Measuring the Sampling Robustness of Complex Networks

Katchaguy Areekijseree
Syracuse University
kareekij@syr.edu

Sucheta Soundarajan
Syracuse University
susounda@syr.edu

***Abstract*—When studying a network, it is often of interest to understand the robustness of that network to noise. Network robustness has been studied in a variety of contexts, examining network properties such as the number of connected components and the lengths of shortest paths. In this work, we present a new network robustness measure, which we refer to as 'sampling robustness'. The goal of the sampling robustness measure is to quantify the extent to which a network sample collected from a graph with errors is a good representation of a network sample collected from that same graph, but without errors. These errors may be introduced by humans or by the system (e.g., mistakes from the respondents or a bug in an API program), and may affect the performance of a data collection algorithm and the quality of the obtained sample. Thus, when data analysts analyze the sampled network, they may wish to know whether such errors will affect future analysis results.**

We demonstrate that sampling robustness is dependent on a few, easily-computed properties of the network: the leading eigenvalue, average node degree and clustering coefficient. In addition, we introduce regression models for estimating sampling robustness given an obtained sample. As a result, our models can estimate the sampling robustness with MSE $<$ 0.0015 and the model has an R-squared of up to 75%.

1. Introduction

Within the field of data mining, studying the structure of complex networks has become an interesting and important task. Data analysts can gain several insights by analyzing real-world networks. For example, one can study how fast information flows through a network, how people form a community, or identify products to recommend to consumers. However, before performing any graph analysis task, one must collect appropriate network data.

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08 $15.00
http://dx.doi.org/10.1145/3341161.3342873

Depending on the domain, there may be many ways to collect network data. For example, one can collect social data through pen-and-paper questionnaires or by interviewing subjects, or for online networks, one can query platforms through a provided API. In some applications, a data analyst or a data collector may have no initial knowledge about a network except the identity of a single seed node (e.g. the first person that she will interview). A network sample can be expanded by querying already-observed nodes to learn their neighbors. In this work, we assume that all edges incident to the queried node are observed and added to the network sample. We refer to this process as *network sampling* or *sampling through crawling*. Over the past few decades, many crawling algorithms have been introduced and used, such as breadth-first search, depth-first search, many variants of random walk, and so on.

In real world application scenarios, errors may occur during this data collection process. When a data collector performs a query, a list of neighboring nodes is returned in response. However, this list may be incomplete. Such errors can occur for many reasons: for example, a participant who answers a questionnaire may make a mistake on their answer, a web crawler may have a bug and fail to extract links from web pages, or an adversary may tamper with the API and alter the information exchanged between two parties. These errors may then lead to errors in a subsequent network analysis. Therefore, it is important for a data analyst to know if a collected sample is trustworthy.

In this work, we introduce a new network robustness measure, which we call *sampling robustness*. To the best of our knowledge, while there are many ways to evaluate network robustness in general, there is no existing work that measures a network's robustness with respect to sampling. For a crawler of choice C, the sampling robustness of network G, denoted by $R_p(G, C)$, is defined as the expected similarity between two samples: one produced by crawler C on an error-free version of G, and one produced by C on a version of G in which each edge is missing with probability p. Intuitively, if a network is robust to this error, the performance of a crawler C will be mostly unaffected by missing edges when it crawls network G.

In this work, we model error as random edge deletion, though our definition could easily be adapted to other types of error. Our goal is to investigate how the sampling robustness of a network changes due to random edge deletion, and

analyze network sampling robustness with respect to the network's structural properties, allowing a data analyst to predict whether a network will have high or low sampling robustness by measuring only a small set of parameters. We observe that sampling robustness is correlated with the network's leading eigenvalue, average node degree, and global clustering coefficient. Thus, with these simple measurements, one can estimate a network's robustness.

Our contributions can be summarized as follows:

1) We introduce and define a new robustness measure, *sampling robustness*, which measures the robustness of a network with respect to sampling by a crawler when edges from the original network are dropped at random. Each edge has probability p that it will be removed from a list of returned edges after each query.
2) We show that the sampling robustness highly depends on the network structure. Networks of different types have different level of sampling robustness.
3) We observe that sampling robustness is highly correlated with the leading eigenvalue, average degree and average clustering coefficient calculated from sampled networks.
4) We present regression models for estimating sampling robustness given a sampled network.

2. Related Work

In this section, we explore previous literature that relates to network sampling and network robustness.

Network Crawling: In the past decades, there have been many network crawling techniques proposed for various fields of study. Many researchers have used basic graph traversal techniques like breadth-first search (BFS) or depth-first search (DFS) for web crawling [1], [2]. For example, Mislove, et al. use a BFS crawler to collect samples of many online social networks (Orkut, Youtube, Live Journal, and Flickr) in [1]. BFS crawlers have also been used for collecting hyperlinks on the WWW in [3]. Snowball sampling, a variation in BFS in which only a fraction of each node's neighbors are added to the queue, has been used for finding hidden populations, a necessary task in some social science research [4], [5]. Such methods are appealing in part because of their simplicity. Random walk crawlers are another popular method. Random walks have been used for crawling peer-to-peer networks [6], [7], the WWW [8], [9] and online social networks [10], [11]. Variations on random walks are useful for collecting a uniform sample [12], [13].

The literature also contains analyses of network crawling methods. In [14], Leskovec and Faloutsos study the characteristics of different sampling methods. Their objective is to determine which method generates samples with the least bias. Similar studies are presented in [15], [16]. An analysis of BFS crawler is presented [17]. The results shows that BFS crawler is biased towards nodes with high degree. Areekijseree et al. present a series of extensive experiments in [18] which focus of the crawlers' performance. Several crawlers are evaluated on synthetic and real networks. They observe that the performance of a crawler is highly dependent on network properties.

Saroop and Karnik study the performance of different crawlers on the task of node coverage, focusing on the crawler that is the best to crawl Twitter network [19]. Other properties like *in-* and *out-* degree distributions are also taken into account when network is crawled. Similarly, Baeza-Yates, et al. focus on the crawlers for collecting web pages [20]. Several crawlers are studied based on how well they find important pages. Results show that the BFS crawler has poor performance, as compared to other crawlers like PageRank and OPIC. Hu and Lau present an excellent survey of several popular sampling methods for both down-sampling and crawling [21]. Several important network properties, theoretical studies and different types of evaluation criterion are discussed in detail.

Network Robustness: To the best of our knowledge, there is no existing work on network robustness on network crawling or data collection. However, there are numerous studies on other forms of network robustness. Work on network robustness has a long history and it has been heavily studied by researchers from different backgrounds, including computer science, biology, physics and mathematics. In general, network robustness is defined as the ability of a network to keep functioning when there is a random failure or targeted attack [22]. For example, a telecommunications network is considered to have high robustness if the network continues its functions and services when some devices fail. Intuitively, robustness is all about back-up possibilities or alternatives paths [23]. Interest in network robustness was sparked by the study of Albert el al. [22]. They study the effect of random failures and targeted attack. They measure network robustness by the diameter of the network and size of the largest connected component. The results show that scale-free networks have a high degree of tolerance against random failure, as opposed to random networks. However, scale-free networks are very sensitive to targeted attack. The diameter of the network drastically increases and the network breaks into several components when the hub nodes are attacked. Cohen et al. are interested in finding the critical point (exact fraction of nodes to be removed, which causes the networks to break into isolated fragments) under a targeted attack in [24]. Other measures for capturing network robustness are proposed, including shortest-path [25], path diversity [26], eigengap [27], spectral radius [28].

3. Sampling Robustness

In this section, we begin by giving a brief description of the data collection process and random error. Next, we provide the details of different crawling techniques. Lastly, we define sampling robustness of a network G.

3.1. Network Data Collection

Let $G = (V, E)$ be an undirected, unobserved network, where V is the set of nodes and E is the set of edges. At the start of the data collection process, only the identity of a single seed node is available. A data collector collects a network sample by adopting a crawling technique C. Given

a seed node and a query budget, the crawler expands the sample by iterative querying observed nodes. For each query response, all the neighboring nodes and incident edges are returned and added to the sampled network. The crawler selects the next queried node from the list of observed nodes in the obtained sample and repeats until the query budget is exhausted. We assume that the data collector queries each observed node at most once. A crawler C thus generates a sampled network $S = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$ are a list of nodes and edges observed, respectively.

3.2. Random Error

Error can originate from many different sources, such as mistakes or missing answers from survey respondents, a misreading of instruments by the data collector, or a bug in the data collection program. In this work, we consider the case of error caused by *random edge deletion*. With this type of error, each query misses some fraction of edges. To model this type of error, each returned edge has a probability p that it will be removed from a list of returned edges after each query. If there is an edge between node A and B, this edge may be missed when a crawler queries on node A, but this edge may be discovered when B is queried.

3.3. Network Crawling Technique

In this section, we describe each crawling method in details. We select three popular crawling algorithms: BFS, random walk and MOD. These crawlers were selected as they represent three important categories of crawling algorithms [18]. To collect a network sample, each crawler is given the same seed node and the same total query budget (in our experiments, we set a budget to be 10% of the total nodes of a network).

Breadth-first search (BFS): The BFS crawler selects the node that has been in the list of unqueried nodes the longest (First-in, First-out). After each query, all of the neighboring nodes that have not been queried are added to the queue. BFS crawler uniformly expands its frontier and is good at capturing a complete view of the networks.

Random walk (RW): In each iteration, the crawler transitions to a neighbor of the node that was just queried at random. The crawler performs a query if it lands on an unqueried node. The random walk crawler is capable of finding many nodes from different regions (e.g. communities). Here, the random walk crawler cannot teleport.

Maximum observed degree (MOD): This crawler selects the unqueried node with the highest degree. The MOD crawler finds hub nodes in a few iterations [29].

3.4. Measuring Sampling Robustness

We define a novel network measure, *sampling robustness*, which measures the extent to which a sample generated by a crawling algorithm in the presence of errors (either in the original network- e.g., a communications network in which edges flicker in and out of existence- or in the crawling process itself- e.g., errors in the crawling process) is representative of a sample generated by the same algorithm without errors. If a network G has high sampling robustness, the performance of a crawler C on network G will be consistent regardless of whether there are errors in the original network or in the data collection process. Here, we assume such errors take the form of edges missing uniformly at random, but the definitions and analysis that we present can easily be generalized to other types of errors.

Definition 3.1. Sampling Robustness

$$R_p(G, C) = \frac{sim(M(S), M(S^{'}))}{\bar{R}_0}$$

We denote the sampling robustness of G when p fraction of edges are missing uniformly at random as $R_p(G, C)$, which is shown in Definition 3.1 In this paper, we let S represents the *complete* sample (i.e., the sample produced by running C on G), and let $S^{'}$ represents the sample obtained by the crawler C with errors (i.e., the sample produced by running C on G with missing edges). The numerator is defined by computing the similarity between two samples, S and S', produced by a crawler C: the first on the original network G without errors, and the second on a version of G in which p fraction of edges have been removed at random.

In the denominator, we account for potential randomness in sampling (including the choice of seed node from which the crawler begins). We normalize this value by $\bar{R}_0$, which represents the average similarity between two samples in the case where there are no missing edges (p = 0). To calculate this, we generate multiple *error-free* samples and compute the average similarity of these samples against each others.

Performance measure: In Definition 3.1, $M(S)$ is an application-specific function which characterizes the performance of the crawler C when it generates a sample (e.g., if one is interested in the sampling robustness of a network for the community detection application, $M(S)$ could represent the set of communities detected on S). Note that $M(\cdot)$ can be any function, as depends on the sampling goal. This means that different types of outputs can be returned by $M(\cdot)$. Some examples are as follows:

- Numbers - e.g. the number of nodes or edges found
- A set - e.g. the distinct nodes in the sample.
- A set of sets - e.g. communities in the sample.
- A distribution - e.g. degree distribution of the sample.

Thus, the appropriate *similarity* measure depends on the output of $M(\cdot)$. Some examples are as follows:

$$sim(\cdot, \cdot) = \begin{cases} 1 - d_{canberra/L_1/L_2}, & numbers \\ Jaccard\ similarity, & a\ set \\ Normalized\ Mutual\ Info. & a\ set\ of\ sets \\ 1 - KS\ statistic, & distribution \end{cases}$$

Our code and implementation can be found at https://github.com/kareekij/sampling-robustness.

4. Sampling Robustness and Network Type

TABLE 1: Statistics of network. $|V|$ is a number of nodes, $|E|$ is a number of edges, $\bar{d}$ is an average degree, $\bar{cc}$ is a average clustering coefficient and λ_1 is a leading eigenvalue of adjacency matrix A of the network. These networks can be downloaded from www.networkrepository.com

Type	Network	$\|V\|$	$\|E\|$	$\bar{d}$	$\bar{cc}$	λ_1
CA	Erdos992	4991	7428	2.977	0.08352	15.13
	HepTh	8638	24806	5.743	0.4816	31.03
	GrQc	4158	13422	6.456	0.5569	45.62
BIO	CE-GN	2215	53680	48.47	0.1843	96.22
	CE-PG	1692	47309	55.92	0.4467	152.6
	SC-GT	1708	33982	39.79	0.3491	109.9
SOCFB	Amherst41	2235	90954	81.39	0.3104	137.1
	Colgate88	3482	155043	89.05	0.2673	141.9
	Bowdoin47	2250	84386	75.01	0.289	124.2
SOC	Hamsterster	2000	16097	16.1	0.54	50.02
	Advogato	5054	39374	15.58	0.2526	70.51
	Wiki-Elec	7066	100727	28.51	0.1418	138.1
Tech	PGP	10680	24316	4.554	0.2659	42.44
	Router-RF	2113	6632	6.277	0.2464	27.67
	WhoIS	7476	56943	15.23	0.4889	150.9

Our definition of sampling robustness requires access to the original network G. Note that this is something of a contradiction: if one has the entire original network G, one need not concern oneself with sampling, or even with robustness! So in practice, if one has collected a sample, with errors, from a graph G, how can one determine whether that sample is likely to be a good representation of the sample one would have obtained had the sampling process not contained errors?

Naturally, the sampling robustness of a network must depend on that network's properties. As shown in [18], networks of the same type tends to have similar properties. For example, an average degree of collaboration network (e.g. number of names appears on the manuscript) is around 5 while the average degree of the social network (number of friends) is around 20. By considering network type, we can roughly classify networks by their properties. We computed the sampling robustness of 15 networks on 5 categories. In this paper, we use $M(\cdot)$ as a size of the sample or node coverage, which defined as

$$M(S) = |\{v \in V_s^{'}, V_s^{'} \subseteq V\}|$$

This function measures the performance of a crawler in terms of the number of nodes discovered after a crawler performs some specified number of queries. To compare the similarity of two samples S and $S^{'}$ (each produced with the same number of queries), we use the Canberra Distance, since the output is normalized between 0 and 1. So, the similarity between S and $S^{'}$ is defined as

$$sim(M(S), M(S^{'}) = 1 - d_{canberra}(|V_s^{'}|, |V_{s'}^{'}|)$$

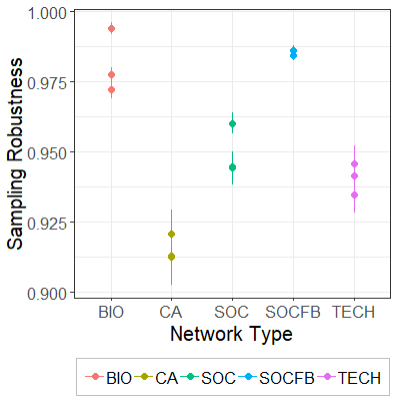


Figure 1: Aggregate results of $R_p(G, BFS)$ when p is ranging between 0.1-0.5. Each point represent a network from various categories (along x-axis). Sampling robustness highly depends on network type.

Statistics of each network are listed in Table 1. Results are illustrated in Figure 1. Each point represents the sampling robustness of a network, as computed over 10 trials. p is varied between 0.1 and 0.5. We use a BFS crawler to collect samples. Similar results were obtained for other tested crawlers (random walk and MOD, specifically).

As we can clearly see in Figure 1, networks of different types tend to have different levels of sampling robustness and networks in the same category have similar sampling robustness. Biological and Facebook networks tend to be the most robust, as opposed to collaboration networks, which have the lowest sampling robustness.

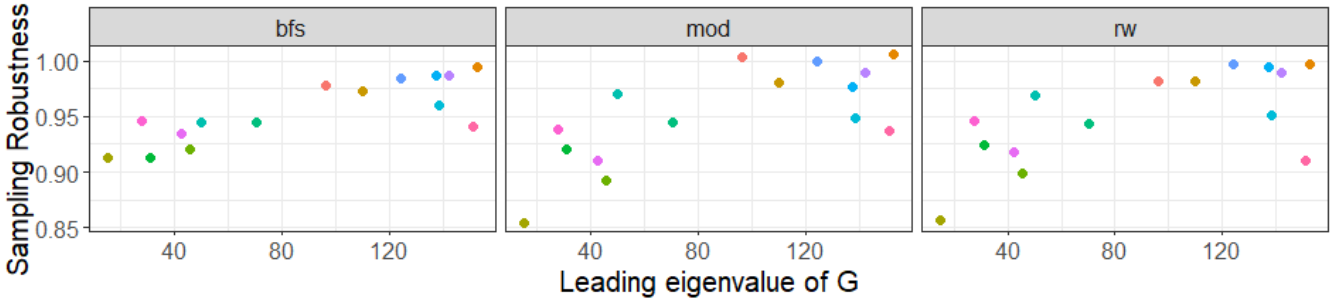
5. Characterizing Sampling Robustness

As noted in the previous sections, computing sampling robustness requires generating an error-free sample S. In real-world applications, obtaining this sample is not practical. In the previous section, we showed that one can roughly estimate sampling robustness from the network category. In this section, we will demonstrate that $R_p(G, C)$ highly depends on the structural properties of both the original network G as well as the obtained network samples $S^{'}$.

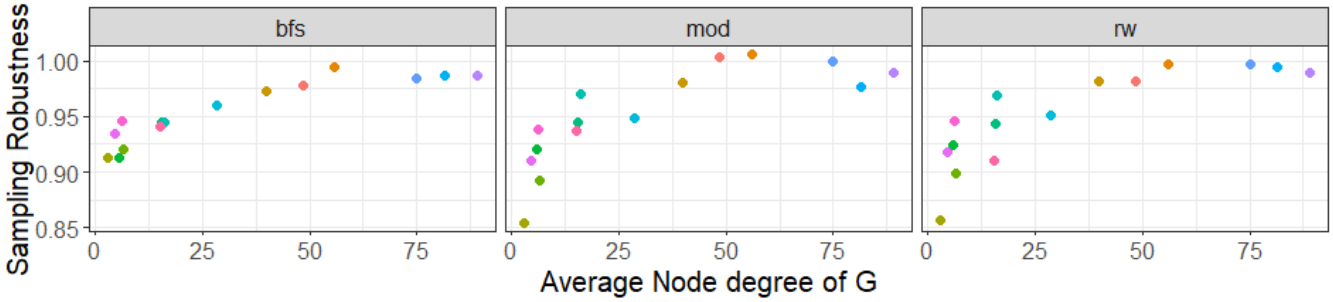
Our earlier work demonstrated that the structural properties of the network play the important role in network sampling [18]. Specifically, certain network properties enhance (or degrade) the efficiency of a crawler.

When sampling error occurs, certain edges may be invisible to the crawler. The ability of the crawler to expand the sampled network may thus drop, because the crawler makes its query decisions based on the nodes and edges in the sampled network that it has observed so far. What, then, are the properties that make a network robust to sampling?

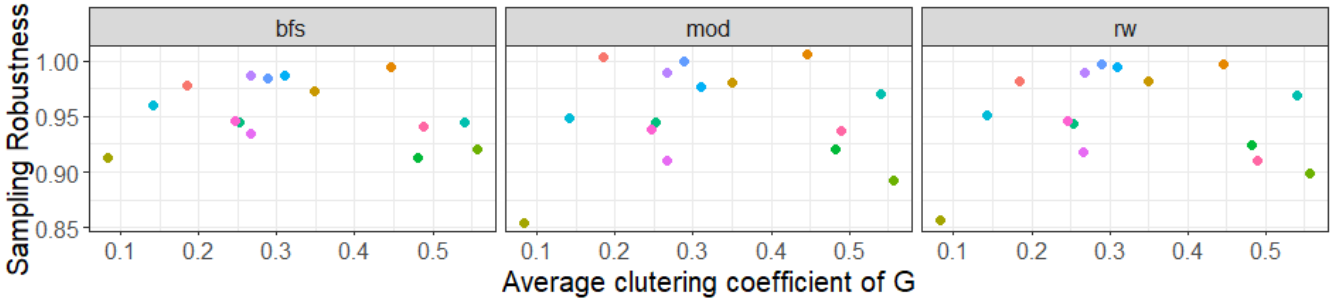
Here, we investigate three properties that we believe support a crawler in expanding a sample's boundary; leading eigenvalue λ_1, average node degree $\bar{d}$, and average clustering coefficient. Leading eigenvalue and average node degree are closely related, since λ_1 is bounded by the degree



(a) Aggregate results of $R_{p:0.1\to0.5}$ against leading eigenvalue λ_1



(b) Aggregate results of $R_{p:0.1\to0.5}$ against leading average degree.



(c) Aggregate results of $R_{p:0.1\to0.5}$ against average clustering coefficient.

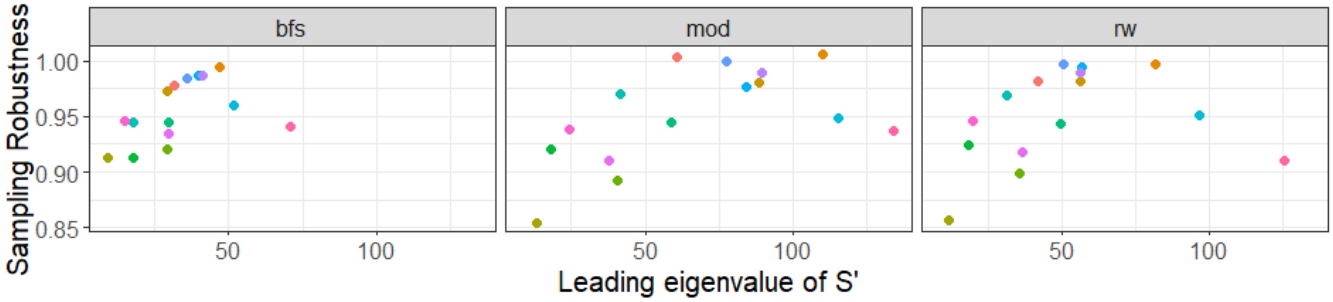
Figure 2: Each point represents a network. Sampling robustness highly depends on λ_1 and $\bar{d}$, but not $\bar{cc}$ of a network.

of a network [30]. Intuitively, if the average degree of a network is k, a naive crawler should discover approximately k nodes on average for each query. Thus, the higher average degree, the higher amount of nodes a crawler discovers which makes network more robust. Similarly, a network with high average clustering coefficient indicates densely connections between nodes. Thus, a network with high average clustering coefficient will help a crawler to find many nodes in a few iterations which makes network robust to missing edges. We observe that sampling robustness is highly dependent on these network properties, as computed in both the original network as well as the sample generated with errors. Figure 2 and 3 illustrate the correlation between sampling robustness and each property on original network G and obtained sample network $S^{'}$, respectively.

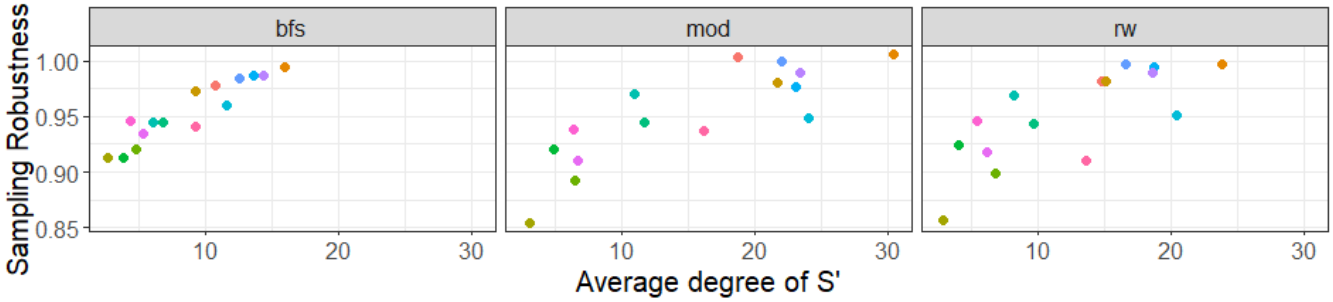
The *largest* eigenvalue of the adjacency matrix A, denoted by λ_1, plays an important role in forecasting epidemic spreading processes. As shown in [30], λ_1 is related to the epidemic threshold τ, which governs how quickly disease can spread through a network via the SIR model. It has been shown that $\tau = \frac{1}{\lambda_1}$, so, one can predict whether an epidemic will die out on any given network by considering only a single parameter. In the SIR model, where β is the birth rate and γ is the curing rate, the epidemic will die out iff $\frac{\beta}{\gamma} < \tau$.

The epidemic process and network data collection process have similar dynamics. Both starts from a single seed node and gradually expand outwards. In both cases, we look at the population of interest after t time steps: i.e., how many people get the diseases or how many distinct users a crawler discovers through crawling.

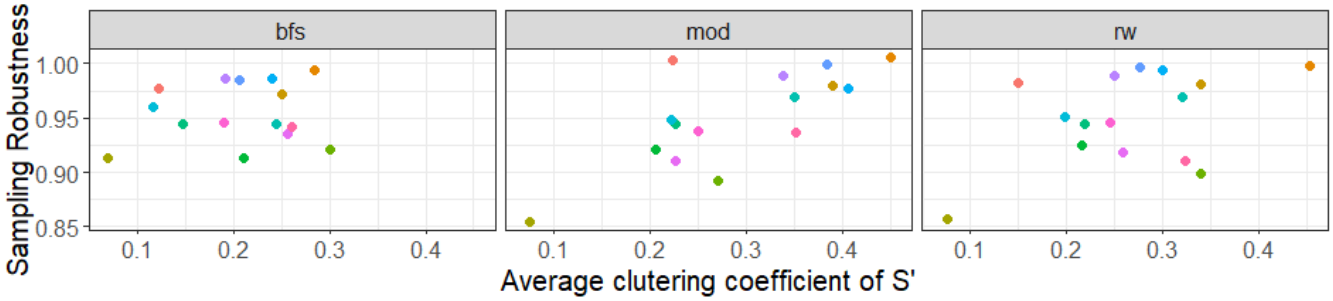
We can consider network crawling to be a simpler version of epidemic model, where γ is a constant and β is a *crawl rate* (e.g., number of requests per second, number of new nodes added to sample). So, we can use λ_1 to indicate how fast the crawler can expand the sample. Since λ_1 is bounded by the average degree, the larger λ_1 means the higher average



(a) Aggregate results of $R_{p:0.1\to0.5}$ against leading eigenvalue λ_1



(b) Aggregate results of $R_{p:0.1\to0.5}$ against leading average degree.



(c) Aggregate results of $R_{p:0.1\to0.5}$ against average clustering coefficient.

Figure 3: Each point represents a network. Sampling robustness highly depends on observed λ_1, $\bar{d}$ and $\bar{cc}$ of $S^{'}$.

degree. As we will see in Figure 2b, higher average degree indicates that a crawler can easily expand its sample.

In Figure 2a, we plot sampling robustness against leading eigenvalue λ_1 of the adjacency matrix A of the network G. The leading eigenvalues of the error-containing sample $S^{'}$ are shown in Figure 3a. In these figures, each point is the sampling robustness of a network, computed as an average over 10 experiments. Each sub-figure illustrates a case when different crawling technique is used. The relationship between sampling robustness and leading eigenvalue is highly correlated. As expected, we observe that a network with higher leading eigenvalues (low threshold τ) is more robust. This indicates that a crawler can easily expand its sample even the edges are missing.

Next, the average degree indicates the number of neighbors of each node, on average (e.g. average number of friends of people on social network). Intuitively, a crawler can more quickly expand its sample if the average degree is large, and can continue to do so even if some of the edges are lost.

Figure 2b illustrates the average sampling robustness against average degree of a network. The average observed degree of samples are shown in Figure 3b. The results are aggregated over 10 runs when p is varied from 0.1 to 0.5. Each sub-plot represents results from different types of crawler. As we expected, the sampling robustness is also highly correlated with the average degree of the networks as well as the average observed degree of the obtained samples.

Lastly, we consider the average clustering coefficient of the network and the samples. This property measures how well nodes are connected. A higher clustering coefficient indicates that neighboring nodes are densely connected to each others. Intuitively, when nodes are densely connected (near clique structure), the crawler will discover nodes quickly, and is more robust against missing edges. In Figure 2c, we observe that the clustering coefficient of a network G is not correlated with its sampling robustness. However, we

do observe that the clustering coefficient of a sample $S^{'}$ is correlated with the sampling robustness. This may be because a large portion of the nodes in network G have degree 1 due to the power-law distribution, and these nodes bring down the average clustering coefficient overall. On the other hand, our selected crawlers are known to be biased toward hub nodes [17], [31], so the sampled networks contain nodes with high degree connecting to each others. The sampled network represents the inner-core structure of the network rather than the periphery, which is a better indicator for measuring robustness.

6. Sampling Robustness Estimation

In this section, we introduce a regression model which we can use to estimate a sampling robustness of any network given an obtained sample. We describe how to estimate error probability p and show how we construct our models for estimating sampling robustness in subsection 6.1. Then, we evaluate the models and report the results in subsection 6.2.

6.1. Model Training

6.1.1. Regression Model. Here, we present a model for each of the BFS, random walk, and MOD crawlers. Given an obtained sample $S^{'}$ that is generated by a crawler C, we can estimate the robustness of the original graph G from the observed properties of the sampled network $S^{'}$ as

$$\hat{R}_p = c_1 p + c_2 \bar{d}^{'} + c_3 \lambda_1^{'} + c_4 \bar{cc}^{'} + c_5 (cc^{'} \times \bar{d}^{'}) + b$$

From the previous section, there is a high correlation between sampling robustness and the observed structural properties of a sampled network. Thus, we build and present a linear regression model for estimating the sampling robustness. We train our model from the sampled networks which we obtained in the previous experiment. In total, there are around 2,200 sampled networks used in the model training.

TABLE 2: Coefficients and intercept of each model

Model	c_1	c_2	c_3	c_4	c_5	b
M1-RW	-0.1843	0.0127	-0.0009	0.4374	-0.0245	0.8661
M2-BFS	-0.1951	0.0119	-0.0006	0.2165	-0.0250	0.9313
M3-MOD	-0.2199	0.0094	-0.0006	0.3928	-0.0152	0.8801

This model represent the relationship between the response variable ($\hat{R}_p$) and predictor variables ($\bar{d}^{'}$, $\bar{cc}^{'}$, $\lambda_1^{'}$, p). However, before the estimation, uses need to estimate the error probability p, which can be done as follows.

6.1.2. Estimating error probability. The probability p can be estimated by performing multiple queries on the same node and counting the number of times a particular edge is duplicated. Let k be the number of times a crawler queries node u, and e be one of the edges incident to node u in G. So, p can be estimated by $p = 1 - \frac{k_e}{k}$, where k_e is the number of times edge e is seen after k queries on u. Users can estimate p with a small k. In our analysis, we assume that these multiple queries are performed using a small amount of budget, and is done after obtaining the samples.

6.2. Model Evaluation

To test our model, we use samples generated from the networks listed in Table 3. These networks were not used in generating the regression model. We use these networks as the original networks G, and use the BFS, random walk, and MOD crawlers to generate samples. The error probability p ranges from 0.1 and 0.5. For each network, ten network samples are generated using each crawler, for each of p. In total, we have around 600 sampled networks.

TABLE 3: Statistics of network used for model testing.

Network	$\|V\|$	$\|E\|$	$\bar{d}$	$\bar{cc}$	λ_1
Hamilton46	2312	96393	83.38	0.2983	135.93
Trinity100	2613	111996	85.72	0.2903	135.83
Epinion	26588	100120	7.53	0.1351	66.206
Caida2007	26475	53381	4.03	0.2082	69.643

We evaluate our models through mean square error (MSE) and R-squared (R^2). MSE measures the quality of the estimate (lower is better) while R^2 measures how well the model fits the data (higher is better).

TABLE 4: Evaluation of the each regression model in term of Mean Square Error (MSE) and R-square (R^2).

	M1	M2	M3
MSE	0.00127	0.00089	0.00142
R^2	0.7258	0.7147	0.7440

The evaluation results of each model are shown in Table 4. Our proposed models are capable of estimating the sampling robustness of a network G from a sample $S^{'}$ with very small MSE (< 0.0015) and a R^2 of up to 0.75.

Through this method, users can estimate the sampling robustness of any network given an obtained sample. We evaluate the model and the results show that these model have good R^2 and it estimates sampling robustness with a small error. This lets the user understand whether the results of an analysis performed on a particular sample with errors are a good representation of the results one would have gotten from analyzing a sample without errors.

7. Conclusion

We presented a novel network robustness measure called "sampling robustness", which measures how much the performance of a network crawler changes when the edges are missing during the data collection process. We demonstrated that different network types have different level of robustness,

and that sampling robustness is highly dependent on the structural properties of the original graph. In addition, it is also correlated with the structural properties calculated from the obtained network samples. We presented models for estimating sampling robustness based on the observed properties in the samples. These models are capable of predicting sampling robustness with MSE less than 0.0015 and an R-squared of up to 75%.

In this work, we demonstrated the correlation between one type of sampling robustness and graph properties. However, our definition of sampling robustness is dependent on a particular performance measure. In our future work, we are interested in further investigating the relationship between sampling robustness and graph properties under different performance measures. We will also explore whether a theoretical analysis can give us a better understanding and estimation.

8. Acknowledgements

This material is based upon work supported in part by the U. S. Army Research Office under grant number #W911NF1810047.

References

[1] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 29–42.

[2] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *International conference on WWW*, 2007.

[3] T. Fu, A. Abbasi, and H. Chen, "A focused crawler for dark web forums," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 6, pp. 1213–1231, 2010.

[4] X. M. Woodley and M. Lockard, "Womanism and snowball sampling: Engaging marginalized populations in holistic research," *The Qualitative Report*, vol. 21, no. 2, pp. 321–329, 2016.

[5] G. A. Dusek, Y. V. Yurova, and C. P. Ruppel, "Using social media and targeted snowball sampling to survey a hard-to-reach population: A case study," *International Journal of Doctoral Studies*, vol. 10, pp. 279–299, 2015.

[6] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," in *IEEE INFOCOM*, vol. 1. Citeseer, 2004, pp. 120–130.

[7] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On unbiased sampling for unstructured peer-to-peer networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 2, pp. 377–390, 2009.

[8] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, "On near-uniform url sampling," *Computer Networks*, vol. 33, no. 1-6, pp. 295–308, 2000.

[9] C. Cooper and A. Frieze, "Crawling on web graphs," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 419–427.

[10] M. Kurant, M. Gjoka, C. T. Butts, and A. Markopoulou, "Walking on a graph with a magnifying glass: stratified sampling via weighted random walks," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 281–292.

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

[12] R.-H. Li, J. X. Yu, L. Qin, R. Mao, and T. Jin, "On random walk based graph sampling," in *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 2015, pp. 927–938.

[13] F. Chiericetti, A. Dasgupta, R. Kumar, S. Lattanzi, and T. Sarlós, "On sampling nodes in a network," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 471–481.

[14] J. Leskovec and 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.

[15] E. Ochodkova, M. Kudelka, and D. Ivan, "Sampling as a method of comparing real and generated networks," in *The Euro-China Conference on Intelligent Data Analysis and Applications*. Springer, 2017, pp. 117–127.

[16] N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 2, 2014.

[17] M. Kurant, A. Markopoulou, and P. Thiran, "On the bias of bfs (breadth first search)," in *Teletraffic Congress (ITC), 2010 22nd International*. IEEE, 2010, pp. 1–8.

[18] K. Areekijseree, R. Laishram, and S. Soundarajan, "Guidelines for online network crawling: A study of data collection approaches and network properties," in *Proceedings of the 10th ACM Conference on Web Science*. ACM, 2018, pp. 57–66.

[19] A. Saroop and A. Karnik, "Crawlers for social networks & structural analysis of twitter," in *Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on*. IEEE, 2011, pp. 1–8.

[20] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez, "Crawling a country: better strategies than breadth-first for web page ordering," in *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 2005, pp. 864–872.

[21] P. Hu and W. C. Lau, "A survey and taxonomy of graph sampling," *arXiv preprint arXiv:1308.5865*, 2013.

[22] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *nature*, vol. 406, no. 6794, p. 378, 2000.

[23] W. Ellens and R. E. Kooij, "Graph measures and network robustness," *arXiv preprint arXiv:1311.5064*, 2013.

[24] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Breakdown of the internet under intentional attack," *Physical review letters*, vol. 86, no. 16, p. 3682, 2001.

[25] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical review E*, vol. 65, no. 5, p. 056109, 2002.

[26] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, S. Qian, and J. P. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation," *Telecommunication systems*, vol. 52, no. 2, pp. 705–736, 2013.

[27] E. Estrada, "Network robustness to targeted attacks. the interplay of expansibility and degree distribution," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 52, no. 4, pp. 563–574, 2006.

[28] H. Chan and L. Akoglu, "Optimizing network robustness by edge rewiring: a general framework," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1395–1425, 2016.

[29] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, "Pay few, influence most: Online myopic network covering," in *Computer Communications Workshops*, 2014.

[30] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.* IEEE, 2003, pp. 25–34.

[31] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Unbiased sampling of facebook," *preprint arXiv*, vol. 906, 2009.