Quick Detection of Nodes with Large Degrees

Konstantin Avrachenkov, Nelly Litvak, Marina Sokol, Don Towsley

Quick Detection of Nodes with Large Degrees*

Konstantin Avrachenkov[1], Nelly Litvak[2], Marina Sokol[1], and Don Towsley[3]

[1] INRIA, 2004 Route des Lucioles, Sophia-Antipolis, France
[2] University of Twente, the Netherlands
[3] University of Massachusetts Amherst, USA

Abstract. Our goal is to quickly find top k lists of nodes with the largest degrees in large complex networks. If the adjacency list of the network is known (not often the case in complex networks), a deterministic algorithm to find the top k list of nodes with the largest degrees requires an average complexity of $O(n)$, where n is the number of nodes in the network. Even this modest complexity can be very high for large complex networks. We propose to use the random walk based method. We show theoretically and by numerical experiments that for large networks the random walk method finds good quality top lists of nodes with high probability and with computational savings of orders of magnitude. We also propose stopping criteria for the random walk method which requires very little knowledge about the structure of the network.

1 Introduction

We are interested in quickly detecting nodes with large degrees in very large networks. Firstly, node degree is one of centrality measures used for the analysis of complex networks. Secondly, large degree nodes can serve as proxies for central nodes corresponding to the other centrality measures as betweenness centrality or closeness centrality [8, 9]. In the present work we restrict ourself to undirected networks or symmetrized versions of directed networks. In particular, this assumption is well justified in social networks. Typically, friendship or acquaintance is a symmetric relation. If the adjacency list of the network is known (not often the case in complex networks), a deterministic algorithm to find the top k list of nodes with the largest degrees requires an average complexity of $O(n)$, where n is the number of nodes in the network. We assume that the degree is available when accessing a node (if this is not the case, the complexity should be counted in terms of links). However, even linear complexity can be very high for very large, possibly varying, complex networks. In the present work we suggest using random walk based methods for detecting a small number of nodes with the largest degree. The main idea is that the random walk very quickly comes across large degree nodes. In our numerical experiments random walks outperform the standard deterministic algorithms by orders of magnitude in terms of

* This research was sponsored by INRIA Alcatel-Lucent Joint Lab, by the NSF under CNS-1065133, and the U.S. Army Research Laboratory under Cooperative Agreement W911NF-09-2-0053.

computational complexity. For instance, in our experiments with the web graph of the UK domain (about 18 500 000 nodes) the random walk method spends on average only about 5 400 steps to detect the largest degree node. Potential memory savings are also significant since the method does not require knowledge of the entire network. In many practical applications we do not need a complete ordering of the nodes and even can tolerate some errors in the top list of nodes. We observe that the random walk method obtains many nodes in the top list correctly and even those nodes that are erroneously placed in the top list have large degrees. Therefore, as typically happens in randomized algorithms [12, 13], we trade off exact results for very good approximate results or for exact results with high probability and gain significantly in computational efficiency.

The paper is organized as follows: in the next section we introduce our basic random walk with uniform jumps and demonstrate that it is able to quickly find large degree nodes. Then, in Section 3 using configuration model we provide an estimate for the necessary number of steps for the random walk. In Section 4 we propose stopping criteria that use very little information about the network. In Section 5 we show the benefits of allowing few erroneous elements in the top k list. Finally, we conclude the paper in Section 6.

2 Random walk with uniform jumps

Let us consider a random walk with uniform jumps which serves as a basic algorithm for quick detection of large degree nodes. The random walk with uniform jumps is described by the following transition probabilities [1]

$$p_{ij} = \begin{cases} \frac{\alpha/n+1}{d_i+\alpha}, \text{ if } i \text{ has a link to } j, \\ \frac{\alpha/n}{d_i+\alpha}, \quad \text{if } i \text{ does not have a link to } j, \end{cases} \tag{1}$$

where d_i is the degree of node i. The random walk with uniform jumps can be regarded as a random walk on a modified graph where all the nodes in the graph are connected by artificial edges with a weight α/n. The parameter α controls the rate of jumps. Introduction of jumps helps in a number of ways. As was shown in [1], it reduces the mixing time to stationarity. It also solves a problem encountered by a random walk on a graph consisting of two or more components, namely the inability to visit all nodes. The random walk with jumps also reduces the variance of the network function estimator [1]. This random walk resembles the PageRank random walk. However, unlike the PageRank random walk, the introduced random walk is reversible. One important consequence of the reversibility of the random walk is that its stationary distribution is given by a simple formula

$$\pi_i(\alpha) = \frac{d_i + \alpha}{2|E| + n\alpha} \qquad \forall i \in V, \tag{2}$$

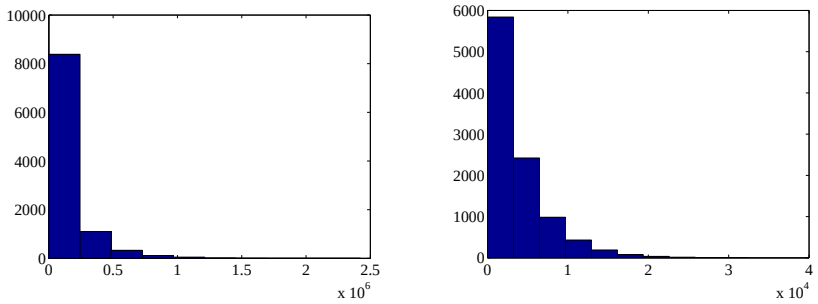
from which the stationary distribution of the unperturbed random walk can easily be retrieved. We observe that the modification preserves the monotonicity of

the stationary distribution with respect to the node degree, which is particularly important for our application.

We illustrate on several network examples how the random walk helps us quickly detect large degree nodes. We consider as examples one synthetic network generated by the preferential attachment rule and two natural large networks. The Preferential Attachment (PA) network combines 100 000 nodes. It has been generated according to the generalized preferential attachment mechanism [6]. The average degree of the PA network is two and the power law exponent is 2.5. The first natural example is the symmetrized web graph of the whole UK domain crawled in 2002 [4]. The UK network has 18 520 486 nodes and its average degree is 28.6. The second natural example is the network of co-authorships of DBLP [5]. Each node represents an author and each link represents a co-authorship of at least one article. The DBLP network has 986 324 nodes and its average degree is 6.8.

We carry out the following experiment: we initialize the random walk (1) at a node chosen according to the uniform distribution and continue the random walk until we hit the largest degree node. The largest degrees for the PA, UK and DBLP networks are 138, 194 955, and 979, respectively. For the PA network we have made 10 000 experiments and for the UK and DBLP networks we performed 1 000 experiments (these networks were too large to perform more experiments).

In Figue 1 we plot the histograms of hitting times for the PA network. The first remarkable observation is that when $\alpha = 0$ (no restart) the average hitting time, which is equal to 123 000 steps, is nearly three orders of magnitude larger than 3 720, the hitting time when $\alpha = 2$. The second remarkable observation is that 3 720 is of the same order of magnitude as the value $1/\pi_{max}(\alpha) = (2|E| + n\alpha)/(d_{max} + \alpha) = 2\,857$, which corresponds to the average return time to the largest degree node in the random walk with jumps.



(a) $\alpha = 0$ (b) $\alpha = 2$

Fig. 1. Histograms of hitting times in the PA network.

We were not able to collect a representative number of experiments for the UK and DBLP networks when $\alpha = 0$. The reason for this is that the random walk gets stuck either in disconected or weakly connected components of the networks. For the UK network we were able to make 1 000 experiments with $\alpha = 0.001$ and obtain the average hitting time 30 750. Whereas if we take $\alpha = 28.6$ for the UK network, we obtain the average hitting time 5 800. Note that the expected return time to the largest degree node in the UK network is given by $1/\pi_{max}(\alpha) = (2|E| + n\alpha)/(d_{max} + \alpha) = 5\,432$. For the DBLP graph we conducted 1 000 experiments with $\alpha = 0.00001$ and obtained an average hitting time of 41 131. Whereas if we take $\alpha = 6.8$, we obtain an average hitting time of 14 200. The expected return time to the largest degree node in the DBLP network is given by $1/\pi_{max}(\alpha) = (2|E| + n\alpha)/(d_{max} + \alpha) = 13\,607$. The two natural network examples confirm our guess that the average hitting time for the largest degree node is fairly close to the average return time to the largest degree node. Let us also confirm our guess with asymptotic analysis.

Theorem 1 *Without loss of generality, index the nodes such that node 1 has the largest degree,* $(1, i) \in E, i = 2, ..., s, s = d_1 + 1$*, and let* ν *denote the initial distribution of the random walk with jumps. Then, the expected hitting time to node 1 starting from any initial distribution is given by*

$$E_\nu[T_1] = \frac{\sum_{i=2}^{n} d_i + (n-1)\alpha}{d_1 + 2\alpha(1 - 1/n)} + o\left(\min_{i=2,...,s}\{(d_i + \alpha), n\}\right), \tag{3}$$

Proof: The expected hitting time from distribution ν to node 1 is given by the formula

$$E_\nu[T_1] = \nu[I - P_{-1}]^{-1}\underline{1}, \tag{4}$$

where P_{-1} is a taboo probability matrix (i.e., matrix P with the 1-st row and 1-st column removed). The matrix P_{-1} is substochastic but is very close to stochastic. Let us represent it as a stochastic matrix minus some perturbation term:

$$P_{-1} = \tilde{P} - \varepsilon Q = \tilde{P} - \begin{bmatrix} \frac{1+2\alpha/n}{d_2+\alpha} & 0 & & & & 0 \\ 0 & \ddots & & & & \\ & & \frac{1+2\alpha/n}{d_s+\alpha} & & & \\ & & & \frac{2\alpha/n}{d_{s+1}+\alpha} & & \\ & & & & \ddots & 0 \\ 0 & & & & 0 & \frac{2\alpha/n}{d_n+\alpha} \end{bmatrix}$$

We add missing probability mass to the diagonal of $\tilde{P}$, which corresponds to an increase in the weights for self-loops. The matrix $\tilde{P}$ represents a reversible Markov chain with the stationary distribution

$$\tilde{\pi}_j = \frac{d_j + \alpha}{\sum_{i=2}^{n} d_i + (n-1)\alpha}.$$

Now we can use the following result from the perturbation theory (see Lemma 1 in [2]):

$$[I - \tilde{P} + \varepsilon Q]^{-1} = \frac{\underline{1}\tilde{\pi}}{\tilde{\pi}(\varepsilon Q)\underline{1}} + X_0 + \varepsilon X_1 + \dots , \tag{5}$$

where $\tilde{\pi}$ is the stationary distribution of the stochastic matrix $\tilde{P}$. In our case, the quantity $\max_{i=2,\dots,s}\{1/(d_i+\alpha), 1/n\}$ will play the role of ε. We apply the series (5) to approximate the expected hitting time. Towards this goal, we calculate

$$\tilde{\pi}(\varepsilon Q)\underline{1} = \sum_{j=2}^{n} \tilde{\pi}_j \varepsilon q_{jj}$$

$$= \sum_{j=2}^{s} \frac{d_j+\alpha}{\sum_{i=2}^{n} d_i + (n-1)\alpha} \frac{1+2\alpha/n}{d_j+\alpha} + \sum_{j=s+1}^{n} \frac{d_j+\alpha}{\sum_{i=2}^{n} d_i + (n-1)\alpha} \frac{2\alpha/n}{d_j+\alpha}$$

$$= \frac{d_1(1+2\alpha/n) + (n-d_1-1)(2\alpha/n)}{\sum_{i=2}^{n} d_i + (n-1)\alpha} = \frac{d_1 + 2\alpha(1-1/n)}{\sum_{i=2}^{n} d_i + (n-1)\alpha}.$$

Observing that $\nu\underline{1}\tilde{\pi}\underline{1} = 1$, we obtain (3).

□

Indeed, the asymptotic expression (3) is very close to $(2|E| + n\alpha)/(d_1 + \alpha)$, which is the expected return time to node 1.

Based on the notion of the hitting time we propose an efficient method for quick detection of the top k list of largest degree nodes. The algorithm maintains a top k candidate list. Note that once one of the k nodes with the largest degrees appears in this candidate list, it remains there subsequently. Thus, we are interested in hitting events. We propose the following algorithm for detecting the top k list of largest degree nodes.

Algorithm 1 Random walk with jumps and candidate list

1. *Set k, α and m.*
2. *Execute a random walk step according to* (1)*. If it is the first step, start from the uniform distribution.*
3. *Check if the current node has a larger degree than one of the nodes in the current top k candidate list. If it is the case, insert the new node in the top-k candidate list and remove the worst node out of the list.*
4. *If the number of random walk steps is less than m, return to Step 2 of the algorithm. Stop, otherwise.*

The value of parameter α is not crucial. In our experiments, we have observed that as long as the value of α is neither too small nor not too big, the algorithm performs well. A good option for the choice of α is a value slightly smaller than the average node degree. Let us explain this choice by calculating a probability of jump in the steady state

$$\sum_{j=1}^{n} \pi_j(\alpha) \frac{\alpha}{d_j+\alpha} = \sum_{j=1}^{n} \frac{d_j+\alpha}{2|E|+n\alpha} \frac{\alpha}{d_j+\alpha} = \frac{n\alpha}{2|E|+n\alpha} = \frac{\alpha}{2|E|/n+\alpha}.$$

If α is equal to $2|E|/n$, the average degree, the random walk will jump in the steady state on average every two steps. Thus, if we set α to the average degree or to a slightly smaller value, on one hand the random walk will quickly converge to the steady state and on the other hand we will not sample too much from the uniform distribution.

The number of random walk steps, m, is a crucial parameter. Our experiments indicate that we obtain a top k list with many correct elements with high probability if we take the number of random walk steps to be twice or thrice as large as the expected hitting time of the nodes in the top k list. From Theorem 1 we know that the hitting time of the large degree node is related to the value of the node's degree. Thus, the problem of choosing m reduces to the problem of estimating the values of the largest degrees. We address this problem in the following section.

3 Estimating the largest degrees in the configuration network model

The estimations for the values of the largest degrees can be derived in the configuration network model [7] with a power law degree distribution. In some applications the knowledge of the power law parameters might be available to us. For instance, it is known that web graphs have power law degree distribution and we know typical ranges for the power law parameters.

We assume that the node degrees $D_1, \ldots, D_n$ are i.i.d. random variables with a power law distribution F and finite expectation $E[D]$. Let us determine the number of links contained in the top k nodes. Denote

$$F(x) = P[D \leq x], \quad \bar{F}(x) = 1 - F(x), \quad x \geq 0.$$

Further let $D_{(1)} \geq \ldots \geq D_{(n)}$ be the order statistics of $D_1, \ldots, D_n$. Under the assumption that D_j's obey a power law, we use the results from the extreme value theory as presented in [11], to state that there exist sequences of constants (a_n) and (b_n) and a constant δ such that

$$\lim_{n\to\infty} n\bar{F}(a_n x + b_n) = (1 + \delta x)^{-1/\delta}. \tag{6}$$

This implies the following approximation for high quantiles of F, with exceedance probability close to zero [11]:

$$x_p \approx a_n \frac{(pn)^{-\delta} - 1}{\delta} + b_n.$$

For the jth largest degree, where $j = 2, \ldots, k$, the estimated exceedance probability equals $(j-1)/n$, and thus we can use the quantile $x_{(j-1)/n}$ to approximate the degree $D_{(j)}$ of this node:

$$D_{(j)} \approx a_n \frac{(j-1)^{-\delta} - 1}{\delta} + b_n. \tag{7}$$

The sequences (a_n) and (b_n) are easy to find for a given shape of the tail of F. Below we derive the corresponding results for the commonly accepted Pareto tail distribution of D, that is,

$$\bar{F}(t) = Cx^{-\gamma} \quad \text{for } x > x', \tag{8}$$

where $\gamma > 1$ and x' is a fixed sufficiently large number so that the power law degree distribution is observed for nodes with degree larger than x'. In that case we have

$$\lim_{n\to\infty} n\bar{F}(a_n x+b_n) = \lim_{n\to\infty} nC(a_n x+b_n)^{-\gamma} = \lim_{n\to\infty} (C^{-1/\gamma}n^{-1/\gamma}a_n x+C^{-1/\gamma}n^{-1/\gamma}b_n)^{-\gamma},$$

which directly gives (6) with

$$\delta = 1/\gamma, \quad a_n = \delta C^{\delta} n^{\delta}, \quad b_n = C^{\delta} n^{\delta}. \tag{9}$$

Substituting (9) into (7) we obtain the following prediction for $D_{(j)}$, $j = 2, \ldots, k$, in the case of the Pareto tail of the degree distribution:

$$D_{(j)} \approx n^{1/\gamma}[C^{1/\gamma}(j-1)^{-1/\gamma} - C^{1/\gamma} + 1]. \tag{10}$$

It remains to find an approximation for $D_{(1)}$, the maximal degree in the graph. From the extreme value theory it is well known that if $D_1, \ldots, D_n$ obey a power law then

$$\lim_{n\to\infty} P\left(\frac{D_{(1)} - b_n}{a_n} \le x\right) = H_\delta(x) = \exp(-(1+\delta x)^{-1/\delta}),$$

where, for Pareto tail, a_n, b_n and δ are defined in (9). Thus, as an approximation for the maximal node degree we can choose $a_n x + b_n$ where x can be chosen as either an expectation, a median or a mode of $H_\delta(x)$. If we choose the mode, $((1+\delta)^{-\delta} - 1)/\delta$, then we obtain an approximation, which is smaller than the one for the 2nd largest degree. Further, the expectation $(\Gamma(1-\delta) - 1)/\delta$ is very sensitive to the value of $\delta = 1/\gamma$, especially when γ is close to one, which is often the case in complex networks. Besides, the parameter γ is hard to estimate with high precision. Thus, we choose the median $(\log(2))^{-\delta} - 1)/\delta$, which yields

$$D_{(1)} \approx a_n \frac{(\log(2))^{-\delta} - 1}{\delta} + b_n = n^{1/\gamma}[C^{1/\gamma}(\log(2))^{-1/\gamma} - C^{1/\gamma} + 1]. \tag{11}$$

For instance, in the PA network $\gamma = 2.5$ and $C = 3.7$, which gives according to (11) $D_{(1)} \approx 127$. (This is a good prediction even though the PA network is not generated according to the configuration model. We also note that even though the extremum distribution in the preferential attachment model is different from that of the configuration model their ranges seem to be very close [10].) This in turn suggests that for the PA network m should be chosen in the range 6 000-18 000 if $\alpha = 2$. As we can see from Figure 2 this is indeed a good range for the number of random walk steps. In the UK network $\gamma = 1.7$ and $C = 90$, which gives $D_{(1)} \approx 82\,805$ and suggests a range of 20 000-30 000 for m if $\alpha = 28.6$. Figure 3 confirms that this is a good choice. The degree distribution of the DBLP network does not follow a power law so we cannot apply the above reasoning to it.

4 Stopping criteria

Suppose now that we do not have any information about the range for the largest k degrees. In this section we design stopping criteria that do not require knowledge about the structure of the network. As we shall see, knowledge of the order of magnitude of the average degree might help, but this knowledge is not imperative for a practical implementation of the algorithm.

Let us now assume that node j can be sampled independently with probability $\pi_j(\alpha)$ as in (2). There are at least two ways to achieve this practically. The first approach is to run the random walk for a significant number of steps until it reaches the stationary distribution. If one chooses α reasonably large, say the same order of magnitude as the average degree, then the mixing time becomes quite small [1] and we can be sure to reach the stationary distribution in a small number of steps. Then, the last step of a run of the random walk will produce an i.i.d. sample from a distribution very close to (2). The second approach is to run the random walk uninterruptedly, also with a significant value of α, and then perform Bernoulli sampling with probability q after a small initial transient phase. If q is not too large, we shall have nearly independent samples following the stationary distribution (2). In our experiment, $q \in [0.2, 0.5]$ gives good results when α has the same order of magnitude as the average degree.

We now estimate the probability of detecting correctly the top k list of nodes after m i.i.d. samples from (2). Denote by X_i the number of hits at node i after m i.i.d. samples. We note that if we use the second approach to generate i.i.d. samples, we spend approximately m/q steps of the random walk. We correctly detect the top k list with the probability given by the multinomial distribution

$$P[X_1 \geq 1, ..., X_k \geq 1] =$$

$$\sum_{i_1 \geq 1, ..., i_1 \geq 1} \frac{m!}{i_1! \cdots i_k! (m - i_1 - ... - i_k)!} \pi_1^{i_1} \cdots \pi_k^{i_k} (1 - \sum_{i=1}^{k} \pi_i)^{m - i_1 - ... - i_k}$$

but it is not feasible for any realistic computations. Therefore, we propose to use the Poisson approximation. Let Y_j, $j = 1, ..., n$ be independent Poisson random variables with means $\pi_j m$. That is, the random variable Y_j has the following probability mass function $P[Y_j = r] = e^{-m\pi_j}(m\pi_j)^r / r!$. It is convenient to work with the complementary event of not detecting correctly the top k list. Then, we have

$$P[\{X_1 = 0\} \cup ... \cup \{X_k = 0\}] \leq 2P[\{Y_1 = 0\} \cup ... \cup \{Y_k = 0\}]$$

$$= 2(1 - P[\{Y_1 \geq 1\} \cap ... \cap \{Y_k \geq 1\}]) = 2(1 - \prod_{j=1}^{k} P[\{Y_j \geq 1\}])$$

$$= 2(1 - \prod_{j=1}^{k} (1 - P[\{Y_j = 0\}])) = 2(1 - \prod_{j=1}^{k} (1 - e^{-m\pi_j})) =: a, \qquad (12)$$

where the first inequality follows from [12, Thm 5.10]. In fact, in our numerical experiments we observed that the factor 2 in the first inequality is very conservative. For large values of m, the Poisson bound works very well as proper approximation.

For example, if we would like to obtain the top 10 list with at most 10% probability of error, we need to have on average 4.5 hits per each top element. This can be used to design the stopping criteria for our random walk algorithm. Let $\bar{a} \in (0,1)$ be the admissible probability of an error in the top k list. Now the idea is to stop the algorithm after m steps when the estimated value of a for the first time is lower than the critical number $\bar{a}$. Clearly,

$$\hat{a}_m = 2(1 - \prod_{j=1}^{k}(1 - e^{-X_j}))$$

is the maximum likelihood estimator for a, so we would like to choose m such that $\hat{a}_m \leq \bar{a}$. The problem, however, is that we do not know which X_j's are the realisations of the number of visits to the top k nodes. Then let $X_{j_1}, ..., X_{j_k}$ be the number of hits to the current elements in the top k candidate list and consider the estimator

$$\hat{a}_{m,0} = 2(1 - \prod_{i=1}^{k}(1 - e^{-X_{j_i}})),$$

which is the maximum likelihood estimator of the quantity

$$2(1 - \prod_{i=1}^{k}(1 - e^{-m\pi_{j_i}})) \geq a.$$

(Here π_{j_i} is a stationary probability of the node with the score X_{j_i}, $i = 1, \ldots, k$). The estimator $\hat{a}_{m,0}$ is computed without knowledge of the top k nodes or their degrees, and it is an estimator of an upper bound of the estimated probability that there are errors in the top k list. This leads to the following stopping rule.
Stopping rule 0. *Stop at $m = m_0$, where*

$$m_0 = \arg\min\{m : \hat{a}_{m,0} \leq \bar{a}\}.$$

The above stopping criterion can be simplified even further to avoid computation of $\hat{a}_{m,0}$. Since

$$\hat{a}_{m,1} := 2(1 - (1 - e^{-X_{j_k}})^k) \geq \hat{a}_{m,0} \geq \hat{a},$$

where X_{j_k} is the number of hits of the worst element in the candidate list. The inequality $\hat{a}_m \leq \bar{a}$ is guaranteed if $\hat{a}_{m,1} \leq \bar{a}$. This leads to the following stopping rule for the random walk algorithm.

Stopping rule 1. *Compute* $x_0 = \arg\min\{x \in \mathbb{N} : (1 - e^{-x})^k \geq 1 - \bar{a}/2.\}$ *Stop at*

$$m_1 = \arg\min\{m : X_{j_k} = x_0\}.$$

We have observed in our numerical experiments that we obtain the best trade off between the number of steps of the random walk and the accuracy if we take α around the average degree and the sampling probability q around 0.5. Specifically, if we take $\bar{a}/2 = 0.15$ ($x_0 = 4$) in Stopping rule 1 for top 10 list, we obtain 87% accuracy for an average of 47 000 random walk steps for the PA network; 92% accuracy for an average of 174 468 random walk steps for the DBLP network; and 94% accuracy for an average of 247 166 random walk steps for the UK network. We have averaged over 1000 experiments to obtain tight confidence intervals.

5 Relaxation of top *k* lists

In the stopping criteria of the previous section we have strived to detect all nodes in the top k list. This costs us a lot of steps of the random walk. We can significantly gain in performance by relaxing this strict requirement. For instance, we could just ask for list of k nodes that contains 80% of top k nodes [3]. This way we can take an advantage of a generic 80/20 rule that 80% of result can be achieved with 20% of effort.

Let us calculate the expected number of top k elements observed in the candidate list up to trial m. Define by X_j the number of times we have observed node j after m trials and

$$H_j = \begin{cases} 1, \text{ node } j \text{ has been observed at least once,} \\ 0, \text{ node } j \text{ has not been observed.} \end{cases}$$

Assuming we sample in i.i.d. fashion from the distribution (2), we can write

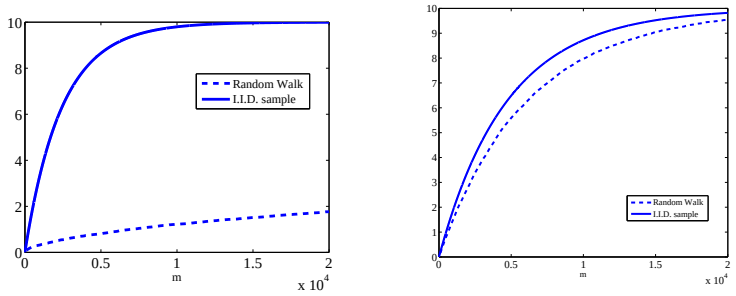
$$E[\sum_{j=1}^{k} H_j] = \sum_{j=1}^{k} E[H_j] = \sum_{j=1}^{k} P[X_j \geq 1] = \sum_{j=1}^{k} (1 - P[X_j = 0]) = \sum_{j=1}^{k} (1 - (1 - \pi_j)^m). \tag{13}$$

In Figure 2 we plot $E[\sum_{j=1}^{k} H_j]$ (the curve "I.I.D. sample") as a function of m for $k = 10$ for the PA network with $\alpha = 0$ and $\alpha = 2$. In Figure 3 we plot $E[\sum_{j=1}^{k} H_j]$ as a function of m for $k = 10$ for the UK network with $\alpha = 0.001$ and $\alpha = 28.6$. The results for the UK and DBLP networks are similar in spirit.

Here again we can use the Poisson approximation

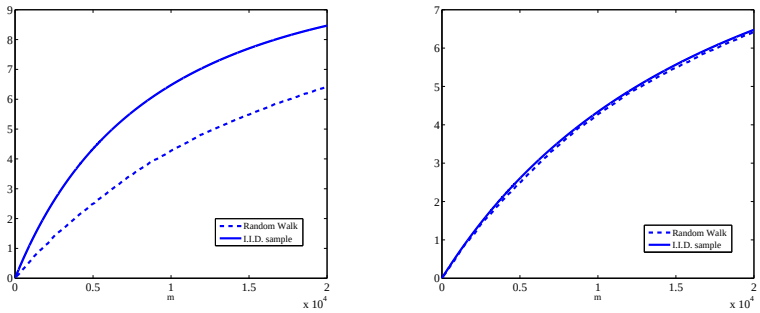
$$E[\sum_{j=1}^{k} H_j] \approx \sum_{j=1}^{k} (1 - e^{-m\pi_j}).$$

In fact, the Poisson approximation is so good that if we plot it on Figures 2 and 3, it nearly covers exactly the curves labeled "I.I.D. sample", which correspond to



(a) $\alpha = 0$ (b) $\alpha = 2$

Fig. 2. Average number of correctly detected elements in top-10 for PA.



(a) $\alpha = 0.001$ (b) $\alpha = 28.6$

Fig. 3. Average number of correctly detected elements in top-10 for UK.

the exact formula (13). Similarly to the previous section, we can propose stopping criteria based on the Poisson approximation. Denote

$$b_m = \sum_{i=1}^{k} (1 - e^{-X_{j_i}}).$$

Stopping rule 2. *Stop at $m = m_2$, where*

$$m_2 = \arg\min\{m : b_m \geq \bar{b}\}.$$

Now if we take $\bar{b} = 7$ in Stopping rule 2 for top-10 list, we obtain on average 8.89 correct elements for an average of 16 725 random walk steps for the PA network; we obtain on average 9.28 correct elements for an average of 66 860 random walk steps for the DBLP network; and we obtain on average 9.22 correct

elements for an average of 65 802 random walk steps for the UK network. (We have averaged over 1000 experiments for each network.) This makes for the UK network the gain of more than two orders of magnitude in computational complexity with respect to the deterministic algorithm.

6 Conclusions and future research

We have proposed the random walk method with the candidate list for quick detection of largest degree nodes. We have also supplied stopping criteria which do not require knowledge of the graph structure. In the case of large networks, our algorithm finds top k list of largest degree nodes with few mistakes with the running time orders of magnitude faster than the deterministic algorithms. In future research we plan to obtain estimates for the required number of steps for various types of complex networks.

References

1. K. Avrachenkov, B. Ribeiro and D. Towsley, "Improving random walk estimation accuracy with uniform restarts", in Proceedings of WAW 2010, also Springer LNCS v.6516, pp.98-109, 2010.
2. K. Avrachenkov, V. Borkar and D. Nemirovsky, "Quasi-stationary distributions as centrality measures for the giant strongly connected component of a reducible graph", *Journal of Comp. and Appl. Mathematics*, v.234, pp.3075-3090, 2010.
3. K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova and M. Sokol, "Quick detection of top-k personalized pagerank lists", in Proceedings of WAW 2011.
4. P. Boldi and S. Vigna, "The WebGraph framework I: Compression techniques", in Proceedings of WWW 2004.
5. P. Boldi, M. Rosa, M. Santini and S. Vigna, "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks", in Proceedings of WWW 2011.
6. S.N. Dorogovtsev, J.F.F. Mendes and A.N. Samukhin, "Structure of growing networks: Exact solution of the Barabasi-Albert model", *Phys. Rev. Lett.*, v.85, pp.4633-4636, 2000.
7. R. van der Hofstad, *Random graphs and complex networks*, Lechture Notes, Available at *http://www.win.tue.nl/rhofstad/NotesRGCN.pdf*, 2009.
8. Y. Lim, D.S. Menasche, B. Ribeiro, D. Towsley and P. Basu, "Online estimating the k central nodes of a network", in Proceedings of IEEE NSW 2011.
9. A.S. Maiya and T.Y. Berger-Wolf, "Online sampling of high centrality individuals in social networks", in Proceedings of PAKDD 2010.
10. A.A. Moreira, J.S. Andrade Jr. and L.A.N. Amaral, "Extremum statistics in scale-free network models", *Phys. Rev. Lett.*, v.89, 268703 4 pages, 2002.
11. G. Matthys and J. Beirlant, "Estimating the extreme value index and high quantiles with exponential regression models", *Statistica Sinica*, v.13, no.3, pp.853-880, 2003.
12. M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.
13. R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.