NYU CS-GY 6763: Algorithmic Machine Learning and Data Science Fall 2023

Finding d_{max} on a network using random walks

Team: [redacted]

Introduction

As our world becomes increasingly interconnected, the study of graphs and networks provides a deeper comprehension of the intricate relationships inherent in complex systems. From unraveling the structure of social interactions to modeling biological pathways and analyzing information flow in technological networks, the versatility of graph theory provides a powerful lens through which to comprehend real-world phenomena (Barnes and Harary, 1983; Wellman, 1983; Bornholdt and Schuster, 2001). In particular, studying different characteristics of a network's topology can reveal important aspects governing the interaction mechanisms of the system it describes. For instance, the study of clustering coefficients helps us quantify the tendency of nodes to form clusters or tightly interconnected groups, which indicate the formation of communities within the graph. In addition, this helps us uncover the modular structure and resilience of networks in the face of node removal or perturbations.

Due to the significant computational resources and time required to analyze real-world networks, one efficient approach used for calculating clustering coefficients are random walks (da Fontoura Costa and Travieso, 2007; Hardiman and Katzir, 2013). Random walks are stochastic processes that navigate through a graph by moving from one node to another based on random choices. This approach provides good estimators for desired metrics without the need of analyzing the whole network (Lovász, 1993; Xia et al., 2019).

In this study, we build on the ideas of Hardiman and Katzir (2013) and we investigate whether it is possible to bound the number of steps needed on a random walk before being able to predict the node with the highest degree in the network, which we denote as $v_{d_{max}}$, with high probability. Furthermore, our approach combines the use of random walks with random networks which serve as a baseline for understanding the structural properties of their empirical counterparts providing insights into the emergence of features such as connectivity, average path length, and clustering (Chen and Chen, 2007). While previous work has examined probabilistically predicting the maximum degree in a network (Schank and Wagner, 2005; Saramäki et al., 2007), to the best of our knowledge, no paper has investigated methods to retrieve the node in question. Finding $v_{d_{max}}$ can have significant relevance for people analyzing a given network in different scenarios. For instance, one may want to know who is the user with the most connections or what power station is the biggest liability in a power grid. Of course, the degree of the node is not the sole measure of it's importance, with other measures of centrality also being used widely (De-Marcos et al., 2016; Chakraborty et al., 2017; Farooq et al., 2018), however, we focus our research direction on degree centrality.

Random and scale-free networks

We begin by elaborating on the properties of random networks, their construction, and the potential efficacy of the random walk approach for the question of finding $v_{d_{max}}$. Firstly, a random network $G_r(n, p)$ is fully described using only two properties; its number of nodes n, and the probability that two nodes are connected via an edge p (Erdős et al., 1959; 1960). In such a network, the degree of each node is the sum of an indicator function denoting whether a given node is connected to another node in the network. More formally, we can see that:

$$\mathbb{E}[d_i] = \mathbb{E}\left[\sum_{j=1}^{n} \mathbb{1}[A_{v_i,v_j} = 1]\right] = np$$

Here, A is the adjacency matrix of the network. As the probability of two nodes being connected is equal to p, the expected degree of a given node v_i is equal to np. As such, the degree distribution of a random network is that of a Binomial distribution, as illustrated in Figure 1A. Here, the red line denotes the degree distribution of 1000 different random networks with $n = 1000, p = 0.25$. Also illustrated in the figure is the distribution of the maximum degrees in each of these 1000 networks, shown by the solid blue line. As we can see, the maximum degree of a random network does not deviate considerably from its expected degree (shown with a red dashed line). Furthermore, the maximum degree does not exhibit a high variance, falling close to $1.178 * \mathbb{E}[d_i]$. Considering these attributes, we would expect that it would be relatively difficult for a random walk to locate $v_{d_{max}}$ reliably in a small number of steps since all nodes in the network have a degree close to their expectation.

Scale-free networks, on the other hand, do not demonstrate this quality. In many of the networks that govern real-world interactions, new nodes in a network prefer to link to more well-connected nodes, a process which is termed "preferential attachment". For instance consider citation networks, where the more cited a given paper is, the more likely one is to read it, and subsequently cite it in their own work. Among the many models which aim to capture this interaction pattern is the Barabasi Albert model (Barabási and Bonabeau, 2003). In this model, a clique of size m is created, and every subsequent node added to the network establishes m new edges to the network. Each of these m edges is connected to a given node in the network with a probability proportional to its degree. As such, nodes with higher degree accumulate more edges over time. The degree distribution of such a network can be seen in Figure 1C. As can be seen, the distribution of d_{max} is far less concentrated than in a random network, and is far greater than the expected degree of such a network. The degree distribution induced by a Barabasi Albert (BA) network is said to followed a power-law degree distribution which is independent of both the number of nodes in the initial clique m and the number of nodes in the network n.

Figure 1 **(B)** illustrates the shortest paths between any two nodes in a random network $G_r(100, 0.25)$. As can be seen, while there is a slight trend in increasing shortest path lengths as the sum of the degrees of the pair of nodes decreases, this trend is not as prominent as that of a BA network $G_{sf}(n = 100, m = 5)$ illustrated in Figure 1D. This suggests that a random walk on a BA network may be more conducive towards finding the node of maximal degree, as the shortest path between any node and the node of maximum degree is shorter on average in a BA network (1.53 steps) than a random network (1.64 steps). While a random walk would not necessarily walk along the shortest

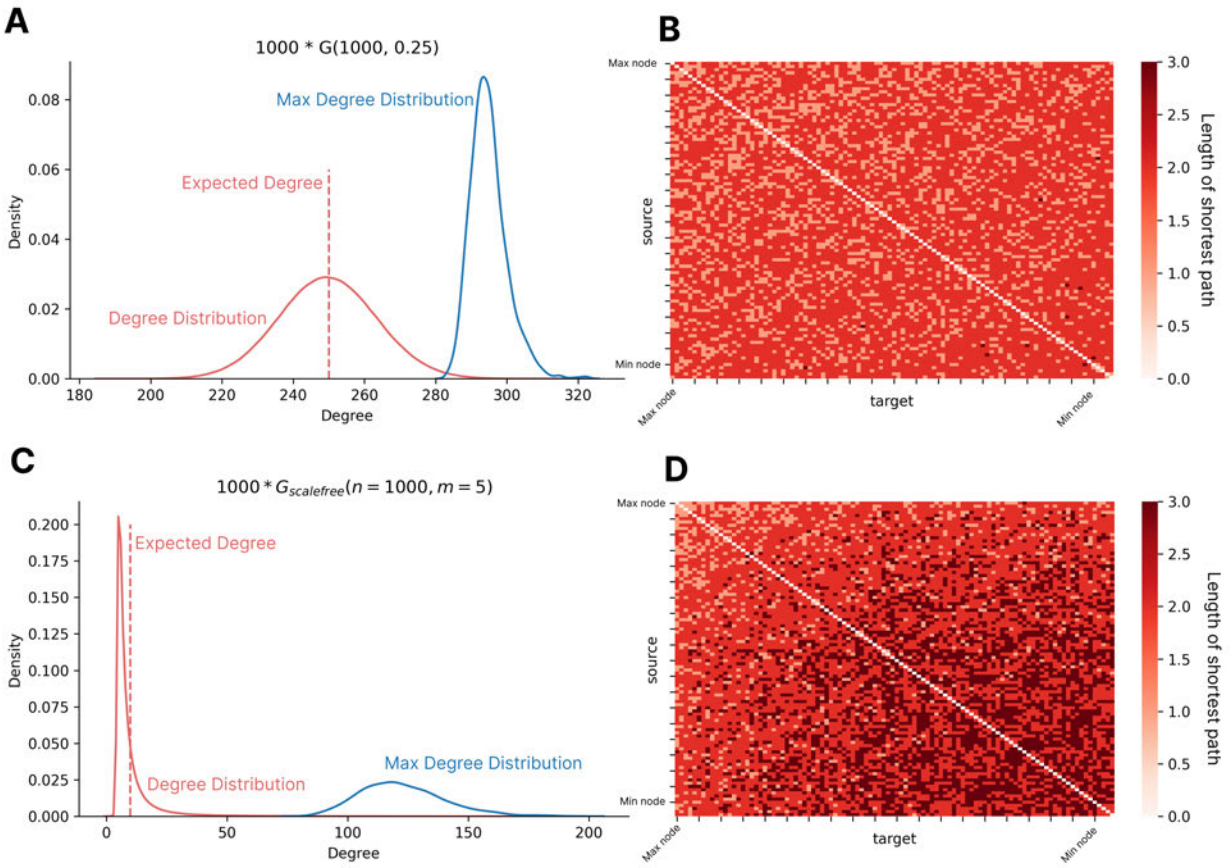

Figure 1: (**A, C**) The red solid line denotes the degree distribution of 1000 different random networks $G_r(1000, 0.25)$ in (**A**) and 1000 different scale free networks $G_{sf}(1000, m = 5)$ in (**C**). The red dashed line denotes the expected degree of these networks. The blue solid line denotes the distribution of the maximum degrees in each of these networks. (**B, D**) The length of the shortest path between two nodes in $G_r(100, 0.25)$ (**B**) and a BA network $G_{sf}(100, 5)$ (**D**). Each heat map is sorted by the degree of the nodes on both axes.

path to the node of maximum degree, having a shorter shortest path between any node and $v_{d_{max}}$, on average, suggests that $v_{d_{max}}$ should appear more frequently along the walk.

Random walks and their bounds

In this section, we describe the process of computing lower bounds for the number of steps needed before finding $v_{d_{max}}$ with high probability. We note that our analysis is constrained to undirected graphs in which the movement of from a given node to any of its neighbors has equal probability. Specifically, we describe the efficacy of the following pseudocode algorithm:

```
current_node = random(G.nodes)
current_max = current_node.degree
max_evolution = [current_max]

for i in range(number_of_steps):
```

```
    for j in range(mixing_time): #burn in time
        next_node = random(current_node.neighbours)
        current_node = next_node

    next_node = random(current_node.neighbours)

    if next_node.degree > current_max:
        current_max = next_node.degree

    current_node = next_node
    max_evolution.append(current_max)
return max_evolution
```

Random networks

At the start of a random walk, it is important to note that we are not sampling from the stationary distribution of the network, as any random walk requires a minimum number of steps r_d before the induced distribution observed via the random walk is within an error ϵ of the stationary distribution. This number of steps r_d is termed the "mixing time" of the network. An alternative, and fundamentally easier, question would be to estimate would the number of tries t needed of sampling the stationary distribution, before finding $v_{d_{max}}$ with high probability. The probability that we don't find $v_{d_{max}}$ after t tries can be expressed as follows. Let event M be the event that we don't find $v_{d_{max}}$ after t tries:

$$\mathbb{P}[M] = \left(1 - \frac{d_{max}}{D}\right)^t \leq \delta \qquad\qquad \hat{t} \geq O\left(\frac{log(1/\delta)D'}{d'_{max}}\right)$$

Here, we can see that our estimator $\hat{t}$ depends linearly on $log(1/\delta)$ and on the ratio of two random variables $\frac{D'}{d'_{max}}$. As such, if we can lower bound D' and upper bound d'_{max}, then we can obtain a lower bound for $\hat{t}$. Beginning with an upper bound of d'_{max}:

$$\mathbb{P}[d_{max} \geq C] = \quad 1 - \mathbb{P}[d_{max} < C] = \delta_2$$

$$\mathbb{P}\left[d_{\max} \geq (1+\epsilon)\,\mu\right] = \sum_{i=1}^{n} \mathbb{P}\left[d_i \geq (1+\epsilon)\,\mu\right] = n \cdot e^{\frac{-\epsilon^2\mu}{2+\epsilon}} \leq n \cdot e^{\frac{-\epsilon^2\mu}{3}} \leq \delta_2$$

$$\epsilon \geq \sqrt{3 \cdot \frac{\log\left(\frac{n}{\delta_2 p}\right)}{n}}$$

$$C = (1+\epsilon)\,\mu = O\left(np + p\sqrt{n\log\left(\frac{n}{\delta_2 p}\right)}\right) = O(np)$$

Here, we use a Chernoff bound on the probability that the degree of a given node exceeds C, since the degree of a given node is the sum of independent $\{0, 1\}$ indicator functions, and use a union bound over all nodes in the network to bound d_{max}. We take C to be our estimator for the upper bound of d_{max}. We would like δ_2 to be small, or in other words, the probability that $d_{max} \geq C$ to be small. Next, turning our attention to bounding D', we find that:

$$\mathbb{P}\left[|D - \mu| \geq \frac{\sigma}{\sqrt{\delta_3}}\right] \leq \delta_3 \qquad \mu = \mathbb{E}[D] = \mathbb{E}\left[\sum_{i=1}^{n} d_i\right] = n^2 p$$

Since d_i, d_j are pairwise independent for all $i, j \in n$:

$$\text{Var}[D] = \text{Var}\left[\sum_{i=1}^{n} d_i\right] = \sum_{i=1}^{n} \text{Var}[d_i]$$

$$\text{Var}[d_i] = \mathbb{E}[d_i^2] - \mathbb{E}[d_i]^2 = np - (np)^2 \leq np$$

$$\text{Var}[D] \leq n^2 p$$

$$\mathbb{P}\left[|D - \mu| \geq \frac{\sigma}{\sqrt{\delta_3}}\right] = 1 - \mathbb{P}\left[\mu - \frac{\sigma}{\sqrt{\delta_3}} \leq D \leq \mu + \frac{\sigma}{\sqrt{\delta_3}}\right] \leq \delta_3$$

$$D' = \mu - \frac{\sigma}{\sqrt{\delta_3}} = n^2 p - \frac{n\sqrt{p}}{\sqrt{\delta_3}} = O(n^2 p)$$

Having bounded both random variables, we can see that by plugging into the bound of $\hat{t}$:

$$\hat{t} \geq O\left(\frac{log(1/\delta)D'}{d'_{max}}\right) = O\left(\frac{log(1/\delta)n^2 p}{np}\right) = O\left(log(1/\delta)n\right)$$

In other words, even when sampling from the stationary distribution directly, we still need $O(log(1/\delta)n)$ tries before finding $v_{d_{max}}$. As such, in the case of random networks, it is more efficient to compute a linear scan of the nodes in the network rather than use a random walk.

Scale-free networks

Taking the same approach of sampling from the stationary distribution, we can find bounds on the number of steps t to find $v_{d_{max}}$ in a BA network. Firstly, according to Barabási and Bonabeau (2003) the probability of a node having a degree k can be expressed as:

$$p_k \approx k^{-\gamma}$$

where, for a BA network, $\gamma = 3$ Barabási and Bonabeau (2003). Here, γ is termed the degree exponent, a quantity characterizing the network topology. This quantity is independent of the size

of the initial clique m. Therefore, we can estimate an upper bound for d'_{max} in a scale free network as follows:

$$\mathbb{P}[d_{max} \leq C] = n * \mathbb{P}[d_i \leq C] = n * \int_0^C k^{-\gamma} dk = \frac{nC^{1-\gamma}}{\gamma - 1}$$

$$C = \left(\frac{\delta(\gamma-1)}{n}\right)^{\frac{1}{1-\gamma}}$$

$$C = O\left(\sqrt[\gamma-1]{\frac{n}{\delta}}\right) = O\left(\sqrt[\gamma-1]{n}\right) = O(\sqrt{n}) \quad \text{for } \gamma = 3$$

In a BA network, the sum of degrees D is deterministic, since at each time interval when a new node is introduced to the network, the number of added edges is a constant m. Therefore, the sum of degrees D can be expressed as follows:

$$D = 2|E| = 2\left[\binom{m}{2} + m(n-m)\right] = O(n) \quad \text{for } n \gg m$$

This is because, at time interval 0, there is an m sized clique and therefore $\binom{m}{2}$ edges, and each of the $n - m$ new nodes adds m new edges each. Therefore, we can see that, when sampling from the stationary distribution, the number of tries t needed before seeing $v_{d_{max}}$ can be expressed as follows:

$$\hat{t} \geq O\left(\frac{log(1/\delta)D}{d'_{max}}\right) = O\left(\frac{log(1/\delta)n}{\sqrt{n}}\right) = O(log(1/\delta)\sqrt{n})$$

This suggests that there may be some value to conducting a random walk on a BA network, since it would require less steps than a random network. While computing the exact mixing time of a given network can be done experimentally, we rely on previous work which suggest that the mixing time of "social networks" which have scale-free properties is on the order of $O(log^2(n))$ Mohaisen et al. (2010). In other words, after taking $O(log^2(n))$ steps, the node at which the random walk is located is no longer dependent on the initial node. Therefore, we would need a burn-in period of $O(log^2(n))$ steps before beginning our random walk. Therefore, the estimated number of steps $\hat{t}$ can be expressed as:

$$\hat{t} \geq O(log(1/\delta)\sqrt{n} * log^2(n))$$

We can see that, for $\delta \leq 0.1$, this value is greater than n for $n > 10$. Therefore, when accounting for mixing time, a simple linear scan through the network's nodes continues to be a more efficient solution for finding the node of maximum degree. Nonetheless, if the nodes in the network are opaque to the user, this algorithm may still be useful.

Simulation and experimental results

We begin by experimentally testing the bounds computed in the previous section on both random and BA networks. Figure 2 illustrates the average performance of 1000 iterations of the algorithm

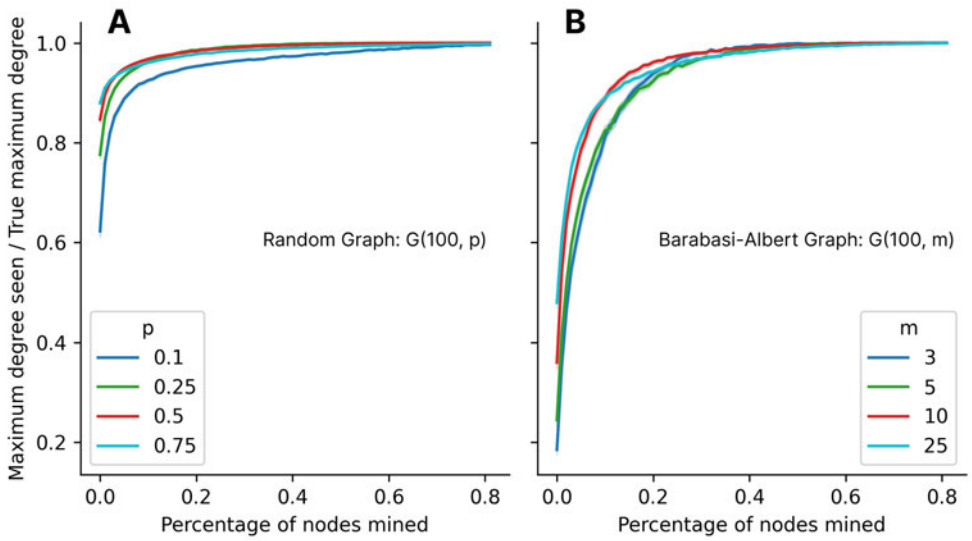

Figure 2: Average performance of 1000 random walks in locating $v_{d_{max}}$ vs. number of unique nodes visited on a random walk through a random network (**A**) and BA network (**B**). Each network is tested at different values of p and m, respectively.

described in the previous section as a function of the number of "mined" nodes, or the number of unique nodes visited, on a random walk through a given network. The performance is estimated as the ratio of the maximum degree seen thus far on the walk and the true maximum degree of the network. We illustrate the performance on a random network $G_r(n = 100, p)$ for $p = [0.1, 0.25, 0.5, 0.75]$ in Figure 2A and the performance on a BA network $G_{sf}(n = 100, m)$ for $m = [3, 5, 10, 25]$ in Figure 2B. As can be seen in Figure 2A, as the likelihood of two nodes being connected by an edge increases, the node at which we start a random walk is closer to the maximum degree node. However, as the random walk progresses, the algorithm is able to find a node with a degree at least within 95% of the true maximum degree while uniquely visiting less than half of all nodes in the network. As for BA networks, regardless of the choice of m, all random walks arrive at the true maximum degree network after visiting 60% of the nodes in the network.

Next, we test the algorithm on three external datasets compiled from the Stanford SNAP dataset Leskovec and Krevl (2014) (i) a page-page network of Facebook pages (22,470 nodes, 171,002 edges), where the nodes are pages and while the links are mutual likes between sites; (ii) User-User network of Twitch users, where the nodes are users and links are mutual friendships between them (168,114 nodes, 6,797,557 edges); (iii) Patent citation dataset where nodes are patents and links are citations made by one patent to another (3774768 nodes, 16518948 edges). This is originally a directed graph, but we convert it to an undirected graph.

The results of this analysis are summarized in Figure 3. Subplots A, C, and E describe the average performance and 95% confidence intervals of 1000 random walks in locating the node of maximum degree in the Facebook, Twitch, and patent citation datasets respectively. On the other hand, subplots B, D, and F describe the degree distribution of these datasets on a log-log scale. As can be seen in the figures on the left-hand side, the algorithm is able to locate the node of maximum degree while only uniquely visiting less than 10% of the network's nodes in the case of the Facebook and Twitch datasets, and less than 2.5% of the nodes in the case of the patents dataset. Each of these networks exhibit scale-free properties, as can be seen by their power-law degree distributions

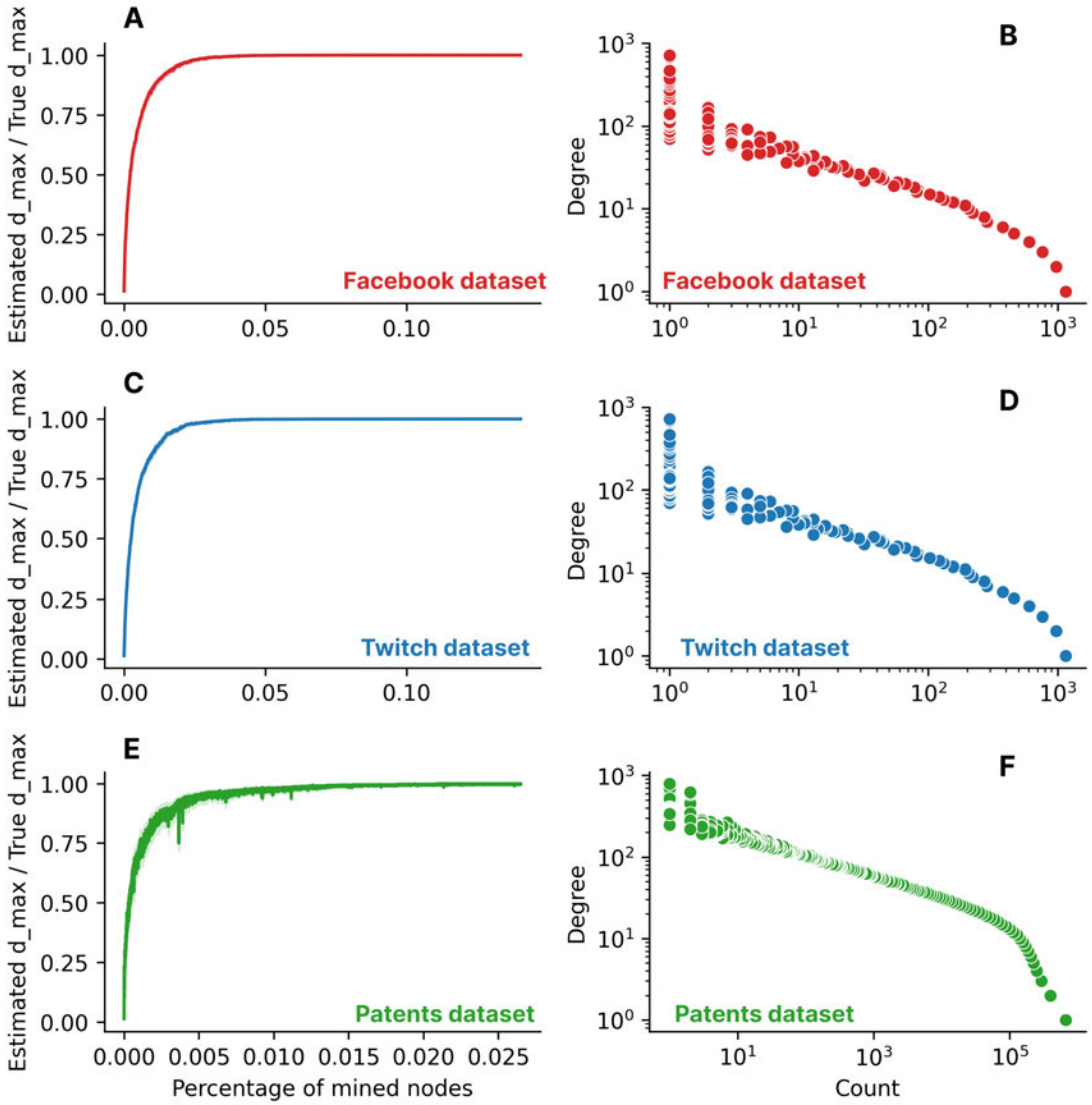

Figure 3: (**A, C, E**) Average performance of 1000 random walks in locating the node of maximum degree in the Facebook, Twitch, and patent citation datasets respectively. (**B, D, F**) Degree distribution of these networks on a log-log scale.

shown in the right-hand side figures.

We've also built an interactive visualization tool using D3, a JavaScript library, allowing users to play around with random walks of varying lengths and to see the performance of these walks on random, scale-free, and lattice networks. This tool can be found here.

Discussion and conclusions

In this project, we aimed to estimate lower bounds for the number of steps needed on a random walk before returning the node with maximum degree in the network with high probability. As demonstrated, the lower bounds computed with this approach exceed the number of nodes n in most cases. Hence, if a user were to have access to a list of the nodes in the network, and their properties, it would be more efficient to complete a linear scan on the set of nodes to find the node with maximum degree. However, if this information is opaque to the user, the algorithm described in this project may be useful.

Furthermore, this algorithm may be similarly useful if one wished to find some node v whose degree is relatively close to that of $v_{d_{max}}$. As shown in Figure 2, in both random and scale free networks, the estimated maximum degree node exhibits a sharp rise at the start of the random walk, finding nodes with a relatively high degree in a small number of steps.

Importantly, the results demonstrated in Figures 2 and 3 illustrate the performance against that of the number of unique nodes visited, rather than the length of the random walk. Therefore, the performance here is plotted against a proxy of memory required to store the random walk's properties, rather than the time of walk. Future work may examine theoretical bounds on the number of unique nodes visited on a random walk before finding $v_{d_{max}}$.

Acknowledgements

We would like to thank Professor Musco and the teaching assistants of the course, Apoorv, Raphael, Feyza and Teal, for all their support throughout this semester. Despite the difficulty of the material, you have kept the class engaging, light, and incredibly interesting. We hope to cross paths with you again in the future.

References

Barabási, A.-L. and Bonabeau, E. (2003). Scale-free networks. *Scientific american*, 288(5):60–69.

Barnes, J. A. and Harary, F. (1983). Graph theory in network analysis. *Social networks*, 5(2):235–244.

Bornholdt, S. and Schuster, H. G. (2001). Handbook of graphs and networks. *From Genome to the Internet, Willey-VCH (2003 Weinheim)*.

Chakraborty, T., Dalmia, A., Mukherjee, A., and Ganguly, N. (2017). Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, 50(4):1–37.

Chen, Q. and Chen, S. (2007). A highly clustered scale-free network evolved by random walking. *Physica A: Statistical Mechanics and its Applications*, 383(2):773–781.

da Fontoura Costa, L. and Travieso, G. (2007). Exploring complex networks through random walks. *Physical Review E*, 75(1):016102.

De-Marcos, L., García-López, E., García-Cabot, A., Medina-Merodio, J.-A., Domínguez, A., Martínez-Herráiz, J.-J., and Diez-Folledo, T. (2016). Social network analysis of a gamified e-learning course: Small-world phenomenon and network metrics as predictors of academic performance. *Computers in Human Behavior*, 60:312–321.

Erdős, P., Rényi, A., et al. (1959). On random graphs i. *Publ. math. debrecen*, 6(290-297):18.

Erdős, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60.

Farooq, A., Joyia, G. J., Uzair, M., and Akram, U. (2018). Detection of influential nodes using social networks analysis based on network metrics. In *2018 international conference on computing, mathematics and engineering technologies (icomet)*, pages 1–6. IEEE.

Hardiman, S. J. and Katzir, L. (2013). Estimating clustering coefficients and size of social networks via random walk. In *Proceedings of the 22nd international conference on World Wide Web*, pages 539–550.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Lovász, L. (1993). Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4.

Mohaisen, A., Yun, A., and Kim, Y. (2010). Measuring the mixing time of social graphs. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 383–389.

Saramäki, J., Kivelä, M., Onnela, J.-P., Kaski, K., and Kertesz, J. (2007). Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105.

Schank, T. and Wagner, D. (2005). Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9(2):265–275.

Wellman, B. (1983). Network analysis: Some basic principles. *Sociological theory*, pages 155–200.

Xia, F., Liu, J., Nie, H., Fu, Y., Wan, L., and Kong, X. (2019). Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(2):95–107.