NORTHWESTERN UNIVERSITY

Estimating Network Metrics via Random Walk Sampling

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Statistics

By

Minhui Zheng

EVANSTON, ILLINOIS

June 2019

ABSTRACT

In this thesis we present methods for estimating network metrics via random walk sampling. More specifically, we generalize the Hansen-Hurwitz estimator and the Horvitz-Thompson estimator to estimate the shortest path length distribution (SPLD), closeness centrality ranking, and clustering coefficients of a network. Those are important metrics to a network, but when a network is large measuring the exact value is computationally expensive. Therefore we adopt random walk sampling to collect information as we explore the network, and then provide estimations for these metrics.

Inspired by the strong ability of random walks to uncover shortest paths in a network, we first propose estimators for the shortest path length distribution (SPLD). There are two problems associated with this estimating process: 1) pairs of nodes (dyads) are sampled with unequal probabilities by a random walk, 2) the actual shortest path length (SPL) cannot by observed from the induced subgraph. To deal with the unequal selection probabilities issue, we generalize the Hansen-Hurwitz estimator and Horvitz-Thompson estimator (and their ratio forms) and apply them to the sampled dyads. Based on theory of Markov chains we prove that the selection probability of a dyad is proportional to the product of the degrees of the two nodes. To approximate the actual SPL for a dyad, we

use the observed SPL in the induced subgraph for networks with large degree variability, i.e., the standard deviation is at least two times of the mean, and we estimate the SPL using landmarks for networks with small degree variability. We find that an estimator based on a random walk with at least 20% sampling budget can achieve high estimation accuracy, and save 94% to 96% of the computational time. To the best of our knowledge, this is the first non-parametric algorithm that can estimate the SPLD of a network without knowledge of the degree distribution.

Using the same techniques to account for unequal selection probabilities and approximating SPLs between sampled nodes, we then estimate the closeness centrality of sampled nodes. But in order to estimate the closeness centrality ranking of a node, which is more of interest rather than the closeness centrality value, we introduce two more steps in the algorithm. We first apply a weighted kernel estimator to estimate the smooth population cumulative distribution function (CDF) of closeness centrality, and then compute the estimated closeness centrality rank of a node from that estimated CDF. This algorithm provides a continuous function as an estimate for the population CDF of closeness centrality and an accurate estimate for the rank of closeness centrality of each node in the network.

We finally look at the clustering coefficients of a network. The clustering coefficient of a graph measures the average probability that two neighbors of a node are themselves neighbors. People have defined the global clustering coefficient (GCC) and the local clustering coefficient (LCC). The global clustering coefficient (GCC) is the fraction of

paths of length two that are closed in the network, and the local clustering coefficient (LCC) for a single node is the fraction of pairs of neighbors of the node that are connected. The average LCC (ALCC) is the unweighted average LCC, while the GCC is equal to a weighted average of the LCCs of the nodes, where the weight is proportional to $\frac{\binom{k_i}{2}}{\sum_{j=1}^{n}\binom{k_j}{2}}$ where k_i is the degree of node i. We generalize the Hansen-Hurwitz estimator to estimate the GCC and the ALCC. By simulation studies and applications to real networks, we find that if we can observe all neighbors of a sampled node and count the exact number of connections among the neighbors, the estimators for both the GCC and the ALCC will be unbiased with small variance.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor, Prof. Bruce D. Spencer, for the continuous support of my Ph.D study and research, for his patience, motivation, and immense knowledge. He has been a tremendous mentor for me in research and a great friend in my life. His passion for research and wisdom for life have inspired me from various perspectives. His guidance has helped me in all the time of my current research and the writing of this dissertation. I truly enjoy our countless meetings and discussions, from which I learned how to become a good researcher. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my dissertation committee, Prof. Denise Scholtens and Prof. Sandy Zabell, not only for their insightful comments and encouragement, but also for the questions which incented me to improvement my research.

In addition, I would like to thank Prof. Eric D. Kolaczyk and Prof. Luis Carvalho from Boston University. Prof. Eric D. Kolaczyk has offered me countless help and supported my pursuit of Ph.D study while Prof. Luis Carvalho has opened the door of doing research for me.

I would also like to thank my fellow and friends Wenqian Wang, Mindy Hong, Mena Whalen, Yajun Liu, Yang Yu, and Dr. Jingsi Zhang for the hard work we put together in this journey and all the fun we had in the past five years. I would like to offer special thanks to my friend, Lingzi Lu, who although no longer with us, continues to inspire me by her desire for knowledge and passion for life.

Last but not least, I would like to thank my parents and my boyfriend, Lih-Chiao Hsu, for words cannot express how grateful I am to have their unlimited love and support that encouraged me to strive toward my goal.

Table of Contents

ABSTRACT 3

Acknowledgements 6

Table of Contents 8

List of Tables 13

List of Figures 15

Chapter 1. Introduction 20

1.1. Problem Statement 20

1.1.1. Estimating Shortest Path Length Distributions 20

1.1.2. Estimating Closeness Centrality Ranking 22

1.1.3. Estimating Clustering Coefficients 24

1.2. Contributions 26

Chapter 2. Background 30

2.1. Preliminary Definitions 30

2.2. Random Walk Sampling 32

2.2.1. Sampling Algorithm 32

2.3. Scale-free Networks 33

2.4. The Horvitz-Thompson Estimator and the Hansen-Hurwitz Estimator 37

Chapter 3. Related Work 40
3.1. Approximating Shortest Path Length 40
3.1.1. Ability of Random Walks to Uncover Shortest Paths 40
3.1.2. Estimating Shortest Distances by Landmarks 41
3.2. Estimating Shortest Path Length Distributions (SPLD) 44
3.2.1. The Small World Effect 45
3.2.2. Previous Estimating Methods for SPLD 46
3.2.3. Estimating SPLD in Configuration-model Networks 47
3.3. Estimating Closeness Centrality Ranking 49
3.4. Estimating Clustering Coefficients 53

Chapter 4. Estimation of Shortest Path Length Distributions 57
4.1. Overview 57
4.2. Intuition 58
4.3. Problem Definition 59
4.4. Estimating Methods 59
4.4.1. The Unweighted Estimator 60
4.4.2. The Hansen-Hurwitz Estimator 60
4.4.2.1. Inclusion Probability of a Dyad 62
4.4.3. The Horvitz-Thompson Estimator 68
4.4.4. Approximating actual SPLs between sampled nodes 73
4.5. Application of Estimating Methods 77
4.5.1. Estimation of $c.v.$ 78
4.5.2. Estimation of ψ_r 78

4.5.3. Estimation of π_r 79
4.6. Evaluation Metrics 79
4.6.1. Box plots 79
4.6.2. Mean Absolute Difference (MAD) 80
4.6.3. Root Mean Square Error (RMSE) 81
4.6.4. Kullback-Leibler Divergence (KL) 82
4.7. Simulation Study 83
4.7.1. Conditions for Random Walks to Uncover Shortest Paths 83
4.7.2. Sampling designs for Random Walks 89
4.7.2.1. Length of Random Walks for Networks with Large $c.v.$ 89
4.7.2.2. Size of Landmarks and Length of Random Walks for Networks with Small $c.v.$ 92
4.7.2.3. Number of Random Walks 100
4.7.2.4. Comparison of Estimators 101
4.7.3. Evaluation of Estimation 105
4.8. Adjusted Estimator by Weighted Average 108
4.9. Real Networks 112
4.10. Summary of Results 117
4.11. Discussion and Future Work 119

Chapter 5. Estimation of Closeness Centrality Ranking 122
5.1. Overview 122
5.2. Definitions 123
5.3. Estimating Methods 124

5.3.1. Estimating closeness centrality of sampled nodes 124
5.3.2. Estimating the population CDF of closeness centrality 128
5.3.3. Estimating closeness centrality rank of a given node 132
5.4. Evaluation Metrics 132
5.4.1. Mean Absolute Error (MAE) 133
5.4.2. Kolmogorov-Smirnov Distance (KS) 133
5.4.3. Percentage Mean Absolute Error (PMAE) 134
5.5. Simulation Study 135
5.5.1. Estimating closeness centrality of sampled nodes 135
5.5.2. Estimating the population CDF of closeness centrality 139
5.5.3. Estimating closeness centrality ranking of a given node 143
5.5.4. Length of Random Walks 147
5.6. Real networks 150
5.7. Summary of Results 155
5.8. Discussion and Future Work 156

Chapter 6. Estimation of Clustering Coefficients 158
6.1. Overview 158
6.2. Definitions 159
6.3. Estimating Method 160
6.3.1. Estimating Global Clustering Coefficient Based on Weighted Average of LCC 161
6.3.2. Estimating Global Clustering Coefficient Based on Triangles 163
6.3.2.1. Application of Bootstrap to Correct for Bias 166

6.3.3. Estimating Average Local Clustering Coefficient 168
6.4. Evaluation Metrics 169
6.4.1. Histograms 169
6.4.2. Normalized Root Mean Square Error (NRMSE) 170
6.5. Simulation Study 170
6.5.1. Influence of Estimating number of neighbors on Estimation Performance 171
6.5.2. 'Draw-by-draw' Probabilities for Triplets 173
6.5.3. Bootstrap Bias Correction 176
6.5.3.1. Estimating Bias by Bootstrap 176
6.5.3.2. Comparison of $\hat{C}_{HH2}$, $\hat{C}^{B1}_{HH2}$, and $\hat{C}^{B2}_{HH2}$ 177
6.5.4. Comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$ 179
6.5.4.1. Computational Time 179
6.5.4.2. Estimation Performance 180
6.5.5. Length of Random Walks 182
6.6. Real Networks 184
6.7. Summary of Results 192
6.8. Discussion and Future Work 192
Bibliography 195

List of Tables

4.1 Comparison of combinations of sampling budget of the random walk (β) and landmark size (γ) to achieve $RMSE \approx 0.01$ in a network with $n = 5000$ and $c.v. = 0.8$ (small $c.v.$). 99

4.2 Numerical comparison of the unweighted sample SPLD observed from the induced subgraphs (UW), the generalized Hansen-Hurwitz ratio estimates based on approximated SPL (HH.ra), and the generalized Hansen-Hurwitz ratio estimates based on actual SPL (HH.ra.l). 107

4.3 Reduction in $RMSE$ by using the adjusted estimator (AE). 112

4.4 Basic information of real networks. 114

4.5 Numerical evaluation measures of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$). 115

5.1 Numerical comparison of estimators for closeness centrality of sampled node. Left: network of size $n = 1000$ with $c.v. = 2.5$; right: network of size $n = 1000$ with $c.v. = 0.7$. 139

5.2	Numerical comparison of estimators for population CDF of closeness centrality. Left: network of size $n = 1000$ with $c.v. = 2.5$; right: network of size $n = 1000$ with $c.v. = 0.7$.	143
5.3	Numerical comparison of estimators for closeness centrality ranking: network of size $n = 1000$ with $c.v. = 2.4$.	147
5.4	Numerical comparison of estimators for closeness centrality ranking: network of size $n = 1000$ with $c.v. = 0.7$.	147
5.5	Basic information and estimation summary of real networks.	152
6.1	Numerical comparison of $\hat{C}^{AE}_{HH1}$ and $\hat{C}_{HH1}$, true global clustering coefficient $C = 0.0085$.	173
6.2	Relative bias for $\hat{\lambda}(G)$ and $\hat{\tau}(G)$.	176
6.3	Numerical comparison of $\hat{C}_{HH2}$, $\hat{C}^{B1}_{HH2}$, and $\hat{C}^{B2}_{HH2}$, true global clustering coefficient $C = 0.0085$.	179
6.4	Numerical comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$, true global clustering coefficient $C = 0.0085$.	181
6.5	Basic information of real networks.	185
6.6	Numerical comparison of $\hat{C}^{AE}_{HH1}$, $\hat{C}_{HH1}$, and $\hat{C}_{HKG}$ for GCC.	188
6.7	Numerical comparison of $\hat{C}^{AE}_{WS}$, $\hat{C}_{WS}$, and $\hat{C}_{HKL}$ for ALCC.	191

List of Figures

2.3.1	Scale-free network vs. Erdős-Rényi random graphs [18].	36
3.3.1	Plot of reverse ranking versus closeness centrality.	52
4.2.1	Illustration of a RW sample path. Green nodes: starting nodes; blue nodes: nodes visited by the random walks; purple edges: edges used by the walks to explore the graph [6].	59
4.6.1	Box plots of estimated SPLDs on the histogram of population SPLD.	80
4.7.1	Erdős-Rényi network v.s. scale-free network: distribution of difference between sample SPL and population SPL.	86
4.7.2	Networks with Gamma degree distribution: distribution of difference between sample SPL and population SPL.	88
4.7.3	Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus sampling budget (β) of the random walk in a networks with large $c.v.$, measured by MAD, $RMSE$, and KL (low values are better).	92
4.7.4	Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus size (γ) of landmarks in a network with small $c.v.$, measured by MAD, $RMSE$ and KL (low values are better).	96

4.7.5 Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus sampling budget (β) of the random walk in a network with small $c.v.$, measured by MAD, $RMSE$, and KL (low values are better). 98

4.7.6 Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus size (γ) of landmarks for different sampling budgets (β) of the random walks in a network with $n = 5000$ and $c.v. = 0.8$ (small $c.v.$). 99

4.7.7 Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus number (H) of random walks, measured by MAD, $RMSE$, and KL (low values are better). 101

4.7.8 Estimation performance versus estimators, measured by MAD, $RMSE$, and KL (low values are better). 104

4.7.9 RMSE of the generalized Hansen-Hurwitz ratio (HH.ra) estimator and the Horvitz-Thompson ratio estimator (HT.ra.s) versus sampling budget (β). 105

4.7.10 Box plots of the unweighted sample SPLD observed from the induced subgraphs (UW), the generalized Hansen-Hurwitz ratio estimates based on approximated SPL (HH.ra), and the generalized Hansen-Hurwitz ratio estimates based on actual SPL (HH.ra.l) 108

4.8.1 Box plots of the generalized Hansen-Hurwitz estimator (HHE), the configuration-model estimator based on estimated degree distribution

(CME), and the he configuration-model estimator based on actual degree distribution (CME with a.d.). 109

4.8.2 Box plots of the generalized Hansen-Hurwitz estimator (HHE), the configuration-model estimator based on estimated degree distribution (CME), and the adjusted estimator (AE) with weight found by (4.83). 111

4.9.1 Box plots of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$)(part 1). 116

4.9.2 Box plots of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$)(part 2). 117

5.5.1 Closeness centrality of sampled nodes: scatter plots of estimated values v.s. actual values. Network: $n = 1000$, $c.v. = 2.5$. 137

5.5.2 Closeness centrality of sampled nodes: scatter plots of estimated values v.s. actual values. Network: $n = 1000$, $c.v. = 0.7$. 138

5.5.3 Smoothed estimated population CDFs of closeness centrality. Network: $n = 1000$, $c.v. = 2.5$. 141

5.5.4 Smoothed estimated population CDFs of closeness centrality. Network: $n = 1000$, $c.v. = 0.7$. 142

5.5.5 Average of estimated rank with confidence interval. Network: $n = 1000$, $c.v. = 2.5$. 145

5.5.6 Average of estimated rank with confidence interval. Network: $n = 1000$, $c.v. = 0.7$. 146

5.5.7 Estimating performance versus sampling budget β (low values are better). 150

5.6.1 Average of estimated rank with confidence interval for real network (part 1). 153

5.6.2 Average of estimated rank with confidence interval for real network (part 2). 154

6.4.1 Histogram of $\hat{C}_{HH1}$, true clustering coefficient $C = 0.0085$. 170

6.5.1 Left: Plot of $\frac{e_i^*}{e_i}$ versus $\frac{k_i^*}{k_i}$; right: Histogram of $\frac{e_i^*/e_i}{k_i^*/k_i}$. 172

6.5.2 Left: histogram of $\hat{C}_{HH1}$; right: histogram of $\hat{C}_{HH1}^{AE}$, true global clustering coefficient $C = 0.0085$. 173

6.5.3 Plot of Q_w versus $k_i k_j k_v$. 174

6.5.4 A small network with 5 nodes. 175

6.5.5 Histograms of estimated bias and relative bias for $\hat{C}_{HH2}$ by bootstrap. 177

6.5.6 Histograms of $\hat{C}_{HH2}$ and adjusted $\hat{C}_{HH2}$ by bootstrap, true global clustering coefficient $C = 0.0085$. 178

6.5.7 Comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$ by bar plot for sampling fraction $\beta = 0.2, 0.3, 0.4, 0.5$, true global clustering coefficient $C = 0.0085$. 181

6.5.8 NRMSE of estimator for GCC ($\hat{C}_{HH1}$) and estimator for ALCC ($\hat{C}_{WS}$) versus sampling budget β (low values are better). 184

6.6.1 Comparison of $\hat{C}^{AE}_{HH1}$, $\hat{C}_{HH1}$, and $\hat{C}_{HKG}$ by bar plot for global clustering coefficient C. 188

6.6.2 Comparison of $\hat{C}^{AE}_{WS}$, $\hat{C}_{WS}$, and $\hat{C}_{HKL}$ by bar plot for average local clustering coefficient C_{WS}. 191

CHAPTER 1

Introduction

1.1. Problem Statement

1.1.1. Estimating Shortest Path Length Distributions

In a large network, the shortest paths between nodes are of particular importance because they are likely to provide the fastest and strongest interaction between nodes [1]. Although measures such as diameter and mean distance [2] [3] [4] have been studied extensively, the entire shortest path length distribution (SPLD) has received little attention. While the shortest path for a pair of nodes is measurable by existing algorithms such as breadth-first search, measuring the shortest paths for all pairs of nodes in a large network is computationally expensive [5].

In this dissertation, we first study the problem of estimating SPLDs in networks via random walk sampling. In particular, for each possible value of the shortest path length (SPL), we estimate the fraction of dyads with that value of SPL. There are two aspects to the problem. First, if a dyad is observed in the sample, the observed SPL in the sample may exceed the actual SPL in the population. Second, the dyads observed in a random walk sample have unequal chances of being included in the sample. With regard to the former aspect, Ribeiro, Basu, and Towsley [6] have shown that in a network with large degree variability, random walks often uncover the shortest paths. In other

words, for two nodes in a network where the variance of degree distribution is very large, the observed shortest path in the subgraph induced by a random walk sample is usually the true shortest path in the population. This property is present in scale-free networks where the degree distribution follows the power law. In this dissertation, we've shown that this property extends to networks whose degree distribution has a large coefficient of variation ($c.v.$), i.e., whose ratio of standard deviation to mean is large. On the other hand, Potamias et al. [5] have shown that in large networks, when calculating the actual distance is computationally expensive, one can use precomputed information to obtain fast estimates of the actual distance in very short time. More specifically, one can first choose a small fraction of nodes as landmarks and compute distances from every node to them. When the distance between a pair of nodes is needed, it can be estimated quickly by combining their precomputed distances to the landmarks.

With regard to dyads' unequal probabilities of being selected in the sample, we draw upon classical sampling theory for estimating totals from samples of elements included with unequal probabilities. The estimators we use are Hansen-Hurwitz estimator [7] and Horvitz-Thompson estimator [8]. Both estimators will be used in original form and ratio form to estimate the fraction of dyads with a particular value of SPL. The ratio form is defined with the numerator equal to the estimator of the number of dyads with a particular value of SPL and the denominator equal to the estimator of the total number of dyads. To develop the Hansen-Hurwitz estimator, we derive from theory of Markov chains [2] [9] [10] that the expected number of appearances of a dyad in a random walk sample with a sufficiently large number of steps is approximately proportional to the product

of the degrees of the two nodes. This result allows application of the Hansen-Hurwitz estimator to the sample including a duplicate selection of nodes. To develop the Horvitz-Thompson estimator, we approximate the random walk sampling of nodes by an adjusted multinomial sampling model in t draws, with t equal to the number of steps in the random walk. Then we apply the Horvitz-Thompson estimator to the sample excluding duplicate nodes.

1.1.2. Estimating Closeness Centrality Ranking

Closeness centrality measures how close a node is to other nodes in a given network. It gives high values for more central nodes and low values for less central ones. A node with high value of closeness centrality might have better access to information at other nodes or more direct influence on other nodes [2]. Mathematically, the closeness centrality of a node is computed as the inverse of the mean distance from the node to other nodes.

In reality, we are more interested in the relative importance of a node rather than its closeness centrality value. That is, we are more interested in the rank of a node's closeness centrality. In order to find the exact closeness centrality rank of a node, we need to first compute the closeness centrality of all nodes in the network, and then compare the closeness centrality of that node to other nodes to find its rank. The computation complexity for this process is $O(m \cdot n)$ for a network with n nodes and m edges, which can be very expensive for large networks.

The second problem we study in this dissertation is to estimate the closeness centrality ranking of a node in a network via random walk sampling. There are three stages in the estimating process: 1) estimating closeness centrality of nodes from a random walk sample; 2) estimating the population cumulative distribution function (CDF) of closeness centrality from the estimated closeness centrality of the sampled nodes; 3) for a given node, computing its estimated closeness centrality rank from the estimated population CDF of closeness centrality.

In order to estimate the closeness centrality of a sampled node, we first apply the Hansen-Hurwitz estimator to estimate the network size and the sum of geodesic distances from that node to all other nodes, and then take their ratio as an estimate for closeness centrality. The issue with this process is that it is time-consuming to measure the exact geodesic distances between sampled nodes, so we use approximations. For networks with large $c.v.$, we use the observed geodesic distances in the induced subgraph to approximate the actual geodesic distances. For networks with small $c.v.$, we use distances computed from distances to landmarks as an approximation.

There are two issues associated with the process of estimating the population CDF of closeness centrality . First, the nodes are sampled with unequal probabilities by the random walk, so the unweighted empirical distribution of the estimated closeness centrality of sampled nodes is a biased estimator for the population CDF. To solve this problem, we adopt an weighted estimator with weight proportional to the inverse of the node degree. Second, the weighted empirical distribution function is a discrete function with number

of values equal to the number of nodes in the sample, so we cannot get an accurate estimate of closeness centrality for each node in the population due to the existence of duplicate values. To smooth the empirical distribution function, we apply a weighted kernel estimator with Gaussian kernel and weight proportional to the inverse of the node degree.

1.1.3. Estimating Clustering Coefficients

The clustering coefficient, also called transitivity, of a graph measures the average probability that two neighbors of a node are themselves neighbors. Social networks tend to have large clustering coefficients. Typically the probability that two neighbors of a node are themselves neighbors is between about 10% and 60% [2] (p.262). However in many cases, the value of clustering coefficient can be sharply different from what it is expected to be in a random network where edges are formed at random between pairs of nodes with a fixed probability. Two networks can have very different clustering coefficients even if they have the same degree distribution. Measuring clustering coefficients is important and has attracted much attention since the definition was proposed in 1988. But it is computationally expensive to measure the exact value of clustering coefficient of a network. More specifically, the running time is $\mathcal{O}(n^3)$ [11] for a network with n nodes.

In this dissertation we finally study the problem of estimating clustering coefficients via random walk sampling. People have defined the global clustering coefficient (GCC) and the local clustering coefficient (LCC). The global clustering coefficient (GCC) is the

fraction of paths of length two that are closed in the network. The local clustering coefficient (LCC) for a single node is the fraction of pairs of neighbors of the node that are connected. The average LCC (ALCC) is the unweighted average LCC. The GCC is equal to a weighted average of the LCCs of the nodes, where the weight is proportional to $\frac{\binom{k_i}{2}}{\sum_{j=1}^{n}\binom{k_j}{2}}$ where k_i is the degree of node i. An equivalent definition of the GCC is the fraction of node triplets with 2 or more edges that are triangles (i.e., have 3 edges). We developed estimation methods for the GCC and the ALCC by generalizing the Hansen-Hurwitz estimator.

The two definitions of the GCC motivate different estimators. The definition in terms of weighted average LCC leads to a strategy of considering each sampled node: first going over each node, (1) counting the number of connections among its neighbors and (2) computing the total number of pairs of its neighbors, and then taking the ratio of the sum of (1) across all nodes to the sum of (2) across all nodes. We used the Hansen-Hurwitz estimator to get unbiased estimators for the numerator and denominator respectively and then take the ratio. One problem with this approach is that we need to observe all neighbors of a sampled node to count the number of connections among them, and this will increase cost. We consider using the number of connections in the induced subgraph to estimate the actual number, but this can lead to potential bias and increase standard error for the estimator.

The definition involving trianges leads to an estimator that is the ratio of the estimated number of triangles to the number of triples with at least 2 edges. The numerator and denominator are based on the Hansen-Hurwitz estimator. The problem with this approach is that unless the sample size is extremely large, i.e., the random walk is extremely long, the theoretical expected number of times a triplet is selected deviates from its empirical value and this will result in bias in the estimator. We introduce using bootstrap to correct for this bias.

Estimation of the ALCC is much simpler. We estimate the sum of LCCs and the network size, and then use their ratio to estimate the ALCC.

1.2. Contributions

This dissertation presents methods for estimating shortest path length distribution, closeness centrality ranking, and clustering coefficients of a network via random walk sampling. For a large network, it is computationally expensive to measure the exact value of these metrics. We adopt random walk sampling to collect information when exploring the network and use our proposed estimators to provide efficient and accurate estimates for these metrics. The main contributions of this thesis are listed as follows:

- *The expected number of appearances of a dyad (pair of nodes) in a sufficiently long random walk is approximately proportional to the product of the degrees of the two nodes*

When applying the Hansen-Hurwitz estimator to estimate number of dyads with each path length and to estimate the total number of dyads during the estimating process for SPLD, we need to know the expected number of appearances of a dyad in a random walk to a proportional degree. It is widely known that expected number of appearances of a node in a sufficiently long random walk is proportional to its degree. Inspired by that, we use theory of Markov chains [2] [9] and results from Anderson's [10] to prove that The expected number of appearances of a dyad (pair of nodes) in a sufficiently long random walk is approximately proportional to the product of the degrees of the two nodes.

- *Approximation of SPLs between each pair of sampled nodes*

  During the estimating process for SPLD and for closeness centrality ranking, we need to know the SPL between each pair of sampled nodes, but we can not measure the exact value of it from the subgprah induced by the random walk. Based on analytical results from Ribeiro, Basu, and Towsley [6] and Potamias et al. [5], we propose to approximate the SPL between a pair of sampled nodes by its observed SPL in the induced subgprah for networks with large $c.v.$ ($c.v. > 2$) and by the minimum of the sum of the nodes' shortest distances to the pre-selected landmarks for networks with small $c.v.$ ($c.v. < 2$). Applications to simulated networks and real networks show that this approximation is appropriate to most networks.

- *A non-parametric estimating algorithm for shortest path length distribution (SPLD)*

In order to estimate the SPLD, previous researchers have proposed various estimating algorithms, but most of them are based on the knowledge of the degree distribution. For example, Katzav et al. [1] showed two complementary analytical approaches for calculating the distribution of shortest path lengths in Erdős-Rényi networks, and Nitzan et al. [12] presented some analytical results for the DSPL between random pairs of nodes in configuration model networks. To the best of our knowledge, the estimating algorithm for SPLD proposed in this thesis is the first non-parametric algorithm that can estimate the SPLD of a network without knowledge of the degree distribution.

- *An accurate estimate of closeness centrality ranking for each node in the network*

  In the previous work for estimation of closeness centrality ranking, researchers are either only considering the top k nodes with the highest closeness centrality [13] [14] or using an sigmoid curve with a general value of slope for all networks [15]. In this thesis, we propose first using the weighted kernel estimator based on estimated closeness centrality of sampled nodes to estimate the population CDF of closeness centrality, which is different for each network, and then computing an accurate estimate for the closeness centrality ranking for each node in the network by the estimated CDF.

- *Accurate estimation of clustering coefficients*

  We generalize the usage of Hansen-Hurwitz estimator to estimate the global clustering coefficient (GCC) and the average local clustering coefficient (ALCC).

By simulation studies and applications to real networks, we find that if we can observe all neighbors of a sampled node and count the exact number of connections among the neighbors, the estimators for both the GCC and the ALCC will be unbiased with small variance.

CHAPTER 2

Background

2.1. Preliminary Definitions

Let $G = (V, E)$ be a finite graph (network), where V is the set of nodes with $|V| = n$ and S is the set of edges with $|E| = m$. Let $i \in \{1, ..., n\}$ denote a node in the graph, and $r \in \{1, ..., N\}$ denote dyad (i, j), $i, j = 1, ..., n$, $j \neq i$, in the graph, where $N = \binom{n}{2}$ is the number of dyads in the graph. An *induced subgraph* $G^* = (V^*, E^*)$ of G, is a graph formed from a subset of the nodes $V^* \subset V$ and all of the edges $E^* \subset E$ connecting pairs of nodes in V^*.

The *adjacency matrix* $\boldsymbol{A}$ [2] (p.111) of a graph is the matrix with element A_{ij} such that

$$A_{ij} = \begin{cases} 1 \text{ if there is an edge from node } i \text{ to node } j, \\ 0 \text{ otherwise.} \end{cases}$$

A graph is *undirected* if $A_{ij} = A_{ji}$ for all i and j, i.e., the adjacency matrix $\boldsymbol{A}$ is symmetric. In this paper, we only consider undirected networks without self-edges, so the adjacency matrix is symmetric and the diagonal elements are all zero.

The *degree* [2] (p.133) of node i, denoted as k_i, in a graph is the number of edges connected to it. For an undirected graph, the degree can be written in terms of the

adjacency matrix as

$$k_i = \sum_{j=1}^{n} A_{ij} = \sum_{j=1}^{n} A_{ji} \tag{2.1}$$

We define p_k to be the fraction of nodes in the network to have degree k, and the *degree distribution* to be the collection of the p_k's for $k = 0, 1, ..., n-1$. We denote $< k >$ as the first moment and $< k^2 >$ as the second moment of the degree distribution.

A *path* [2] (p.136) in a network is any sequence of nodes such that every consecutive pair of nodes in the sequence is connected by an edge in the network. A graph is *connected* if and only if there exists a path between any pair of nodes. A graph is *primitive* if $A^k > 0$ for some positive integer $k < (n-1)n^n$. In a primitive graph, a path of length k exits between every pair of nodes for some positive integer k. The *length* [2] (p.136) of a path in a network is the number of edges traversed along the path. The *shortest path* [2] (p.139), also known as *geodesic path*, is a path between two nodes such that no shorter path exists. The *diameter* L of a graph is the longest shortest path between any two nodes. Note that the diameter is finite for connected graphs.

Let $l_r = l_{ij} \in \{1, ..., L\}$ denote the true *shortest path length (SPL)*, also known as the *geodesic distance*, of dyad r in the population graph G. The *mean distance* M of a graph, is the average of shortest path lengths of all dyads in the graph. We define f_l to be the fraction of dyads in the network to have SPL l, and the *Shortest Path Length Distribution (SPLD)* to be the collection of f_l's for $l = 1, 2, ..., L$

2.2. Random Walk Sampling

Random walk sampling is a class of network sampling methods that have arisen recently and has been applied widely in large networks, due to its strong ability of 'crawling' in the network. In this dissertation, we define a single random walk $\{X_t\}$ with length t (t steps) in a given graph $G = (V, E)$ as follows:

1) Select a node u with equal probability $1/n$ from V;

2) If node u has k_u neighbors, i.e., node u has degree k_u, include one of its neighbors, say v, with equal probability $1/k_u$ into the sample;

3) In turn, conditionally independent of previous steps, one of v's neighbor nodes is selected with equal probability $1/k_v$ from the set of v's neighbors;

4) Repeat this process until the desired length t of the random walk is reached.

In the real world, some random walks are self-avoiding, in which case an edge or a node cannot be visited twice. However, in this dissertation we only consider random walks that are allowed to go along edges more than once, visit nodes more than once, or retrace their steps along an edge just traversed. In other words, we may have duplicates in our random walk sample.

2.2.1. Sampling Algorithm

For a given network $G = (V, E)$, we first take a simple random sample of H distinct nodes $U = \{u_1, ..., u_H\}$, and start a random walk from each of them. The H random walks are independent after the starting nodes. We define the sampling budget, denoted by β, $0 < \beta < 1$, to be the ratio of total steps of the H random walks to the networks

size n, and let each random walk take $B = \beta n/H$ steps.

Let $X(h) = (X_1^{(h)}, ..., X_B^{(h)}),\ h = 1, ..., H$, denote the sequence of nodes visited by the h^{th} walker. Let $V(h)$ denote the set of distinct nodes visited by the h^{th} walker, and $|V(h)|$ denote the number of nodes in set $V(h)$. Note that $|V(h)| \leq B$ as a node can be revisited during the random walk. Let $E(h)$ denote the set of edges in E that have both endpoints in $V(h)$.

Let $V^* = \bigcup_{h=1}^{h=H} V(h)$ denote the set of distinct nodes visited by the any of the H random walks, and E^* denote the set of edges in E that have both of their endpoints in V^*. Then $G^* = (V^*, E^*)$ is the induced subgraph obtained by connecting nodes in V^* using edges in E^*. The observed shortest path length between any two sampled nodes will be measured from G^*.

2.3. Scale-free Networks

Many of the research papers in graph theory concern the Erdős-Rényi random graphs. A *Erdős-Rényi random graph* $G(n, p)$ is a graph with n nodes and each edge is assigned independently to to each pair of distinct nodes with probability $p \in (0, 1)$ [16] (p.156). By this definition, the degree distribution for a Erdős-Rényi random graph follows a binomial distribution:

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}. \tag{2.2}$$

As demonstrated by Newman [2] (p.402), in the limit of large n, $G(n,p)$ has a Poisson degree distribution:

$$\lim_{n\to\infty} p_k = e^{-c}\frac{c^k}{k!}, \tag{2.3}$$

where $c = (n-1)p$ is the mean degree of $G(n,p)$. According to the property of Poisson distribution, the variance of degree distribution is always equal to the mean of degree distribution.

The model is widely studied because of its simple structure. However, recent empirical results [17] show that for many real-world networks the degree distribution significantly deviates from a Poisson distribution. In particular, for many real-world networks, the degree distribution has a power-law tail

$$p_k \propto k^{-\alpha}, \tag{2.4}$$

where α is the *exponent* of the power law. Such networks are called *scale-free*. Typically, the values in α from real networks are in range $[2,3]$, although values slightly outside this range are possible and are observed occasionally [2] (p.248).

Scale-free networks possess some unusual properties as compared to other networks. One of the nicest properties is the existence of hubs. The definition for hubs is vague in the literature. In this dissertation we define a *hub* in a network to a node whose degree is in the upper tail of the degree distribution. Intuitively, nodes with small degrees are usually connected through hubs. Therefore hubs in a network play an important role in

information exchange and shortening the shortest paths between nodes. As we will discuss later, scale-free networks have a smaller average geodesic distance than other networks. The existence of hubs is a significant difference between random networks and scale-free networks. In random networks, the expected degree is comparable for every node, and thus fewer hubs emerge.

The emergence of hubs can be explained by the growth algorithm of a scale-free network. A widely used model is the *preferential attachment* model [17]:

The network begins with an initial connected network of m_0 nodes. New nodes are added to the network one at a time. Each new node is connected to $m \leq m_0$ existing nodes with a probability that is proportional to the number of edges that the existing nodes already have. Formally, the probability that the new node is connected to node i is $\frac{k_i}{\sum_j k_j}$, where k_i is the degree of the node i and the sum is taken over all pre-existing nodes j. Numerical simulations [17] indicated that this network evolves into a scale-free network with $\alpha = 3$.

In Figure 2.3.1 below, we illustrate the comparison between scale-free networks and Erdős-Rényi random graph.

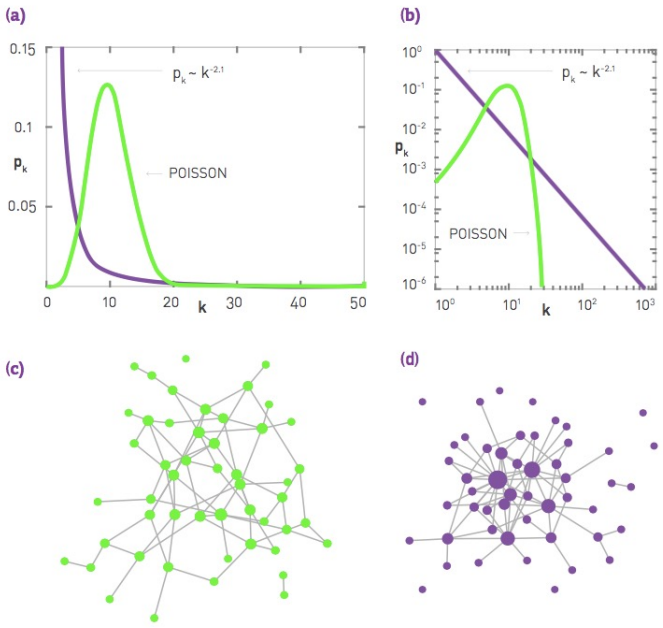


Figure 2.3.1. Scale-free network vs. Erdős-Rényi random graphs [18].

(a) Comparing a Poisson function with a power-law function ($\alpha = 2.1$) on a linear plot. Both distributions have $< k >= 11$.

(b) The same curves as in (a), but shown on a log-log plot, allowing us to inspect the difference between the two functions in the high-k regime.

(c) An Erdős-Rényi random network with $< k >= 3$ and $n = 50$, illustrating that most nodes have comparable degree around $< k >$. The variation in degrees is very small.

(d) A scale-free network with $\alpha = 2.1$ and $< k >= 3$, illustrating that numerous small-degree nodes coexist with a few highly connected hubs. The size of each node is proportional to its degree, therefore the large ones are hubs in the network.

2.4. The Horvitz-Thompson Estimator and the Hansen-Hurwitz Estimator

Suppose we have a population of elements $\{1, 2, ..., M\}$ and y_i is the characteristic of interest associated with element i, $i = 1, ..., M$. Let $t_y = \sum_{i=1}^{M} y_i$ denote the total of y_i's. In order to estimate t_y from samples of elements selected with unequal probabilities, we can use the Horvitz-Thompson estimator for samples drawn without replacement and the Hansen-Hurwitz estimator for samples drawn with replacement.

Suppose a sample of size m is drawn without replacement from the population, and the inclusion probability for element y_i is $\pi_i > 0$. Let Z_i be an indicator variable such that $Z_i = 1$ if element i is in the sample and 0 otherwise. The Horvitz-Thompson estimator [8] of the population total t_y is

$$\hat{t}_y^{HT} = \sum_{i=1}^{M} \frac{Z_i y_i}{\pi_i}, \tag{2.5}$$

with mean

$$E(\hat{t}_y^{HT}) = t_y, \tag{2.6}$$

and variance

$$Var(\hat{t}_y^{HT}) = \sum_{i=1}^{M} \sum_{k>1}^{M} (\pi_i \pi_j - \pi_i k)(\frac{y_i}{\pi_i} - \frac{y_k}{\pi_k})^2. \tag{2.7}$$

Next suppose a sample of size m is drawn with replacement in m independent draws from the population, and that on each draw the probability of selecting element y_i is β_i. Let Q_i denote the number of times element y_i selected in the sample, so that

$Q_1, ..., Q_M \sim$ multinomial$(\beta_1, ..., \beta_M; m)$, $E(Q_i) = m\beta_i$, and $\sum_{i=1}^{N} Q_i = m$.The Hansen-Hurwitz estimator [7] of the population total $t_y = \sum_{i=1}^{M} y_i$ is

$$\hat{t}_y^{HH} = \frac{1}{m} \sum_{i=1}^{M} \frac{Q_i y_i}{\beta_i}, \tag{2.8}$$

with mean

$$E(\hat{t}_y^{HH}) = t_y, \tag{2.9}$$

and variance

$$Var(\hat{t}_y^{HH}) = \frac{1}{m} \sum_{i=1}^{M} \beta_i (y_i/\beta_i - t_y)^2. \tag{2.10}$$

More generally, we will consider sample selections that could be dependent with varying selection probabilities for different draws. Thus, we define a more general form of $\hat{t}_y^H H$ as

$$\hat{t_y}^{GHH} = \sum_{i=1}^{M} \frac{Q_i y_i}{E(Q_i)}. \tag{2.11}$$

This is always unbiased for t_y as long as $E(Q_i) > 0$. The variance of $\hat{t}_y^{GHH}$ can be estimated if the sample is selected with replication.

Note that we can also estimate the total from a sample obtained by sampling with replacement by a Horvitz-Thompson estimator. If we reduce the sample obtained by sampling with replacement to a subsample by excluding the duplicates, we will get the subsample consisting of distinct elements from the population, which is analogous to a sample

obtained by sampling without replacement but with random sample size. Therefore we can apply the idea of estimating the population total by Horvitz-Thompson estimator to the subsample, provided we can calculate π_i terms.

CHAPTER 3

Related Work

3.1. Approximating Shortest Path Length

3.1.1. Ability of Random Walks to Uncover Shortest Paths

The strong ability of random walks to discover the shortest paths in networks with large degree variability was shown by Ribeiro, Basu, and Towsley [6]. They found that the ability of random walks to find shortest paths bears no relation to the paths they take, but instead relies on the large variance of the degree distribution of the network.

They proved two important results for networks with large degree variability. First, even with a relatively small number of steps, a single random walk is able to traverse a large fraction of edges. Let $< k^r >$ denote the r^{th} moment of the degree distribution. They show that for a single random walk with t steps, the number of edges discovered by the random walk is approximately $\frac{<k^2>-<k>}{<k>}t$, which is very large for networks with large variance in degree distribution. Second, two random walks cross with high probability after a small percentage of nodes have been visited. The first result indicates that the observed SPLs in the induced subgraph are very likely to be the true SPLs in the population. With a large fraction of edges visited by the random walk, the true shortest paths are very likely to be observed. The second result implies that a single random walk has the potential to explore a large area in the population network, instead of staying around

the small area close to itself. This property provides the possibility of using a single random walk to uncover the true SPLs. We will verify this property in later sections. These observations provide the possibility of using random walks to uncover shortest paths in networks with large degree variability.

Their simulation results on some real networks are also very promising. For most real-world networks they tested, more than 65% of the shortest paths observed in the sampled graph by random walk sampling are the true shortest paths in the parent graph, and more than 90% of the shortest paths observed in the sampled graph by random walk sampling are within one hop of the true shortest paths in the parent graph. The only exception is a network whose degree variability measured by $\frac{<k^2>-<k>}{<k>}t$ is much smaller than other networks.

3.1.2. Estimating Shortest Distances by Landmarks

Computing the shortest distance, i.e., the length of the shortest path between arbitrary pairs of nodes, has been a prominent problem in computer science. In an unweighted graph with n nodes and m edges, the shortest distances between one node and all other nodes can be computed by the Breadth First Search (BFS) algorithm in time $O(m+n)$ [5]. To measure the distances between all pairs of nodes, one can implement the BFS algorithm n times in time $O(n^2+mn)$, which is quadratic in the number of nodes. Therefore, in large networks, computing the exact shortest distances between all pairs of nodes

is computationally expensive. To improve the efficiency, several fast approximation algorithms have been developed recently.

Most of the approximation algorithms are landmark-based methods. They start from selecting a small set of nodes called landmarks. Then the actual distances from each landmark to all other nodes in the graph are computed by BFS and stored in memory. By using the precomputed shortest distances from the landmarks, the distance between an arbitrary pair of nodes can be computed in almost constant time. The algorithm proposed by Potamias et al. [5] is one of the landmark-based methods to quickly estimate the the length of the point to point shortest path.

Their algorithm is based on the triangle inequalities for the geodesic distance. That is, given any three nodes s, u, and t, the geodesic distances between them satisfy the following inequalities:

$$l_{st} \leq l_{su} + l_{ut}, \tag{3.1}$$

$$l_{st} \geq |l_{su} - l_{ut}|. \tag{3.2}$$

Note that if u lies on one of the shortest paths from s to t, then inequality (3.1) holds with equality.

In the pre-computing step, a set of d landmarks D are selected from the graph, and the actual distances between each landmark and all other nodes are computed by BFS. In

the estimating step, by the above inequalities, the actual geodesic distance between node s and t satisfies:

$$L \leq l_{st} \leq U, \tag{3.3}$$

where

$$L = max_{j \in D} |l_{sj} - l_{jt}|, \tag{3.4}$$

$$U = min_{i \in D} \{l_{si} + l_{it}\}. \tag{3.5}$$

By experiments, Potamias et al. [5] proposed simply using the upper bound U as an estimate to the geodesic distance. That is,

$$l_{st} \approx min_{i \in D} \{l_{si} + l_{it}\}. \tag{3.6}$$

This algorithm takes $O(d)$ time to approximate the distance between a pair of nodes and requires $O(dm + dn)$ space for the pre-computation data.

Note that the approximation will be very precise if many shortest paths pass through the landmarks. That is, the best set of landmarks consists of the most "central" nodes in the graph, and more specifically, the nodes with high betweenness centralities. In graph G, let n_{st}^i be the number of shortest paths between node s and node t passing node i, and g_{st} be the total number of shortest paths between s and t, the *betweenness centrality* of node i is defined to be $\sum_{st} \frac{n_{st}^i}{g_{st}}$. Intuitively, it measures the fraction of shortest paths

passing node i. Generally, nodes with high degrees usually have high betweenness centralities but nodes with high betweenness centralities don't always have high degrees. One example would be a graph consisting of two clusters which are connected trough a single node. The connecting node has only degree 2 but its betweenness centrality is really high.

Measuring the betweenness centrality of a node requires the information of shortest paths between all nodes in the sample, which can not be observed from the sample. As an alternative, Potamias, Bonchi, Castillo, *et al.* [5] came up with two basic strategies based on other centrality measures for selecting landmarks: (i) high degree nodes and (ii) nodes with high estimated *closeness centrality*, where the closeness centrality is the inverse of the average distance from a node to all other nodes. They defined the estimation error to be the average of $|\hat{l} - l|/l$ across all pairs of sampled nodes, where l is the actual distances and $\hat{l}$ is the approximation. Regarding to the size of the set of landmarks, they found from the application to some real networks that, with 100 landmarks, the estimation error is at less than 10% in 3 of the 5 real networks, and between 10% and 20% in the other 2 real networks.

3.2. Estimating Shortest Path Length Distributions (SPLD)

The shortest paths are of particular importance because they are likely to provide the fastest and strongest interaction between nodes in a network [1]. Up to now, measures such as the diameter and the mean distance have been studied extensively, but the entire shortest path length distribution (SPLD) has apparently attracted little attention. This distribution is of great importance as it's closely related to dynamic properties such as

velocities of network spreading processes [19]. More specifically, it plays a key role in the temporal evolution of dynamical processes on networks, such as signal propagation, navigation, and epidemic spreading [20].

3.2.1. The Small World Effect

One of the most interesting and widely studied of network phenomena is *the small world effect*: in many networks, the distances between nodes are surprisingly small. The first empirical study of this phenomenon goes back to Stanley Milgram's letter-passing experiment in the 1960s, in which he asked each of the randomly chosen "starter" individuals to try forwarding a letter to a designated "target" person living in the town of Sharon, MA, a suburb of Boston. It turned out that the letters made it to the target in a remarkably small number of steps, around six on average. Therefore, this phenomenon is also called "six degrees of separation".

With complete network data and measuring methods available these days, it is possible to measure or estimate the distances between nodes, and the small world effect has been verified explicitly. In mathematical terms, the small-world effect is the condition that the mean distance M is small. In fact, following the mathematical models, the mean distance for Erdős-Rényi random graphs was shown to scale as $\log n$ [2] (p.422).

What's more, analytical results have shown that the mean distances for scale-free networks are even smaller. Chung and Lu [3], showed that for certain families of random graphs with given expected degrees the average distance is almost surely of order

$\log n / \log \tilde{d}$. Here $\tilde{d}$ denotes the second-order average degree defined by $\tilde{d} = \frac{\sum w_i^2}{\sum w_i}$, where w_i denotes the expected degree of the i^{th} node. More specifically, for scale-free networks with $\alpha > 3$, they proved that the average distance is almost surely of order $\log n / \log \tilde{d}$. However, many Internet, social, and citation networks are scale-free networks with exponents in the range $2 < \alpha < 3$, for which the mean distance is almost surely of order $\log \log n$, but have diameter of order $\log n$ (subject some mild constraints for the average distance and maximum degree, see Chung and Lu [3] for details). This was followed by the study by Cohen and Havlin [4], who showed, using analytical argument, that the mean distance $M \sim \log \log n$ for $2 < \alpha < 3$, $M \sim \log n / \log \log n$ for $\alpha = 3$, and $M \sim \log n$ for $\alpha > 3$.

To summarize, the small world effect on scale-free networks with $2 < \alpha < 3$ yields the nice property that the mean distance and the diameter are of scale $\log \log n$ and $\log n$ respectively. For instance, a scale-free network of size $n = 10000$ has diameter only around 9. A small diameter leads to a small range of SPL, and thus it's practical to estimate the SPLD, which consists of the percentage of dyads with a particular value of SPL for each possible value of SPL.

3.2.2. Previous Estimating Methods for SPLD

Katzav et al. [1] showed two complementary analytical approaches for calculating the distribution of shortest path lengths in Erdős-Rényi networks, based on recursion equations for the shells around a reference node and for the paths originating from it. However, Erdős-Rényi graphs are not widely observed in real networks and are often only of research

interest because of their simple structure. In practice, we are more interested in a wider class of networks.

Other researchers such as Bauckhage, Kersting, and Rastegarpanah [19] have characterized shortest path histograms of networks by the Weibull distributions. Empirical tests with different graph topologies, including scale-free networks, have confirmed their theoretical prediction. However, each real network has its own parameter values of the Weibull distribution, and it is hard to find those values without full access to the network. What's more, even if we can measure the shortest distance between any pair of nodes in a network, it is very time-consuming when the network is large [5]. In chapter 4 we will propose a method for estimating the SPLD of a population graph by the sample data generated by random walks.

3.2.3. Estimating SPLD in Configuration-model Networks

Nitzan et al. [12] presented some analytical results for the DSPL between random pairs of nodes in configuration model networks. A configuration model is a method for generating random networks from given degree sequence. More specifically, in a configuration network, the degrees of all nodes are pre-defined and represented as half-links or stubs. When forming a network, two nodes are chosen uniformly at random and connected with an edge using up one of each node's stubs. This process continues until we run out of stubs and the result is a network with the pre-defined degree sequence. Nitzan et al. [12] proposed that if the degree distribution of an empirical network is known, the configuration model is ideal to be used as a null model for analyzing some network properties,

including DSPL. Therefore this is a model-based approach to estimate the DSPL. Applying this idea to our case where the degree distribution is unknown, a possible approach is to estimate the DSPL of a network based on its estimated degree distribution.

For a population network G of size n with known degree distribution p_k, according to the analytical results of Nitzan et al. [12], the tail distribution of the shortest path lengths can be expressed as

$$P(d > l) = P(d > 0) \prod_{l'}^{l} m_{n,l'}, \tag{3.7}$$

where

$$m_{n,l} = \sum_{k=1}^{n-2} p_k (\tilde{m}_{n-1,^k l-1})^k, \tag{3.8}$$

and

$$\tilde{m}_{n,l} = \sum_{k=1}^{n-2} \frac{k}{c} p_k (\tilde{m}_{n-1,^k l-1})^{k-1} \tag{3.9}$$

with c being the average degree, for $l \geq 2$, and

$$m_{n,1} = \sum_{k=1}^{n-1} p_k (1 - \frac{1}{n-1})^k, \tag{3.10}$$

and

$$\tilde{m}_{n,1} = \sum_{k=1}^{n-1} \frac{k}{c} p_k (1 - \frac{1}{n-1})^{k-1} \tag{3.11}$$

for $l = 1$.

Then the proportion of dyads with SPL l can be computed from

$$f_l = P(d > l-1) - P(d > l), \tag{3.12}$$

for $l = 1, 2, ..., n-1$.

When the network is not fully accessible and the true degree distribution is unknown, we can estimate the degree distribution p_k by the Hansen-Hurwitz ratio estimator

$$\hat{p}_k = \frac{\sum_{i \in s} x_i^k k_i^{-1}}{\sum_{i \in s} k_i^{-1}}, \tag{3.13}$$

where $x_i^k = 1$ if the degree of node i is k and zero otherwise.

If we use the estimated degree distribution $\hat{p}_k$ instead of p_k in equations (3.7) - (3.12), then we will get an configuration-model estimator (CME) from equation (3.12), denoted as $\hat{f}_l^m$. In this estimating process, the bias can come either from estimating the degree distribution or from computing the DSPL from the estimated degree distribution, but the variance only comes from estimating the degree distribution.

3.3. Estimating Closeness Centrality Ranking

Closeness centrality measures the mean distance from a node to other nodes. It gives low value for more central nodes and high value for less central ones. A node with low value of closeness centrality might have better access to information at other nodes or more direct influence on other nodes [2]. For instance, in a social network, a person with low closeness centrality might find it easier to spread their opinions to other people or receive comments from other people. The closeness centrality of a node can be computed

by taking the ratio of the network size and the sum of distances from the node to all other nodes in the network. It can be computed by breadth first search (BFS) and the the time complexity for it is $O(m)$ where m is the number of edges in the network.

In reality, we are more interested in the relative importance of a node rather than its closeness centrality value. That is, we are more interested in the rank of closeness centrality of a node. It takes two steps to compute the rank: 1) computing the closeness centrality of all nodes; 2) for a given node, comparing its closeness centrality to those of other nodes to find its rank. The first step takes $O(n \cdot m)$ time and the second step takes $O(n)$ so the time complexity for the entire process is $O(n \cdot m)$. Therefore it will be computationally expensive to find the exact rank of a node in a large network.

Due to the high complexity of computing closeness centrality of all nodes in a networks, it has attracted researchers toward several areas related to measuring the closeness centrality of a node. Kas et al. [21] and Yen, Yeh, and Chen [22] developed some methods for updating closeness centrality in dynamic networks. Other researchers such as Cohen et al. [23] and Wang [24], proposed some faster algorithms to approximate closeness centrality. Another popular area is identifying a few top nodes with highest closeness centrality. Okamoto, Chen, and Li [13] combined existing methods on calculating exact values and approximate values of closeness centrality and presented new algorithms to rank the top-k vertices with the highest closeness centrality. Ufimtsev and Bhowmick [14] presented a fast and scalable algorithm for identifying the high closeness centrality node

by group testing.

When it comes to closeness centrality ranking, Wehmuth and Ziviani [25] proposed a method called DACCER (Distributed Assessment of the Closeness CEntrality Ranking) to approximate the closeness centrality ranking. They indicated that the ranking based on local neighborhood volume computed by DACCER is highly correlated with the node ranking based on the traditional closeness centrality, in both simulated and real world networks. However, they didn't go further showing how people can use DACCER to estimated the closeness centrality ranking of a node.

The most relevant work for closeness centrality ranking is done by Saxena, Gera, and Iyengar [15]. They proposed a heuristic method to fast estimate the closeness rank of a node in $O(\alpha \cdot m)$ time complexity, where $\alpha = 3$. They observed that in real world scale-free social networks, the reverse ranking versus closeness centrality follows a sigmoid curve as shown in Figure 3.3.1. Mathematically, they found that a 4-parameter logistic equation can fit the curve well:

$$R_{rev}(u) = n + \frac{1-n}{1 + (\frac{C(u)}{c_{mid}})^p}, \tag{3.14}$$

where $R_{rev}(u)$ is the reverse ranking of node u, $C(u)$ is the closeness centrality of node u, c_mid is the closeness centrality of middle ranked node in the network, and p denotes slope of the logistic curve at the middle point. Once the reversed ranking is estimated,

the actual ranking R_{act} can be computed as

$$R_{act} = n - R_{rev} + 1. \tag{3.15}$$

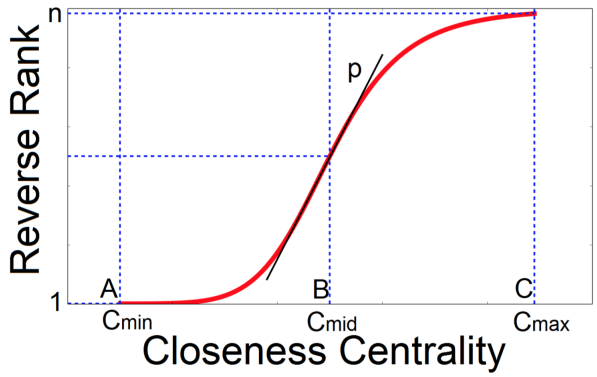


Figure 3.3.1. Plot of reverse ranking versus closeness centrality.

Due to properties of the sigmiod function, c_{mid} can be computed by $(c_{min} + c_{max})$ where c_{min} is the minimum closeness centrality and c_{max} is the maximum closeness centrality in the network. After analyzing the relationship between closeness centrality and node degree, they proposed estimating c_{max} and c_{min} by $\hat{c}_{max} = C(u)$ where u is the node with largest degree in the network and $\hat{c}_{min} = C(w)$ where w is a node chosen uniformly at random from all the nodes farthest away from u u. After measuring the slope p of the sigmoid curse for 20 real world networks, they observed that the slope ranges from 10 to 15 and that the slight variation in the estimation of p does not cause more error in the ranking, so the average of p from the real world networks can be used for estimation.

This approximation method is simple and works well for scale-free networks but still has some issues: 1) Estimating c_{mid} requires first finding the node with largest degree in the network and then finding all nodes farthest away from that node, which can be time-consuming in large networks; 2) Based on the 20 real world networks they studied, they found a range and provided a single estimation for the slope p for all network, but for other real world networks the slope can go beyond that range; 3) This sigmoid curve estimation is only verified for scale-free networks, while there are still real world networks which are not scale-free. In chapter 5 we will present an estimation method for closeness centrality ranking that will 1) only require information from a random walk sample, 2) provide different estimated curves for different networks, and 3) work for networks which are either scale-free or not scale-free.

3.4. Estimating Clustering Coefficients

Clustering coefficients serve as an important measure for network transitivity. The history of clustering coefficients goes back to 1998. Watts and Strogatz [26] first introduced *local clustering coefficient* c_i for a single node i, which is the fraction of connected neighbors of node i, and then proposed using the average of local clustering coefficients (ALCC) to measure network clustering. Later on, Newman, Watts, and Strogatz [27] proposed using *global clustering coefficient* (GCC) , which is the fraction of paths of length two that are closed in the network, as an alternative measure of network clustering. While ALCC is the unweighted average of c_i, GCC is the weighted average of c_i with weight proportional to $k_i(k_i - 1)$, where k_i is the degree of node i.

These are two different measures and can give substantially different numbers for a given network [2] (p.204). The major difference between the two measures is that GCC captures the totality of network members' experience, which may be dominated by low clustering among high degree nodes [28]. On the other hand, ALCC tends to be dominated by the nodes with low degrees, since they have small denominators in computing c_i [2] (p.204). Nowadays, people prefer using the second measurement due to its simplicity in interpretation and computation.

The running time to compute the exact clustering coefficient for a network is $\mathcal{O}(n^3)$ [11]. Therefore it is computationally expensive to run the naive computing algorithm for large networks and estimation is needed. There are two major directions of estimation: estimating with access to the entire network and estimating via sampling. In most real cases, only part of the network is accessible and information can be collected as we crawl the network by some sampling method such as random walk sampling. In this work we will focus on estimating through sampling, with random walk sampling in particular.

Schank and Wagner [29] developed an efficient algorithm to estimate GCC and ALCC via sampling. The running time is only $\mathcal{O}(1)$ for estimating ALCC and $\mathcal{O}(n)$ for GCC. However, the algorithm assumes that the adjacency matrix of the entire network is accessible and triplets can be sampled with appropriate pre-computed probabilities. As we discussed, this is not very practical in most real cases.

Some other researchers have developed estimation methods via sampling when the entire network is not accessible. In these methods, relevant information is collected as we explore nodes through random walk. Ribeiro and Towsley [30] showed that Frontier sampling, which performs m dependent random walks in the graph, can provide a better estimation performance than regular random walk in networks which are disconnected or have loosely connected components. Others like Gjoka et al. [31] and Cem and Sarac [32] have proposed using Metropolis-Hastings random walk sampling that samples nodes uniformly on large online social networks to estimate clustering coefficient. These methods require the inclusion of all neighbors of sampled nodes in order to perform estimation, which will greatly increase the sampling cost in large networks.

To make it more efficient, Katzir and Hardiman [11] proposed using random walk but without the requirement of including neighbors of sampled nodes. The only information they need in addition to the sampled nodes themselves is if there's a tie between the nodes before and after the focal nodes. They first derived unbiased estimators for both GCC and ALCC and then used simulation to show that the estimators are unbiased with small variance with only a small sampling fraction. But decreasing sampling cost will result in less information in the sample, which will potentially increase estimations error. Consider the following two cases: 1) we are able to observe all neighbors of sampled nodes and obtain the actual number of connections among the neighbors in the population graph; 2) we are not able to observe all neighbors of sampled nodes, but instead we can induce a subgraph by connecting the sampled nodes if they are connected in the population, and observe number of connections among the neighbors in the induced subgraph. Apparently,

the information in the first case in richer than the information in the second case. We will show in chapter 6 that for most real networks, the estimation performance in the first case is much better than the second case.

CHAPTER 4

Estimation of Shortest Path Length Distributions

4.1. Overview

In a network, the shortest paths between nodes are of great importance as they allow the fastest and strongest interaction between nodes. However measuring the shortest paths between all nodes in a large network is computationally expensive. In this chapter we propose a method to estimate the shortest path length (SPL) distribution of a network by random walk sampling. To deal with the unequal inclusion probabilities of dyads (pairs of nodes) in the sample, we generalize the usage of Hansen-Hurwitz estimator and Horvitz-Thompson estimator (and their ratio forms) and apply them to the sampled dyads. Based on theory of Markov chains we prove that the selection probability of a dyad is proportional to the product of the degrees of the two nodes. To approximate the actual SPL for a dyad, we use the observed SPL in the induced subgraph for networks with large degree variability, i.e., the standard deviation is at least two times of the mean, and for networks with small degree variability, estimate the SPL using landmarks for networks with small degree variability. By simulation studies and applications to real networks, we find that 1) for large networks, high estimation accuracy can be achieved by using a single random or multiple random walks with total number of steps equal to at least 20% of the nodes in the network; 2) About 94% to 96% reduction in computational time can be achieved by using sampling and approximation of SPLs between sampled nodes; 3)

the estimation performance increases as the network size increases but tends to stabilize when the network is large enough; 4) a single random walk performs as well as multiple random walks; 5) the generalized Hansen-Hurwitz ratio estimator is most preferable to use in practice due to its high estimation accuracy and easiness in computation.

4.2. Intuition

Recall that in a scale-free network, most nodes with small degrees are connected through hubs. Our approach is based on the following intuition: random walks in scale-free networks usually take steps along the shortest paths between pairs of nodes. This nice behavior is attributed to the existence of hubs.

Consider an extreme case of a network with only one hub to which all other nodes are connected. Then the random walk always goes back to the hub before moving to another node, which indeed is following the shortest path of length 2 between the nodes before and after the hub. Next consider a network with multiple hubs, but still, all other nodes are connected only to the hubs. In this case a random walk starting from any node will have to go back to the hub to which the node is connected to get to another node, which forces the random walk to travel along the shortest path for a pair of nodes.

More generally, if there are some but very few connections between nodes which are not hubs, a random walk might have the chance to traverse a path that is not the shortest path between two nodes, but the chance is small. Figure 4.2.1 shows how multiple random walks recover shortest paths in a scale-free network.

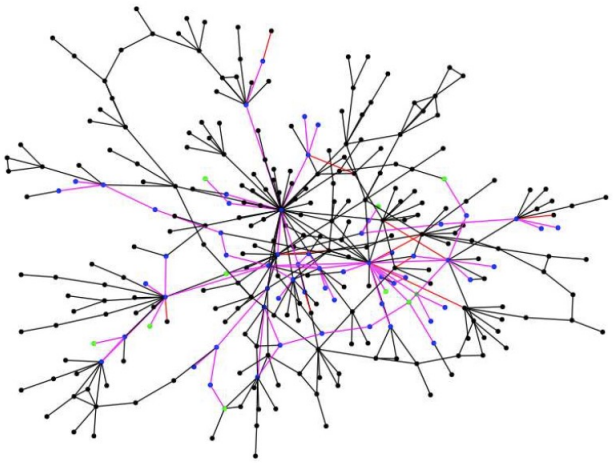


Figure 4.2.1. Illustration of a RW sample path. Green nodes: starting nodes; blue nodes: nodes visited by the random walks; purple edges: edges used by the walks to explore the graph [6].

4.3. Problem Definition

Consider a connected and undirected network $G = (V, E)$ with n nodes, m edges, and diameter L. Then the shortest path length distribution (SPLD) of G is defined as

$$f_l = \frac{N_l}{N}, l = 1, ..., L \tag{4.1}$$

where N_l is the number of dyads with SPL l, and $N = \sum_{l=1}^{L} N_l = {}_nC_2$ is the total number of dyads (pairs of nodes) in G.

4.4. Estimating Methods

In order to estimate the fraction f_l of dyads with SPL l, we need to first estimate N_l, the number of dyads with SPL l in the population graph. Let $\hat{N}_l$ denote the estimate for N_l, and f_l can be estimated by $\hat{f}_l = \frac{\hat{N}_l}{N}$. Note that sometimes we want to use a ratio

estimator $\hat{f}_l^r = \frac{\hat{N}_l}{\hat{N}}$, in which case we also estimate N, the total number of dyads in the population graph.

4.4.1. The Unweighted Estimator

A naive approach to estimate population SPLD is to simply use the SPLD of the induced subgraph G^* as an estimate. Let N_l^* denote the number of dyads with SPL l in G^*, and N^* denote the total number of dyads in G^*, the unweighted estimator for f_l is

$$\hat{f}_l^{uw} = \frac{N_l^*}{N^*},\ l = 1, ..., L. \tag{4.2}$$

However, this simple estimator may suffer from two sources of bias. First, the dyads are sampled with unequal probabilities due to the nature of random walk sampling. More specifically, dyads with shorter SPLs are more likely to be sampled than those with longer SPLs. Therefore, with the unweighted estimator, f_l for small value of l is likely to be over estimated, and f_l for large value of l is likely to be under estimated. Second, the observed SPL in G^* might be longer than the actual SPL G, and thus f_l for small value of l is likely to be under estimated, and f_l for large value of l is likely to be over estimated.

4.4.2. The Hansen-Hurwitz Estimator

Let $s = \{X(1), X(2), ..., X(H)\}$ denote the set of sequences of nodes visited by H random walks, including duplicates, and let $|s| = H \cdot B$ denote the size of s. Let $I(X_b^{(h)} = i)$ denote an indicator variable taking the value 1 if node i is visited at b^{th} step in h^{th} random walk, and zero otherwise. Let $q_i = \sum_{h=1}^{H} \sum_{b=1}^{B} I(X_b^{(h)} = i),\ i = 1, ..., n$ denote the number of times node i appears in sample s, and define $\phi_i = E(q_i)/|s|$. We assume $0 < E(q_i) < |s|$

$\forall i$, and thus $0 < \phi_i < 1\ \forall i$. Since $\sum_{i=1}^{n} q_i = |s|$, $\sum_{i=1}^{n} \phi_i = 1$. Therefore, the ϕ_i's form a probability distribution over the n nodes.

Let $r,\ r = 1, ..., N$ represent dyad $(i, j),\ i = 1, ..., n-1,\ j = i+1, ..., n$ in the population graph. Let $S = \{(X_{b_1}^{(h_1)}, X_{b_2}^{(h_2)}) : h_1, h_2 \in \{1, ..., H\}, b_1, b_2 \in \{1, ..., B\}, X_{b_1}^{(h_1)} \neq X_{b_2}^{(h_2)}\}$ denote the set of dyads whose members are any two distinct nodes in s. That os, S is the sequence of dyads visited by the H random walks, including duplicates. Define $Q_r = q_i q_j,\ i = 1, ..., n-1,\ j = i+1, ..., n$ as the number of times dyad r appears in sample S, and let $|S| = \sum_{r=1}^{N} Q_r$ denote the size of S. Notice that there may be duplicates in the sample of nodes s, but to a dyad, we only include pairs consisting of two different nodes, therefore $|S|$ is a random variable with $|S| = \binom{|s|}{2} - \sum_{i=1}^{n} \binom{q_i}{2}$. Define $\psi_r = \frac{E(Q_r)}{E(|S|)}$ and assume $0 < E(Q_r) < |S|\ \forall r$, therefore $0 < \psi_r < 1\ \forall r$. Since $\sum_{r=1}^{N} Q_r = |S|$, $\sum_{r=1}^{N} \psi_r = 1$. Therefore, the ψ_r's form a probability distribution over the N dyads.

Let $l_r \in \{1, ..., L\}$ denote the true SPL of dyad r in the population graph. Let y_r^l, $r \in \{1, ..., N\}$ and $l \in \{1, ..., L\}$, denote an indicator variable taking value $y_r^l = 1$ if $l_r = l$ and zero otherwise. Thus $N_l = \sum_{r=1}^{N} y_r^l$ is the number of dyads with SPL l in the population, and $N = \sum_{l=1}^{L} \sum_{r=1}^{N} y_r^l$ is the total number of dyads in the population.

According to 2.11, the generalized Hansen-Hurwitz estimator for N_l is

$$\hat{N}_l^{HH} = \frac{1}{|S|} \sum_{r=1}^{N} \frac{Q_r y_r^l}{\psi_r},\ l = 1, ..., L \tag{4.3}$$

The generalized Hansen-Hurwitz estimator for N is

$$\hat{N}^{HH} = \frac{1}{|S|} \sum_{r=1}^{N} \frac{Q_r}{\psi_r}, \ l = 1, ..., L \tag{4.4}$$

In order to apply (4.3) and (4.4) we need to compute or estimate ψ_r. We first recall some definitions and results for Markov chains. We call a sequence of random variables $\{X_t : t = 1, 2, ...\}$ a *discrete-time Markov chain (DTMC)* if it satisfies

$$P(X_{t+1} = i_{t+1} | X_t = i_t, X_{t-1} = i_{t-1}, .., X_1 = i_1) = P(X_{t+1} = i_{t+1} | X_t = i_t), \tag{4.5}$$

for all $t \geq 1$ and $i_1, i_2, ..., i_{t+1} \in \Omega$, where Ω is a finite or countable state space.

4.4.2.1. Inclusion Probability of a Dyad

A DTMC is *finite* if Ω is finite. A DTMC is *homogeneous* if it satisfies

$$P(X_{t+1} = j | X_t = i) = P_{i,j} \text{ for all } i, j \in \Omega, \text{ independent of } t. \tag{4.6}$$

We call the probabilities $P_{i,j}$'s the *transition probabilities*. Let $\boldsymbol{P}$ denote a matrix with element $P_{i,j}$ at its position of i^{th} row and j^{th} column. We call $\boldsymbol{P}$ the *transition matrix* for a homogeneous DTMC. Since we will only consider finite DTMCs in this paper, we denote $\Omega = \{1, 2, ..., n\}$ for simplicity.

Let $p_i(t)$ denote the probability that $\{X_t\}$ is in state i at time t, and let $\boldsymbol{p}(t) = (p_1(t), p_2(t), ..., p_n(t))^T$ denote the vector of probabilities. For a finite homogeneous DTMC

we have

$$\boldsymbol{p}^T(t+1) = \boldsymbol{p}^T(t)\boldsymbol{P}. \tag{4.7}$$

A probability vector $\boldsymbol{p} = (p_1, p_2, ..., p_n)^T$ is called a *stationary distribution* for a homogeneous DTMC with transition matrix $\boldsymbol{P}$, if it satisfies

$$\boldsymbol{p}^T = \boldsymbol{p}^T\boldsymbol{P}. \tag{4.8}$$

State j is said to be *accessible* from state i if $P^n_{i,j} > 0$ for some $n \geq 0$. If state i is accessible from state j and state j is accessible from state i, i and j are said to *communicate*. A DTMC is called *irreducible* if all of its states communicate with each other. A state i is *aperiodic* if the greatest common divisor of $\{n \geq 0 : P^n_{i,i} > 0\}$ is 1. A DTMC is called *aperiodic* if all of its states are aperiodic.

Proposition 1 [2] (p.157-159): A single random walk $\{X_t\}$ on a graph $G = (V, E)$ of size n is a finite homogeneous DTMC with a stationary distribution $\boldsymbol{p} = (\frac{k_1}{K}, ..., \frac{k_n}{K})^T$, where $K = \sum_w k_w$.

Proof: Consider a random walk $\{X_t\}$ that starts at a certain node and takes t steps. Suppose $\{X_t\}$ is at node i at time $t-1$, then the probability that it will be at node $j \neq i$ at time t is $1/k_i$, by the definition of random walk sampling in section 2.2, given that i is connected to j, i.e., $A_{ij} = 1$. That is

$$P(X_t = j | X_{t-1} = i) = \frac{A_{ij}}{k_i}. \tag{4.9}$$

Therefore, $\{X_t\}$ is a homogeneous DTMC with finite state space $\{1, 2, ..., n\}$ and transition probabilities $P_{i,j} = \frac{A_{ij}}{k_i}$. Let $\boldsymbol{P}$ denote the transition matrix of $\{X_t\}$, then $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\boldsymbol{D}$ is the diagonal matrix with elements k_i's for $i = 1, ..., n$.

Let $\boldsymbol{p} = (\frac{k_1}{K}, \frac{k_2}{K}, ..., \frac{k_n}{K})^T$, where $K = \sum_w k_w$.

(4.10)

$$\boldsymbol{p}^T\boldsymbol{D}^{-1}\boldsymbol{A} = \begin{pmatrix} \frac{k_1}{K} & \frac{k_2}{K} & \dots & \frac{k_n}{K} \end{pmatrix} \begin{pmatrix} \frac{A_{11}}{k_1} & \frac{A_{12}}{k_1} & \dots & \frac{A_{1n}}{k_1} \\ \frac{A_{21}}{k_2} & \frac{A_{22}}{k_2} & \dots & \frac{A_{2n}}{k_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{A_{n1}}{k_n} & \frac{A_{n2}}{k_n} & \dots & \frac{A_{nn}}{k_n} \end{pmatrix}$$

(4.11)
$$= \begin{pmatrix} \sum_{i=1}^n \frac{k_i}{K}\frac{A_{i1}}{k_i} & \sum_{i=1}^n \frac{k_i}{K}\frac{A_{i2}}{k_i} & \dots & \sum_{i=1}^n \frac{k_i}{K}\frac{A_{in}}{k_i} \end{pmatrix}$$

(4.12)
$$= \begin{pmatrix} \frac{1}{K}\sum_{i=1}^n A_{i1} & \frac{1}{K}\sum_{i=1}^n A_{i2} & \dots & \frac{1}{K}\sum_{i=1}^n A_{in} \end{pmatrix} = \begin{pmatrix} \frac{k_1}{K} & \frac{k_2}{K} & \dots & \frac{k_n}{K} \end{pmatrix} = \boldsymbol{p}^T$$

That is, $\boldsymbol{p}^T = \boldsymbol{p}^T\boldsymbol{P}$. Since $p_i > 0$ and $\sum_j p_j = 1$, $\boldsymbol{p}$ is a stationary distribution for $\{X_t\}$.

Proposition 2: If G is connected and has at least one triangle, the finite homogeneous DTMC $\{X_t\}$ from **Proposition 1** is irreducible and aperiodic.

Proof: Since G is connected, any node in the is accessible by any other node. That is, all states of $\{X_t\}$ communicate with other, and thus $\{X_t\}$ is irreducible. For any node in G, it can be either in a triangle or not. Suppose i is any node in a triangle, then

starting from itself, i can be reached by either 2 steps or 3 steps, that is $P^2_{i,i} > 0$ and $P^3_{i,i} > 0$. Therefore i is an aperiodic state. Consider any node j which is not in a triangle and suppose that its shortest distance to node i is l, then starting from itself, j can be reached by either $2l+2$ steps or $2l+3$ steps, that is $P^{2l+2}_{i,i} > 0$ and $P^{2l+3}_{i,i} > 0$. Therefore j is also an aperiodic state. Since all states in $\{X_t\}$ are aperiodic, $\{X_t\}$ is aperiodic.

Proposition 3: If a single random walk $\{X_t\}$ initiates from its stationary distribution $\boldsymbol{p}$ on a connected graph G with at least one triangle, then $\phi_i = E(q_i)/t = k_i/K$, and $lim_{t\to\infty}\psi_r = \alpha k_i k_j$, where $\alpha = 2[(\sum_w k_w)^2 - \sum_w k_w^2]^{-1}$, and $K = \sum_{w=1}^n k_w$.

Proof: Let $\boldsymbol{q} = (q_1, q_2, ..., q_n)^T$, where q_i = number of times node i appears in the sample, and $\boldsymbol{p} = (p_1, p_2, ..., p_n)^T$, where $p_i = \frac{k_i}{K}$. According to Anderson's (1989) results for irreducible and aperiodic Markov chains,

$$E(\boldsymbol{q}) = \boldsymbol{p}t, \tag{4.13}$$

and

$$\lim_{t\to\infty} \frac{Cov(\boldsymbol{q})}{t} = C, \tag{4.14}$$

where C is a square matrix with constant elements.

From (4.13), we have $\frac{E(q_i)}{t} = \frac{k_i}{K}$, for $i = 1, ..., n$.

In general $a_n = O(b_n)$ indicates $\lim_{t\to\infty} a_n/b_n = c$, where c is a constant, and $a_n = o(b_n)$ indicates $\lim_{t\to\infty} a_n/b_n = 0$, so we have

$$Cov(q_i, q_j) = o(t^2), \text{ and } Var(q_i) = o(t^2)\ \forall i. \tag{4.15}$$

The expected number of times dyad r appears in sample S is

$$E(Q_r) = E(q_i q_j) = E(q_i)E(q_j) + Cov(q_i, q_j) = p_i p_j t^2 + o(t^2) \tag{4.16}$$

The expected number of dyads (including duplicates) in sample S is

$$E(|S|) = \binom{t}{2} - \sum_{i=1}^{n} E(\frac{q_i(q_i-1)}{2}) \tag{4.17}$$

$$= \binom{t}{2} - \frac{1}{2}\sum_{i=1}^{n}(E(q_i^2) - E(q_i)) \tag{4.18}$$

$$= \binom{t}{2} - \frac{1}{2}\sum_{i=1}^{n}(E^2(q_i) - E(q_i) + Var(q_i)) \tag{4.19}$$

$$= \binom{t}{2} - \frac{1}{2}\sum_{i=1}^{n} tp_i(tp_i - 1) + o(t^2) \tag{4.20}$$

$$= \frac{1}{2}t(t-1) - \frac{1}{2}(t^2\sum_{i=1}^{n} p_i^2 - t) + o(t^2) \tag{4.21}$$

$$= \frac{1}{2}(1 - \sum_{i=1}^{n} p_i^2)t^2 + o(t^2) \tag{4.22}$$

In the long run, the expected fraction that dyad r appears in sample S is

$$\lim_{t\to\infty} \psi_r = \lim_{t\to\infty} \frac{E(Q_r)}{E|S|} \tag{4.23}$$

$$= \lim_{t\to\infty} \frac{2p_i p_j t^2 + o(t^2)}{(1-\sum_{i=1}^{n} p_i^2)t^2 + o(t^2)} \tag{4.24}$$

$$= \frac{2p_i p_j}{1-\sum_{i=1}^{n} p_i^2} \tag{4.25}$$

$$= \frac{2\frac{k_i k_j}{(\sum_w k_w)^2}}{1-\frac{\sum_w k_w^2}{(\sum_w k_w)^2}} \tag{4.26}$$

$$= \frac{2k_i k_j}{(\sum_w k_w)^2 - \sum_w k_w^2} \tag{4.27}$$

For simplicity we can write $lim_{t\to\infty}\psi_r = \alpha k_i k_j$, where $\alpha = 2[(\sum_w k_w)^2 - \sum_w k_w^2]^{-1}$.

Therefore, the generalized Hansen-Hurwitz estimator for N_l is

$$\hat{N}_l^{HH} = \frac{1}{|S|}\sum_{r=1}^{N} \frac{Q_r y_r^l}{\alpha k_i k_j},\ l = 1, ..., L, \tag{4.28}$$

and the generalized Hansen-Hurwitz estimator for N is

$$\hat{N}^{HH} = \frac{1}{|S|}\sum_{r=1}^{N} \frac{Q_r}{\alpha k_i k_j},\ l = 1, ..., L. \tag{4.29}$$

The generalized Hansen-Hurwitz estimator for the fraction of dyads with SPL l is

$$\hat{f}_l^{HH} = \frac{\hat{N}_l^{HH}}{N} = \frac{\sum_{r=1}^{N} \frac{Q_r y_r^l}{\alpha k_i k_j}}{|S|N},\ l = 1, ..., L, \tag{4.30}$$

and the generalized Hansen-Hurwitz ratio estimator for the fraction of dyads with SPL l is

$$\hat{f}_l^{HH.ra} = \frac{\hat{N}_l^{HH}}{\hat{N}^{HH}} = \frac{\sum_{r=1}^{N} \frac{Q_r y_r^l}{k_i k_j}}{\sum_{r=1}^{N} \frac{Q_r}{k_i k_j}},\ l = 1, ..., L \tag{4.31}$$

4.4.3. The Horvitz-Thompson Estimator

In the Hansen-Hurwitz estimator illustrated above, we take the average of all observed dyads, including duplicates, to estimate N_l and N. Alternatively, we can consider applying the Horvitz-Thompson estimator to the subsample obtained by excluding duplicate observations.

Let $s^* = V^*$ denote set of distinct nodes visited by H random walks, and $|s^*| = \sum_{h=1}^{H} |V(h)|$ denote the sample size of s^*. Since s^* is derived from s by excluding the duplicates, $|s^*|$ is a random variable depending on s. Let $z_i,\ i = 1, ..., n$ denote the number of times node i appears in sample s^*. In our case z_i is an indicator variable such that $z_i = 1$ if $i \in s^*$ and zero otherwise. Let $\tau_i = E(z_i)$ denote the inclusion probability of node i in the subsample s^*, which is indeed the probability that node i ever appears in sample s. Since $\sum_{i=1}^{n} z_i = |s^*|$, we have $\sum_{i=1}^{n} \tau_i = E(|s^*|)$.

Let S^* denote the set of all pairs of nodes in s^*, and let $|S^*|$ denote the size of S^*. Let $Z_r,\ i = 1, ..., n-1,\ j = i+1, ..., n$ denote the number of times dyad $r = (i, j)$ appears in sample S^*. In our case Z_r is an indicator variable such that $Z_r = 1$ if $r \in S^*$ and

zero otherwise. Let $\pi_r = E(Z_r)$ denote the inclusion probability of dyad r in the subsample S^*, which is indeed the probability that dyad r ever appears in sample S. Since $\sum_{r=1}^{N} Z_r = |S^*|$, we have $\sum_{r=1}^{N} \pi_r = E(|S^*|)$.

Due to the lack of knowledge about the full network $G = (V, E)$ as well as computational considerations, we will use an approximation for estimating π_r, $r \in S^*$. If a single random walk $\{X_t\}$ initiates from its stationary distribution $\boldsymbol{p}$ on a connected graph G with at least one triangle, in the long run,

$$\pi_r \approx \tau_i \tau_j, \text{ for } r = 1, 2, ..., N, \tag{4.32}$$

where

$$\tau_i = \frac{|s^*|}{\sum_{i=1}^{n} \theta_i} \theta_i \text{ for } i = 1, 2, ..., n, \tag{4.33}$$

and

$$\theta_i = 1 - (1 - \frac{k_i}{\sum_w k_w})^t \text{ for } i = 1, 2, ..., n. \tag{4.34}$$

Heuristic proof: To derive the expected number of appearances of dyads in S, we used (4.14) but did not need to use the form of the matrix C. A simple sampling model that satisfies (4.13) and (4.14) is multinomial sampling with t draws and probability $p_i = \frac{k_i}{\sum_w k_w}$ for node i to be sampled at each draw. For multinomial sampling,

$$E(q_i) = tp_i, \tag{4.35}$$

and

$$Cov(q_i, q_j) = \begin{cases} -tp_i p_j, \; i \neq j, \\ tp_i(1-p_i), \; i = j, \end{cases} \tag{4.36}$$

and hence (4.13) and (4.14) are satisfied.

Under multinomial sampling, the probability that node i is ever included in the sample by step t is

$$\theta_i = 1 - (1-p_i)^t. \tag{4.37}$$

The joint probability that i and j are both included in the sample is

$$\theta_r = \theta_{ij} = \sum_{x=1}^{t-1} P(z_i = 1 | q_j = x) P(q_j = x). \tag{4.38}$$

Note that

$$P(q_j = x) = \binom{t}{x} p_j^x (1-p_j)^{t-x}, \tag{4.39}$$

and

$$P(z_i = 1 | q_j = x) = 1 - (1 - \frac{p_i}{1-p_j})^{t-x}, \tag{4.40}$$

so

$$\theta_r = \sum_{x=1}^{t-1} \binom{t}{x} p_j^x (1-p_j)^{t-x} [1 - (1 - \frac{p_i}{1-p_j})^{t-x}] \tag{4.41}$$

$$= \sum_{x=1}^{t-1} \binom{t}{x} p_j^x (1-p_j)^{t-x} - \sum_{x=1}^{t-1} \binom{t}{x} p_j^x (1-p_i-p_j)^{t-x} \tag{4.42}$$

$$= 1 - (1-p_i j)^t - p_j^t - [(1-p_i)^t - (1-p_i-p_j)^t - p_j^t] \tag{4.43}$$

$$= 1 - (1-p_i)^t - (1-p_j)^t + (1-p_i-p_j)^t. \tag{4.44}$$

Since

$$\theta_i \theta_j = [1 - (1-p_i)^t][1 - (1-p_j)^t] \tag{4.45}$$

$$= 1 - (1-p_i)^t - (1-p_j)^t + (1-p_i-p_j+p_i p_j)^t \tag{4.46}$$

$$\approx 1 - (1-p_i)^t - (1-p_j)^t + (1-p_i-p_j)^t \text{ if } p_i p_j \text{ is negligible,} \tag{4.47}$$

and as $p_i p_j$ is verified to be negligible by simulations in this case, we can estimate θ_r by

$$\theta_r \approx \theta_i \theta_j. \tag{4.48}$$

The only problem in approximation by multinomial sampling is that we assume the draws are independent, while it is not the case in random walk sampling since a node can't be sampled twice consecutively. Therefore, θ_i under the multinomial sampling model over estimates τ_i, the inclusion probability of node i in random walk sampling. To adjust for the overestimation, we can use the one of the following two approaches to estimate τ_i,

and then estimate π_r by

$$\pi_r \approx \tau_i \tau_j. \tag{4.49}$$

Approach 1: Using the fact $\sum_{r=1}^{n} \tau_i = E(|s^*|)$ as a constraint for τ_i, we can estimate τ_i by

$$\tau_i = \frac{|s^*|}{\sum_{i=1}^{n} \theta_i} \theta_i, \tag{4.50}$$

Approach 2: Using the fact $\sum_{i \in s^*} \tau_i^{-1} = n$, we can choose the exponent $t^* < t$ for the random walking sampling such that

$$(\sum_{i \in s^*} \frac{1}{1-(1-\phi_i)^{t^*}} - n)^2 \tag{4.51}$$

is minimized, and estimate τ_i by

$$\tau_i = 1 - (1-\phi_i)^{t^*}. \tag{4.52}$$

Simulation results have shown that both (4.50) and (4.52) can provide a good estimation for τ_i.

The Horvitz-Thompson estimator for N_l is

$$\hat{N}_l^{HT} = \sum_{r=1}^{N} \frac{Z_r y_r^l}{\pi_r}, \ l = 1, ..., L, \tag{4.53}$$

and the Horvitz-Thompson estimator for N is

$$\hat{N}^{HT} = \sum_{r=1}^{N} \frac{Z_r}{\pi_r} \tag{4.54}$$

The Horvitz-Thompson estimator for the fraction of dyads with SPL l is

$$\hat{f}_l^{HT} = \frac{\hat{N}_l^{HT}}{N} = \frac{\sum_{r=1}^{N} \frac{Z_r y_r^l}{\pi_r}}{N},\ l = 1, ..., L \tag{4.55}$$

and the Horvitz-Thompson ratio estimator for the fraction of dyads with SPL l is

$$\hat{f}_l^{HT.ra} = \frac{\hat{N}_l^{HT}}{\hat{N}^{HT}} = \frac{\sum_{r=1}^{N} \frac{Z_r y_r^l}{\pi_r}}{\sum_{r=1}^{N} \frac{Z_r}{\pi_r}},\ l = 1, ..., L \tag{4.56}$$

4.4.4. Approximating actual SPLs between sampled nodes

As discussed in previous sections, in a network with n nodes and m edges, the time complexity to measure the actual distances between all pairs of nodes is $O(mn + n^2)$. This is computationally expensive for large networks. With our proposed estimators discussed above, we only need measure the distances between sampled nodes to estimate the SPLD of the population graph. Let β^* denote the ratio of number of nodes in the induced subgraph to the number of nodes in the population graph, where $0 < \beta^* \leq \beta$ and β is the sampling budget. The computation time of actual distances between all sampled nodes is $O(\beta^* mn + \beta^* n^2)$. For $\beta^* = 20\%$, only measuring the actual distances between sampled nodes will bring a 80% reduction in computation time.

However, according to some approximation methods for SPLs discussed in previous sections, we can approximate the actual SPLs between sampled nodes instead of actually measuring them. And by doing that we can achieve further reduction in computation time. In the following section we will revise the approximation methods from Ribeiro, Basu, and Towsley [6] and Potamias, Bonchi, Castillo, *et al.* [5] and apply them to our

random walk samples.

1) For networks with large *c.v.*, approximate actual SPLs by observed SPLs in the induced subgraph.

Based on theoretical and simulation results from Ribeiro, Basu, and Towsley [6], in scale-free networks, random walks have strong ability to uncover the true shortest paths, so the actual SPLs between sampled nodes can be approximated by the their observed SPLs in the subgraph induced by the random walk sample. More specifically, for a pair of sampled nodes (i, j), the actual SPL l_{ij} between them in the population graph G can be approximated by the observed SPL in the induced subgraph G^*.

More generally, it is the existence of hubs in scale-free networks that makes random walks able to find the shortest paths, as discussed in section 4.1. Therefore in this paper, we generalize the condition for random walks to uncover shortest paths to networks with relatively large variance in degree distribution, compared to the mean degree $< k >$. Let $c.v. = \frac{\sqrt{Var(k)}}{<k>} = \frac{\sqrt{<k^2>-<k>^2}}{<k>}$ denote the *coefficient of variation* of the degree distribution as a measure of the relative variance. A large $c.v.$ is needed in order for the random walks to uncover the shortest paths, and we will discuss in section 5.1 about how large the $c.v.$ needs to be.

In an induced subgraph with $\beta^* n$ nodes, the computing time for single source shortest paths is reduced to $O(\beta^* m + \beta^* n)$ by BFS within the induces subgraph. Applying BFS

to $\beta^* n$ sampled nodes in the induced subgraph, the time complexity for computing SPLs between all sampled nodes is $O(\beta^{*2} mn + \beta^{*2} n^2)$. Comparing to measuring the actual distance between sampled nodes, i.e., applying BFS to sampled nodes in the population graph, doing BFS only in the induced subgraph can save us $(1 - \beta^*) \times 100\%$ in computation time.

2) For networks with small *c.v.*, approximate actual SPLs using landmarks.

For networks with small *c.v.* in degree distribution, since random walks can't find the shortest paths in the induced subgraph, we need to implement breadth-first search (BFS) on sampled nodes in the population graph to find the shortest paths. However, based on findings by Potamias, Bonchi, Castillo, *et al.* [5], the BFS doesn't have to be applied to all sampled nodes. Instead, one can apply BFS to only a fraction of the sampled nodes to find their shortest distances to all other nodes, and use that information to estimate the shortest distances between other sampled nodes. More specifically, one can first select a set of nodes as landmarks, denoted as D, pre-compute the SPLs from landmarks to all other nodes by BFS in the population graph, and estimate the SPL between any arbitrary pair of nodes s and t by $min_{j \in D}\{l_{sj} + l_{jt}\}$. The estimation will be very precise if many shortest paths contain the selected landmarks. From their experiments, using 100 nodes with highest degrees from the population seems a fairly good strategy for choosing landmarks.

In this work, we propose selecting landmarks from the sample. This is because we are only interested in the SPLs between nodes in the sample, and landmarks from the sample will be more likely to be on the shortest paths between nodes in the sample. Also it is costly to select landmarks from the population since we need to observe the degrees of all nodes. Let γ denote the the ratio of number of landmarks to number of nodes in the induced subgraph G^*. From the sample we will choose the top $\gamma\beta^* n$ nodes in their actual degrees as landmarks. We will discuss the size of landmark set, i.e., the value of γ, in later sections.

In an induced subgraph with $\beta^* n$ nodes and $\gamma\beta^* n$ landmarks, the computing time for SPLs between a single landmark and all other nodes in the sample is still $O(m+n)$, since the BFS needs to be implemented in the population graph to compute the actual distances. Invoking the BFS $\gamma\beta^* n$ times, the computing time for SPLs between all landmarks and all other nodes in the sample is $O(\gamma\beta^* mn + \gamma\beta^* n^2)$. Comparing to measuring the actual distance between sampled nodes, i.e., applying BFS to all sampled nodes in the population graph, doing BFS only to the landmarks can save us $(1-\gamma)\times 100\%$ in computation time. This is for the pre-computing stage.

For the estimation stage, for any arbitrary pair of nodes, it only takes $\gamma\beta^* n$ time to go through the distances from these two nodes to each landmark and choose the minimum sum as the estimated SPL. Note that with BFS applied to landmarks, the distances between $\gamma\beta^* n$ landmarks and all other nodes in the sample have already been identified, therefore we just need to estimate the distances between $(1-\gamma)\beta^* n$ nodes that are not

used as landmarks. Applying $\gamma\beta^* n$ numerical search to $\binom{(1-\gamma)\beta^* n}{2} \approx \frac{1}{2}(1-\gamma)^2\beta^{*2}n^2$ pairs of nodes in the sample, the computing time for estimating distances between sampled nodes that are not landmarks is about $O(\gamma(1-\gamma)^2\beta^{*3}n^3)$ after we have the pre-computation data.

4.5. Application of Estimating Methods

In practice, sometimes we are only able to crawl part of the network, so we are restricted to observing the degrees of the sampled nodes. To apply the estimators in section 4.4 to estimating the SPLD for a network, we need to estimate ψ_r's and π_r's of the sampled nodes and $c.v$ of degree distribution by the degrees of nodes in the sample.

Following the mathematical expressions of $c.v.$, ψ_r, and π_r, we can estimate them by the estimated first moment k_1 and the second moment k_2 of the degree distribution. The estimation for k_1 and k_2 can be achieved by Hansen-Hurwitz ratio estimator. Suppose a single random walk $\{X_t\}$ initiates from its stationary distribution $\boldsymbol{p} = (\frac{k_1}{K}, \frac{k_2}{K}, ..., \frac{k_n}{K})^T$ on a connected graph G with at least one triangle such that

$$\phi_i = \frac{k_i}{K} = \frac{k_i}{nk_1}. \tag{4.57}$$

Then we can estimate the first moment k_1 by

$$\hat{k}_1 = \frac{\hat{K}}{\hat{n}} = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{k_i}{\phi_i}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{\phi_i}} = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{\frac{k_i}{k_i}}{K}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{\frac{k_i}{K}}} = \frac{|s|}{\sum_{i\in s}k_i^{-1}}. \tag{4.58}$$

Similarly, we can estimate the second moment k_2 by

$$\hat{k}_2 = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{k_i^2}{\phi_i}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{\phi_i}} = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{k_i^2}{\frac{k_i}{K}}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{\frac{k_i}{K}}} = \frac{\sum_{i\in s}k_i}{\sum_{i\in s}k_i^{-1}}. \tag{4.59}$$

4.5.1. Estimation of $c.v.$

We can estimate $c.v.$ by

$$\hat{c.v.} = \frac{\sqrt{\hat{k}_2 - (\hat{k}_1)^2}}{\hat{k}_1}. \tag{4.60}$$

4.5.2. Estimation of ψ_r

For Hansen-Hurwitz estimator, we can estimate α in $\psi_r = \alpha k_i k_j$ by

$$\hat{\alpha} = \frac{2}{(n\hat{k}_1)^2 - n\hat{k}_2}, \tag{4.61}$$

and can therefore estimate ψ_r by

$$\hat{\psi}_r = \frac{2}{(n\hat{k}_1)^2 - n\hat{k}_2}k_i k_j. \tag{4.62}$$

Note that for Hansen-Hurwitz ratio estimator (4.31), we can just plug in the observed degrees k_i and k_j of sampled nodes, and don't need to estimate any selection probabilities.

4.5.3. Estimation of π_r

For Horvitz-Thompson estimator, we can estimate τ_i identified in (6.16) by

$$\hat{\tau}_i = \frac{|s^*|}{n\hat{\bar{\theta}}}\hat{\theta}_i, \tag{4.63}$$

where

$$\hat{\theta}_i = 1 - (1 - \frac{k_i}{n\hat{k}_1})^t \tag{4.64}$$

and

$$\hat{\bar{\theta}} = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{\hat{\theta}_i}{\phi_i}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{\phi_i}} = \frac{\frac{1}{|s|}\sum_{i\in s}\frac{\hat{\theta}_i}{k_i/K}}{\frac{1}{|s|}\sum_{i\in s}\frac{1}{k_i/K}} = \frac{\sum_{i\in s}\frac{\hat{\theta}_i}{k_i}}{\sum_{i\in s}\frac{1}{k_i}}. \tag{4.65}$$

Consequently, we can estimate π_r by

$$\hat{\pi}_r = \hat{\tau}_i\hat{\tau}_j. \tag{4.66}$$

4.6. Evaluation Metrics

To evaluate the performance of an estimator, we take K random walk samples from the population graph G, compute the estimate from each sample, and then apply the following four evaluating metrics to get an overall assessment for the estimator.

4.6.1. Box plots

We first plot the histogram of the population SPLD. For each value of the population SPL, we place a box plot of sample estimates on the corresponding position of the histogram.

Figure 4.6.1 is an example of box plots of Hansen-Hurwitz ratio estimates based on 100 samples taken from a scale-free network of size 1000. For each sample, a single random walk of 200 steps is used to produce the induced subgraph for the sample SPL to be observed.

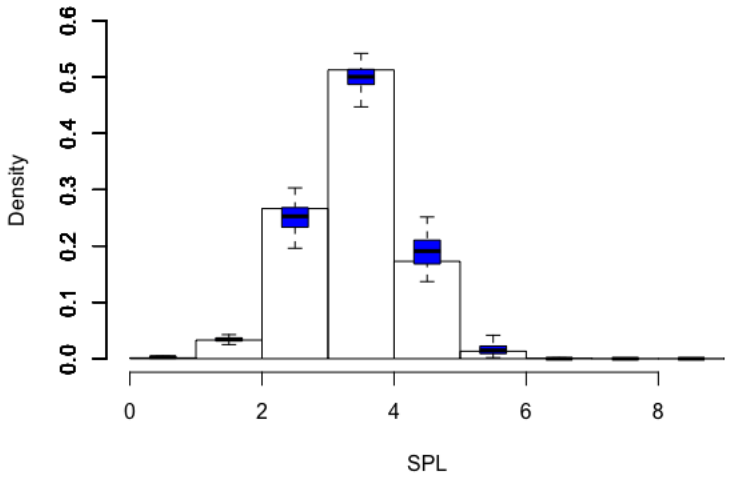


Figure 4.6.1. Box plots of estimated SPLDs on the histogram of population SPLD.

4.6.2. Mean Absolute Difference (MAD)

For each value of population SPL l, the Mean Absolute Difference (MAD) for the estimated fraction $\hat{P}(l)$ is

$$mad(l) = E(|\hat{P}(l) - P(l)|). \tag{4.67}$$

The empirical MAD for SPL l from K samples is

$$MAD(l) = \frac{1}{K}\sum_{k} |\hat{P}_k(l) - P(l)|, \tag{4.68}$$

with estimated variance

$$\hat{Var}(MAD(l)) = \frac{1}{K}\frac{\sum_k (|\hat{P}_k(l) - P(l)| - MAD(l))^2}{K-1}. \tag{4.69}$$

Averaging all possible values of population SPL, the MAD for the estimated SPLD $\hat{P}$ is

$$MAD = \frac{1}{L}\sum_{l} MAD(l), \tag{4.70}$$

with estimated standard error

$$\hat{se}(MAD) = \frac{1}{L}\sqrt{\sum_{l} \hat{Var}(MAD(l))} \tag{4.71}$$

4.6.3. Root Mean Square Error (RMSE)

For each value of population SPL l, the Root Mean Square Error (RMSE) for the estimated fraction $\hat{P}(l)$ is

$$rmse(l) = \sqrt{E[(\hat{P}(l) - P(l))^2]}. \tag{4.72}$$

The empirical RMSE for SPL l from K samples is

$$RMSE(l) = \sqrt{\frac{1}{K}\sum_{k} (\hat{P}_k(l) - P(l))^2}, \tag{4.73}$$

with estimated variance

$$\hat{Var}(RMSE(l)) = \frac{1}{K}\frac{\sum_k(\sqrt{(\hat{P}_k(l) - P(l))^2} - RMSE(l))^2}{K-1}. \tag{4.74}$$

Averaging all possible values of population SPL, the RMSE for the estimated SPLD $\hat{P}$ is

$$RMSE = \frac{1}{L}\sum_l RMSE(l), \tag{4.75}$$

with estimated standard error

$$\hat{se}(RMSE) = \frac{1}{L}\sqrt{\sum_l \hat{Var}(RMSE(l))}. \tag{4.76}$$

4.6.4. Kullback-Leibler Divergence (KL)

To measure the difference between two discrete distributions: estimated SPLD $\hat{P}_k$ from the k^{th} sample, and population SPLD P, we can use the symmetrised Kullback-Leibler divergence:

$$KL(k) = \sum_l \hat{P}_k(l) log\frac{\hat{P}_k(l)}{P(l)} + \sum_l P(l) log\frac{P(l)}{\hat{P}_k(l)}. \tag{4.77}$$

The average Kullback-Leibler divergence over all K samples is

$$KL = \frac{1}{K}\sum_k KL(k), \tag{4.78}$$

with estimated standard error

$$\hat{se}(KL) = \sqrt{\frac{1}{K}\frac{\sum_k (KL(k) - KL)^2}{K-1}} \tag{4.79}$$

In practice, since the values of KL are much almost ten times as large as the values of MAD and $RMSE$, we will use $KL/10$ to keep the three numerical measures in the same scale.

4.7. Simulation Study

In this section, we present several simulation studies to assess the performance of the methods we proposed in Section 4. More specifically, by using the evaluation techniques discussed in section 4.6, we 1) test on different values of $c.v.$ of degree distribution to explore the conditions for random walks to uncover shortest paths; 2) test on various lengths and numbers of random walks and different estimators to find the best sampling design; 3) compare our estimates based on approximated SPLs to the unweighted sample SPLDs and estimates based on actual SPLs to evaluate the estimation performance.

4.7.1. Conditions for Random Walks to Uncover Shortest Paths

In Section 4.4.4, we generalized the condition for random walks to uncover shortest paths to having a large $c.v.$ of degree distribution. In this section, we will first verify the strong ability of random walks from scale-free networks in uncovering shortest paths. And based on that, we will explore the range of $c.v.$ which allows the random walks to perform well in uncovering shortest paths in general networks. To assess the performance, we will look at the proportion of shortest paths uncovered by the random walk sample. We will use

networks with gamma degree distributions as an example of general networks.

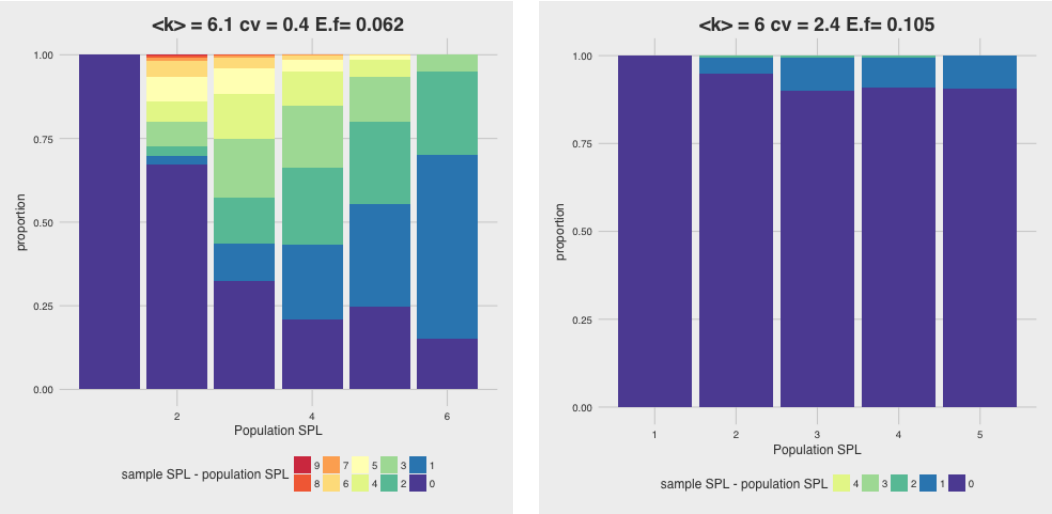
In addition, as discussed by Ribeiro, Basu, and Towsley [6], in networks with large degree variability, the fraction of edges with at least one its endpoints visited by the random walk is large. In this paper, we are more concerned about the fraction of edges in the induced subgraph, i.e., with edges with both endpoints visited by the random walk, because they are what we use to measure sample SPLs. If more edges are included in the induced subgraph, it is more likely to observe the true shortest paths from the sample. Let $E.f$ denote the fraction of edges with both of its endpoints visited by the random walk, that is, the fraction of edges in the induced subgraph. One should expect large values of $E.f$ for networks with large value of $c.v$.

For each network of size 1000, a single random walk of 200 steps is implemented to produce the induced subgraph. For each dyad in the subgraph, we take the difference between its sample SPL (SPL observed in the induced subgraph) and population SPL (SPL observed in the population graph, i.e., true SPL). Note that the sample SPL is always as large as or larger than the population SPL, as a node may take more steps in the subgraph to reach another node than it would in the population graph. Therefore the value of this difference has a range $\{0, 1, 2, ...\}$. For each value of population SPL, we plot the distribution of difference between sample SPL and population SPL. The proportion of uncovered shortest paths by the random walk sample is equal to the proportion of zero difference between sample SPL and population SPL. Therefore, we expect a large proportion with zero difference to show that the random walk sample is performing well

in uncovering the true SPL.

1) Scale-free networks v.s. Erdős-Rényi networks

We first compare a Erdős-Rényi network and a scale-free network, both of which have average degree around 6. In Figure 4.7.1, we observe a large proportion of zero difference for each value of SPL in the scale-free network, which indicates that random walks have strong ability in uncovering the true shortest paths. However, in the Erdős-Rényi network, we don't see a large proportion of zero difference, for any value of SPL greater than 1. Therefore the ability of random walks to uncover the true shortest paths in the Erdős-Rényi is very weak. This is to be expected, since the *c.v.* of degree distribution of the scale-free network is much larger than that of the Erdős-Rényi network. What's more, we notice that $E.f$ in the scale-free network is larger than that in the Erdős-Rényi network, which also explains why random walks are doing a better job in uncovering shortest paths in the scale-free network.



(a) Erdős-Rényi, $n = 1000$, $\beta = 0.2$ (b) scale-free, $n = 1000$, $\beta = 0.2$

Figure 4.7.1. Erdős-Rényi network v.s. scale-free network: distribution of difference between sample SPL and population SPL.

2) General Networks

A more general condition for random walks to uncover shortest paths is that the degree distribution has a large coefficient of variation ($c.v.$). To explore how large the $c.v.$ needs to be in order for the random walk to perform well in uncovering the shortest paths, we compare 4 networks with gamma degree distributions.

As one would expect, as the $c.v.$ increases from 0.8 in network (c) to 2.4 in network (f), $E.f$ increases, which means more edges are observed in the induced subgraph, and therefore the proportion of zero difference between sample SPL and population SPL increases. When $c.v.$ reaches 1.8 in network (e), the distribution of difference between sample SPL and population SPL looks very close to that for the scale-free network in Figure 4.7.1.

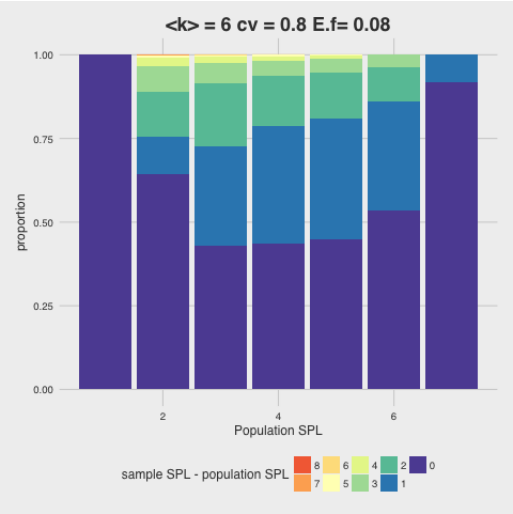
When $c.v.$ increases from 1.8 in network (e) to 2.4 in network (f), there is still an increase in the proportion of zero difference between sample SPL and population SPL, but not very substantial. One should also notice that $c.v.$ for the scale-free network in Figure 4.7.1 is 2.4. Combining the empirical results from some real networks in section 6, we get some insight about the value of $c.v.$ we need for the random walk to perform well in uncovering shortest paths:

1) If the $c.v.$ is much smaller than 2, the random walk is not able to uncover the shortest paths;

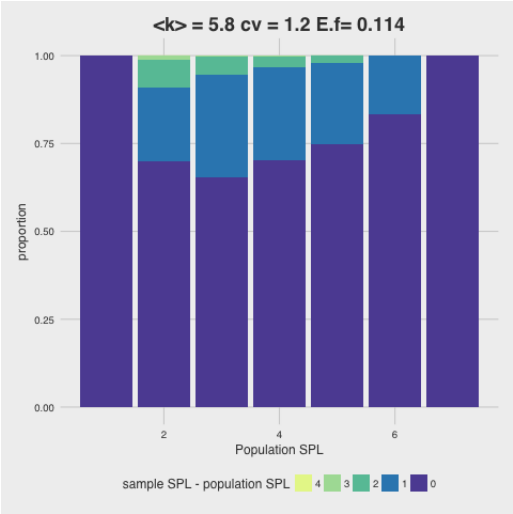
2) If the $c.v.$ is around 2, the random walk has the ability to uncover the shortest paths, but the performance may vary from case to case;

3) If the $c.v$ is much larger than 2, the random walk has strong ability to uncover most of the shortest paths between the sampled nodes.

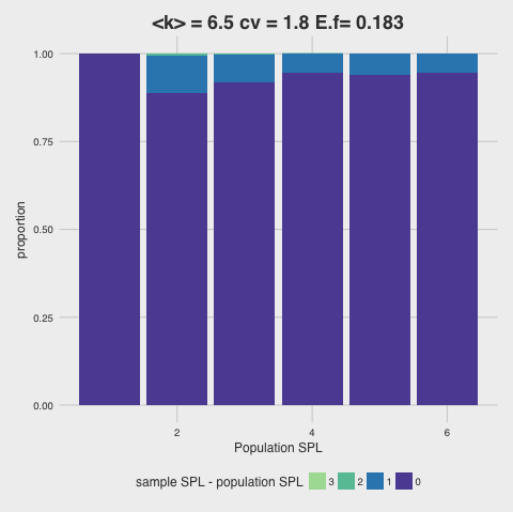
As network (f) has the same value of $c.v.$ as the scale-free network (b), we will use degree sequence generated from $Gamma(0.125, 40) + 1$ to generate networks as an example for networks with large $c.v.$ in the rest of this simulation section. And we will use degree sequence generated from $Gamma(1, 5) + 1$ (setting for network (c)) to generate networks as an example for networks with small $c.v.$. In order to evaluate the estimation performance, for a given network, a specific sampling design and a specific estimator, a total of $K = 100$ random walk samples will be drawn from the network. An estimate will be computed from each of the samples. Then the 100 estimates will be used to construct the the box plots and calculate the three numerical measures discussed in section 4.6.



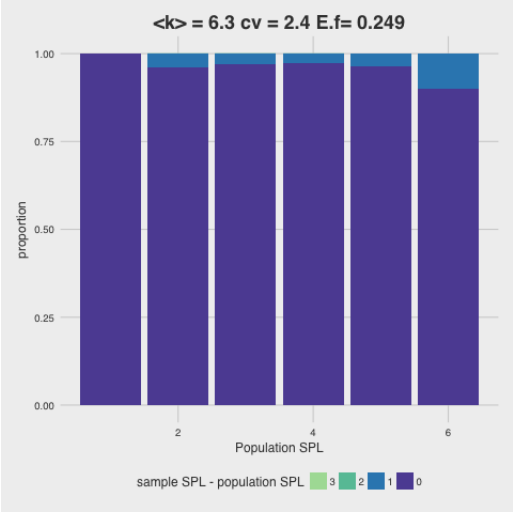
(c) Gamma(1,5)+1, $n = 1000$, $\beta = 0.2$



(d) Gamma(0.5,10)+1, $n = 1000$, $\beta = 0.2$



(e) Gamma(0.25,20)+1, $n = 1000$, $\beta = 0.2$



(f) Gamma(0.125,40)+1, $n = 1000$, $\beta = 0.2$

Figure 4.7.2. Networks with Gamma degree distribution: distribution of difference between sample SPL and population SPL.

4.7.2. Sampling designs for Random Walks

In this section, we will explore random walk sampling designs for estimating the population SPLD. We will also compare the performance of different estimators. Basically, we will answer the following four questions:

1) For networks with large *c.v.*, how many steps do we need in a single random walk in order to get a good estimation?

2) For networks with small *c.v.*, how many nodes do we need to use as landmarks and how many steps do we need in a single random walk in order to get a good estimation?

3) Will multiple random walks outperform a single random walk, given fixed sampling budget?

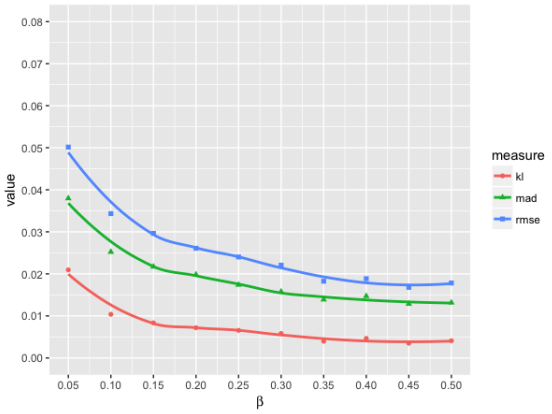
4) For a fixed sampling design, how will the performance differ by using different estimators?

4.7.2.1. Length of Random Walks for Networks with Large *c.v.*

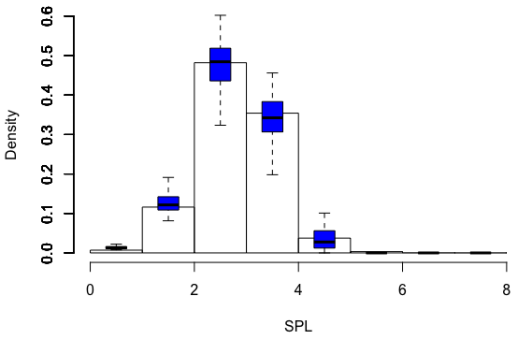
For networks with large *c.v.* in degree distribution, we use the observed SPLs in the induced subgraph to approximate the actual SPLs between sampled nodes. In order to see the effect of length of a single random walk on the estimation performance, we implement single random walks with sampling budget $\beta = 0.05(0.05)0.5$, where $x = a(r)b$ means x increasing from a to b, with r increment at each time. This process is applied to networks with $c.v. = 2.4$ and size $n = 1,000$, $n = 5,000$, and $n = 10,000$. The estimator we use here is the generalized Hansen-Hurwitz ratio estimator, denoted as HH.ra.

In Figure 4.7.3, the values of the three numerical measures of accuracy keep decreasing, as we increase the sampling budget from 0.05 to 0.5. That means, the estimation performance is improving as the single random walk gets longer, which is to be expected. However, the improvement is dramatic as the sampling rate reaches 0.2, and becomes moderate beyond that. Therefore, it is appropriate to set the minimum sampling budget β to be around 0.2 for the estimation to perform well. Let's now assume $\beta^* = \beta = 0.2$, then the computing time of approximating SPLs between all sampled nodes is $0.04n(m+n)$. Comparing it to the computing time of actual distances between all sampled nodes $0.2n(m+n)$, approximating the SPLs leads to 80% reduction in computational time. Recalling that the original computing time for DSPL is $n(m+n)$, using sampling and approximation of SPLs achieves 96% reduction in computational time for networks with large $c.v.$.

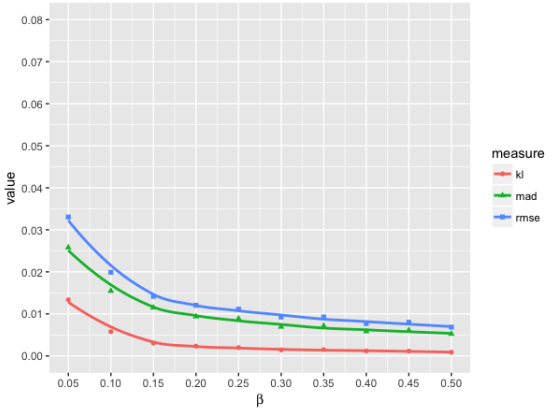
Another thing we can notice from Figure 4.7.3 is that the estimation performance is better in larger networks. More specifically, as we increase the network size, the estimates stay unbiased and their variance gets smaller. One possible reason for this phenomenon is the small world effect. For a fixed sampling budget, the sample size increases linearly with the network size, while the shortest path lengths only increases in the log scale. Therefore even with the same sampling budget, a random walk in a large network is relatively "longer" than that in a small network, and thus has a stronger ability in uncovering the shortest paths.



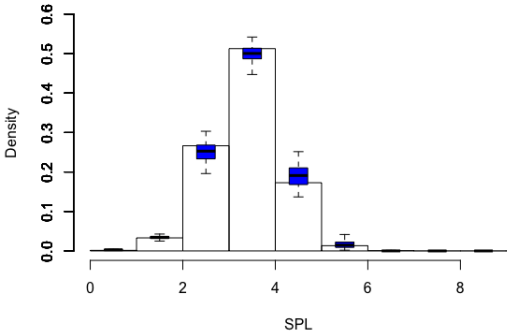
(a) $n = 1000$, $c.v. = 2.4$



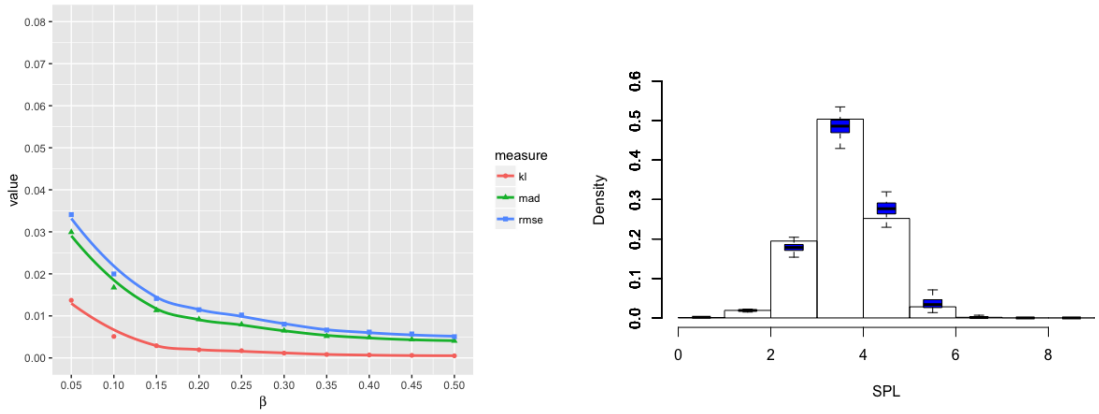
(d) $n = 1000$, $c.v. = 2.4$, $\beta = 0.2$



(b) $n = 5000$, $c.v. = 2.4$



(e) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$



(c) $n = 10000$, $c.v. = 2.4$ (f) $n = 10000$, $c.v. = 2.4$, $\beta = 0.2$

Figure 4.7.3. Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus sampling budget (β) of the random walk in a networks with large $c.v.$, measured by MAD, $RMSE$, and KL (low values are better).

4.7.2.2. Size of Landmarks and Length of Random Walks for Networks with Small $c.v.$

For networks with small $c.v.$ in degree distribution, due to the lack of powerful hubs, random walks lack strong ability to uncover shortest paths. As discussed in section 4.4.4, an alternative way is to use landmarks to estimate the SPLs between sampled nodes. We proposed using nodes in the sample with high degrees as landmarks, and a remaining question is the size of landmark set.

In order to see the effect of landmark size and single random walk length on the estimation performance, we will:

1) Fix the sampling budget at $\beta = 0.2$ and let $\gamma = 0.05(0.05)0.5$ to find the minimum fraction γ_0 for good estimation;

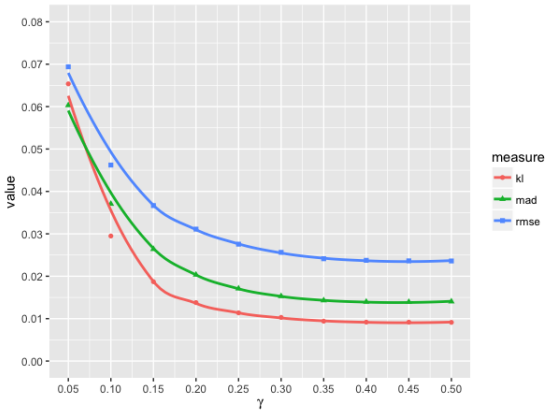
2) Fix the fraction of landmarks at $\gamma = \gamma_0$, implement single random walks with sampling budget $\beta = 0.05(0.05)0.5$, and check if a random walk with $\beta < 0.2$ is also acceptable.

The above process is applied to networks with $c.v. = 0.8$ and size $n = 1,000$, $n = 5,000$, and $n = 10,000$, as shown in Figure 4.7.4 and 4.7.5. The estimator we use here is the generalized Hansen-Hurwitz estimator, denoted as HH.ra.

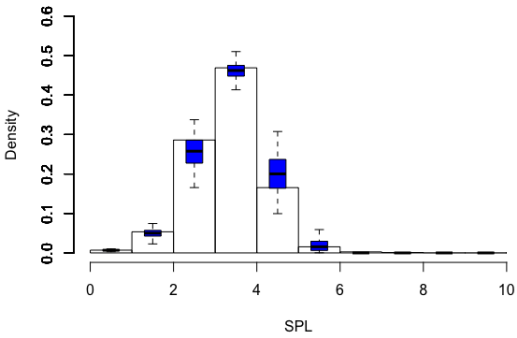
In Figure 4.7.4, the values of the three numerical measures are decreasing as γ increases from 0.05 to 0.2, and stay almost stable after 0.3. Thus we can use $\gamma_0 = 0.3$ as the minimum fraction of landmarks. In Figure 4.7.5, for large networks when $n = 5,000$ or $n = 10,000$, the estimation performance is very good if we use a sampling budget as large as $\beta = 0.2$. We can also use s smaller sampling budget such as 0.15 or even 0.1 for large networks since the estimation error will not increase too much. If we assume $\beta^* = \beta = 0.2$ and use $\gamma = 0.3$, the pre-computing time of approximating SPLs between all sampled nodes is $0.06n(m+n)$. Comparing it to the computing time of actual distances between all sampled nodes $0.2n(m+n)$, approximating the SPLs leads to 70% reduction in computational time. Recalling that the original computing time for DSPL is $n(m+n)$, using sampling and approximation of SPLs achieves 94% reduction in computational time for networks with small $c.v.$.

Similar to networks with large $c.v.$, for networks with small $c.v.$ we also notice that the estimation performance is better in larger networks. A possible reason is that as we increase the network size and fix sampling budget and landmark fraction, the number of landmarks is getting larger. And with more landmarks it is more likely to get a precise estimation of the SPLs between sampled nodes.

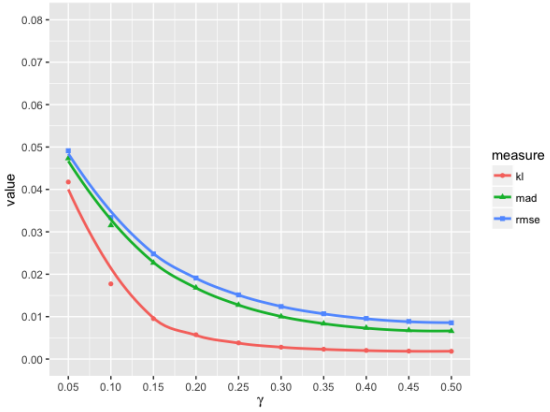
On the other hand, Figure 4.7.6 shows the change of RMSE as we increase the landmark size γ for different values of random walk length β. As expected, the lines for larger β are below the lines for smaller β. The means if the random walk is longer, less landmarks are needed. To save computation time of breadth-first search, we want the value of $\beta\gamma$ to be as small as possible. The questions remains whether to use large β and small γ or to use small β and large γ. Ideally the latter is better because by doing that we can also save the sampling cost. Suppose we want the RMSE to be as small as 0.01, there are four available combinations of β and γ listed in Table 4.1 to achieve this accuracy. Among them $\beta = 0.1$ and $\gamma = 0.5$ is the best because it achieves both the smallest sampling budget and the shortest computation time for BFS.



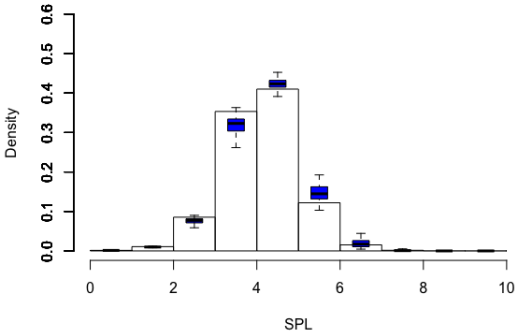
(a) $n = 1000$, $c.v. = 0.8$, $\beta = 0.2$



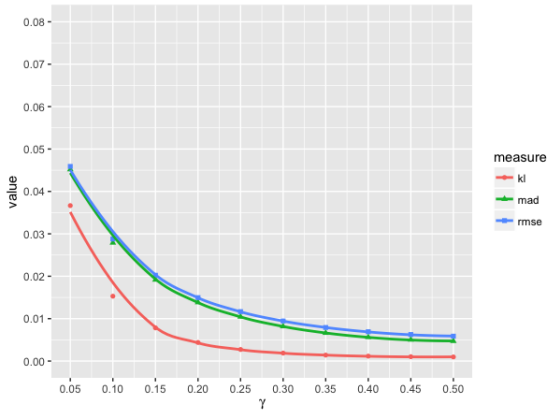
(d) $n = 1000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

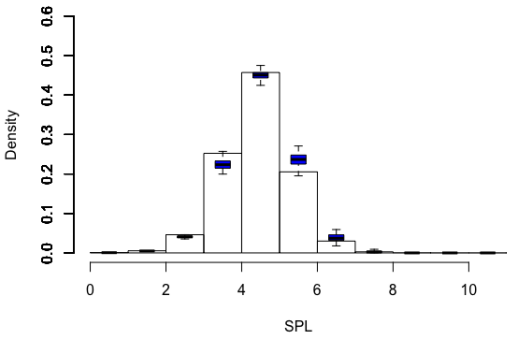


(b) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$



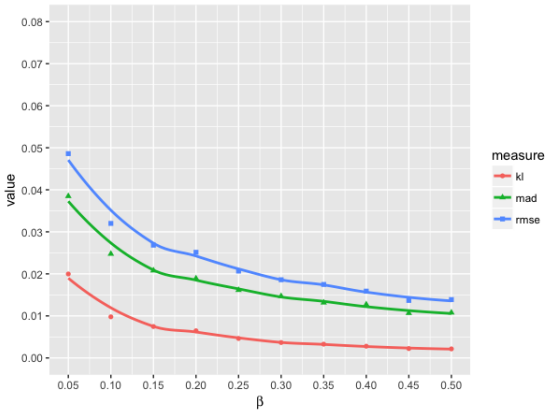
(e) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$



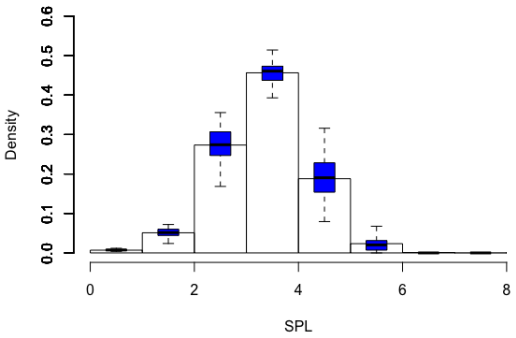


(c) $n = 10000$, $c.v. = 0.8$, $\beta = 0.2$ (f) $n = 10000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

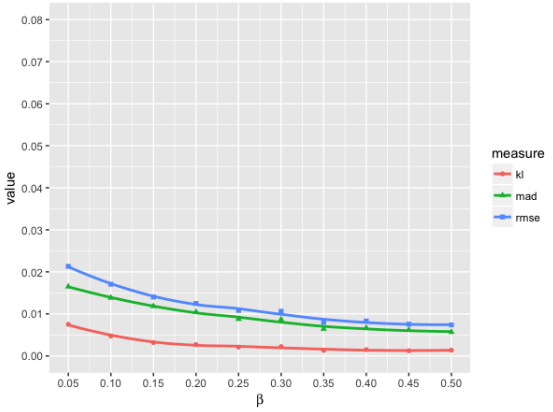
Figure 4.7.4. Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus size (γ) of landmarks in a network with small $c.v.$, measured by MAD, $RMSE$ and KL (low values are better).



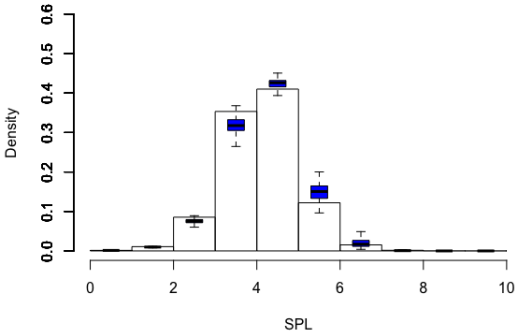
(a) $n = 1000$, $c.v. = 0.8$, $\gamma = 0.3$



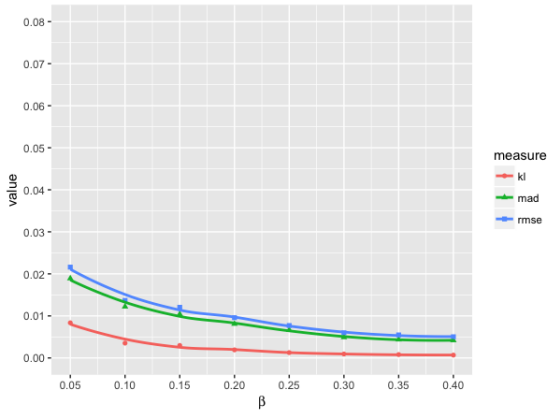
(d) $n = 1000$, $c.v. = 0.8$, $\gamma = 0.3$, $\beta = 0.2$



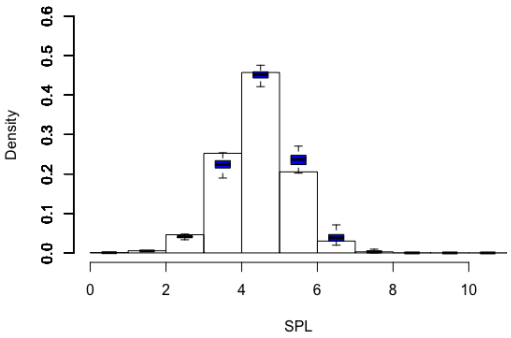
(b) $n = 5000$, $c.v. = 0.8$, $\gamma = 0.3$



(e) $n = 5000$, $c.v. = 0.8$, $\gamma = 0.3$, $\beta = 0.2$

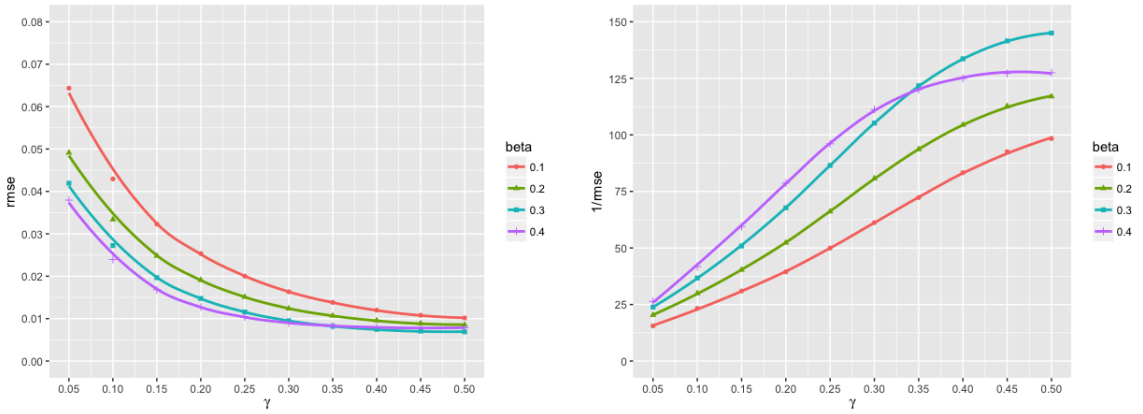


(c) $n = 10000$, $c.v. = 0.8$, $\gamma = 0.3$



(f) $n = 10000$, $c.v. = 0.8$, $\gamma = 0.3$, $\beta = 0.2$

Figure 4.7.5. Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus sampling budget (β) of the random walk in a network with small $c.v.$, measured by MAD, $RMSE$, and KL (low values are better).



(a) $RMSE$ (b) $1/RMSE$

Figure 4.7.6. Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus size (γ) of landmarks for different sampling budgets (β) of the random walks in a network with $n = 5000$ and $c.v. = 0.8$ (small $c.v.$).

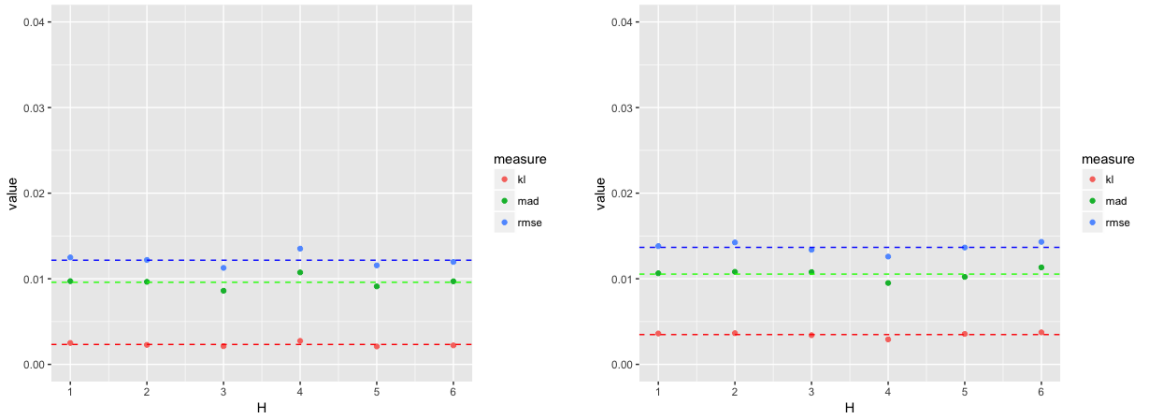
β	γ	$\beta\gamma$
0.4	0.25	0.1
0.3	0.3	0.09
0.2	0.375	0.075
0.1	0.5	0.05

Table 4.1. Comparison of combinations of sampling budget of the random walk (β) and landmark size (γ) to achieve $RMSE \approx 0.01$ in a network with $n = 5000$ and $c.v. = 0.8$ (small $c.v.$).

4.7.2.3. Number of Random Walks

To compare the estimation performance with a single random walk and multiple random walks, we fix the total sampling budget and take H independent random walk samples with H ranging from 1 to 6. For networks with large $c.v.$, we fix the total sampling budget at $\beta_0 = 0.2$. For networks with small $c.v.$, we fix the total sampling budget at $\beta_0 = 0.2$ and use $\gamma_0 = 0.3$ as the landmark fraction.

As we can observe in Figure 4.7.7, for both networks, the three numerical measures are stable as we increase the number of random walks from 1 to 6. Therefore, when keeping the total sampling budget fixed, using multiple random walks will not improve the estimation performance. In the case of networks with large $c.v$, the reason for this phenomenon is explained by Ribeiro, Basu, and Towsley [6]. As they have shown in their work, if the network has a large variance in degree distribution, two random walks intersect with high probability, and thus the subgraph induced by multiple random walks will be very similar to that induced by a single random walk. In the case of networks with small $c.v.$, where we use landmarks to estimate the SPLs between sampled nodes, although the landmarks found by a single random walk and those by multiple random walks are not necessarily the same, our simulation showed that they have similar and high betweenness centralities. We can therefore infer that they will play similar roles in estimating the distances between other nodes.



(a) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$ (b) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

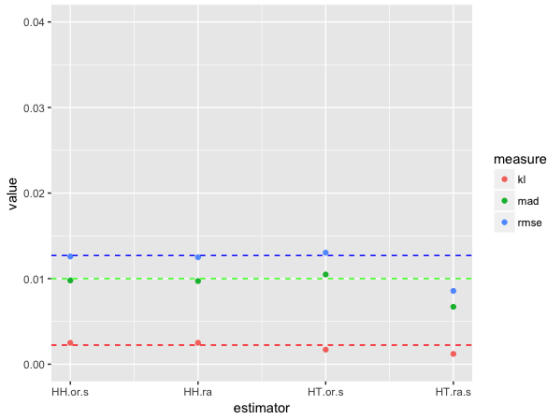
Figure 4.7.7. Estimation performance of the generalized Hansen-Hurwitz ratio estimator versus number (H) of random walks, measured by MAD, $RMSE$, and KL (low values are better).

4.7.2.4. Comparison of Estimators

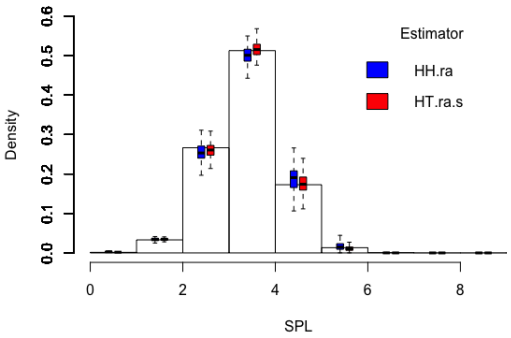
In this section, we compare the performances of the four estimators proposed in section 4.4. For generalized Hansen-Hurwitz estimator, Horvitz-Thompson estimator, and Horvitz-Thompson ratio estimator, ψ_r's and π_r's are estimated by the expressions discussed in section 4.5, therefore the estimates are denoted by HH.or.s, HT.or.s, and HT.ra.s, respectively. For generalized Hansen-Hurwitz ratio estimator, we just need to use the actual degrees of sampled nodes to compute the estimates, thus the estimates are denoted as HH.ra. The comparison based on numerical evaluations measures and comparison based on box plots are shown in Figure 4.7.8.

From the numerical comparison, one can observe that the Horvitz-Thompson ratio estimator is doing a slightly better job than the other three estimators. As we can observe from the comparison of box plots, the Horvitz-Thompson ratio estimator exhibits smaller variance than the Hansen-Hurwitz ratio estimator. There are two reasons for this phenomenon. According to the Rao-Blackwell theorem [33] (p.342), if $\hat{\theta}$ is an unbiased estimator of θ and $\theta^* = E(\hat{\theta}|T)$ where T is the sufficient statistic for θ, then θ^* is also an unbiased estimator of θ and $Var(\theta^*) \leq Var(\hat{\theta})$, and the inequality is strict unless θ is a function of T. That is, for any unbiased estimator that is not a function of the sufficient statistic, one may always obtain an unbiased estimator, depending on the sufficient statistic, that is better in terms of smaller variance. For the finite population sampling situation, the *minimal sufficient statistic* T is the unordered set of distinct, labeled observations [34]. Therefore, the Hansen-Hurwitz estimator $\hat{t}^{HH}$ is not a function of the minimal sufficient statistic while the Horvitz-Thompson estimator $\hat{t}^{HT}$ is. Note that both $\hat{t}^{HH}$ and $\hat{t}^{HT}$ are unbiased estimators for t. Based on the Rao-Blackwell theorem, we can always find another unbiased estimator $W = E(\hat{t}^{HH}|T)$ such that W has a smaller variance than $\hat{t}^{HH}$, while we cannot find such an estimator for $\hat{t}^{HT}$ as $\hat{t}^{HT} = E(\hat{t}^{HT}|T)$. Therefore $\hat{t}^{HT}$ is expected to have a smaller variance than $\hat{t}^{HH}$. Second, since the ratio form ensures that the estimated fractions for all values of SPL sum to 1, it stabilizes the estimators and therefore has a smaller variance than the original form. Theses two reasons make it not surprising for the Horvitz-Thompson ratio estimator to perform best among the four estimators.

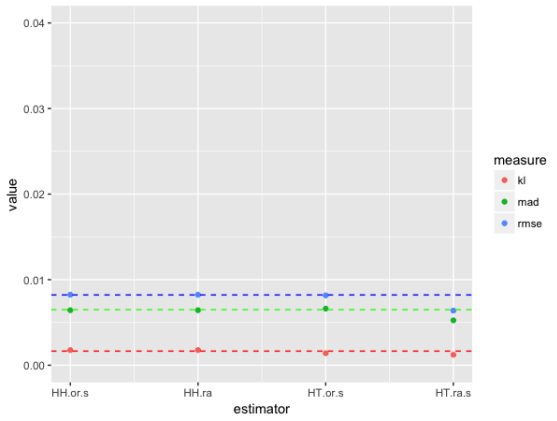
In Figure 4.7.9, we compare the performance of the Horvitz-Thompson ratio estimator and the generalized Hansen-Hurwitz ratio estimator by plotting their RMSE versus the sampling budget β. As one can observe, for the Horvitz-Thompson ratio estimator, we can use a smaller sampling budget to achieve the same estimation precision as the generalized Hansen-Hurwitz ratio estimator. For example, in network (a), the estimation precision by the generalized Hansen-Hurwitz ratio estimator with 20% sampling budget can be achieved by the Horvitz-Thompson ratio estimator with only about 12.5% sampling budget. People can choose to use the Horvitz-Thompson ratio estimator to save sampling cost while achieving a high estimation accuracy. But given that the generalized Hansen-Hurwitz ratio estimator requires less computational work and its estimation performance is close to that of the Horvitz-Thompson ratio estimator, it is still preferable to use in practice.



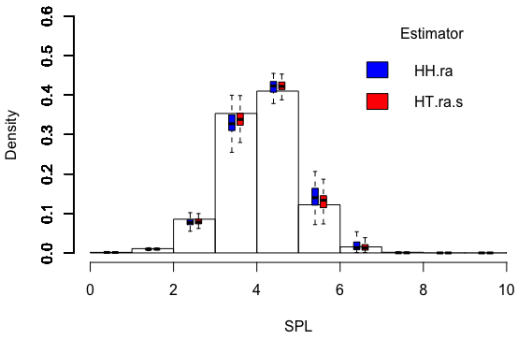
(a) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$



(b) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$

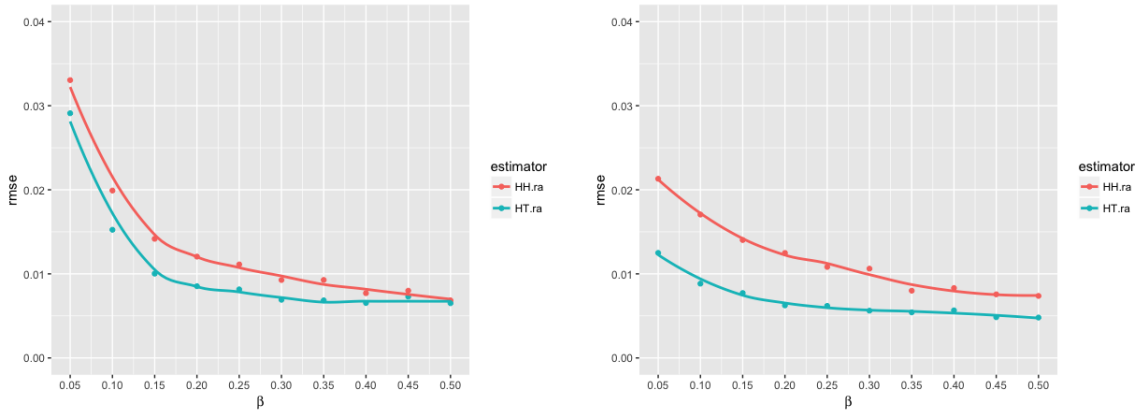


(c) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$



(d) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

Figure 4.7.8. Estimation performance versus estimators, measured by MAD, $RMSE$, and KL (low values are better).



(a) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$ (b) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

Figure 4.7.9. RMSE of the generalized Hansen-Hurwitz ratio (HH.ra) estimator and the Horvitz-Thompson ratio estimator (HT.ra.s) versus sampling budget (β).

4.7.3. Evaluation of Estimation

In order to evaluate how well our estimates from section 4.4 perform in estimating the population SPLD, we first compare the generalized Hansen-Hurwitz ratio estimates, denoted by HH.ra, to the unweighted sample SPLDs observed from the induced subgraphs, denoted by UW. Note that by using HH.ra, we are correcting bias from UW, but the bias to be corrected for networks with large $c.v.$ and networks with small $c.v$ are different. For networks with large $c.v.$, we only correct the bias from unequal sampling probabilities, because we are still using the observed SPLs between sampled nodes from the induced subgraph. For networks with small $c.v.$, we correct bias from both unequal sampling probabilities and not observing the true SPLs between sampled nodes, as we use landmarks

to estimate those SPLs.

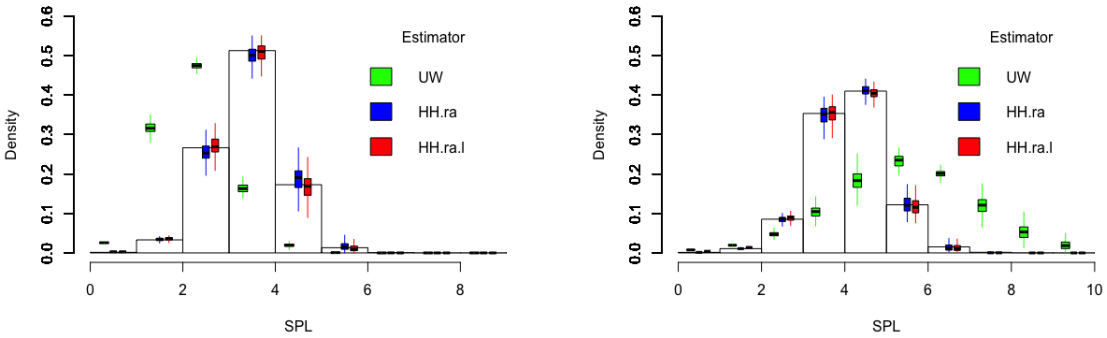
From the numerical comparison of UW and HH.ra in Table 4.2, one can observe that for both networks, about 90% of the estimation error in UW is reduced by using HH.ra. In Figure 4.7.10, one can observe that for networks with large $c.v.$ as shown in (a), the box plots for UW are shifted to the left of the population SPLD. This is because dyads with shorter SPLs are more likely to the be sampled than dyads with longer SPLs, and thus the fractions of dyads with shorter SPLs are over estimated while the fractions of dyads with long SPLs are under estimated. Therefore for networks with large $c.v.$, bias from unequal sampling probabilities is dominating in the estimation error of UW. For networks with small $c.v.$ as shown in (b), the box plots for UW are shifted to the right of the population SPLD. This is because the many observed SPLs are longer than the true SPLs. Therefore in networks with small $c.v.$, bias from not observing the true SPLs between sampled nodes is dominating in the estimation error of UW, and thus correcting it is necessary. For both networks, after applying HH.ra, the box plots stay at the right positions on the histogram with short whisker, which means the estimates are unbiased and have small variance.

On the other hand, in order to see how much we can improve if we can actually observe the true SPLs between sampled nodes, we compare our HH.ra based on approximated SPLs, to the generalized Hansen-Hurwitz estimates based on the true SPLs between sampled nodes, denoted by HH.ra.l. As we can observe, there will still be some improvement if we use the latter, but the improvement will not be huge. More specifically, in Table

4.2, the improvement from HH.ra to HH.ra.l is only about 10%. In Figure 4.7.10, we can also see that the box plots for HH.ra and those for HH.ra.l are really close. Therefore in practice, we will prefer to base our estimates on the approximated SPLs for saving computation time and not loosing much estimation accuracy.

	$n = 5000$, $c.v. = 2.4$ $\beta = 0.2$			$n = 5000$, $c.v. = 0.8$ $\beta = 0.2$, $\gamma = 0.3$		
	MAD	RMSE	KL	MAD	RMSE	KL
UW	.114	.115	0.161	.101	.103	.213
HH.ra	.010	.012	.0025	.011	.014	.003
HH.ra.l	.009	.011	.0023	.010	.013	.003

Table 4.2. Numerical comparison of the unweighted sample SPLD observed from the induced subgraphs (UW), the generalized Hansen-Hurwitz ratio estimates based on approximated SPL (HH.ra), and the generalized Hansen-Hurwitz ratio estimates based on actual SPL (HH.ra.l).



(a) $n = 5000$, $c.v. = 2.4$, $\beta = 0.2$ (b) $n = 5000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

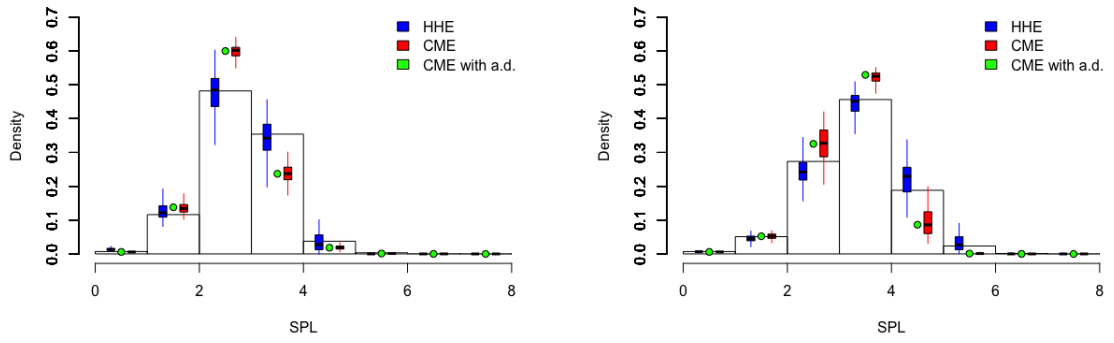
Figure 4.7.10. Box plots of the unweighted sample SPLD observed from the induced subgraphs (UW), the generalized Hansen-Hurwitz ratio estimates based on approximated SPL (HH.ra), and the generalized Hansen-Hurwitz ratio estimates based on actual SPL (HH.ra.l)

4.8. Adjusted Estimator by Weighted Average

While the estimators we proposed and discussed in the previous sections are non-parametric estimators which are not based on the degree distribution of the population network, Nitzan et al. [12] presented some analytical results for the DSPL between random pairs of nodes in configuration model networks, as discussed in section 3.5. In Figure 4.8.1, we compared the estimation performance of the generalized Hansen-Hurwitz ratio estimator (HHE), $\hat{f}_l^{GHH.r}$, and that of the configuration-model estimator (CME), $\hat{f}_l^m$.

For both networks, the CME is biased. More specifically the proportions of dyads with shorter SPLs are over estimated and the proportions of dyads with longer SPLs are under

estimated. To investigate the source of bias, we also computed the CME using the actual degree distribution. As one can observe, the bias in the CME is not corrected by using the actual degree distribution, so we can infer that the bias is from computing the DSPL from the estimated degree distribution, instead of from estimating the degree distribution. On the other hand, the HHE is almost unbiased in network (a) with large $c.v.$ and slightly biased in network (b) with small $c.v.$ where the proportions of dyads with shorter SPLs are under estimated and the proportions of dyads with longer SPLs are over estimated. Since the bias of the HHE and the bias of the CME have different directions, we consider using the weighted average of them to adjust for any possible bias of the HHE.



(a) $n = 1000$, $c.v. = 2.4$, $\beta = 0.2$ (b) $n = 1000$, $c.v. = 0.8$, $\beta = 0.2$, $\gamma = 0.3$

Figure 4.8.1. Box plots of the generalized Hansen-Hurwitz estimator (HHE), the configuration-model estimator based on estimated degree distribution (CME), and the he configuration-model estimator based on actual degree distribution (CME with a.d.).

Letting α, $0 < \alpha < 1$, denote the weight for the HHE, we define the adjusted estimator(AE), denoted as $\hat{f}_l^{adj}$, to be

$$\hat{f}_l^{adj} = \alpha \hat{f}_l^{GHH.r} + (1-\alpha)\hat{f}_l^{m}. \tag{4.80}$$

Recall that the goal of using the weighted average is to improve the estimation performance, so we will use root mean square error (RMSE) as a measure of that. Theoretically the optimal weight α^* will be to the weight by which the RMSE of the weighted average is minimized, i.e.,

$$\alpha^* = \underset{\alpha}{argmin} RMSE(\hat{f}_l^{adj}) \tag{4.81}$$

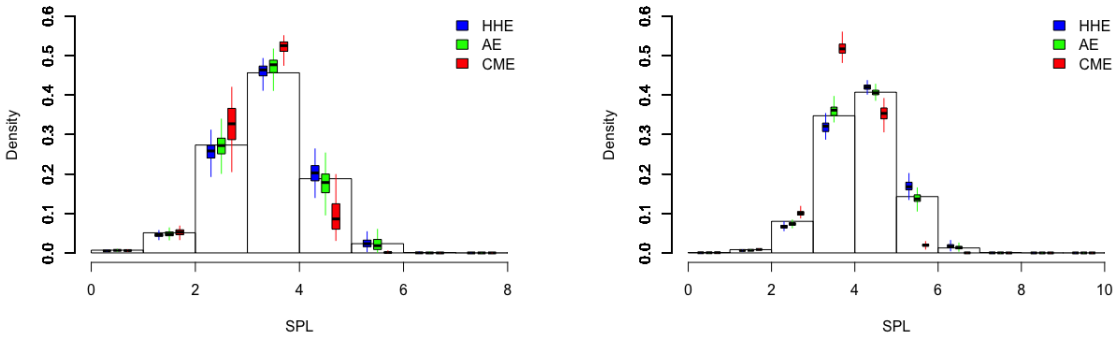
In practice, an intuitive approach to find the theoretical α^* is to solve equation

$$M = \alpha M^{GHH.r} + (1-\alpha)M^{m} \tag{4.82}$$

for α, where M is the true average SPL of graph G, $M^{GHH.r}$ is the average SPL computed from HHE $\hat{f}_l^{GHH.r}$, and M^m is the average SPL computed from CME $\hat{f}_l^m$. That is

$$\alpha^* = \frac{M - M^m}{M^{GHH.r} - M^m}. \tag{4.83}$$

In Figure 4.8.2, we plotted the adjusted estimates with weight found by Eq.(4.83) for (a) a simulated network with $c.v. = 0.8$ and (b) a real network: P2P network with $c.v. = 0.9$. For both networks, the weighted average has smaller bias variance for each SPL, so the RMSE of the weighted average will be smaller than the HHE.



(a) simulated network $n = 1000$, $c.v. = 0.8$ (b) $P2P$ network, $c.v. = 0.9$

Figure 4.8.2. Box plots of the generalized Hansen-Hurwitz estimator (HHE), the configuration-model estimator based on estimated degree distribution (CME), and the adjusted estimator (AE) with weight found by (4.83).

From results in Figure 4.8.2, using equation (4.83) is a reasonable approach to find the optimal weight α^*. However in practice, we won't know the true average SPL M without full access to the population network. Therefore we need to seek for a practical solution for weight that works for most networks. In general, if we want to pick one value for α for all networks, it's better to set the value large rather than small. If the value of α we use is large, then it will be a little too conservative to some networks with small $c.v.$, i.e., we are not making the maximum improvement, but we are still making some improvement for those networks and are not over-correcting the bias for networks with large $c.v.$. However, if we use a small value of α, then more weight is put on to the CME, and we will be over-correcting the bias for networks with large $c.v.$. After calculating α^*'s for 6 simulated networks with various values of $c.v.$, we found that $\alpha = 0.8$ is a reasonable

choice. The RMSE reduced by using the optimal α and that by using $\alpha = 0.8$ is listed on Table 4.3.

p_k	$c.v.$	α^*	RMSE by HHE	RMSE by CME	Reduction in RMSE ($\alpha = \alpha^*$)	Reduction in RMSE ($\alpha = 0.8$)
$Gamma(4, 1.25)$	0.41	0.42	0.0212	0.0205	42.13%	24.73%
$Gamma(2, 2.5)$	0.57	0.57	0.0196	0.0249	28.61%	21.67%
$Gamma(1, 5)$	0.87	0.6	0.0260	0.0252	28.04%	18.45%
$Gamma(0.8, 6.25)$	0.97	0.62	0.0330	0.0292	25.57%	16.52%
$Gamma(0.5, 10)$	1.18	0.86	0.0224	0.0367	8.59%	10.33%
$Gamma(0.25, 20)$	1.66	0.85	0.0260	0.0496	9.77%	10.79%

Table 4.3. Reduction in $RMSE$ by using the adjusted estimator (AE).

4.9. Real Networks

In this section, we test our SPLD estimation methods on data from eight real world networks. These data are available on the SNAP (Stanford Network Analysis Project) website. To simplify the analysis, we only consider nodes in the largest connected component. Table 4.4 summarizes the basic information for each network used in out test. These networks vary in size, number of edges, average degree, and most importantly, coefficient of variation. We compare the HH.ra estimates based on observed SPLs from induced subgraph (obs SPL) versus those based on estimated SPLs by landmarks (est SPL). For estimates based on observed SPLs from induced subgraph, we use a single random walk with 20% sampling budget. For estimates based on estimated SPLs by landmarks, we use a single random walk with 20% sampling budget and 30% of the sampled nodes as

landmarks. The results are shown in Table 4.5, Figure 4.9.1, and Figure 4.9.1.

As shown in plot (a), (b), and (c) in Figure 4.9.1, the estimates based on observed SPLs of the first three real networks, Oregon, AS-733, and Email-Enron are very good. This is not surprising, as the $c.v.$'s for those networks are all much larger than 2, which indicates the existence of hubs. In addition, the performance on Email-Enron network is the best among these three, as measured by the small values in MAD, RMSE, and KL in Table 4.5. This is also to be expected, since Email-Enron network has the largest size among the three. According to our discussion in previous sections, our estimation method tends to perform better for larger networks.

When the $c.v.$ gets closer to 2, the performance of estimation based on observed SPLs varies from case to case. For example, the SPLD of CA-HepPh network is a little over-estimated, while the SPLD of Wiki-Vote network is very well estimated. As one can observe, the average distance in CA-HepPh network is longer than the average distance in Wiki-Vote network, therefore random walks in CA-HepPh network are having a harder time in finding the true shortest paths. The performance of estimation is getting worse as the $c.v.$ decreases to some value below 1.5, and even below 1. For networks CA-HepTh, CA-GrQc, and P2P, the SPLDs are highly over estimated. The worst case happens to the P2P network, which only has $c.v. = 0.9$. Since there's no powerful hub in networks (f), (g), and (h) , its really hard for random walks to find the shortest paths.

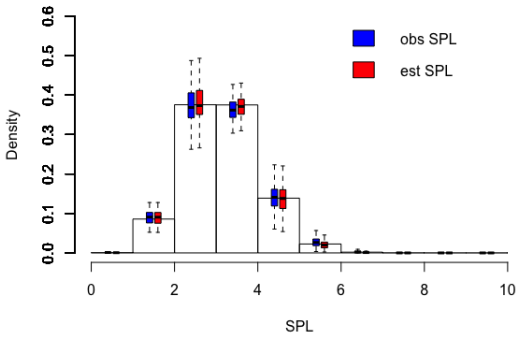
Alternatively, we can base the estimates on the estimated SPLs by landmarks. As one can notice, for networks whose estimates based on the observed SPLs are good, such as (a), (b), (c), and (e), there won't be much improvement if we base the estimates on the estimated SPLs. However, for networks with small value in $c.v.$, whose estimates based on the observed SPLs are far from the true SPLDs, such as (f), (g), and (h), using estimated SPLs will correct the bias from not observing true SPLs in the induced subgraph and therefore result in much better estimation performance.

Network	nodes	edges	$< k >$	cv	$E.f$
Oregon	10.7K	22K	4.1	7.6	0.162
AS-733	6.4K	13.2K	4.3	5.8	0.140
Email-Enron	33.7K	361.7K	21.5	3.5	0.298
CA-HepPh	11.2K	235.2K	42	2.29	0.361
Wiki-Vote	7.1K	103.7K	29.3	2.06	0.254
CA-HepTh	8.6K	49.6K	11.5	1.12	0.107
CA-GrQc	4.2K	26.8K	12.9	1.34	0.129
P2P	10.9K	40K	7.4	0.9	0.093

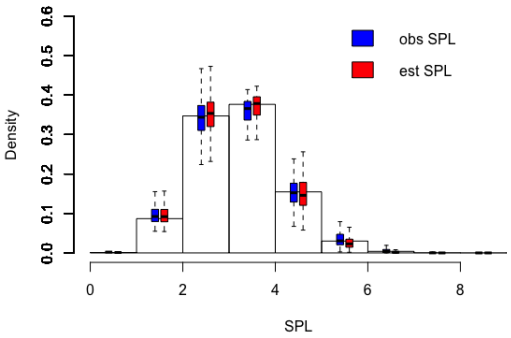
Table 4.4. Basic information of real networks.

	HH.ra by	MAD	RMSE	KL
Oregon	obs SPL	.012	.014	.0032
	est SPL	.011	.014	.0029
AS-733	obs SPL	.016	.021	.0055
	est SPL	.016	.020	.0051
Email-Enron	obs SPL	.0069	.009	.0023
	est SPL	.0085	.010	.0032
CA-HepPh	obs SPL	.026	.032	.026
	est SPL	.016	.022	.011
Wiki-Vote	obs SPL	.014	.018	.0028
	est SPL	.015	.018	.0029
CA-HepTh	obs SPL	.028	.034	.054
	est SPL	.010	.015	.012
CA-GrQc	obs SPL	.031	.038	.062
	est SPL	.015	.024	.0225
P2P	obs SPL	.086	.087	.13
	est SPL	.009	.010	.0012

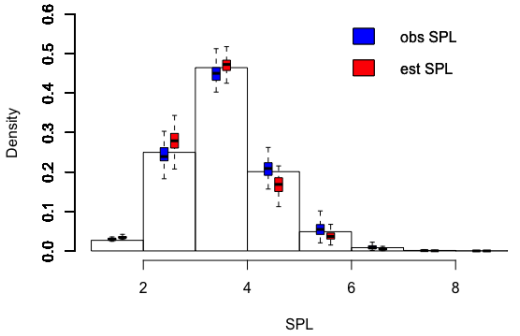
Table 4.5. Numerical evaluation measures of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$).



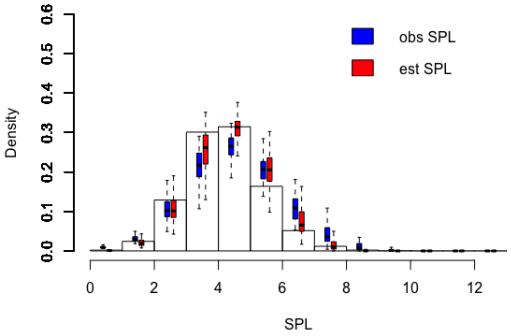
(a) Oregon ($c.v. = 7.6$)



(b) AS-733 ($c.v. = 5.8$)

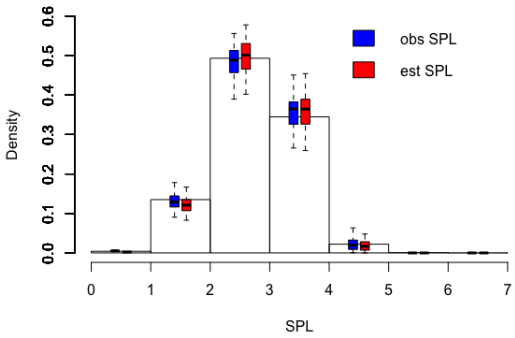


(c) Email-Enron ($c.v. = 3.5$)

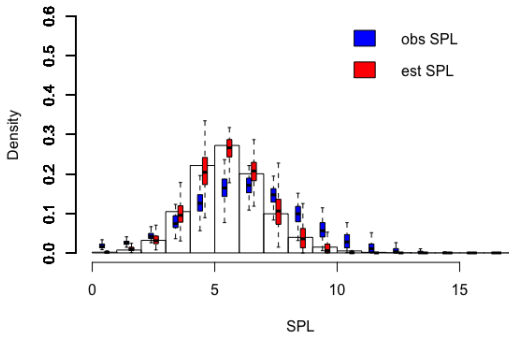


(d) CA-HepPh ($c.v. = 2.29$)

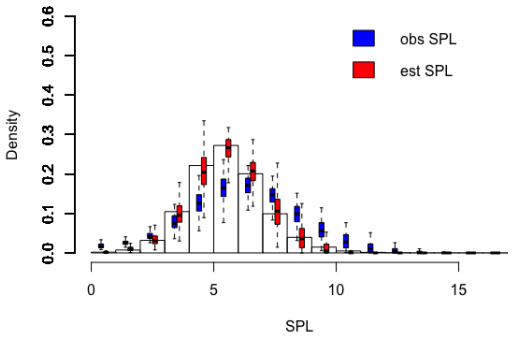
Figure 4.9.1. Box plots of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$)(part 1).



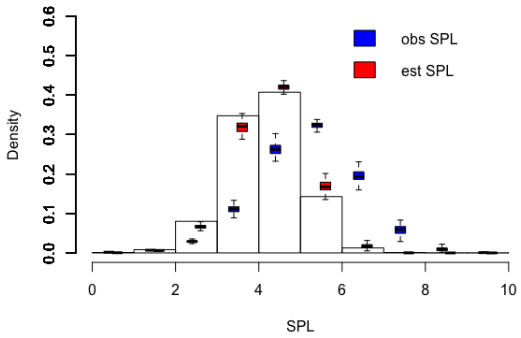
(e) Wiki-Vote ($c.v. = 2.06$)



(f) CA-HepTh ($c.v. = 1.12$)



(g) CA-GrQc ($c.v. = 1.34$)



(h) P2P ($c.v. = 0.9$)

Figure 4.9.2. Box plots of estimated SPLDs of real networks: HH.ra by observed SPL ($\beta = 0.2$) v.s. HH.ra by estimated SPL by landmarks ($\beta = 0.2$, $\gamma = 0.3$)(part 2).

4.10. Summary of Results

By applying the estimators proposed in section 4.4 and evaluation metrics discussed in section 4.6 to the simulated networks studied in section 4.7, we have the following findings:

- When a network has $c.v. > 2$ in degree distribution, random walks have strong ability to discover the actual shortest paths between sampled nodes. Therefore we can use the observed SPLs between sampled nodes in the induced subgraph to approximate their actual SPLs.
- When a network has $c.v. < 2$ in degree distribution, random walks don't have strong ability to discover the actual shortest paths between sampled nodes. Therefore we need to do breadth-first search in the population graph to get the actual SPLs, but only to a fraction, such as 30%, of the sampled nodes (known as "landmarks"), and use that information to approximate the SPLs between other sampled nodes.
- The estimation performance improves as sampling budget increases, with dramatic improvement as the sampling budget reaches 20% and moderate improvement beyond that.
- If we use 20% as sampling budget, using sampling and approximation of SPLs between sampled nodes will achieve 96% reduction in computational time for networks with large $c.v.$ and 94% reduction for networks with small $c.v.$.
- If we fix the total sampling budget, such as 20%, using a single random walk performs equally well as using multiple random walks.
- To a small degree, the Horvitz-Thompson ratio estimator outperforms the other estimators, but the generalized Hansen-Hurwitz ratio estimator is still preferable to use in practice given its high estimation accuracy and easiness in computation.
- The estimation performance improves as the network size increases, but tends to be stable once the network is large enough, such as of size $n = 5000$ or larger.

- When the Hansen-Hurwitz ratio estimator is slightly biased for networks with small *c.v.*, we can use the weighted average of it and a configuration-model estimator proposed by Nitzan, Katzav, Kühn, *et al.* [12] to adjust for bias. A reasonable weight to use for the Hansen-Hurwitz ratio estimator is 0.8.

4.11. Discussion and Future Work

In this chapter we studied methods for estimating SPLDs in networks via random walk sampling. More specifically we applied Hansen-Hurwitz estimator, Horvitz-Thompson estimator, and their ratio forms to estimate SPLDs by subgraphs induced by random walk samples. There are two problems associated with this estimating process: 1) we are not able to observe the actual SPLs between sampled nodes from the induced subgraph; 2) pairs of nodes are sampled with unequal probabilities so the unweighted estimator is biased. To solve the first problem, we used approximations for SPLs. We approximated the actual SPLs between sampled nodes by their observed SPLs in the induced subgraph for networks with large *c.v.* and by the minimum of the sum of their distances to landmarks for networks with small *c.v.*. To deal with the second problem, we used weighted estiamtors: the Hansen-Hurwitz estimator for samples with duplicate and the Horvitz-Thompson estimator for samples without duplicates. For the Hansen-Hurwitz estimator, by theory of Markov chains we showed that the inverse of the weight of a pair is approximately proportional to the product of the degrees of the two nodes when the random walk is sufficiently long. For the Horvitz-Thompson estimator we approximated the random walk sampling of nodes by an adjusted multinomial sampling model, and computed the

weight accordingly.

By applying our proposed estimators to some simulated networks and real networks, we have found that 1) for large networks, high estimation accuracy can be achieved by using a single random or multiple random walks with total number of steps equal to at least 20% of the nodes in the network; 2) about 94% to 96% reduction in computational time can be achieved by using sampling and approximation of SPLs between sampled nodes; 3) the estimation performance increases as the network size increases but tends to stabilize when the network is large enough; 4) a single random walk performs as well as multiple random walks; 5) the Horvitz-Thompson ratio estimator performs best among the four estimators.

While the estimator for networks with large $c.v.$ is unbiased, the estimator for networks with small $c.v.$ is slightly biased, due to the lack of estimation accuracy for SPLs between sampled nodes. In section 4.8, we proposed using the weighted average of the Hansen-Hurwitz ratio estimator and a configuration-model estimator developed by Nitzan, Katzav, Kühn, *et al.* [12] to adjust for bias. After developing the theoretical optimal weight and applying it to several simulated networks we found that 0.8 is a reasonable weight to use for the Hansen-Hurwitz ratio estimator, and thus the weight for the configuration-model estimator is 0.2. Based on our simulation results, this adjustment is helpful in reducing the bias. However, it only works for some real networks and cannot fully eliminate the bias. Therefore one future direction of this work is to develop alternative methods to further reduce the bias in estimating SPLDs for networks with small $c.v.$. There are

two possible approaches: 1) applying or inventing an algorithm to approximate the SPLs between sampled nodes that can work better than using landmarks; 2) developing a better way to adjust for bias from estimator of SPLD based on approximated SPLs using landmarks.

CHAPTER 5

Estimation of Closeness Centrality Ranking

5.1. Overview

Closeness centrality measures how close a node is to other nodes in a given network. In reality people are more interested in the rank of closeness centrality as it measures the relative importance of a node in the network. People need to compute the closeness centrality of all nodes in the network in order to find the exact rank of a node, and this is computationally expensive for large networks. In this chapter we study the problem of estimating closeness centrality rank via random walk sampling through three stages: 1) estimating closeness centrality of nodes from a random walk sample by Hansen-Hurwitz ratio estimator and approximated geodesic distances; 2) estimating the population cumulative distribution function (CDF) of closeness centrality by weighted kernel estimator; 3) for a given node, computing its estimated closeness centrality rank from the estimated CDF of closeness centrality. Application to simulated networks and real networks show that the weighted rank estimator performs well in estimating the closeness centrality ranking of a given node: 1) for networks with large $c.v.(c.v. > 2)$, the estimator is unbiased with moderate standard deviation; 2) for networks with small $c.v.(c.v. < 2)$, the estimator is slightly biased due to the bias in geodesic distance estimation, but not to a large extent.

5.2. Definitions

Closeness centrality is a metric that measures how close a node is to other nodes in a given network. Mathematically, in a network with n nodes, the *closeness centrality* of node i, denoted as c_i, is defined as the inverse of the mean geodesic distance from node i to all nodes in the network:

$$c_i = \frac{n}{\sum_{j=1}^{n} l_{ij}}, \tag{5.1}$$

where l_{ij} is the geodesic distance between node i and node j in the population network.

In reality, we are more interested in the rank of closeness centrality of a node in a network. It can measure the relative importance of a node in information delivery in a network. The *closeness centrality rank* of node i, denoted by R_i, $R_i \in \{1, 2, ..., n\}$, is the relative position of node i in the network, based on its value of closeness centrality. For instance, in a network with n nodes, $R_i = 1$ if node i has the largest value of c_i in the network, and $R_i = n$ if node i has the smallest value of c_i in the network.

Mathematically, we can compute the rank of closeness centrality R_i of node i from its closeness centrality c_i and the CDF of closeness centrality of the network. Let C denote the random variable for closeness centrality, and let $F(c) = P(C \leq c)$ denote the *cumulative distribution function (CDF)*, also called *distribution function*, of closeness centrality of the network. In a network with n nodes, we can compute the rank of closeness centrality

R_i of node i by

$$R_i = (n+1) - nF(c_i). \tag{5.2}$$

5.3. Estimating Methods

When the network is large, computing the closeness centrality for all nodes can be time consuming, therefore the actual CDF of the network will be unknown. In this case we consider using sampling to estimate the CDF of closeness centrality of the network first, and then plug in the value of closeness centrality of a particular node to estimate its rank of closeness centrality. More specifically, the estimation process has three stages:

1) Estimate closeness centrality of sampled nodes;

2) Estimate the CDF of closeness centrality of population network from the estimated closeness centrality of sampled nodes;

3) For a particular node, plug in its closeness centrality value into the estimated CDF to estimate its rank of closeness centrality.

5.3.1. Estimating closeness centrality of sampled nodes

Recall that V^* is the set of distinct nodes visited by the H random walks. Let $|V^*|$ denote the the size of V^*. Let d_{ij} denote the observed geodesic distance between node i and node j in the induced subgraph G^*. For any sampled node i, we define its observed closeness centrality in the sample as

$$c_i^{obs} = \frac{|V^*|}{\sum_{j \in V^*} d_{ij}}. \tag{5.3}$$

However, c_i^{obs} is biased since 1) nodes in V^* are sampled with unequal probabilities by random walk sampling and 2) the observed geodesic distance d_{ij} is not always equal to the actual geodesic distance l_{ij}. To solve the first problem, we generalize the Hansen-Hurwitz estimators to account for the unequal selection probabilities. To solve the second problem, we adopt different strategies to approximate the actual geodesic distance l_{ij} for networks with different values of $c.v.$. For networks with large $c.v.$, we approximate l_{ij} by the observed geodesic distance d_{ij}. For networks with large $c.v.$, we approximate l_{ij} using landmarks.

1) Generalized Hansen-Hurwitz Estimator

Let $s = \{X(1), X(2), ..., X(H)\}$ denote the set of sequences of nodes visited by H random walks, including duplicates, and let $|s| = H \cdot B$ denote the size of s. Let $I(X_b^{(h)} = i)$ denote an indicator variable taking the value 1 if node i is visited at the b^{th} step in the h^{th} random walk, and zero otherwise. Let $q_i = \sum_{h=1}^{H} \sum_{b=1}^{B} I(X_b^{(h)} = i),\ i = 1, ..., n$, denote the number of times node i appears in sample s and let $\boldsymbol{q} = (q_1, q_2, ..., q_n)^T$.

Let $t_i = \sum_{j=1}^{n} l_{ij}$ denote the total geodesic distances from node i to all nodes in the network, then $c_i = \frac{n}{t_i}$. Using the generalized Hansen-Hurwitz estimator, we can estimate t_i and n by

$$\hat{t}_i = \sum_{j=1}^{n} \frac{q_j l_{ij}}{E(q_j)}, \tag{5.4}$$

and

$$\hat{n} = \sum_{j=1}^{n} \frac{q_j}{E(q_j)}, \tag{5.5}$$

and therefore we can estimate c_i by

$$\frac{n}{\hat{t}_i} = \frac{n}{\sum_{j=1}^{n} q_j l_{ij}/E(q_j)}, \tag{5.6}$$

or the ratio estimator

$$\frac{\hat{n}}{\hat{t}_i} = \frac{\sum_{j=1}^{n} q_j/E(q_j)}{\sum_{j=1}^{n} q_j l_{ij}/E(q_j)}. \tag{5.7}$$

Let $\boldsymbol{p} = (p_1, p_2, ..., p_n)^T$, where $p_i = \frac{k_i}{K}$ with $K = \sum_w k_w$. As shown in [35], a single random walk $\{X_t\}$ on a connected graph $G = (V, E)$ with at least one triangle is an irreducible and aperiodic Markov chain with a stationary distribution $\boldsymbol{p}$. According to Anderson's (1989) results for irreducible and aperiodic Markov chains, $E(\boldsymbol{q}) = \boldsymbol{p}t$. Applying this in our case, we will have $E(q_j) = |s|\frac{k_j}{K}$. We define the Hansen-Hurwitz estimator for c_i by actual geodesic distances as

$$\hat{c}_i^{HH.l} = \frac{n}{\frac{K}{|s|}\sum_{j=1}^{n} q_j l_{ij}/k_j}, \tag{5.8}$$

and define the Hansen-Hurwitz ratio estimator for c_i by actual geodesic distances as

$$\hat{c}_i^{HH.ra.l} = \frac{\sum_{j=1}^{n} q_j/k_j}{\sum_{j=1}^{n} q_j l_{ij}/k_j}. \tag{5.9}$$

2) Approximating geodesic distances between sampled nodes

Ribeiro, Basu, and Towsley [6] have shown that random walks have strong ability in uncovering the shortest paths in networks with high degree variability. Zheng and Spencer [35] have shown this property generalized to networks with large $c.v.$ (e.g., $c.v. > 2$). For a network with large $c.v$, we propose using the observed geodesic distance d_{ij} in the induced subgraph to estimated the actual geodesic distance l_{ij} in the population graph for dyad (i, j).

For networks with small $c.v.$ in degree distribution, since random walks can't find the shortest paths in the induced subgraph, we need to implement breadth-first search (BFS) on sampled nodes in the population graph to find the shortest paths. However, based on findings by Potamias, Bonchi, Castillo, *et al.* [5], the BFS doesn't have to be applied to all sampled nodes. Instead, one can apply BFS to only a fraction of the sampled nodes to find their shortest distances to all other nodes, and use that information to estimate the shortest distances between other sampled nodes. More specifically, one can first select a set of nodes as landmarks, denoted as D, pre-compute the SPLs from landmarks to all other nodes by BFS in the population graph, and estimate the SPL between any arbitrary pair of nodes i and j by $min_{v \in D}\{l_{iv} + l_{vj}\}$.

We define the Hansen-Hurwitz estimator for c_i by estimated geodesic distances as

$$\hat{c}_i^{HH} = \frac{n}{\frac{K}{|s|}\sum_{j=1}^{n} q_j \hat{l}_{ij}/k_j}, \tag{5.10}$$

and define the Hansen-Hurwitz ratio estimator for c_i by estimated geodesic distances as

$$\hat{c}_i^{HH.ra} = \frac{\sum_{j=1}^{n} q_j/k_j}{\sum_{j=1}^{n} q_j \hat{l}_{ij}/k_j}, \tag{5.11}$$

where $\hat{l}_{ij} = d_{ij}$ for networks with $c.v. > 2$ and $\hat{l}_{ij} = min_{v \in D}\{l_{iv} + l_{vj}\}$ for networks with $c.v. < 2$.

5.3.2. Estimating the population CDF of closeness centrality

In this section, we develop methods to estimate the CDF of closeness centrality of the population graph using the estimated closeness centralities of sampled nodes. Intuitively, one would use the empirical distribution of the estimated closeness centrality of sampled nodes (empirical estimated CDF) to approximate the shape of distribution of closeness centrality of population graph. This will give us a general idea of how well the estimated CDF fits the true CDF, but we cannot get an accurate estimate of CDF for each node in the population using the empirical estimated CDF. This is because the empirical CDF is a discrete function with number of values equal to the elements in the data set. Since the sample size is always smaller than the population size, the number of possible values in the empirical estimated CDF is smaller than that in the true population CDF, and thus there will be many equal values for the estimates of CDF. In order to deal with this issue, we apply kernel estimator to smooth the empirical estimated CDF and get a continuous estimated CDF.

1) Empirical estimated CDF

If the elements are sampled with equal probabilities, we can approximate the empirical CDF of population by the empirical CDF of sampled nodes. The unweighted estimated empirical CDF is

$$\hat{F}(c) = \frac{\sum_{i \in s} I(C_i \leq c)}{|s|}. \tag{5.12}$$

If the elements are sampled with unequal probabilities, the empirical CDF of sampled nodes deviates from the empirical CDF of population. Instead we need to use a weighted empirical CDF of sampled nodes to approximate the empirical CDF of population. The weighted estimated empirical CDF is

$$\hat{F}^w(c) = \frac{\sum_{i \in s} w_i I(C_i \leq c)}{\sum_{i \in s} w_i}, \tag{5.13}$$

where $w_i = K/k_i$.

2) Smoothed Estimated CDF

Kernel density estimation is a commonly-used technique to get a continuous estimator for the population density from a sample of discrete values. Mathematically, for a positive real number h, the *kernel density estimator* $\hat{p}(\cdot)$ is defined as

$$\hat{p}(y) = \frac{1}{nh} \sum_{j=1}^{n} K(\frac{y - Y_j}{h}), -\infty < y < \infty, \tag{5.14}$$

where $K(\cdot)$ is a continuous, nonnegative density function satisfying $\int_{-\infty}^{\infty} K(u)du = 1$, also known as the *kernel*, h is known as the *smoothing parameter* or the *bandwidth*, and n is

the sample size.

Some commonly-used kernels include Gaussian kernel, biweight kernel, Epanechnikov kernel, and triangular kernel. Different kernels lead to different density estimates, but they tend to be very similar. In this paper we choose to use the Gaussian kernel:

$$K(u) = \frac{1}{\sqrt{2\pi}} exp\{-\frac{1}{2}u^2\}. \tag{5.15}$$

The choice of smoothing parameter h is another issue in using the kernel estimator. In general, large value in h will lead to a smoother estimator, but it might not be accurate enough to capture the important features of the data. In contrast, small value in h will capture the data feature very well but it might not satisfy the desired level of smoothness. Therefore a good choice of h is a balance of desired level of smoothness and the ability of capturing the important features of the data. In this work, after analyzing the results for several different values of h, we choose to use $h = 0.01$.

Once we have the kernel density estimator $\hat{p}(\cdot)$, we can compute the smoothed distribution function $\hat{F}_K(y)$ by

$$\hat{F}_K(y) = \int_{-\infty}^{y} \hat{p}(t)dt, -\infty < y < \infty. \tag{5.16}$$

Consider the case of a Gaussian kernel, we have

$$\hat{p}(y) = \frac{1}{nh} \sum_{j=1}^{n} \phi(\frac{y - Y_j}{h}), -\infty < y < \infty, \tag{5.17}$$

where $\phi(\cdot)$ is the density function of the standard normal distribution, and therefore

$$\hat{F}_K(y) = \frac{1}{n}\sum_{j=1}^{n}\int_{-\infty}^{y}\frac{1}{h}\phi(\frac{t-Y_j}{h})dt, -\infty < y < \infty. \tag{5.18}$$

After some algebra using the change-of-variable $u = (t - Y_j)/h$, we have

$$\hat{F}_K(y) = \frac{1}{n}\sum_{j=1}^{n}\Phi(\frac{y-Y_j}{h}), -\infty < y < \infty, \tag{5.19}$$

where $\Phi(\cdot)$ is the distribution function of the standard normal distribution.

Applying Eq. (5.19) to our case of estimating the CDF of closeness centrality, the smoothed estimated CDF is defined as

$$\hat{F}_s(c) = \frac{1}{|s|}\sum_{i\in s}\Phi(\frac{c-C_i}{h}). \tag{5.20}$$

Here C_i is a random variable that denotes the value of closeness centrality of sampled node i. In section 6.2, we will compare the estimation performance by using the actual closeness centrality c_i and that by using the estimated closeness centrality $\hat{c}_i^{HH.ra}$ or $\hat{c}_i^{HH.ra.l}$.

While the above methodology is based on a sample where the elements are sampled with equal probabilities, we can also extend the usage of kernel estimator to the case where elements are sampled with unequal probabilities. More specifically, we define the smoothed weighed estimator for CDF as

$$\hat{F}_s^w(c) = \frac{1}{\sum_{i\in s} w_i}\sum_{i\in s} w_i\Phi(\frac{c-C_i}{h}), \tag{5.21}$$

where $w_i = 1/\phi_i = K/k_i$.

5.3.3. Estimating closeness centrality rank of a given node

For a particular node i in a network with closeness centrality c_i, if $F(c)$ is known, we can compute its rank of closeness centrality R_i by plugging c_i into Eq.(5.2). However in many real cases, $F(c)$ is unknown and we need to estimate it through Eq.(5.20) or Eq.(5.21). Therefore, we define the estimated rank of node i through $\hat{F}_s(c)$as

$$\hat{R}_i = (n+1) - \lfloor n\hat{F}_s(c_i) \rfloor, \tag{5.22}$$

where $\lfloor x \rfloor$ means rounding x down to its closest integer, and define the estimated rank of node i through $\hat{F}_s^w(c)$as

$$\hat{R}_i^w = (n+1) - \lfloor n\hat{F}_s^w(c_i) \rfloor. \tag{5.23}$$

5.4. Evaluation Metrics

In this section, we list some numerical metrics for evaluating the estimation performances of 1) closeness centrality of sampled nodes, 2) CDF of closeness centrality of population graph and, 3) closeness centrality rank of a given node.

5.4.1. Mean Absolute Error (MAE)

To evaluate the estimation performance of closeness centrality of a sampled node $\hat{c}_i$, we use mean absolute error. Mean absolute error (MAE) measures the average absolute distance between the estimate and the parameter.

For k^{th} sample, the MAE for $\hat{c}_i$ is

$$MAE_k^c = \frac{1}{|s|} \sum_{i \in s} |\hat{c}_i - c_i|. \tag{5.24}$$

Across K samples, the average of MAE^c is

$$mean(MAE^c) = \frac{1}{K} \sum_{k=1}^{K} MAE_k^c, \tag{5.25}$$

and the standard deviation of MAE^c is

$$s.d.(MAE^c) = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (MAE_k^c - mean(MAE^c))^2}. \tag{5.26}$$

5.4.2. Kolmogorov-Smirnov Distance (KS)

To measure the estimation performance of smoothed CDF of closeness centrality of population graph $\hat{F}_s(c)$, we use Kolmogorov-Smirnov distance. Kolmogorov-Smirnov distance (KS) measure the maximum distance between two distribution functions. For k^{th} sample, the KS for $\hat{F}_s(c)$ is

$$KS_k = \underset{c_i}{sup} |\hat{F}_s(c_i) - F(c_i)|. \tag{5.27}$$

Across K samples, the average of KS is

$$mean(KS) = \frac{1}{K}\sum_{k=1}^{K} KS_k, \tag{5.28}$$

and the standard Deviation of KS is

$$s.d.(KS) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(KS_k - mean(KS))^2}. \tag{5.29}$$

5.4.3. Percentage Mean Absolute Error (PMAE)

To evaluate the estimation performance of closeness centrality ranking of a given node $\hat{R}_i$, we use percentage mean absolute error (PMAE).

For k^{th} sample, MAE for $\hat{R}_i$ is

$$MAE_k^R = \frac{1}{n}\sum_{i=1}^{n} |\hat{R}_i - R_i|, \tag{5.30}$$

and the $PMAE$ for $\hat{R}_i$ is

$$PMAE_k^R = \frac{MAE_k^R}{n} \times 100\% \tag{5.31}$$

Across K samples, the average of $PMAE^R$ is

$$mean(PMAE^R) = \frac{1}{K}\sum_{k=1}^{K} PMAE_k^R, \tag{5.32}$$

and the standard deviation of $PMAE^R$ is

$$s.d.(PMAE^R) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(PMAE_k^R - mean(PMAE^R))^2}. \tag{5.33}$$

5.5. Simulation Study

In this section, we test our estimators developed in section 5.3 using some graphical tools and the numerical metrics discussed in section 5.4 for the three stages in estimating the rank of closeness centrality respectively. We do the evaluation for both a network of size $n = 1000$ with large $c.v.$ ($c.v. = 2.5$) and a network $n = 1000$ with small $c.v.$ ($c.v. = 0.7$). From each population network, $K = 100$ random walks samples are taken with sampling budget $\beta = 0.3$, i.e., each random walk takes $B = 300$ steps. For the network with large $c.v.$, we use the observed geodesic distances in the induced subgraph to approximate the actual geodesic distances between sampled nodes. For the network with small $c.v.$, we use distances computed from landmarks to approximate the actual geodesic distances between sampled nodes.

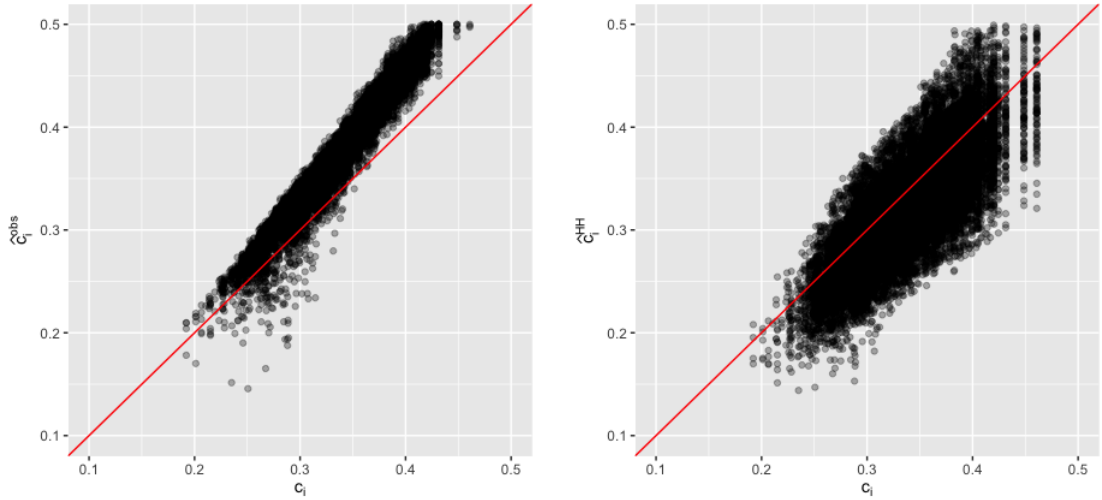
5.5.1. Estimating closeness centrality of sampled nodes

In Figure 5.5.1 and 5.5.2, we plot the estimated closeness centrality of sampled nodes versus their actual closeness centrality for a network with large $c.v.$ ($c.v. = 2.5$) and that for a network with small $c.v.$ ($c.v. = 0.7$). The four estimators are a) the observed closeness centrality from the induced subgraph $\hat{c}_i^{obs}$, b) Hansen-Hurwitz estimator based on approximated geodesic distances $\hat{c}_i^{HH}$, c) Hansen-Hurwitz ratio estimator based on approximated geodesic distances $\hat{c}_i^{HH.ra}$, and d) Hansen-Hurwitz ratio estimator based on actual geodesic distances $\hat{c}_i^{HH.ra.l}$.

For the network with large $c.v.$ ($c.v. = 2.5$), $\hat{c}_i^{obs}$ and $\hat{c}_i^{HH}$ do not perform very well. For $\hat{c}_i^{obs}$, most estimated c_i's are greater than the actual c_i's, especially for nodes with

large c_i's, so the bias is not negligible. $\hat{c}_i^{HH}$ is less biased but the variance is large. On the other hand, $\hat{c}_i^{HH.ra}$ and $\hat{c}_i^{HH.ra.l}$ perform well in terms of being unbiased and having small variance. $\hat{c}_i^{HH.ra.l}$ performs slightly better than $\hat{c}_i^{HH.ra}$ for nodes with small c_i's, but not to a remarkable extent. This can also be verified by the numerical error measures listed on the left table from Table 5.1. The MAE's of $\hat{c}_i^{obs}$ and $\hat{c}_i^{HH}$ are much greater than the MAE's of $\hat{c}_i^{HH.ra}$ and $\hat{c}_i^{HH.ra.l}$, while the MAE of $\hat{c}_i^{HH.ra}$ is only slightly larger than the MAE of $\hat{c}_i^{HH.ra}$. Therefore, for a network with large $c.v.$, approximating the actual geodesic distances using the observed geodesic distances in the induced subgprah is reasonable, and $\hat{c}_i^{HH.ra}$ is preferable to use in practice.

For the network with small $c.v.$ ($c.v. = 0.7$), the estimation performances are slightly different. For $\hat{c}_i^{obs}$ and $\hat{c}_i^{HH}$, most estimated c_i's are smaller than the actual c_i's, and the variance is large. $\hat{c}_i^{HH.ra}$ and $\hat{c}_i^{HH.ra.l}$ perform similarly, but some c_i's are under estimated by $\hat{c}_i^{HH.ra}$, especially for nodes with small c_i's. This is due to the fact that by using landmarks, the approximated geodesic distance is always greater than or equal to the actual geodesic distance, therefore the denominator in Eq. (5.11) tends to be greater than the denominator in Eq. (5.9). Also from the right table from Table 5.1, we can observe that the average MAE can be reduced by about 40% if we use $\hat{c}_i^{HH.ra.l}$ instead of $\hat{c}_i^{HH.ra}$. $\hat{c}_i^{HH.ra}$ is slightly biased, and this bias may result in bias in the estimation for CDF, which we will discuss it in section 6.2.



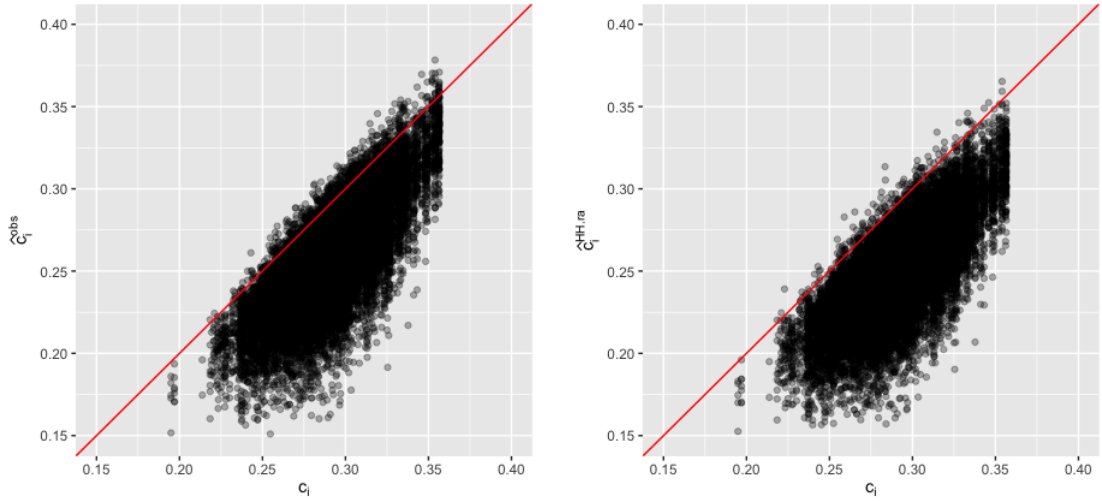
a) $\hat{c}_i^{obs}$ v.s. c_i

b) $\hat{c}_i^{HH}$ v.s. c_i

c) $\hat{c}_i^{HH.ra}$ v.s. c_i

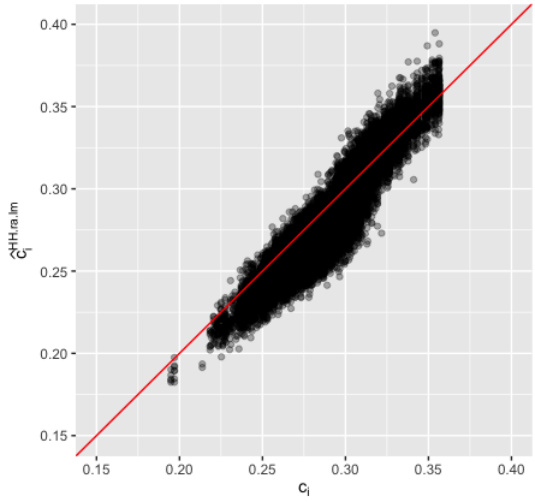
d) $\hat{c}_i^{HH.ra.l}$ v.s. c_i

Figure 5.5.1. Closeness centrality of sampled nodes: scatter plots of estimated values v.s. actual values. Network: $n = 1000$, $c.v. = 2.5$.

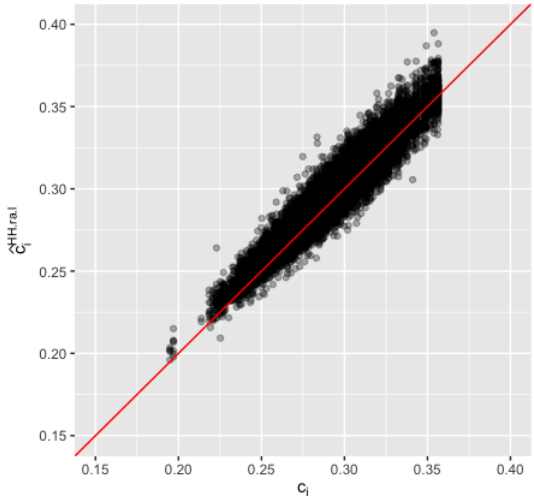


a) $\hat{c}_i^{obs}$ v.s. c_i

b) $\hat{c}_i^{HH}$ v.s. c_i



c) $\hat{c}_i^{HH.ra}$ v.s. c_i



d) $\hat{c}_i^{HH.ra.l}$ v.s. c_i

Figure 5.5.2. Closeness centrality of sampled nodes: scatter plots of estimated values v.s. actual values. Network: $n = 1000$, $c.v. = 0.7$.

Estimator	$mean(MAE^c)$	$s.d.(MAE^c)$	Estimator	$mean(MAE^c)$	$s.d.(MAE^c)$
c_i^{obs}	0.0389	0.0062	c_i^{obs}	0.0326	0.006
c_i^{HH}	0.0315	0.0192	c_i^{HH}	0.0422	0.0063
$c_i^{HH.ra}$	0.0112	0.0046	$c_i^{HH.ra}$	0.0113	0.0018
$c_i^{HH.ra.l}$	0.0091	0.0030	$c_i^{HH.ra.l}$	0.0069	0.0018

Table 5.1. Numerical comparison of estimators for closeness centrality of sampled node. Left: network of size $n = 1000$ with $c.v. = 2.5$; right: network of size $n = 1000$ with $c.v. = 0.7$.

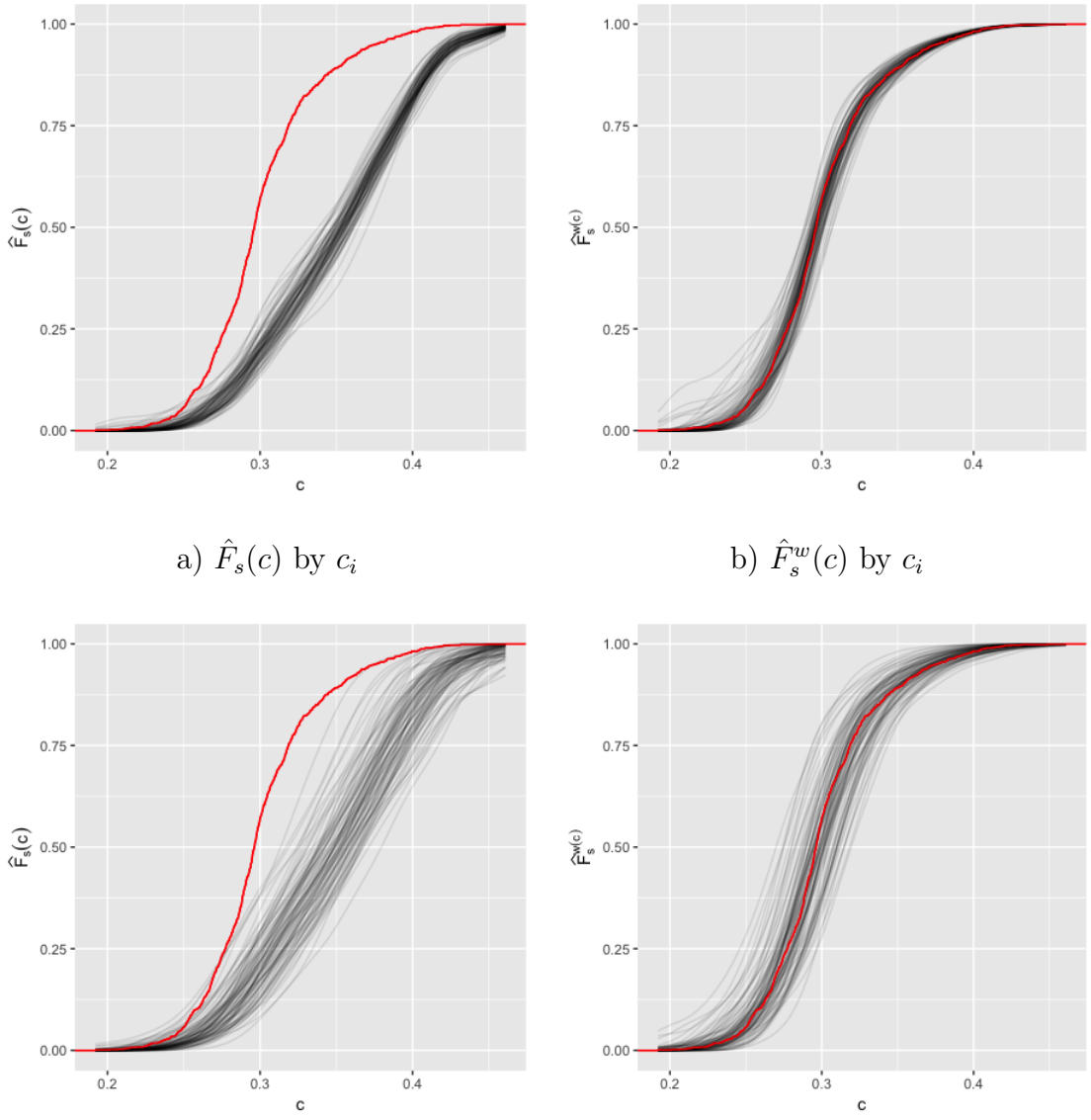
5.5.2. Estimating the population CDF of closeness centrality

In Figure 5.5.3 and Figure 5.5.4, we plot the smoothed estimated CDF's along with the actual CDF for a network with large $c.v.$ ($c.v. = 2.5$) and that for a network with small $c.v.$ ($c.v. = 0.7$). The four estimators are a) unweighted smoothed estimator $\hat{F}_s(c)$ based on c_i, b) weighted smoothed estimator $\hat{F}_s^w(c)$ based on c_i, c) unweighted smoothed estimator $\hat{F}_s(c)$ based on $\hat{c}_i$, and d) weighted smoothed estimator $\hat{F}_s^w(c)$ based on $\hat{c}_i$. Here $\hat{c}_i$ refers to the Hansen-Hurwitz ratio estimator $\hat{c}_i^{HH.ra}$ based on approximated geodesic distances.

For the network with large $c.v.$ ($c.v. = 2.5$), the unweighted estimates deviate from the actual CDF, as we can observe from plot a) and plot c) in Figure 5.5.3 . This is because nodes are sampled with unequal probabilities. More specifically, nodes with large degrees are more likely to be sampled by random walks. So the unweighted estimators are biased.

On the other hand, the weighted estimators perform much better. Both of them are unbiased, as we can observe from plot b) and plot d). The one based on c_i performs slightly better than the one based on $\hat{c}_i$ with a smaller variance. From the left table on Table 5.2, we can also notice that there's a dramatic reduction in KS if we change from the unweighted estimators to the weighted estimators, but the reduction is not very substantial if we change from the weighted estimator based on $\hat{c}_i$ to the weighted estimator based on c_i.

For the network with small $c.v.$ ($c.v. = 0.7$), the results are slightly different. For estimators based on c_i, the estimation performances are similar to those from the network with large $c.v.$. That is, the unweighted estimator is biased and the weighted estimator is unbiased with small variance. But for estimators based on $\hat{c}_i$, as we discussed in section 6.1, since $\hat{c}_i$'s are biased for some nodes with small c_i's, the estimation of CDF is affected. For the unweighted estimator based on $\hat{c}_i$, the bias from $\hat{c}_i$ and bias from unequal selection probabilities cancel out to some extent for nodes with small c_i's, so the estimated CDFs based on $\hat{c}_i$ are actually less biased than those based on c_i. For the weighted estimator based on $\hat{c}_i$, the bias from $\hat{c}_i$ makes the CDFs slightly over estimated for nodes with small c_i's. The bias from $\hat{c}_i$ makes the KS distance of the unweighted estimator based on $\hat{c}_i$ similar to the KS distance of the weighted estimator based on $\hat{c}_i$, as shown on the right table from Table 5.2.



a) $\hat{F}_s(c)$ by c_i

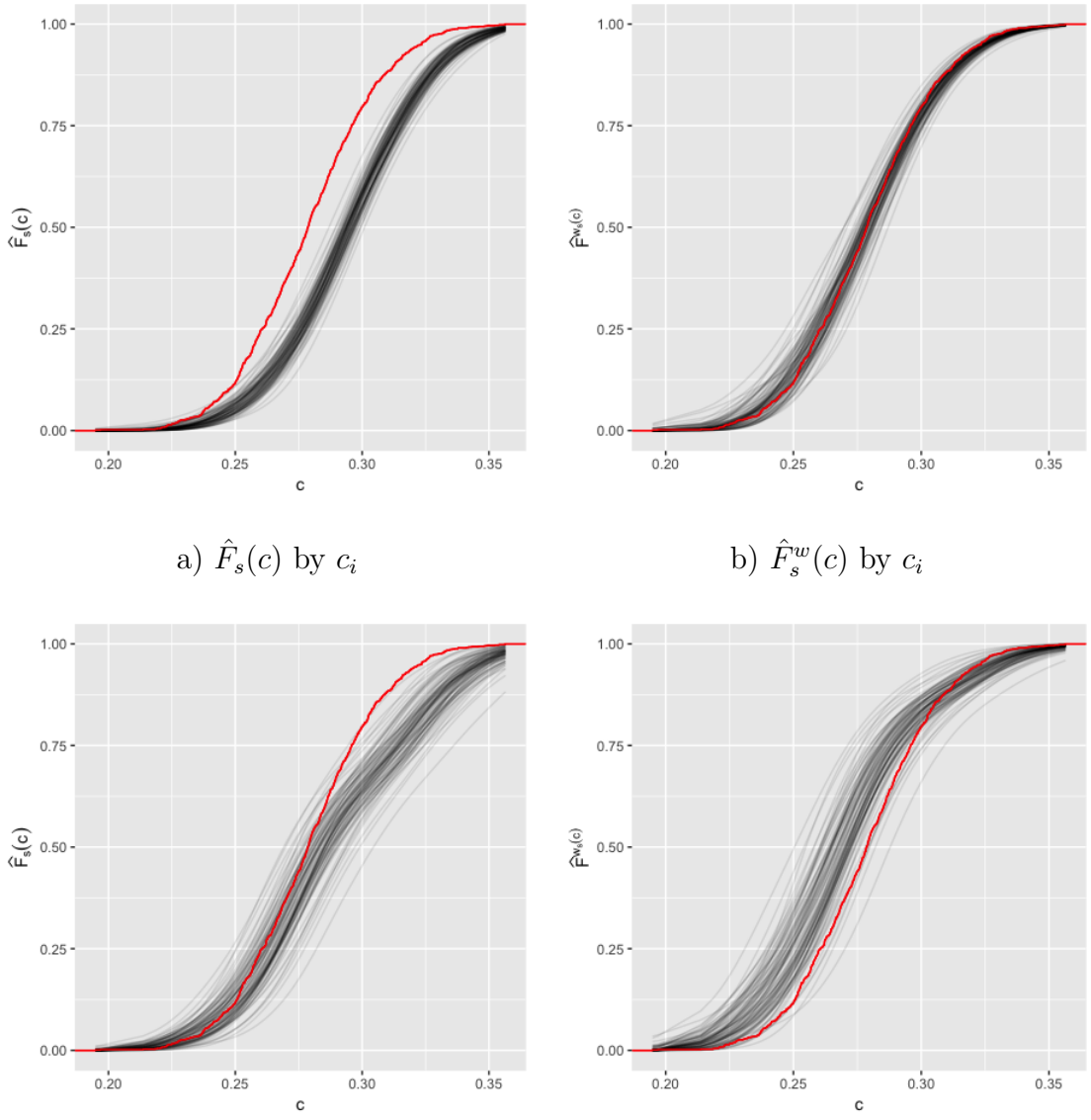
b) $\hat{F}_s^w(c)$ by c_i

c) $\hat{F}_s(c)$ by $\hat{c}_i$

d) $\hat{F}_s^w(c)$ by $\hat{c}_i$

Figure 5.5.3. Smoothed estimated population CDFs of closeness centrality.

Network: $n = 1000$, $c.v. = 2.5$.



a) $\hat{F}_s(c)$ by c_i

b) $\hat{F}_s^w(c)$ by c_i

c) $\hat{F}_s(c)$ by $\hat{c}_i$

d) $\hat{F}_s^w(c)$ by $\hat{c}_i$

Figure 5.5.4. Smoothed estimated population CDFs of closeness centrality.

Network: $n = 1000$, $c.v. = 0.7$.

Estimator	$mean(KS)$	$s.d.(KS)$	Estimator	$mean(KS)$	$s.d.(KS)$
$\hat{F}_s(c)$ by c_i	0.4689	0.0361	$\hat{F}_s(c)$ by c_i	0.2423	0.0331
$\hat{F}_s^w(c)$ by c_i	0.0725	0.0351	$\hat{F}_s^w(c)$ by c_i	0.0585	0.0293
$\hat{F}_s(c)$ by $\hat{c}_i$	0.4481	0.0786	$\hat{F}_s(c)$ by $\hat{c}_i$	0.1717	0.0456
$\hat{F}_s^w(c)$ by $\hat{c}_i$	0.1313	0.0780	$\hat{F}_s^w(c)$ by $\hat{c}_i$	0.1656	0.0725

Table 5.2. Numerical comparison of estimators for population CDF of closeness centrality. Left: network of size $n = 1000$ with $c.v. = 2.5$; right: network of size $n = 1000$ with $c.v. = 0.7$.

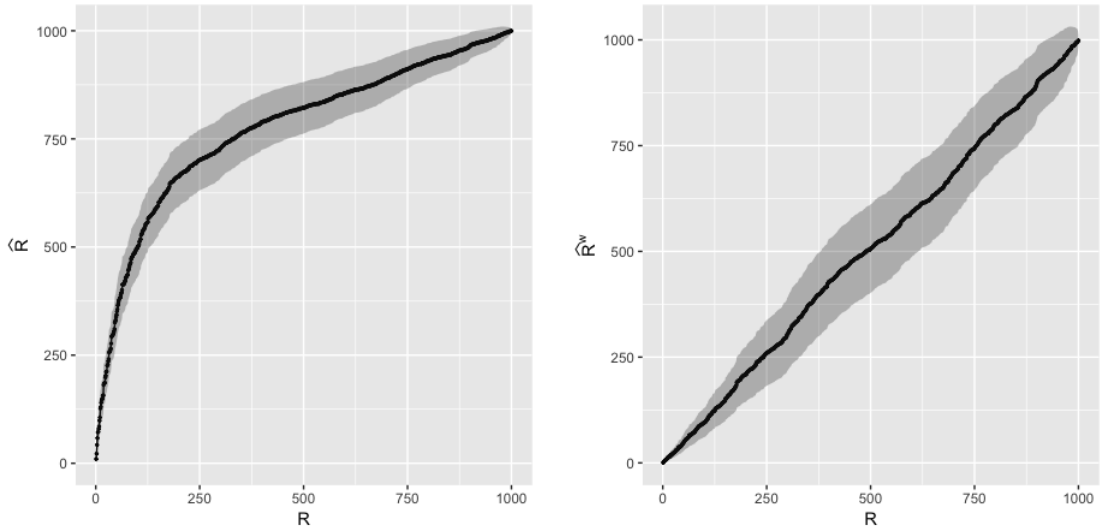
5.5.3. Estimating closeness centrality ranking of a given node

As discussed in section 4.3.3, once we have the smoothed estimated CDF, we can estimate the closeness centrality rank of a given node through Eq. (5.22) or Eq. (5.23). In this section, we use plot of average estimated rank with confidence band and $PMAE$ of $\hat{R}_i$ to evaluate the estimation performance of different estimators for R_i.

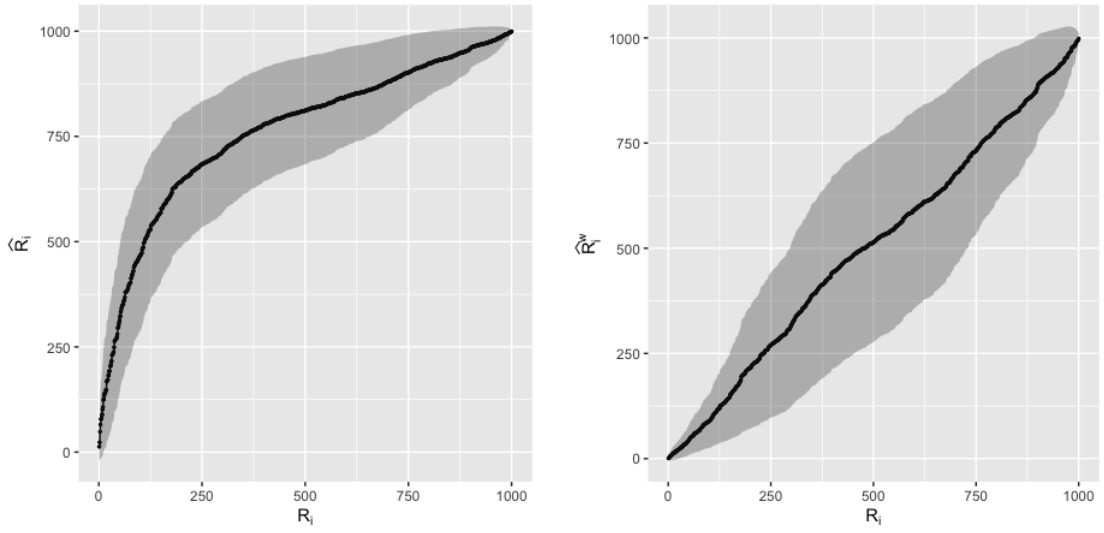
We display the plots for the network with large $c.v.$ ($c.v. = 2.5$) in Figure 5.5.5. If we base the estimator on actual closeness centrality c_i, as shown in plot a) and plot b), the unweighted estimator R_i is biased and the weighted estimator R_i^w is unbiased. This is consistent with the results of estimation of CDFs. The variances for both estimators are small as the confidence bands are narrow. If we base estimator on estimated closeness centrality $\hat{c}_i$, as shown in plot c) and plot d), we still have the same pattern for average estimated rank as we have for the one using c_i, but the variances increase since the confidence bands get wider. This indicates that for network with large $c.v.$, using the

observed geodesic distances in the induced subgraph to approximate the actual geodesic distances will keep the unbiased estimator unbiased, and slightly increase the variance of the estimator. If we refer to Table 5.3, we can also notice that the estimation performance has a dramatic improvement if we change from the unweighted estimator to the weighted estimator, but approximating geodesic distances don't have a substantial influence on the estimation performance.

When it comes to the network with small $c.v.$ ($c.v. = 0.7$), the results are similar to the results for estimated CDFs as discussed in section 6.2. The estimators based on c_i behave similarly to those for the network with large $c.v.$. One thing to notice is that the bias for the unweighted estimator is milder in this case. For the estimators based on $\hat{c}_i$, since we have two sources of bias, and they can cancel out to some extent, the unweighted estimator is less biased than expected and the weighted estimator is more biased than expected. The biases for the two estimators are in opposite directions, but if measured by $PMAE$, they have similar estimation performances as shown on Table 5.4.

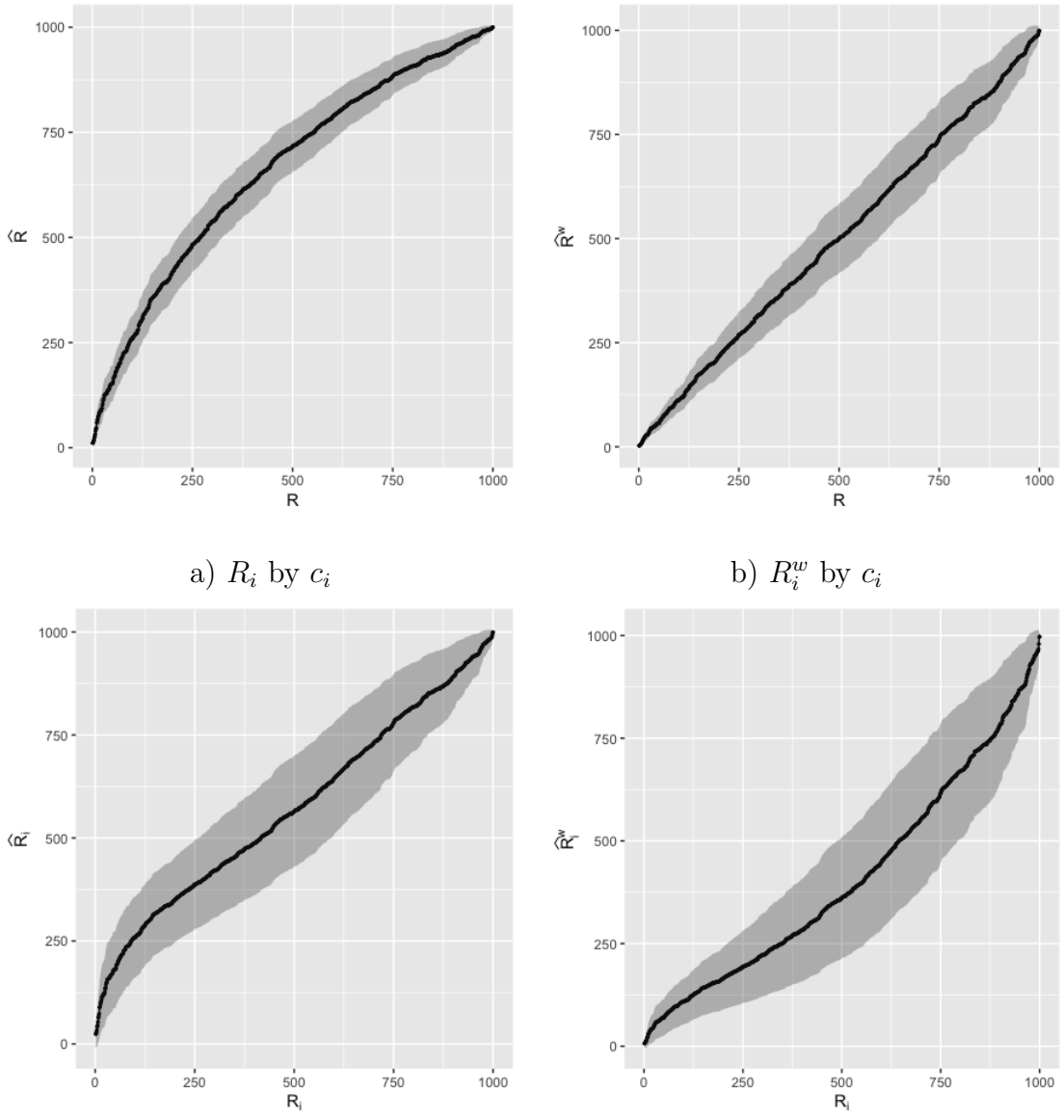


a) R_i by c_i b) R_i^w by c_i



c) R_i by $\hat{c}_i$ d) R_i^w by $\hat{c}_i$

Figure 5.5.5. Average of estimated rank with confidence interval. Network: $n = 1000$, $c.v. = 2.5$.



a) R_i by c_i

b) R_i^w by c_i

c) R_i by $\hat{c}_i$

d) R_i^w by $\hat{c}_i$

Figure 5.5.6. Average of estimated rank with confidence interval. Network: $n = 1000$, $c.v. = 0.7$.

Estimator	$mean(PMAE^R)$	$s.d.(PMAE^R)$
$\hat{R}_i$ by c_i	27.23%	2.43%
$\hat{R}_i^w$ by c_i	3.28%	1.97%
$\hat{R}_i$ by $\hat{c}_i$	25.95%	5.69%
$\hat{R}_i^w$ by $\hat{c}_i$	6.64%	4.81%

Table 5.3. Numerical comparison of estimators for closeness centrality ranking: network of size $n = 1000$ with $c.v. = 2.4$.

Estimator	$mean(PMAE^R)$	$s.d.(PMAE^R)$
$\hat{R}_i$ by c_i	16.00%	2.33%
$\hat{R}_i^w$ by c_i	2.80%	1.75%
$\hat{R}_i$ by $\hat{c}_i$	8.60%	3.65%
$\hat{R}_i^w$ by $\hat{c}_i$	10.06%	4.79%

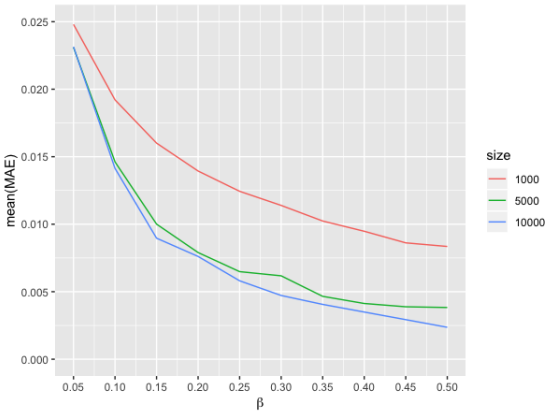
Table 5.4. Numerical comparison of estimators for closeness centrality ranking: network of size $n = 1000$ with $c.v. = 0.7$.

5.5.4. Length of Random Walks

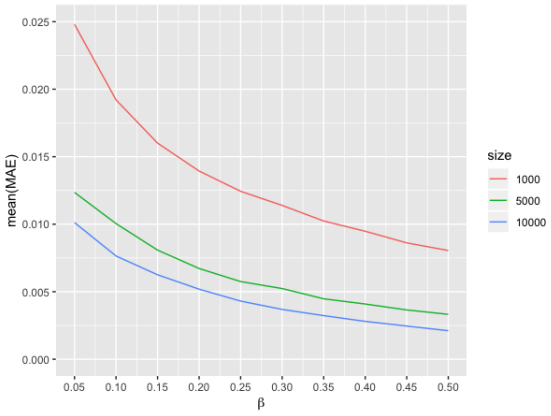
In Figure 5.5.7, we plot the estimation performance versus sampling budgets for networks with sizes of 1000, 5000, and 10000 and $c.v.$ equal to 2.4 and 0.8. We display the estimation performance of the three steps in the estimating process for closeness centrality ranking: 1) the estimation performance of closeness centrality of sampled nodes is measured by mean absolute error (MAE), 2) the estimation performance of population CDF is measured by Kolmogorov-Smirnov distance (KS), and 3) the estimation performance

of closeness centrality ranking is measured by percentage mean absolute error ($PMAE$).

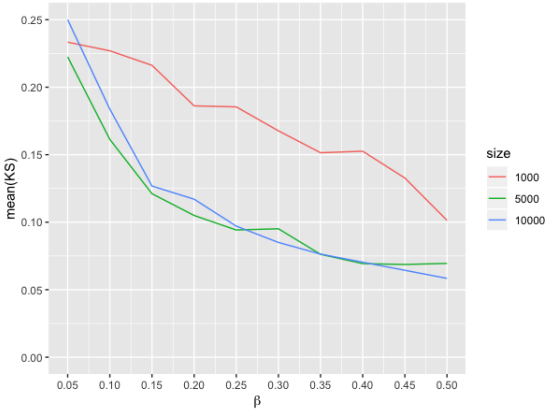
First of all, we observe that the estimation performance improves as we increase the network size from 1000 to 5000, but tends to stabilize once the network size is large enough, as we don't observe any substantial difference between $n = 5000$ and $n = 10000$. The following analysis is based on networks of size 5000 and 10000. For networks with large $c.v.$ ($c.v. = 2.4$), the estimation performance for the three steps, as we can observe from plots (a), (c), and (e), improves as we increase the sampling budget. The improvement is dramatic before $\beta = 0.2$ and moderate after $\beta = 0.2$. We can also observe this behavior pattern in the estimation performance of the first step for networks with small $c.v.$ ($c.v. = 0.8$), as shown in plot (b). Therefore, we can set $\beta = 0.2$ as the minimum sampling budget to achieve a good estimation performance for closeness centrality ranking.



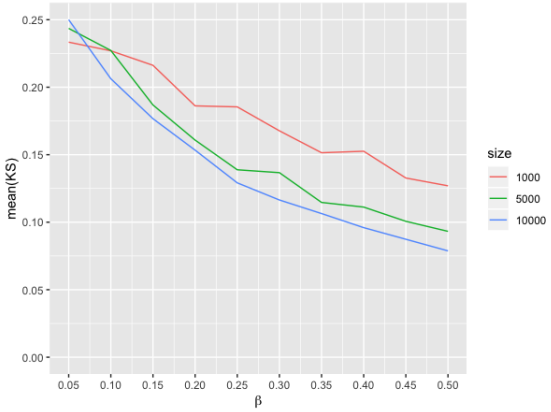
a) MAE, $c.v. = 2.4$



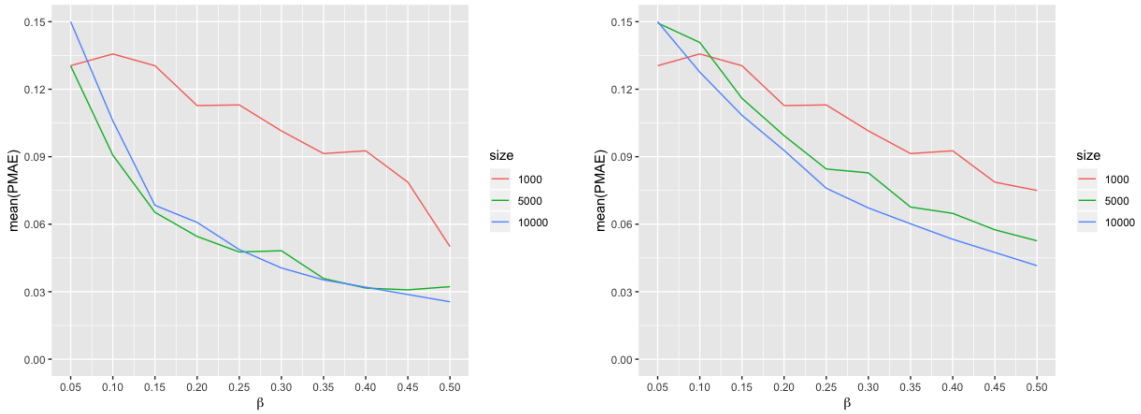
b) MAE, $c.v. = 0.8$



c) KS, $c.v. = 2.4$



d) KS, $c.v. = 0.8$



e) $PMAE$, $c.v. = 2.4$ f) $PMAE$, $c.v. = 0.8$

Figure 5.5.7. Estimating performance versus sampling budget β (low values are better).

5.6. Real networks

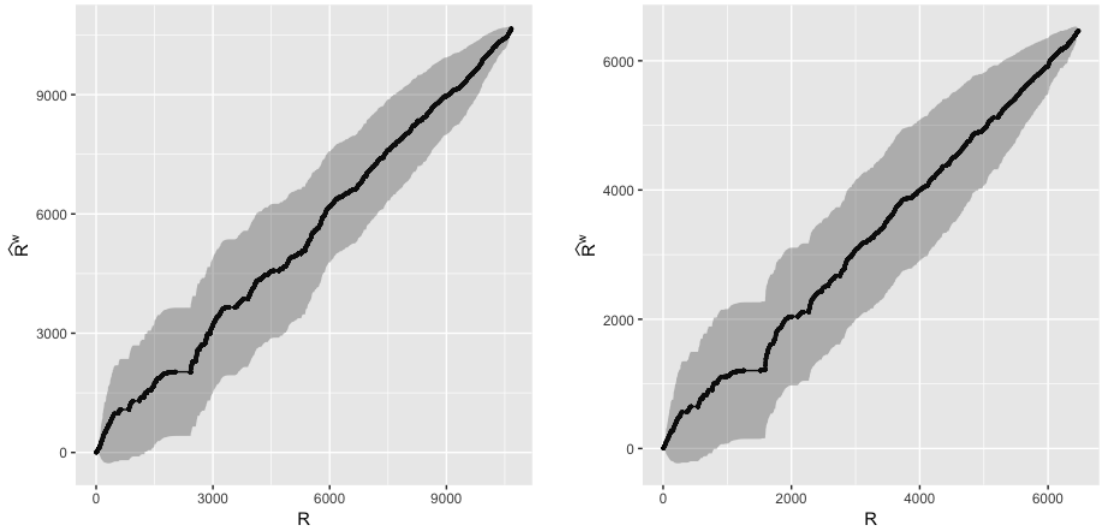
In this section, we apply our proposed estimators for closeness centrality rank to some real networks. These data are available on the SNAP (Stanford Network Analysis Project) website. To simplify the analysis, we only consider nodes in the largest connected component. The first four columns of Table 5.5 summarize the basic information for each network used in this section. These networks vary in size, number of edges, average degree, and coefficient of variation. For each network, $K = 100$ random walk samples with sampling budget $\beta = 0.3$ are taken, and an estimate is computed from each sample by the weighted rank estimator $\hat{R}_i^w$ based on c_i estimated from the Hansen-Hurwitz ratio estimator $\hat{c}_i^{HH.ra}$. To approximate the actual geodesic distances, we use the observed geodesic distances in the induced subgprah for networks with $c.v. > 2$, and use estimations through landmarks for networks with $c.v. < 2$. To measure the estimation performance,

we use the average estimated rank with confidence band as graphical tool and the MAE of $\hat{R}_i^w$ as numerical metric.

As we can observe from Figure 5.6.1 and Figure 5.6.2, the estimator is almost unbiased for all real networks. For some networks, such as Oregon, AS-733, and P2P network, there are some minor fluctuations on the plots of average estimated ranks, but there's no major deviation from a diagonal line with intercept 0 and slope 1. As for variance, the estimator is very stable for some networks such as Wiki-Vote and P2P, where the confidence bands are very narrow. While the variances for some networks are slightly larger, such as CA-HepTh and CA-GrQc, they are still in moderate range. If we refer to the numerical measure listed on the last two columns on Table 5.5, we can also notice that the average $PMAE$s for $\hat{R}_i^w$ are all under 10%, except for CA-GrQc whose average $PMAE$ is only slightly above 10% Both the graphs and the numerical metrics show that our proposed estimator $\hat{R}_i^w$ for closeness centrality ranking performs well in real networks.

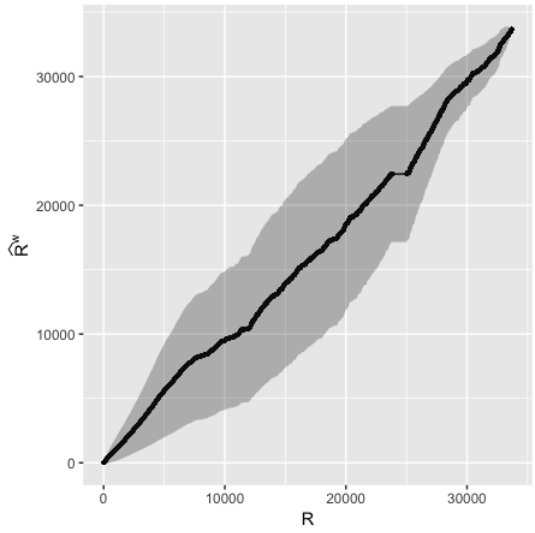
Network	nodes	edges	$< k >$	$c.v.$	$mean(PMAE^R)$	$s.d.(PMAE^R)$
Oregon	10.7K	22K	4.1	7.6	5.18%	3.26%
AS-733	6.4K	13.2K	4.3	5.8	5.67%	3.96%
Email-Enron	33.7K	361.7K	21.5	3.5	5.37%	4.14%
CA-HepPh	11.2K	235.2K	42	2.29	9.54%	6.20%
Wiki-Vote	7.1K	103.7K	29.3	2.06	3.73%	2.84%
CA-HepTh	8.6K	49.6K	11.5	1.12	7.02%	5.30%
CA-GrQc	4.2K	26.8K	12.9	1.34	11.02%	6.97%
P2P	10.9K	40K	7.4	0.9	5.97%	2.54%

Table 5.5. Basic information and estimation summary of real networks.

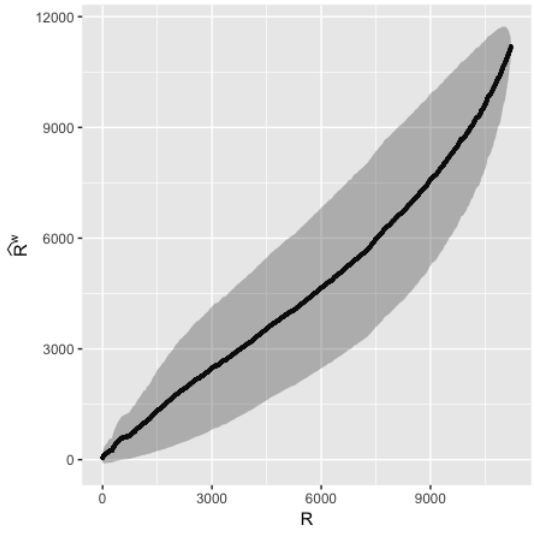


a) Oregon

b) AS-733

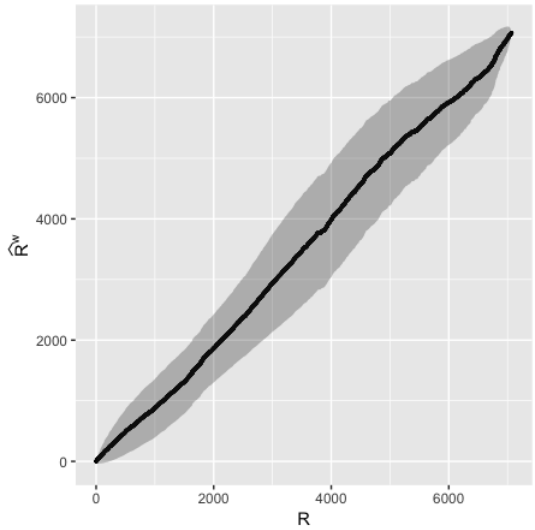


c) Email-Enron

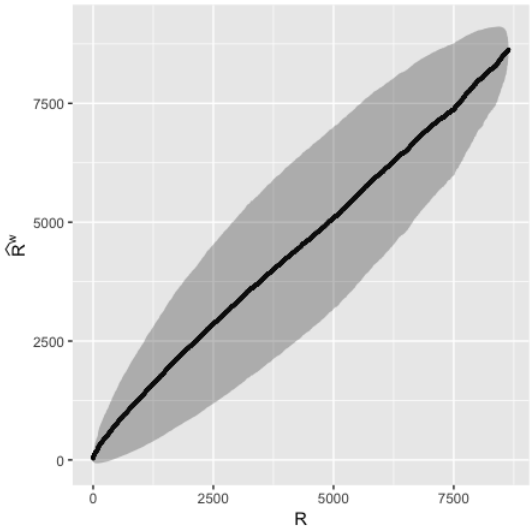


d) CA-HepPh

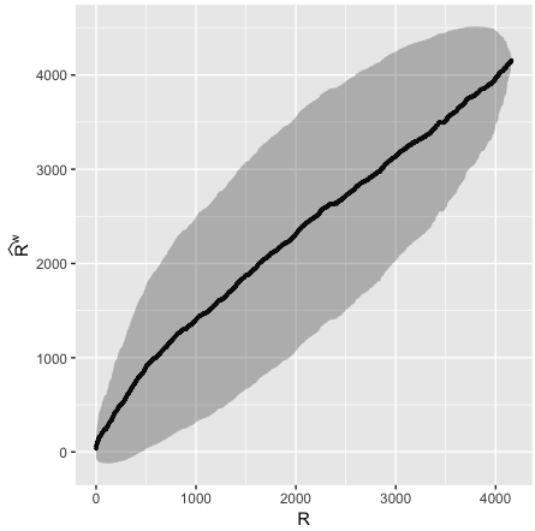
Figure 5.6.1. Average of estimated rank with confidence interval for real network (part 1).



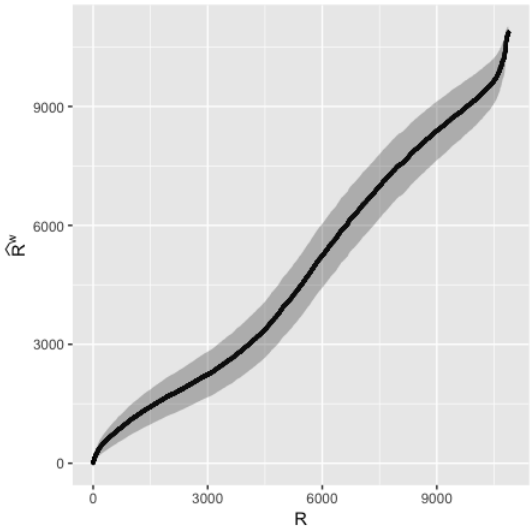
e) Wiki-Vote



f) CA-HepTh



g) CA-GrQc



h) P2P

Figure 5.6.2. Average of estimated rank with confidence interval for real network (part 2).

5.7. Summary of Results

By applying the estimators and evaluation metrics to several simulated networks and real networks, we develop the following findings:

- The Hansen-Hurwitz ratio estimator provides unbiased estimate with small variance for the closeness centrality of sampled nodes. For networks with $c.v. > 2$, using observed geodesic distances in the induced subgraph as an approximation for actual geodesic distances doesn't bring notable bias to the estimator. For network with $c.v. < 2$, using estimated geodesic distances by landmarks brings some bias to nodes with small closeness centrality, but not to a large extent.
- The weighted kernel estimator performs well in estimating CDF of closeness centrality of the population network. For network with $c.v. > 2$, the unweighted estimator is biased and the bias is reduced substantially in the weighted estimator. For networks with $c.v. < 2$, there two sources of bias in the unweighted estimator: bias from estimating closeness centrality and bias from unequal selection probabilities. The two biases cancel out to some extent to make the unweighted estimator less biased than expected. On the other hand, the weighted estimator is also slightly biased due to the existence of bias from estimating closeness centrality. The interaction of biases makes the estimation performance for the unweighted estimator and that for the weighted estimator similar for networks with $c.v. < 2$.
- The estimation performance of closeness centrality rank follows the pattern of CDF estimation. In general, the rank computed from the weighted kernel estimator for CDF provides good estimation for the rank. More specifically, the estimator for

networks with $c.v. > 2$ is unbiased with moderate standard deviation, and the estimator for networks with $c.v. < 2$ is slightly biased but not to a large extent.

5.8. Discussion and Future Work

In this chapter we proposed estimating closeness centrality ranking of a node in a network via random walk sampling. We first applied Hansen-Hurwitz ratio estimator to estimate the closeness centrality of nodes sampled by a random walk, then used weighted kernel estimator to estimate the population CDF of closeness centrality, and finally computed the estimated rank by the estimated CDF. There are three challenges in the estimating process: 1) the actual geodesic distances between sampled nodes are unknown; 2) the nodes are sampled with unequal probabilities so their unweighted empirical distribution of closeness is biased for the population CDF; 3) the empirical distribution is a discrete function with number of possible values equal to the number of nodes in the sample, so we cannot get an accurate estimation of rank for each node in the population.

We adopted a weighted estimator to deal with the unequal selection probabilities and applied kernel estimator to smooth the empirical distribution. Therefore the second and third problems are perfectly solved. To deal with the first challenge, we used different strategies for networks with different values in $c.v.$. For networks with large $c.v.(c.v. > 2)$, we used the observed geodesic distances in the induced subgraph to approximate the actual distances. It works very well as random walks have strong ability in finding the shortest paths in networks with large $c.v.$. It follows that the estimation performance for

the three stages is consistently good as expected.

For networks with small $c.v.(c.v. < 2)$, we estimated the geodesic distance between a pair of sampled nodes by taking the minimum of the sum of their geodesic distances to the pre-selected landmarks. Since the distance estimated by this algorithm is always equal to or greater than the actual distances, it brings some bias in estimating closeness centrality of sampled nodes, and thus slightly affects the estimation of CDF of closeness centrality and the ranking estimation. A possible direction for future work is to invent or apply an algorithm that will provide better estimation for geodesic distances in network with small $c.v.$, so that the bias can be further reduced in estimating closeness centrality of sampled nodes, population CDF of closeness centrality, and finally closeness centrality ranking.

CHAPTER 6

Estimation of Clustering Coefficients

6.1. Overview

The clustering coefficient of a graph measures the average probability that two neighbors of a node are themselves neighbors. However it is computationally expensive to measure the exact value of clustering coefficient of a network. In this chapter we study the problem of estimating the clustering coefficient via random walk sampling. We generalize the Hansen-Hurwitz estimator to estimate the global clustering coefficient (GCC) and the average local clustering coefficient (ALCC). By simulation studies and applications to real networks, we find that 1) If we can observe all neighbors of a sampled node and count the exact number of connections among the neighbors, the estimators for both the GCC and the ALCC will be unbiased with small variance; 2) If we can only observe the neighbors of a sampled node in the induced subgraph, we need to estimate the number of connections among the neighbors in the population graph using that number in the induced subgraph. The error from this estimating process may lead to bias in the GCC estimator and the ALCC estimator and largely increase the estimation error. Therefore in practice it is highly recommended to observe all neighbors of a sampled node to maintain a high estimation accuracy.

6.2. Definitions

The clustering coefficient of a graph measures the average probability that two randomly selected neighbors of a node are themselves neighbors. Mathematically, we define the *global clustering coefficient (GCC)* [2] (p.199), denoted by C, to be the fraction of paths of length two that are closed in the network. The global clustering coefficient C can be computed through two equivalent definitions.

For a given node i in a graph $G = (V, E)$, the number of paths of length two going through node i is the number of pairs of neighbors of i, which is $\binom{k_i}{2} = \frac{k_i(k_i-1)}{2}$, where k_i is the degree of node i. The number of closed paths of length two going through node i is the number of edges among the neighbors of i, which we denote by e_i. Therefore we can compute the global clustering coefficient C by

$$C = \frac{\sum_{i=1}^{n} e_i}{\sum_{i=1}^{n} \binom{k_i}{2}} = \frac{2\sum_{i=1}^{n} e_i}{\sum_{i=1}^{n} k_i(k_i-1)}. \tag{6.1}$$

On the other hand, a triangle in a network involves three paths of length two that are closed. Let $\lambda(G)$ denote the number of triangles in network G, and $\tau(G)$ denote the number of paths of length two that are open. So we can alternatively compute the global clustering coefficient C by

$$C = \frac{3\lambda(G)}{3\lambda(G) + \tau(G)}. \tag{6.2}$$

In addition, we can define a local clustering coefficient for a single node as the average probability that a pair of the node's neighbors are neighbors of one another. Mathematically, we define the *local clustering coefficient* [2] (p.201) of node i, denoted by c_i, to be the fraction of connected neighbors of node i, i.e.,

$$c_i = \frac{e_i}{\binom{k_i}{2}}. \tag{6.3}$$

Based on that, Watts and Strogatz [26] proposed another measure of clustering coefficient for the entire network, which is the unweighted mean of local clustering coefficients for each node, and we call it *average local clustering coefficient (ALCC):*

$$C_{WS} = \frac{1}{n}\sum_{i=1}^{n} c_i. \tag{6.4}$$

Note that the ALCC C_{WS} defined by Eq.(6.4) is an unweighted average of LCCs, and the GCC C defined by Eq.(6.1) is a weighted average of LCCs with weight equal to $\frac{\binom{k_i}{2}}{\sum_{j=1}^{n}\binom{k_j}{2}}$ for node i with degree k_i. To distinguish the two estimators for GCC based on two definitions, we call it estimator based on weighted average LCC if it's based on Eq.(6.1) and estimator based on triangles if it's based on Eq.(6.2.)

6.3. Estimating Method

In this section, we develop estimating methods for the global clustering coefficient C through Def. (6.1) and Def. (6.2), and for the average local clustering coefficient C_{WS} through Def. (6.4), by applying generalized Hansen-Hurwitz estimator as defined in Eq. (2.11).

6.3.1. Estimating Global Clustering Coefficient Based on Weighted Average of LCC

Let $s = \{X(1), X(2), ..., X(H)\}$ denote the set of sequences of nodes visited by H random walks, including duplicates, and let $|s| = H \cdot B$ denote the size of s. Let $I(X_b^{(h)} = i)$ denote an indicator variable taking the value 1 if node i is visited at the b^{th} step in the h^{th} random walk, and zero otherwise. Let $q_i = \sum_{h=1}^{H} \sum_{b=1}^{B} I(X_b^{(h)} = i),\ i = 1, ..., n$, denote the number of times node i appears in sample s and let $\boldsymbol{q} = (q_1, q_2, ..., q_n)^T$.

Suppose for a sampled node i we can observe the degree k_i and the number of edges e_i among the neighbors of i. Let $t^e = \sum_{i=1}^{n} e_i$ denote the total number of closed paths of length two and let $t^k = \sum_{i=1}^{n} \frac{k_i(k_i-1)}{2}$ denote the total number of paths of length two, and notice that

$$C = \frac{t^e}{t^k}. \tag{6.5}$$

Using the generalized Hansen-Hurwitz estimator, we can estimate t^e and t^k by

$$\hat{t}^e = \sum_{i=1}^{n} \frac{q_i e_i}{E(q_i)}, \tag{6.6}$$

and

$$\hat{t}^k = \sum_{i=1}^{n} \frac{q_i k_i (k_i - 1)/2}{E(q_i)}, \tag{6.7}$$

and therefore we can estimate C through Def. (6.1) by

$$\frac{\hat{t}^e}{\hat{t}^k} = \frac{\sum_{i=1}^{n} \frac{q_i e_i}{E(q_i)}}{\sum_{i=1}^{n} \frac{q_i k_i (k_i-1)/2}{E(q_i)}}. \tag{6.8}$$

Let $\boldsymbol{p} = (p_1, p_2, ..., p_n)^T$, where $p_i = \frac{k_i}{K}$ with $K = \sum_w k_w$. As shown in [35], a single random walk $\{X_t\}$ on a connected graph $G = (V, E)$ with at least one triangle is an irreducible and aperiodic Markov chain with a stationary distribution $\boldsymbol{p}$. According to Anderson's (1989) results for irreducible and aperiodic Markov chains, $E(\boldsymbol{q}) = \boldsymbol{p}t$. Applying this in our case, we will have $E(q_i) = |s|\frac{k_i}{K}$, and therefore we can estimate C through Def. (6.1) by

$$\frac{\hat{t}^e}{\hat{t}^k} = \frac{\sum_{i=1}^{n} q_i e_i / k_i}{\sum_{i=1}^{n} q_i (k_i - 1)/2}. \tag{6.9}$$

In some real cases however, we only observe the neighbors of node i when they are also visited by the random walk, and therefore we can observe the connections between those neighbors. In other words, the actual value of e_i can't be observed by random walk sampling and thus needs to be estimated. Recall that G^* denotes the subgraph induced by the sampled nodes. Let k_i^* denote the observed degree of sampled node i in G^* and let e_i^* denote the observed connections between neighbors of node i in G^*. We propose estimating e_i by

$$\hat{e}_i = \frac{k_i}{k_i^*} e_i^*. \tag{6.10}$$

We define the Hansen-Hurwitz estimator for global clustering coefficient C based on weighted average of LCC through Def. (6.1) by e_i as

$$\hat{C}^{AE}_{HH1} = \frac{\sum_{i=1}^{n} q_i e_i / k_i}{\sum_{i=1}^{n} q_i (k_i - 1)/2}, \tag{6.11}$$

and define the Hansen-Hurwitz estimator for global clustering coefficient C based on weighted average of LCC through Def. (6.1) by $\hat{e}_i$ as

$$\hat{C}_{HH1} = \frac{\sum_{i=1}^{n} q_i \hat{e}_i / k_i}{\sum_{i=1}^{n} q_i (k_i - 1)/2}. \tag{6.12}$$

6.3.2. Estimating Global Clustering Coefficient Based on Triangles

Let w, $w = 1, ..., M$, represent a triple of nodes (i, j, v), $i = 1, ..., n-2$, $j = i+1, ..., n-1$, $v = j+1, ..., n$, in the population graph. Suppose we have a single random walk $X = \{X_1, X_2, ..., X_B\}$, define $S = \{(X_{b_1}, X_{b_2}, X_{b_3}), b_1 < b_2 < b_3, b_1, b_2, b_3 \in \{1, ..., B\}, X_{b_1} \neq X_{b_2} \neq X_{b_3}\}$ to be the sequence of triplets visited by the random walk. The three nodes in each triplet are distinct, but any node may appear multiple times in the random walk, therefore it's also possible to have duplicates of triplets in S. Suppose we have multiple random walks, i.e., $H > 1$, recall that $s = \{X(1), X(2), ..., X(H)\}$ with $X(h) = (X_1^{(h)}, ..., X_B^{(h)})$, $h = 1, ..., H$, represents the set of sequences of nodes visited by H random walks, including duplicates. Define $S = \{(X_{b_1}^{(h_1)}, X_{b_2}^{(h_2)}, X_{b_3}^{(h_3)}) : h_1 \leq h_2 \leq h_3, h_1, h_2, h_3 \in \{1, ..., H\}, b_i < b_j \text{ if } h_i = h_j, i, j = 1, 2, 3, i \neq j, b_1, b_2, b_3 \in \{1, ..., B\}, X_{b_1}^{(h_1)} \neq X_{b_2}^{(h_2)} \neq X_{b_3}^{(h_3)}\}$ to be the sequence of triplets visited by the H random walks.

Define $Q_w = q_i q_j q_v,\ i = 1, ..., n-2,\ j = i+1, ..., n-1,\ v = j+1, ..., n$, as the number of times triplet w appears in S, and let $|S| = \sum_{w=1}^{M} Q_w$ denote the size of S. Notice that there may be duplicates in the sample of nodes, but to form a triplet, we only include three distinct nodes, therefore $|S|$ is a random variable with $|S| = \binom{|s|}{3} - \sum_{i=1}^{n}\binom{q_i}{3} - \sum_{i=1}^{n}[\binom{q_i}{2}\sum_{j\neq i} q_j]$. Define $\psi_w = \frac{E(Q_w)}{E(|S|)}$ and assume $0 < E(Q_w) < |S|$ $\forall w$, therefore $0 < \psi_w < 1\ \forall w$. Since $\sum_{w=1}^{M} Q_w = |S|$, $\sum_{w=1}^{M} \psi_w = 1$. Therefore, the ψ_w's form a probability distribution over the M triplets.

Let A^* denote the adjacency matrix of the induced subgraph G^*. Let $y_w^\lambda,\ w \in \{1, ..., M\}$, denote an indicator variable taking value $y_w^\lambda = 1$ if triple $w = (i, j, v)$ forms an triangle (has three edges) in G^* and zero otherwise. Let $y_w^\tau,\ w \in \{1, ..., M\}$, denote an indicator variable taking value $y_w^\tau = 1$ if triple $w = (i, j, v)$ forms a path of length two that is open (has two edges) in G^* and zero otherwise.

The generalized Hansen-Hurwitz estimator for $\lambda(G)$ is

$$\hat{\lambda}(G) = \sum_{w=1}^{M} \frac{Q_w y_w^\lambda}{E(Q_w)}, \tag{6.13}$$

where

$$y_w^\lambda = A_{ij}^* A_{jv}^* A_{iv}^*. \tag{6.14}$$

The generalized Hansen-Hurwitz estimator for $\tau(G)$ is

$$\hat{\tau}(G) = \sum_{w=1}^{M} \frac{Q_w y_w^{\tau}}{E(Q_w)}, \tag{6.15}$$

where

$$y_w^{\tau} = A_{ij}^* A_{jv}^* (1 - A_{iv}^*) + A_{ij}^* (1 - A_{jv}^*) A_{iv}^* + (1 - A_{ij}^*) A_{jv}^* A_{iv}^* \tag{6.16}$$

If we already know the value of y_w^{λ} for triplet w, we can also compute y_w^{τ} by

$$y_w^{\tau} = A_{ij}^* A_{jv}^* + A_{ij}^* A_{iv}^* + A_{jv}^* A_{iv}^* - 3y_w^{\lambda}. \tag{6.17}$$

According to Def. (6.2), the generalized Hansen-Hurwitz estimator for C is

$$\frac{3\hat{\lambda}^(G)}{3\hat{\lambda}^(G) + \hat{\tau}^(G)} = \frac{3\sum_{w=1}^{M} \frac{Q_w y_w^{\lambda}}{E(Q_w)}}{3\sum_{w=1}^{M} \frac{Q_w y_w^{\lambda}}{E(Q_w)} + \sum_{w=1}^{M} \frac{Q_w y_w^{\tau}}{E(Q_w)}}. \tag{6.18}$$

According to the simulation results in Section 6.2, $E(Q_w) \propto k_i k_j k_v$ on average in the long run. Therefore, we define the Hansen-Hurwitz estimator for global clustering coefficient C based on triangles through Def. (6.2) as

$$\hat{C}_{HH2} = \frac{3\sum_{w=1}^{M} \frac{Q_w y_w^{\lambda}}{k_i k_j k_v}}{3\sum_{w=1}^{M} \frac{Q_w y_w^{\lambda}}{k_i k_j k_v} + \sum_{w=1}^{M} \frac{Q_w y_w^{\tau}}{k_i k_j k_v}}. \tag{6.19}$$

According to simulations in Section 6.2, $\psi_w \propto k_i k_j k_v$ on average across $K = 100$ random walks with $B = 80$ steps in a network of size $n = 100$. This implies $\psi_w = \alpha k_i k_j k_v$, where $\alpha = \sum_{i \neq j \neq v} k_i k_j k_v$. However ψ_w deviates from $\alpha k_i k_j k_v$ in a single random walk with moderate length. This is due to the fact that nodes are not sampled independently

in random walk sampling. In fact, since random walk sampling is implemented by going through connections among nodes, a triplet with at more connection will be more likely to be sampled than a triplet with less or no connection, even if the the products of nodes degrees for the two triplets are equal. So the empirical ψ_w for the former tends to be higher than than $\alpha k_i k_j k_v$ and the empirical ψ_w for the latter tends to be lower than than $\alpha k_i k_j k_v$. Therefore, if we use $\alpha k_i k_j k_v$ for ψ_w in the estimation process, $\lambda(G)$ and $\tau(G)$ will be biased. More specifically, since the triplets we use to compute $\lambda(G)$ have three connections and the triplets we use to compute $\tau(G)$ have two connections, $\alpha k_i k_j k_v$ tends to be smaller than the empirical ψ_w for those triplets, and thus $\lambda(G)$ and $\tau(G)$ will be over estimated.

According to Eq. (6.18), $\hat{C}_{HH2}$ will be unbiased if $\frac{E(\hat{\lambda}(G))}{\lambda(G)} = \frac{E(\hat{\tau}(G))}{\tau(G)}$, even though $\hat{\lambda}(G)$ and $\hat{\tau}(G)$ are biased. But based on simulations results presents in Table 6.2, $\frac{E(\hat{\lambda}(G))}{\lambda(G)} \neq \frac{E(\hat{\tau}(G))}{\tau(G)}$, so $\hat{C}_{HH2}$ will be biased.

6.3.2.1. Application of Bootstrap to Correct for Bias

In order to correct for bias of $\hat{C}_{HH2}$, we apply a bootstrap. The bias for $\hat{C}_{HH2}$ is

$$bias(\hat{C}_{HH2}) = E(\hat{C}_{HH2}) - C, \tag{6.20}$$

and the relative bias for $\hat{C}_{HH2}$ is

$$r.bias(\hat{C}_{HH2}) = \frac{E(\hat{C}_{HH2}) - C}{C}. \tag{6.21}$$

For a subgraph G^* of size n^* induced by sample s, let C_s denote its actual global clustering coefficient defined by Def. (6.2). To apply the bootstrap, we take K random walk subsamples from G^* with $B = \beta n^*$, and from each of the subsamples we estimate C_s by Eq. (6.19), denoted as $\hat{C}_s^b$, $b = 1, ..., K$.

For each sample s, we define the estimated bias for $\hat{C}_{HH2}$ as

$$\hat{bias}(\hat{C}_{HH2}) = \frac{1}{K}\sum_{b=1}^{K}\hat{C}_s^b - C_s, \tag{6.22}$$

and the estimated relative bias for $\hat{C}_{HH2}$ as

$$\hat{r.bias}(\hat{C}_{HH2}) = \frac{\frac{1}{K}\sum_{b=1}^{K}\hat{C}_s^b - C_s}{C_s}. \tag{6.23}$$

We can correct for bias of $\hat{C}_{HH2}$ through either bias or relative bias. We define the Hansen-Hurwitz estimator for C based on triangles with bias correction by bootstrap as

$$\hat{C}_{HH2}^{B1} = \hat{C}_{HH2} - \hat{bias}(\hat{C}_{HH2}), \tag{6.24}$$

and define the Hansen-Hurwitz estimator for C based on triangles with relative bias correction by bootstrap as

$$\hat{C}_{HH2}^{B2} = \frac{\hat{C}_{HH2}}{\hat{r.bias}(\hat{C}_{HH2}) + 1}. \tag{6.25}$$

6.3.3. Estimating Average Local Clustering Coefficient

Let $t^c = \sum_{i=1}^{n} c_i$ denote the sum of local clustering coefficient. Using the generalized Hansen-Hurwitz estimator, we can estimate t^c by

$$\hat{t}^c = \sum_{i=1}^{n} \frac{q_i c_i}{E(q_i)}, \tag{6.26}$$

where $c_i = e_i / \binom{k_i}{2}$, and estimate network size n by

$$\hat{n} = \sum_{i=1}^{n} \frac{q_i}{E(q_i)}. \tag{6.27}$$

According to Anderson [10] and Zheng and Spencer [35], $E(q_i) = |s|\frac{k_i}{K}$, therefore we can estimate C_{WS} by

$$\frac{\hat{t}^c}{\hat{n}} = \frac{\sum_{i=1}^{n} q_i c_i / k_i}{\sum_{i=1}^{n} q_i / k_i}. \tag{6.28}$$

In some real cases, we cannot observe e_i and need to estimate it through $\hat{e}_i = \frac{k_i}{k_i^*} e_i^*$. We define the Hansen-Hurwitz estimator for average local clustering coefficient through Def.(6.4) based on e_i as

$$\hat{C}_{WS}^{AE} = \frac{\sum_{i=1}^{n} q_i c_i / k_i}{\sum_{i=1}^{n} q_i / k_i}, \tag{6.29}$$

and define the Hansen-Hurwitz estimator for average local clustering coefficient through Def.(6.4) based on $\hat{e}_i$ as

$$\hat{C}_{WS} = \frac{\sum_{i=1}^{n} q_i \hat{c}_i / k_i}{\sum_{i=1}^{n} q_i / k_i}, \tag{6.30}$$

where $\hat{c}_i = \hat{e}_i / \binom{k_i}{2}$.

6.4. Evaluation Metrics

To evaluate the performance of an estimator, we take K random walk samples from the population graph G, compute the estimate from each sample, and estimate the sampling distribution by a histogram. On the numerical side, in addition to bias and standard deviation, we use normalized root mean square error (NRMSE) to get an overall assessment for the estimator.

6.4.1. Histograms

We plot histograms of the estimates to show the estimated probability distribution of the estimates. In general, a symmetric and unimodal histogram will be ideal. A more concentrated histogram indicates smaller variance of the estimator than a histogram which is more spread out. We also plot the true value of clustering coefficient (red vertical dash line) along with the histogram to indicate if the estimator is biased. In Figure 6.4.1 we plot the histogram of $\hat{C}_{HH1}$ based on $K = 100$ estimates, and the true value of global clustering coefficient $C = 0.0085$.

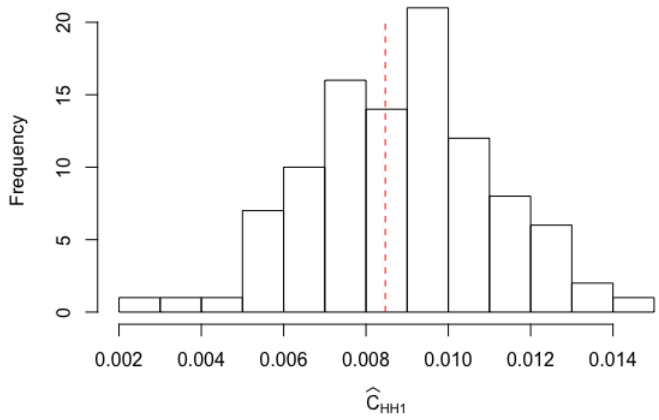


Figure 6.4.1. Histogram of $\hat{C}_{HH1}$, true clustering coefficient $C = 0.0085$.

6.4.2. Normalized Root Mean Square Error (NRMSE)

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

$$NRMSE(\hat{\theta}) = \frac{\sqrt{E(\hat{\theta} - \theta)^2}}{\theta}. \tag{6.31}$$

6.5. Simulation Study

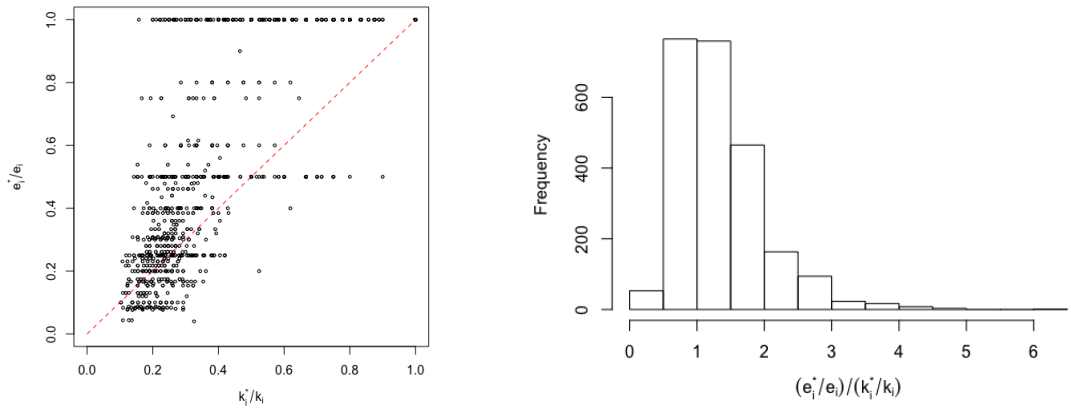
In this section, we present several simulation studies to assess the performance of methods proposed in section 6.3. More specifically, by simulations, we 1) assess the adequacy of assumption $\frac{e_i^*}{e_i} = \frac{k_i^*}{k_i}$ and compare estimation performance based on e_i and $\hat{e}_i$ for $\hat{C}_{HH1}$ and $\hat{C}_{WS}$; 2) discuss the 'draw-by-draw' probabilities for triplets and their influence on bias for $\hat{C}_{HH1}$; 3) evaluate the effectiveness of bootstrap to bias-correction; 4)

compare the performance of estimators proposed in Section 4 and the estimator proposed by [11]; 5) analyze the effect of different sampling budget (β) and find the best sampling design for $\hat{C}_{HH1}$ and $\hat{C}_{WS}$.

6.5.1. Influence of Estimating number of neighbors on Estimation Performance

In this simulation, we first check the adequacy of assumption $\frac{e_i^*}{e_i} = \frac{k_i^*}{k_i}$, and then evaluate the effect of error $\hat{e}_i$ by comparing estimation performance of $\hat{C}_{HH1}^{AE}$ and $\hat{C}_{HH1}$. In order to make histograms and compute $NRMSE$ for the two estimators, $K = 100$ random walk samples are taken from a scale-free network of size $n = 1000$. For each random walk sample, a single random walk of length $B = 300$ is implemented. The true value of clustering coefficient of the population network is $C = 0.0085$.

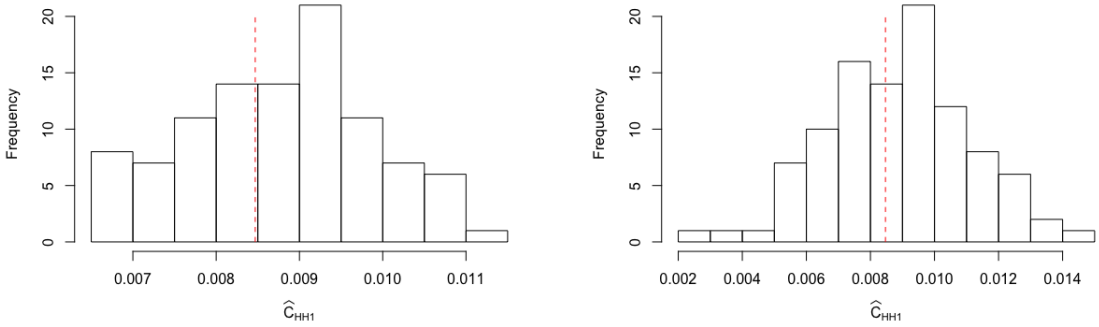
Since Eq. (6.10) is based on assumption $\frac{e_i^*}{e_i} = \frac{k_i^*}{k_i}$, we want to check if this equality holds for most sampled nodes through simulation. In Figure 6.5.1 (a), we plot $\frac{e_i^*}{e_i}$ versus $\frac{k_i^*}{k_i}$ for all sampled nodes. As we can observe, there's a lot of dispersion in the plot. In Figure 6.5.1 (b), we plot the histogram of ratio $\frac{e_i^*/e_i}{k_i^*/k_i}$. We can see that in this case, for a large proportion of the sampled nodes, this ratio is around 1. More specifically, for 55% of the sampled nodes, ratio $\frac{e_i^*/e_i}{k_i^*/k_i}$ is within range $(0.5, 1.5)$.



(a) Plot of $\frac{e_i^*}{e_i}$ versus $\frac{k_i^*}{k_i}$ (b) Histogram of $\frac{e_i^*/e_i}{k_i^*/k_i}$

Figure 6.5.1. Left: Plot of $\frac{e_i^*}{e_i}$ versus $\frac{k_i^*}{k_i}$; right: Histogram of $\frac{e_i^*/e_i}{k_i^*/k_i}$.

The error in $\hat{e}_i$ may increase bias and standard deviation in the estimating process. In Figure 6.5.2 we plot (a) the histogram of $\hat{C}_{HH1}$ and (b) the histogram of $\hat{C}_{HH1}^{AE}$. In Table 6.1, we list the numerical comparisons of the two estimators. As we can observe, in this case both estimators are unbiased, but $\hat{C}_{HH1}$ has larger standard deviation than $\hat{C}_{HH1}^{AE}$. In Section 7, we will show that for some real networks, using estimated e_i increases both bias and standard deviation of the the estimator. Therefore it is preferable to use actual e_i if it is observable.



(a) Histogram of $\hat{C}_{HH1}$ (b) Histogram of $\hat{C}^{AE}_{HH1}$

Figure 6.5.2. Left: histogram of $\hat{C}_{HH1}$; right: histogram of $\hat{C}^{AE}_{HH1}$, true global clustering coefficient $C = 0.0085$.

Estimator	Bias	SD	NRMSE
$\hat{C}^{AE}_{HH1}$	0.0002	0.0011	0.134
$\hat{C}_{HH1}$	0.0003	0.0022	0.262

Table 6.1. Numerical comparison of $\hat{C}^{AE}_{HH1}$ and $\hat{C}_{HH1}$, true global clustering coefficient $C = 0.0085$.

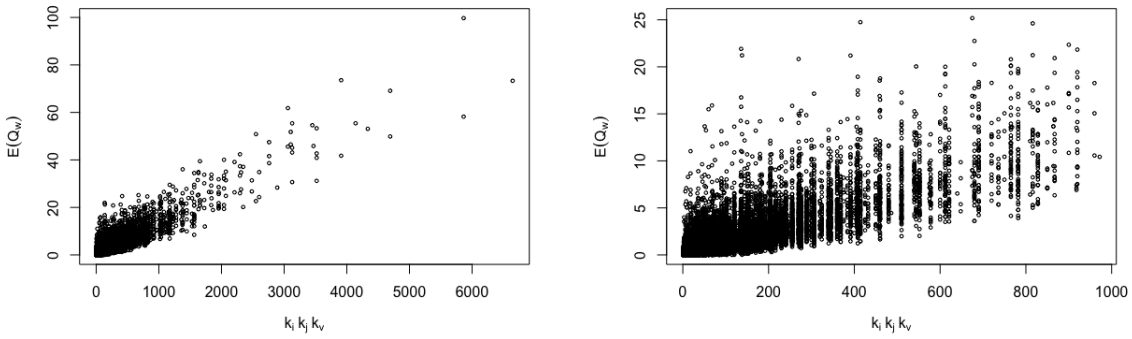
6.5.2. 'Draw-by-draw' Probabilities for Triplets

In order to find the theoretical ψ_w on average in the long run, we did the following simulation. From a scale-free network of size $n = 100$, we take $K = 100$ random walk samples, each of which takes $B = 80$ steps. For each triple $w = (i, j, v)$ in G, we plot $E(Q_w)$, its average number of occurrence in the 100 random walks, i.e., total number of occurrence in the 100 random walks divided by 100, versus $k_i k_j k_v$, the product of

corresponding node degrees, as shown in Figure 6.5.3. As we can observe, there's roughly a linear relationship between $E(Q_w)$ and $k_ik_jk_v$, and $cor(E(Q_w), k_ik_jk_v) = 0.92$. Therefore, we can infer that $E(Q_w) \propto k_ik_jk_v$ in the long run. Since ψ_w is defined as $\frac{E(Q_w)}{E(|S|)}$ and ψ_w forms a probability distribution, we have

$$\psi_w = \alpha k_i k_j k_v, \tag{6.32}$$

where $\alpha = \sum_{i \neq j \neq v} k_ik_jk_v$, and we call ψ_w the 'draw-by-draw' probability for triplet w.



(a) for all triplets (b) for triplets where $k_ik_jk_v < 1000$

Figure 6.5.3. Plot of Q_w versus $k_ik_jk_v$.

However, there two problems with using $\alpha k_ik_jk_v$ for ϕ_w in the estimating process. First, as we can observe from plot (b), the variation of $E(Q_w)$ is very large when $k_ik_jk_v$ is small. Second, as discussed in section 4.3.2, in a single random walk, unless the random walk is extremely long (for example, a random walk with $B = 2000$ steps from a network of size $n = 100$), the empirical ψ_w is not equal to $\alpha k_ik_jk_v$. Due to the nature of random walk sampling, the empirical ψ_w for a triplet with more edges between its nodes tends

to be larger than $\alpha k_i k_j k_v$ and the empirical ψ_w for a triplet with fewer edges between its nodes tends to be smaller than $\alpha k_i k_j k_v$, even if their products of degrees are equal. In Figure 6.5.4, we demonstrate this phenomenon using a small network with only 5 nodes $\{a, b, c, d, e\}$. Consider a extreme case where the random walk only takes 3 steps, then ϕ_w for triplet $w = \{a, b, d\}$ is zero while ψ_w for triplet $w = \{a, b, c\}$ is greater than zero, even if the products of node degrees for these two triplets are both 4. This is because there are two connection among $w = \{a, b, c\}$ and the random walk is able to traverse the three nodes in 3 steps, while for $w = \{a, b, d\}$ there's only one connection and the the random walk is not able to visit all three nodes in 3 steps.

Figure 6.5.4. A small network with 5 nodes.

Because of that, if we use $\alpha k_i k_j k_v$ for ψ_w in the estimation process, $\lambda(G)$ and $\tau(G)$ will be over estimated. To verify this, we computed estimates for $\lambda(G)$ and for $\tau(G)$ from the $K = 100$ random walks samples with $B = 80$ from a scale-free network of size $n = 100$, and listed the summary in Table 6.2. As we can observe, the relative bias of $\hat{\tau}$ is greater than than the relative bias of $\hat{\lambda}$, i.e., $\frac{E(\hat{\tau})}{\tau} > \frac{E(\hat{\lambda})}{\lambda}$, therefore C tend to be underestimated according to Eq.(6.18).

Estimator $\hat{\theta}$	θ	$E(\hat{\theta})$	$\frac{E(\hat{\theta})}{\theta}$	Relative Bias
$\hat{\lambda}$	31	75	2.42	142%
$\hat{\tau}$	951	3472	3.65	265%

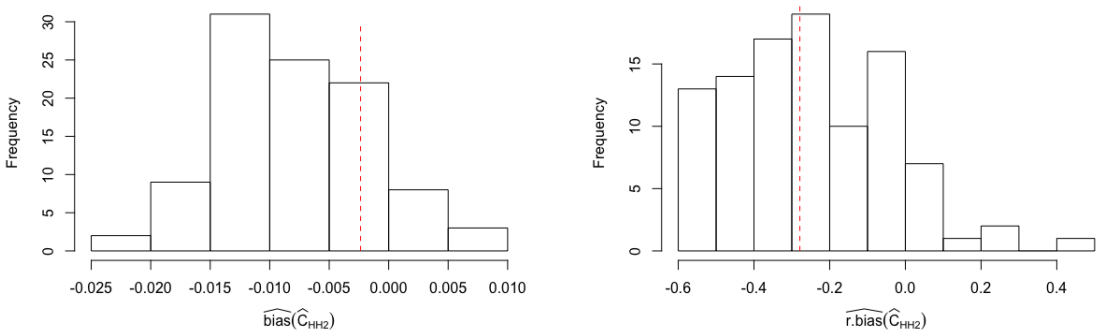
Table 6.2. Relative bias for $\hat{\lambda}(G)$ and $\hat{\tau}(G)$.

6.5.3. Bootstrap Bias Correction

As discussed in Section 4.3.3, we will apply bootstrap to correct for bias. In this simulation study, we take $K = 100$ random walk samples from a scale-free network of size $n = 1000$. The true value of clustering coefficient of the population network is $C = 0.0085$. For each random walk sample, a single random walk of length $B = 300$ is implemented. From each subgraph of size n^* induced by random walk sample, $K = 100$ random walk subsamples are taken with $B = 0.3n^*$ steps. For each sample, the bias and relative bias for $\hat{C}_{HH2}$ will be estimated through Eq.(6.22) and Eq.(6.23) by using the estimates from the subsamples. We will correct the bias for $\hat{C}_{HH2}$ through Eq.(6.24) and Eq.(6.25).

6.5.3.1. Estimating Bias by Bootstrap

In Figure 6.5.5, we plot the histograms for $\hat{bias}(\hat{C}_{HH2})$ and histogram for $r.\hat{bias}(\hat{C}_{HH2})$ by bootstrap, along with the true empirical bias. As one can observe from plot (a), $bias(\hat{C}_{HH2})$ is underestimated by $\hat{bias}(\hat{C}_{HH2})$ while $r.\hat{bias}(\hat{C}_{HH2})$ is almost unbiased estimator for $r.bias(\hat{C}_{HH2})$.

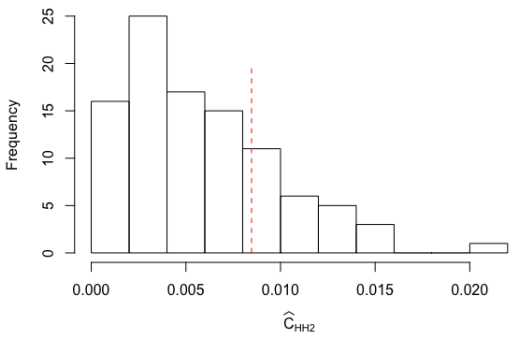


(a) Histogram of estimated bias (b) Histogram of estimated relative bias

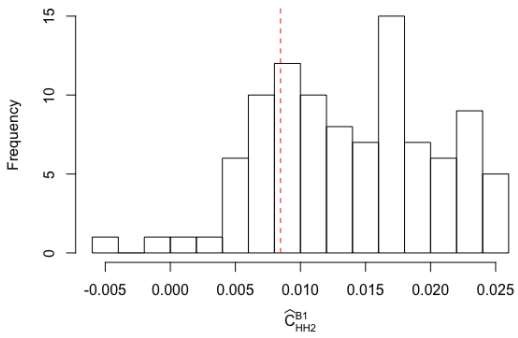
Figure 6.5.5. Histograms of estimated bias and relative bias for $\hat{C}_{HH2}$ by bootstrap.

6.5.3.2. Comparison of $\hat{C}_{HH2}$, $\hat{C}^{B1}_{HH2}$, and $\hat{C}^{B2}_{HH2}$

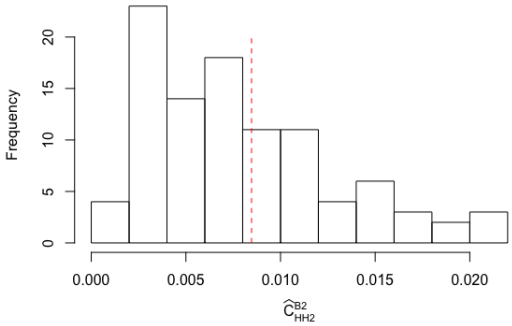
In Figure 6.5.6, we plot histograms of estimates for $\hat{C}_{HH2}$, $\hat{C}^{B1}_{HH2}$ and $\hat{C}^{B2}_{HH2}$. In Table 6.3, we list the numerical comparison of the three estimators. As we can observe, the bias of $\hat{C}_{HH2}$ is not negligible. Using $\hat{C}^{B1}_{HH2}$ is over-correcting the bias so the bias is changed to another direction. Using $\hat{C}^{B2}_{HH2}$ is effective in reducing bias as it reduced the bias to -0.0003. But on the other hand, the standard deviation of $\hat{C}^{B2}_{HH2}$ is larger than that of $\hat{C}_{HH2}$, which is a common consequence of bias correction. People may still prefer using $\hat{C}_{HH2}$ in real cases as 1) it has smaller $NRMSE$ and 2) the computational time is shorter without doing the bootstrapping.



(a) Histogram of $\hat{C}_{HH2}$



(b) Histogram of $\hat{C}^{B1}_{HH2}$



(c) Histogram of $\hat{C}^{B2}_{HH2}$

Figure 6.5.6. Histograms of $\hat{C}_{HH2}$ and adjusted $\hat{C}_{HH2}$ by bootstrap, true global clustering coefficient $C = 0.0085$.

Estimator	Bias	SD	NRMSE
$\hat{C}_{HH2}$	-0.0023	0.0047	0.6243
$\hat{C}^{B1}_{HH2}$	0.0055	0.0069	1.0401
$\hat{C}^{B2}_{HH2}$	-0.0003	0.0057	0.6778

Table 6.3. Numerical comparison of $\hat{C}_{HH2}$, $\hat{C}^{B1}_{HH2}$, and $\hat{C}^{B2}_{HH2}$, true global clustering coefficient $C = 0.0085$.

6.5.4. Comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$

In this section, we compare our estimators $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$ for estimating global clustering coefficient C. We will first compare their computational time and them compare their estimation performance by looking at $NRMSE$.

6.5.4.1. Computational Time

For a random walk sample with $H = 1$ random walk and β sampling fraction, let β^* denote the fraction of unique sampled nodes, i.e., the size of the induced sub graph is $\beta^* n$. For $\hat{C}_{HH1}$ we only need to go over each sampled node and count the connections among its neighbors in the sub graph. Therefore the computational time for $\hat{C}_{HH1}$ is $\beta^* n \binom{\bar{k}^*}{2} \approx \frac{1}{2}\beta^* \bar{k}^* n$, where $\bar{k}^*$ is the average degree in the induced sub graph.

For $\hat{C}_{HH2}$, intuitively we need to go over each sampled triplet to check if it is a triangle or a path of length 2, and the computational time is $\binom{\beta^* n}{3} \approx \frac{1}{6}(\beta^* n)^3$. However among all triplets, the fraction of triangles or the fraction of paths of length 2 is so small, i.e., most triplets have at most one connection, that going through each triplet is not necessary. In

fact, we can quickly get a list of triangles from command *triangles* in R package *igraph*, so we just need to look into those triplets for the λ part. For the τ part, we just need to find the triplets with exactly two connections. According to [2], the $(i,j)^{th}$ element of $A*A$ gives the number of paths of length 2 between node i and node j. If $(i,j)^{th}$ element of $A*A*(1-A)$ is greater than zero, then there's are least one path of length between node i and node j, but i and j are not directly connected. Instead of going over each triplet, we only need to go over each non-zero element in the lower or upper triangle of matrix $A*A*(1-A)$ to find the third element in each path of length two. Therefore the computational time for $\hat{C}_{HH2}$ will be at most $\binom{\beta^* n}{2} \approx \frac{1}{2}(\beta^* n)^2$.

Since the computational time for $\hat{C}_{HH1}$ is in order $\mathcal{O}(n)$ and the computational time for $\hat{C}_{HH2}$ is in order $\mathcal{O}(n^2)$, it can be much more expensive to compute $\hat{C}_{HH2}$ if the network is large. For instance, in our example with $n = 1000$, $B = 300$, and $K = 100$, we have $\beta^* \approx 0.2$ and $\bar{k}^* \approx 3$. The computational time will be 300 for $\hat{C}_{HH1}$ and be at most 2000 for $\hat{C}_{HH2}$. In reality, the network size is usually much larger than $n = 1000$, therefore the computational time for $\hat{C}_{HH2}$ will be even greater for real networks.

6.5.4.2. Estimation Performance

In Table 6.4, we list the numerical comparison for estimation performance of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$. The simulation is based on $K = 100$ random walk samples scale-free network of size $n = 1000$. Each of the random walks takes $B = 300$ steps and the true global clustering coefficient is $C = 0.0085$. As we can observe, $\hat{C}_{HH1}$ is unbiased and has smaller standard deviation than $\hat{C}_{HH2}$, and thus has a smaller $NRMSE$. In order to check if this

is consistent across different lengths of random walks, we also compare the $NRMSE$ for the two estimators with different sampling budgets β in Figure 6.5.7. Since $\hat{C}_{HH1}$ has the smaller $NRMSE$ across different sampling budgets, it is preferable for use in practice.

Estimator	Bias	SD	NRMSE
$\hat{C}_{HH1}$	0.0003	0.0022	0.2620
$\hat{C}_{HH2}$	-0.0023	0.0047	0.6243

Table 6.4. Numerical comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$, true global clustering coefficient $C = 0.0085$.

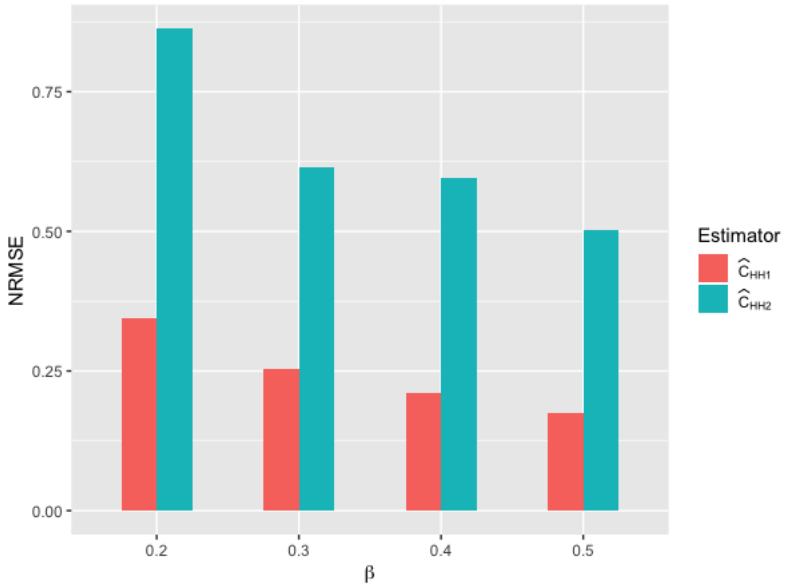
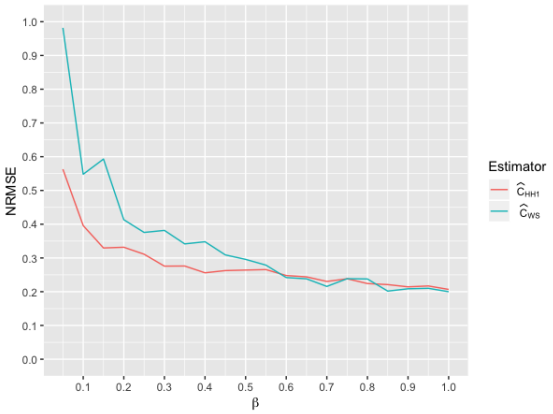


Figure 6.5.7. Comparison of $\hat{C}_{HH1}$ and $\hat{C}_{HH2}$ by bar plot for sampling fraction $\beta = 0.2, 0.3, 0.4, 0.5$, true global clustering coefficient $C = 0.0085$.

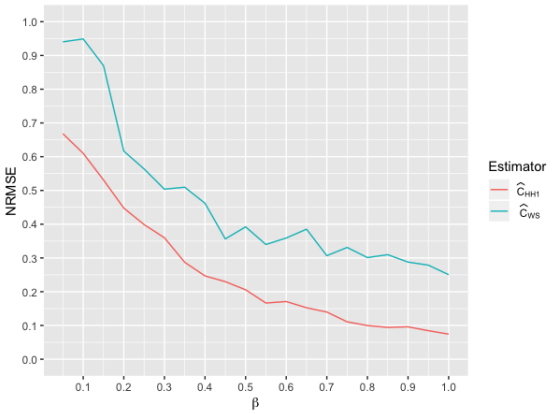
6.5.5. Length of Random Walks

In Figure 6.5.8 we plot the estimation performance measured by normalized root mean square error ($NRMSE$) for two estimators: estimator for the global clustering coefficient based on weighted average LCC ($\hat{C}_{HH1}$) and estimator for the average local clustering coefficient ($\hat{C}_{WS}$). The plots are made for networks with sizes equal to 1000, 5000, and 10000, and $c.v.$ equal to 0.8 and 2.4.

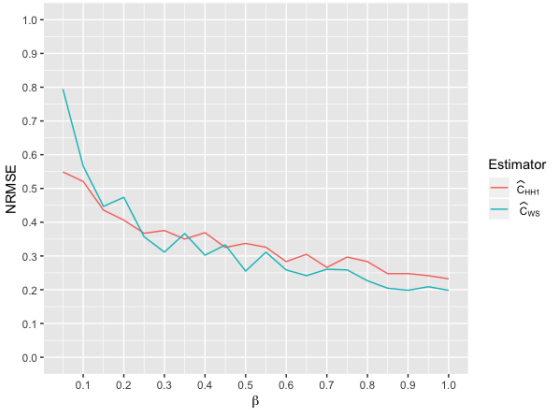
For networks with large $c.v.$ ($c.v. = 2.4$), as shown in plots (a), (c), and (e), the estimation performances of the GCC estimator $\hat{C}_{HH1}$ are similar in networks with difference sizes, while the estimation performances of the ALCC estimator $\hat{C}_{WS}$ improves as the network size increases. For both estimators, the $NRMSE$ decreases as the sampling budget β increases, but when $\beta > 0.3$ the decrease is not as dramatic as it is when $\beta < 0.3$. For networks with small $c.v.$ ($c.v. = 0.8$), as shown in plots (b), (d), and (f), the estimation performances of both $\hat{C}_{HH1}$ and $\hat{C}_{WS}$ are similar in networks with different sizes. As β increases, $NRMSE$ decreases, and the rate of decrease in $NRMSE$ also decreases. While we don't observe a change point in the rate of decrease in networks with small $c.v.$, we recommend to set $\beta = 0.3$ as the minimum sampling budget based on our observation from networks with large $c.v.$.



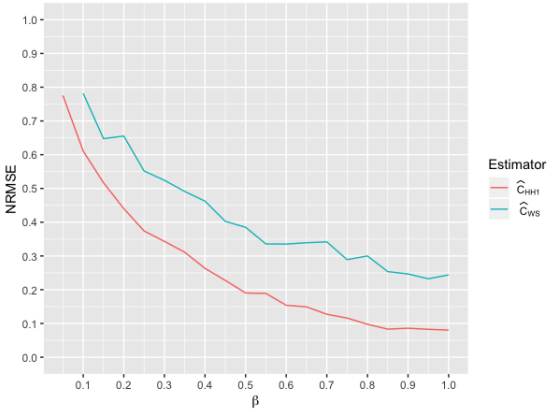
a) $n = 1000$, $c.v. = 2.4$



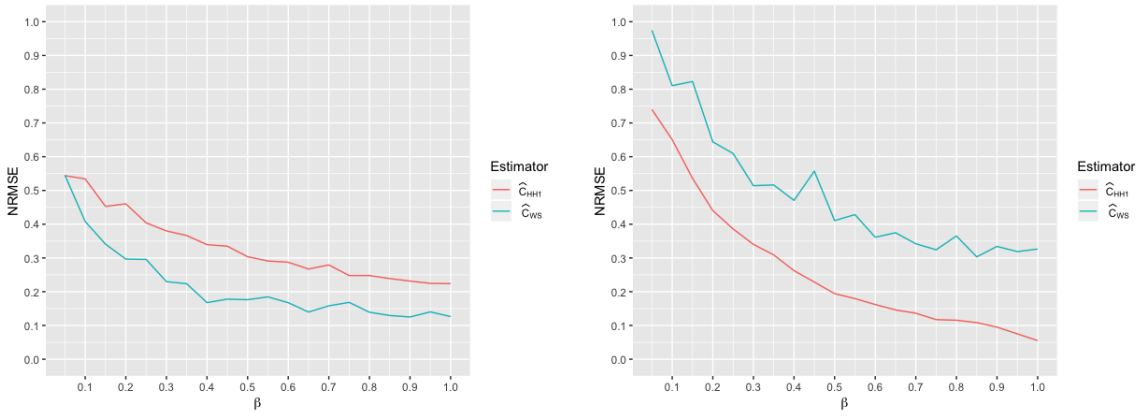
b) $n = 1000$, $c.v. = 0.8$



c) $n = 5000$, $c.v. = 2.4$



d) $n = 5000$, $c.v. = 0.8$



e) $n = 10000$, $c.v. = 2.4$ f) $n = 10000$, $c.v. = 0.8$

Figure 6.5.8. NRMSE of estimator for GCC ($\hat{C}_{HH1}$) and estimator for ALCC ($\hat{C}_{WS}$) versus sampling budget β (low values are better).

6.6. Real Networks

In this section, we apply our proposed estimators for global clustering coefficient C and average local clustering coefficient C_{WS} to some real networks. These data are available on the SNAP (Stanford Network Analysis Project) website. To simplify the analysis, we only consider nodes in the largest connected component. Table 6.5 summarizes the basic information for each network used in this section These networks vary in size, number of edges, average degree, coefficient of variation, and most importantly, C and C_{WS}. For each network, $K = 100$ random walk samples with sampling budget $\beta = 0.3$ are taken, and an estimate is computed from each sample and each estimator. For numerical comparison, we use the mean, bias, relative bias ($r.bias$), standard deviation ($s.d.$), coefficient of variation ($c.v.$), root mean square error ($RMSE$), and normalized root mean square error

($NRMSE$) to measure the estimation performance. For graphical comparison, we use bar plot of $NRMSE$ to measure the estimation performance.

Network	nodes	edges	$< k >$	$c.v.$	C	C_{WS}
Oregon	10.7K	22K	4.1	7.6	0.0093	0.2970
AS-733	6.4K	13.2K	4.3	5.8	0.0096	0.2522
Email-Enron	33.7K	361.7K	21.5	3.5	0.0581	0.5092
CA-HepPh	11.2K	235.2K	42	2.29	0.6594	0.6216
Wiki-Vote	7.1K	103.7K	29.3	2.06	0.1255	0.1396
CA-HepTh	8.6K	49.6K	11.5	1.12	0.2811	0.4816
CA-GrQc	4.2K	26.8K	12.9	1.34	0.6289	0.5566
P2P	10.9K	40K	7.4	0.9	0.0054	0.0062

Table 6.5. Basic information of real networks.

For global clustering coefficient C, we compare the estimation performance of $\hat{C}^{AE}_{HH1}$, $\hat{C}_{HH1}$ and the estimator proposed by Katzir and Hardiman [11], which we denote as $\hat{C}_{HKG}$. The numerical comparison is listed in Table 6.6 and the bar plots are shown in Figure 6.6.1. As we can observe, $\hat{C}^{AE}_{HH1}$ and $\hat{C}_{HKG}$ are unbiased for all networks, and $\hat{C}_{HH1}$ is biased for most networks. On the other hand, $\hat{C}^{AE}_{HH1}$ has smaller $s.d.$ than $\hat{C}_{HKG}$ and thus performs the best among the three estimators, except for network Wiki-Vote. This is due to the fact that $\hat{C}^{AE}_{HH1}$ uses the largest amount of information from the population network among the three estimators. More specifically, to compute $\hat{C}^{AE}_{HH1}$, we observe the actual e_i for each sampled node. For $\hat{C}_{HH1}$, we only observe e^*_i in the induced subgraph and use that to estimate e_i, and for $\hat{C}_{HKG}$ we only observe if there's a connection between

the node before the focal node and the node after the focal node, for each node visited by the random walk. The richer the information is, the more precise the estimator is. In addition, the $NRMSE$s for all networks are very small (under or close to 0.2). Therefore in practice, if we are able to observe all neighbors of each sampled node, $\hat{C}^{AE}_{HH1}$ is preferable.

For some networks, such as Email-Enron, CA-HepPh, CA-HepTh, and CA-GrQc, the estimation performance of $\hat{C}_{HH1}$ is close to that of $\hat{C}^{AE}_{HH1}$. This is because $\hat{C}_{HH1}$ is almost unbiased for for those networks, as we can observe from Table 6.6.1. Therefore bias is the main source of estimation error for $\hat{C}_{HH1}$. For network Wiki-Vote, $\hat{C}_{HH1}$ performs best among the three estimators.

Network	Estimator	*bias*	*r.bias*	*s.d.*	*c.v.*	$RMSE$	$NRMSE$
Oregon	$\hat{C}^{AE}_{HH1}$	0.0000	-0.0001	0.0007	0.0730	0.0007	0.0726
	$\hat{C}_{HH1}$	0.0055	0.5959	0.0011	0.1228	0.0057	0.6083
	$\hat{C}_{HKG}$	0.0000	-0.0030	0.0026	0.2760	0.0026	0.2746
AS-733	$\hat{C}^{AE}_{HH1}$	0.0001	0.0087	0.0008	0.0812	0.0008	0.0813
	$\hat{C}_{HH1}$	0.0048	0.4964	0.0013	0.1312	0.0049	0.5133
	$\hat{C}_{HKG}$	0.0005	0.0570	0.0034	0.3523	0.0034	0.3551
Email-Enron	$\hat{C}^{AE}_{HH1}$	-0.0001	-0.0015	0.0027	0.0327	0.0027	0.0325
	$\hat{C}_{HH1}$	0.0094	0.1100	0.0031	0.0366	0.0099	0.1159
	$\hat{C}_{HKG}$	0.0002	0.0027	0.0046	0.0537	0.0046	0.0535
CA-HepPh	$\hat{C}^{AE}_{HH1}$	0.0003	0.0004	0.0102	0.0155	0.0102	0.0155
	$\hat{C}_{HH1}$	-0.0282	-0.0427	0.0163	0.0248	0.0325	0.0493
	$\hat{C}_{HKG}$	-0.0005	-0.0008	0.0190	0.0288	0.0189	0.0286
Wiki-Vote	$\hat{C}^{AE}_{HH1}$	-0.0206	-0.1640	0.0016	0.0127	0.0206	0.1645
	$\hat{C}_{HH1}$	-0.0124	-0.0987	0.0025	0.0195	0.0126	0.10067
	$\hat{C}_{HKG}$	0.0122	0.0974	0.0104	0.0829	0.0160	0.1279
CA-HepTh	$\hat{C}^{AE}_{HH1}$	-0.0116	-0.0414	0.0629	0.2239	0.0637	0.2266
	$\hat{C}_{HH1}$	-0.0620	-0.2206	0.0667	0.2373	0.0908	0.3232
	$\hat{C}_{HKG}$	-0.0108	-0.0384	0.0635	0.2259	0.0641	0.2281
CA-GrQc	$\hat{C}^{AE}_{HH1}$	-0.0541	-0.0860	0.0797	0.1267	0.0959	0.1526
	$\hat{C}_{HH1}$	-0.1006	-0.1600	0.0981	0.1560	0.1402	0.2229
	$\hat{C}_{HKG}$	-0.0555	-0.0883	0.0876	0.1392	0.1033	0.1643

P2P	$\hat{C}^{AE}_{HH1}$	0.0000	-0.0043	0.0002	0.0451	0.0002	0.0451
	$\hat{C}_{HH1}$	-0.0021	-0.3893	0.0004	0.0691	0.0021	0.3953
	$\hat{C}_{HKG}$ 0.0000	0.0029	0.0015	0.2816	0.0015	0.2802	

Table 6.6. Numerical comparison of $\hat{C}^{AE}_{HH1}$, $\hat{C}_{HH1}$, and $\hat{C}_{HKG}$ for GCC.

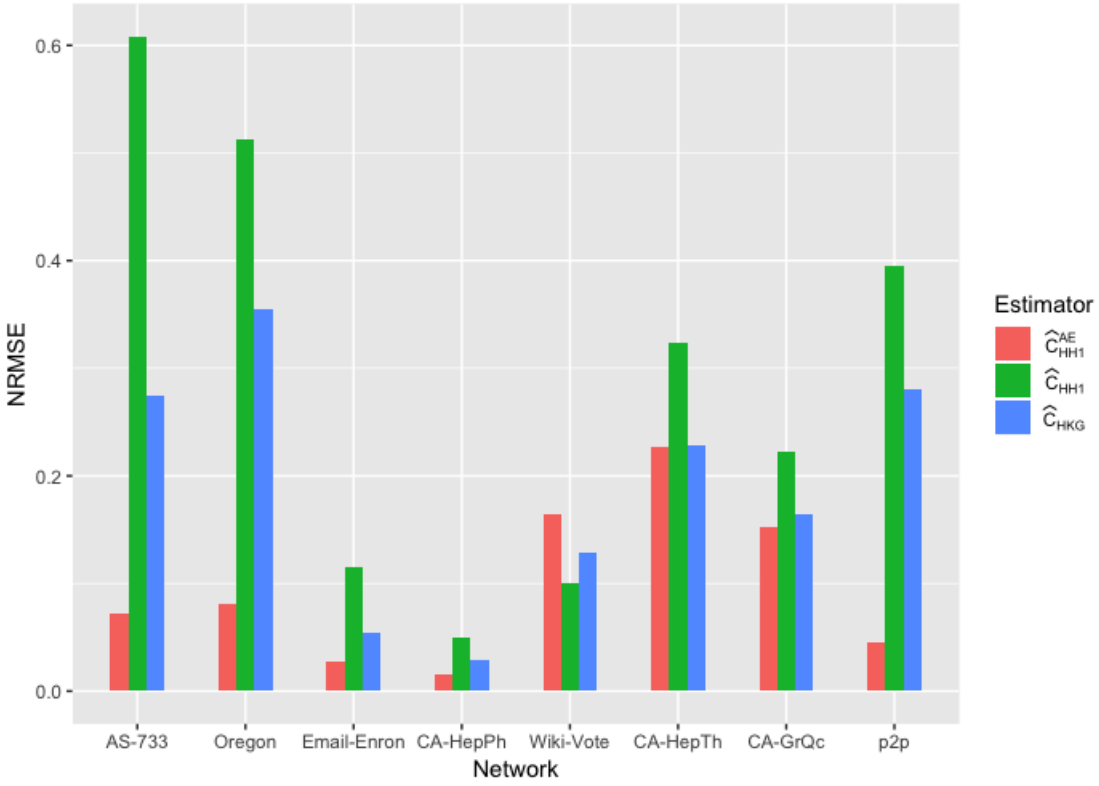


Figure 6.6.1. Comparison of $\hat{C}^{AE}_{HH1}$, $\hat{C}_{HH1}$, and $\hat{C}_{HKG}$ by bar plot for global clustering coefficient C.

For local clustering coefficient C_{WS}, we compare the estimation performance of $\hat{C}^{AE}_{WS}$, $\hat{C}_{WS}$ and the estimator proposed by Katzir and Hardiman [11], which we denote as $\hat{C}_{HKL}$. The numerical comparison is listed in Table 6.7 and the bar plots are shown in Figure

6.6.2. As we can observe, $\hat{C}_{WS}^{AE}$ and $\hat{C}_{HKL}$ are unbiased for all networks, and $\hat{C}_{WS}$ is biased for most networks. On the other hand, $\hat{C}_{WS}^{AE}$ has smaller $s.d.$ than $\hat{C}_{HKL}$ and thus performs the best among the three estimators with the smallest $NRMSE$. Again this is due to the fact that $\hat{C}_{WS}^{AE}$ uses the largest amount of information from the population network among the three estimators. Except for network P2P, the $NRMSE$s are all very small (under 0.2). Therefore in practice, $\hat{C}_{WS}^{AE}$ is most preferable if we are able to observe all neighbors of each sampled node. For some networks, such as AS-733 and Oregon, where $\hat{C}_{WS}$ is almost unbiased, the estimation performance of $\hat{C}_{WS}$ is close to that of $\hat{C}_{WS}^{AE}$. For network Wiki-Vote, $\hat{C}_{WS}$ performs best among the three estimators.

Network	Estimator	*bias*	*r.bias*	*s.d.*	*c.v.*	*RMSE*	*NRMSE*
Oregon	$\hat{C}_{WS}^{AE}$	0.0009	0.0029	0.0217	0.0730	0.0216	0.0727
	$\hat{C}_{WS}$	-0.0057	-0.0191	0.0214	0.0721	0.0221	0.0743
	$\hat{C}_{HKL}$	-0.0001	-0.0002	0.0223	0.0752	0.0222	0.0748
AS-733	$\hat{C}_{WS}^{AE}$	0.0006	0.0025	0.0265	0.1049	0.0263	0.1044
	$\hat{C}_{WS}$	-0.0065	-0.0260	0.0259	0.1027	0.0266	0.1054
	$\hat{C}_{HKL}$	0.0001	0.0002	0.0314	0.1243	0.0312	0.1237
Email-Enron	$\hat{C}_{WS}^{AE}$	0.0016	0.0032	0.0294	0.0577	0.0293	0.0575
	$\hat{C}_{WS}$	-0.0899	-0.1766	0.0264	0.0518	0.0937	0.1839
	$\hat{C}_{HKL}$	0.0011	0.0021	0.0306	0.0601	0.0305	0.0598
CA-HepPh	$\hat{C}_{WS}^{AE}$	0.0049	0.0079	0.0321	0.0517	0.0323	0.0520
	$\hat{C}_{WS}$	-0.1570	-0.2527	0.0381	0.0613	0.1616	0.2599
	$\hat{C}_{HKL}$	0.0026	0.0042	0.0432	0.0695	0.0430	0.0692
Wiki-Vote	$\hat{C}_{WS}^{AE}$	0.0173	0.1242	0.0205	0.1469	0.0268	0.1922
	$\hat{C}_{WS}$	0.0063	0.0453	0.0197	0.1413	0.0207	0.1482
	$\hat{C}_{HKL}$	0.0234	0.1677	0.0325	0.2327	0.0400	0.2867
CA-HepTh	$\hat{C}_{WS}^{AE}$	-0.0013	-0.0027	0.0238	0.0493	0.0237	0.0492
	$\hat{C}_{WS}$	-0.0830	-0.1724	0.0252	0.0523	0.0867	0.1801
	$\hat{C}_{HKL}$	-0.0037	-0.0077	0.0295	0.0613	0.0296	0.0615
CA-GrQc	$\hat{C}_{WS}^{AE}$	0.0006	0.0011	0.0409	0.0736	0.0407	0.0732
	$\hat{C}_{WS}$	-0.0777	-0.1395	0.0418	0.0750	0.0881	0.1583
	$\hat{C}_{HKL}$	-0.0002	-0.0003	0.0475	0.0853	0.0472	0.0848

P2P	$\hat{C}_{WS}^{AE}$	0.0001	0.0211	0.0019	0.3096	0.0019	0.3088
	$\hat{C}_{WS}$	-0.0020	-0.3285	0.0018	0.2922	0.0027	0.4387
	$\hat{C}_{HKL}$	0.0003	0.0422	0.0029	0.4672	0.0029	0.4667

Table 6.7. Numerical comparison of $\hat{C}_{WS}^{AE}$, $\hat{C}_{WS}$, and $\hat{C}_{HKL}$ for ALCC.

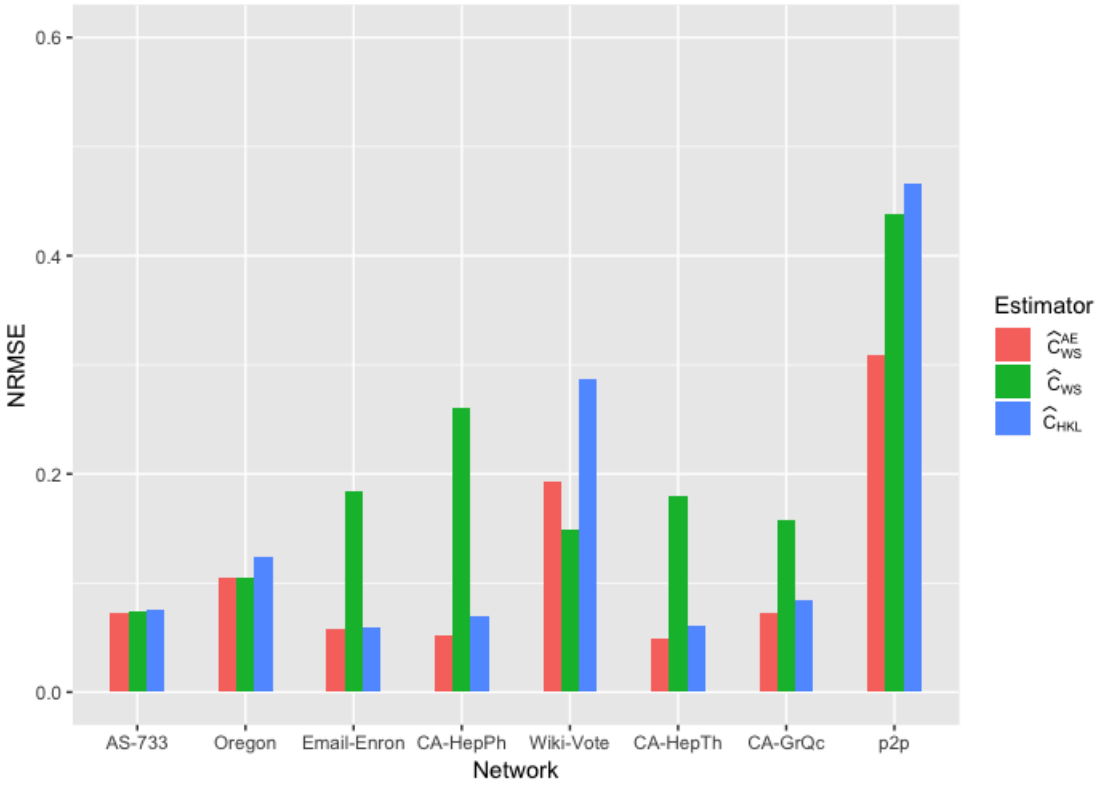


Figure 6.6.2. Comparison of $\hat{C}_{WS}^{AE}$, $\hat{C}_{WS}$, and $\hat{C}_{HKL}$ by bar plot for average local clustering coefficient C_{WS}.

6.7. Summary of Results

By applying the estimators and evaluation techniques to several simulated networks and real networks, we developed the following findings:

- For the GCC, if we can observe the actual nmber of connections among neighbors of each sampled node, the estimator based on weighted average of LCCs is unbiased with small variance.
- For the GCC, bootstrapping the estimator based on triangles can reduce bias but at the cost of increasing the standard error, and can increase the NRMSE. It also has a longer computing time than the estimator based on weighted average of LCCs. Therefore estimation based on weighted average of LCCs is preferable to estimation based on triangles.
- For the ALCC, if we can observe the actual number of connections among neighbors of each sampled node, the estimator is unbiased with small variance.
- For both the GCC and the ALCC, if we can not observe the actual number of connections among neighbors of each sampled node, and need to estimate it using the number observed from the induced subgraph, the bias and NRMSE of the estimator tend to increase appreciably. Therefore in practice, it is recommended to observe the actual number of connections among neighbors of each sampled node to minimize the estimation error, even if it will increase sampling cost.

6.8. Discussion and Future Work

In this chapter we applied random walk sampling and generalized the usage of Hansen-Hurwitz estimator to estimate the global clustering coefficient (GCC) and the average local clustering coefficient (ALCC) of a network. For the GCC, we developed two estimators based on weighted average LCC and on triangles respectively. The former is unbiased with small variance if we can observe all neighbors of sampled nodes. The latter is biased

due to lack of knowledge of the exact inclusion probability of a triplet when the random walk has moderate length. We can use bootstrap to correct for this bias but at the cost of increasing the standard error and $NRMSE$. For the ALCC, we can get an unbiased estimator with small variance if we can observe all neighbors of sampled nodes. We applied the estimator for the GCC based on weighted average LCC and the estimator for ALCC to some real networks, and found that our estimators outperform the estimators proposed by Katzir and Hardiman [11] if we can observe all neighbors of sampled nodes.

There are two directions for future work. 1) In this work, when we are not able to observe all neighbors of a sampled node, we used the ratio of its observed degree in the induced subgraph and its actual degree to estimate the ratio of its observed number connections among neighbors in the induced subgraph and its actual number of connections among neighbors. This estimation brings bias in the estimator for the GCC based weighted average LCC and the estimator for the ALCC. We can try to reduce the bias by finding an alternative estimation for the ratio of connections among neighbors. 2) The estimator for the GCC based on triangles has two issues. First the computational times is too long, and second it is biased. To deal with the bias issue, we used bootstrap but it increased the standard deviation so that the $NRMSE$ of the estimator adjusted by bootstrap is greater than the $NRMSE$ of the original estimator. In order to reduce both bias and standard deviation, some adjustment is needed for the theoretical inclusion probabilities of triplets so that they are closer to the empirical ones.

Bibliography

[1] E. Katzav, M. Nitzan, D. ben Avraham, P. Krapivsky, R. Kühn, N. Ross, and O. Biham, "Analytical results for the distribution of shortest path lengths in random networks," *EPL (Europhysics Letters)*, vol. 111, no. 2, p. 26 006, 2015.

[2] M. Newman, *Networks: an introduction*. Oxford university press, 2010.

[3] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *Proceedings of the National Academy of Sciences*, vol. 99, no. 25, pp. 15 879–15 882, 2002.

[4] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Physical review letters*, vol. 90, no. 5, p. 058 701, 2003.

[5] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis, "Fast shortest path distance estimation in large networks," in *Proceedings of the 18th ACM conference on Information and knowledge management*, ACM, 2009, pp. 867–876.

[6] B. Ribeiro, P. Basu, and D. Towsley, "Multiple random walks to uncover short paths in power law networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, IEEE, 2012, pp. 250–255.

[7] M. H. Hansen and W. N. Hurwitz, "On the theory of sampling from finite populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 333–362, 1943.

[8] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[9] K. Sigman, "Limiting distribution for a markov chain," 2009. [Online]. Available: `http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-MCII.pdf`.

[10] T. W. Anderson, "Second-order moments of a stationary markov chain and some applications," pp. 1–16, 1989.

[11] L. Katzir and S. J. Hardiman, "Estimating clustering coefficients and size of social networks via random walk," *ACM Transactions on the Web (TWEB)*, vol. 9, no. 4, p. 19, 2015.

[12] M. Nitzan, E. Katzav, R. Kühn, and O. Biham, "Distance distribution in configuration-model networks," *Physical Review E*, vol. 93, no. 6, p. 062 309, 2016.

[13] K. Okamoto, W. Chen, and X.-Y. Li, "Ranking of closeness centrality for large-scale social networks," in *International Workshop on Frontiers in Algorithmics*, Springer, 2008, pp. 186–195.

[14] V. Ufimtsev and S. Bhowmick, "An extremely fast algorithm for identifying high closeness centrality vertices in large-scale networks," in *Proceedings of the 4th Workshop on Irregular Applications: Architectures and Algorithms*, IEEE Press, 2014, pp. 53–56.

[15] A. Saxena, R. Gera, and S. Iyengar, "A faster method to estimate closeness centrality ranking," *arXiv preprint arXiv:1706.02083*, 2017.

[16] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, 1st. Springer Publishing Company, Incorporated, 2009, ISBN: 038788145X, 9780387881454.

[17] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[18] A.-L. Barabasi, "Network science the scale-free property," 2014. [Online]. Available: `http://barabasi.com/f/623.pdf`.

[19] C. Bauckhage, K. Kersting, and B. Rastegarpanah, "The weibull as a model of shortest path distributions in random networks," in *Proc. Int. Workshop on Mining and Learning with Graphs, Chicago, IL, USA*, 2013.

[20] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical review letters*, vol. 86, no. 14, p. 3200, 2001.

[21] M. Kas, M. Wachs, K. M. Carley, and L. R. Carley, "Incremental algorithm for updating betweenness centrality in dynamically growing networks," in *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*, ACM, 2013, pp. 33–40.

[22] C.-C. Yen, M.-Y. Yeh, and M.-S. Chen, "An efficient approach to updating closeness centrality and average path length in dynamic networks," in *2013 IEEE 13th International Conference on Data Mining*, IEEE, 2013, pp. 867–876.

[23] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Computing classic closeness centrality, at scale," in *Proceedings of the second ACM conference on Online social networks*, ACM, 2014, pp. 37–50.

[24] D. E. J. Wang, "Fast approximation of centrality," *Graph Algorithms and Applications*, vol. 5, no. 5, p. 39, 2006.

[25] K. Wehmuth and A. Ziviani, "Daccer: Distributed assessment of the closeness centrality ranking in complex networks," *Computer Networks*, vol. 57, no. 13, pp. 2536–2548, 2013.

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[27] M. E. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 1, pp. 2566–2572, 2002.

[28] A. M. Verdery, J. C. Fisher, N. Siripong, K. Abdesselam, and S. Bauldry, "New survey questions and estimators for network clustering with respondent-driven sampling data," *Sociological methodology*, vol. 47, no. 1, pp. 274–306, 2017.

[29] T. Schank and D. Wagner, *Approximating clustering-coefficient and transitivity*. Universität Karlsruhe, Fakultät für Informatik, 2004.

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

[31] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *2010 Proceedings IEEE Infocom*, Ieee, 2010, pp. 1–9.

[32] E. Cem and K. Sarac, "Estimating clustering coefficients via metropolis-hastings random walk and wedge sampling on large osn graphs," in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, IEEE, 2016, pp. 1–2.

[33] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2.

[34] D Basu, "Role of the sufficiency and likelihood principles in sample survey theory," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 441–454, 1969.

[35] M. Zheng and B. D. Spencer, "Estimating shortest path length distributions via random walk sampling," *arXiv preprint arXiv:1806.01757*, 2018.