree Rank

In this section, we will explain four methods to estimate degree rank of a node using local information.

Table 1: Notation

Notation	**Description**
G	A network, $G \in \mathcal{G}(f)$
n	Total number of nodes in the network
m	Total number of edges in the network
n'	Estimated number of nodes in the network
n_j	Total number of nodes having degree j in the network
u, v, w	Nodes in the network
d_u	Degree of node u
d_{max}	Maximum degree in the network
d_{min}	Minimum degree in the network
d_{avg}	Average degree of the network
S	Set of sampled nodes
s	Sample size, $s = \|S\|$
d'_{max}	Estimated maximum degree/maximum degree in S
d'_{min}	Estimated minimum degree/minimum degree in S
d'_{avg}	Estimated average degree/average degree of S
$R_{act}(u)$	Actual rank of node u in the network
$R_{est}(u)$	Estimated rank of node u in the network
$R_{local}(u)$	Rank of node u in sample S
$R_G(u)$	A random variable that denotes the rank of node u in G
$R_S(u)$	A random variable that denotes the rank of node u in S

5.1 Using Power Law Degree Distribution (PL Method)

In 1999, Barabasi and Albert observed that degree distribution f of real world scale-free networks follows power law [8]. The probability $f(j)$ of a node having degree j is given as $f(j) = cj^{-\gamma}$, where c and γ are constants for a network. Due to the power law characteristic, only a few nodes manage to get the higher degree in the network. In real world scale-free networks, the range of the exponent is $2 < \gamma < 3$. The degree rank of a node can be computed using power law equation if its parameters are known. In this section, we propose a method to estimate these parameters that can be used further to estimate degree rank of the node.

Theorem 1. *In a scale-free network G ($G \in \mathcal{G}(f)$), the power law exponent of degree distribution can be computed as, $\gamma \approx 2 + \frac{d_{min}}{d_{avg} - d_{min}}$, where d_{min} and d_{avg} represent minimum and average degree of the network respectively.*

Proof. Let network G follows power law degree distribution $f(j) = cj^{-\gamma}$. First, we derive an equation to estimate the value of c. The sum of probabilities of a node having degree j ($d_{min} \leq j \leq d_{max}$) is equal to 1. The probability function of degree distribution can be written as,

$$\sum_{j=d_{min}}^{d_{max}} f(j) = 1.$$

We switch to integration[1] to compute c:

$$\int_{d_{min}}^{d_{max}} f(j)dj = 1,$$
$$\int_{d_{min}}^{d_{max}} c \cdot j^{-\gamma} dj = 1.$$

After integration, we obtain the value for c to be

$$c \cdot \frac{(d_{max})^{1-\gamma} - (d_{min})^{1-\gamma}}{1-\gamma} = 1$$

[1]Here, discrete probability values are considered as continuous probability density function, as this introduces a very small error.

$$c = \frac{1-\gamma}{(d_{max})^{1-\gamma} - (d_{min})^{1-\gamma}}. \tag{1}$$

To compute γ, the average degree of the network, (d_{avg}), is used. Using $f(j) = c \cdot j^{-\gamma}$, it can be computed as

$$d_{avg} = \sum_{j=d_{min}}^{d_{max}} j \cdot f(j)$$

$$d_{avg} = \int_{d_{min}}^{d_{max}} j \cdot \left(c \cdot j^{-\gamma}\right) dj.$$

After integration, we have that

$$d_{avg} = c \cdot \tfrac{d_{max}^{2-\gamma} - d_{min}^{2-\gamma}}{2-\gamma}.$$

Putting value of c from equation (1) in this equation,

$$d_{avg} = \frac{1-\gamma}{2-\gamma} \cdot \frac{d_{max}^{2-\gamma} - d_{min}^{2-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}$$

$$d_{avg} = \frac{\gamma-1}{\gamma-2} \cdot \frac{d_{max}^{\gamma-2} - d_{min}^{\gamma-2}}{d_{max}^{\gamma-1} - d_{min}^{\gamma-1}} \cdot d_{max} \cdot d_{min}$$

where, $d_{min} << d_{max}$, and $2 < \gamma < 3$ for scale-free real networks [8].

$$d_{avg} \approx \tfrac{\gamma-1}{\gamma-2} \tfrac{d_{max}^{\gamma-2}}{d_{max}^{\gamma-1}} \cdot d_{max} \cdot d_{min}$$

$$d_{avg} \approx \tfrac{\gamma-1}{\gamma-2} \cdot d_{min}$$

i.e. $\gamma \approx 2 + \tfrac{d_{min}}{d_{avg} - d_{min}}$. □

We next present the expected degree rank of a node.

Theorem 2. *In a network G ($G \in \mathcal{G}(f)$), the expected degree rank of a node u can be computed as, $E[R_G(u)] \approx n\left(\frac{d_{max}^{1-\gamma} - (d_u+1)^{1-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}\right) + 1$, where γ is the power law exponent of the degree distribution of network G.*

Proof. In a given network G, the actual rank of a node u having degree d_u can be computed as,

$$R_{act}(u) = \sum_{j=d_u+1}^{d_{max}} n_j + 1$$

where n_j represents total number of nodes having degree j in network G. Let N_j be a random variable that represents total number of nodes having degree j in G. Then, the expected value of N_j can be computed as, $E[N_j] = n \cdot f(j)$. Thus the expected degree rank of a node u can be computed as

$$E[R_G(u)] = E\left[\sum_{j=d_u+1}^{d_{max}} N_j + 1\right]$$

$$E[R_G(u)] = \sum_{j=d_u+1}^{d_{max}} E[N_j] + 1$$

$$E[R_G(u)] = \sum_{j=d_u+1}^{d_{max}} n \cdot f(j) + 1$$

$$E[R_G(u)] \approx n \int_{d_u+1}^{d_{max}} f(j)dj + 1.$$

Since $f(j) = cj^{-\gamma}$, after the integration of $E[R_G(u)] \approx n \int_{d_u+1}^{d_{max}} c \cdot j^{-\gamma} dj + 1$ we have

$$E[R_G(u)] \approx nc\frac{d_{max}^{1-\gamma} - (d_u+1)^{1-\gamma}}{1-\gamma} + 1.$$

Replacing the value of c from equation (1), we obtain

$$E[R_G(u)] \approx n(\frac{d_{max}^{1-\gamma} - (d_u+1)^{1-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}) + 1$$

as desired. □

And so, using Theorem 2 and given general estimators about the network, we can estimate the degree rank of nodes.

Corollary 2.1. *In a network G ($G \in \mathcal{G}(f)$), the degree rank of a node u can be estimated as,*

$$R_{est}(u) = n' \left(\frac{(d'_{max})^{1-\gamma} - (d_u+1)^{1-\gamma}}{(d'_{max})^{1-\gamma} - (d'_{min})^{1-\gamma}}\right) + 1,$$

where $\gamma = 2 + \frac{d'_{min}}{d'_{avg} - d'_{min}}$, *and* n', d'_{min}, d'_{max}, *and* d'_{avg} *denote the estimated value of network size, minimum degree, maximum degree, and average degree of the network respectively.*

In one of our previous works, we have validated this method on BA networks [49, 50]. The proposed method estimates the rank with high accuracy for BA networks but does not give good results for real world networks, as they follow power law degree distribution with a droop head and a heavy tail. We further compute variance in degree rank estimation using power law degree distribution and the results show that there is a high variance for lower degree nodes [51].

Next, we propose few more sampling based approaches that perform better on real world networks. These are discussed below.

5.2 Using Uniform Sampling (US Method)

In this section, the uniform sampling technique is used to collect a small sample of actual dataset. In uniform sampling, the probability of sampling a node is equal to $1/n$, where n is total number of nodes. Uniform samples preserve the characteristics of actual dataset. So, the collected samples follow similar degree distribution as observed in real world networks. Here we assume that the network G is generated using degree distribution f_1 and $G \in \mathcal{G}(f_1)$. Now, Theorem 3 can be used to estimate the rank of a node using uniform samples. The expected global rank of a node can be estimated by extrapolating its local rank in the collected sample set.

Theorem 3. *In a network* G ($G \in \mathcal{G}(f_1)$), *if sample* S *is collected uniformly, the expected local rank of node* u *can be computed as,* $E[R_S(u)] \approx \frac{s}{n} E[R_G(u)]$, *where* $R_G(u)$ *and* $R_S(u)$ *are random variables that denote the rank of node* u *in network* G *and sample* S *respectively.*

Proof. We are interested in computing the rank of a node u having degree d_u. Let's take a random variable N_j, that denotes the number of nodes having degree j in the network. The expected value of N_j can be computed as, $E[N_j] = n \cdot f_1(j)$.

The expected rank of node u in network G can be computed as:

$$E[R_G(u)] = E[\sum_{j=d_u+1}^{d_{max}}(N_j) + 1]$$

$$E[R_G(u)] = \sum_{j=d_u+1}^{d_{max}} (n \cdot f_1(j)) + 1 \tag{2}$$

Now, we have a uniform sample S of size s. In network G, the probability p to sample a node y uniformly at random having degree greater than d_u ($d_y > d_u$) can be defined as,

$$p = \frac{\sum_{j=d_u+1}^{d_{max}}(n \cdot f_1(j))}{\sum_{j=1}^{d_{max}}(n \cdot f_1(j))}$$

Using equation (2),

$$p = \frac{E[R_G(u)]-1}{\sum_{j=1}^{d_{max}}(n \cdot f_1(j))}$$
$$p = \frac{E[R_G(u)]-1}{n\sum_{j=1}^{d_{max}}(f_1(j))}$$

Using the property of probability distribution $\sum_{j=1}^{d_{max}} f_1(j) = 1$.

$$E[R_G(u)] = p \cdot n + 1 \tag{3}$$

The expected value of local rank of node u in sample S can be computed as,

$$E[R_S(u)] = \sum_{j=0}^{s} \binom{s}{j} p^j (1-p)^{(s-j)} j + 1$$

$$E[R_S(u)] = s \cdot p + 1 \tag{4}$$

Using equations (3) and (4),

$$E[R_S(u)] = \frac{s}{n} E[R_G(u)] + \frac{n-s}{n}$$

Where, $0 \leq (n-s)/n < 1$, if $s \leq n$. So,

$$E[R_S(u)] \approx \frac{s}{n} E[R_G(u)]$$

□

In a network G, $R_{act}(u) \approx E[R_G(u)]$ and $R_{local}(u) \approx E[R_S(u)]$. $R_{local}(u)$ denotes the rank of node u in sample S, and $R_{local}(u) = \sum_{j=i+1}^{d'_{max}}(n'_j) + 1$, where n'_j is the number of nodes having degree j in sample S. Using theorem 3, the actual rank of node u can be computed as,

$$R_{act}(u) \approx \frac{n}{s} R_{local}(u) \tag{5}$$

Corollary 3.1. *In a network G, using uniform samples, degree rank of a node u can be estimated as,* $R_{est}(u) = \frac{n'}{s} R_{local}(u)$*, where* n' *is the estimated network size.*

5.3 Using Metropolis-Hastings Random Walk (MH Method)

In most of the online networks, uniform sampling is not possible as node ids are not known well in advance. These networks can be sampled using graph sampling techniques like breadth first traversal, random walk, etc. These sampling methods are biased towards higher degree nodes and fail to generate uniform samples. In this method, we use metropolis-hastings random walk that generates sample equivalent to uniform samples, that can be used for rank estimation.

Metropolis-Hastings Random Walk: This technique was first proposed by Metropolis et al. [24] in 1953. In this method, the probability distribution of random walk is modified so that the collected samples retain the properties of the actual distribution of the dataset. At each time step, the crawler will move to the next node with probability p and will stay at the same node with probability $(1 - p)$. So, the probability distribution can be modified as,

$$P_{u \to v} = \begin{cases} \frac{1}{d_u} \cdot min(1, \frac{d_u}{d_v}), & if\ v\ is\ the\ neighbor\ of\ u, \\ 1 - \sum_{w \neq u} P_{u \to w}, & if\ v = u, \\ 0, & otherwise. \end{cases}$$

This probability distribution collects more samples of lower degree nodes and fewer samples of higher degree nodes, so the collected samples are not biased towards higher degrees. Gjoka et al. studied that in real world network, the samples collected using metropolis-hastings random walk are equivalent to uniform samples, and can be used to study the network parameters [52]. Corollary 3.1 can be used to estimate degree rank using MHRW samples.

5.4 Using Random Walk (RW Method)

The classical random walk is a well known easier method to collect the samples in large dynamic networks. In **Random Walk**, a crawler starts from a randomly chosen node. It moves to the next node that is chosen uniformly at random among the neighbors of the current node [23]. The probability to move to node v from node u is defined as,

$$P_{u\to v} = \begin{cases} \frac{1}{d_u}, & if\ v\ is\ a\ neighbor\ of\ u, \\ 0, & otherwise. \end{cases}$$

In a random walk, the probability of a node being sampled converges to a stationary distribution, $p(u) = d_u/2m$. So, the collected samples are biased towards high degree nodes. We propose Theorem 4 to estimate degree rank using random walk samples.

First, notice that in a random walk, the probability of a node being sampled is directly proportional to its degree. These samples can be converted to uniform samples using a new probability distribution, where the probability of picking a node is inversely proportional to its degree $p(u) \propto 1/d_u$, known as re-weighted random walk sampling technique [25].

Theorem 4. *In a network G ($G \in \mathcal{G}(f_1)$), using random walk sample S, the degree rank of node u can be computed as, $R_{act}(u) \approx \frac{n}{k} \cdot R_{local}(u)$, where $R_{local}(u) = \sum_{j=d_u+1}^{d'_{max}} (q(j) \cdot k) + 1$, and k is a constant, $q(j)$ is the re-sampling probability function $q(j) = \frac{n'_j/j}{\sum_{i=d'_{min}}^{d'_{max}} n'_i/i}$, and n'_j represents total number of nodes having degree j in sample S.*

Proof. The probability q to resample a j degree node can be computed as, $q(j) = \frac{n'_j/j}{\sum_{i=d'_{min}}^{d'_{max}} n'_i/i}$, where n'_j represents total number of nodes having degree j in sample S.

To estimate the degree rank of a node, collect $q(j) \cdot k$ samples of each degree j from S, where k is a constant. So, total number of new sampled nodes $|S'| = \sum_{j=d'_{min}}^{d'_{max}} (q(j) \cdot k) = k$. Then, the rank of node u in S' can be computed as, $R_{local}(u) = \sum_{j=d_u+1}^{d'_{max}} (q(j) \cdot k) + 1$.

As the new sample set S' follows uniform distribution, the rank of node u can be computed using equation (5), $R_{act}(u) \approx \frac{n}{k} \cdot R_{local}(u)$. □

In the experiments, the value of k is chosen as $k = min(1/q)$, so that the regenerated samples also contain higher degree nodes and their rank is estimated with high accuracy.

Corollary 4.1. *In a network G, using random walk samples, the degree rank of a node u can be estimated as,*

$$R_{est}(u) = n' \cdot \frac{\sum_{j=d_u+1}^{d'_{max}} \left(\frac{n'_j}{j} \cdot k\right)}{\sum_{j=d'_{min}}^{d'_{max}} \left(\frac{n'_j}{j} \cdot k\right)},$$

where n'_j represents total number of nodes having degree j in sample S, and k is a constant.

6 Simulation Results

In this section, we will discuss the datasets, error functions, and simulation results.

6.1 Datasets

All proposed methods are simulated on both synthetic as well as on real world scale-free networks. Synthetic networks are generated using Barabási-Albert (BA) model $G(n, k)$, where each new coming node makes k preferential connections with already existing nodes [8]. The probability $p(u)$ to make a connection with an existing node u is directly proportional to the degree of node u, as $p(u) = d_u / \sum_v d_v$. So, the nodes having higher degrees acquire more links over time and it gives birth to power law degree distribution. All synthetic datasets are explained in Table 2.

Table 2: Datasets

Network	#Nodes	#Edges
BA1	100000	999900
BA2	200000	1999900
BA3	300000	2999900
BA4	400000	3999900
BA5	500000	4999900

All real world datasets are explained below:

1. Academia Online Social Network: Academia.edu is an online website where academics share research papers. This is an extracted social network from this website [53, 54]. It contains 200167 nodes and 1022440 edges.

2. Actor Collaboration Network: In actor collaboration network, nodes represent actors and they are connected by an edge if they both have performed in the same movie [8]. This network contains 374511 nodes and 15014839 edges.

3. Catster Friendship Network: This social Network is created using the friendships between catster.com website users [55]. catster provides a platform to cat owners and lovers, where they can connect with each other and share the information. It contains 148826 nodes and 5447464 edges.

4. DBLP Collaboration Network: This is a coauthorship network extracted from DBLP computer science bibliography, where edge denotes that the authors have common publications [56]. This network contains 317080 nodes and 1049866 edges.

5. Delicious online social network: Delicious is a social bookmarking web service for storing, sharing, and discovering web bookmarks. The dataset contains all links among users [53,57]. It contains 536108 nodes and 1365961 edges.

6. Digg Friendship Network: This friendship network was extracted from Digg website in 2009 [58]. It contains 261489 nodes and 1536577 edges.

7. Dogster Friendship Network: This social Network is created using the friendships between users of the website http://www.dogster.com [53]. catcher provides a platform to cat owners and lovers, where they can connect with each other and share the information. It contain 426485 nodes and 8543321 edges.

8. Douban online social network: Douban is an online social network that provides user review and recommendation services for movies, books, and music. This is the friendship network extracted from the website [53, 57]. It contains 154908 nodes and 327162 edges.

9. European Email Communication Network: This is the email communication network of a European research institution, where a node represents an individual person and an edge represents that at least one email has been exchanged between them [59]. This dataset was collected from October 2003 to May 2005 (18 months).

10. Facebook Network: Facebook is the most popular online social networking site today. This dataset is the induced subgraph of Facebook, where users are represented by nodes and friendships are represented by edges [60, 61]. It contains 3097165 nodes and 23667394 edges.

11. Friendster Network: This is the induced subgraph of Friendster online social network [53]. Nodes represent users and a directed edge (a,b) indicates that user a has added user b to his friendship lists. The network is converted to a undirected network for study. It contains 5689498 nodes and 14067887 edges.

12. Foursquare Network: Foursquare is a location-based social networking software for mobile devices that can be accessed using GPS. This dataset is an induced subgraph of friendships of Foursquare [62]. It contains 639014 nodes and 3214985 edges.

13. Gowalla Social Network: This network is extracted from a location-based social network called, Gowalla [63]. This was used to share the locations among its users. In this network, a node represents a user and an edge indicates the friendship between the user. It contains 196591 nodes and 950327 edges.

14. Google Plus Social Network: This is an induced subgraph of Google plus online social network [64]. It contains 107614 nodes and 12238285 edges.

15. Hollywood Collaboration Network: This is an undirected collaboration network of Hollywood movie actors where nodes are actors, and there is an edge between two actors if they have appeared in a movie together [65].It contains 1069126 nodes and 56306653 edges.

16. Hyves: Hyves is a popular social networking site in the Netherlands that is mainly used by Dutch visitors. This is the induced subgraph

of this that was collected in December 2010 [62]. The network is undirected and unweighted. It contains 1402673 nodes and 2777419 edges.

17. last.fm Network: Last.fm is a music website that has more than 40 million active users [53, 66]. This is the induced friendship networks of the bloggers from this website. It contains 1191805 nodes and 4519330 edges.

18. Livemocha Network: Livemocha was an online language learning website and this network is extracted from the social connections of the website [67]. This contains 104103 nodes and 2193082 edges.

19. Pokec Online Social Network (soc-pokec): Pokec is a popular online social network in Slovakia. The dataset contains a list of user relationships [53]. It contains 1632803 nodes and 22301964 edges.

20. Youtube Social Network: This is the induced subgraph of Youtube social network [62]. This network contains 1134885 nodes and 2987468 edges.

6.2 Error Functions

The accuracy of all methods is evaluated using absolute and weighted error functions. These are discussed below:

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

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

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

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

2. Weighted Error: In real life applications, the significance of the error depends on two important parameters: 1. rank of the node, and 2. network size. The same rank difference has more impact for the higher ranked nodes than the lower ranked nodes. Similarly, the same error in the rank will be perceived higher in smaller networks than the larger networks. We consider both of these parameters and propose a weighted error function. It is defined as,

$$Err_{wtd}(x) = \frac{Err_{abs}(x)}{n} \cdot \frac{(n - R_{act}(u) + 1)}{n} \cdot 100\%$$

Where, $\frac{(n - R_{act}(u) + 1)}{n} \times 100$ denotes percentile of node u. The weighted error increases linearly with the percentile and decreases with the network size, if the absolute error is constant.

6.3 Results and Discussion

In this section, we will discuss simulation results of all proposed methods. The network parameters like size, average, maximum, and minimum degree are estimated using the methods discussed in Section 2. The network size estimation method converges approximately at 1% samples. Each experiment is repeated 10 times and the average value is considered for further experiments.

To measure the performance of the proposed methods, the average error is calculated for each degree and it is averaged over all degrees to compute the overall error in rank estimation. Each value (absolute and weighted error) is computed by taking the average of 20 iterations of the experiment. Results for US, MH, and RW methods are shown when 1% nodes are sampled. All methods are validated on 20 real world social networks, and the summarized results are shown in Table 3. The detailed results are shown in Appendix I

Table 3: Average Estimation Error on 20 Real World Social Networks

Method	Err_{paae}	Err_{wtd}
PL	1.51	1.14
US	0.13	0.12
MH	0.50	0.41
RW	0.16	0.13

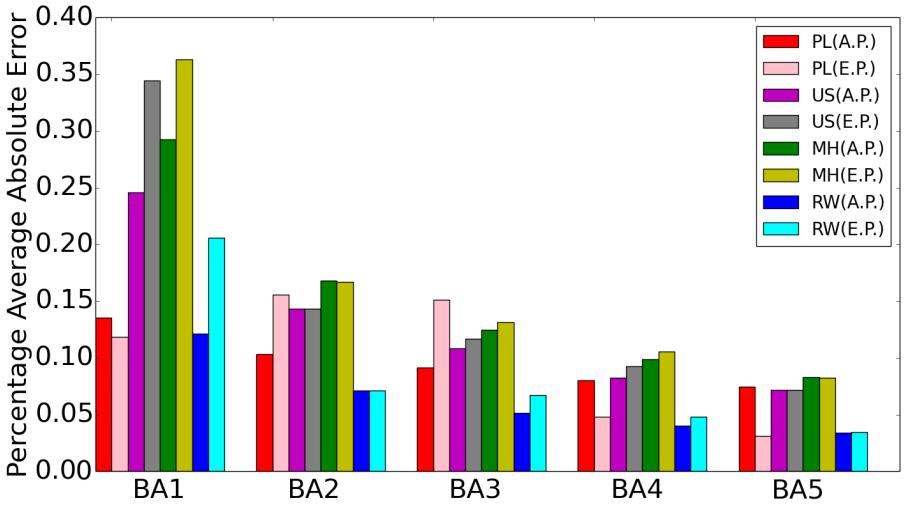
Results show that US method performs best on real world networks. As uniform sampling is not possible in real world networks, RW method is the most feasible and accurate method. In random walk samples, the probability of sampling a node is directly proportional to its degree once the samples are stabled. But in our experiments, we have not removed samples before mixing time and results are shown for the starting 1% samples. It makes the proposed random walk method even faster. MH method gives more error than both US and RW methods because MH random walk is not able to

generate perfect uniform samples for small sample size. The efficiency of the sampling methods increases with the sample size. The performance of PL method is poor on real world networks as they do not follow the perfect power law.

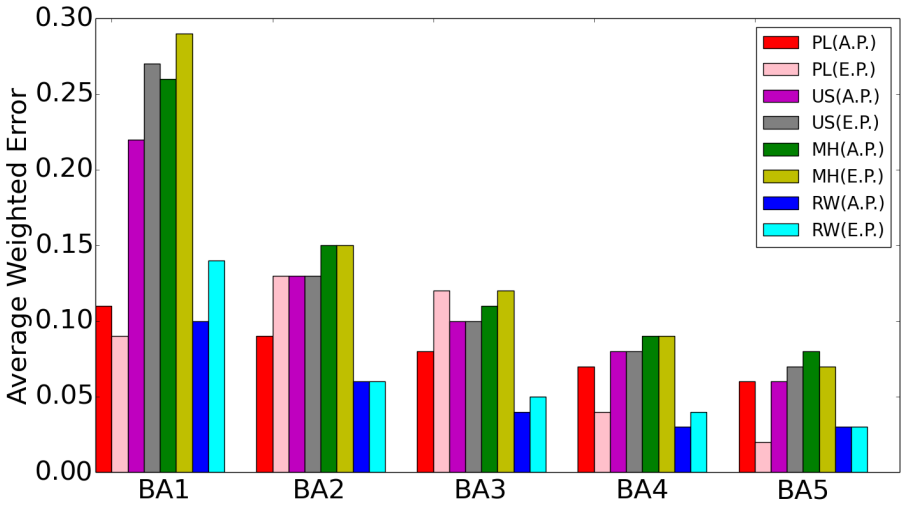
RW versus PL Method: In RW method, same samples can be used to estimate the network size and degree rank, so it is faster than the other sampling methods. RW method is also faster than PL method because PL method estimates the average degree using smoothed random walk that is not required in RW method. But if the network parameters are already known, PL method can be used to estimate the rank in $O(1)$ time.

The detailed results are shown for 10 networks and their estimated parameters are shown in Table 4. The average error is shown using actual parameters (A.P.) as well as estimated parameters (E.P.) to observe the error caused due to the estimation of network parameters. Figures 1 and 2 show percentage average absolute error and average weighted error for BA and real world networks respectively. In these figures, first 2 bars show the average error of PL method using actual and estimated parameters respectively. Then it is followed by US, MH, and RW methods. It can be observed that in BA networks, RW method outperforms all other methods. In real world networks, the accuracy of RW method depends on the density and structure of the network. It gives more accurate results in sparse networks than the dense networks. The Same pattern is observed in MH method, as it also collects samples using a walk over the network.

We further study, the behavior of estimation error with degree rank. Figure 3 shows absolute error versus degree rank for real world (a. Actor, and b. DBLP) networks and BA network. In all methods, estimation error increases with the rank. Figures 3(a) and 3(b) show that for high ranked nodes, RW method outperforms other methods. US method gives more error for high ranked nodes, as the linear extrapolation technique starts assigning rank from n/s. If α percent nodes are sampled, it will start ranking nodes from $100/\alpha$, that will induce more error for high ranked nodes. Figure 3(c) shows that PL method outperforms other methods in BA networks, that is followed by RW method. PL method estimates the rank in $O(1)$ time once the preprocessing steps are done. We observe that PL method works well for BA networks but it gives a huge error for real world networks. It happens because of these two reasons. Firstly, real world networks do not follow the perfect power law. Secondly, the rank of a d degree node is computed by integrating the probability distribution function from $d+1$ to d_{max}, but in real world networks

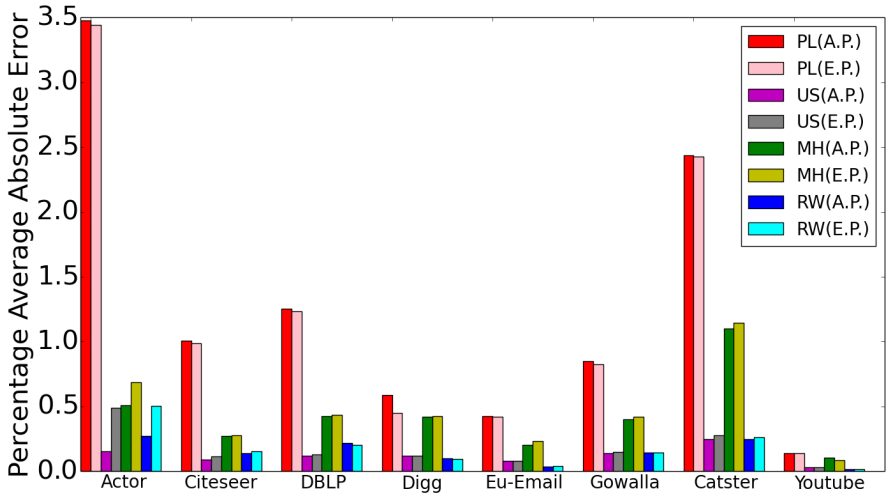


(a) Absolute Error

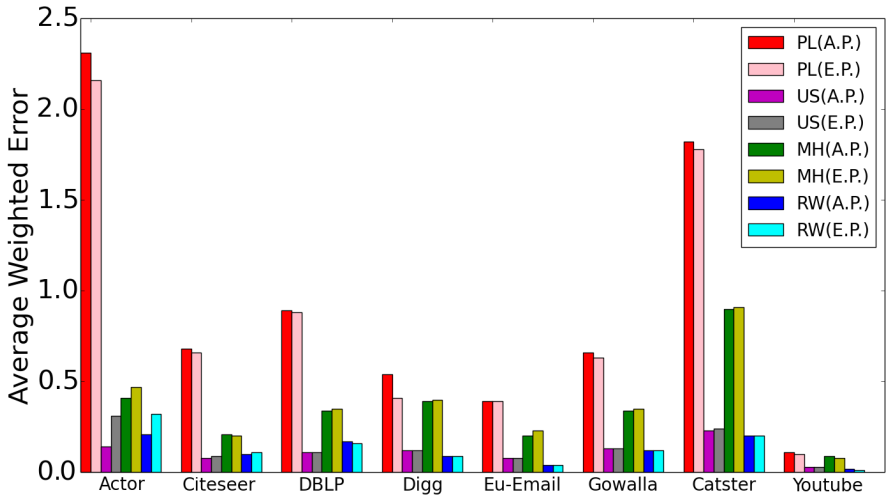


(b) Weighted Error

Figure 1: Average Estimation Error for BA Networks



(a) Absolute Error



(b) Weighted Error

Figure 2: Average Estimation Error for Real World Networks

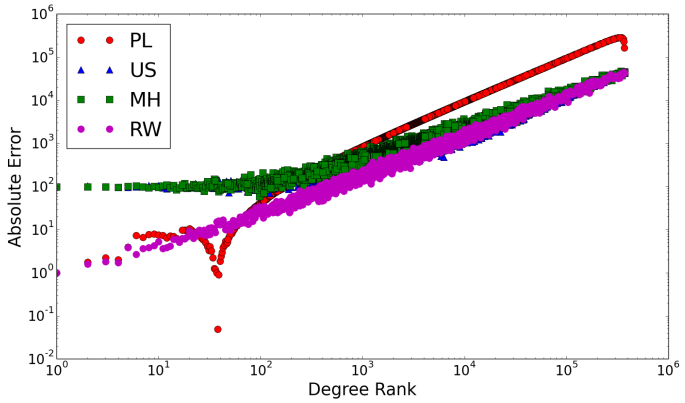
Table 4: Estimated Network Parameters

Network	Number of Nodes		Average Degree	
	Actual	Estimated	Actual	Estimated
BA1	100000	106773	20.00	19.68
BA2	200000	199303	20.00	19.75
BA3	300000	292649	20.00	191.09
BA4	400000	406837	20.00	20.06
BA5	500000	500688	20.00	20.30
Actor	374511	417560	80.18	92.53
DBLP	317080	315587	6.62	7.20
Digg	261489	260435	11.75	17.00
Eu-Email	224832	223151	3.02	2.96
Gowalla	196591	199568	9.67	10.92
Youtube	1134885	1136445	5.26	10.15

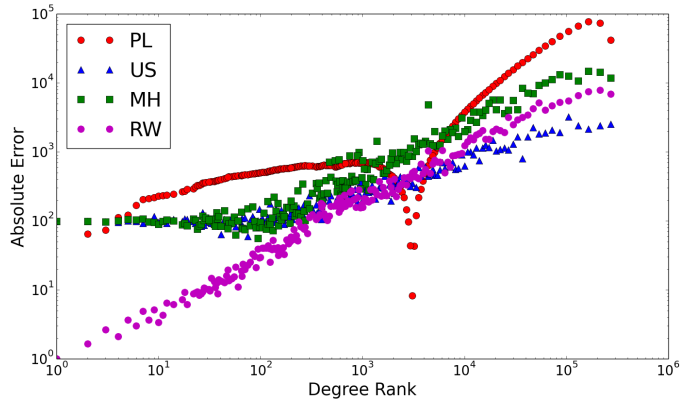
nodes of some degrees might not be present. This further adds up to more error.

In figure 3(a) and 3(b), it can be observed that the rank estimation error in PL method decreases, goes very close to zero, and it further increases. This gives a dip in the absolute error when it is plotted with degree rank. This happens due to the error in the estimated slope of the degree distribution. The actual and estimated number of nodes versus degrees are shown in figure 4 for DBLP network. So in this network, first, the estimated rank will be lower than the actual rank, then, it will be close to zero when the estimated rank is approximately equal to the actual rank, and finally, the estimated rank will be higher than the actual rank. Due to this reason, it shows a strange dip in the absolute error.

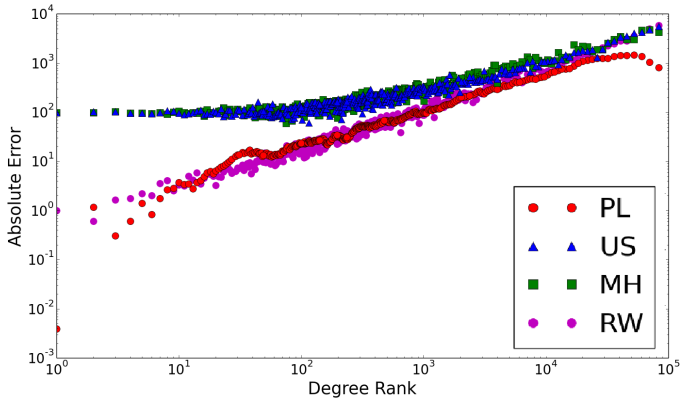
Next, we study how the estimation error changes with the network size. To study the same, a BA network is evolved by maintaining the same density. In Figure 5, percentage average absolute error and average weighted error are plotted against the network size. Plots show that the error decreases with an increase in the network size. It can also be observed that RW method outperforms other methods as the network size increases.



(a) Actor Network



(b) DBLP Network



(c) BA1 Network

Figure 3: Absolute Estimation Error versus Degree Rank on log-log scale

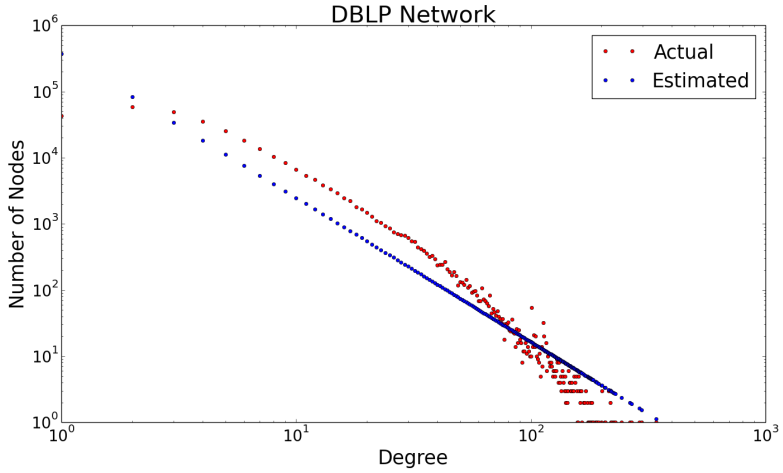


Figure 4: Actual and Estimated Number of Nodes versus Degree

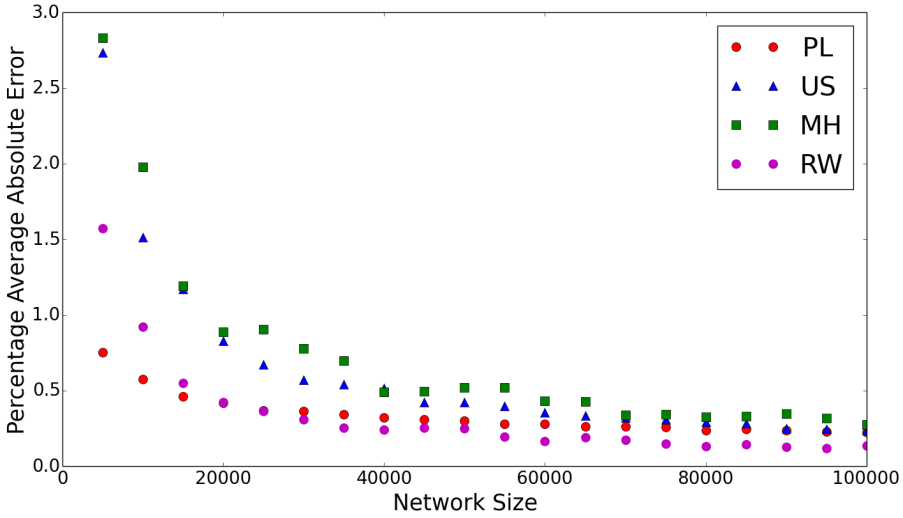
7 Random Networks

In this section, we study degree ranking methods for random networks. In 1959, Erdős and Rényi proposed a model to generate random networks, called Erdős and Rényi (ER) model [68]. In ER model, a network is started with n nodes, and an edge is placed between each pair of nodes with some fixed probability p. The degree distribution g of random networks follows poisson law. The probability of a node having degree j can be approximated as $g(j) \rightarrow \frac{(d_{avg})^j e^{-d_{avg}}}{j!}$ as $n \rightarrow \infty$, where d_{avg} is average degree of the network. We will discuss a ranking method based on poisson law degree distribution. The rest of the methods (US, MH, and RW method) can be directly applied to random networks, as they have no dependency on the degree distribution function. Network size and average degree are estimated using the same techniques that we have discussed earlier.

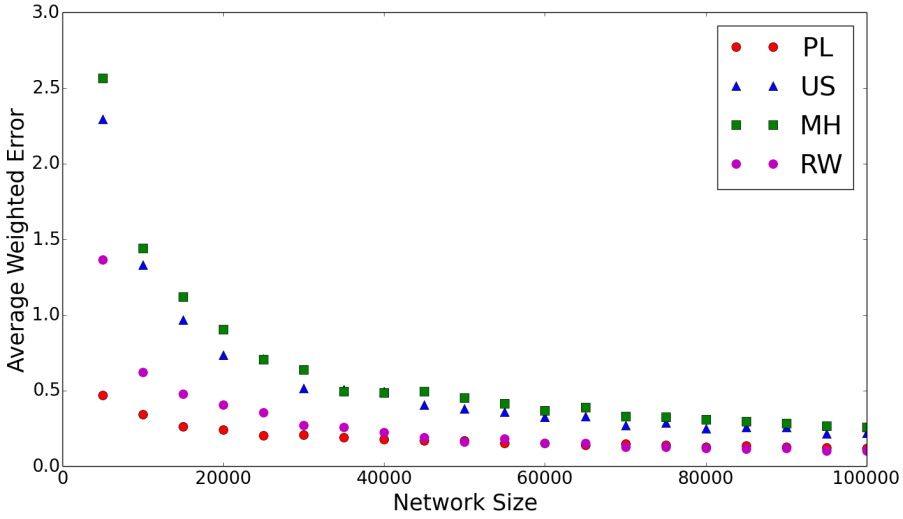
7.1 Using Poisson Degree Distribution (PD Method)

This method uses Poisson degree distribution of random networks to estimate degree rank of a node.

Lemma 5. *In a random network G ($G \in \mathcal{G}(g)$), the expected degree rank of a node u can be computed as, $E[R_G(u)] = n \cdot e^{-d_{avg}} \sum_{j=d_u+1}^{d_{max}} \frac{(d_{avg})^j}{j!} + 1$.*



(a) Absolute Error



(b) Weighted Error

Figure 5: Average Estimation Error versus Network Size for BA networks

Table 5: Estimated parameters for Erdős and Rényi Networks

Network	Number of Nodes		Average Degree	
	Actual	Estimated	Actual	Estimated
ER1	100000	99874	11.50	11.24
ER2	200000	202731	12.34	12.08
ER3	300000	300503	12.71	12.49
ER4	400000	398168	12.99	121.01
ER5	500000	505675	13.19	13.07

Proof. In a given network G that follows Poisson degree distribution, the actual rank of a node u can be computed as,

$$R_{act}(u) = \sum_{j=d_u+1}^{d_{max}} n_j + 1$$

where, n_j represents total number of nodes having degree j in network G.

Let N_j be a random variable that represents the total number of nodes having degree j in the network G ($G \in \mathcal{G}(g)$). The expected value of N_j is $E[N_j] = n \cdot g(j)$. Then, the expected degree rank of a node u can be computed as,

$$E[R_G(u)] = E\left[\sum_{j=d_u+1}^{d_{max}} N_j + 1\right]$$

$$E[R_G(u)] = \sum_{j=d_u+1}^{d_{max}} E[N_j] + 1$$

$$E[R_G(u)] = \sum_{j=d_u+1}^{d_{max}} n \cdot g(j) + 1$$

As we know $g(j) \to \frac{(d_{avg})^j e^{-d_{avg}}}{j!}$ as $n \to \infty$, so to compute the expected

rank we use $g(j) = \frac{(d_{avg})^j e^{-d_{avg}}}{j!}$,

$$E[R_G(u)] = n \cdot \sum_{j=d_u+1}^{d_{max}} \frac{(d_{avg})^j e^{-d_{avg}}}{j!} + 1$$

$$E[R_G(u)] = n \cdot e^{-d_{avg}} \sum_{j=d_u+1}^{d_{max}} \frac{(d_{avg})^j}{j!} + 1,$$

as desired. □

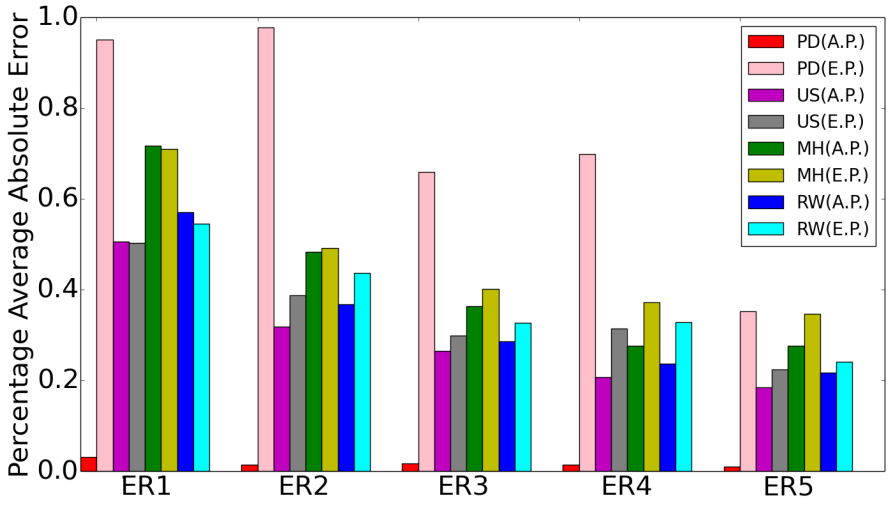
Corollary 5.1. *In a random network G, the degree rank of a node u can be estimated as,*
$R_{est}(u) = n \cdot e^{-d'_{avg}} \sum_{j=d_u+1}^{d'_{max}} \frac{(d'_{avg})^j}{j!} + 1$, *where d'_{max} and d'_{avg} are estimated maximum and average degree of the network respectively.*

7.2 Discussion

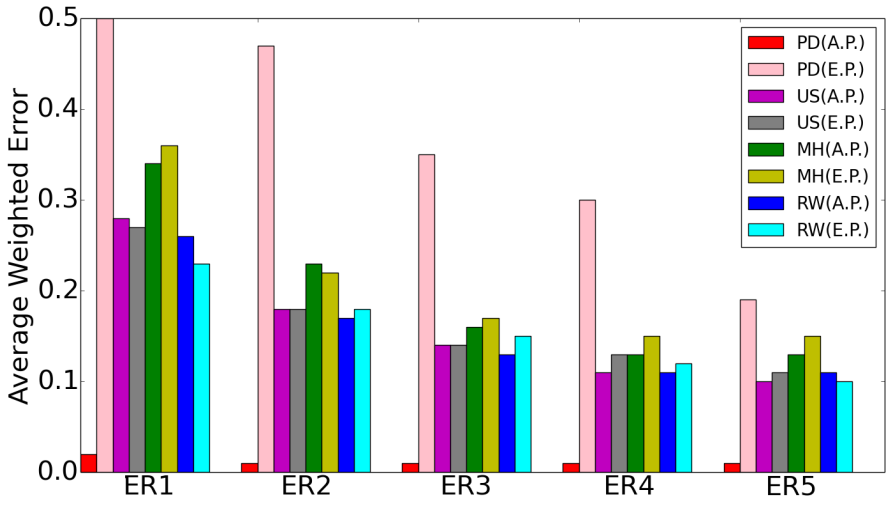
The proposed methods (PD, US, MH, RW) are verified on the generated ER networks, and their details are given in table 5. Figure 6 shows percentage average absolute error and average weighted error using actual and estimated parameters. In PD method, the error computed using estimated parameters is very high as the rank is highly dependent on the average degree. The number of nodes for each degree j is directly proportional to $(d_{avg})^j$, so, a small estimation error leads to more cumulative error. Rest of the results are similar to scale-free networks. The average error of RW method is very close to US method, and it can be efficiently used for large size random networks. The performance of RW method improves with network size. It is also observed that the estimation error decreases as the network size increases. In random networks, absolute error versus degree rank shows a different behavior due to the poisson degree distribution. Figure 7 shows that the estimation error first increases with the rank and then decreases.

8 Conclusion

In this work, we have proposed four methods to estimate degree rank of a node without having the entire structure of the network. With time, the size of real world dynamic networks is increasing very fast. It is not feasible



(a) Absolute Error



(b) Weighted Error

Figure 6: Average Estimation Error for ER Networks

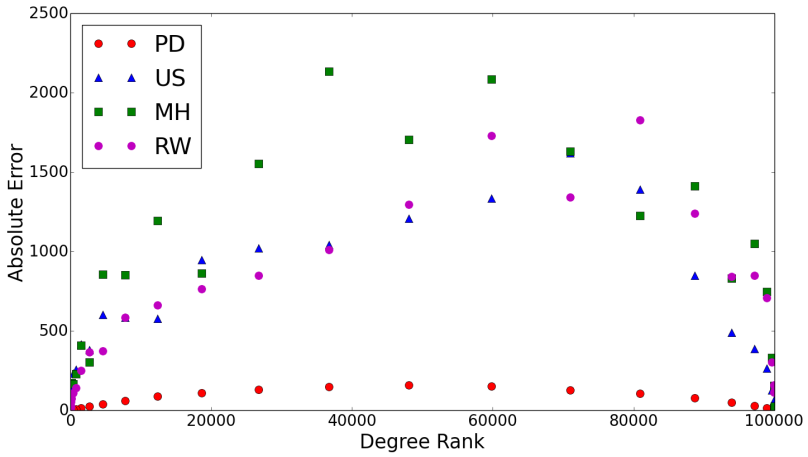


Figure 7: Absolute Estimation Error versus Degree Rank for ER1 Network

to collect the entire network to study its global properties. The proposed methods collect a small sample set using random walk or its variations and estimate global degree rank of the node.

The accuracy of the proposed methods is evaluated using absolute and weighted error functions. It is observed that the accuracy of RW method is very close to US method. RW method is the most feasible and accurate method for real world networks. In RW method, percentage average absolute error is 0.16% and average weighted error is 0.13%. All proposed methods estimate the rank of higher degree nodes more accurately than the lower degree nodes. These methods are further extended to random networks. Results show that they can be efficiently used to estimate degree rank in random networks.

One can further extend this to estimate the rank of a node based on other centrality measures like closeness centrality, betweenness centrality, katz centrality, pagerank, coreness, and so on. The complexity to compute these global centrality measures is very high. So, the local algorithms to compute the global rank of the nodes will be of great interest.

References

[1] M. E. Shaw, Some effects of unequal distribution of information upon group performance in various communication nets, Journal of abnormal and social psychology 49 (4) (1954) 547–553.

[2] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, T. Zhou, Identifying influential nodes in complex networks, Physica a: Statistical mechanics and its applications 391 (4) (2012) 1777–1787.

[3] G. Sabidussi, The centrality index of a graph, Psychometrika 31 (4) (1966) 581–603.

[4] L. C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977) 35–41.

[5] K. Stephenson, M. Zelen, Rethinking centrality: Methods and examples, Social Networks 11 (1) (1989) 1–37.

[6] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.

[7] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine in: Seventh international world-wide web conference (www 1998), april 14-18, 1998, brisbane, australia, Brisbane, Australia.

[8] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, science 286 (5439) (1999) 509–512.

[9] B. Hu, J. Gong, Simulation of epidemic spread in social network, in: Management and Service Science, 2009. MASS'09. International Conference on, IEEE, 2009, pp. 1–4.

[10] J. Leskovec, L. A. Adamic, B. A. Huberman, The dynamics of viral marketing, ACM Transactions on the Web (TWEB) 1 (1) (2007) 5.

[11] Y. Gupta, A. Saxena, D. Das, S. Iyengar, Modeling memetics using edge diversity, in: Complex Networks VII, Springer, 2016, pp. 187–198.

[12] A. Saxena, S. Iyengar, Y. Gupta, Understanding spreading patterns on social networks based on network topology, in: Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on, IEEE, 2015, pp. 1616–1617.

[13] D. J. Watts, P. S. Dodds, Influentials, networks, and public opinion formation, Journal of consumer research 34 (4) (2007) 441–458.

[14] B. Hou, Y. Yao, D. Liao, Identifying all-around nodes for spreading dynamics in complex networks, Physica A: Statistical Mechanics and its Applications 391 (15) (2012) 4012–4017.

[15] Y. Yu, S. Fan, Node importance measurement based on the degree and closeness centrality, Journal of Information & Commputational Science, pp-1281-1291.

[16] S. Fortunato, M. Boguñá, A. Flammini, F. Menczer, Approximating pagerank from in-degree, in: Algorithms and models for the web-graph, Springer, 2006, pp. 59–71.

[17] G. Ghoshal, A.-L. Barabási, Ranking stability and super-stable nodes in complex networks, Nature communications 2 (2011) 394.

[18] A. Saxena, R. Gera, S. Iyengar, A faster method to estimate closeness centrality ranking, arXiv preprint arXiv:1706.02083.

[19] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 631–636.

[20] G. Haralabopoulos, I. Anagnostopoulos, Real time enhanced random sampling of online social networks, Journal of Network and Computer Applications 41 (2014) 126–134.

[21] S. Even, Graph algorithms, Cambridge University Press, 2011.

[22] L. A. Goodman, Snowball sampling, The annals of mathematical statistics (1961) 148–170.

[23] L. Lovász, Random walks on graphs: A survey, Combinatorics, Paul erdos is eighty 2 (1) (1993) 1–46.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of state calculations by fast computing machines, The journal of chemical physics 21 (6) (1953) 1087–1092.

[25] M. H. Hansen, W. N. Hurwitz, On the theory of sampling from finite populations, The Annals of Mathematical Statistics 14 (4) (1943) 333–362.

[26] M. J. Salganik, D. D. Heckathorn, Sampling and estimation in hidden populations using respondent-driven sampling, Sociological methodology 34 (1) (2004) 193–240.

[27] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 635–644.

[28] Z. Zhou, N. Zhang, Z. Gong, G. Das, Faster random walks by rewiring online social networks on-the-fly, ACM Transactions on Database Systems (TODS) 40 (4) (2016) 26.

[29] A. Nazi, Z. Zhou, S. Thirumuruganathan, N. Zhang, G. Das, Walk, not wait: Faster sampling over online social networks, Proceedings of the VLDB Endowment 8 (6) (2015) 678–689.

[30] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 390–403.

[31] E. Voudigari, N. Salamanos, T. Papageorgiou, E. J. Yannakoudakis, Rank degree: An efficient algorithm for graph sampling, in: Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on, IEEE, 2016, pp. 120–129.

[32] B. Davis, R. Gera, G. Lazzaro, B. Y. Lim, E. C. Rye, The marginal benefit of monitor placement on networks, in: Complex Networks VII, Springer, 2016, pp. 93–104.

[33] M. Kurant, C. T. Butts, A. Markopoulou, Graph size estimation, arXiv preprint arXiv:1210.0460.

[34] S. J. Hardiman, L. Katzir, Estimating clustering coefficients and size of social networks via random walk, in: Proceedings of the 22nd international conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2013, pp. 539–550.

[35] E. Cem, K. Sarac, Estimation of structural properties of online social networks at the extreme, Computer Networks 108 (2016) 323–344.

[36] C. Musco, H.-H. Su, N. Lynch, Ant-inspired density estimation via random walks, in: Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, ACM, 2016, pp. 469–478.

[37] R. Lucchese, D. Varagnolo, Networks cardinality estimation using order statistics, in: American Control Conference (ACC), 2015, IEEE, 2015, pp. 3810–3817.

[38] L. Chen, A. Karbasi, F. W. Crawford, Estimating the size of a large network and its communities from a random sample, in: Advances in Neural Information Processing Systems, 2016, pp. 3072–3080.

[39] S. Ye, S. F. Wu, Estimating the size of online social networks, International Journal of Social Computing and Cyber-Physical Systems 1 (2) (2011) 160–179.

[40] C. Cooper, T. Radzik, Y. Siantos, A fast algorithm to find all high degree vertices in power law graphs, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 1007–1016.

[41] A. Marchetti-Spaccamela, On the estimate of the size of a directed graph, in: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 1988, pp. 317–326.

[42] A. Dasgupta, R. Kumar, T. Sarlos, On estimating the average degree, in: Proceedings of the 23rd international conference on World wide web, ACM, 2014, pp. 795–806.

[43] T. Eden, D. Ron, C. Seshadhri, Sublinear time estimation of degree distribution moments: The arboricity connection, arXiv preprint arXiv:1604.03661.

[44] J. Lu, D. Li, Sampling online social networks by random walk, in: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, ACM, 2012, pp. 33–40.

[45] E. Cem, K. Sarac, Estimating the size and average degree of online social networks at the extreme, in: Communications (ICC), 2015 IEEE International Conference on, IEEE, 2015, pp. 1268–1273.

[46] E. Cem, K. Sarac, Average degree estimation under ego-centric sampling design, in: Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on, IEEE, 2016, pp. 152–157.

[47] B. Ribeiro, D. Towsley, On the estimation accuracy of degree distributions from graph sampling, in: Decision and Control (CDC), 2012 IEEE 51st Annual Conference on, IEEE, 2012, pp. 5240–5247.

[48] B. Ribeiro, P. Wang, F. Murai, D. Towsley, Sampling directed graphs with random walks, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1692–1700.

[49] A. Saxena, V. Malik, S. Iyengar, Rank me thou shalln't compare me, arXiv preprint arXiv:1511.09050.

[50] 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.

[51] A. Saxena, V. Malik, S. Iyengar, Estimating the degree centrality ranking of a node, arXiv preprint arXiv:1511.05732.

[52] M. Gjoka, M. Kurant, C. T. Butts, A. Markopoulou, Walking in Facebook: A case study of unbiased sampling of OSNs, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[53] R. A. Rossi, N. K. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
URL `http://networkrepository.com`

[54] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, Y. Elovici, Link prediction in social networks using computationally efficient topological

features, in: IEEE Third International Confernece on Social Computing (SocialCom), IEEE, 2011, pp. 73–80.

[55] Catster friendships network dataset – KONECT (Oct. 2016).
URL http://konect.uni-koblenz.de/networks/petster-friendships-cat

[56] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowledge and Information Systems 42 (1) (2015) 181–213.

[57] R. Zafarani, H. Liu, Users joining multiple sites: Distributions and patterns.

[58] T. Hogg, K. Lerman, Social dynamics of digg, EPJ Data Science 1 (1) (2012) 1–26.

[59] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1) (2007) 2.

[60] A. L. Traud, E. D. Kelsic, P. J. Mucha, M. A. Porter, Comparing community structure to characteristics in online collegiate social networks, SIAM Rev. 53 (3) (2011) 526–543.

[61] A. L. Traud, P. J. Mucha, M. A. Porter, Social structure of Facebook networks, Phys. A 391 (16) (2012) 4165–4180.

[62] R. Zafarani, H. Liu, Social computing data repository at ASU (2009).
URL http://socialcomputing.asu.edu

[63] 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.

[64] J. J. McAuley, J. Leskovec, Learning to discover social circles in ego networks., in: NIPS, Vol. 2012, 2012, pp. 548–56.

[65] P. Boldi, S. Vigna, The WebGraph framework I: Compression techniques, in: Proc. of the Thirteenth International World Wide Web Conference (WWW 2004), ACM Press, Manhattan, USA, 2004, pp. 595–601.

[66] I. Konstas, V. Stathopoulos, J. M. Jose, On social networks and collaborative recommendation, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, 2009, pp. 195–202.

[67] R. Zafarani, H. Liu, Social computing data repository at ASU (2009). URL `http://socialcomputing.asu.edu`

[68] P. Erdős, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hungar. Acad. Sci 5 (1960) 17–61.

[69] M. De Choudhury, H. Sundaram, A. John, D. D. Seligmann, Social synchrony: Predicting mimicry of user actions in online social media, in: Computational Science and Engineering, 2009. CSE'09. International Conference on, Vol. 4, IEEE, 2009, pp. 151–158.

Appendix

I Results on real world scale-free networks

Table 6: Absolute and Weighted Error in the Estimated Ranking using all Methods on Real World Social Networks

Network	Type	Ref	Nodes	Edges	Avg Deg	PL Error		US Error		MH Error		RW Error	
						Abs	Wt	Abs	Wt	Abs	Wt	Abs	Wt
Friendster	Social	[53]	5689498	14067887	4.95	0.06	0.06	0.01	0.01	0.05	0.05	0.01	0.01
Academia	Social	[54]	200167	1022440	10.22	1.46	1.01	0.15	0.14	0.47	0.35	0.19	0.14
Dogster	Social	[53]	426485	8543321	40.06	1.27	0.95	0.1	0.09	0.34	0.28	0.07	0.06
Facebook1	Social	[61]	3097165	23667394	15.28	01.07	0.78	0.03	0.02	0.19	0.17	0.05	0.04
Gowalla	Social	[63]	196591	950327	9.67	01.05	0.66	0.12	0.11	0.5	0.41	0.12	0.1
Hyves	Social	[62]	1402673	2777419	3.96	0.2	0.15	0.02	0.02	0.07	0.07	0.01	0.01
Foursquare	Social	[62]	639014	3214985	10.06	1.1	0.95	0.08	0.07	0.52	0.45	0.06	0.05
Last.fm	Social	[66]	1191805	4519330	7.58	0.24	0.21	0.03	0.03	0.09	0.08	0.01	0.01
Livemocha	Social	[67]	104103	2193082	42.13	2.96	2.26	0.42	0.38	1.01	0.78	0.31	0.24
Delicious	Social	[53]	536108	1365961	5.1	0.31	0.25	0.05	0.05	0.17	0.15	0.05	0.04
Douban	Social	[53]	154908	327162	4.22	1.35	1.16	0.19	0.18	0.41	0.38	0.13	0.12
Actor	Collaboration	[8]	374511	15014839	80.18	3.48	2.31	0.15	0.14	0.51	0.41	0.27	0.21
DBLP	Collaboration	[56]	317080	1049866	6.62	1.25	01.09	0.12	0.11	0.43	0.34	0.21	0.16
Digg	Social	[69]	261489	1536577	11.75	0.59	0.54	0.12	0.12	0.42	0.39	0.1	0.09
Eu-Email	Communication	[59]	224832	339925	3.02	0.42	0.39	0.08	0.08	0.21	0.2	0.04	0.04
Gplus	Social	[64]	107614	12238285	227.45	5.93	4.64	0.46	0.41	21.08	2.34	1.09	01.07
Catster	Social	[53]	148826	5447464	73.21	2.43	11.02	0.25	0.23	1.1	0.9	0.25	0.2
Youtube	Social	[62]	1134885	2987623	5.27	0.14	0.11	0.03	0.03	0.1	0.09	0.02	0.02
Pokec	Social	[53]	1632803	22301964	27.32	2.74	1.78	0.05	0.04	0.14	0.1	0.07	0.05
Hollywood	Collaboration	[65]	1069126	56306653	105.33	2.54	1.91	0.08	0.07	0.3	0.26	0.15	0.12
Summary						1.51	1.14	0.13	0.12	0.50	0.41	0.16	0.13