This dissertation is dedicated to
my family.
SAMPLING AND ESTIMATION ON LARGE ONLINE SOCIAL NETWORKS

by

EMRAH CEM, BS, MS

DISSERTATION
Presented to the Faculty of
The University of Texas at Dallas
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY IN
COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS
December 2016
ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my advisor, Dr. Kamil Sarac, for his continuous support, patience, and motivation. His guidance helped me in all the time of research and writing of this dissertation.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. András Faragó, Dr. Jason P. Jue, and Dr. Vibhav Gogate, for their insightful comments and encouragement. My sincere thanks also go to Dr. R. Chandrasekaran for always being around for guidance and keeping our motivation as high as possible by his nice conversations.

I would like to thank my friends Arunmoezhi Ramachandran, Nazim Ahmed, Richard Antiabong, Alex K. Mills, Kenneth Platz, Shreyas Gokhale, Aravind Natarajan, Erman Pattuk, Mehmet Kuzu, Mehmet Engin Tozal, and Huseyin Ulusoy for all the great times that we have shared.

I also thank the National Science Foundation for their financial support of the research under Award No. 1027520.

Last but not the least, I would like to thank my wife Sevinc Cem. This dissertation would not be possible without your support, encouragement, and patience. I am just so grateful to have you in my life. Thanks for the undying love and support you have provided me and for believing in me that I can finish this dissertation. This last word of acknowledgment I have saved for my little angel Linda Begum Cem, who has been one of my sources of inspiration in completing this dissertation.

September 2016
PREFAE

This dissertation was produced in accordance with guidelines which permit the inclusion as part of the dissertation the text of an original paper or papers submitted for publication. The dissertation must still conform to all other requirements explained in the “Guide for the Preparation of Master’s Theses and Doctoral Dissertations at The University of Texas at Dallas.” It must include a comprehensive abstract, a full introduction and literature review, and a final overall conclusion. Additional material (procedural and design data as well as descriptions of equipment) must be provided in sufficient detail to allow a clear and precise judgment to be made of the importance and originality of the research reported.

It is acceptable for this dissertation to include as chapters authentic copies of papers already published, provided these meet type size, margin, and legibility requirements. In such cases, connecting texts which provide logical bridges between different manuscripts are mandatory. Where the student is not the sole author of a manuscript, the student is required to make an explicit statement in the introductory material to that manuscript describing the student’s contribution to the work and acknowledging the contribution of the other author(s). The signatures of the Supervising Committee which precede all other material in the dissertation attest to the accuracy of this statement.
Studying the structural characteristics of online social networks (OSNs) provide useful information about the underlying OSN applications. Due to their very large sizes, sampling is a commonly used approach. One of the challenges in such studies is to develop proper statistical estimators under the limitations imposed by the OSN service providers. This dissertation explores various sampling schemes in estimating structural properties of OSNs.

First, we empirically show that the performance of the estimators highly depend on the studied characteristic and the underlying structure of the graph. Second, we propose estimators for the network size and the average degree under a very limited data access model which we call random neighbor access (RNA) model. The motivation is to understand the performance of the estimators when the OSN service providers limit the access to their data significantly.

Third, we propose various estimators for average degree under the ego-centric sampling. Each estimator utilizes different information in the sampled ego-networks. Finally, we propose estimators for clustering coefficient measures by combining Metropolis-Hastings random walk with wedge sampling.
5.7 Related Work ......................................................... 120
5.8 Conclusion .......................................................... 121
CHAPTER 6 CONCLUSION .............................................. 122
APPENDIX ERGODIC THEOREM ....................................... 125
REFERENCES ............................................................ 126
VITA
## LIST OF FIGURES

2.1 Jensen-Shannon divergence scores for synthetic graphs. Backend querying results (top) outperform Frontend querying results (bottom). .................................................. 17

2.2 Degree probability mass functions for the population graphs. .......................... 19

2.3 Barabási-Albert population graph. Degree vs. betweenness centrality. .......... 20

2.4 Barabási-Albert population graph. Degree vs. clustering coefficient. .......... 21

2.5 Clustering coefficient cumulative density functions for the population graphs. . 22

2.6 Scatter plots of degree vs clustering coefficient. .................................................. 25

2.7 Jensen-Shannon divergence scores for real-world graphs. .......................... 27

3.1 An example data collection from the OSN graph using the RN-QUERY calls. . 31

3.2 Proposed Estimators: Illustration of which estimator is used based on the sampling scheme and probing type. .......................................................... 31

3.3 Illustration of the proposed sampling designs with a sample size of 6 on a small population graph. .......................................................... 45

3.4 The accuracy of binomial approximation to the number of collisions $c_u$ for various $\omega$ and $d_u$ values. .......................................................... 49

3.5 Relation between number of RN-QUERY calls and degree for a given confidence level and relative error. (a) Relative error vs. the minimum $\omega$ needed for 90% confidence level (b) Confidence level vs. the minimum $\omega$ needed for the relative error $\epsilon = 0.1$. .......................................................... 52

3.6 Normalized estimates for various number of RN-QUERY calls and vertex degrees (left). Normalized root mean square errors of the estimations for various vertex degrees (right). .......................................................... 53

3.7 Relative error and coefficient of variation of number of visits to four vertices with different degrees in CONDMAT graph. .......................................................... 55

3.8 Average degree estimation on real-world graphs using SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators with no thinning, $t=1$. ................. 64

3.9 Degree distributions and statistics of the real-world graphs. ........................ 64

3.10 Average degree estimation on real-world graphs using SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators with thinning, $t=5$. ................. 65
3.11 Average degree estimation on synthetic graphs using SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators \textbf{with no thinning, }t=1. \hfill 67

3.12 Degree distributions and degree statistics of the synthetic graphs. \hfill 67

3.13 NRMSE of RNA-AVG-DEGREE estimator for various sampling fractions. \hfill 68

3.14 Network size estimation on real-world graphs using SRW-HHB-NETSIZE and RNA-HHB-NETSIZE estimators. \hfill 70

3.15 Network size estimation on real-world graphs using SRW-VCB-NETSIZE and RNA-MVCB-NETSIZE estimators. \hfill 71

3.16 Network size estimation on synthetic graphs using SRW-HHB-NETSIZE and RNA-HHB-NETSIZE estimators. \hfill 72

3.17 Network size estimation on synthetic graphs using SRW-VCB-NETSIZE and RNA-MVCB-NETSIZE estimators. \hfill 74

3.18 Average degree estimation on synthetic graphs using SRW-AVG-DEGREE-ADAPTIVE and RNA-AVG-DEGREE-ADAPTIVE estimators \textbf{with no thinning, }t=1. \hfill 77

3.19 Network size estimation on real-world graphs using SRW-HHB-NETSIZE-ADAPTIVE and RNA-HHB-NETSIZE-ADAPTIVE estimators. \hfill 78

3.20 Average degree estimation of dynamic friendship graph of digg.com, a social news website, over the period of approximately 4 years (from August 2005 to July 2009). \hfill 79

4.1 An example graph illustrating the two different cases for an ego-centric graph sample with 2 ego vertices (vertex 1 and 9). Vertex degree is available as a vertex label (on the left) and vertex degree is not available as a vertex label (on the right) cases are shown. \hfill 83

4.2 Free Neighborhood Information: Errors of estimators utilizing different information in the ego-centric graph sample of real-world graphs. \hfill 96

4.3 Non-free neighborhood information: Errors of estimators utilizing different information in the ego-centric graph sample of real-world graphs. \hfill 96

4.4 Comparison of average degree estimators EGOS and EDGES on Enron Email graph when the degree information is not available as a vertex label. Results for \textit{free-neighborhood} data are on the left, results for \textit{non-free neighborhood} data are on the right. \hfill 98

4.5 Effect of considering the ego-nodes in the estimation. \hfill 100

4.6 Effect of degree assortativity coefficient in the accuracy of the estimations. \hfill 101

5.1 Example population graph. \hfill 106

5.2 Comparison of the $c_{na}$ estimators. a) NRMSE of the estimators. b) The average running times of the estimators over 1000 independent experiments. \hfill 113
5.3 Convergence of the MH-WEDGE-CLUSTERING for $c_{na}$ on soc-epinions graph. 113

5.4 Comparison of the $c_t$ estimators. a) NRMSE of the estimators. b) The running times of the estimators. 119

5.5 Convergence of the MH-WEDGE-CLUSTERING for $c_t$ on soc-epinions graph. 120
LIST OF TABLES

2.1 Details of real-world networks used in the experiments. .......................... 24
3.1 Comparison of the estimators for SRW sampling with the adapted estimators for
the RNA model. Estimators for the RNA model do not include the safety-margin
dependency reduction technique to make it easier to compare with the estimators
for SRW. ............................. 47
3.2 Comparison of the descriptive statistics of $\text{Binomial}(\frac{\omega}{2}, 1/d_u)$ and the number
of collisions $c_u$ for various $\omega = \{10, 50, 100\}$ and $d_u = \{2, 5, 10, 5, 100\}$. .......................... 50
3.3 Details of real-world networks used in the experiments. .......................... 60
4.1 Table of Notation for the Network Model and Sampling Design. .......................... 88
4.2 Real-world topologies used in experimental evaluations. .......................... 95
5.1 Table of Notation for the Network Model and Sampling Design. .......................... 106
5.2 Population Parameters. ............................. 114
5.3 Estimation of $c_{na}$ with $mhrw$ for $\alpha = 0.05$ and $\epsilon = 0.1$ (relative), $n \approx 5730$. .......................... 117
5.4 Estimation of $c_t$ with $mhrw$ for $\alpha = 0.05$ and $\epsilon = 0.1$ (relative) $n \approx 8690$. .......................... 119
CHAPTER 1
INTRODUCTION

1.1 Motivation

The Internet has made new sources of vast amounts of data available and the concept of big data has gained momentum. The traditional storage and processing approaches are not scalable enough any more. Therefore, the development of efficient tools and techniques to solve large-scale problems have gained significant attention in the research community. One of the popular research fields in the big data analytics is the large-scale graph analytics. Its popularity can be attributed to the fact that many technological, social, transportation, and biological systems can be naturally represented as a graph. Usually, once the natural phenomena is represented as a graph, the next step is to study its structural properties to better understand the underlying mechanism producing specific structures in the graph. For instance, in proteomics, the large-scale study of proteins, it is believed that studying networks of the interactions of proteins will help to better understand the inner working of cells and provide better treatments for complex diseases [57].

One of the main challenges in large-scale Online Social Network (OSN) analytics is that collecting the entire graph from the underlying application is too costly and often not allowed by the OSN service provider due to privacy and business concerns. The data access limitations imposed by the OSN service providers generally make well-known sampling schemes such as independent random vertex sampling or random edge sampling inapplicable or inefficient. Therefore, designing estimators under alternative sampling designs and understanding the factors affecting their accuracy are important.
In this thesis, we focus on the implications of increasing data access limitations by OSN service providers in the estimation performance for important structural properties of OSN networks. We propose sampling schemes and corresponding estimators for some important structural properties of the graphs under various data access constraints. We evaluate their performances on various real-world large-scale OSN networks.

1.2 Outline of the Contributions

This thesis focuses on developing estimators for the structural properties of large network graphs under various data access limitations. The following outline describes what we address in each chapter.

Chapter 2: We discuss the important sampling design considerations while estimating structural properties of large graphs. We conduct large-scale experiments to show that 1) the accuracy of estimators based on the most popular sampling designs are highly dependent on the studied characteristic and underlying structure of the studied graph, and 2) the common practice of utilizing a readily available sampling design to generate a sample graph and use that subgraph to answer queries about the population graph should be done carefully.

Chapter 3: We propose estimators for the size and the average degree of OSN networks under a rather limited data access model which we call the random neighbor access (RNA) model. The main motivation is to observe how accurately the important properties of OSN networks can be estimated when the OSN providers limit third party access to network data significantly.

Chapter 4: We propose various estimators for the average degree under the ego-centric sampling design. Each estimator utilizes different information in the sampled ego-centric graph, we compare their performance through experiments on real-world graphs.

Chapter 5: We propose estimators for two important clustering coefficient measures 1) network average clustering coefficient and 2) transitivity, using Metropolis-Hastings random
walk (MHRW) and wedge sampling. Wedge sampling has been shown to be efficient in estimating the clustering coefficient measures [95], but it assumes the availability of independent vertex sampling which is unavailable or inefficient in OSN networks.

**Chapter 6:** We summarize our findings and give insights about the possible improvements and further developments.
CHAPTER 2
IMPACT OF SAMPLING DESIGN IN ESTIMATION OF GRAPH CHARACTERISTICS

2.1 Introduction

Recently, there has been a significant interest in modeling the real world complex systems as large scale graphs. The resulting graphs, in general, lack a simple structure and consist of numerous interconnected or interacting entities. These graphs are then studied to understand the structural and functional characteristics of the underlying real world systems as well as the interactions among the entities appearing in these systems.

World Wide Web graph, Internet topology maps at various levels, protein interaction networks, online social networks, and actor collaboration networks are among the examples to the complex systems that are represented as large graphs. Note that in complex systems context the terms graph and network are used interchangeably to refer to a graph data structure and we follow the same convention in this study.

Studying these complex systems might allow us to model the evolution of the Internet; predict the damage of an epidemic; reveal various characteristics of online social cliques; or identify the hub sites in World Wide Web. However, representing these systems as graphs on computers and analyzing their characteristics is usually uneconomical, impractical, or impossible due to the size and/or inherent limitations of these systems. To illustrate, capturing and studying the entire World Wide Web as a graph is extremely challenging.

---

if not beyond the capabilities of current computers [13]; discovering the complete router level topology of an Internet Service Provider (ISP) network is rather difficult unless it is provided by the ISP which is typically not the case due to security and privacy concerns; and mapping all interactions in protein interaction networks is experimentally difficult and time consuming [44].

Given that many of these systems are difficult to capture and/or impractical to analyze in their entirety, most studies resort to sampling to gather necessary information for the analysis. In general, the objective of network sampling can be divided into two: (1) designing sampling schemes that are compliant with the process that generates the population graph, i.e., the sampling design should produce sample graphs from the probability distribution over graphs given by the underlying graph generation process and (2) designing sampling schemes that will help us study the characteristics of a given population graph rather than the characteristics of the process generating that graph. The second objective above can be further divided into two: (2a) estimating characteristics of population graphs such as degree distribution, average clustering coefficient, or variation in path length, and (2b) obtaining a subgraph that preserves important topological characteristics of the population graph, i.e., representative subgraph sampling. Note that the objective (1) above is different from the objective (2b). Note also that it is difficult to come up with a sampling scheme that would satisfy both objectives (2a) and (2b) simultaneously [2].

Another practical issue that may impact network sampling in the context of the second objective above is the available querying mechanisms. In a given application domain, we may have two types of querying mechanisms available as backend querying and frontend querying. We say that backend querying is available if one can obtain the value of an entity characteristic from the population graph, e.g., degree of a sampled node refers to its degree in the population graph. We say that frontend querying is available if one can obtain the value of an entity characteristic from the sample graph, e.g., degree of a sampled node refers to its
degree in the *sample* graph. Note that in the context of *representative subgraph sampling*, we are mainly using *frontend* querying as the goal in this case is to build a representative subgraph.

Given that graph sampling may have various different types of objectives as mentioned above, it is important to know how to carefully design a sampling method. We believe that the effectiveness of a given graph sampling design depends on the following design considerations: (a) the characteristic of interest under study, (b) topology of the population graph (e.g., whether it follows scale free, small world, random, or semi-hierarchical graph models), and (c) available querying type (*frontend* vs *backend*). Note that design considerations that we address is comprehensive, but may not be exhaustive. As an example, a given graph sampling method may perform good for one characteristic of interest but may not perform well for some other characteristic under *frontend* querying [2]. Similarly, a given graph sampling method may preserve a given characteristic (e.g., type of degree distribution) of a random population graph in the sample graph but may not preserve the same characteristic of a scale-free population graph in the sample graph [97]. Finally, as we demonstrate in our evaluations, the available querying type may significantly impact the performance of a given sampling design in studying various characteristics of the population graph.

Based on our prior work in Internet topology mapping and our familiarity with studies in online social network (OSN) analysis, we realize that the above mentioned design considerations are not necessarily utilized in many studies that involve sampling from a population graph. In other words, in many studies in Internet topology measurement and OSN analysis (and possibly many others), there is a tendency to use readily available sampling schemes to quickly put together a sample subgraph and use it for analysis. The potential problems with such an approach include (1) use of a sampling method that is not inline with proper design considerations for the analysis study at hand and (2) misuse of the data obtained during the sampling process. An example for the first problem would be the direct use of metropolized
random walk with frontend querying, i.e., without carefully designing an estimator, to study shortest path length distribution of the population graph. An example for the second problem is to use a sampling design that is effective in producing samples for studying a particular characteristic (e.g., path length distribution) but use the resulting samples for studying some other characteristic (e.g., degree distribution) without ensuring the suitability of the sampling method for studying those additional characteristics.

In this study, we conduct a large scale experimental study to empirically demonstrate the relation between various sampling designs and the accuracy of resulting samples in estimating various characteristics of population graphs. We specifically employ the common practice of utilizing a readily available sampling design to generate a sample graph and use that subgraph with available querying to study various characteristics of the population graph. We then use a divergence metric to compare the similarity between the characteristic of interest obtained by the sampling process and the corresponding population characteristic. For the experiments, we use several types of synthetic graphs and several real-world graphs as population graphs. Therefore, the knowledge of the population graph enables us to compute the divergence scores as a metric to measure the accuracy of using various sampling methods in studying various graph characteristics. Note that if this were possible in practice, obviously there would not be a need for sampling at all.

Based on the conducted experiments, we observed that the performance of a sampling method significantly depends on the above design considerations. Therefore, while making inferences about a population graph using a sample graph within the application constraints, these design considerations should not be ignored. This study calls for further research in understanding the theory behind the ability to make inferences about population graphs using sample graphs obtained by a given sampling design.

The rest of the chapter is organized as follows. Section 2.2 presents the related work. Section 2.3 discusses our sampling framework. Section 2.4 demonstrates the experimental results motivating this study. Finally, Section 2.5 concludes the chapter.
2.2 Related Work

Early work in network sampling proposed statistical estimation techniques to accurately capture some fundamental characteristics of the underlying networks such as population size [28]. More recently, network sampling is utilized in several popular networking application domains such as Internet topology measurements (ITM), online social networks (OSN), and peer-to-peer networks (P2P) domains.

In ITM domain, sampling is used to estimate the size of the Internet [106]. Various studies also looked at the statistical properties of the sample networks obtained by traceroute-based path samples [1, 20]. In OSN domain, sampling is used in obtaining nodal or topological characteristics of the underlying social network. Most of the studies have focused on obtaining an unbiased sample or removing sampling bias in the collected sample network [69, 64, 65, 73, 88]. Other work demonstrated that certain biases can be exploited to increase the inclusion probability of desired properties during sampling [76]. In P2P domain, sampling has been studied with the goal of eliminating sampling bias due to skewed node degree distribution in overlay networks [41, 87, 100].

Another goal in network sampling has been to collect representative subnetworks from a population graph [54, 61, 40]. In this context, a subnetwork that matches many popular topological characteristics of the population graph is considered representative. Some of the most popular characteristics considered include degree distribution, clustering coefficient and betweenness characteristics. On the other hand, there has been concerns that building a subnetwork that matches an arbitrary set of popular topological characteristics of the population network may not always be feasible [76, 2].

In our work, we experimentally show that the utilized sampling strategy has an important effect on the representativeness of the resulting subnetwork even when we consider a single topological characteristic to match. We consider both backend and frontend querying and
utilize a small but diverse set of sampling schemes including (1) simple node and link sampling, (2) walk based sampling, and (3) path based sampling to demonstrate our points.

2.3 Sampling in Complex Systems

In the context of complex networks, sampling is typically used to collect a subgraph from the underlying system so as to estimate topological and functional properties of the population graph. As we argued in the introduction section, the accuracy of the obtained results in such a sampling scheme depends on various elements that we refer as design considerations that contribute to the sampling process. In this section, we present several examples for each design consideration as well as the sampling methods that were considered in this study. The examples presented below are commonly used in various sampling studies in sampling of complex systems.

2.3.1 Topological Characteristics of Graphs

A graph $G(V, E)$ is an ordered pair of a vertex set $V = \{v_1, v_2, \ldots\}$ and an edge set $E = \{e_1, e_2, \ldots\}$ such that $e_k = (v_i, v_j)$ and $v_i, v_j \in V$. In reality $V$ is the set of objects that are meaningful in a particular domain and $E$ is the set representing interactions or relations among these objects. Topological characteristics of graphs are those features that are related to the structural layout of the graph, e.g., degree distribution, clustering coefficient, betweenness, path length distribution, and diameter. In this study we focus on the topological characteristics of simple undirected graphs where $e_k = (v_i, v_j) \Rightarrow e_k = (v_j, v_i)$ without self edges. Below, we formally define the graph characteristics that we used in our experiments namely degree, clustering coefficient, and path length distributions.

Degree of a vertex $v_i$, also called the number of neighbors, denotes the number of edges incident to $v_i$ and it is represented by $d_i$. Degree distribution of a graph $G(V, E)$, is the probability distribution of degrees of vertices in $V$. In many networks, degree of a vertex
shows the popularity or importance of that vertex. For example, high degree vertices in OSN are those people having many friends, in AS level Internet topology networks they, most of the time, correspond to Internet Exchange Points (IXPs).

Clustering coefficient is a measure of closed connections appearing among vertex groups in a graph. Clustering coefficient $c_i$ of a vertex $v_i$ is defined as $\frac{2|R_i|}{d_i(d_i-1)}$, where $R_i$ is the number of connected pairs between all neighbors of $v_i$. Clustering coefficient distribution of a graph $G(V,E)$ is the probability distribution of clustering coefficients of vertices in $V$. High clustering coefficient indicates existence of tightly associated cliques in OSN and alternative bypassing paths around a particular router in router level Internet topology graphs. In our experiments, we assumed that vertices with degree $\leq 1$, have clustering coefficient of 0.

A path between $v_i$ and $v_j$ is a sequence of consecutive edges starting from $v_i$ and ending at $v_j$. Given that each edge in a graph is assigned a weight, a shortest path between $v_i$ and $v_j$ is one of the paths having the minimum total weight. In the most common case edges in a simple undirected graph are equally weighted which we also adopt in this study. Note that, there might be more than one shortest path between any two vertices. Path length, $l_{i,j}$, between $v_i$ and $v_j$ is the number of edges appearing on a shortest path between $v_i$ and $v_j$. Path length distribution of a graph $G(V,E)$ is the probability distribution of path lengths of vertex pairs in $V$. Path length between any two vertices in OSN is a measure indicating how easy to reach from one person to another via the intermediate people. Maximum path length in the router level Internet topology maps is called the diameter of the Internet and used as an indication of the maximum number of times that a packet can be enqueued/dequeued between two hosts.

Note that degree of a vertex involves 1-hop information around the vertex in the graph. Clustering coefficient of a vertex involves 2-hop information. Path length distribution is a characteristic that depends on the global structure of the network. As a result, we believe that the selected characteristics correspond to a small but diverse set for analysis purposes.
2.3.2 Graph Models

Many types of graphs with different distinguishing characteristics have appeared in the literature. Some graph types, however, occur so frequently in natural and man-made systems that they have been studied and used widely. In this part we introduce three popular graph types which frequently appear in the literature namely random, scale free, small world networks.

Random graphs are type of graphs that are obtained through a random process. Erdős-Rényi [26] (ER) is a widely used model for generating random graphs. ER model, $G(|V|, p)$, builds a random graph $G(V, E)$ by first creating $|V|$ isolated vertices and then, independently connecting each pair of vertices with probability $p$. All graphs having $|V|$ vertices and $|E|$ edges have the same probability of being generated, $p^{|E|} (1 - p)^{|V| - |E|}$, in ER model. However, as parameter $p \in (0, 1)$ increases dense graphs are more likely to be generated over sparse graphs. At $p = 0.5$ the model generates any possible graph of $|V|$ vertices with the same probability, $(0.5)^{|V|}$.

A graph, $G(V, E)$, having a power law degree distribution is called a scale free graph. Power law degree indicates the existence of a few vertices with very high degrees along with many vertices with less degrees. Degree distribution is said to conform power law if it is in the form of $P\{d_i = d\} \propto d^{-\alpha}$ where $\alpha$ is the scaling parameter [19]. Barabási-Albert [9] (BA) model is a widely used method for generating scale free networks. BA model is based on two principles (i) incremental growth and (ii) preferential attachment. Incremental growth assumes that the final graph, $G(V, E)$, is generated by introducing vertices one by one. Preferential attachment implies that the more connected a vertex is, the more likely it is to grab new links to be added to the network. In BA model, each new vertex is connected to $m$ existing vertices with a probability proportional to the number of edges that the existing vertices already have.
A graph, $G(V, E)$, having a large average clustering coefficient and a small average shortest path is called a small-world network. Large clustering coefficient implies the existence of highly clustered cliques in the graph. Small shortest paths refers to the existence of hub vertices or hub cliques within the graph. Small shortest paths denote that the average shortest path grows with respect to the logarithm of the number of vertices, i.e., $\mu_P \propto \log(|V|)$ where $\mu_P$ is the average path length and $|V|$ is the vertex set size. Watts-Strogatz [112] (WS) is the most widely used model for generating small-world networks. Given the number of vertices $|V|$ and the mean vertex degree $\mu_D$, the model creates a circular lattice where each vertex is linked to $\mu_D/2$ closest neighbor vertices on clockwise and counter clockwise. Then, with probability $\beta$ each edge of each vertex $v_i$ is rewired to a randomly selected vertex $v_k$ such that $i \neq k$. WS model guarantees high clustering coefficients by the initial circular lattice and decreases the shortest path length by randomly rewiring edges. The parameter $\beta$ controls the properties of the final graph. As $\beta = 0$ the graph would remain as a lattice with high clustering coefficient and high average shortest path. As $\beta \to 1$ the final graph resembles to ER graphs with $p = \mu_D/(|V| - 1)$.

Each of these graphs are used as underlying population graph and we use different sampling techniques to collect network samples from each of these graphs.

### 2.3.3 Sampling Methods

A sampling method refers to a systematic approach that is used to observe a set of sampling units from the population graph, where sampling unit refers to the entities in the population graph that is selected at each step during a sampling process. Typically, vertices and edges are used as observation units but other structural units such as end-to-end paths, star structures, triangle structures, etc., can also be considered as sampling units. In this study, we use vertices, edges, and end-to-end shortest paths as observation units.

In practice, the available observation units may be decided by the practical and operational limitations in the underlying application domain. As an example, in OSN analysis domain,
implementing a random vertex sampling may be extremely costly if the underlying OSN system has a sparsely populated identifier space. Most OSN applications, on the other hand, allow a random walk starting from a chosen vertex in the system allowing us to sample vertices and edges. On the other hand, in ITM domain directly observing a router or a subnet from a remote location is prohibited due to security and privacy concerns.

In network sampling applications, sampling methods are mainly determined based on the data collection mechanisms supported by the underlying application domain. It is often the case that a supported sampling method may not be an ideal one and may often result in sampling bias or high variance for the estimation process of the graph characteristic under study. In this study, we consider several sampling methods as discussed below.

Random vertex (RV) sampling selects a number of vertices from a population graph with equal probability. Although the technique is simple and effective in estimating the direct properties of vertices, e.g., degree distribution, with backend querying, it may not be readily supported by some complex systems such as the Internet or may be extremely costly to implement in some complex systems such as an OSN. We also use a version of RV sampling, called Induced RV (IRV) sampling which includes the existing edges from the population graph among the selected vertices into the sample graph. IRV sampling is also known as induced subgraph sampling. Random edge (RE) sampling selects a number of edges from the population graph with equal probability and also selects the incident vertices. This sampling method is also known as incident subgraph sampling. Induced random edge (IRE) sampling first performs RE sampling, then includes the existing edges from the population graph among the selected vertices if not selected during RE sampling. In general, RV, IRV, RE, and IRE sampling schemes may produce disconnected sample graphs.

Random walk (RW) sampling [73] initially designates a random vertex as the current vertex. At each step, it selects a new vertex among the immediate neighbors of the current vertex and designates it as the new current vertex. Random walk sampling process is governed by the stochastic process \( \{X_n, n \geq 0\} \) with transition probability
where $X_k$ is the random variable denoting the vertex selected at step $k$. For connected graphs with enough many steps the process converges to the stationary distribution $\pi(v_i) = d_i/2|E|$. That is, the process introduces bias by selecting larger degree vertices with higher probabilities compared to the smaller degree ones.

Metropolized random walk (MRW) removes the bias introduced by naive RW by rearranging the one step transition probability matrix of the RW stochastic process. Specifically, the transition matrix for MRW would be

$$P_{MRW}\{X_{k+1} = v_j|X_k = v_i\} = \begin{cases} 
1/d_i & \text{if } (v_i,v_j) \in E \\
0 & \text{otherwise}
\end{cases}$$ (2.1)

Thereby, MRW sampling reduces the probability of transitioning to high degree vertices to remove the bias introduced by the RW sampling.

Random shortest path (RSP) sampling selects paths among all possible shortest paths with equal probability. The total number of shortest paths on a simple graph $G(V,E)$ is at least $|V|(|V| - 1)$. Usually shortest path algorithms picks one path over the others in case there are multiple shortest paths between two vertices. Hence, the number of shortest paths is exactly $|V|(|V| - 1)$ in practice. Since enumerating all shortest paths in a graph is costly, RSP can be approximated by using shortest paths of randomly selected (source,destination) pairs uniformly without replacement. The sample graph in this scheme is constructed by carefully merging the shortest paths among the pairs of vertices. Variations of this approach in the ITM domain include sampling among K vantage points, i.e., (K,K) sampling, or sampling between K sources and M destinations, i.e., (K,M) sampling where (K$\ll$M).
2.3.4 Query Types

Query type is another important aspect of network sampling. One can analyze the topological properties of the sampled components in two ways: *backend querying* and *frontend querying*. In backend querying, a topological characteristic of a sampled unit is queried (or extracted) from the underlying population graph though, the entity appears in the sample graph.

In frontend querying, a topological characteristic of an entity is queried (or extracted) directly from the sample graph.

Since sample graphs do not preserve all information related to the population graph, frontend querying introduces another source of bias as opposed to backend querying. Therefore, the bias quantification and elimination should be handled differently in frontend and backend querying cases.

2.4 Experimental Results

In this section, we conduct experiments to observe the impact of the issues listed in the previous section on network sampling. Our goal is to present instances or evidence that sampling bias may emerge from various directions including the mismatch between sampling method and the characteristic under study; mismatch between the sampling method and the graph model; mismatch between the characteristic of interest and the utilized sampling unit, or frontend vs backend querying capabilities in sampling. The recent sampling literature includes studies that develop estimators for removing sampling bias for certain types of sampling methods and for certain types of topological characteristics [38, 64, 88]. Given that not all sampling schemes have proper estimators defined, in this work, we do not consider the use of such estimators.

We use the graph models listed in Section 2.3.2 as population graphs and take samples from them using the sampling methods described in Section 2.3.3. Then, using each query
type described in Section 2.3.4, we compute the distribution of characteristic (among the ones listed in Section 2.3.1) under study for each sample. Finally, to measure the difference between the sample and the population distributions, we use Jensen-Shannon (JS) divergence, a modification of Kullback-Leibler (KL) divergence, that is symmetric and robust with respect to noise and the size of histogram bins [92]. JS divergence is defined as

\[
d(H||K) = \frac{1}{2} \sum_i \left( h_i \log_2 \left( \frac{h_i}{m_i} \right) + k_i \log_2 \left( \frac{k_i}{m_i} \right) \right)
\]

where \( m_i = (h_i + k_i)/2 \).

In Equation 2.2, \( H \) and \( K \) are population and sample distributions defined over the same sample space. \( h_i \) and \( k_i \) are the relative outcome frequencies for discrete distributions and relative histogram bin frequencies for continuous distributions belonging to \( H \) and \( K \). Note that the Jensen-Shannon divergence takes values between 0 and 1 since we use the base 2 logarithm. In the experiments, we take samples of size 10% from each population graph by using sampling methods described in Section 2.3. Each divergence score presented in this study is the average of 30 experiments. In general, as the sample size increases, the estimation gets closer and closer to its true population values and 10% is a commonly used sample size in the literature.

### 2.4.1 Results on Synthetic Networks

In this section, we generate random, scale-free, and small-world population graphs by using Erdős-Rényi (ER), Barabási-Albert (BA), and Watts-Strogatz (WS) models, respectively. These models generate graphs with different topological characteristics which enable us to observe the effects of the graph models on the results. Unless otherwise stated, for each model, we generate synthetic graphs of size 50,000 with average vertex degree 20 to compare the graph models properly. To get the desired average degree, we set the model parameters as \( \text{BA}(m = 10) \), \( \text{ER}(p = 0.0004) \), \( \text{WS}(\mu_D = 20, \beta = 0.5) \). We set the number of vantage
points to 3% of the population size in K-K sampling method. In K-M sampling, we set the number of source and destination vertices to 3% and 7% of the population size, respectively. Since K-K and K-M sampling methods are mostly utilized in Internet measurement studies and the number of available ‘vantage points’ is often very limited, we selected parameters so as to reflect the real-world usage better as well as to sample 10% of the population graph.

Fig. 2.1 presents the Jensen-Shannon divergence values of our experiment results for the three characteristics. We omit random vertex (RV) sampling results in the figure because it is not applicable for the frontend querying (FE) since no edges are sampled and it gives same results as IRV sampling for the backend querying (BE). Note that random walk based sampling methods require the information on the neighboring vertices be available to the sampling process in order to sample the next step vertex. Hence, one might question the use of FE querying for such cases. However, companies in various domains use a random walk based method to sample their proprietary networks and publicize the resulting sample graph for analysis purposes without providing any further access to their population networks.

Based on the results in Fig. 2.1, we can conclude that for any sampling method, BE querying outperforms FE querying when it is applicable to the synthetic graph models.
Although this is an expected result, the difference can be attributed to the fact that the vertex based sampling methods tend to under-sample the edges of a population graph. On the other hand, computation of many graph characteristics including degree, clustering coefficient, and path length involves edge information as well. The edge related information of the population graphs is available to BE querying while FE querying can only rely on the edge information provided by the sample graph. Because RV, IRV, and MRW sampling methods select vertices uniformly at random, they generally perform well when used with BE querying. On the other hand, RE, IRE, and RW sampling with BE querying does not produce similar results because they tend to over sample high degree vertices compared to low degree ones. RSP and K-M methods perform better than the K-K method though, RSP is slightly better than K-M for most of the cases. Path based sampling bias mostly occurs around the source and destination vertices. That is, the probability of a vertex being sampled is higher for those vertices which are located closer to the source and destination vertex sets in terms of hop count. In our experiments both the source and destination sets are constructed uniformly at random. On the other hand, K-M and RSP possess more diversity in terms of the destination set compared to the K-K method. The bias due to the diversity around the destination sets is reflected in Fig. 2.1.

Note that quite often real world systems do not support (or provide service for) RV and IRV sampling, e.g., WWW and OSN. Hence, Fig. 2.1 suggests that MRW sampling with BE querying is a practical and well-performing method for clustering coefficient, degree, and path length distribution estimation.

**Degree Distribution**

Fig. 2.1-b and Fig.2.1-e show the degree distribution divergence scores of different sampling methods on our graph models for BE and FE querying, respectively. Fig. 2.1-b shows that RV and MRW with BE querying outperform all other methods. This observation is not
surprising because RV with BE querying is always the best sampling method to estimate any local vertex characteristic. Unfortunately, RV sampling with BE querying is not available in many domains including OSN, ITM, and WWW. MRW with BE querying also performs well because it could be considered as an approximation of RV with BE querying for degree distribution. MRW sampling with BE querying is supported in WWW domain, but it is not directly supported in OSN and ITM domains due to privacy concerns and technical limitations, respectively.

![Probability Mass Functions](image)

(a) BA  
(b) ER  
(c) WS

Figure 2.2. Degree probability mass functions for the population graphs.

Interestingly, all sampling methods with BE querying performed worse on BA population graphs compared to ER and WS models. Analyzing the probability mass functions (Fig. 2.2) of those population graphs shows that ER and WS models produce almost symmetric degree distributions while BA model naturally produces a power law graph. That is, BA graphs have a small number of high degree vertices along with high number of small degree vertices. RE and IRE tend to sample the high degree vertices with higher probability. Similarly, naive RW tend to visit high degree vertices with high probability. As shown in Fig. 2.3, in BA graphs, those high degree vertices also have high betweenness centrality values. That is, the number of shortest paths passing over those high degree vertices is significantly larger than the paths passing through the vertices having small degree. Hence, path based sampling methods (RSP, K-K, and K-M) sample high degree nodes with higher probabilities as in the other sampling methods. This analysis clearly shows that the underlying graph models have significant impacts on sampling where all other parameters are held fixed.
Fig. 2.1-e demonstrates that all sampling methods almost equally perform bad with frontend querying on all graph models for degree distribution. The divergence can be attributed to the fact that not all edges of a vertex could be sampled and included into the sample graph. As a result, in all cases degrees of vertices are under-sampled and this artifact is reflected in the divergence results. On the other hand, the sampling methods perform marginally better on BA graphs. The reason is simply BA graphs have many small degree vertices and their edges are sampled better compared to those having large degrees. As a special case in Fig. 2.1-e IRE sampling performs better on BA graphs. In addition to the existence of many small degree vertices in BA graphs, IRE adds additional edges into the sample graph if any two vertices are already sampled. Hence, the degree of a vertex is closer to the population value with IRE sampling.

**Clustering Coefficient Distribution**

Fig. 2.1-a and Fig.2.1-d show the clustering coefficient distribution divergence scores of different sampling methods on our graph models for backend and frontend querying, respectively. Both figures strongly articulate the underlying graph models’ impact on performance of the utilized sampling methods.

Fig. 2.1-a demonstrates that all sampling methods perform acceptably well with BE querying on ER and WS graphs. On the other hand, all methods but IRV and MRW
relatively perform worse on BA graphs. Our further analysis shows that the difference between BA graph model and ER and WS graph models can be attributed to the relation between the degree and clustering coefficient distributions of these models. Specifically, BA graphs consist of small number of of high degree vertices and high number of small degree vertices. Furthermore, as shown in Fig. 2.4, low degree vertices in BA graphs have higher clustering coefficient values. In fact, this outcome is quite natural because of the preferential attachment concept incorporated into the BA graph generation algorithm. That is, a newly introduced vertex tends to get connected to a higher degree vertex rather than its relatively lower degree first hop neighbors. RE and IRE methods tend to under-sample lower degree vertices because they have less edges. RSP, K-K, and K-M sampling methods also tend to under-sample low degree vertices because those vertices have lower betweenness centrality (see Fig. 2.3). As a result, the divergence score increases for these methods because they cannot fairly sample vertices having high clustering coefficient. Note that the arguments do not apply to IRV and MRW on BA graphs because IRV samples vertices uniformly at random and MRW removes the degree bias which indirectly reduces the clustering coefficient bias.

Fig. 2.1-d shows that FE querying performs worse than BE querying for clustering coefficient as well. This behavior may again be attributed to the vertex based sampling
methods under-sampling the edges hence underestimating the clustering coefficient values in many FE querying cases as shown in Fig. 2.5. Further analysis shows that we can explain the differences of the divergence scores among different population graph models by the existence of many vertices having clustering coefficient zero. The vertices with degree one have zero clustering coefficient by definition. Hence, if a vertex with degree one is sampled and added into the sample graph, it is sampled with all of its edges (one) and its true clustering coefficient value (zero). Moreover, since all sampling methods under-sample the edges in the population graphs, the number of vertices having zero clustering coefficient values increases in the sample graph. As a result, sampling schemes introduce vertices having artificial zero clustering coefficient. As shown in Fig. 2.5, around 70% of the vertices in the population BA graph and 93% of the vertices in the population ER graph have zero clustering coefficient values. Because of edge under-sampling, those vertices with zero clustering coefficient values in the population graph are compensated in the sample graph hence resulting in less divergence.

On the other hand, the same argument is not applicable to WS model because the clustering coefficient distribution has a larger range and variance.

Path Length Distribution

Fig. 2.1-c and Fig.2.1-f show the path length divergence scores of different sampling methods on our graph models for BE and FE querying, respectively.
Fig. 2.1-c shows that all sampling methods perform well with BE querying on ER and WS graphs. On the other hand, all methods but IRV and MRW relatively perform worse on BA graphs. Further analysis shows that the difference between BA graph model and ER and WS graph models can be attributed to the vertex degrees. Since BA graphs consist of small number of of high degree vertices and high number of small degree vertices; RE, IRE, and RW sampling methods tend to under-sample low degree vertices occurring at the periphery of the graph. A similar argument is applicable to RSP, K,K, and K,M sampling methods because these methods favor vertices having high betweenness centrality which in turn have high degrees. As a result, the sampled vertices are the ones mostly appearing in the core of the graph rather than periphery and they have comparably shorter path lengths in the population graph.

The divergence scores conveyed in Fig. 2.1-e are especially surprising. All methods except RSP with frontend querying perform bad on ER and WS graphs. Fig. 2.5 shows the cumulative densities of the population graphs as well as the sample graphs obtained by all sampling methods. One counter intuitive observation is that the lengths of observed sample paths as well as the diameter of the sample graph are much higher than the population graph except for RE sampling. The artifact simply occurs due to vertex based sampling methods under-sampling edges in the population graphs. Many of the shortest paths appearing in a population graph are not sampled because of one or more unsampled edges. As a result, path lengths between many sampled vertices in sample graphs get increased. On the other hand, further analysis shows that the reason that RE sampling behaves differently is that this sampling scheme generates highly disconnected graphs having smaller shortest paths. Finally, RSP performs better than K-K and K-M schemes because it does not have the bias due to sampling vertices closer to the preselected source and destination vertices.

BA graphs have core vertices with high degrees and high betweenness centrality. These vertices are, in general, sampled into the sample graph as well. Specific to IRE sampling
method, all edges induced to the neighboring sample vertices of the core vertices are also sampled. Since these vertices carry most of the shortest paths between other vertices, the path length distribution is preserved in the sample graph. As a result, IRE with FE querying performs very well on BA graphs (see Fig. 2.1-f).

2.4.2 Results on Real-World Networks

In this section, we present our experimental results on real world graphs shown in Table 2.1. CAIDA, CO-AUTHOR, ENRON, and EPINIONS graphs can be obtained at http://snap.stanford.edu. The overall results are presented in Fig. 2.7 where the rows correspond to BE and FE query types and columns correspond to degree distribution, clustering coefficient distribution, and path length distribution characteristics, respectively. The top row is for BE querying and the bottom row is for FE querying cases. Similar to the synthetic topology based experiments, our results for path length distribution with BE querying assumes that selected nodes know their path length distribution in the topology graph. In practice, this information is rather difficult to collect and therefore the results here show the best case scenario.

The bar charts in Fig. 2.7 present the summary results for the experiments. We use several cases from Fig. 2.7 to illustrate the effects of design considerations as well as some interesting observations.

Table 2.1. Details of real-world networks used in the experiments.

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Application domain</th>
<th># of nodes</th>
<th># of edges</th>
<th>degree (min:mode:max)</th>
<th>average clustering coefficient</th>
<th>diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAIDA</td>
<td>Internet topology</td>
<td>26,475</td>
<td>106,762</td>
<td>1:2:2628</td>
<td>0.010</td>
<td>17</td>
</tr>
<tr>
<td>CO-AUTHOR</td>
<td>Collaboration (CONDMAT)</td>
<td>21,363</td>
<td>182,628</td>
<td>1:5:281</td>
<td>0.633</td>
<td>15</td>
</tr>
<tr>
<td>ENRON</td>
<td>Email exchanges</td>
<td>33,696</td>
<td>361,622</td>
<td>1:3:1383</td>
<td>0.497</td>
<td>12</td>
</tr>
<tr>
<td>EPINIONS</td>
<td>Social(Trust)</td>
<td>75,877</td>
<td>508,836</td>
<td>1:2:3044</td>
<td>0.228</td>
<td>13</td>
</tr>
<tr>
<td>FACEBOOK [107]</td>
<td>Social(Friendship)</td>
<td>63,392</td>
<td>816,886</td>
<td>1:10:1098</td>
<td>0.279</td>
<td>15</td>
</tr>
<tr>
<td>GNUTELLA</td>
<td>P2P</td>
<td>65,561</td>
<td>147,878</td>
<td>1:2:95</td>
<td>0.010</td>
<td>11</td>
</tr>
<tr>
<td>YEAST-PPI [55]</td>
<td>Biological</td>
<td>1,458</td>
<td>1,993</td>
<td>1:2:56</td>
<td>0.070</td>
<td>19</td>
</tr>
</tbody>
</table>
Effect of population graph

The variations of the divergence scores across different application domains (see Fig. 2.7-d) show that the underlying population graph has an impact on the results when the same sampling method and the same query type are utilized. This is also visible in other charts in the same figure.

Fig. 2.7-a, 2.7-b, and 2.7-c show that, for each network, relative performances of sampling methods for all three characteristics are similar with BE querying. For instance, in EPINIONS network, for all three characteristics, IRV (and hence RV) sampling method performs the best; RE, IRE, and RW sampling methods perform the worst. This observation is also apparent for other networks. With BE querying, the performance of the sampling method depends on which vertices it selects. As long as there is a correlation between different characteristics, we expect to observe similar results. As depicted in Fig. 2.6, as the vertex degree increases, clustering coefficient of the vertex decreases, which is very natural when the definition of clustering coefficient is considered. As the vertex degree increases, to keep the clustering coefficient the same, the number of edges between the neighbors should increase quadratically. However, this is not satisfied, especially for high degree vertices, in any real-networks that we considered. The similar performance results in Fig. 2.7-a and Fig.2.7-b can be explained in a similar way. However, this result does not imply that if a sampling design performs well for one characteristic, it will also perform well for any other characteristic [2].

Note that this trend is not observed in FE querying. In FE querying, performance of the sampling method depends on the topology of the sampled graph since no information can be
extracted from the population. Therefore, observing similar results for different characteristics is not expected especially if the nature of the characteristics are different.

**Effect of characteristic**

The mismatch between the characteristic of interest and the sampling design will affect the results of a sampling study. For instance, walk based sampling methods perform poorly for the estimation of path length distribution with frontend querying (see Fig. 2.7-f). This is because they mostly create sparse subgraphs by under-sampling edges causing overestimation of shortest path lengths. However, in Fig. 2.7-f we observe a different behavior for YEAST-PPI network. This difference can be attributed to the sparsity of the population graph. Moreover, by further analysis, we noticed that the topology of the YEAST-PPI network has lots of chain structure, which fits to the nature of the walk-based designs since the walk-based designs select only one edge of a vertex at each visit and create sparse sample graphs.

Divergence scores of MRW sampling for GNUTELLA network in Fig. 2.7-d, Fig.2.7-e, and Fig.2.7-f are 0.031, 0.236, and 0.950 for clustering coefficient, degree and path length distributions, respectively. This result shows that the characteristic of interest is important for the performance of the utilized sampling design. By analyzing the clustering coefficient distribution of GNUTELLA network, which is not shown here due to space limitations, we observed that more than 90% of vertices have zero clustering coefficient similar to the ER graph. Since the MRW sampling method under-samples the edges, it introduces vertices having artificial zero clustering coefficient. The vertices with zero clustering coefficient values in the population graph are compensated in the sample graph by those vertices hence resulting in less divergence. High divergence score for the path length distribution can be attributed to the mismatch between the sampling design and the characteristic of interest.
Figure 2.7. Jensen-Shannon divergence scores for real-world graphs. The variability in the performance of sampling designs depicts the effects of the following design considerations: (a) the characteristic of interest under study, (b) topology of the population graph, and (c) available querying type.

Effect of query type

Fig. 2.7 depicts that the BE querying (top row) results are significantly different than FE querying results (bottom row) in most cases. For instance, in Fig. 2.7-c and Fig.2.7-f, for YEAST-PPI network, even though IRV and RE sampling methods perform better than path-based sampling methods under BE querying, the situation is the reverse for FE querying. This result shows that query type also has an impact on the performance of the sampling design.

In addition to above results, we also observe that in some cases FE querying performs better than BE querying, which we did not observe in synthetic graph analysis. For instance in Fig. 2.7-b and Fig.2.7-e, for CAIDA and YEAST-PPI networks, degree distribution results with FE querying outperform the degree distribution results with BE querying in walk-based sampling methods. We believe that this is a quite interesting result. Existence of such cases may motivate researchers to do further analysis to understand the interaction
between the sampling design, the characteristic of interest, and the topology of the network. Understanding this interaction may give clues about the feasibility of making inferences about a given population characteristic using the sample graph.

2.5 Future Work & Conclusions

In this study we demonstrate that the design of a sampling scheme plays an important role on estimating various network characteristics. Our empirical evaluations clearly shows that network characteristic estimation problem depends on many factors and should be handled individually per characteristic and per sampling scheme supported by the underlying system. **Even though we consider important factors of sampling biases in this study, they may not be exhaustive. Therefore, we plan to conduct a theoretical study supporting our findings in this empirical study. Furthermore, we plan to develop working instances of our proposed sampling framework and device new statistical estimators for different features of real world complex systems including OSN, ITM, P2P, and WWW.**
CHAPTER 3

ESTIMATION OF STRUCTURAL PROPERTIES OF ONLINE SOCIAL NETWORKS AT THE EXTREME

Online social networks (OSNs) such as Facebook, Twitter, and LinkedIn have become an important part of our daily life. They not only create a platform to connect people but also provide a means to analyze and characterize the behavior of their users. Estimating the structural properties of the underlying networks of OSNs using the statistical estimation theory has drawn a lot of attention [21, 58, 47, 117, 65, 63, 64, 39, 37]. Understanding the structure of these OSN graphs might help to develop algorithms for detecting trusted or influential users [96, 81]; mitigate email spam [34]; defend against Sybil attacks [118]; and improve Internet search [27].

The main challenge in estimating the characteristics of OSNs for third parties is the limited access to the data of interest through a public application program interface (API). In principle, sampling enables us to make population level estimations if data is collected with known statistical properties. However, limited data access can make it challenging to develop practical sampling schemes with known statistical properties. Data access limitation in OSNs is generally in two different forms: 1) query rate limitation and 2) authorization. Query rate limitation means that the number of API calls that can be made within a specified time period is limited by the OSN service provider. This limitation affects how fast the graph can be sampled. The practitioners generally partially overcome this limitation by using a parallel

---

crawling approach to collect the data [39, 17]. Even if the entire dataset is accessible and parallel crawling is used, accessing all the data is generally not efficient and sampling is a common way of increasing the efficiency at the expense of accuracy. Authorization is also an important feature of today’s OSN services as it provides privacy for the OSN users. However, this feature makes statistical estimation impossible without making further assumptions about the unobserved data.

In this study we consider the estimation problem with a limited data access model called the random neighbor access (RNA) model. The only public API query that is available under the RNA model is the random neighbor query (RN-QUERY). The RN-QUERY returns the id of a neighbor selected uniformly at random and independently for a given vertex id (see Fig. 3.1). The RNA model is relevant because OSN service providers are imposing more and more limitations for third party access to their data and users are becoming more aware of privacy issues and share the data with limited number of people [23]. As one of the possible extreme cases of this trend, we consider the RNA model. Our main motivation is to investigate the predictability of important structural properties under the RNA model. Although the model allows very limited data access, the probabilistic nature of the access model to the data allows us to make statistical estimations without further assumptions on the data. Studying the RNA model helps us understand the performance limitations of the estimators developed for an extreme scenario. In this study, we specifically focus on the estimation of two important characteristics namely the average degree and the network size. Fig. 3.2 illustrates the difference of the proposed estimator based on the sampling scheme and the probing type.

Sampling has been commonly used in OSN contexts but existing schemes are not applicable under the RNA model. Random walk (RW) [78, 7, 88, 90, 38, 39, 64, 115, 110] is the most commonly used sampling scheme in OSNs. It is well-known that the simple random walk (SRW) sampling in which a neighbor is selected uniformly at random at each step is biased
Figure 3.1. An example data collection from the OSN graph using the RN-QUERY calls. Note that we call the RN-QUERY twice for the vertex with id 4 and get different neighbors by chance.

Figure 3.2. Proposed Estimators: Illustration of which estimator is used based on the sampling scheme and probing type. Probing type is applicable only on ERSRW sampling scheme. We propose average degree estimator under only ERSRW sampling, so choosing the sampling scheme step is not applicable.

towards high degree nodes, but its bias is known and can be corrected for connected and non-bipartite graphs using the vertex degrees [50, 93]. An implicit assumption in the previous works is that the degrees of vertices visited by SRW are observable. However, we only observe the ids of the visited vertices under the RNA model. Therefore, previous estimators are not directly applicable. More recent studies [88, 90, 37, 115] propose various versions of RWs to improve the efficiency of the estimator for certain graphs. However, these estimators are also not directly applicable under the RNA model as the efficiency is generally achieved by assuming access to more data such as the availability of independent node sampling. Breadth-first search (BFS) and depth-first search (DFS) sampling methods are also easy to
apply on OSNs, but it is not known how to correct their bias towards high degree nodes for arbitrary graphs [67, 66]. They are also not very efficient under the RNA model as the neighbor selection is not deterministic under the RNA model.

The main contributions of this study are as follows:

1. We propose various sampling schemes and corresponding average degree and network size estimators under a rather limited probabilistic data access model called RNA model.

2. We propose an estimator for the reciprocals of the vertex degrees which we use in the estimation of the average degree and the network size properties. We provide both theoretical analysis and simulation based experiments on its accuracy.

3. We conduct extensive simulation studies on both real world and synthetic graphs (1) to measure the performance of the proposed estimators and (2) to study the factors that affect the accuracy of these estimators.

Our main findings are as follows:

1. The network size estimation is a challenging task under the RNA model and requires large number of RN-QUERY calls (e.g., more than the actual network size, even 40-50 times the network size in some cases). However, the average degree estimation can be done efficiently with the selection of a reasonable sampling design parameter.

2. Applying dependency reduction technique to the samples does not necessarily improve the estimation performance.\(^2\) For instance, in the estimation of average degree, dependency reduction technique reduces the precision of the estimation; while in the

\(^2\)We use the term estimation performance as a combined measure of precision and accuracy. Precision is a measure of the stability (hence inversely proportional to the variance of the estimator) while accuracy is a measure of being close to the true parameter (hence proportional to the bias of the estimator) on the average.
estimation of the network size, it increases the accuracy, especially when the sampling fraction \( f > 1 \).

3. The dynamic nature of the underlying graph adds one more layer to the complexity of the estimation problem. The accuracy of the estimation is limited by how fast the samples can be collected and how fast the property of interest changes. As opposed to the static graph case, larger sample sizes do not provide better estimation results especially when the property of interest increases or decreases over time as the old data becomes unrepresentative of the current data.

The outline of the chapter is as follows: Section 3.1 presents the background and the related work. Section 3.2 presents the RNA model and sampling designs. Section 3.3 presents the estimators for the RNA model. Section 3.4 presents our experimental evaluations. Section 3.5 discusses the practical issues. Section 3.6 concludes the chapter.

### 3.1 Background and Related Work

The RNA model enables us to perform a walk, but not a SRW\(^3\) on the underlying graph. Nevertheless, the estimation techniques proposed under SRW sampling form the basis for those under the RNA model as the sampling schemes that we use under the RNA model, namely RSRW and ERSRW, are the modifications of the SRW. In this section, we first introduce the graph and sampling terminologies. Then, we briefly discuss SRW sampling and present the well-known and the state-of-the-art estimators for the average degree and network size properties under SRW sampling. We also discuss graph traversal methods (e.g., breadth-first search (BFS) and depth-first search (DFS)) and two relevant Monte Carlo (MC) methods and explain why they are not practical or applicable under the RNA model.

\( ^3 \)In SRW, the vertex degrees of visited vertices are assumed to be known.
3.1.1 Graph Terminology

We denote an OSN graph by \( G = (V, E) \) where \( V \) is the finite set of nodes/vertices and \( E \) is the set of links/edges. We use \( N = |V| \) and \( M = |E| \) to denote the network size and the number of edges, respectively. We use \( d_v \) to denote the degree of \( v \in V \). We use \( \tau_d = \sum_{u \in V} d_u \) and \( \mu_d = \frac{\tau_d}{N} \) to denote the total and the average degree of \( G \), respectively.

3.1.2 Sampling Terminology

We use \( S \) and \( s_i \) to denote the list of sampled nodes and the \( i^{th} \) sampled node, respectively. We use \( n = |S| \) and \( f = n/N \) to denote the sample size and the sample fraction, respectively. We use \( \mathcal{S} \) to denote the set of distinct vertices in \( S \). We use \( \Delta_u \) to denote the number of times vertex \( u \) is sampled and call it the multiplicity of vertex \( u \). We say that a collision occurs when the \( i^{th} \) and the \( j^{th} \) sampled vertices are the same for \( i < j \). Finally, we use \( C = \sum_{u \in \mathcal{S}} \binom{\Delta_u}{2} \) to denote the total number of collisions in the sample. We use \( \binom{n}{k} \) to denote the binomial coefficient, i.e., the number of ways of picking \( k \) unordered outcomes from \( n \) possibilities. For instance, if \( S = \{u, v, t, u, t, u\} \), then \( C = \binom{3}{2} + \binom{1}{2} + \binom{2}{2} = 4 \) since \( \Delta_u = 3, \Delta_v = 1, \) and \( \Delta_t = 2 \).

3.1.3 SRW Sampling Design

RW is a finite state time-reversible Markov chain and it has a well-established theory [73]. RW starts from a vertex and repeatedly moves to a random neighbor node. If the neighbor node is selected uniformly at random, it is called an SRW [119, 115]. More formally, let \( \{s_t\} \) be the Markov chain representing the sequence of sampled nodes by SRW. The transition probability matrix \( P_{SRW} = \{p_{uv}\}_{u,v \in V} \) of SRW, where \( p_{uv} \) is the probability of moving from vertex \( u \) to vertex \( v \), is given by
Researchers develop variations of RW such as Metropolis-Hastings RW \([99, 87]\), multiple dependent RW \([88]\), multigraph RW \([37]\), RW with jumps \([51, 71]\) and weighted RW \([64]\), to sample certain types of graphs more efficiently. However, the RNA model does not allow to make these modifications efficiently. For instance, they require independent node selection and/or the knowledge of the degrees of both the current node and the candidate node in order to efficiently choose the next node in the RW (see Eq. 3.8).

Stationary distribution of a walk is an important concept that defines the probability of being at any particular vertex. The fundamental theorem of Markov chains\([85]\) states that every finite, irreducible, aperiodic Markov chain has a unique stationary distribution. Irreducibility and aperiodicity are achieved on connected and non-bipartite graphs. A walk is called stationary random walk if the initial vertex in the walk is selected from the stationary distribution \(\pi = \{\pi_u\}_{u \in V} = \{d_u/2M\}_{u \in V}\).

Even though the initial vertex selection from the stationary distribution is not trivial, we briefly discuss how it is generally achieved by practitioners in Section 3.5.

Another important property of a RW is that there is a strong dependence on the consecutive samples. Thus, using estimators that assume independent vertex selection may yield inaccurate results if the samples obtained by a random walk are used directly. Kurant et al. \([63]\) discuss some practical dependency reduction techniques such as safety-margin and shifted-thinning. In the safety-margin technique with parameter \textit{margin}, the information obtained from the pairs of sampled nodes \(s_k\) and \(s_l\) that are close to each other, i.e., \(|k - l| \leq \text{margin}\), is ignored and the estimator is adjusted accordingly. For instance, the information might be whether there is a collision or not. In the shifted-thinning technique with thinning parameter \(t\), sampled vertices are divided into \(t\) groups. The first group contains every \(t^{th}\) sample starting...
from the first sample, the second group contains every $t^{th}$ sample starting from the second sample, and so on. Then, an estimation is performed for each group independently and the average of $t$ estimations is considered as the final estimation. We use these techniques in our estimators to reduce the dependency in the sample.

### 3.1.4 Estimators for SRW Sampling Design

In this section, we introduce the existing estimators for the average degree and network size properties when the data is collected by a SRW sampling and the degree information is observable at the sampled vertices. The estimators are not exhaustive but gives an overview of the known statistical techniques and recent advancements. The estimators that we introduce here form the basis for our estimators. We adapt the ideas presented in this section to the RNA model in Section 3.3.

#### Average Degree Estimation

The GENERALIZED estimator [102], a.k.a. self normalized importance sampling, is one of the most popular estimators for the average characteristics such as average age of users or average number of friends. Even though it is not restricted for vertex characteristics, we only discuss it for vertex characteristics for simplicity. It requires that the selection probability of each sampled vertex is independent, strictly positive and is known up to a constant.\(^4\) It is an asymptotically unbiased estimator and is defined as

\[
\hat{\mu}_y = \frac{\sum_{u \in S} y_u / w_u}{\sum_{u \in S} 1 / w_u},
\]

(3.2)

where $S$ and $y_u$ are the list of sampled nodes and the property of interest, respectively. $w_u$ is the selection weight of vertex $u$, i.e., the selection probability of vertex $u$ up to a constant.

\(^4\)An alternative is the Hansen-Hurwitz (HH) estimator which additionally requires the network size and the exact value of the selection probability of each sampled vertex.
In a stationary SRW, vertex $u \in V$ is sampled with probability $d_u/2M$. Therefore, we can select $w_u$ as $d_u$. As we estimate the average degree, we replace $y_u$ with $d_u$ and get the following estimator:

$$
\hat{\mu}_d = \frac{n}{\sum_{u \in S} 1/d_u}.
$$

(3.3)

The issue with this estimator is the dependency of the sampled vertices. The ergodic theorem\[85] states that the effect of the dependence in the sample disappears in the limit of the large sample size. However, we reduce the dependency of the sampled vertices by applying a shifted-thinning [63] to the random walk as we are interested in the finite sample sizes. Applying shifted-thinning to Eq. 3.3 gives the following average degree estimator which we call SRW-AVG-DEGREE:

$$
\hat{\mu}_{d, st} = \frac{1}{t} \sum_{i=1}^{t} \left( \sum_{j=1}^{n/t} \frac{n/t}{1/d_{s_i+(j-1)t}} \right),
$$

(3.4)

where $s_i$ denotes the $i^{th}$ sampled node. Note that Eq. 3.4 with $t = 1$ is equivalent to Eq. 3.3.

Note that SRW-AVG-DEGREE estimator still requires the degree information at the sampled vertices; hence, it is not directly applicable under the RNA model. In Section 3.3.2, we present an estimator called RNA-AVG-DEGREE estimator that modifies SRW-AVG-DEGREE estimator to estimate the average degree by using only RN-QUERY calls.

A recent study [21] proposes an average degree estimator under a prescribed distribution and shows how to realize it using the samples collected by a modified RW. However, realizing the prescribed distribution under the RNA model is not efficient as the accuracy highly depends on the knowledge of the vertex degrees. Therefore, we do not consider to adapt the estimator into the RNA model.

Another recent study by Lu et al. [75] shows that random edge (RE) sampling and its corresponding harmonic mean estimator, which we refer as GENERALIZED estimator in this
study, can reduce the estimation variance significantly as compared to random vertex sampling. The study also shows that although RW sampling is an approximation to the random edge sampling, harmonic mean estimator under random edge sampling always outperforms the same estimator under RW sampling. Although these are important theoretical findings, independent sampling schemes such as random edge sampling and random vertex sampling are not applicable under the RNA model and we focus on how well we can make estimations under the constraints of the RNA model although it depends on the graph conductance.

**Network Size Estimation**

In this section, we present commonly used and the state of the art network size estimators. We discuss why they are not directly applicable under the RNA model.

Hansen-Hurwitz (HH) estimator is a design-unbiased estimator for any total that can be represented as a sum of local property values under unequal-probability, independent, and with-replacement sampling designs. More formally, let \( \tau_y = \sum_{u \in V} y_u \), then HH estimator for \( \tau_y \) is given by

\[
\hat{\tau}_y = \frac{1}{n} \sum_{u \in S} y_u \pi_u. \tag{3.5}
\]

We can obtain a network size estimator by setting \( y_u = 1 \) and \( \pi_u = \frac{d_u}{2M} \) for all \( u \in S \):

\[
\hat{N} = \frac{2M}{n} \sum_{u \in S} \frac{1}{d_u} = \frac{2M}{\hat{\mu}_d}. \tag{3.6}
\]

Note that this is a natural estimator as \( N = 2M/\mu_d \). One of the issues with this estimator is that \( M \) is usually not known. In Section 3.3.3, we propose an estimator for \( M, \hat{M} \), under the RNA model (see Eq. 3.11). Therefore, we can get a network size estimator, which we call SRW-HHB-NETSIZE estimator, by replacing \( M \) with \( \hat{M} \) in Eq. 3.6 and \( \hat{\mu}_d \) with \( \hat{\mu}_d^{st} \). Thus, SRW-HHB-NETSIZE estimator becomes

\[
\hat{N} = \frac{2\hat{M}}{\hat{\mu}_d^{st}}. \tag{3.7}
\]
Given that \( \hat{\mu}^{st}_d \) requires the vertex degrees (see Eq. 3.4), SRW-HHB-NETSIZE estimator is not directly applicable under the RNA model. In Section 3.3.3, we propose an estimator called RNA-HHB-NETSIZE estimator that modifies SRW-HHB-NETSIZE estimator to estimate the network size by using only RN-QUERY calls.

Another estimator that is commonly used, especially in ecology, is called mark-and-recapture (also known as capture-recapture) estimator [86]. To use mark-and-recapture estimator, we first select \( k_1 \) samples from the population uniformly at random without replacement, and then mark and release them to the population. Then, we select \( k_2 \) samples from the same population uniformly at random without replacement. Let \( c \) denote the number of marked units selected in the second sample. Then, mark-and-recapture estimator for network size, which is asymptotically unbiased, is given by \( \hat{N}_{MR} = k_1 k_2 / c \). As the estimator requires uniform node selection which is very inefficient under the RNA model (see 3.1.6), this estimator is not practical under the RNA model.

Algorithm 1 Vertex collision based network size estimator under SRW sampling (SRW-VCB-NETSIZE) by Katzir et al. [58]

1: function SRW-VCB-NETSIZE(G,n)
2: \( \Psi \leftarrow 0, \Phi \leftarrow 0, C \leftarrow 0, \delta_i = 0 \)
3: for \( i = 1 \) to \( n \)
4: \( u \leftarrow \pi(G) \) \quad \triangleright sample a vertex from the stationary distribution \( \pi(G) \)
5: \( C \leftarrow C + \delta_u \)
6: \( \Psi \leftarrow \Psi + d_u \)
7: \( \Phi \leftarrow \Phi + 1/d_u \)
8: \( \delta_u \leftarrow \delta_u + 1 \)
9: end for
10: return \( \frac{\Phi \Psi}{\Omega} \)
11: end function

In a recent study [58], Katzir et al. propose a vertex collision based estimator for network size, which we call SRW-VCB-NETSIZE, when the vertices are sampled independently from the stationary distribution of the graph \( G \), denoted by \( \pi(G) \) (see Algorithm 1). They show that the SRW-VCB-NETSIZE estimator is approximately asymptotically unbiased for the network
size. The SRW-VCB-NETSIZE estimator keeps track of the sum of degrees \( \Psi = \sum_{u \in S} d_u \), the sum of inverse degrees \( \Phi = \sum_{u \in S} \frac{1}{d_u} \), and the total number of collisions \( C \) in the sample (between line 3 and 9) and uses these values to estimate the network size. Note that we use \( \delta_{s_i} \) to keep the number of times \( s_i \) is sampled before iteration \( i \) and use this variable to compute the total number of collisions on the fly. They show that the performance of the proposed estimator under this sampling design is superior to the collision-based estimator under the sampling design that selects vertices independently and uniformly at random. Note that the SRW-VCB-NETSIZE estimator uses the samples collected independently from the stationary distribution of the underlying graph and their degrees. As both of these are not available under the RNA model, SRW-VCB-NETSIZE is not applicable under the RNA model. In Section 3.3.3 we propose RNA-MVCB-NETSIZE estimator, an adapted version of the SRW-VCB-NETSIZE estimator, that is applicable under the RNA model. RNA-MVCB-NETSIZE estimator uses vertex multiplicities of sampled vertices instead of vertex degrees and utilizes only the samples that are not highly correlated in the random walk.

In another study, Lu et al. [74] characterize the relative bias of a collision based network size estimator \( \hat{N}_c = \left( \gamma^2 + 1 \right) \binom{n}{2} \frac{1}{C} \) under SRW sampling, where \( \gamma \) is the coefficient of variation of the degrees. Using a similar approach, we can estimate the degree of a vertex using \( \hat{d}_v = \left( \frac{\omega}{2} \right) \frac{1}{C} \), where \( \omega \) and \( c \) are the number of RN-QUERY calls made for the vertex \( v \) and the number of collisions among the neighbors of vertex \( v \), respectively. However, in our estimators for both network size and average degree, we use RNA-REC-DEGREE estimator (see Algorithm 2) which directly estimates the reciprocals of the degrees and hence does not have the bias issue discussed by Lu et al. The bias in our estimators for average degree and network size are originated from the GENERALIZED and HH estimators discussed in Section 3.1.4.

In a more recent study, Hardiman et al. [47] propose a vertex neighbor collision based network size (SRW-VNCB-NETSIZE) estimator using the weighted neighbor collisions and the
safety-margin technique so that sampled vertices will be approximately uncorrelated. More specifically, the estimator uses the number of common neighbors of approximately uncorrelated node pairs. They show that the estimation performance of SRW-VNCB-NETSIZE is better than that of SRW-VCB-NETSIZE estimator. However, SRW-VNCB-NETSIZE estimator requires knowing not only the degrees of the sampled vertices but also the ids of the neighbors of the visited vertices. This is very inefficient under the RNA model. Therefore, we do not consider the adapted version of SRW-VNCB-NETSIZE estimator as a practical estimator under the RNA model.

3.1.5 Graph Traversal Methods

Graph traversal methods are alternative crawling-based graph sampling algorithms. Unlike RW, they sample the graph without replacement. BFS [24, 116, 114, 80, 4], DFS [24, 61], Forest-Fire (FF) [71], and Snowball sampling [111] are popular graph traversal methods. All these graph traversal methods have biases that are hard to characterize for general graphs. For instance, the studies by Kurant et al. [66, 67] focus on the bias of BFS in random graphs with arbitrary degree distribution. The characterization of the bias in other graph traversal algorithms is still an active research area and is beyond the scope of this study.

3.1.6 Monte-Carlo Methods

The basic idea in all Monte-Carlo (MC) methods is to sample the data from an easy-to-sample distribution (i.e., proposal distribution), and use these samples to simulate the target distribution. We briefly discuss two MC methods: Rejection Sampling (RS) [70] and Metropolis-Hastings RW (MHRW) [79]. In the context of this study, the target distribution is the uniform distribution as we compute average and the sum. In order to select one sample from the target distribution, several samples might be collected from the proposal

\[ \text{Network size can be considered as the sum over nodes with local value one.} \]
distribution and this may make the MC method inefficient. We next discuss why RS and MHRW are inefficient under the RNA model.

The rejection sampling (a.k.a. accept-reject sampling) is a popular MC method to get samples from an arbitrary distribution and can be used to get a uniform sample of vertex ids by randomly generating a vertex id and checking if RN-QUERY succeeds (returns a neighbor id). If RN-QUERY succeeds, the vertex id is included in the sample; otherwise, it is discarded. Obviously, the rejection sampling is efficient only if the vertex id space is dense which is not the case in the context of this study (see Section 3.2.1, Assumption 5).

MHRW is a Markov Chain Monte Carlo (MCMC) method that can remove the bias introduced by SRW by rearranging the one step transition probability matrix of SRW. Intuitively, for a large enough sample size, the walk visits each vertex in the graph with equal probability. More specifically, the transition probability matrix $P_{\text{MHRW}} = \{p_{uv}\}_{u,v \in V}$ of MHRW is given by

$$p_{uv} = \begin{cases} \min \left(1, \frac{d_u}{d_v} \right) & \text{if } (u, v) \in E \\ 0 & \text{if } u \neq v, (u, v) \notin E, \end{cases}$$

(3.8)

and $p_{uu} = 1 - \sum_{u \neq v} p_{uv}$. MHRW can be implemented in two-steps to select the next sample. In the first step, a candidate vertex $v$ is selected uniformly at random among the neighbors. In the second step, $v$ is accepted as a sample with probability $p_{uv}$, otherwise $u$ is accepted as a sample. The transition probabilities of MHRW depend on the degrees of both the current vertex and the proposed vertex. In the RNA model, these are not available and estimating both is not very efficient especially when the candidate vertex is rejected in the second step since all RN-QUERY calls made to estimate the degree of the candidate vertex might be wasted.
3.1.7 Other Related Work

Estimation of the structural properties of graphs has been popular in social sciences, especially in the survey sampling field, during 1970s and 1980s [29, 31, 30, 32, 33, 43, 15]. These papers have introduced the statistical methodology for estimating properties of graphs based on various simple sampling methods such as induced subgraph sampling based on random vertex selection, random edge sampling, and star sampling [15]. There are also studies that propose link-tracing sampling methods and inference techniques for hard-to-reach populations such as snowball-sampling [42] and respondent-driven sampling [50, 93, 45, 35, 36, 108]. Additionally, there are studies focusing on estimation of the properties of ego-centric networks rather than the properties of the complete networks [77, 84].

Recently, there has been an interest in the development of various random-walk based sampling methods in the sampling and estimation problems in large scale real world networks such as OSN, web, and search engines [88, 90, 109, 48, 64, 47, 38, 39, 7, 117, 11, 51, 8].

The research community has also been interested in studying various characteristics of such large scale networks including degree distribution [97, 98, 69, 88, 38, 71, 68], clustering coefficient [2, 47, 69, 88, 38], power-law exponent [24], average path length [5, 3], centrality measures [69, 5], and diameter [24, 71].

Researchers have also studied the task of how to choose an effective sampling scheme, how to evaluate different sampling schemes for studying a particular estimation problem [71, 2], and the impact of underlying graph structure and the studied graph property on the effectiveness of the sampling algorithms [16].

3.2 The RNA Model and Sampling Designs

We introduce the RNA model through the assumptions on the underlying graph and the data access limitations to the underlying graph. We also present the proposed sampling designs for the RNA model.
3.2.1 The RNA Model

The RNA model consists of the following assumptions about the underlying graph and the data access limitations:

1. The underlying graph is static.

2. The underlying graph is connected, non-bipartite, and undirected.

3. The graph is accessible only through the RN-QUERY calls. RN-QUERY returns one of the neighbors uniformly at random for a given valid vertex id \( i \). We say that RN-QUERY fails if \( i \) is not a valid vertex id.

4. We initially have access to one of the valid node ids.

5. The vertex id space is sparse, the number of valid ids is much less than the size of the id space.

The first assumption is for mathematical convenience. Providing bounds for estimators on dynamic graphs require modeling the dynamics of the network which is beyond the scope of this study [87, 100, 113]. However, we still provide a simulation experiment and discuss the factors affecting the accuracy of the estimators under dynamic networks (see Section 3.5.3). The second assumption is required for the estimators to have good asymptotic properties.\(^6\) The third assumption is the restriction on the data collection from the underlying graph. The fourth assumption is to get the first user id in the walk. The last assumption does not affect our estimation as we perform a walk on the graph. However, it makes estimation through rejection sampling impractical; otherwise one can simply generate a random id and make an RN-QUERY using that id to get an independent sample from the graph. In fact, the last assumption is very common with popular OSNs. For instance, Facebook has increased the id

\(^6\)These are the requirements of the underlying graph for the random walk to have a stationary distribution.
The $i^{th}$ RN-QUERY call is made for vertex id $u$ and it returns vertex id $v$.

![Illustration of RSRW](image1)

(a) Illustration of RSRW

![Illustration of ERSRW with $\omega = 2$](image2)

(b) Illustration of ERSRW with $\omega = 2$

Figure 3.3. Illustration of the proposed sampling designs with a sample size of 6 on a small population graph.

space of its users from 32 bit to 64 bit in 2009 and hence the id range has increased from $[0, 2^{32}]$ to $[0, 2^{64}]$. The implication of this is that most of the RN-QUERY calls fail if random ids are selected from the id range and RN-QUERY calls are made.

3.2.2 Sampling Designs

We introduce two sampling designs that are applicable under the RNA model. The adaptive version of the ERSRW sampling design is discussed in Section 3.5.2 together with the corresponding estimators.

**Restricted Simple Random Walk (RSRW)**

In a typical simple random walk (SRW) setting, it is assumed that the degrees of the visited vertices are observed. Under the RNA model, we can still perform a random walk on the underlying graph but we only observe the vertex ids of the visited vertices. We call this version of the random walk as the RSRW. Note that the restriction refers to the ability to observe only the vertex ids not to the ability to perform a random walk. Fig. 3.3a shows an
example RSRW on a small population graph. The obtained sample is \( S = \{5, 2, 1, 5, 3, 5\} \).

The challenge here is to propose an estimator using only the information in \( S \).

**Exploratory Restricted Simple Random Walk With Fixed Probing (ERSRW)**

Note that RSRW goes to a neighbor vertex after every RN-QUERY call. An alternative sampling scheme is to make several RN-QUERY calls for the current vertex before going to a neighbor vertex. The aim of making several RN-QUERY calls for the same vertex is to estimate the reciprocal of the vertex degree which is used to correct the sampling bias.

We make a constant number \( \omega \) of RN-QUERY calls for the current vertex before going to a neighbor vertex in the walk. Fig. 3.3b shows an example ERSRW on a small population graph. The obtained sample is \( S = \{5, 1, 2, 5, 4, 2\} \). Note that we make two RN-QUERY calls for vertices 4, 1, and 5 and we only go to the neighbor vertex in the \( \omega^\text{th} \) RN-QUERY call. For instance, the first RN-QUERY call returns id 1, but we do not go to that vertex and perform an RN-QUERY for that vertex since \( \omega = 1 \). In general, for a given \( \omega \) and \( n \), we basically perform an RSRW of size \( \lfloor n/\omega \rfloor \) and we sample \( \omega \) neighbors of each vertices with replacement in the walk.7

### 3.3 Estimation under the RNA Model

In this section we present our estimators that use the data collected by the sampling designs we introduced in Section 3.2.2. Table 3.1 shows how our estimators for the RNA model differs from the estimators for the SRW model. The estimators shown in the table do not use the dependency reduction techniques for the simplicity of comparison. We propose RNA-AVG-DEGREE estimator for the average degree. RNA-AVG-DEGREE estimator uses the data collected by ERSRW sampling. We propose two different estimators for the network size: 1)

\[ \text{If } n \text{ is not divisible by } \omega, \text{ we ignore the last } (n \mod \omega) \text{ RN-QUERY calls made for the last vertex in the random walk.} \]
RNA-MVCB-NETSIZE and 2) RNA-HHB-NETSIZE. RNA-MVCB-NETSIZE estimator uses the data collected by RSRW sampling while RNA-HHB-NETSIZE estimator uses the data collected by ERSRW sampling.

Table 3.1. Comparison of the estimators for SRW sampling with the adapted estimators for the RNA model. Estimators for the RNA model do not include the safety-margin dependency reduction technique to make it easier to compare with the estimators for SRW.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Estimator for SRW sampling (without dependency reduction)</th>
<th>Estimator for the RNA model (without dependency reduction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Degree</td>
<td>SRW-AVG-DEGREE ( \frac{n}{\sum_{u \in S} 1/d_u} )</td>
<td>RNA-AVG-DEGREE ( \frac{n}{\sum_{u \in S} 1/d_u} )</td>
</tr>
<tr>
<td></td>
<td>SRW-VCB-NETSIZE ( \frac{\sum_{u \in S} d_u \sum_{v \in S} 1/d_v}{C} )</td>
<td>RNA-MVCB-NETSIZE ( \frac{\Delta_u \sum_{v \in S} 1/\Delta_v}{C} )</td>
</tr>
<tr>
<td>Network Size</td>
<td>SRW-HHB-NETSIZE ( \frac{2M}{n} \sum_{u \in S} 1/d_u )</td>
<td>RNA-HHB-NETSIZE ( \frac{2M}{n} \sum_{u \in S} 1/d_u )</td>
</tr>
</tbody>
</table>

Before proceeding with the average degree and network size estimators, we first introduce an estimator for the reciprocal of the vertex degrees that uses the data collected by ERSRW sampling. We use this estimator in both RNA-AVG-DEGREE and RNA-HHB-NETSIZE estimators to correct the sampling bias due to random walk. Next, we give the details of the average degree and network size estimators.

### 3.3.1 On-the-fly Estimator for the Reciprocal of the Vertex Degree: RNA-REC-DEGREE

Given that SRW-AVG-DEGREE and SRW-HHB-NETSIZE estimators require the reciprocal of the vertex degrees which is not available under the RNA model, we need to estimate them. In this section, we propose an estimator for the reciprocal of the vertex degree based on the
sample collected by ERSRW sampling method. The main motivation is that the RNA model allows us to perform a random walk which visits vertices with higher degrees with higher probability. Therefore, the plug-in estimator gives an overestimation, but this bias can be corrected if the vertex degrees are estimated accurately. To estimate the vertex degree, we sample the neighbor ids of the queried vertex independently and uniformly at random with replacement by making several RN-QUERY calls for the same vertex. Then, we estimate the reciprocal of the vertex degree, based on the number of RN-QUERY calls and collisions. We make a constant number of RN-QUERY calls per sampled vertex and denote it by $\omega$ (see Algorithm 2). We discuss the version with adaptive number of RN-QUERY calls per sampled vertex in Section 3.5.2.

**Algorithm 2** On-the-fly estimator for the reciprocal of the degree of vertex $u$ under ERSRW with fixed probing.

1: function RNA-REC-DEGREE($G, u, \omega$)  
2: \hspace{1em} nbors $\leftarrow []$  
3: for $j = 1$ to $\omega$  
4: \hspace{1em} nbors[$j$] $\leftarrow$ RN-QUERY($G, u$)  
5: end for  
6: $c \leftarrow \text{NUMOFCOLLISIONS}(nbors)$  
7: return $\frac{c}{(\omega)^2}$  
8: end function

Although we use fixed number of RN-QUERY calls for estimating the reciprocal of the degree of each sampled vertex, one important question in this context is how many RN-QUERY calls are needed to get a given precision and confidence level. Intuitively, a vertex with high degree requires more RN-QUERY calls than a vertex with a small degree to get the same precision or confidence level. In the rest of this section, we develop the theory to establish the relation between the number of RN-QUERY calls and the degree for a given precision and confidence level. Note that an alternative approach might be to make variable number of RN-QUERY calls per vertex depending on the vertex degree. One possible approach is to make RN-QUERY calls until a fixed number of collisions occur. With
this approach, we make more RN-QUERY calls for higher degree nodes. Our preliminary experimental results shows that this approach performs worse especially when the degree distribution is very skewed and sampling fraction is small. The main reason is that too many RN-QUERY calls are made for very high degree nodes. As a future work, we plan to further investigate under what circumstances this approach outperforms the fixed RN-QUERY per vertex approach.

![Figure 3.4. The accuracy of binomial approximation to the number of collisions $c_u$ for various $\omega$ and $d_u$ values. Each plot shows the normalized frequency of $c_u$ over 100,000 simulations (shown as filled-green bars) and the probability mass function (PMF) of $\text{Binom}(\binom{\omega}{2}, 1/d_u)$ (shown as black step lines).](image)

We make the analysis for an arbitrary vertex $u$ without loss of generality. We use $c_u$ to denote the number of collisions in $\omega$ RN-QUERY calls for vertex $u$. We use the idea that $c_u$ follows approximately $\text{Binom}(\binom{\omega}{2}, 1/d_u)$ as each ordered pair of sampled vertices have collision probability of $1/d_u$. $c_u$ is approximately binomial as the collisions of ordered pair of sampled vertices are not independent. Based on this approximation, we provide an estimator for $1/d_u$ and a lower bound on $\omega$ for a given accuracy and precision in Theorem 1. As the quality of the approximation in Theorem 1 depends on how well $c_u$ is approximated by $\text{Binom}(\binom{\omega}{2}, 1/d_u)$, we provide empirical results for various values of $\omega$ and $d_u$ in Fig. 3.4.
Table 3.2. Comparison of the descriptive statistics of $\text{Binomial}(\binom{\omega}{2}, 1/d_u)$ and the number of collisions $c_u$ for various $\omega = \{10, 50, 100\}$ and $d_u = \{2, 5, 10, 50, 100\}$.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$d_u = 2$</th>
<th>$\omega = 50$</th>
<th>$\omega = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>var</td>
<td>skew</td>
</tr>
<tr>
<td>Binomial</td>
<td>22.500</td>
<td>11.250</td>
<td>0</td>
</tr>
<tr>
<td>$c_u$</td>
<td>22.529</td>
<td>11.434</td>
<td>2.367</td>
</tr>
<tr>
<td>Binomial</td>
<td>9.000</td>
<td>7.200</td>
<td>0.224</td>
</tr>
<tr>
<td>$c_u$</td>
<td>9.010</td>
<td>7.251</td>
<td>1.423</td>
</tr>
<tr>
<td>Binomial</td>
<td>4.500</td>
<td>4.050</td>
<td>0.398</td>
</tr>
<tr>
<td>$c_u$</td>
<td>4.504</td>
<td>4.064</td>
<td>1.219</td>
</tr>
<tr>
<td>Binomial</td>
<td>0.900</td>
<td>0.882</td>
<td>1.022</td>
</tr>
<tr>
<td>$c_u$</td>
<td>0.897</td>
<td>0.875</td>
<td>0.377</td>
</tr>
<tr>
<td>Binomial</td>
<td>0.450</td>
<td>0.445</td>
<td>1.468</td>
</tr>
<tr>
<td>$c_u$</td>
<td>0.446</td>
<td>0.436</td>
<td>1.645</td>
</tr>
</tbody>
</table>

The figure shows that binomial approximation is very good especially when both $\omega$ and $d_u$ are greater than 10. Another observation is that although the approximations are not perfect for small $d_u$ (e.g. $d_u = 2$), the mean and the variance of $c_u$ is still well approximated by $\text{Binom}(\binom{\omega}{2}, 1/d_u)$ (see Table 3.2). The main difference between them is the skewness. This helps us to use the variance of the binomial distribution as the approximate variance of the number of collisions and obtain approximate confidence intervals.

Theorem 1. The estimator $\hat{d}_u^{rec} = \frac{c_u}{\binom{\omega}{2}}$ for $1/d_u$ is approximately unbiased and has mean square error $MSE(\hat{d}_u^{rec}) \approx (\binom{\omega}{2})^{-1} \frac{1}{d_u} \left(1 - \frac{1}{d_u}\right)$ and it guarantees for any $0 \leq \epsilon \leq 1$ and $0 \leq \alpha \leq 1$:

$$P \left[ \left| \frac{\hat{d}_u^{rec} - 1/d_u}{1/d_u} \right| \leq \epsilon \right] \geq 1 - \alpha$$

when $\omega$ satisfies
\[ \omega \geq \frac{z_{\alpha} \sqrt{2(d_u - 1)}}{\epsilon}, \]

where \( z_{\alpha} \) denotes the \( 100(1 - \alpha) \)th percentile of a standard normal distribution.

We first show the following two lemmas which are used to prove the Theorem 1:

**Lemma 2.** The estimator \( \hat{d}_{u}^{\text{rec}} \) is approximately unbiased for \( 1/d_u \), i.e., \( \text{Bias}(\hat{d}_{u}^{\text{rec}}) \approx 0 \).

**Proof.** \( c_u \) follows approximately a binomial distribution and \( \binom{\omega}{2} \) is a constant; therefore, \( \mathbb{E}[d_{u}^{\text{rec}}] \) can be computed as:

\[
\mathbb{E}[\hat{d}_{u}^{\text{rec}}] = \mathbb{E} \left[ c_u \binom{\omega}{2}^{-1} \right] = \binom{\omega}{2}^{-1} \mathbb{E}[c_u] \approx \binom{\omega}{2}^{-1} \binom{\omega}{2} \frac{1}{d_u} = \frac{1}{d_u}
\]  

(3.9)

Therefore, \( \text{Bias}(\hat{d}_{u}^{\text{rec}}) = \mathbb{E}[\hat{d}_{u}^{\text{rec}} - 1/d_u] \approx 0 \). \( \square \)

**Lemma 3.** The approximate variance of the estimator \( \hat{d}_{u}^{\text{rec}} \), \( \text{Var}(\hat{d}_{u}^{\text{rec}}) \), is \( \binom{\omega}{2}^{-1} \frac{1}{d_u} \left( 1 - \frac{1}{d_u} \right) \).

**Proof.** The result simply follows from the fact that \( \omega \) is a constant and \( c_u \) follows approximately \( \text{Binom}(\binom{\omega}{2}, 1/d_u) \). \( \square \)

**Proof of Theorem 1.** In Lemma 2, we show that \( \hat{d}_{u}^{\text{rec}} \) is approximately unbiased for \( 1/d_u \). Then, in Lemma 3, we obtain the variance of \( \hat{d}_{u}^{\text{rec}} \) which is approximately \( \text{MSE}(\hat{d}_{u}^{\text{rec}}) \) as \( \text{MSE}(\hat{d}_{u}^{\text{rec}}) = \text{Var}(\hat{d}_{u}^{\text{rec}}) + \text{Bias}^2(\hat{d}_{u}^{\text{rec}}) \) and \( \text{Bias}^2(\hat{d}_{u}^{\text{rec}}) \approx 0 \). In order to show the lower bound for \( \omega \), we use the normal distribution approximation to the binomial distribution. We use \( \text{Normal}(\text{mean}, \text{variance}) \) to denote the normal distribution with a given mean and variance.

As a rule of thumb, a binomial distribution \( \text{Binom}(n, p) \) is well-approximated by a normal distribution \( \text{Normal}(np, np(1 - p)) \) when \( np > 5 \) and \( n(1 - p) > 5 \). Therefore, \( d_{u}^{\text{rec}} \) can be approximated by the \( \text{Normal} \left( 1/d_u, \binom{\omega}{2}^{-1} \frac{1}{d_u} \left( 1 - \frac{1}{d_u} \right) \right) \) when \( \binom{\omega}{2}/d_u > 5 \) and \( \binom{\omega}{2}(1 - \frac{1}{d_u}) > 5 \), or approximately when \( \omega > 2\sqrt{10d_u} \).\(^8\) Note that for reasonable \( \epsilon (<0.1) \) and \( z_{\alpha}^2 (>1.64) \)

\(^8\)\(1/d_u \leq (1 - 1/d_u) \) for \( d_u \geq 2 \). Therefore, the bound is obtained using \( \binom{\omega}{2}/d_u > 5 \). We handle \( d_u = 1 \) case separately. We estimate \( 1/d_u \) as 1 when all RN-QUERY calls return the same node.
values, the lower bound for $\omega$ in Theorem 1 is tighter. Therefore, normality assumption is safe to make. Next, by combining the normal approximation and setting the desired margin of error to $\frac{\epsilon}{d_u}$, we get the lower bound on $\omega$ as:

$$\omega \geq 1 + \sqrt{1 + \frac{8z_{\alpha/2}^2(d_u-1)}{\epsilon^2}} \approx \frac{\sqrt{8z_{\alpha/2}^2(d_u-1)}}{2} = \frac{z_{\alpha/2}}{\epsilon}\sqrt{2(d_u-1)}.$$

(3.10)

Figure 3.5. Relation between number of RN-QUERY calls and degree for a given confidence level and relative error. (a) Relative error vs. the minimum $\omega$ needed for 90% confidence level (b) Confidence level vs. the minimum $\omega$ needed for the relative error $\epsilon = 0.1$.

Fig. 3.5a and 3.5b illustrate the relation between degree, precision, and confidence level obtained in Eq. 3.10. Fig. 3.5a illustrates the required number of RN-QUERY calls to get an estimate within a desired relative error with 90% confidence level. Fig. 3.5b illustrates the required number of API calls to get an estimate with a desired confidence level within $\epsilon = 0.1$ relative error. Fig. 3.6 illustrates the mean and coefficient of variance of 10,000 independent degree estimations using various number of RN-QUERY calls for different vertex degrees and the normalized root mean square error (NRMSE) where the normalization is over the true parameter, i.e., $NRMSE(\hat{\theta}) = E[(\hat{\theta} - \theta)]^{1/2}/\theta$. As the vertex degree increases, negative bias is observed for small $\omega$. This observation can be explained by the high likelihood of
Figure 3.6. Normalized estimates for various number of RN-QUERY calls and vertex degrees (left). Normalized root mean square errors of the estimations for various vertex degrees (right).

no collision when $\omega$ is much less than the population degree of a vertex. When there is no collision, the estimator becomes 0. Fig. 3.6 also shows the NRMSE of 10,000 estimations for various degrees with varying $\omega$. As expected, we get a higher NRMSE for the estimation of a higher degree with the same $\omega$.

Algorithm 3 Average degree estimator for the RNA model.

3.3.2 Average Degree Estimator : RNA-AVG-DEGREE

In Section 3.1.4, we discussed SRW-AVG-DEGREE, a GENERALIZED estimator with shifted-thinning. Although SRW-AVG-DEGREE is a well-known estimator under SRW sampling, it
can not be directly used to estimate the average degree as the degrees of sampled vertices are
not observable under the RNA model. In this section, we introduce RNA-AVG-DEGREE
estimator that is a simple adaptation of SRW-AVG-DEGREE estimator to the RNA model by
estimating the reciprocal of the vertex degrees using RNA-REC-DEGREE. The pseudocode
of the RNA-AVG-DEGREE is shown in Algorithm 3. In line 2, we first determine \( n_\omega \), the
number of nodes visited by the walk. The algorithm can also be implemented in such a
way that the it stops when the sampling budget \( n \) is reached. In line 3, we initiate the list
recTotal which keeps the sum of the reciprocal of the vertex degree estimates for each \( t \)
thinned walk. Recall that shifted-thinning with parameter \( t \) creates \( t \) thinned walks, where
thinned walk \( i \in [1, t] \) consists of vertices in \( \{s_{i+0t}, s_{i+1t}, \ldots \} \). Then, between lines 5-9, we
perform ERSRW and compute the sum of the estimates of reciprocal of the vertex degrees
for each thinned walk. Note that we use line 9 for ease of presentation. It can be eliminated
by using one of the RN-QUERY calls in RNA-REC-DEGREE at each iteration. In our
implementation, we use the last RN-QUERY calls in RNA-REC-DEGREE to determine next
step in the walk. Finally, on line 10, the average of the \( t \) estimates based on \( t \) thinned walks
is computed.

Under a fixed sampling budget, there is a tradeoff between the estimation performance of
vertex degrees and number of vertices visited by the walk as the number of vertices in the
walk will be approximately \( n/\omega \). The larger the \( \omega \) is, the better the vertex degree estimations
are, but the smaller the length of the walk. We explore this tradeoff using simulations in
Section 3.4.3.

### 3.3.3 Network Size Estimators

In this section we propose two different estimators for the network size: 1) Multiplicity and
Vertex Collision Based Network Size (RNA-MVCB-NETSIZE) Estimator and 2) Hansen-
RNA-MVCB-NETSIZE Estimator

In this section, we propose a network size estimator under the RNA model by adapting the SRW-VCB-NETSIZE estimator using the well-known Ergodic Theorem in the Markov chain theory (see A). The estimator has two requirements: (a) vertices need to be sampled from the stationary distribution of the underlying graph independently and (b) vertex degrees need to be known or estimated up to a constant. We use a stationary RSRW and apply the safety-margin dependency reduction technique proposed in [63] to meet the first requirement. We estimate the vertex degrees up to a constant using the vertex multiplicities based on the Ergodic Theorem to meet the second requirement.

Figure 3.7. Relative error and coefficient of variation of number of visits to four vertices with different degrees in CONDMAT graph. Each point is calculated based on 1,000 independent simulations.

Theorem 9 simply states that the multiplicity of a vertex becomes proportional to its degree in the limit of a large sample size. Although the multiplicity of a vertex (Δ) is not guaranteed to be proportional to the degree of that vertex for small sample sizes, it is still a reasonable characteristic to consider under the RSRW which a very limited sampling design. Therefore, we use the multiplicity of a vertex in Algorithm 1 in place of the vertex degree. In order to illustrate how Δ(i, t) converges to its theoretical value for finite sample sizes, we conduct an experiment on CONDMAT real world graph (see Table 3.3 for the details of the
The results are shown in Fig. 3.7. We select four nodes with different degrees. We perform 1,000 independent random walks of length 400,000. For each walk, we keep track of how many times each of these four vertices is visited after every 5,000 steps. Finally, we find the relative error with respect to the right hand side of Eq. A.1 in the Ergodic Theorem and plot the average of the 1,000 relative errors. We observe that the relative error and the coefficient of variation is small for vertices with larger degrees even for the initial steps of the RW. We want to emphasize that the convergence results in Fig. 3.7 are specific to CONDMAT graph. In general, the convergence of a random walk is a well-studied topic and it has been shown that it depends on the “spectral gap” of the underlying graph which is defined as $1 - \lambda_2$ where $\lambda_2$ is the second largest eigenvalue of the transition matrix of the underlying graph[73].

**Algorithm 4** Multiplicity and vertex-collision based network size estimator for the RNA model.

```plaintext
1: function RNA-MVCB-NETSIZE(G, init, n, margin)
2:     Φ ← 0, C ← 0, curr ← init, s ← []
3:     for i = 1 to n
4:         s[i] ← curr
5:         if m[curr] does not exist
6:             m[curr] ← 1
7:         else
8:             m[curr] ← m[curr] + 1
9:         end if
10:     curr ← RN-QUERY(G, curr)
11: end for
12: for i = 1 to n
13:     for j = 1 to n
14:         if |i - j| > margin
15:             Φ ← Φ + m[s[i]]/m[s[j]]
16:         end if
17:     end for
18: end for
19: return Φ
20: end function
```
The pseudocode of the network size estimator using vertex multiplicities is given in Algorithm 4. Between lines 3-10, the multiplicity of sampled nodes are computed. Between lines 12-21, the estimator is computed using the multiplicities of the sampled nodes. On line 14, the algorithm checks whether the samples are apart from each other in the walk for the dependency reduction. If it is true, than the RNA-MVCB-NETSIZE estimator as shown in Table 3.1 is computed. Recall that the estimators in Table 3.1 do not include the dependency reduction techniques for the sake of clear comparison. Although the proposed algorithm has \( O(n^2) \) running time complexity, we present this version due to its readability. The complexity can easily be reduced to \( O(n) \) by using a similar idea as in [63].

**RNA-HHB-NETSIZE Estimator**

In this section, we propose another estimator named RNA-HHB-NETSIZE for the network size. The basic idea is as follows: we have proposed an estimator for the average degree \( \mu_d \) under ERSRW sampling in Section 3.3.2. If we can develop an estimator for the number of edges \( M \) under the same sampling scheme, then the network size \( N = 2M/\mu_d \) can be estimated.

At first, estimating \( M \) seems to be as hard as estimating \( N \). However, obtaining edges uniformly at random is easier and more efficient than obtaining a vertex uniformly at random under ERSRW sampling. Note that we make RN-QUERY calls for two reasons under ERSRW sampling. First, we make \( \omega \) RN-QUERY calls for the same vertex to estimate the reciprocal of the vertex degree. Second, we use the last RN-QUERY call to go to a random neighbor, i.e., to perform a walk on the graph. If we consider the RN-QUERY calls made for the walk and take the current vertex and randomly selected neighbor vertex tuples as a sample edge at each step, we can sample the edges uniformly in the long run. Then, the challenge is to estimate the number of edges when we can sample the edges uniformly. This is a very similar problem to network size estimation, but the sampled objects are edges instead of nodes and
Algorithm 5 Hansen-Hurwitz-based network size estimator for the RNA model.

1: function RNA-HHB-NETSIZE(G, init, n, ω, margin, t)  
2:     \( n_ω \leftarrow \lfloor n / ω \rfloor \)  
3:     denom \leftarrow 0  
4:     curr \leftarrow init, prev \leftarrow null  
5:     e \leftarrow [] \quad \triangleright e \text{ keeps the unordered sampled edges}  
6:     c \leftarrow [] \quad \triangleright c \text{ keeps the edge multiplicities}  
7:     for \( i = 1 \) to \( n_ω \)  
8:         if prev is not null  
9:             e[i - 1] \leftarrow (prev, curr)  
10:     end if  
11:     if \( i - (margin + 1) \geq 0 \)  
12:         c[e[i - 1]] \leftarrow c[e[i - 1]] + 1  
13:     end if  
14:     prev \leftarrow curr  
15:     curr \leftarrow RN-QUERY(G, curr)  
16:     end for  
17:     for \( i = 1 \) to \( n_ω - 1 \)  
18:         if \( i - (margin + 1) \geq 1 \)  
19:             c[e[i - (margin + 1)]] \leftarrow c[e[i - (margin + 1)]] + 1  
20:     end if  
21:     if \( i + margin + 1 \leq n_ω \)  
22:         c[e[i + margin]] \leftarrow c[e[i + margin]] - 1  
23:     end if  
24:     denom \leftarrow denom + c[e[i]]  
25:     end for  
26:     return \( \frac{2(n_ω - margin)(n_ω - margin - 1)}{RNA-AVG-DEGREE(G, init, n, ω, t) \cdot denom} \)  
27: end function

they are selected with equal probability. Using Katzir’s estimator with weights equal to 1 (hence both \( Φ \) and \( Ψ \) becomes \( n_ω \)), we get the following estimator for \( \hat{M} \): \( \hat{M} = \frac{n_ω^2}{2C_E} \), where \( C_E \) is the number of collisions in the sampled edges. Then, by applying the safety-margin technique, we get

\[
\hat{M} = \frac{\sum_{1 \leq i,j \leq n_ω} 1_{\{j-i\geq margin\}}}{\sum_{1 \leq i,j \leq n_ω} 1_{\{s_{e_i} = s_{e_j}\}1_{\{j-i\geq margin\}}}} = \frac{(n_ω - margin)(n_ω - margin - 1)}{\sum_{1 \leq i,j \leq n_ω} 1_{\{s_{e_i} = s_{e_j}\}1_{\{j-i\geq margin\}}}}, \quad (3.11)
\]

where \( s_{e_i} \) denotes the \( i^{th} \) sampled edge. Finally, dividing \( 2\hat{M} \) by RNA-AVG-DEGREE estimator proposed in Section 3.3.2, we get the estimator for \( N \). The pseudocode of the efficient implementation is given in Algorithm 5. In lines 8-10, sampled unordered edges are
stored in list $e$. In lines 11-13, multiplicities of sampled edges are computed. $c[e[i]]$ denotes the multiplicity of $i^{th}$ sampled edge. In lines 17-25, twice the total number of edge collisions that are at least margin step away from each other, i.e., the denominator of Eq. 3.11, is computed and stored in denom. Finally, in line 26, estimation for the average degree and the number of edges is combined to get the estimation for the network size. For ease of presentation, we compute the RNA-AVG-DEGREE estimator in line 26, but it can also be computed in the first for loop of the algorithm (between lines 7-16).

3.4 Experimental Results

In this section we evaluate the performance of the proposed estimators. In general, comparing two estimators under a different data access model may not be fair as one of the models may allow more access to the data for a given estimation task and this may make the estimation problem easier. For this reason, we do not compare the performance of our estimators with the estimators developed under a different sampling design such as independent vertex/edge sampling or induced subgraph sampling. Instead, we compare the proposed estimators with the corresponding SRW based estimators in which vertex degrees are observed (see Table 3.1). Comparing the performance of the SRW and RNA based estimators enables us to observe the two phenomena: 1) the performance of the estimators with the true degree information of the sampled vertices and 2) the performance drop due to the estimation of the vertex degrees in the RNA based estimators.

3.4.1 Dataset

We evaluate the performance of our estimators on both real-world networks from various application domains and synthetic networks.

---

9Edge $(u,v)$ and edge $(v,u)$ is considered to be the same.

10Collision on indices $(i,j)$ and $(j,i)$ are counted separately.
Table 3.3. Details of real-world networks used in the experiments. All except Facebook (New-Orleans) network was obtained from [72].

<table>
<thead>
<tr>
<th>Network Name</th>
<th># of nodes in LCC</th>
<th># of edges in LCC</th>
<th>avg. degree</th>
<th>degree assortativity coefficient (r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNUTELLA</td>
<td>62,561 (1.000*)</td>
<td>147,878 (1.000*)</td>
<td>4.727</td>
<td>-0.09</td>
</tr>
<tr>
<td>CONDMAT</td>
<td>21,363 (0.923)</td>
<td>91,342 (0.977)</td>
<td>8.551</td>
<td>0.13</td>
</tr>
<tr>
<td>EPINIONS</td>
<td>75,877 (1.000)</td>
<td>508,836 (1.000)</td>
<td>10.695</td>
<td>-0.04</td>
</tr>
<tr>
<td>ENRON</td>
<td>33,696 (0.918)</td>
<td>180,811 (0.984)</td>
<td>10.732</td>
<td>-0.12</td>
</tr>
<tr>
<td>FACEBOOK [107]</td>
<td>63,392 (0.995)</td>
<td>816,886 (1.000)</td>
<td>25.773</td>
<td>0.18</td>
</tr>
</tbody>
</table>

* the fraction of the number of nodes in the largest connected component (LCC)
** the fraction of the number of edges in the LCC

Real-World Networks: The real-world network datasets that we use in the experiments are shown in Table 3.3. We extract and use only the largest connected component of each graph. The description of the real world-graphs are as follows:

- **GNUTELLA**: Gnutella peer-to-peer file sharing network from August 2002. Vertices are hosts in Gnutella network and there exists an edge between two vertices if they have connections.


- **EPINIONS**: Who-trusts-whom network of users of Epinions.com. Although the trust relationship is directed in the original data set, we convert it into an undirected graph where there exists an edge between two vertices if there exists at least one directed edge between them for the sake of analysis.

- **ENRON**: Enron email communication network. Vertices are the email addresses and there exists an undirected edge between two vertices if there is at least one email communication between them.
• FACEBOOK: New Orleans Regional Facebook friendship network collected between December 29, 2008 and January 3, 2009 in a breadth-first-search fashion.

**Synthetic Networks:** We use two synthetic graph generation models: 1) Erdős-Renyi (ER) and 2) Barabási-Albert (BA). In the experiments, we generate synthetic networks of size 10,000 with average degrees 4, 10, and 100.

ER graph generation model creates purely random networks (we shortly refer to them as ER graphs) and has two general forms. In one, first the number of edges and vertices are fixed, then these edges connect randomly selected vertices. In the other, which we use in this study, an edge is created between each pair of vertices with probability \( p \). One can create ER graphs with size \( N \) and average degree \( \mu_d \) by setting the \( p \) value to \( \frac{\mu_d}{N-1} \) as the expected number of edges in the generated network is \( \binom{N}{2} p \).

BA graph generation model creates scale-free networks (we shortly refer to them as BA graphs) and the model incorporates two important concepts: 1) preferential attachment and 2) growth. Growth means that the model creates a network with increasing size over time. Preferential attachment means that the vertices with higher degree have more chance to connect to the new nodes. These two concepts are very common in real-world networks. One can create BA graphs with size \( N \) and average degree \( \mu_d \) by setting the parameter for the number of edges attached to the newly joining node to \( \frac{\mu_d}{2} \). The reason is that the number of edges in the network converges to \( N\mu_d/2 \) as \( N \to \infty \) and hence \( \mu_d \) is obtained as the approximate average degree.

Both ER and BA graph generation models generate non-assortative graphs in the limit of large graph size. In other words, the degree of a node is not correlated with the degrees of its neighbors. This is an important feature as we perform a random walk on the underlying graph, we know that the vertex degrees of sampled nodes are not correlated. Therefore, the effect of correlation in the degrees of sampled nodes disappear on the estimation performance. This helps us eliminate one possible side effect on the estimation performance as we generate graphs with different parameters.
3.4.2 Evaluation Metric

We evaluate the performance of the proposed estimators using simulations as follows: We collect 1,000 independent samples of a certain size. Then, we use the relevant estimator to create 1,000 independent estimates of a certain property. Then, we compute the 5th and 95th percentile of these 1,000 estimations. The region between these two intervals is known as 90% Monte Carlo confidence interval. We also compute the 50th percentile (median) of these 1,000 estimations. Then, we normalize these percentile values with the true parameter to illustrate how well the estimations are concentrated around the true parameter. We repeat this experiment for various sampling fractions. For the average degree, we use the sampling fraction range of [0.01,1]. For the network size, we use the sampling fraction range of [0.01,50]. If the region between 5th and 95th percentiles concentrates around 1, the estimation performance gets better.

3.4.3 Average Degree Estimation

We compare the RNA-AVG-DEGREE estimator with the SRW-AVG-DEGREE estimator. In the RNA model, it is not possible to get the vertex degree with 100% certainty. Therefore, we need to determine a fair cost function for obtaining the actual vertex degree in the SRW-AVG-DEGREE estimator. One approach could be that a single RN-QUERY call returns the actual degree of a vertex as well as its id. However, this assumption will trivially make SRW-AVG-DEGREE estimator better because we use $\omega$ RN-QUERY calls in the RNA model in order to estimate the vertex degree. Instead, we use the same walk length (of size $\lfloor n/\omega \rfloor$) for both estimators. This corresponds to getting the actual vertex degree with a sampling cost of $\omega$ in SRW-AVG-DEGREE estimator. With this approach, we can observe the effect of degree estimations on the performance of the average degree estimation by the difference of SRW-AVG-DEGREE and RNA-AVG-DEGREE confidence intervals.

The summary of results for the average degree estimation is as follows
• The estimation precision depends on the degree structure of the underlying graph (e.g., average degree, degree variance, degree skewness), the value of $\omega$, and the sampling fraction.

• A good selection of $\omega$ depends on the degree structure of the underlying graph.

• Applying shifted-thinning dependency technique to the walk decreases estimation performance.

Real-world graphs: The results for real-networks with no thinning, $t = 1$, is shown in Fig. 3.8. We first note that the confidence intervals for SRW-AVG-DEGREE estimator get wider as $\omega$ increases. This is because the length of SRW $n/\omega$ decreases and the shorter walk length causes less precise estimations. We also observe that the confidence intervals for SRW-AVG-DEGREE and RNA-AVG-DEGREE overlaps. This is probably because $\omega = 10$ is large enough for the degree estimations which can be validated by the small true average degrees of the population graphs shown in Table 3.3. As we will show in the synthetic graph cases, when the true average degree is high (e.g., 100), RNA-AVG-DEGREE performs worse than SRW-AVG-DEGREE for $\omega = 10$.

Another observation is that the estimation in GNUTELLA graph is much better than all others for all $\omega$ that we consider. An important reason might be that GNUTELLA graph has a lower average degree, degree variance, and skewness (see Fig. 3.9). An interesting observation is that a small downward bias is observed when $\omega = 10$ and $f > 0.8$, especially in ENRON and EPINIONS graphs. This might be due to high variance and skewness in the degree distribution. Approximately 85% of the vertices have degrees less than the true average degree in ENRON and EPINIONS graphs; while approximately 70% of the vertices have degrees less than the true average degree in the other real-world graphs. Nevertheless, we should not observe this downward bias if we repeat the experiments infinitely many times. The same phenomena occurs when estimating the mean of a highly skewed distribution with
values are shown on each row: normally distributed sampling distribution. We also observe that high degree variance with a
Furthermore, the more skewed the distribution is, the larger sample size we need to get a
for the variance and the skewness of the degree distribution, respectively.

Figure 3.9. Degree distributions and statistics of the real-world graphs. var and skew stand
ranging from 1% to 100% are shown on each plot. Estimation results with three different ω values are shown on each row: ω = [10, 50, 100].

Figure 3.8. Average degree estimation (real-world graphs) with no thinning, t=1: 5th percentile, median, and 95th percentile of 1,000 estimations are shown for both SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators. Estimation results for the sampling fractions ranging from 1% to 100% are shown on each plot. Estimation results with three different ω values are shown on each row: ω = [10, 50, 100].

i.i.d. samples. If the sample size is not large enough, the sampling distribution is also skewed.
Furthermore, the more skewed the distribution is, the larger sample size we need to get a
normally distributed sampling distribution. We also observe that high degree variance with a
small walk length causes a very wide confidence intervals for small $f < 0.1$. For instance, see ENRON, EPINIONS, and FACEBOOK graphs with $\omega = 100$.

![Figure 3.10. Average degree estimation (real-world graphs) with thinning, $t=5$: $5^{th}$ percentile, median, and $95^{th}$ percentile of 1,000 estimations are shown for both SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators. Estimation results for the sampling fractions ranging from 1% to 100% are shown on each plot. Estimation results with three different $\omega$ values are shown on each row: $\omega = [10, 50, 100]$. We also conduct average degree estimations with thinning parameter $t = 5$ for the RNA-AVG-DEGREE and keep $t = 1$ for SRW-AVG-DEGREE to see whether decreasing the dependency of samples at the expense of the length of the walk improves estimation performance (see Fig. 3.10). Recall that shifted-thinning with the walk size $n$ and the thinning parameter $t$ makes $t$ estimations based on a sample size of $n/t$. Experimental results show that the confidence intervals for RNA-AVG-DEGREE estimator, especially for small sample sizes, are wider. This means that applying the dependency reduction decreases the performance of the RNA-AVG-DEGREE.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Sampling fraction ($f = n/N$)</th>
<th>Avg degree estimate</th>
<th>Avg degree estimate</th>
<th>Avg degree estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>gnutella</td>
<td>0.0 0.2 0.4 0.6 0.8 1.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
</tr>
<tr>
<td>condMat</td>
<td>0.0 0.2 0.4 0.6 0.8 1.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
</tr>
<tr>
<td>epinions</td>
<td>0.0 0.2 0.4 0.6 0.8 1.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
</tr>
<tr>
<td>enron</td>
<td>0.0 0.2 0.4 0.6 0.8 1.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
</tr>
<tr>
<td>facebook-neworleans</td>
<td>0.0 0.2 0.4 0.6 0.8 1.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
<td>1.0 1.5 2.0 2.5 3.0</td>
</tr>
</tbody>
</table>

SRW-AVG-DEGREE | RNA-AVG-DEGREE
**Synthetic graphs:** To better illustrate the effects of the true average degree, degree variance, and skewness on the estimation performance, we conduct experiments on synthetic graphs with different average degree, degree variance, and skewness. Estimation results and the degree statistics of the synthetic graphs are shown in Fig. 3.11 and Fig. 3.12, respectively. The graphs have approximately zero degree assortativity coefficient (|r| < 0.04 for all synthetic graphs).

The estimation results are shown in Fig. 3.11. We observe that when $\omega = 10$ and $\mu_d = 100$, estimations between SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators differ widely in both ER and BA graphs. This result shows that too small $\omega$ in RNA-AVG-DEGREE is not good enough to estimate the reciprocals of the vertex degrees. Recall that the difference between RNA-AVG-DEGREE and SRW-AVG-DEGREE estimators is that RNA-AVG-DEGREE estimator estimates the reciprocals of the vertex degrees while SRW-AVG-DEGREE estimator uses the actual vertex degrees. We also observe that the good selection of $\omega$ is related to the population average degree. For ER graphs with $\mu_d = 4$ and $\mu_d = 10$, the best estimation is obtained when $\omega = 10$ and increasing $\omega$ decreases the precision for all sampling fractions. However, for the ER graph with $\mu_d = 100$, the best estimation is obtained when $\omega = 100$ and decreasing $\omega$ decreases the precision for all sampling fractions. For the BA graphs with $\mu_d = 4$ and $\mu_d = 10$, we still observe that $\omega = 10$ is the best option among all three options. However, for the BA graph with $\mu_d = 100$, the estimation results are almost equal for $f > 0.1$. For $f < 0.1$, the best estimation is obtained when $\omega = 50$ as the walk length is short and approximately 50% of the nodes have degrees between 50 and 70 (see Fig. 3.12, $BA(\mu_d = 100)$). This result shows that the optimal selection of $\omega$ not only depends on the average degree but also depends on the degree structure of the underlying graph.

We also conduct experiments on synthetic graphs to see the effect of $\omega$ on the NRMSE of the RNA-AVG-DEGREE estimator (see Fig. 3.13). We choose $\omega$ in the range [2,30] and $f$ in
We see that optimal estimation results with three different $\omega$ values are shown on each row: $\omega = [10, 50, 100]$.

Figure 3.12. Degree distributions and degree statistics of the synthetic graphs generated by Erdős-Rényi (ER) and Barabási-Albert (BA) models. The variance and the skewness of the degree distribution are abbreviated as var and skew, respectively.

the range $[0.01, 3]$. We perform 1,000 simulations for each sampling fraction and $\omega$. Then, we plot the NRMSE of the estimator. We also plot the average of the best $\omega$ values over 1,000 simulations for each sampling fraction. We see that optimal $\omega$ values highly depend on the degree structure of the underlying graph. We also observe that the optimal value of $\omega$ increases much faster in BA graphs which have higher degree variance and skewness. In ER
Figure 3.13. NRMSE of RNA-AVG-DEGREE for various sampling fractions $0.01 < f < 3$ and $\omega = [2, 30]$. The average of 1,000 independent simulations are shown. The black lines show the average of the best $\omega$ values over 1,000 simulations for each sampling fraction. The best $\omega$ is the one that gives the minimum NRMSE among all $\omega$ values that we use for a given sampling fraction.

graph, the optimal value of the $\omega$ is close to the true average degree and increases slightly for the large sample sizes.

**A guide on the selection of $\omega$:** The experimental results show that the selection of $\omega$ has big impact on the estimation performance. We provide a rough guidance on how to select $\omega$.

The first important factor in the selection of $\omega$ is the true average degree of the underlying population network. If the underlying population network has a low average degree, $\omega$ should not be too large. Otherwise, we may spend too much sampling budget to estimate individual vertex degrees but we may not visit enough nodes via random walk. For instance, we need to almost double $\omega$ to reduce the relative error of the degree estimation from 0.2 to 0.1 at the 90% confidence level (see Fig. 3.5-a). The degree variance of the underlying graph is also an important factor in the selection of $\omega$. To clarify, suppose that the underlying graph is known to have a degree distribution close to uniform distribution. In this case, spending more RN-QUERY calls on estimating individual vertex degrees rather than visiting more nodes via RW would be a good decision. On the other hand, suppose that the underlying graph has a degree distribution with high variance. In this case, estimating individual vertex degrees accurately is not enough to make accurate estimations, the walk should also visit more nodes to explore the underlying graph.
3.4.4 Network Size Estimation

In this section, we evaluate the performance of the network size estimators. We compare the proposed network size estimators with their corresponding SRW-based estimators. More specifically, we compare SRW-HHB-NETSIZE estimator with RNA-HHB-NETSIZE estimator by keeping the random walk length the same (similar to the comparison of SRW-AVG-DEGREE and RNA-AVG-DEGREE estimators). We also compare SRW-VCB-NETSIZE estimator with RNA-MVCB-NETSIZE estimator by keeping the number of RN-QUERY calls the same for both estimators since there is no \( \omega \) in RNA-MVCB-NETSIZE estimator. We apply the safety-margin dependency reduction technique in the estimators. We consider three different margin parameters \{1, 10, 100\}.

The summary of results for the average degree estimation is as follows:

- RNA-HHB-NETSIZE estimator with a good selection of \( \omega \) outperforms RNA-MVCB-NETSIZE estimator for \( f < 10 \) in real-world graphs.
- RNA-MVCB-NETSIZE estimator is very precise but requires high sampling fraction \( (f > 10) \) for accurate estimations. Interestingly, for the synthetic graphs, \( f = 2 \) seems large enough for both accurate and precise estimations.
- Applying the safety-margin dependency technique generally improves the accuracy although the precision decreases due to ignoring some sample data, especially for \( f < 10 \).

**Real-world Networks:** Performance of the SRW-HHB-NETSIZE and RNA-HHB-NETSIZE estimators for real-world graphs are shown in Fig. 3.14. We select \( \omega = 10 \) for RNA-HHB-NETSIZE estimator. Similar to the average degree estimations, the estimators perform almost identical shown as overlapped 90% confidence intervals in Fig. 3.14. We observe that confidence intervals get wider as the margin parameter increases. This is expected as the estimators ignore more sample data as the margin parameter increases and the precision decreases.
Figure 3.14. Network size estimation on real-world graphs using SRW-HHB-NETSIZE and RNA-HHB-NETSIZE: 5\(^{th}\) percentile, median, and 95\(^{th}\) percentile of 1,000 estimations are shown for each estimator with \(t = 1\). For RNA-HHB-NETSIZE estimator, \(\omega = 10\) is used. Estimation results for the sampling fractions ranging from 1\% to 5000\% are shown on each plot. A safety-margin dependency reduction technique with different margin values are shown on each row: margin = \([1, 10, 100]\).

For both estimators, we observe that increasing margin causes more accurate estimations with less precision especially for \(f < 10\). For instance, the network size is underestimated with high precision for margin = 1 for all real-world graphs. However, the estimations are centered around the true network size for margin = 100 with wider confidence intervals. This illustrates the tradeoff between reducing the dependency and utilizing more data (the bias-variance tradeoff). If we ignore too much data, the precision becomes low but samples become more independent. On the other hand, if we do not ignore enough data, the precision becomes high, but sample becomes more dependent.
Performance of the SRW-VCB-NETSIZE and RNA-MVCB-NETSIZE estimators for real-world graphs are shown in Fig. 3.15. For RNA-MVCB-NETSIZE estimator, we observe very precise but inaccurate estimations for all graphs that we consider regardless of the margin parameter for $f < 10$. The main reason of this observation is that multiplicity values of vertices do not become proportional to the vertex degrees until we have a large enough sample size. In SRW-VCB-NETSIZE estimator, we observe the bias-variance tradeoff for small sample sizes. When margin = 1 (the top row in Fig. 3.15), we observe an underestimation with a high precision for small sample sizes ($f < 0.1$) for all graphs. When margin = 10 (the second row), we observe that estimations are not very precise but are around the true parameter. When margin = 100 (the bottom row), we observe much less precision due to
We conjecture that this is directly related to the average degree of the considered real-world graphs being close to 10. Ignoring too much data. In these graphs, \( \text{margin} = 10 \) seems to be better than the others. We do not observe underestimation when \( \mu_d = 1 \) for \( f < 20 \) in both ER and BA graphs. This can be explained by the underestimation of \( M \) due to small \( n_\omega \) and large number of collisions due to small average degree in Eq. 3.11. We do not observe underestimation when \( f > 1 \) and \( \mu_d = 100 \). This might be due to the balance of the underestimation of \( M \) and small \( C_E \) due to small average degree in Eq. 3.11.

![Graph](image-url)

Figure 3.16. Network size estimation on synthetic graphs using SRW-HHB-NETSIZE and RNA-HHB-NETSIZE: 5\(^{th}\) percentile, median, and 95\(^{th}\) percentile of 1,000 estimations are shown for each estimator with \( t = 1 \). For RNA-HHB-NETSIZE estimator, \( \omega = 10 \) is used. Estimation results for the sampling fractions ranging from 1\% to 5000\% are shown on each plot. A safety-margin dependency reduction technique with different \( \text{margin} \) values are shown on each row: \( \text{margin} = [1, 10, 100] \).

**Synthetic graphs:** Experimental results for SRW-HHB-NETSIZE and RNA-HHB-NETSIZE estimators on synthetic graphs are shown in Fig. 3.16. We observe that the estimation performance does not differ between ER and BA graphs when the average degree and margin parameters are the same although the graphs are structurally different. For \( \text{margin} = 1 \) case, we see underestimation when \( \mu_d = 4 \) and \( \mu_d = 10 \) for \( f < 20 \) in both ER and BA graphs. This can be explained by the underestimation of \( M \) due to small \( n_\omega \) and large number of collisions due to small average degree in Eq. 3.11. We do not observe underestimation when \( f > 1 \) and \( \mu_d = 100 \). This might be due to the balance of the underestimation of \( M \) and small \( C_E \) due to small average degree in Eq. 3.11.

![Graph](image-url)
to high average degree. When we increase *margin* to 10 or 100, we see that estimations for \( \mu_d = 4 \) or \( \mu_d = 10 \) cases in both ER and BA graphs are more accurate and precise than that of \( \mu_d = 100 \). These results illustrate that optimal *margin* parameter depends on the average degree of the underlying population graph as well as the sampling fraction.

Experimental results for SRW-VCB-NETSIZE and RNA-MVCB-NETSIZE estimators on synthetic graphs are shown in Fig. 3.17. Similar to the estimation results on real-world graphs, all estimations on synthetic graphs are very precise especially when sampling fraction \( f > 1 \). A very interesting observation is that RNA-MVCB-NETSIZE estimator is very accurate and precise when \( f > 1 \) and outperforms SRW-VCB-NETSIZE estimator. We also observe that SRW-VCB-NETSIZE estimator tends to overestimate the network size when \( 1 < f < 30 \) on ER graphs, while it underestimates the network size when \( 0.1 < f < 5 \) on BA graphs. Therefore, when \( 1 < f < 5 \), we consistently underestimate the network size in ER graphs, while we consistently overestimate the network size. This creates a practical challenge as the degree structure of the graph may cause consistent underestimations or overestimations. Without knowing the degree structure, we may not decide to which case the estimations belong to. Recall that in RNA-HHB-NETSIZE and SRW-HHB-NETSIZE estimators, degree structure did not affect the estimation accuracy as much. The selection of *margin* parameter and the average degree of the underlying graph was the main factors affecting the performance of the estimators.

**Comparison of RNA-HHB-NETSIZE with RNA-MVCB-NETSIZE:** In real-world graphs, we observe that with a good selection of \( \omega \) (e.g., \( \omega = 100 \)), RNA-HHB-NETSIZE outperforms RNA-MVCB-NETSIZE in terms of accuracy when \( f < 10 \) (see Fig. 3.14 and Fig. 3.15). However, when \( f \approx 10 \), we see that RNA-MVCB-NETSIZE estimator starts making estimations with high precision and accuracy (see GNUTELLA and CONDMAT in Fig. 3.15). In the synthetic graphs, we observe that RNA-HHB-NETSIZE estimator is very precise and accurate when \( f > 20 \) (even larger in some graphs, e.g., ER(\( \mu_d = 100 \))
3.5 Discussion

In this section, we discuss several practical issues related to our problem context including adaptive number of RN-QUERY calls per vertex and estimation in dynamic graphs.

### 3.5.1 General Discussion on Practical Issues

One of the assumptions of our estimators is that the sample is collected by a stationary random walk. In practice, this can be achieved by ignoring a number of initial steps in the random walk (known as burn-in period). In general, the number of steps until convergence, i.e., the mixing time, depends on the underlying structure of the graph and can be bounded

---

**Figure 3.17.** Network size estimation on synthetic graphs using SRW-VCB-NETSIZE and RNA-MVCB-NETSIZE: 5th percentile, median, and 95th percentile of 1,000 estimations are shown for each estimator. Estimation results for the sampling fractions ranging from 1% to 5000% are shown on each plot. A safety-margin dependency reduction technique with different margin values are shown on each row: margin = [1, 10, 100].

and $BA(\mu_d = 100)$ in Fig. 3.16), while the similar performance is obtained in RNA-MVCB-NETSIZE when $f > 2$ and margin > 10 (see Fig. 3.17).
by a function of the modulus of the second largest eigenvalue of the graph. Mohaisen et al. [82] study this dependency and the mixing time of various OSN graphs. Gjoka et al. [39] apply two formal convergence diagnostic techniques on OSN graphs. Note that the burn-in period is relevant for both RSRW and ERSRW sampling schemes as they are just variations of a random walk.

To estimate the network size, one can be tempted to save unique vertex ids visited by SRW until no new ids are obtained instead of using the multiplicity based estimator which requires high sample sizes ($f \approx 10$). However, this idea has much higher cost ($((1 - o(1)) \log n < f < O(4m))$) for general undirected and non-bipartite graphs [73],\(^{11}\) therefore, we do not consider a complete topology discovery as a viable alternative.

Another point is that any nodal property that is not correlated with the vertex ids are impossible to estimate under the RNA model. For instance, if we study the average age of an OSN users, we should either observe or estimate the age of each sampled user together with its degree in order to correct the bias. Even though we can estimate the vertex degrees under the RNA model, estimating the age of a user (and hence the bias correction) is not possible as there is no information revealed by the RN-QUERY.

### 3.5.2 Estimation with Adaptive Number of RN-QUERY Calls

Note that our proposed estimators use a sampling scheme that makes a fixed number of RN-QUERY calls per visited vertex. In this section, we propose average degree and network size estimators that use an alternative sampling scheme making variable number of RN-QUERY calls per visited vertex. Finally, we run simulation experiments using our real-world graphs to evaluate the performance of these estimators as compared to the estimators using fixed number of RN-QUERY calls.

\(^{11}\)Time to visit all vertices in a graph with a SRW is also known as the cover time of the graph.
Algorithm 6 On-the-fly estimator for the reciprocal of the degree of the vertex $u$ under ERSRW sampling with adaptive probing.

1: function RNA-REC-DEGREE-ADAPTIVE($G, u, c$)  
2:     $nbors \leftarrow \{\}$  
3:     $coll \leftarrow 0$  
4:     $\omega \leftarrow 0$  
5:     while $coll < c$  
6:         $randNbor = RN-QUERY(G, u)$  
7:         if $nbors$ does not have key $randNbor$  
8:             $nbors[randNbor] \leftarrow 0$  
9:         end if  
10:         $coll \leftarrow coll + nbors[randNbor]$  
11:         $nbors[randNbor] \leftarrow nbors[randNbor] + 1$  
12:     return $coll \binom{c}{2}$  
13: end function

ERSRW with Adaptive Probing: In this sampling design, RN-QUERY calls are made for the current vertex in the walk until the number of collisions exceeds the threshold $c$. Note that this type of probing is more likely to make more RN-QUERY calls for a vertex with high degree. However, it is not obvious whether this property improves the overall performance of the estimation. The walk is biased towards high degree nodes and these nodes are more likely to consume more sampling budget. Therefore, the walker may visit less number of vertices. Basically, there is a tradeoff between visiting many nodes and better reciprocal degree estimation for high degree vertices.

Algorithm 6 shows the pseudocode of the estimator for the reciprocals of the vertex degrees. Note that the estimator is very similar to RNA-REC-DEGREE but this time $\omega$ is not fixed. In our simulations, we use RNA-REC-DEGREE-ADAPTIVE, instead of RNA-REC-DEGREE, as a subroutine in the average degree and the network size estimations. We call the corresponding estimators as RNA-AVG-DEGREE-ADAPTIVE and RNA-HHB-NETSIZE-ADAPTIVE, respectively.

Fig. 3.18 and Fig. 3.19 show our simulation results. The main findings are as follows: estimation accuracy is generally worse compared to the RNA-AVG-DEGREE and RNA-HHB-
Figure 3.18. Average degree estimation (synthetic graphs) with no thinning, \( t=1 \): 5\(^{th} \) percentile, median, and 95\(^{th} \) percentile of 1,000 estimations are shown.

NETSIZE estimators that use ERSRW with fixed probing. The parameter \( c \) has a crucial effect on the bias-variance tradeoff. Choosing too small \( c \) gives an underestimation of the average degree with high precision while choosing large \( c \) gives a less biased estimation with higher variance (see Fig. 3.18). We also observe the same phenomena in the estimation of the network size (see Fig. 3.19).

### 3.5.3 Estimation on Dynamic Graphs

One of the assumptions of our estimators is that the underlying network is static. However, it is still important to understand when our statistical model break and estimations become inaccurate. We perform a simulation study on a publicly available real-world social graph data which represents the friendship relation of the users of a social news website named
SRW-HHB-NETSIZE-ADAPTIVE
RNA-HHB-NETSIZE-ADAPTIVE
SRW-HHB-NETSIZE-ADAPTIVERNA-HHB-NETSIZE-ADAPTIVE

Figure 3.19. Network size estimation (real-world graphs): 5th percentile, median, and 95th percentile of 1,000 estimations are shown for both SRW-HHB-NETSIZE-ADAPTIVE and RNA-HHB-NETSIZE-ADAPTIVE estimators with $t = 1$. For RNA-HHB-NETSIZE-ADAPTIVE estimator, $c = 10$ and margin = 10 is used.

digg.com [62]. The dataset contains a total of 1.7 million friendship edges and their creation time between August 2005 and July 2009.

We perform the experiment as follows: 1) For the first 10 days, we initialize the network without sampling. 2) Then, we make RN-QUERY calls periodically, which is controlled by a parameter named query interval. 3) We estimate the average degree every 24 hours using RNA-AVG-DEGREE-ADAPTIVE estimator with $c = 10$. While performing the average degree estimation, we only use the degree estimations of the last couple of nodes, which is controlled by the window size parameter. Fig. 3.20 illustrates the estimation results where x-axis represents the simulation time since the beginning of the experiment. We see that the average degree of the graph does not change much in the first 800 days and then it gradually increases. When the true average degree does not change over time, we observe that
Figure 3.20. Average degree estimation of dynamic friendship graph of digg.com, a social news website, over the period of approximately 4 years (from August 2005 to July 2009). Red dotted lines represent the 95th percentile, median, and the 5th percentile of 1000 estimations per day from top to bottom in each graph. The graph illustrates the effects of the estimation parameters on the bias-variance tradeoff over time. We use the abbreviation \textit{min} for minutes in the query interval.

estimations perform almost the same and the effects of the parameters are not observable. In the rest, we focus on the days after the true average degree starts increasing.

First, we observe the effect of window size on the bias-variance tradeoff. When the window size is too small, we observe high variance in the estimation, which can be attributed to the small sample size considered in the estimation. When the window size is too large, we observe a bias in the estimation as the past data is not representative of the current network state. Note that it is intuitive that when the actual average degree increases (decreases) over time, using a large window size causes underestimation (overestimation).

The selection of query interval is also important as it determines (together with the window size parameter) how representative our sample data used in the estimation. Basically,
low query interval value helps to take samples fast and the window fills fast with more recent data; and hence the sample is more representative of the current data. We observe that the increase in the estimation is not enough to catch the true value when the query interval is 10 minutes and window size is 1,000 or 10,000. When the query interval is reduced to 1 minute, we observe that the increase in the estimation is better in catching the true value although with the window size=10,000 still results in an underestimation.

In summary, the estimation accuracy highly depends on how fast the true value changes and whether the change has certain patterns such as gradual increase or decrease. In addition, the limitations imposed by the OSN service providers on the frequency of queries affects the estimation accuracy. Note that these factors are independent of the sampling design. In order to achieve a good accuracy, the sampling design parameters such as query interval and window size should be selected carefully.

3.6 Conclusion

In this study, we have proposed sampling schemes and estimators for the network size and the average degree properties under the RNA model. Using simulations on both real-world and synthetic graphs, we have demonstrated that network size estimation requires high sampling fractions \((f > 10)\) under the RNA model for accurate and precise estimations. However, the average degree can be estimated with high accuracy and precision using practical sample sizes \((f < 0.2)\) with a reasonable selection of sampling design parameter. We have also empirically shown that applying dependency reduction technique on the average degree estimator decreases the precision; while applying it in the network size estimators may increase the accuracy of the estimations. We also discussed the several practical issues of estimation under the RNA model. We believe this study will motivate future studies to develop estimators under various other interesting sampling designs and not to stick with
the typical sampling designs such as independent vertex/edge sampling or induced subgraph sampling when they are not efficient under the specific problem constraints.
CHAPTER 4

AVERAGE DEGREE ESTIMATION ON LARGE GRAPHS UNDER EGO-CENTRIC SAMPLING

Estimating the structural characteristics of large graphs from a sample is a classical problem. Researchers have focused on developing efficient and accurate estimators for structural characteristics of large graphs. Although the term important is ambiguous, some common various characteristics including graph size [47, 65, 58, 116], average degree [21, 24], degree distribution [89, 38, 88, 97], average clustering coefficient [47, 88, 38], global clustering coefficient [47], diameter [24, 71], and centrality measures [69].

Although it is very common to use a random walk or its variation to collect data from popular large scale graphs such as Online Social Network (OSN) graphs [90, 64, 88, 39], some OSNs such as LinkedIn does not allow its regular users to perform a random walk on the underlying graph through API calls. As a result, proposing estimators under various sampling designs, other than random walk, is beneficial to such OSNs. In addition, other disciplines that require a sample from a graph may have other limitations on data collections that restrict using a completely new sampling design.

In this study, we develop estimators for average degree characteristic of a large graph under ego-centric sampling design with uniform ego vertex selection. The estimators can easily be generalized to other vertex characteristics. For a given graph $G = (V, E)$, ego-centric graph (or network) of a vertex $i \in V$ refers to the subgraph induced by the vertex $i$ and its immediate neighbors called alters. Ego-centric sampling design consists of the following

---

1© 2016 IEEE. Reprinted, with permission, from Emrah Cem and Kamil Sarac, Average degree estimation under ego-centric sampling design, Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on, April 2016
Figure 4.1. An example graph illustrating the two different cases for an ego-centric graph sample with 2 ego vertices (vertex 1 and 9). Vertex degree is available as a vertex label (on the left) and vertex degree is not available as a vertex label (on the right) cases are shown.

steps: an ego vertex is sampled and its ego-centric graph is observed. Then, this procedure is repeated until either a fixed number of ego-vertices are sampled or a fixed number of observations are made, depending on the sampling problem under consideration. Observing a graph may have different meanings. In this study, we consider an observation scheme in which observing the ego-centric graph of a vertex $i$ implies observing all the edges and the vertices in the ego-centric graph of $i$. Observing an edge means observing the ids of the vertices incident on this edge and observing a vertex means observing its id and its labels. For instance, the vertex label can be the age or gender of a user in an OSN network.

We develop statistical estimators for ego-centric sampling under two different scenarios: 1) vertex degree is available as a vertex label and 2) vertex degree is not available as a vertex label (see Fig. 4.1). The main difference between these scenarios is that the degrees of all vertices in the ego-centric graphs of the sampled ego vertices are observed in the former, while the degrees of only the ego-vertices are observed in the latter. Estimation of the average degree of a graph under these two scenarios has both theoretical and practical value. The fundamental question that we ask is "When we have several ego-centric graph samples, which
information should we utilize to get the best estimator, information only from the ego vertices, only from the alters, or both?“.

The main contributions of this study are as follows:

• We propose four estimators for the average degree characteristic under the ego-centric sampling design and compare their accuracy to EGOS estimator, a plug-in estimator for the random vertex sampling).

• We theoretically analyzed the asymptotic behavior of the proposed estimators.

• We performed extensive simulation on both real-world and synthetic graphs to measure the accuracy of the estimators.

Our experimental results show that even when observing the ego-centric graphs of the sampled vertices is free, utilizing them in the estimation (which is done by ALL, ALL-HT, and ALTERS estimators) does not necessarily increase the accuracy of the estimation. The accuracy of the estimation not only depends on the amount of the utilized observation but also on the interaction between the underlying graph structure and the way the information is utilized in the estimator.

The rest of this work is organized as follows. Section 2 presents preliminaries. Section 3 presents the network model and the ego-centric sampling design. Section 4 proposes various estimators for average degree. Section 5 presents the experimental results. Section 6 presents the related work. Finally, Section 7 concludes the chapter.

4.1 Preliminaries

In this section, we discuss design-based and model-based approaches along with their advantages and disadvantages. We also discuss how sampling design plays an important role in estimation and how its quality is measured. Lastly, we introduce estimation techniques that are commonly used under sampling designs which select units with unequal probabilities.
**Design-based vs. Model-based Approach in Sampling:** In design-based approach, the values of the variable of interest in the population are considered as fixed but unknown constants and randomness comes into play only through the sampling design. The estimator is a random variable whose value depends on which sample is selected. In model-based approach, the values of the variables of interest are considered to be the realizations of a vector of random variables. Design-based approaches are widely used as they are unbiased independent of the nature of the population. They also avoid the potentially catastrophic effects of important but unknown auxiliary variables. Benefits of the model-based approach include dealing with non-sampling errors and data obtained without any proper sampling design, making good use of auxiliary information, and suggesting good designs to use for certain populations. The common problem with the model-based approach is that even though models become mathematically complex, they are not realistic enough for real-world populations.

We follow the design-based estimation procedure in this study since the structural characteristics of social networks vary, e.g., social networks are assortative, whereas technological and biological networks are disassortative [83]. Thus, making estimation without any dependence on the network structure seems a better choice.

**Sampling Design:** The procedure by which a sample of the units is selected from the population is called sampling design. In the rest of the chapter, while referring to the name of the sampling design we will omit the word design, e.g. random walk sampling. Researchers studied the estimation of several characteristics under various sampling designs such as random walk sampling, induced graph sampling, star sampling, uniform independent vertex/edge sampling, weighted vertex/edge sampling, and snowball sampling. Since different sampling designs generally have different selection/inclusion probabilities for the studied units (vertices, edges, triangles, etc.), they have different bias characteristics. For instance, random walk sampling is widely used in OSNs since it is applicable in popular OSNs and
computing the inclusion probabilities of simple objects such as vertices, edges, triangles is tractable. Therefore, statistical estimations about their properties such as total, average, or proportion quantities can be done. Some sampling designs, on the other hand, have intractable calculations of inclusion probabilities, even for the vertices, resulting in combinatorial issues. Snowball sampling after one-stage is such a sampling design. There are several definitions of snowball sampling in the literature and we use the definition in [59].

In general, the key factor in applying a sampling design to a problem lies in its applicability as well as the computational tractability of computing inclusion probabilities of the studied units. We consider a sampling design, ego-centric sampling, that lies in between labeled star sampling, which has already been considered, and two-stage snowball sampling, which has computationally intractable inclusion probabilities.

Probability vs. Non-probability Sampling: Sampling designs can be divided into two categories based on whether the probability of selection of units can be computed or not. A probability sampling is any method of sampling that utilizes some form of random selection which gives a known, positive selection probability to every element. Otherwise, it is called non-probability sampling. The advantage of probability sampling is that it is possible to develop unbiased estimators for the population parameter by weighing the values of sampled units based on their selection probabilities.

If there are units with selection probability of zero (also called missing data) in the target population, then generalizing the estimation to the target population might give biased results. There are two-general techniques to overcome the effects of missing data: model-based methods and imputation. We do not consider missing data in this study as it is a broad topic that deserves attention in a study of its own.

Ego-centric sampling is a probability sampling design. We are aware that the results can only be generalized to the set of elements within the sampling frame, i.e., the material or device from which sample is obtained, if there are unobservable units in the target population.
How Good is the Estimator: Several different objectives for sampling have been considered in the network/graph sampling literature. Some studies focused on the estimation of a single characteristic [21, 109, 47, 22]. Some others tried to obtain a sample graph preserving several important characteristics simultaneously [54]. Note that designing an unbiased estimator for a characteristic does not depend on obtaining a sample graph preserving that characteristic. Induced subgraph sampling is an example which underestimates the average degree in general, but an unbiased estimator can be developed using the observed degrees of the nodes. Nevertheless, depending on the characteristic under study, finding an efficient estimator might be challenging under the constraint of data availability.

We evaluate the performance of the estimators based on how well the single studied characteristic is estimated since the performance of the estimator depends heavily on the studied characteristic [16] and preserving several characteristics simultaneously may not be possible [2], e.g., average degree and density.

Estimation Under Unequal Probability Sampling: Different units in the population graph might have different probabilities of inclusion in the sample depending on the sampling design that is used. These probabilities should be taken into consideration to make accurate estimation. Horvitz-Thompson (HT) [53], Hansen-Hurwitz (HH) [46], and generalized mean estimator (GEN) [102] are three commonly used estimators under unequal probability sampling. While the first two are unbiased estimators for population totals, the last one is an asymptotically unbiased estimator for population averages, i.e., bias approaches to zero as the sample size increases. In GEN estimator, it is sufficient to know the selection probabilities, denoted as $p$, up to a constant to reduce the sampling bias.

We compute the inclusion/selection probabilities of vertices and edges under ego-centric sampling design and utilize the HT, HH, and GEN estimators to remove the sampling bias in the estimation of the average degree.
Table 4.1. Table of Notation for the Network Model and Sampling Design.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>the number of vertices in $G$</td>
</tr>
<tr>
<td>$M$</td>
<td>the number of edges in $G$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>neighbors of vertex $i$</td>
</tr>
<tr>
<td>$N_i^+$</td>
<td>the closed neighborhood of vertex $i$ (i.e., $N_i \cup {i}$)</td>
</tr>
<tr>
<td>$k_i$</td>
<td>the degree of vertex $i$</td>
</tr>
<tr>
<td>$\langle k \rangle$</td>
<td>the average degree of $G$</td>
</tr>
<tr>
<td>$C_{(i,j)}$</td>
<td>the common neighbors of vertex $i$ and $j$, $(i, j) \in E$</td>
</tr>
<tr>
<td>$C_{(i,j)}^+$</td>
<td>$C_{(i,j)} \cup {i, j}$</td>
</tr>
<tr>
<td>$n$</td>
<td>the sample size</td>
</tr>
<tr>
<td>$v$</td>
<td>the number of distinct units in the sample</td>
</tr>
<tr>
<td>$\pi_i$</td>
<td>inclusion probability of vertex $i$</td>
</tr>
<tr>
<td>$p_i$</td>
<td>selection probability of vertex $i$</td>
</tr>
<tr>
<td>$S^e$</td>
<td>the list of sampled ego vertices (possibly repeating)</td>
</tr>
<tr>
<td>$s_i$</td>
<td>$i^{th}$ sampled ego vertex</td>
</tr>
<tr>
<td>$G_i^e$</td>
<td>ego-centric graph of $i$</td>
</tr>
</tbody>
</table>

4.2 Network Model and Sampling Design

4.2.1 Network Model

We represent the network as a simple graph $G = (V, E)$ with a finite set of vertices $V$ and edges $E$. We provide the graph related notations in Table 4.1.

4.2.2 Ego-Centric (EC) Sampling Design

In ego-centric sampling, $n$ ego vertices are selected and their ego-centric graphs are observed. Within the ego-centric graph of $i$, $i$ is called ego vertex and all other vertices are called alters. Sampling related notations are given in Table 4.1. Definition of ego-centric graph that we used in this study has been used in the literature as (radius-1) ego-centered network [84].
Ego-centric sampling lies in between labeled-star sampling [59] and two-stage snowball sampling [42] in terms of being able to observe the neighborhood of ego vertices.

Below is the summary of the sampling designs considered in this study:

- **How are ego-vertices sampled?** We consider the selection of ego vertices uniformly at random with replacement.

- **How many ego-vertices are sampled?** We consider two cases: 1) Free neighborhood information: we only spend 1 unit of cost per ego-centric graph; therefore $n$ ego vertices are selected, 2) Non-free neighborhood information: We spend $|k_i + 1|$ units of cost when vertex $i$ is selected as an ego vertex. After selecting an ego vertex obtaining each alter also requires 1 unit of cost. Therefore, the total number of ego vertices that we select depends on the sample with the expected value of approximately $n/(\langle k \rangle + 1)$.

- **Is the degree information of all vertices in an ego-centric graph observable?** We consider two cases: 1) Degree information available as a vertex label: The degrees of all vertices in the ego-centric graph are observable as observing a vertex implies observing its labels, 2) Degree information not available as a vertex label: Although the degrees of vertices are not observable, the degree of the ego vertex can be obtained from the number of incident edges. For alters, we only have incident edges within the ego centric graph. If an alter has an edge to a vertex that is not a direct neighbor of the ego vertex, that edge is not observable.

### 4.3 Estimation

In this section, we first provide the statistical estimators that we use to correct/reduce the sampling bias. Then, we provide several estimators for the average degree, i.e., $\langle k \rangle = 1/N \sum_{i=1}^{N} k_i$, under the ego-centric sampling design. Our estimators differ in the information that they utilize in the sample.
4.3.1 Estimation Under Unequal Probability Sampling

Different units in the population graph might have different probabilities of inclusion in the sample depending on the sampling design that is used. These probabilities should be taken into consideration to make accurate estimation. Horvitz-Thompson (HT) [53], Hansen-Hurwitz (HH) [46], and generalized mean estimator (GEN) [102] are three commonly used estimators under unequal probability sampling. While the first two are unbiased estimators for population totals, the last one is an asymptotically unbiased estimator for population averages, i.e., bias approaches to zero as the sample size increases. In GEN estimator, it is sufficient to know the selection probabilities, denoted as $p$, up to a constant to reduce the sampling bias.

4.3.2 Degree Information is Available as a Vertex Label

In this section, we propose estimators for the average degree when the vertex degree is available as a vertex label. We first give the EGOS estimator which is a plug-in estimator utilizing only the degrees of ego vertices:

$$\langle k \rangle_{EGOS} = \frac{1}{n} \sum_{i=1}^{n} k_{s_i}. \quad (4.1)$$

Since ego vertices are selected uniformly at random, the estimator is unbiased. It is also well-known that if we sample $n = 2/(\epsilon^2 \delta)$ ego vertices, the naive estimator gives an estimate $\langle \hat{k} \rangle$ such that $|\langle \hat{k} \rangle - \langle k \rangle| < \epsilon$ with probability $1 - \delta$. On the other hand, when we also utilize the degrees of alters in the estimator, the plug-in estimator overestimates the average degree due to the bias towards high degree vertices. In the rest of this section, we propose estimators that correct this bias in the limit by reweighing the degrees of vertices utilized in the estimation. More specifically, we propose three estimators: 1) ALTERS estimator, a GEN estimator that utilizes only the degrees of alters and ignores the ego vertices, 2)
ALL estimator, a GEN estimator that utilizes the degrees of both ego vertices and alters, and corrects the bias using draw-by-draw selection probabilities of vertices, and 3) ALL-HT estimator that also utilizes the degrees of both ego vertices and alters but corrects the bias using the inclusion probabilities of vertices.

We first provide ALTERS estimator. This estimator is similar to the estimator proposed by Dasgupta et al. [22] but we do not require the knowledge of the network size. Another similar estimator is proposed by Wang et al. [109] but they propose a density estimation using a walk based sampling while we are interested in the average value and the random walk is not applicable. ALTERS estimator is given by

\[
\hat{k}_{\text{ALTERS}} = \frac{\frac{1}{n} \sum_{i=1}^{n} \sum_{j \in N_{s_i}} k_i k_j}{\frac{1}{n} \sum_{i=1}^{n} \sum_{j \in N_{s_i}} \frac{1}{k_j}} = \frac{\sum_{i=1}^{n} k_{s_i}}{\sum_{i=1}^{n} \sum_{j \in N_{s_i}} \frac{1}{k_j}}.
\] (4.2)

A vertex is utilized in this estimator whenever one of its neighbors is sampled as an ego vertex. Therefore, the bias is proportional to the vertex degree and the estimator corrects this bias by reweighing the degree of a vertex by a value proportional to its selection probability.

**Lemma 4.** \(\hat{k}_{\text{ALTERS}}\) is an asymptotically unbiased estimator of \(\langle k \rangle\).

**Proof.** From Strong Law of Large Numbers (SLLN), for any function \(f\) satisfying \(\sum_{v \in V} |f(v)| < \infty\),

\[
\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} f(s_i) \xrightarrow{a.s.} \frac{1}{|V|} \sum_{v \in V} f(v),
\] (4.3)

where \(\xrightarrow{a.s.}\) denotes the almost sure convergence. Therefore, if we do not simplify the \(1/n\) term in both numerator and the denominator, we get

\[
\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} k_{s_i} \xrightarrow{a.s.} \frac{1}{N} \sum_{i=1}^{N} k_i = \langle k \rangle
\] (4.4)
\[ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \frac{1}{k_j} \xrightarrow{a.s.} \frac{1}{N} \sum_{v \in V} \sum_{j \in \mathcal{N}_v} \frac{1}{k_j} = 1. \] (4.5)

The last step of (4.5) can be obtained by observing that for each vertex \( i \in V \), there are exactly \( k_i \) terms of \( 1/k_i \) in the overall sum. Finally, the result \( \langle \hat{k} \rangle_{ALTERS} \xrightarrow{a.s.} \langle k \rangle \) follows from the Slutsky’s theorem [70]. We omit the proof of the remaining lemmas in the study as they can be obtained using the similar idea in this proof. \( \square \) \( \square \)

Note that ALTERS estimator estimates \( n \) in the denominator of (4.2) instead of using the known value of \( n \) unlike the naive estimator. Although it is counterintuitive to use an estimate of a known value in an estimator, depending on the degree correlation of ego vertices and their alters, \( \langle \hat{k} \rangle_{ALTERS} \) estimator may outperform \( \langle \hat{k} \rangle_{EGOS} \) estimator. For instance, when the numerator overestimates (underestimates) the total degree of \( n \) ego vertices by chance, if the denominator also overestimates (underestimates) \( n \), then they may balance each other and give an estimate close to the true average degree of the population graph.

We propose another estimator, called ALL, that utilizes the degrees of all the observed vertices, i.e., both ego vertices and alters. This is also a GEN estimator similar to ALTERS estimator, but the bias correction changes since the degree of vertex \( i \) is also utilized in the ALL estimator when vertex \( i \) is selected as an ego vertex unlike the ALTERS estimator. Using these, we provide the following asymptotically unbiased estimator for the average degree:

\[ \langle \hat{k} \rangle_{ALL} = \frac{\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \frac{k_j}{k_j+1}}{\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \frac{1}{k_j+1}}. \] (4.6)

We propose another estimator, called ALL-HT, that uses the inclusion probabilities of vertices instead of draw-by-draw selection probabilities. In general, inclusion probability of a vertex under with replacement sampling schemes can be computed as one minus the probability of not observing this vertex in any sample. In ego-centric sample, vertex \( i \) is
observed if a vertex in $\mathcal{N}_i^+$ is selected as an ego vertex; therefore, the inclusion probability of vertex $i$ becomes

$$\pi_i = 1 - \left(1 - \frac{k_i + 1}{N}\right)^n. \quad (4.7)$$

Using these probabilities, we derive the following HT-based asymptotically unbiased estimator for the average degree:

$$\langle \hat{k} \rangle_{\text{ALL-HT}} = \frac{1}{N} \sum_{j \in \bigcup_{i=1}^n \mathcal{N}_i^+} \frac{k_j}{\pi_j}. \quad (4.8)$$

As opposed to the estimator in (4.6), HTbased estimator utilizes each observed unit only once. One of the requirements of this estimator is the knowledge of network size. We skip the proof of this estimator as it is trivial to observe that as we sample more and more vertices, the set of vertices utilized in the estimator converges to the set of vertices in the population graph.

In summary, in this section, we have proposed different estimators for the average degree of a graph. Each of the estimators utilizes different information in the sample. Depending on whether observations are biased or not, observed degrees might be reweighed which affects the variance of these estimators. For instance, EGOS estimator uses unbiased observations and does not have to do bias correction. Therefore, the variance of the EGOS estimator is a simple function of the degree variance of the population graph. On the other hand, the estimators that we propose (ALTERS, ALL, and ALL-HT) use biased observations and perform bias correction by reweighing the observed degree information. These bias correction operations affect the relationship between the degree variance of the population and the variance of these estimators. As the estimators are more involved and the closed form of their variances are not easy to interpret, we measure the accuracy of these estimators through simulations on both real-world and synthetic graphs.
4.3.3 The Degree Information is not Available as a Vertex Label

In this section, we propose the EDGES estimator for the average degree when the vertex degree is not available as a vertex label. Note that all incident edges of ego vertices are still observable, so degrees of egos are indirectly available. Therefore, one can use EGOS estimator in this scenario. However, an interesting question is whether we can utilize the edges between alters in an estimator and improve the accuracy of the estimator. In the rest of this section, we provide such an estimator called EDGES estimator. An additional requirement of EDGES estimator is the availability of the number of common neighbors of alters, i.e., $|C_{(i,j)}|$, when both $i$ and $j$ are alters. In the context of OSNs, OSN service provider can provide this information without revealing the degrees of the alters. When $i$ is an ego vertex and $j$ is an alter vertex, $|C_{(i,j)}|$ is the number of alters in $G_i^e$ that are neighbors of $j$; therefore, we do not need an additional information.

EDGES estimator estimates the average degree by estimating the total number of edges in the population graph and normalizing it by $2/n$ (recall that $\langle k \rangle$ can also be defined as $2M/N$). Note that the selection probability of an edge $(i, j)$ is

$$p_{(i,j)} = \frac{|C_{(i,j)}^+|}{N}. \quad (4.9)$$

Using probabilities in (4.9), we correct the bias by reweighing the count of each edge by its selection probability and derive the following asymptotically unbiased HH-based estimator for average degree:

$$\langle \hat{k} \rangle_{EDGE} = \frac{2}{Nn} \sum_{m=1}^{n} \sum_{(i,j) \in E_{sm}^e} \frac{1}{p_{(i,j)}} = \frac{2}{n} \sum_{m=1}^{n} \sum_{(i,j) \in E_{sm}^e} \frac{1}{|C_{(i,j)}^+|}, \quad (4.10)$$

where $E_{sm}^e$ denotes the set of edges in the ego-centric graph of $m^{th}$ sampled ego vertex.

Note that this estimator does not require $N$ as it cancels out but it requires $|C_{(i,j)}^+|$ to correct the bias.
Table 4.2. Real-world topologies used in experimental evaluations.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>N</th>
<th>M</th>
<th>$\langle k \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP Community</td>
<td>260998</td>
<td>628915</td>
<td>4.82</td>
</tr>
<tr>
<td>Enron Email</td>
<td>36692</td>
<td>183831</td>
<td>10.02</td>
</tr>
<tr>
<td>Epinions</td>
<td>75877</td>
<td>405737</td>
<td>10.69</td>
</tr>
<tr>
<td>Stanford Web</td>
<td>281903</td>
<td>1992636</td>
<td>14.14</td>
</tr>
</tbody>
</table>

4.4 Experimental Evaluations

4.4.1 Experimental Setup

Datasets: We perform our evaluations on both synthetic and real-world graphs. More specifically, we use Erdős-Rényi $G(N, p)$ and Barabási-Albert $G(N, m)$ graph models to generate synthetic graphs. They represent two different family of synthetic graphs. Erdős-Rényi model creates purely random graphs and Barabási-Albert model creates scale-free graphs using a preferential attachment mechanism. We create synthetic graphs of size $N = 10,000$ with average degree approximately 10 by setting $p = 0.001$ and $m = 5$ in Erdős-Rényi and Barabási-Albert models, respectively.

Real-world graphs that we use in the experiments are shown in Table 4.2. The datasets and their descriptions are publicly available at http://snap.stanford.edu/data/.

Error Metric: We use normalized root mean square error (NRMSE), which is a common measure used to quantify the relative error of the estimator $\hat{\theta}$ with respect to its true value $\theta$. It is defined as

$$NRMSE(\hat{\theta}) = \sqrt{\frac{E[(\hat{\theta} - \theta)^2]}{\theta^2}}.$$  \hspace{1cm} (4.11)

When $\hat{\theta}$ is an unbiased estimator of $\theta$, $E[\hat{\theta}] = \theta$, $NRMSE(\hat{\theta}) = std(\hat{\theta})/\theta$. Empirical NRMSE of our simulations, i.e., each data point in graphs, is obtained over 100 runs.
Figure 4.2. *Free neighborhood* information: Errors of estimators utilizing different information in the ego-centric graph sample of real-world graphs. Sample fraction corresponds to the proportion of observed number of ego-vertices, equivalently the observed number of ego-centric graphs, to the population size $N$.

Figure 4.3. *Non-free neighborhood* information: Errors of estimators utilizing different information in the ego-centric graph sample of real-world graphs. Sample fraction corresponds to the proportion of total number of observed vertices to the population size $N$.

### 4.4.2 Estimator Accuracy

In this section, we experimentally compare the accuracy of the estimators. Following the presentation in Section 4.3, we divide the experiments in two parts: one on evaluating estimators for the ego-centric sampling design when the degree information is available as a vertex label and the other on evaluating estimators for the ego-centric sampling design when the degree information is not available as a vertex label.

**The degree information is available as a vertex label:** We first analyze the free neighborhood case in which a fixed number of ego vertices are sampled. Fig. 4.2 shows the estimation errors of all estimators provided in Section 4.3 on real-world graphs shown in Table 4.2. One general observation from the figure is that estimators utilizing all the information that comes for free, namely ALL and ALL-HT, outperform the EGOS and
ALTERS estimators that partially utilize the available information in all the real-worlds graphs that we consider. Especially ALL-HT has much better accuracy for larger sample sizes which can be attributed to its usage of sufficient statistic in the estimation. It only uses the unique observations in the sample. On the other hand, ALL-HT requires the network size information which may not be available. The reader is referred to [102] for more details about using sufficient statistic in the estimation. Another interesting observation is that EGOS and ALTERS estimators have similar estimation errors which might be counterintuitive as ALTERS estimator utilizes much more information. For instance in Stanford Web graph, ALTERS estimator uses ≈ 14 times more data than EGOS estimator. Similar performance can be explained with the fact that the sample size is not the only criteria in determining the accuracy of an estimator. As we discuss in Section 4.3, due to the interaction between the characteristic under study and the selection probabilities of vertices under ego-centric sampling design, EGOS and ALTERS estimators give similar estimates. The only difference is that EGOS estimator divides the total degree of ego vertices by the number of observed ego vertices, while ALTERS estimator estimates the number of observed ego vertices from the data although it is known. ALTERS estimator takes a biased sample and reweighs the observed values to reduce the bias which causes it not to use the full power of the large sample size.

Next, we consider the case where obtaining neighbor information during sampling is not free. In this case, the number of observations in the sample is fixed rather than the number of ego vertices. Fig. 4.3 shows the estimation errors of the estimators on real-world graphs.

One general observation from this graph is that in the applications where neighbor information is not free, EGOS estimator which is simple and does not require any correction performs quite well when compared to the other estimators that use the biased data and corrects it. Although EGOS estimator has the advantage of unbiased data, depending on the relation between degrees of ego vertices and alters, ALTERS, ALL, and ALL-HT may
Figure 4.4. Comparison of average degree estimators EGOS and EDGES on Enron Email graph when the degree information is not available as a vertex label. Results for *free-neighborhood* data are on the left, results for *non-free neighborhood* data are on the right.

outperform the EGOS estimator. For instance, ALL estimator in DBLP Community graph, outperforms the EGOS estimator for sample fractions less than 10%. We also observe that ALL-HT estimator is unstable especially in Stanford web graph with sample sizes less than 5% due to high skewness of the degree distribution. In Stanford Web graph, vertex degrees range from 1 to 38625 and there are only 35 nodes with degree larger than 5000.

**The degree information is not available as a vertex label:** In this part, we compare the accuracy of EDGE estimator with EGOS estimator. Recall that EGOS estimator is still applicable as the incident edges of all egos are observable. ALL and ALL-HT are not applicable as they use the degrees of alters which is not available in this scenario. Fig. 4.4 shows the results of average degree estimation on Enron Email graph. We obtain similar results for other graphs. The results conform that utilizing more information in the sample does not always result in better estimation. Recall that EDGES estimator requires the knowledge of the number of triangles an edge is involved in. Even if this information is available to correct the bias, EGOS estimator outperforms EDGES estimator. However, we do not generalize this result for all types of graphs. The variance of EDGES estimator depends on the variance of the inner sum of (4.10) and it is related to the $C_{(i,j)}^+$ values of the edges on the population graph. If the underlying population graph has a structure such that the sum of the reciprocals of the $C_{(i,j)}^+$ values at each ego-centric graph has a low variance but
the degree distribution has a high variance then EDGES estimator would outperform the EGOS estimator. What we observe through simulations in this section is that this phenomena does not occur in population graphs that we consider in this study.

4.4.3 Impact of Including Ego-nodes in the Estimation

In this section, we show the effect of considering ego-nodes in the estimation by comparing ALTERS and ALL estimators. We run our experiments on synthetic graphs generated by Erdős-Rényi and Barabási-Albert graph models. Both of these models generate graphs with no assortativity in the limit of large graph size, i.e., degree of ego-nodes and alters are not correlated. Results of experiments are shown in Fig. 4.5 as empirical confidence interval curves. Interval curves contain the 5th and 95th percentile of 1000 estimations. Intervals in Barabási-Albert case are larger than those in Erdős-Rényi case. This is expected as the degree variance in graphs generated by Barabási-Albert model are higher and degree assortativity is zero in both cases. Note that when the degree assortativity is zero and the degree variance is high, this results in ego-centric graphs with high degree variance.

Intuitively, when the average degree of a population graph is small, ignoring ego vertices in the estimation is expected to have a larger effect compared to a population graph with high average degree. This is well-supported in Barabási-Albert case. Note that in Barabási-Albert graph there are large number of vertices with small degree and a few number of vertices with very high degree. Therefore, most of the sampled vertices have small degrees and hence small ego-centric graphs. Ignoring ego vertices in these small ego-centric graphs results in more information loss. Fig. 4.5 shows that when the average degree is less than 8, utilizing the ego vertices in the estimation increases the precision of the estimator. However, in Erdős-Rényi case the difference in precision is not significant since the ego-centric graphs of very small sizes are not as common compared to the Barabási-Albert case.
Figure 4.5. Effect of considering the ego-nodes in the estimation. Population graphs of size 10,000 with varying average degrees are generated using Erdős-Rényi (on the left) and Barabási-Albert (on the right) graph models. Intervals includes the 5th and 95th percentile of 1000 estimations.

4.4.4 Impact of Degree Assortativity Coefficient

Degree assortativity of a graph is a preference for its nodes to attach to others that have similar degrees. In the ego-centric sampling design, we observe the attached nodes of selected ego nodes and utilize their information in the estimator. Therefore, degree assortativity of the underlying graph seems to be an important characteristic that determines the accuracy of the estimation. One of the most prominent measures of the degree assortativity is the assortativity coefficient, $r$ [83]. The network is said to be assortative when $r > 0$, non-assortative when $r = 0$, and disassortative when $r < 0$. In ego-centric sampling, when an ego vertex is selected all attached nodes are also observed. Therefore, in an assortative network when an ego vertex with high degree is observed, we expect to observe vertices with high degrees. On the other hand, if an ego vertex with low degree is selected, we expect to observe vertices with low degrees.

In order to show the effect of the degree assortativity, we generate population graphs using Callaway graph model [14]. This is a parametrized graph model. By changing the model parameter between 0 and 1, we can generate graphs with assortativity coefficient in the range $[0, 1/3]$ and this range covers the assortativity coefficient of most social networks [83].
Figure 4.6. Effect of degree assortativity coefficient in the accuracy of the estimations. Population graph of size 10,000 is generated by a Callaway graph model with varying model parameter. Color map encodes the NRMSE values.

Fig. 4.6 shows the affect of assortativity coefficient on the estimation accuracy of ALL estimators on graphs generated by Callaway graph model with varying parameter. We consider the case in which the degree information is available as a vertex label and we utilize the same number of observations in both estimators. Note that the results for EGOS estimator are included in the figure to show the overall effect of changing the Callaway graph parameter on the estimation accuracy. As the vertices whose degrees are utilized in EGOS estimator are sampled independently and uniformly at random, a change in assortativity coefficient can not be the factor that affects the estimation accuracy. By comparing the difference in errors of ALL and EGOS estimators for a fixed sample fraction, we can observe the affect of assortativity coefficient on the estimation accuracy. In the overall, we observe only a slight difference between ALL and EGOS estimators. An important observation is that as the degree assortativity of the underlying graph increases, the error of ALL estimator decreases but the difference between EGOS and ALL estimators increases. This suggests that the efficiency of the ALL estimator decreases with respect to the EGOS estimator as the assortativity of the underlying graph increases.
4.4.5 Discussion

The proposed estimators utilize different units in the sample. One of the important points is that ALTERS, ALL, and ALL-HT estimators utilize more information than EGOS estimator in the free neighborhood case, yet their estimation accuracy is not always better than that of the EGOS estimator as shown in Enron-Email network in Fig. 4.2. This can be explained by the fact that ALTERS, ALL, and ALL-HT estimators correct the bias due to the non-uniform inclusion probability incurred by sampling alters, which makes them less efficient. Therefore, there is a trade-off between obtaining more biased samples and correcting their bias.

Uniform selection of vertices may not be directly available in some cases. In those cases, rejection sampling could be used. In rejection sampling, a valid user id is guessed by randomly generating a user id and then checking if it is valid. If user id space is sparse, most of the guesses result in query misses.

4.5 Related Work

Researchers have studied estimation of various characteristics under various sampling designs. One of the most studied sampling design is random walk (RW) sampling. For a great survey on the properties of random walks on graphs, see [73]. Variations of RW sampling has been proposed to improve the accuracy or efficiency [115, 64, 7]. RW sampling for estimating the degree distribution on directed graphs was proposed in [90]. RW sampling has also been used for estimating content density in P2P networks [100], uniformly sampling web pages from the Internet [51], uniformly sampling web pages from a search engine’s index [8]. Comparison of RW sampling and Metropolized-RW sampling is given [38]. In this study, we focus on another sampling design that has applications especially in OSNs such as LinkedIn in which RW sampling is not applicable.

There are recent studies that use the neighborhood information in the estimation [109, 22], which is closely related to our work in this study. Wang et al. [109] estimates the vertex and
edge label density under Frontier Sampling. DasGupta et al. [22] randomly samples nodes and then uses neighborhood information to improve the estimation accuracy. Kurant [64] et al. uses neighborhood information to find random walk weights but not uses it for estimation. Finally, DasGupta et al. presents a comprehensive study on estimating average degrees of large graphs in a recent study [21], but the approach requires sampling from a prescribed distribution which is not applicable under our assumptions. Our work relates to these studies as ego-centric sampling also uses neighbor information. Unlike previous studies, our focus is on comparing the accuracy of various estimators that utilize different information in the estimation under ego-centric sampling design.

4.6 Conclusion

We have proposed estimators for average degree characteristic of large graphs under ego-centric sampling design. Ego-centric sampling can be used in OSNs such as LinkedIn where other popular sampling designs (e.g., random walk sampling or its variations) are not applicable and neighbor information is fully or partially observable. Using real-world and synthetic graphs, we have compared the accuracy of these estimators under two different sampling budget constraints: free-neighborhood and non-free neighborhood. Our experimental results show that utilizing more information improves the estimation results in some cases. However, there are cases in which plug-in estimator performs better. These results support the intuition that the accuracy of an estimator is not only determined by the amount of information it utilizes but also by the interaction between the underlying graph structure, the characteristic under study, and the sampling design.
CHAPTER 5
ESTIMATING CLUSTERING COEFFICIENTS VIA
METROPOLIS-HASTINGS RANDOM WALK AND WEDGE SAMPLING
ON LARGE OSN GRAPHS

5.1 Introduction

Online social networks (OSNs) have become an important part of our lives. OSN service providers enable their users to share and follow different kinds of information such as photos, videos, news, and messages. One of the key characteristics of OSN services is that users can choose to whom to send the information as in Facebook or from whom they get the information as in Twitter. These interactions with other users create an inherent network graphs which enables researchers study various structural properties of these graphs.

One of the important properties of the OSNs is the clustering of their users. Clustering information have been used in identifying thematic structures of networks [25], spam and fraud detection [10], link classification and recommendation [103]. There are various ways to define the clustering in a graph, but the most commonly used clustering measures are 1) network average clustering coefficient and 2) global clustering coefficient (transitivity) [111]. Network average clustering is the average of the clustering coefficient of nodes in the network where the clustering coefficient of a node \( u \) is the ratio of existing links to max possible links among the neighbors of node \( u \). Transitivity is the ratio of the total number of triangles to the number of connected vertex triples (the wedges) in the network.

Exact algorithms for clustering measures are not efficient for graphs of size millions [94]. One way of overcoming this challenge is to trade-off the speed with accuracy using sampling.
OSNs usually allow to observe the friends/neighbors of their public users.\(^1\) This enables to perform a random walk (RW) on the underlying graph. Although practical challenges exist in collecting data from OSN graphs [60], it is not the focus of this study. One of the advantages of using the RW is that its statistical properties has been well-studied especially in the field of Markov Chains [73]. In graph sampling context, Markov Chain Monte Carlo (MCMC) methods are commonly used as they enable to design walks that sample the vertices/edges with the desired probabilities [54, 39]. We specifically use the Metropolis-Hastings RW (MHRW) which is an application of the Metropolis-Hastings algorithm [49], a popular MCMC method.

The edges and vertices are the most common sampling units in the context of graph sampling. However, there are alternatives such as paths [56] and wedges [95]. Seshadhri et al. [95] shows that wedge sampling is very efficient in estimating the clustering coefficients of graphs and they provide theoretical bounds on the estimation results assuming that independent vertex sampling is available.

In this study, we combine these two techniques: 1) MHRW and 2) the wedge sampling. More specifically, we sample wedges while performing a MHRW. Our main contribution is to make the efficient estimators proposed in [95] applicable for OSN graphs that allow RW sampling, but not independent sampling, using MCMC methods.

**Contributions:** We propose estimators for network average clustering coefficient and transitivity based on random walk data which is applicable under most OSNs. Our estimators do not require sampling bias correction. We conduct extensive experiments on real-world graphs to compare the performance of our estimators with the existing relevant estimators.

---

\(^1\)Some of the users may not be accessible to the third parties. Therefore, the estimation is only for publicly available part of the OSN network and we avoid to generalize the estimation results to the whole OSN network.
Figure 5.1. Example population graph.

Table 5.1. Table of Notation for the Network Model and Sampling Design.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>the number of vertices in $G$</td>
</tr>
<tr>
<td>$M$</td>
<td>the number of edges in $G$</td>
</tr>
<tr>
<td>$W$</td>
<td>the number of wedges in $G$</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>the number of triangles in $G$</td>
</tr>
<tr>
<td>$\mathcal{N}_v$</td>
<td>neighbors of vertex $v$</td>
</tr>
<tr>
<td>$d_v$</td>
<td>the degree of vertex $v$</td>
</tr>
<tr>
<td>$c_{na}$</td>
<td>network average clustering coefficient</td>
</tr>
<tr>
<td>$c_t$</td>
<td>global clustering coefficient (transitivity)</td>
</tr>
</tbody>
</table>

5.2 Definitions and Notation

5.2.1 Graph Notations

Let $G = (V, E)$ be an undirected, connected, non-bipartite graph, where $V$ and $E$ are the set of vertices and the set of edges, respectively. Let $N = |V|$ and $M = |E|$ be the number of vertices and edges, respectively. Let $\mathcal{N}_v$ and $d_v = |\mathcal{N}_v|$ be the set of neighbors and the degree of vertex $v$. The rest of the notation is obtained from study [95]. We use wedge and triangle to refer to a path of length two and the cycle of length three, respectively. For instance, in Fig. 5.1, $2 \rightarrow 6 \rightarrow 5$ is a wedge; $1 \rightarrow 2 \rightarrow 3$ is a triangle. Note that each triangle consists of three wedges. We use $W$ and $\Delta$ to denote the number of wedges and triangles in the graph.
5.2.2 Clustering Coefficient

In this section, we provide the formal definitions of the clustering coefficient measures that we use in this study: 1) network average clustering coefficient and 2) global clustering coefficient (transitivity)

Network Average Clustering Coefficient

The clustering coefficient of node $u$ is $c_u = \frac{l_u}{\left(\frac{d_u}{2}\right)}$ where $l_u$ is the number of edges among the neighbors of $u$. For instance, the clustering coefficient of node 2 and node 3 in Fig. 5.1 are $c_2 = \frac{1}{3}$ and $c_3 = \frac{2}{3}$, respectively. The clustering coefficient of nodes with degree zero or one is considered zero in this study although there is no consensus in the literature.

The network average clustering coefficient, $c_{na}$, is simply the average of clustering coefficients of all nodes in the network. Formally,

$$c_{na} = \frac{1}{N} \sum_{v \in V} c_v. \quad (5.1)$$

Global Clustering Coefficient (Transitivity)

The transitivity, $c_t$, is the relative number of triangles compared to total number of wedges. Formally,

$$c_t = \frac{3\Delta}{W}. \quad (5.2)$$

The transitivity of the network in Fig. 5.1 is 6/15.

Note that $c_{na}$ and $c_t$ can be very different for a given graph. Schank et al. provides some graph families for which $c_{na}$ and $c_t$ differs as much as possible. Therefore, one needs to carefully consider which measure is more relevant for a given context.
5.3 Background

In this section, we first discuss simple random walk (SRW) in the context of graph characteristic estimation. Then, we discuss the MHRW and how to use it to sample a graph with a given distribution. Then, we discuss the estimators for $c_t$ and $c_{na}$ under independent sampling [95]. We will use these estimators in Section 5.4 based on the sample collected by the MHRW.

5.3.1 Simple Random Walk

RW is one of the most popular sampling schemes in OSNs due to its applicability, simple usage, and well-established theory from Markov Chains. One of the important properties of RWs is the distribution of the current position in the graph if the length of the walk is infinitely large. This distribution is known as the *stationary distribution*. When the underlying graph is connected and bipartite, stationary distribution always exists and is independent of the initial position of the walk. These properties of RW make it possible to develop estimators without worrying about where to start the walk from and how to correct the bias if exists.

In *simple RW* (SRW), the next vertex in the walk is selected among the neighbors of the current vertex uniformly at random. SRW has a bias towards high degree nodes. In other words, the higher the degree of a vertex, the more that vertex is visited by the SRW.

Formally, if we start the SRW with a distribution $\pi = \{\pi_u\}_{u \in V}$ and take $k$ steps, then the distribution of the current position becomes $(P^T)^k \pi$, where $P = \{p_{u,v}\}_{u,v \in V}$ is the transition matrix of the graph. The stationary distribution is the one that satisfies the following equation:

$$\pi^* = P^T \pi^*. \quad (5.3)$$

The following lemma is a formal characterization of the vertex selection bias of a SRW.
Lemma 5. The stationary distribution for an SRW over $G$ is $\pi_{u}^{SRW} = \frac{d_u}{2M}$.

Proof. Suppose that the initial distribution is $\pi$ and we take a single step. The probability of being at vertex $u$ after this step can be computed as

$$(P^T\pi)_u = \sum_v p_{v,u}\pi_v = \sum_{v:(v,u) \in E} \frac{d_v}{2M}$$

$$= \sum_{v:(v,u) \in E} \frac{1}{2M} = \frac{d_u}{2M} = \pi_u.$$

The stationary distribution of RW depends on the selection method of the next vertex in the walk. In the next section, we discuss MHRW which an application of Metropolis-Hasting algorithm [79] that enables to obtain a walk with the desired stationary distribution.

5.3.2 MHRW

In some applications, a sample from a certain distribution might be needed. MHRW, as opposed to SRW, chooses the next vertex in the walk based on the state of the current vertex and the candidate next vertex in order to sample nodes with a desired probability, e.g., uniformly at random. More formally, MHRW is a general Markov Chain Monte Carlo (MCMC) technique to modify the stationary distribution of a walk into any given target distribution using a weight function $\omega_u$ over $V$. In order to get the target stationary distribution $\pi_u^{MH} \propto \pi_u^{SRW} \omega_u$, the following transition matrix can be used

$$p_{u,v} = \begin{cases} \frac{1}{d_u} \min(1, \frac{\omega_v}{\omega_u}) & \text{if } v \in \mathcal{N}_u \\ 1 - \sum_{k \in \mathcal{N}_u} p_{u,k} & \text{if } u = v \\ 0 & \text{otherwise}. \end{cases} \quad (5.4)$$
Note that we can obtain any target distribution $\pi_u^{MH}$ by setting

$$\omega_u \propto \frac{\pi_u^{MH}}{\pi_u^{SRW}}.$$  \hspace{1cm} (5.5)

For instance, setting $\omega_u = 1/d_u$ gives a uniform stationary distribution, i.e., $\pi_u^{MH} = 1/N$.

### 5.3.3 Wedge Sampling with Independent Vertex Selection

Seshadhri et al. [95] has proposed algorithms that sample wedges with certain probabilities to estimate $c_t$ and $c_{na}$ assuming that independent vertex selection is available. In the following lemmas, we present the unbiasedness of those estimators.

The Lemma 6 presents how to sample wedges and estimate the clustering coefficients of each sampled vertex unbiasedly. The average of these unbiased estimators for each sampled vertex clustering coefficient gives an unbiased estimator for $c_{na}$.

**Lemma 6.** Let $S = \{s_1, s_2, \ldots, s_n\}$ be the list of uniformly selected $n$ nodes possibly including duplicates. Let $I_{s_i}, s_i \in S$, denote an indicator variable for whether $s_i$ forms a triangle with its uniformly randomly selected two distinct neighbors. Then, \(\hat{c}_{na}^{indep} = \frac{1}{|S|} \sum_{i=1}^{n} I_{s_i}\) is an unbiased estimator for $c_{na}$.

**Proof.** Note that $I_s$ is an unbiased estimator of $c_s$ as a neighbor pair is selected uniformly at random among all distinct neighbor pairs. Then, due to linearity of expectation $E[\hat{c}_{na}^{indep}] = \frac{1}{N} \sum_{v \in V} E[I_v] = \frac{1}{N} \sum_{v \in V} c_v = c_{na}$. \hfill $\square$

The Lemma 7 presents how to sample wedges uniformly. Under this sampling scheme, the ratio of the number of triangles to the number of wedges gives an unbiased estimator for $c_t$.

**Lemma 7.** (Claim 3.1 [95]) Let $W_v = \binom{d_v}{2}$ denote the number of wedges centered at node $v$. Let $W = \sum_{v \in S} W_v$ be the sum of the number of wedges centered at sampled vertices. If vertices are selected with probability $p_v = \frac{W_v}{W}$ and two distinct neighbors $u$ and $w$ are selected, the sampled wedge $(\heartsuit - \bigstar - \spadesuit)$ is uniformly sampled among all wedges.
Proof. A node \( v \) is selected with probability \( p_v \), and then a distinct neighbor pair is selected with probability \( 1/\binom{d_v}{2} \). Therefore, the wedge is selected with probability \( p_v/\binom{d_v}{2} = 1/W \).

In this section, we discussed two important techniques that allow us to develop estimators for both \( c_{na} \) and \( c_t \) for graphs that allow RW sampling, but not independent sampling, such as OSN graphs:

1. MHRW that enables us to sample the nodes with the desired probabilities.

2. Two lemmas providing intuitive unbiased estimators for both \( c_{na} \) and \( c_t \) under independent sampling.

Most OSN providers do not allow direct uniform vertex sampling due to business and security concerns. Although there are statistical techniques such as rejection sampling to get independent samples, they are generally inefficient due sparse user id space [39]. As an alternative method, we propose using MHRW.

5.4 Proposed Estimators

The structure of the algorithm for estimating \( c_{na} \) and \( c_t \) is the same except the transition matrix which is determined by the weight function \( \omega \) that we introduced in Section 5.3.1. Therefore, we first provide the pseudo code of the MH-WEDGE-CLUSTERING algorithm in Algorithm 7, a Metropolis-Hastings algorithm and wedge sampling applied on clustering coefficient estimation, that takes \( \omega \) as an argument. Then, we provide the \( \omega \) required to estimate \( c_{na} \) and \( c_t \) separately

Estimating \( c_{na} \): Based on Lemma 6, we set the target stationary distribution to the uniform distribution \( \pi_{u}^{MH} = 1/N \). Using Eq. 5.5, we get

\[
\omega_u \propto \frac{\pi_u^{MH}}{\pi_u^{SRW}} = \frac{(1/N)}{(d_u/2M)} = \frac{2M}{N} \frac{1}{d_u}.
\] (5.6)
Algorithm 7 Algorithm template for estimating the clustering coefficients ($c_{na}$ and $c_t$).

1: function MH-WEDGE-CLUSTERING($G, n, init, \omega$)
2: 
3: for $i = 1$ to $n$
4: 
5: 
6: if IsTriangle(nbor1, nbor2, curr)
7: 
8: end if
9: 
10: 
11: if $r < \alpha$
12: 
13: end if
14: 
15: return triangleCount/n
16: end function

Since $M$ and $N$ are constants, we select $\omega = \{1/d_u\}_{u \in V}$. Then, using $\omega$ in MH-WEDGE-CLUSTERING algorithm, we get an asymptotically unbiased estimator for $c_{na}$.

Estimating $c_t$: Similarly, we can estimate $c_t$ using Lemma 7. Based on Lemma 7, we set the target stationary distribution to $\pi_u^{MH} = W_u/W$. Using Eq. 5.5, we get

$$\omega_u \propto \frac{\pi_u^{MH}}{\pi_u^{SRW}} = \frac{(W_u/W)}{(d_u/2M)} = \frac{M}{W} (d_u - 1). \quad (5.7)$$

Since $M$ and $W$ are constants, we select $\omega = \{d_u - 1\}_{u \in V}$. Then, using $\omega$ in MH-WEDGE-CLUSTERING algorithm, we get an asymptotically unbiased estimator for $c_t$.

Discussion: Note that we use the SRW as the base walk, i.e., uniform distribution as the proposal distribution, and Metropolis-Hastings algorithm modifies the SRW by accepting or rejecting the proposed next vertex to get the desired stationary distribution. In general, MHRW can modify any type of walk to get the desired stationary distribution, but to keep the presentation simple we did not consider alternative proposal distributions in this study.

There is an interesting difference between the walk performed by the MH-WEDGE-CLUSTERING algorithms for $c_{na}$ and $c_t$. In order to reduce the bias in the estimation, the
walk for $c_{na}$ visits high degree nodes less than SRW by rejecting to visit some higher degree neighbors. The walk for $c_t$ visits the high degree vertices more than SRW by rejecting to visit some lower degree neighbors. In the experimental evaluations, we observe that using MHRW for estimating $c_t$ improves the estimation accuracy for a given sample size, while using it for $c_{na}$ decreases the estimation accuracy when compared to using SRW for the estimation.

5.5 Experimental Results

In this section, we present our experimental results in which we compare the estimation results with most relevant state of the art algorithms to estimate the clustering coefficients. We use the following notations to denote each algorithm:
• *indep:* The algorithms proposed by Seshadhri et al. [95], which assume that independent vertex selection is available. Although that is usually not available or efficient in social networks, we present these as the ideal estimators.

• *mhrw:* Our proposed algorithms described in Section 5.4.

• *rw:* These are the algorithms proposed by Hardiman et al. [47]. The main difference is that they perform SRW with a reference to the last three visited vertices at any given time and then correct the bias after the walk.

5.5.1 Datasets

We use publicly available social network datasets in our experiments [91]. The parameters of these graphs are shown in Table 5.2. The parameters are based on the largest connected components of each network data. We remove the directions of edges if exists.

<table>
<thead>
<tr>
<th>Graph</th>
<th>M</th>
<th>N</th>
<th>$c_{na}$</th>
<th>$c_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>soc-BlogCatalog</td>
<td>2093195</td>
<td>88784</td>
<td>0.353335</td>
<td>0.059889</td>
</tr>
<tr>
<td>soc-LiveMocha</td>
<td>2193083</td>
<td>104103</td>
<td>0.054414</td>
<td>0.014079</td>
</tr>
<tr>
<td>soc-academia</td>
<td>1022882</td>
<td>200167</td>
<td>0.222479</td>
<td>0.023483</td>
</tr>
<tr>
<td>soc-delicious</td>
<td>1365961</td>
<td>536108</td>
<td>0.032175</td>
<td>0.010514</td>
</tr>
<tr>
<td>soc-epinions</td>
<td>100120</td>
<td>26588</td>
<td>0.135164</td>
<td>0.089164</td>
</tr>
<tr>
<td>soc-google-plus</td>
<td>1135669</td>
<td>201949</td>
<td>0.147862</td>
<td>0.238785</td>
</tr>
<tr>
<td>soc-gowalla</td>
<td>950327</td>
<td>196591</td>
<td>0.236724</td>
<td>0.023483</td>
</tr>
<tr>
<td>soc-themarker</td>
<td>1644800</td>
<td>69317</td>
<td>0.186630</td>
<td>0.045944</td>
</tr>
<tr>
<td>socfb-Berkeley13</td>
<td>852419</td>
<td>22900</td>
<td>0.207071</td>
<td>0.113820</td>
</tr>
<tr>
<td>socfb-Duke14</td>
<td>506437</td>
<td>9885</td>
<td>0.245542</td>
<td>0.165923</td>
</tr>
<tr>
<td>socfb-Indiana</td>
<td>1305757</td>
<td>29732</td>
<td>0.204670</td>
<td>0.135006</td>
</tr>
<tr>
<td>socfb-Texas84</td>
<td>1590651</td>
<td>36364</td>
<td>0.193727</td>
<td>0.099890</td>
</tr>
</tbody>
</table>
5.5.2 Sample Size Selection

We select a sample size of \( n \) as

\[
n = \frac{1}{\theta} \frac{(2 + \epsilon) \log(2/\alpha)}{\epsilon^2}
\]

based on the Chernoff-Hoeffding bound as shown in Theorem 8, where \( \epsilon \) and \( \alpha \) are the relative error and the confidence level, respectively.

**Theorem 8.** (Chernoff-Hoeffding Bound [52]) Let \( X_1, X_2, \ldots, X_n \) be independent 0-1 random variables with \( \Pr(X_i = 1) = p_i \). Denote \( \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i, \mu = E[\bar{X}] = \frac{1}{n} \sum_{i=1}^{n} p_i, p = \frac{\mu}{n}, \)

\[
\Pr(|\bar{X} - \mu| > \epsilon \mu) \leq 2 \exp\left(\frac{-\epsilon^2 \mu}{2 + \epsilon}\right)
\] (5.9)

We choose \( n \) in such a way that the indep estimator has the same error bound and the level of confidence in all the graphs, i.e., NRMSE of indep is the same on all graphs (see Fig. 5.2 and 5.4). In our experiments, we set \( \epsilon = 0.05 \) and \( \alpha = 0.1 \). Note that the sample size required to get the same relative error and confidence level in each graph differs as it depends on the value of the parameter of interest \( \theta \), which is either \( c_{na} \) or \( c_t \).

5.5.3 Evaluation Metrics

**Normalized Root Mean Square Error (NRMSE)**

NRMSE is a common measure used to quantify the relative error of an estimator \( \hat{\theta} \) with respect to its true value \( \theta \). It is defined as

\[
NRMSE(\hat{\theta}) = \sqrt{\frac{E[(\hat{\theta} - \theta)^2]}{\theta}}.
\] (5.10)

Empirical NRMSE of our simulations is obtained over 1000 independent runs. We also provide various statistics for the 1000 estimations of the proposed algorithms such as mean, standard deviation, minimum, maximum, and percentiles.
Running Time

The experiments were performed on the machines with the following cpu core: 2.67GHz Intel Xeon CPU X5650 with 12288 KB cache. The algorithm was implemented in python 2.7 and time.time() function was used for measuring the running time. We use random.random() and random.choice() functions to generate random numbers from a range and an element from a set, respectively. We plot the average running time over 1000 independent runs in our experimental evaluations.

The factors affecting the running times of each algorithm based on our implementation are as follows: The running times of all algorithms are inversely proportional to $c_{na}$ based on Eq. 5.8. In addition, the running times for mhrw and rw depend on the degrees of the visited vertices as the walks select random neighbors at each step using random.choice(). The time to select a random neighbor for vertex $v$ using random.choice() is proportional to $\log(d_v)$.

5.5.4 Network Average Clustering Coefficient Estimation

In this section, we compare the performance of the estimators for $c_{na}$. Fig. 5.2-a and 5.2-b shows the NRMSE and the running time of the experiments, respectively.

In Fig. 5.2-a, we observe that rw outperforms the mhrw in all graphs except the friendship graphs, i.e., soc-google-plus and facebook graphs prefixed as socfb. Poor performance of mhrw for $c_{na}$ can be due to the rejection of high degree nodes to balance the sampling probability of high degree and low degree nodes, which may cause slower convergence for the walk. Similar result has also been observed in [39, 87]. We also provide additional statistics for mhrw estimations in Table 5.3.

The comparison of indep and mhrw shows that underlying structure of the graph has significant effect on the performance MCMC methods. Although we perform an MHRW with a stationary distribution equals uniform distribution, the accuracy of mhrw might be arbitrarily worse than indep which samples nodes independently from a uniform distribution.
Table 5.3. Estimation of $c_{na}$ with $mhrw$ for $\alpha = 0.05$ and $\epsilon = 0.1$ (relative), $n \approx 5730$.

<table>
<thead>
<tr>
<th>Graph</th>
<th>$c_{na}$</th>
<th>mean</th>
<th>std</th>
<th>min</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>soc-BlogCatalog</td>
<td>0.353</td>
<td>0.366</td>
<td>0.092</td>
<td>0.033</td>
<td>0.320</td>
<td>0.364</td>
<td>0.415</td>
<td>1.000</td>
</tr>
<tr>
<td>soc-LiveMocha</td>
<td>0.054</td>
<td>0.054</td>
<td>0.017</td>
<td>0.014</td>
<td>0.045</td>
<td>0.051</td>
<td>0.059</td>
<td>0.207</td>
</tr>
<tr>
<td>soc-academia</td>
<td>0.222</td>
<td>0.223</td>
<td>0.017</td>
<td>0.169</td>
<td>0.213</td>
<td>0.223</td>
<td>0.233</td>
<td>0.299</td>
</tr>
<tr>
<td>soc-delicious</td>
<td>0.032</td>
<td>0.033</td>
<td>0.016</td>
<td>0.001</td>
<td>0.022</td>
<td>0.030</td>
<td>0.039</td>
<td>0.167</td>
</tr>
<tr>
<td>soc-epinions</td>
<td>0.135</td>
<td>0.135</td>
<td>0.018</td>
<td>0.076</td>
<td>0.124</td>
<td>0.135</td>
<td>0.145</td>
<td>0.227</td>
</tr>
<tr>
<td>soc-google-plus</td>
<td>0.148</td>
<td>0.148</td>
<td>0.028</td>
<td>0.078</td>
<td>0.129</td>
<td>0.147</td>
<td>0.165</td>
<td>0.414</td>
</tr>
<tr>
<td>soc-gowalla</td>
<td>0.236</td>
<td>0.239</td>
<td>0.038</td>
<td>0.000</td>
<td>0.226</td>
<td>0.240</td>
<td>0.252</td>
<td>0.762</td>
</tr>
<tr>
<td>soc-themarker</td>
<td>0.186</td>
<td>0.191</td>
<td>0.074</td>
<td>0.015</td>
<td>0.154</td>
<td>0.181</td>
<td>0.210</td>
<td>0.769</td>
</tr>
<tr>
<td>socfb-Berkeley13</td>
<td>0.207</td>
<td>0.207</td>
<td>0.009</td>
<td>0.146</td>
<td>0.202</td>
<td>0.207</td>
<td>0.212</td>
<td>0.244</td>
</tr>
<tr>
<td>socfb-Duke14</td>
<td>0.246</td>
<td>0.246</td>
<td>0.011</td>
<td>0.207</td>
<td>0.239</td>
<td>0.245</td>
<td>0.252</td>
<td>0.353</td>
</tr>
<tr>
<td>socfb-Indiana</td>
<td>0.205</td>
<td>0.205</td>
<td>0.007</td>
<td>0.180</td>
<td>0.200</td>
<td>0.204</td>
<td>0.209</td>
<td>0.231</td>
</tr>
<tr>
<td>socfb-Texas84</td>
<td>0.194</td>
<td>0.194</td>
<td>0.007</td>
<td>0.168</td>
<td>0.189</td>
<td>0.193</td>
<td>0.198</td>
<td>0.231</td>
</tr>
</tbody>
</table>

A similar result has been found in [75]. All facebook graphs perform similarly in terms of both NRMSE and the running time which is another observation supporting the idea that relative performance of the algorithms are highly dependent on the underlying structure of the graph.\(^2\)

It is also important to observe how fast the estimation converges to the parameter of interest as the sample size increases. Fig. 5.5 shows the convergence of the MH-WEDGE-CLUSTERING algorithm for $c_{na}$ in soc-epinions graph. The graph shows that even for small sampling fractions, the mean estimation is around the true parameter but with a larger confidence interval. These graphs also show that $mhrw$ needs more samples to get the same error and the same confidence level than $rw$ in soc-epinions graph, which also confirms the result for soc-epinions graphs in Fig. 5.2-a.

\(^2\)Comparing the additional visualizations of the graph statistics on networkrepository.com, it is easy to see that facebook graphs have similar structural properties when compared to other graphs.
5.5.5 Transitivity Estimation

In this section, we compare the performance of the estimators for \( c_t \). Fig. 5.4-a and 5.4-b shows the NRMSE and the running time of the experiments, respectively. We observe that \( mhrw \) outperforms \( rw \) in terms of NRMSE for most of the graphs. Recall that MH-WEDGE-CLUSTERING algorithm performs a walk that promotes visiting high degree vertices more than SRW for \( c_t \), we conjecture that this may cause a faster convergence to the stationary distribution.

In terms of the running time, \( rw \) outperforms both \( indep \) and \( mhrw \). Recall that running time is inversely proportional to \( c_t \) based on Eq. 5.8. The running time of \( rw \) is In addition, \( indep \) requires independent and weighted selection of nodes which we implement using a binary search on the cumulative function over the weights of vertices. \( mhrw \) visits high degree vertices more often than an SRW and the algorithm selects two random neighbors of these high degree nodes which increases the running time.\(^3\) If the limited resource is the sample size, then \( mhrw \) is preferable over \( rw \). On the other hand, if the limited resource is the running time, then \( rw \) can still be a better option.

Fig. 5.5 shows the convergence of the MH-WEDGE-CLUSTERING algorithm for \( c_t \) in soc-epinions graph. We observe that \( indep \) performs slightly better than MCMC algorithms (\( mhrw \) and \( rw \)) in terms of NRMSE in soc-epinions graph. When Fig. 5.2-a and Fig. 5.4 are compared, we also observe that this result is true for all graphs. The difference between \( indep \) and the MCMC algorithms is less. This result confirms that the studied characteristic is an important factor in the accuracy of the estimators.

\(^3\)We provide the neighbors of a node to the random.choice() function to get a random neighbor, which takes more time as the vertex degree increases.
Table 5.4. Estimation of $c_t$ with $mhrw$ for $\alpha = 0.05$ and $\epsilon = 0.1$ (relative) $n \approx 8690$.

<table>
<thead>
<tr>
<th>Graph</th>
<th>$c_t$</th>
<th>mean</th>
<th>std</th>
<th>min</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>soc-BlogCatalog</td>
<td>0.060</td>
<td>0.060</td>
<td>0.002</td>
<td>0.051</td>
<td>0.058</td>
<td>0.060</td>
<td>0.061</td>
<td>0.070</td>
</tr>
<tr>
<td>soc-LiveMocha</td>
<td>0.014</td>
<td>0.014</td>
<td>0.000</td>
<td>0.013</td>
<td>0.014</td>
<td>0.014</td>
<td>0.014</td>
<td>0.016</td>
</tr>
<tr>
<td>soc-academia</td>
<td>0.023</td>
<td>0.024</td>
<td>0.004</td>
<td>0.012</td>
<td>0.021</td>
<td>0.024</td>
<td>0.026</td>
<td>0.044</td>
</tr>
<tr>
<td>soc-delicious</td>
<td>0.011</td>
<td>0.011</td>
<td>0.002</td>
<td>0.000</td>
<td>0.009</td>
<td>0.011</td>
<td>0.012</td>
<td>0.050</td>
</tr>
<tr>
<td>soc-epinions</td>
<td>0.089</td>
<td>0.089</td>
<td>0.003</td>
<td>0.077</td>
<td>0.087</td>
<td>0.089</td>
<td>0.091</td>
<td>0.104</td>
</tr>
<tr>
<td>soc-google-plus</td>
<td>0.239</td>
<td>0.239</td>
<td>0.011</td>
<td>0.190</td>
<td>0.231</td>
<td>0.238</td>
<td>0.245</td>
<td>0.303</td>
</tr>
<tr>
<td>soc-gowalla</td>
<td>0.023</td>
<td>0.024</td>
<td>0.005</td>
<td>0.007</td>
<td>0.021</td>
<td>0.024</td>
<td>0.027</td>
<td>0.065</td>
</tr>
<tr>
<td>soc-themarker</td>
<td>0.046</td>
<td>0.046</td>
<td>0.002</td>
<td>0.039</td>
<td>0.045</td>
<td>0.046</td>
<td>0.047</td>
<td>0.054</td>
</tr>
<tr>
<td>socfb-Berkeley13</td>
<td>0.114</td>
<td>0.114</td>
<td>0.003</td>
<td>0.102</td>
<td>0.112</td>
<td>0.114</td>
<td>0.116</td>
<td>0.125</td>
</tr>
<tr>
<td>socfb-Duke14</td>
<td>0.166</td>
<td>0.166</td>
<td>0.003</td>
<td>0.155</td>
<td>0.164</td>
<td>0.166</td>
<td>0.168</td>
<td>0.178</td>
</tr>
<tr>
<td>socfb-Indiana</td>
<td>0.135</td>
<td>0.135</td>
<td>0.003</td>
<td>0.123</td>
<td>0.133</td>
<td>0.135</td>
<td>0.137</td>
<td>0.145</td>
</tr>
<tr>
<td>socfb-Texas84</td>
<td>0.100</td>
<td>0.100</td>
<td>0.003</td>
<td>0.087</td>
<td>0.098</td>
<td>0.100</td>
<td>0.102</td>
<td>0.112</td>
</tr>
</tbody>
</table>

Figure 5.4. Comparison of the $c_t$ estimators. a) NRMSE of the estimators. b) The running times of the estimators.

5.6 Discussion

There are a few issues with MCMC methods such as the correlated samples and the convergence to the stationarity [68]. We did not go into technical discussion of these issues, but analyzed their effect in the estimation through experiments on real-world graphs. Although estimation errors can be bounded easily with independent sampling using Chernoff-Hoeffding bounds or similar statistical bounds, using MCMC methods, as independent sampling is not applicable
Figure 5.5. Convergence of the MH-WEDGE-CLUSTERING for $c_t$ on soc-epinions graph. Solid black line show $c_t$ for the soc-epinions graph and the other lines are various statistics of 1000 independent estimations.

<table>
<thead>
<tr>
<th>Sampling fraction</th>
<th>Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

or very inefficient, MCMC methods makes the estimation error dependent on the structure of the graph which may not be known beforehand. This is a common issue with the MCMC methods

Although we only consider the undirected graphs in this study, OSN graphs can also be represented as directed graphs and weighted graphs. Our idea can easily be extended to the these graphs but we leave it as a future work.

### 5.7 Related Work

There are various exact algorithms for the computation of clustering measures and enumeration of triangles [101, 18, 6, 12]. However, they are generally computationally expensive or resource intensive for the graphs of size in the order of millions. Hence, fast and accurate approximation algorithms might be preferable in some scenarios. Sampling is a commonly used approach to overcome the scalability issue. Accept-reject sampling, an alternative to Metropolis-Hastings algorithm, is a popular sampling algorithm to obtain a sample from any target distribution. However, it is not used frequently in the OSN context due to its inefficiency [39]. Sparsification [103, 104, 105, 18] is a sampling technique that reduce the number of edges

---

4. independent sampling can be simulated using Accept-Reject sampling, another Monte Carlo method, event though it is not directly applicable.
probabilistically and make estimations based on the reduced graph. The wedge sampling has been used in [95, 94] for estimating clustering coefficient. However, most of these sampling-based studies focuses on the theoretical bounds on the estimation error when the independent sampling is available.

In the context of OSN graph sampling, the independent sampling is usually not available and most of the time random walks are used as an alternative sampling algorithm. Hardiman et al. [47] propose estimators for clustering coefficients using SRW, which can be considered as an alternative estimator to our proposed estimators as they have the similar assumptions about the access to the graph data.

5.8 Conclusion

In this study, we consider the estimation of two important clustering coefficient measures: 1) network average clustering coefficient, 2) global clustering coefficient (transitivity). We propose estimator for each measure by combining a Metropolis-Hastings RW and wedge sampling. We compare our estimators with the SRW-based estimator in terms of NRMSE and the running time. We also include the estimation results for a wedge-sampling based estimator with independent vertex selection as an ideal estimator although independent vertex selection is usually not applicable or efficient in OSN context. We observe that our transitivity estimator outperforms the SRW-based estimator, but SRW-based estimator outperforms our estimator for network average clustering coefficient. This result confirms that the studied characteristic is important in the accuracy of the estimator. We also observe that the structure of the studied graph affects the accuracy of the MCMC algorithms (our proposed algorithms and the SRW-based algorithms) significantly.
 CHAPTER 6
CONCLUSION

In this dissertation, we provide statistical estimators for important structural properties of large-scale Online Social Networks under various data access limitations.

First, we present a large-scale experimental study showing the accuracy of the estimators using the sample collected by popular graph sampling designs. We consider three important structural properties: clustering coefficient, degree, and path-length distribution. Our results show that a best effort approach to graph sampling within the constraints of an application domain may give inaccurate estimations. In addition, the accuracy of the estimators highly depend on the studied characteristic and the underlying structure of the graph.

Second, we study how accurately some important properties of graphs can be estimated under a limited data access model. More specifically, we consider random neighbor access (RNA) model as a rather limited model. In the RNA model, the only query available to get data from the studied graph is the random neighbor query which returns the id of a random neighbor for a given vertex id. We propose various sampling schemes and estimators for average degree and network size under the RNA model. We conduct extensive experiments on both real world OSN graphs and synthetic graphs (1) to measure the performance of the proposed estimators and (2) to identify the factors affecting the accuracy of our estimators. We find that while the average degree estimators can make accurate estimations with reasonable sample sizes under the RNA model, network size estimators require quite large sample sizes for accurate estimations.

Third, we propose estimators for the average degree characteristic of a network under ego-centric sampling. In this sampling design, we first sample a number of vertices called
ego vertices from the underlying graph and then obtain their ego-centric graph. Ego-centric
graph of a sampled vertex is defined as the subgraph induced by the vertices within 1-hop
neighborhood of the sampled ego vertex. We compare the proposed estimators with the estimator that do not utilize the neighborhood information using both real-world and synthetic large-scale graphs. The results show that utilization of the neighborhood information does not always increase the estimation accuracy depending on the sampling budget usage and the structure of the underlying graph.

Fourth, we propose estimators for popular clustering coefficient measures including (1)
network average clustering coefficient and (2) global clustering coefficient (a.k.a. transitivity).
Most studies on estimating clustering coefficients depend on independent vertex sampling
which is either unavailable or inefficient to implement in most OSN applications. Given that most OSN applications support random walk as the only practical sampling mechanism, we propose new estimators that would utilize Metropolis-Hastings RW and wedge sampling to estimate clustering coefficients in such OSN applications. We use several large scale OSN graphs to compare the accuracy of the proposed estimators with that of the existing alternatives. We also study the convergence speed of the proposed estimators. The evaluation results show that our estimator for transitivity outperforms the alternative random walk based estimator in most graphs and our estimator for network average clustering coefficient outperforms the alternatives on a subset of the used graphs (i.e., on Facebook graphs only).

In most cases, we evaluate the performance of the proposed estimators through large-scale synthetic or real-world graphs through simulations. Potential further research field that may help to develop better estimators is to develop better graph generation models for OSN networks. This will enable doing further theoretical studies to understand the concrete structural properties that affect the estimation accuracy. This may give some insights to design better or even near-optimal sampling schemes for the estimation of certain structural characteristics for large-scale OSN networks.
Once the natural phenomena is represented as a graph, the amount of the interesting queries about the structure of the graph is endless. The researchers studying graphs on different application domains might be interested in totally different structural properties of the graphs under totally different data access limitations. Therefore, it is highly likely that some structural properties of large graphs might not have received much attention so far but may become quite important for some application domains. We believe this dissertation gives some insights on the important factors to consider while developing estimators for the structural properties of large-scale networks.
APPENDIX

ERGODIC THEOREM

Theorem 9 ([85], Theorem 1.10.2). Let $C$ be a finite, irreducible, aperiodic Markov Chain and $\Delta(i,t)$ be the number of visits to vertex $i$ in first $t$ steps. Let $\tau_d = \sum_{i=1}^{N} d_i$ be the sum of the vertex degrees. Then,

$$\Delta(i,t) \overset{a.s.}{\to} td_i/\tau_d$$

(A.1)

where $\overset{a.s.}{\to}$ denotes almost sure convergence.

The main implication of the Ergodic Theorem is that the multiplicity of a vertex becomes proportional to its degree in the limit of a large sample sizes. Although this is an asymptotic result, i.e., it holds as the sample size goes to infinity, it is an important property to make use of under the very limited data access model that we consider. Note that RNA-MVCB-NETSIZE estimator needs a property that is proportional to the vertex degree and we choose this property as the vertex multiplicity based on the Ergodic Theorem.
REFERENCES


[74] Lu, J. and D. Li (2012). Bias Correction in a Small Sample from Big Data. TKDE 25(11), 2658–2663.


VITA

Emrah Cem was born in Istanbul, Turkey in 1985. He obtained his B.Sc. and M.Sc. degrees in Computer Science from Koc University in Istanbul, Turkey. He published 4 journal, 6 conference, and 2 workshop papers during his academic studies. His main research interests are large-scale network analysis, statistical estimation on networks, and big-data analysis.

He has been working as a software engineer at imo.im since February 2016. The company develops a cross-platform video/audio call, and messaging application with a global reach. He improves the scalability and the robustness of the infrastructure, and analyzes the application user data. He enjoys using big data technologies such as Spark, Hadoop, Kafka, Redis, etc., to solve large-scale problems.

Publications relevant to this thesis:


Other publications:


