SAMPLING OF RANDOM GRAPHS WITH PRESCRIBED DEGREE SEQUENCE

A Dissertation Presented to

the Faculty of the Department of Physics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Weibin Zhang

May 2019

SAMPLING OF RANDOM GRAPHS WITH PRESCRIBED DEGREE SEQUENCE

Weibin Zhang

APPROVED:

Dr. Kevin E. Bassler, Chairman
Department of Physics

Dr. Gemunu Gunaratne
Department of Physics

Dr. Arthur Weglein
Department of Physics

Dr. Krešimir Josić
Department of Mathematics

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I would like to first sincerely thank my advisor, Dr. Kevin E. Bassler, for his guidance and support throughout my Ph.D. study. It's my fortune to have met and become a student of such a kind, responsible and intellectual professor. He not only teaches me knowledge, skills, and way of thinking, but also shows me the beauty of math, science, and life. I also gratefully acknowledge Dr. Royce K.P. Zia, Dr. Gemunu Gunaratne, Dr. George Reiter, Dr. Arthur Weglein, Dr. Krešimir Josić and Dr. Zoltán Toroczkai for their useful advice to my research. Besides, I appreciate the help and discussions from past and current members in my research group Dr. Charo del Genio, Dr. Florian Greil, Dr. Amy Nyberg, Dr. Shabnam Hossein, Dr. Tianlong Chen, Dr. Pramesh Singh, Mohammadmehdi Ezzatabadipour, Jiahao Guo, Vidushi Adlakha, Negin Alizadeh, Zhenyu Dai, and Erich McMillan. I also want to thank Dr. Lijian Chen, Dr. Jianfa Chen, Dr. Lei Sun, Dr. Xiang Zhang, Dr. Parth Singh, Dr. Fabio Zegarra, Hanming Yuan, Ziping Ye, Le Fang and many other good friends I met here for their support and accompany in both life and study. Finally, I would like to especially thank my parents for their love, care, understanding, and unconditional support to help me pursue my dream.

SAMPLING OF RANDOM GRAPHS WITH PRESCRIBED DEGREE SEQUENCE

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Physics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Weibin Zhang

May 2019

Abstract

Random graph generation is the foundation of the statistical study of complex networks, which are commonly found in social, technological, and biological systems. In empirical studies, often only limited information about the topological structure of a network is available. In order to make the best use of this information, one must sample the ensemble of graphs that satisfy the constraint of the known structure, but are otherwise as random as possible.

Similar to the microcanonical ensemble and canonical ensemble in statistical physics, there are two types of methods to generate an ensemble of graphs with prescribed topological constraints. A hard constraint method generates an ensemble of graphs in which each graph satisfies the constraints exactly. On the other hand, a soft constraint method generates an ensemble of graphs in which the constraints are satisfied only on average within the ensemble.

In this dissertation, inspired by the idea of maximizing entropy, improvements in a hard constraint method called Sequential Importance Sampling (SIS) are developed for the case when the number of connections of each node, the degree sequence, is prescribed. Among the improvements is a more stable method of calculating ensemble averages that allows much larger networks to be sampled. With the improved methods, results from hard constraint methods are compared with those of soft constraint methods. It is found that soft constraint methods significantly overestimate the global clustering coefficient for both regular random graphs and scale-free graphs. This implies that problems exist with many network analyses and that care must be taken about the assumptions of network statistical analyses.

A dynamical model to generate random graphs with prescribed degrees is also considered. The graphs resulting from this model are split graphs. We develop a linear complexity algorithm to decompose any graph into a series of split graphs,

which can potentially be used to improve the efficiency of hard constraint sampling methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A network, or graph, can be a useful way to represent relations between different entities [21]. However, due to the difficulty of conducting an exhaustive survey or privacy concerns [97], the complete structure of a graph is not always available. Often only partial information, like the number of neighbors for each node, i.e. degree, is available. In order to make the best use of the information we have, we want to generate an ensemble of graphs using the known information as structural constraints. But enumeration [89] of all possible graphs satisfying these constraints is too costly [107] for any reasonable size of the graph: given N nodes, there are at most $\sim O(2^{N(N-1)/2})$ possible configurations for undirected, unweighted simple graph. Thus, we have to sample graphs from this ensemble of possible graphs, which in general is a difficult problem [125]. While different properties, like degree correlation or spectrum, can be used as constraints of the ensemble, in this dissertation, we focus on degree

sequence, which is a simple local measurement of a graph. Even for such a seemingly simple constraint, it turns out to be a non-trivial problem to sample graphs that satisfy this constraint effectively.

There are two types of methods to generate an ensemble of graphs satisfying some constraints. Hard constraint methods sample from an ensemble of graphs in which every graph in this ensemble satisfies the constraint exactly, which reminds us of the micro-canonical ensemble in statistical physics [120]. Among this category there are Markov chain Monte Carlo method (MCMC) [155] and direct construction methods including the configuration model (CM) [16] and Sequential Importance Sampling (SIS) [46]. On the other hand, soft constraint methods define an ensemble of graphs in which the ensemble average of the graph property agrees with the constraint, which is very similar to the canonical ensemble [120]. Here there is the Exponential Random Graph Model (ERGM) [145], which is inspired by the principle of maximizing entropy [80, 81], and the Chung-Lu model [37], which can be seen as a simplified version of the ERGM when the maximum degree is small enough. Another way is to let the graph evolve to satisfy the constraints following some dynamics [12]. While the statistical property of the evolved ensemble is known only for limited cases [98], this ensemble has some interesting properties [14].

While many methods exist, they all have room for improvement. In this dissertation we will discuss ways to improve the efficiency of SIS [46], the statistical difference between soft and hard constraint methods, the statistical properties of a dynamic model called preferred degree extreme dynamics [168], and a potential way to improve the efficiency of MCMC using canonical decomposition of graph split graphs [55].

In the rest of this chapter, we'll give a brief introduction of complex networks, and different ways to generate random graphs. More detailed description of the algorithms can be found in the corresponding chapters.

1.2 Networks

Graph or complex network is a useful representation of many kinds of systems we want to study [114, 113, 4, 50, 47, 21]. A graph contains a set of nodes and a set of edges connecting the nodes. A node can represent an entity in a system and an edge can represent the relationship or interaction between entities. In general, both nodes and edges can have their own properties [15, 167, 166, 161]. For any system, if we can abstract the system into a graph, we can use graph theory and knowledge of complex network to describe [146, 157, 119], predict [103, 102] and even control [100] the system.

1.2.1 Real world examples

Many systems can be described as networks [42, 96]. Based on what the network represents, we can roughly classify networks into the following categories: In social networks [26, 30], nodes represent people, and edges can represent whether two persons are friends [110], call each other [148], exchange email [93], or interact on social media [127]. Nodes can also represent organizations. In company ownership network [134, 156, 122] node is company and edge means how much each company owns another company. In the network of financial institutions like banks [27], the edges can represent how much money is transferred between banks. We can also think of

technological networks like transportation networks (road [83], railway [138], airline [68], and sea [84]). Similarly, considering the transportation of energy and information, there are power grid [2, 126], the telephone network [162], and the Internet [53]. When it comes to information networks, there are semantic networks [108], citation networks [70], and web page graphs, in which the most famous one is WWW (World Wide Web) [5]. If we consider networks with different types of nodes [67], we have collaboration network [115] and recommendation network [130]. When considering the biological system, we can find networks in all scales, like gene regulatory network [38], protein interaction network [28] and metabolic network [9] in a microscopic level, neural networks [144, 143] in mesoscopic level, and food web [11] in macroscopic level.

1.2.2 Definition

A graph [114, 24, 37], or network $G(V, E)$ contains a set V of nodes (also called vertices) and a set E of edges (also called links) connecting pairs of nodes. In general, edges can have direction, sign, and weight. If multiple edges exist for the same pair of nodes, they are called multi-edges. If the two ends of an edge point to the same node, this is called a self-loop. A graph is called a simple graph if there are no self-loops and no multi-edges.

In this dissertation, we will focus on undirected, unweighted, simple networks.

1.2.3 Representation

Given a graph $G(V, E)$, we can label the nodes $V = \{1, 2, \cdots, N\}$, where $N = |V|$ is the number of nodes. Then the edge set E can be described as a list of edges

$\{(i,j) \in E\}$. This is called edgelist. Equivalently we can define adjacency matrix A, where $A_{ij} \neq 0$ if i and j are connected, i.e. $(i,j) \in E$, and $A_{ij} = 0$ otherwise. This is a better representation for linear algebra. For undirected graph, $A_{ij} = A_{ji}$. For unweighted graph without multi-edges, $A_{ij} \in \{0,1\}$. For graph with no self-loops, $A_{ii} = 0$.

1.2.4 Notation

1.2.4.1 Degree, degree distribution and degree correlation

The degree d_i of a node i is the number of neighbors it connects to. For an undirected simple graph, using the expression of the adjacency matrix, we can write degree as

$$d_i = \sum_j A_{ij} \ . \tag{1.1}$$

Degree distribution $P(d)$ is the probability of a node having degree d. Degree correlation between different degrees d_i and d_j can be characterized by the joint probability $P(d_i, d_j)$, i.e. the probability that a node of degree d_i and a node of degree d_j are connected.

1.2.4.2 Path and cycle

A path is a sequence of nodes in which every consecutive pair of nodes is connected by an edge. The length of a path is the number of edges traversed along the path. The number of paths of length r from node i to node j is $[A^r]_{ij}$. A path with its end node the same as its starting node is a cycle. The total number of cycles of length r

is

$$\sum_i [A^r]_{ii} = Tr(A^r) \ . \tag{1.2}$$

1.2.4.3 Clustering coefficient

Global clustering coefficient, also known as transitivity, describes the probability that node i connects to j given the condition that i connects to k and k connects to j,

$$CC_g = \frac{3N_\Delta}{N_V} \ , \tag{1.3}$$

where CC_g is the global clustering coefficient, N_Δ is number of triangles, and N_V is number of connected triples.

A similar definition is local clustering coefficient, which describes the probability that the neighbors of a certain node are connected.

$$CC_l(i) = \frac{\sum_{j,k} A_{ij}A_{ik}A_{jk}}{d_i(d_i - 1)} \ . \tag{1.4}$$

In this dissertation, we only consider global clustering coefficient.

1.2.4.4 Others metrics

There are many other metrics [43] to describe a graph, like node and edge centrality [101], motif [6], community structure [119, 104, 60, 61, 34], connected component, cut set and graph spectrum [123]. While each of them captures the important properties of a graph, they are not the focus of this dissertation.

1.3 Random graph generation

1.3.1 History and models

1.3.1.1 The Erdos-Renyi random graph

A straight forward way to generate random graphs is to connect pairs of nodes randomly. Erdos and Renyi (ER) [57, 23] developed and carefully analyzed a model $G(N, M)$ in which M pairs of nodes are chosen uniformly to be connected from all the $N(N-1)/2$ possible pairs of nodes. In other words, for all the $\binom{\binom{N}{2}}{M}$ possible graphs having N nodes and M edges, ER model picks up one of them with equal probability. A similar model is the $G(N, p)$ model, which connects any pair of nodes with independent and identical probability $0 < p < 1$. These two versions of ER model are asymptotically the same if N goes to infinity and Np is fixed. Nowadays both $G(N, M)$ and $G(N, p)$ are called ER model. In this dissertation, we focus on $G(N, p)$ because it is easier to analyze.

Since the elements of the adjacency matrix are independent and identically distributed (i.i.d.), the degree distribution of $G(N, p)$ can be written as [54, 23]

$$P(d) = \binom{N-1}{d} p^d (1-p)^{N-1-d} . \tag{1.5}$$

In the limit of large N and fixed $\bar{d} = (N-1)p$, the degree distribution becomes Poisson distribution

$$P(d) = \frac{\bar{d}^d}{d!} e^{-\bar{d}} . \tag{1.6}$$

Thus, the ER graphs are sometimes also called Poisson random graphs.

Since each node connects to $\bar{d}$ neighbors randomly, a node can reach around $\bar{d}^l$ after l hops. Thus, in order to reach any other node in a network of size N, only

$L \approx \ln N / \ln \bar{d} \propto \ln N$ is needed [23, 158].

Also because of the i.i.d. connection probability, the global clustering coefficient for $G(N, p)$ is just p.

An interesting fact of the ER model is the emergence of the giant component [54, 23], as shown in Figure 1.1. For small $p < 1/N$, almost surely (with probability tending to 1 as $N \to \infty$) the graph doesn't contain any component of size bigger than $O(\ln N)$. However, when $p = 1/N$, a component of size $O(N^{2/3})$ emerges. For $p > 1/N$, there exists a unique giant component having size $O(N)$ and no other component has a size bigger than $O(\ln N)$. As p increases further to $p \geq \ln N/N$, the graph becomes totally connected.

1.3.1.2 Small-world phenomena and the Watts-Strogatz model

Many real-world networks have the so-called "small world" property, which means the network is sparse, the typical distance between any pair of nodes is small, and the network has a relatively high clustering coefficient. To be more specific, it requires the average distance $L \propto \ln N$ and the clustering coefficient remains finite as $N \to \infty$.

While ER graph has a small average shortest distance $L \propto \ln N$, its clustering coefficient $CC_g = p = \bar{d}/(N-1) \to 0$ for fixed $\bar{d}$ and infinite N. In order to deal with this problem, Watts and Strogatz (WS) [158] proposed the small-world model. In WS model, the random graph is constructed by first placing N nodes uniformly on a low dimensional lattice and connecting each node with all of its neighbors within certain distance k, and then rewiring [158] or adding [121] the edges randomly with certain probability p. For $p = 0$, the graph is just a lattice with high average shortest distance and high clustering coefficient. For $p = 1$, the graph is almost an ER random

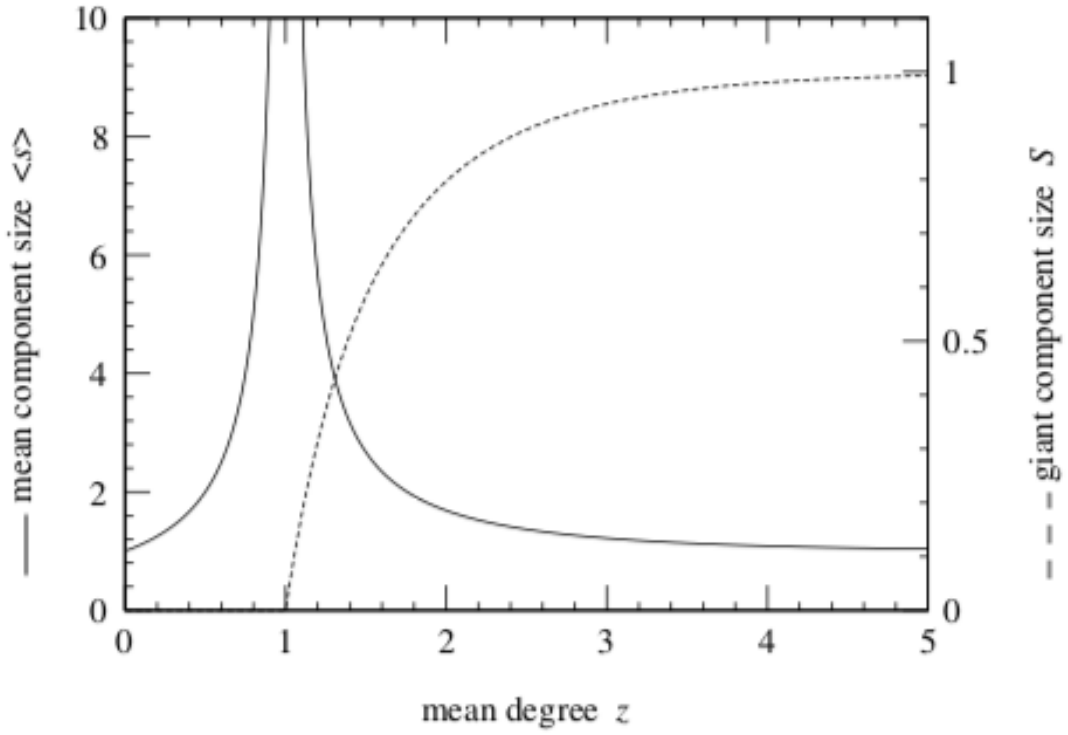


Figure 1.1: Average component size $< s >$ excluding giant component (solid line) and size of giant component S (dashed line) as a function of average degree z in ER graph [113, 54]. Reprinted from "The structure and function of complex networks"[113], by M. E. Newman, 2003, *SIAM review, 45(2)*, p. 199. Copyright 2003 by Society for Industrial and Applied Mathematics.

graph with low average shortest distance and low clustering coefficient. However, as p goes from 0 to 1, the average shortest distance drops quickly while the clustering coefficient remains high [158, 121, 10], as shown in Figure 1.2.

1.3.1.3 Scale-free network and the Barabasi-Albert model

Many real-world networks are observed to show the scale-free property [8, 129, 128, 140], i.e. the degree distribution follows power-law $P(d) \propto d^{-\gamma}$, where typically $2 < \gamma < 3$. Since neither ER model nor WS model has a power-law degree distribution, a lot of new mechanisms, such as the fitness model and the gradient network, are proposed to explain the scale-free property. The most famous one among those is the growth and preferential attachment model, in particular, the Barabasi-Albert (BA) model [8], which mimics the dynamics through which the real world networks are formed.

BA model starts with m_0 isolated nodes. At each time step, we add a new node and randomly connect it with $m < m_0$ old nodes. We keep adding new nodes until there are N nodes in total. The probability of connecting the new node with any old node is linearly proportional to the current degree of the old node.

BA model generates graphs with power-law degree distribution where the exponent $\gamma = 3$ [8, 48, 94]. Its average shortest path scales as $L \sim \ln N / \ln \ln N$ [25] and its clustering coefficient scales as $CC_g \sim N^{-0.75}$. Many variants [51, 94, 49, 65, 3, 48, 131, 75, 92] have been proposed to generate more realistic graphs with a wider range of γ and larger clustering coefficient.

Another family of models to generate scale-free graphs, inspired by protein interactions, is the node copying model [142, 153], which increase the network size by

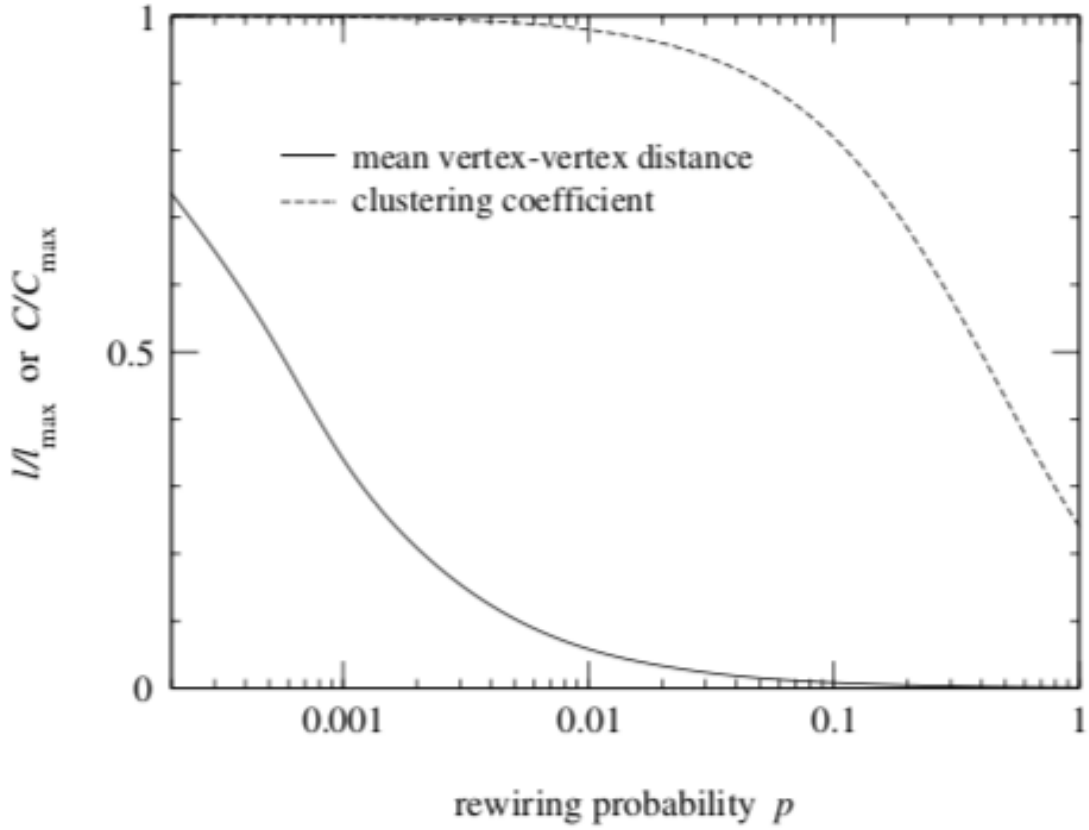


Figure 1.2: Normalized clustering coefficient C/C_{max} and normalized average shortest distance l/l_{max} as a function of rewiring probability p in WS model [113, 158]. Reprinted from "The structure and function of complex networks"[113], by M. E. Newman, 2003, *SIAM review, 45(2)*, p. 210. Copyright 2003 by Society for Industrial and Applied Mathematics.

adding a copy of a node (whose neighbors are the same as the original node) and then mutating the copy.

1.3.2 Null model

While the WS model, BA model, and their different variants can explain part of the properties we observed in real-world networks, it's difficult to tell which model makes more sense in general. After all, even if a model can reproduce all the properties we measured, it still doesn't mean it is the only possible explanation of the real world data. Instead of assuming the network is explicitly generated from a certain dynamics and worrying about the causation, sometimes we just want to use the information we already have and least extra assumptions to generate random graphs. Once we have this ensemble of random graphs, we can infer the conditional probability distribution of unknown properties. In other words, we need a null model which generates graphs satisfying certain constraints but otherwise as random as possible.

1.3.2.1 Dk series

The properties of a graph can range from local properties like degree and degree correlation to global properties like distance and spectrum. A systematic way to describe different levels of constraints is dk-series [125]. A dk-distribution is the joint degree distribution of simple connected subgraphs of size d. A dk-graph is a random graph that has same ik-distribution as the original graph for all $i \leq d$. Thus, compared with the original graph, a 0k-graph has the same average degree $\bar{d}$, a 1k-graph has the same degree distribution $P(d)$, a 2k-graph has the same joint degree distribution $P_{(i,j)\in E}(d_i, d_j)$, and 3k-graph has the same three body correlation $P_V(d_i, d_j, d_k)$ and

$P_\Delta(d_i, d_j, d_k)$. If $d = N$, then Nk-graph is just the original graph.

0k-graphs are basically ER model $G(N, M)$, which is relatively easy to generate. However, generating 1k-graph is already a non-trivial task. While there already exists several ways to generate random graphs with certain degree sequence [16, 145, 111, 112, 7, 35, 118, 106, 154, 89, 46, 88], all of them suffer from some problems [91]. The goal of this dissertation is to analyze, compare and improve the performance of those methods. 2k-graphs are more difficult to generate [13], with very few algorithms available, such as link-swap that preserves degree correlation. Even for this algorithm, there is no theoretical bound for the mixing time. For dk-graphs where $d > 2$, basically the only available method is simulated annealing, which can be not only slow but also inaccurate [17].

The rest of this chapter gives a brief introduction of different methods to generate random graphs with prescribed degree sequence. Depending on whether we require the constraints to be satisfied exactly or approximately, we can classify those methods into two categories: hard constraint methods and soft constraint methods.

1.3.3 Hard constraint methods

Hard constraint method requires every graph generated to have exactly the same degree sequence as the prescribed degree sequence. In analogy to statistical physics, it's similar to micro-canonical ensemble where every state has exactly the same energy.

In order to do statistical inference from this ensemble of random graphs, we prefer the graphs to be sampled uniformly from all possible graphs satisfying the constraints. If uniform sampling is not practical, at least we need to know the relative probability to pick up each graph.

1.3.3.1 Graphicality and Erdos-Gallai theorem

Before we look for simple graphs satisfying a specific degree sequence, we want to make sure such graph exits. A degree sequence is called graphical if there exists at least one simple graph that realizes this degree sequence. To verify whether a sequence is graphical, we can use the Erdos-Gallai theorem [56], which says:

A finite non-increasing sequence of non-negative integers $d_1 \geq d_2 \geq \cdots \geq d_N$ is graphical if and only if $\sum_{i=1}^{N} d_i$ is even and

$$\sum_{i=1}^{k} d_i - k(k-1) \leq \sum_{j=k+1}^{N} \min(d_j, k) \tag{1.7}$$

for all $1 \leq k \leq N$.

A related theorem is the Havel-Hakimi theorem [71, 69], which says:

A finite non-increasing sequence of non-negative integers $d_1 \geq d_2 \geq \cdots \geq d_N$ is graphical if and only if sequence $(d_2 - 1, d_3 - 1, \cdots, d_{d_1+1} - 1, d_{d_1+2}, \cdots, d_N)$ is also graphical.

For a graphical degree sequence, Havel-Hakimi theorem can be used to construct a graph that realizes the degree sequence deterministically.

1.3.3.2 Configuration model (CM)

Configuration model [16, 111, 120] picks up pairs of stubs, or half-edges and connects them randomly. Given a degree sequence $d_1, \cdots, d_N$, we first create N nodes and for each node i give it d_i stubs. Then we randomly pick up two stubs from all the stubs and connect them. We keep doing this until there are no more stubs.

Configuration model generates graph uniformly [111], i.e. with the same probability for all possible graphs having a degree sequence. However, it doesn't guarantee the generated graph to be simple. Since we connect stubs randomly, it's possible that two stubs from the same node are connected (self-loop), or a pair of nodes is connected multiple times (multi-edge). While the proportion of self-loops and multi-edges might be small compared to the total number of edges, it is also unlikely that none of those two situations happens at all [46].

To generate a simple graph, one way is to use configuration model first and delete the self-loops and multi-edges afterward. But then the degree sequence is not preserved. Another way is to revert the last step and do backtracking once a violation of simple graph is detected. However, besides the poor performance of backtracking, the resulting graph is no longer uniformly sampled from the population [109]. Even worse, we lose track of the relative probability of generating that graph. Yet another way is to stop and restart generating a new graph immediately after finding any self-loop or multi-edge. By doing this, the finally generated graph is still sampled uniformly from the population. But since configuration model is unlikely to generate a simple graph, this method could be very slow [20, 16].

1.3.3.3 Sequential importance sampling (SIS)

In order to directly construct graphs without backtracking or rejection, [46] and [20] developed a sequential importance sampling algorithm. The idea of this algorithm is, during the construction of a graph, when we connect a pair of stubs, we want to make sure that after this operation the remaining stubs can still realize a graph.

The SIS algorithm works as follows [46]: Given a degree sequence, we first choose

an arbitrary node and refer to it as Hub. Then we find the Allowed Set of this Hub so that after connecting any node in the Allowed Set with the Hub, the residual degree sequence is still graphical. Once we have the Hub and the Allowed Set, we assign non-zero probability to all the nodes in the Allowed Set and pick up one of them randomly to connect with the Hub. For this same Hub, we keep finding Allowed Set and connecting the Hub with a node in the Allowed Set randomly until the Hub's residual degree reaches 0. Then we can pick up a new Hub and do the same thing. We keep doing this until all the nodes reach zero residual degree.

Define the number of Hubs picked m, the residual degree of Hub i when being picked $\tilde{d}_i$, and probability for Hub i to connect node j in i's Allowed Set p_{ij}. Then the probability to follow this trajectory is just $\prod_{i=1}^{m}\prod_j p_{ij}$. Since for any Hub i, there are $\tilde{d}_i!$ different orders to connect the same set of neighbors, the probability to reach a certain graph G is

$$P(G) = \prod_{i=1}^{m} \tilde{d}_i! \prod_j p_{ij} \ . \tag{1.8}$$

But different graphs should have same importance as long as they all satisfy the constraints, we should compensate for that with weight

$$W(G) = \frac{1}{P(G)} = (\prod_{i=1}^{m} \tilde{d}_i! \prod_j p_{ij})^{-1} \ . \tag{1.9}$$

SIS can generate independent random graphs efficiently without backtracking or rejection. However, since the weight is a product of at least $O(M)$ terms, where M is the number of edges, this weight can have a wide distribution for a large system. In the worst case, a single largest weight can dominate the distribution and make the statistics very unstable.

Usually, the largest weight problem in SIS can be mitigated by resampling [52] the ensemble of samples every now and then. In this way, the difference between the

largest and smallest weight can be bounded. However, in graph sampling problem, different samples may choose different Hubs, making it not clear how to do resampling correctly.

1.3.3.4 Markov chain Monte Carlo method (MCMC)

Instead of constructing a graph from scratch, we can randomize a graph we already have to get a new random graph. This idea leads us to the Markov chain Monte Carlo method (MCMC) [41, 85, 155].

In general, MCMC works as follows. Our goal is to sample states $s \in S$ with distribution $P(s)$. We can achieve this by walking randomly between different states for a long time. If S is ergodic, i.e. there exist path to go from any state to any other state in S with non-zero probability, and the transition probability $T(s \to s')$ satisfies detailed balance $P(s)T(s \to s') = P(s')T(s' \to s)$ for any pair of states (s, s'), then after enough time steps the probability to reach state s is $P(s)$ regardless of the starting point s_0.

Here in hard constraint graph sampling, since the constraints are hard, all states connected by valid moves have the same probability. Thus, we can set the transition probability to be 1, i.e. always accept a move.

If the constraint is degree sequence, a valid move can be degree-preserving link-swap. This is done by picking up two edges randomly, say (a, b) and (c, d), cut them, swap the endpoints and reconnect them. Then we have two new edges (a, d) and (c, b). We can do this as long as we don't introduce self-loops or multi-edges. Otherwise, we simply discard the change and pick up two new edges to try to swap. By performing degree-preserving link-swap (link-swap in short), we changed the topology of the

graph while maintaining the degree sequence unchanged.

To sample random graphs using MCMC, we first get an original graph either from real-world data or generated from some direct construction method like Havel-Hakimi [71, 69] algorithm. Then we keep performing link-swaps for τ_{eq} steps until the system reaches equilibrium. Now we can start sampling by saving graphs every τ_{mix} steps to make sure the states are well mixed, and there is no correlation between different samples.

MCMC is widely used in randomizing graphs. However, it has both theoretical and practical problems. Theoretically, we can only get loose bound for the equilibrium time τ_{eq} and mixing time τ_{mix} in a few cases [66, 41]. One general type of fast mixing graph is a type of graph that can be decomposed into a series of split graphs [59, 152] using canonical decomposition [151, 55], which will be discussed in Chapter 5. While we can measure τ_{eq} and τ_{mix} during the simulation, the result from a finite number of time steps can be misleading especially if the state space S has bottlenecks. In practice, depending on the degree sequence, if we simply pick up two edges randomly, it may be unlikely that they can perform link-swap. This can slow down the simulation.

1.3.4 Soft constraint methods

Instead of requiring every graph generated satisfies the constraints, we can require the ensemble average of generated graphs to satisfy the constraint. Methods having this property are called soft constraint methods.

If hard constraint methods remind us about the micro-canonical ensemble, then soft constraint methods are more similar to canonical ensemble, where the energy of

individual states can fluctuate but the temperature remains the same.

If we decide to use soft constraint methods, then it's no longer necessary to require the degree sequence to be graphical. This may lead to new phenomena.

1.3.4.1 Exponential random graph model (ERGM)

For reader's convenience, this section gives a brief introduction to ERGM. More detailed description can be found in [145, 114, 39].

We want to generate random graph G from all possible graphs of same size with probability P_G, whose graph properties $x_i(G)$ satisfying some constraints on average. That is, $\forall i,\ \sum_G P_G x_i(G) = \bar{x}_i$ and $\sum_G P_G = 1$.

If we want the graphs generated to be as random as possible, we can maximize the entropy of distribution $S = -\sum_G P_G \ln P_G$ while keeping the constraints satisfied on average. Using Lagrange multipliers, we have

$$L = -\sum_G P_G \ln P_G + \sum_i \beta_i (\sum_G P_G x_i(G) - \bar{x}_i) + \alpha (\sum_G P_G - 1) \ . \tag{1.10}$$

To maximize L, we require $\partial L / \partial P_G = 0$ for any G, thus,

$$0 = \frac{\partial L}{\partial P_G} = -\ln P_G - 1 + \sum_i \beta_i x_i(G) + \alpha \ , \tag{1.11}$$

$$P_G = \frac{e^{\sum_i \beta_i x_i(G)}}{Z} \ , \tag{1.12}$$

where

$$Z = \sum_G e^{\sum_i \beta_i x_i(G)} \ . \tag{1.13}$$

In this project, we consider simple, unweighted, undirected graph with the degree sequence $\mathcal{D} = \{d_1,\ d_2, \ldots, d_N\}$ as constraints. Note that in order for ERGM to get

finite solution, $1 < d_i < N-1$ for all i. Expressing the graph in its adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ when node i and node j are connected and $a_{ij} = 0$ when they are not, we have $\forall i,\ d_i = \sum_{j \neq i} a_{ij}$. Thus,

$$P_G = P(A) = \frac{e^{\sum_i \beta_i \sum_{j \neq i} a_{ij}}}{Z} = \frac{e^{\sum_{j \neq i} \beta_i a_{ij}}}{Z} \ . \tag{1.14}$$

Since the graph is simple, the adjacency matrix is symmetric, $a_{ij} = a_{ji}$, thus,

$$P(A) = \frac{e^{\sum_{i<j} (\beta_i + \beta_j) a_{ij}}}{Z} = \frac{\prod_{i<j} e^{(\beta_i + \beta_j) a_{ij}}}{Z} \ , \tag{1.15}$$

where

$$\begin{aligned} Z &= \sum_{a_{ij} \in \{0,1\}} \prod_{i<j} e^{(\beta_i + \beta_j) a_{ij}} \\ &= \prod_{i<j} \Big(\sum_{a_{ij} \in \{0,1\}} e^{(\beta_i + \beta_j) a_{ij}} \Big) \\ &= \prod_{i<j} (1 + e^{\beta_i + \beta_j}) \ . \end{aligned} \tag{1.16}$$

Thus,

$$P(A) = \frac{\prod_{i<j} e^{(\beta_i + \beta_j) a_{ij}}}{\prod_{i<j} (1 + e^{\beta_i + \beta_j})} = \prod_{i<j} \frac{e^{(\beta_i + \beta_j) a_{ij}}}{1 + e^{\beta_i + \beta_j}} \ . \tag{1.17}$$

Now let's define

$$p_{ij} = \frac{e^{\beta_i + \beta_j}}{1 + e^{\beta_i + \beta_j}} = \frac{1}{1 + e^{-(\beta_i + \beta_j)}} \ , \tag{1.18}$$

which can be interpreted as the probability that $a_{ij} = 1$, then

$$P(A) = \prod_{i<j} p_{ij}^{a_{ij}} (1 - p_{ij})^{1 - a_{ij}} \ , \tag{1.19}$$

and the constraints are

$$d_i = \sum_{j \neq i} a_{ij} = \sum_{j \neq i} \frac{1}{1 + e^{-(\beta_i + \beta_j)}} \tag{1.20}$$

for all i.

ERGM is mathematically beautiful. However, if we happen to choose a set of properties with non-linear dependencies, then ERGM may suffer from degeneracy problem, i.e. the distribution of a property become bimodal with low probability near the expected value [76]. Luckily up to now, we haven't observed this degeneracy for degree sequence.

1.3.4.2 Chung-Lu model

ERGM gives us the probability to connect a pair of nodes in 1.18. In sparse case, $p_{ij} \ll 1$, thus, $p_{ij} \approx e^{\beta_i} e^{\beta_j}$. For mathematical simplicity, for now, we temporarily allow self-loops. Now $d_i = \sum_j p_{ij} = e^{\beta_i} \sum_j e^{\beta_j}$. Since $\sum_j e^{\beta_j}$ is a constant, $e^{\beta_i} \propto d_i$. Let's define $e^{\beta_i} = Cd_i$. Since $\sum_{i,j} p_{ij} = \sum_i d_i = 2M$ where M is the number of edges, we have

$$2M = \sum_{i,j} C^2 d_i d_j = C^2 (\sum_i d_i)^2 = C^2 (2M)^2 \ . \tag{1.21}$$

Thus, $C^2 = \frac{1}{2M}$,

$$p_{ij} = \frac{d_i d_j}{2M} \ . \tag{1.22}$$

This is the Chung-Lu model [36].

Chung-Lu model generates graphs efficiently when the largest degree is small enough. However, if the maximum degree d_1 exceeds $\sqrt{\bar{d}}\sqrt{N}$, then the probability to connect two nodes with largest degree $p_{11} > \frac{N\bar{d}}{2M} = 1$. Since a probability should not exceed 1, we should not use Chung-Lu model when d_1 is larger than $O(N^{\frac{1}{2}})$.

This is exactly the case for power-law degree sequence $P(d) \propto d^{-\gamma}$ where $2 < \gamma < 3$. In this range the mean degree is finite but the variance grows as $O(N^{3-\gamma})$, and the

maximum degree grows as $O(N^{\frac{1}{\gamma-1}})$ [45], which is larger than $O(N^{\frac{1}{2}})$.

1.3.4.3 Preferred degree extreme dynamics, XIE and GIE

Another way to generate a random graph is to define some kind of node dynamics and let the system evolve [12]. In preferred degree extreme dynamics [168], each node has its preferred degree. At each time step, a random node is chosen to make itself happier, i.e. make its actual degree closer to its preferred degree. If this node has more neighbors than preferred, it cuts one of its edges randomly. If it has fewer neighbors than preferred, it adds a connection with one of the nodes not yet connect to it randomly. If it happens to have the same number of neighbors as preferred, it does nothing. We can then pick up graphs after the system settled in stationary states.

While preferred degree extreme dynamics can be used to generate random graphs, the exact probability of getting a specific graph is only known in XIE [98]. More general cases are yet to be solved. On the other hand, the extreme dynamics itself has some interesting properties.

If there are only two possible degrees, 0 (eXtreme Introverts) and $N-1$ (eXtreme Extroverts), then we have the eXtreme Introvert Extrovert (XIE) dynamics. After a sufficiently long time, the system reaches an equilibrium where all extroverts are connected and all introverts are not connected [98, 169]. Thus, XIE results in split graph [59, 152]. If we use the difference of number between extroverts and introverts as a control parameter, and the number of crosslinks between extroverts and introverts as order parameter, then the system has a mixed order phase transition [14].

If the introverts and extroverts are not so extreme, we have the Generalized Introvert Extrovert (GIE) dynamics. If the preferred degrees are integers, it seems the system can still reach equilibrium. However, if the preferred degrees are non-integers, the system will not reach equilibrium, and there exist probability current between states [58].

1.3.5 Digression: Graph sampling in graph simplification

When we talk about graph sampling, we mean sampling random graphs from an ensemble of all possible graphs satisfying certain constraints, for example, properties of a real-world graph. This is also called graph generation in some literature [78]. On the other hand, some literature [95, 78] define graph sampling as sampling small subgraphs from an original large graph, so that certain properties of the large graph are preserved [99] in the small subgraphs. While this topic is interesting, it is not the topic we study in this dissertation.

1.4 Dissertation organization

The rest of this dissertation is organized as follows. Chapter 2 talks about the improvement of efficiency and stability of the SIS method. This is a paper [164] prepared for publication. Chapter 3 talks about the difference between ensembles generated using hard constraint methods and soft constraint methods. This is also a paper [165] prepared for publication. This work is done in collaboration with Erich McMillan. Chapter 4 talks about degree distribution, cross-link distribution and correlation in XIE model, and equilibrium in GIE model. This is a collaboration project with

Mohammadmehdi Ezzatabadipour and Dr. Royce K. P. Zia. Some of the results in this chapter are published in [170] while others are in preparation for publication. Chapter 5 talks about split graph and nested networks. Those results can potentially improve the performance of MCMC. The work in this chapter is based on previous work by Dr. Zoltán Toroczkai [55]. Chapter 6 is the conclusion.

Chapter 2

Sequential importance sampling

2.1 Introduction

Ensemble modeling of graphs is a widely used and important technique in Network Science [18]. It is useful in empirical studies when there is limited and/or imperfect knowledge of the structure of the graph [97]. It is also useful in theoretical studies that seek to understand the influence of particular structural constraints on dynamical or other structural properties of graphs [117]. In both cases, statistical analysis of ensembles of graphs that have the known or assumed structural properties, but otherwise vary randomly, is then used to model and predict the behavior of the graph being studied [133, 114, 125]. A common case is to study an ensemble of networks constrained to have prescribed node degrees, which for a graph with N nodes is specified by the degree sequence $\mathcal{D} = \{d_1,\ d_2, \ldots, d_N\}$. The number of graphs realized by a given degree sequence, however, typically grows rapidly with N. Because of this, taking explicit averages over an ensemble of graphs with prescribed degrees is usually

impossible, even numerically, when $N \gtrapprox 10$ [107, 63]. For larger graphs, random samples of graph realizations satisfying the degree constraints are used to calculate ensemble averages. However, generating random graphs with prescribed degrees for this purpose in an unbiased way is a non-trivial problem.

Many kinds of constraints were studied, like degree sequence in undirected graph [46] and directed graph [88], degree correlation [13], and clustering coefficient [125] in undirected graph.

Various methods exist to solve this problem. Soft-constraint methods generate graphs that only satisfy the degree constraints on-average. These include the exponential random graph model (ERGM) [113], and its approximation the Chung-Lu model [36, 35]. Much more difficult hard-constraint methods generate graphs such that each realization precisely satisfies the constraints. Methods in this category include Markov chain Monte Carlo methods (MCMC) [155, 40, 147], and direct construction methods. Among direct construction methods are the configuration model (CM) [111, 113], and ones that use sequential importance sampling (SIS) [46, 20, 163].

However, every known method has problems [91]. Soft-constrained methods can generate an ensemble of graphs that have very different, biased average properties than one generated with hard-constrained methods [76]. MCMC generate samples by performing link-swaps to randomize an initial configuration. The graphs thus generated are correlated and the mixing time needed to generate statistically independent samples is generally not known. In fact, the mixing time is known to be "fast", i.e. increase only algebraically with N, only for a limited class of degree sequences [66]. The CM assigns a number of stubs to each node, equal to the node's degree, and then connects them randomly to form links. This can result in self-loops and

multi-links. For the generation of simple graphs, this causes dead-ends that must be rejected, otherwise uncontrolled sampling biases will occur. This can cause the CM to be ineffective, even effectively useless [31].

SIS takes a similar approach to the CM model, except that they ensure only links that still allow simple graphs to be generated from the remaining stubs to be formed. Thus, SIS generates each graph independently, and it can do so efficiently, without back-tracking or dead-ends, but in general it does not sample the ensemble uniformly. Nevertheless, the relative probability of generating each particular graph can be calculated in some SIS methods, allowing the ensemble to be uniformly sampled by reweighting the samples. Unfortunately, the weights needed for uniform ensemble sampling are generally log-normally distributed [46]. Because of the slow decay of the tail of the log-normal distribution, the number of samples needed to reliably calculate ensemble averages grows exponentially with N. This severely limits the size of the graphs whose ensemble can be reliably sampled.

In this paper, we present methods to improve the efficiency of SIS of simple graphs constrained to have a prescribed degree sequence. As we will see, existing SIS algorithms for this purpose have certain freedoms that can be optimized to improve sampling efficiency. These optimizations can substantially increase the size of the graphs for which ensemble averages can be reliably calculated. Unfortunately, even with these improvements, the size of the graphs whose ensemble can be reliably sampled typically remains limited. For very large graphs, however, we show that a different approach to calculating ensemble averages can be used. In this limit, central limit theorem considerations often allow the joint distribution of the graph properties and the log-weights of the samples to be well approximated as multivariate Gaussian.

When this is true, sampling to estimate the distribution parameters and then calculating the ensemble averages from the estimated distribution rather than directly can produce reliable results.

Depending on the degree sequence $\mathcal{D}$ constraining the ensemble, the efficiency of each of the various random graph generation methods will vary. For particular classes of $\mathcal{D}$ one method may be preferred over another. Although our improvements in SIS and methods of ensemble estimation are expected to be broadly applicable, we focus our efforts in this paper on the especially challenging case of sparse scale-free graphs near their graphicality transition [45]. Scale-free graphs have nodes with degrees that are randomly chosen from a power-law decaying distribution $\rho(d) \propto d^{-\gamma}$ [8]. In the limit of large N, graphs can be constructed for such randomly chosen degree sequences only if $\gamma \geq 2$. For large γ the CM can be used to generate random graphs efficiently. However, as γ gets smaller, and especially for $\gamma < 3$, when structural correlations in the graphs become increasingly important [31], the CM becomes increasingly inefficient. In fact, the problems with all known methods, with possibly the exception of SIS, make generating random graphs near the transition at $\gamma = 2$ difficult. Even in this case though, our methods enable ensembles of large graphs to be reliably sampled. We will demonstrate this by applying our methods to study the global clustering coefficient in the ensemble of networks constrained to have the same degree sequence as the Youtube user friendship network (YoutubeNet) [110]. This network has over 10^6 nodes with degrees approximately distributed as a power-law with $\gamma \approx 2.3$.

The remainder of the paper is organized as follows: Section II gives a brief introduction of SIS and the difficulties we may encounter when using it. Section III

introduces efficient stub sampling, an improvement to the original SIS method. Section IV shows a new way to estimate the ensemble average by measuring the joint distribution parameters. Section V demonstrates the strength of our methods by applying them to a large real-world network, i.e. YoutubeNet, with more than 1 million nodes.

2.2 Sequential importance sampling

SIS algorithms for constructing random simple graphs constrained to have a prescribed degree sequence $\mathcal{D}$ work by directly constructing the graph. They first assign d_i stubs to each node i and then connect pairs of them sequentially to form links until a graph is fully constructed. This can be done by choosing any node, called a "hub", and connecting all of its stubs first, then choosing any other node as hub and connecting all of its unlinked stubs, and repeating until the graph is complete [46]. As long as $\mathcal{D}$ is graphical, i.e. as long as some graph can be constructed with degree sequence $\mathcal{D}$, then the algorithm can be sure to complete construction of a graph, without backtracking. This is accomplished by requiring that the construction of a simple graph can still be completed each time a pair of stubs are connected. This requirement restricts the set of nodes that the hub can connect to. The set of nodes that the hub is allowed to connect to is called the "Allowed Set." The Allowed Set can be efficiently determined by applying a type of Erdős-Gallai graphicality test on the residual degree sequence $\mathcal{D}' = \{d'_1,\ d'_2, \ldots, d'_N\}$ that lists the number of unconnected stubs each node has. This test is an extension of the original Erdős-Gallai test [56] that also works when some of the hub's stubs have already been connected. The test can be completed in a worst case algorithmic complexity of $\mathcal{O}(N)$. The formation of

each link can also be completed in a worst case algorithmic complexity of $\mathcal{O}(N)$. The construction of each complete graph can be completed in a worst case algorithmic complexity of $\mathcal{O}(NM)$, where M is the number of links.

At least for finite N, if the node chosen to connect the hub to is picked randomly from the Allowed Set and each node in the Allowed Set has a finite probability of being picked, then every member of the ensemble of graphs that realize the prescribed degree sequence $\mathcal{D}$ has a finite probability of being constructed. Thus, the ensemble will be sampled ergodically. However, generally it will not be sampled uniformly. The probability of constructing the graphs in the ensemble is not uniform. Fortunately, it can be made uniform by weighting the samples. Note that the relative probability of constructing a particular graph g is

$$P_g = \prod_{i=1}^{m} \bar{d}_i! \prod_{j=1}^{\bar{d}_i} p_{ij} \ , \tag{2.1}$$

where m is the number of hub nodes used in the construction, $\bar{d}_i$ is the residual degree of hub node i when it is chosen as a hub node, and p_{ij} is the probability of choosing node j from the Allowed Set when it is picked to connect to hub i. Therefore, an unbiased estimator of an observable Q from n randomly generated samples $\{s_1, s_2, \ldots, s_n\}$ is

$$\langle Q \rangle = \frac{\sum_{i=1}^{n} w_{s_i} Q(s_i)}{\sum_{i=1}^{n} w_{s_i}} \ , \tag{2.2}$$

where $w_{s_i} = P_{s_i}^{-1}$ are weights for each sample, and the denominator is a normalization factor. In the limit of large n, $\langle Q \rangle$ is equivalent to the ensemble average of Q.

The generation of the sample weights is a random multiplicative process. As such, by central limit theorem arguments, for large graphs at least, they nominally have a

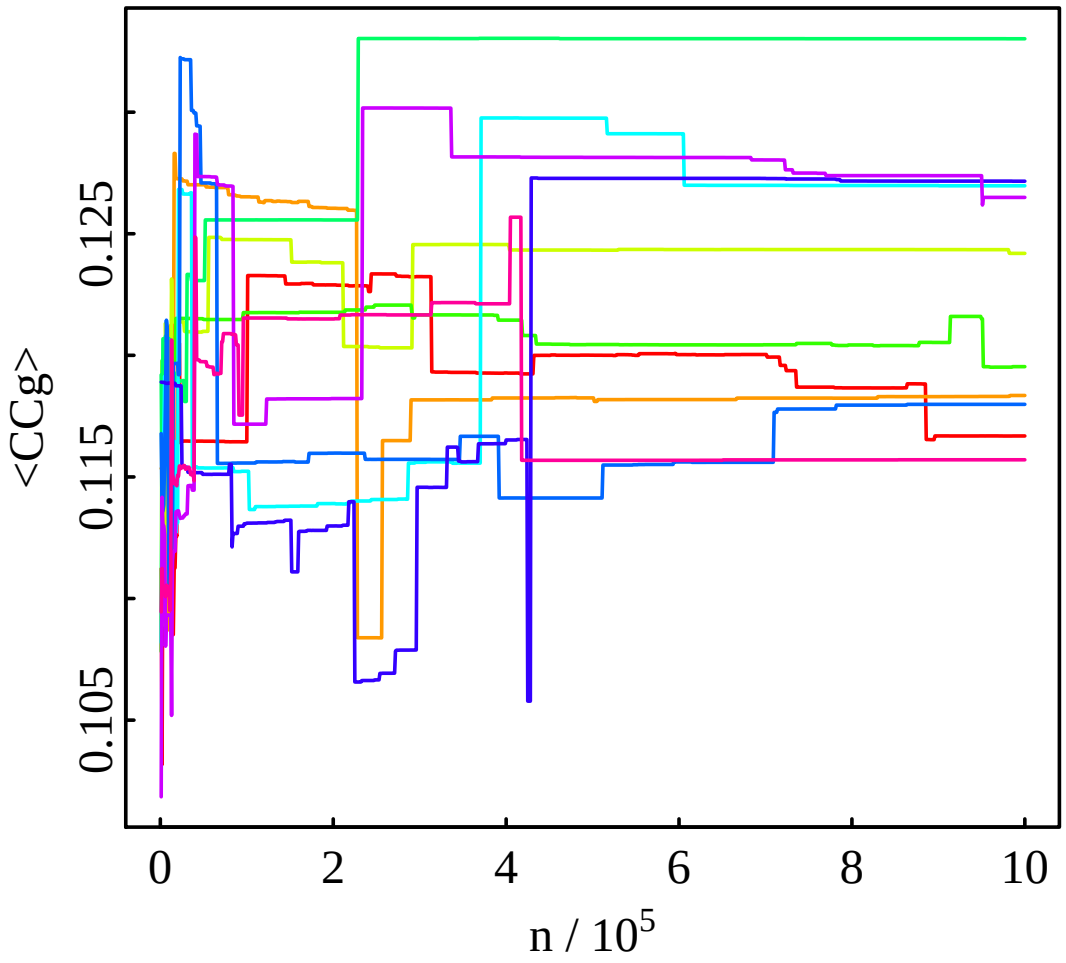


Figure 2.1: Running weighted average of the global clustering coefficient CC_g using SIS. Results of 10 different runs for average of an ensemble of graphs with the same prescribed degree sequence are shown. The sequence is of length $N = 100$ and the degrees were chosen randomly from a power-law distribution with $\gamma = 2$.

log-normal distribution [46]

$$P(w) = \frac{1}{\sigma\sqrt{2\pi}w} e^{-(\ln(w)-\mu)^2/(2\sigma^2)} \ . \tag{2.3}$$

Here, μ and σ are the mean and standard deviation of the log-weights, respectively. Both μ and σ^2 are expected to scale proportionally with M. Unfortunately, a log-normal distribution decays very slowly, more slowly than an exponential, and thus has a "fat tail." Because of the slow decay, sample weights can vary by many orders of magnitude in a typical run for even a modest size network with tens of nodes. This spread of sample weights presents problems for convergence of ensemble averages.

For example, Figure 2.1 shows ten different runs that calculate the running weighted average of the global clustering coefficient, CC_g, for an ensemble of graphs with the same prescribed degree sequence. The CC_g is also known as the transitivity. It is defined as $CC_g = \frac{(number\ of\ triangles)\times 3}{(number\ of\ connected\ triples)}$, where a "connected triple" means three nodes uvw connected with links (u, v) and (v, w). The link (u, w) may or may not be present. The factor of three in the numerator prevents over-counting as each triangle gets counted three times when the possible connected triples are counted. The triangle uvw for instance contains the triples uvw, vwu, and wuv [114]. The CC_g measures how likely that "the friend of my friend is also my friend."

From the figure, clearly, it is difficult to know if and when convergence has occurred, because the running average of an observable can jump when a sample with large weight is suddenly included. In order to ensure convergence a large number of samples n are required. To help estimate the error in a weighted average with sample size n and weights $\{w_i;\ i = 1, 2, \ldots, n\}$, Kish's Effective Sample Size $n_{eff} = \frac{(\sum_{i=1}^{n} w_i)^2}{\sum_{i=1}^{n} w_i^2}$ can be used to estimate the equivalent number of independent unweighted samples [90]. For SIS, assuming a log-normal distribution of sample weights, Eq. 2.3, the

expected moments of the weight distribution are [82]

$$E[w^s] = e^{s\mu + \frac{1}{2}s^2\sigma^2} \ , \tag{2.4}$$

and the expected effective sample size is

$$E[n_{eff}] = \frac{n^2 E[w]^2}{nE[w^2]} = \frac{n^2 e^{2(\mu+\frac{1}{2}\sigma^2)}}{ne^{2\mu+2\sigma^2}} = \frac{n}{e^{\sigma^2}} \ . \tag{2.5}$$

Since $\sigma^2 \sim M$, $E[n_{eff}] \sim n/e^M$. That is, the number of samples required to calculate reliable ensemble averages increases exponentially with the number of links M. This limits the size of graphs that can be effectively sampled with SIS [91].

2.3 Optimized sequential importance sampling

Although the biased nature of SIS graph construction limits the size of graphs that can be reliably sampled, by optimizing the sampling, it becomes possible to reliably sample significantly larger ones. Since by Eq. 2.5 the estimated effective sample size is $\frac{n}{e^{\sigma^2}}$, the smaller the variance of log-weight, the larger the effective sample size and the more efficient the algorithm is. Thus, SIS is optimized when σ is minimal.

Note that the SIS algorithm described in Section 2.2 has two freedoms. The first is that when choosing a hub, any node can be picked. The second is that probabilities of picking the various nodes in the Allowed Set to connect the hub to can be set arbitrarily, as long as every node in the Allowed Set has a non-zero probability of being chosen. Different choices for these two freedoms typically produce different sample weight distributions. The ideal choices will minimize the variance of the weight distribution.

The optimal choices may depend on the prescribed degree sequence $\mathcal{D}$. Here we focus on the difficult case of sequences with power-law distributed degrees $\rho(d) \propto d^{-\gamma}$

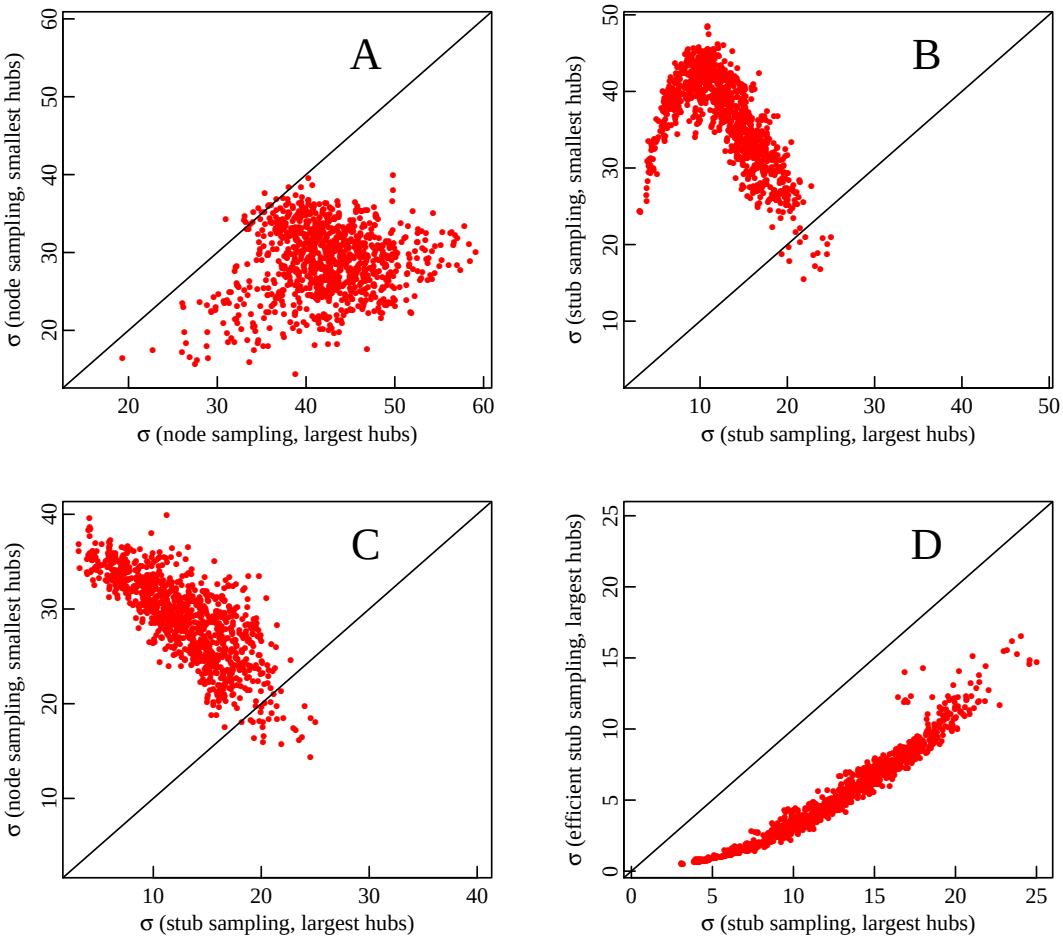


Figure 2.2: Comparison of standard deviation σ of the log-weight distribution for different freedom choices in SIS sampling. Each red dot shows the results for one random power-law distributed degree sequence with $N = 1000$ nodes. A: node sampling with choosing the smallest vs. largest nodes as hubs; B: stub Sampling with choosing the smallest vs. largest nodes as hubs; C: node sampling with smallest nodes as hub vs. stub sampling with largest nodes as hubs; D: efficient stub sampling vs. stub sampling with largest node chosen as hubs in both cases.

with $\gamma = 2$ and no artificial degree cutoff [31], so that the maximum degree possible is $N-1$, the maximum for a simple graph. We have numerically explored a variety of options for both of the freedoms. For choosing the hub node, we have explored either choosing a node with the largest residual degree or one with the smallest residual degree at the time of choosing during the graph construction process. For choosing nodes in the Allowed Set to connect the hub to, we have also explored a number of methods. Here we will discuss results for three: node sampling, where each node in the allowed set is equally likely to be chosen, stub sampling, where each node has a probability proportional to its residual degree d'_i to be chosen, and efficient stub sampling, where each node in the allowed set has a probability of being chosen according to

$$p_{ij} \propto \left(1 + \frac{\bar{d'}(N' - 1 - d'_i)(N' - 1 - d'_j)}{d'_i d'_j (N' - 1 - \bar{d'})}\right)^{-1} . \tag{2.6}$$

For efficient stub sampling, d'_i is the residual degree of the chosen node in the Allowed Set, d'_j is the residual degree of the hub node, N' is the number of nodes left that still have stubs to be connected, and $\bar{d'}$ is the average degree of the nodes left. All of these choices can be implemented so that the SIS algorithm has a worst-case computational complexity of $\mathcal{O}(NM)$.

Node sampling and stub sampling are obvious, simple choices. Node sampling combined with choosing the largest residual degree node as hubs was suggested in [46], while stub sampling combined with choosing the smallest residual degree node as hubs was suggested in [20]. Efficient stub sampling is inspired by the probability of connecting a pair of nodes in a soft-constrained exponential random graph model (ERGM) with prescribed degrees [39]. The specific form in Eq. 2.6 is derived by finding approximate solutions of ERGM using mean-field approach. More details can be found in the appendix. We have also explored using probabilities for choosing

nodes from the Allowed Set, including ones that are proportional to a power of the residual degree and ones that are an exponential function of the residual degree. Results for these various methods are given in the supplemental information. The most optimal choice we have found is efficient stub sampling.

To compare the different choices, we considered 1000 randomly chosen power-law distributed degree sequences for $N = 1000$ nodes. We generated 1000 graphs for each sequence using different algorithmic freedom choices and calculated the standard deviation of the logarithm of the sample weights resulting from each different choice. Figure 2.2 shows the results. Figure 2.2A compares choosing smallest vs. largest nodes for hubs when using node sampling. Each of the red dots represents the results from one degree sequence. When a dot lies below the diagonal line the standard deviation of the log-weight of samples generated by choosing smallest hubs is less that of those generated by choosing largest hubs. As almost all of the dots lie below the diagonal, choosing the smallest hubs is generally better than choosing the largest hubs for node sampling. Choosing the smallest hubs and node sampling gives an assortative preference of connecting pairs of nodes with smaller residual degrees. This result indicates that an assortative preference for link formation leads to a smaller log-weight variance of the samples.

A similar analysis comparing choosing smallest vs. largest nodes for hubs when using stub sampling is shown in Figure 2.2B. In this case, almost all of the dots lie above the diagonal, indicating that choosing the largest hubs is generally better than choosing the smallest hubs for stub sampling. This result also indicates that an assortative preference for link formation leads to a smaller log-weight variance of the samples, as choosing the largest hubs and stub sampling gives an assortative preference of connecting pairs of nodes with larger residual degrees.

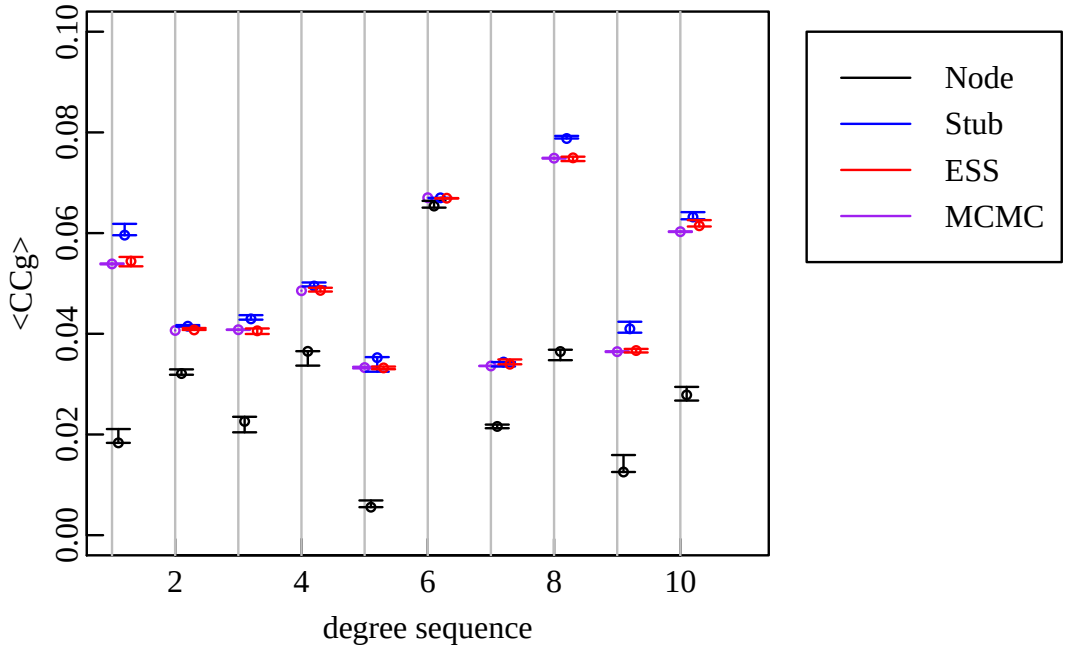


Figure 2.3: Weighted average of the global clustering coefficient CC_g for 10 different random power-law distributed degree sequences with $N = 1000$ nodes, calculated directly, using different sampling methods. Results are shown in black for node sampling with smallest nodes as hubs, blue for stub sampling with largest nodes as hubs, and red for efficient stub sampling with largest nodes as hubs. Purple shows the results for MCMC. (Error bars are 95% confidence interval calculated using bootstrapping method. [64, 44])

Figure 2.2C analogously compares the best choices from Figure 2.2A and Figure 2.2B: choosing smallest nodes for hubs when using node sampling vs. choosing largest nodes for hubs when using stub sampling. Almost all of the dots lie above the diagonal, indicating that stub sampling generally has a smaller log-weight variance and is better than node sampling. Simple stub sampling is, however, not the optimal choice. Figure 2.2D compares efficient stub sampling vs. stub sampling when choosing largest nodes for hubs. For all of the sequences studied the dots clearly lie below the diagonal, indicating that efficient stub sampling is better than simple stub sampling. Choosing largest nodes as hub combined with efficient stub sampling is the most optimal method of SIS graph construction for power-law distributed sequences we have found.

The improvement in sampling reliability that can be obtained by using efficient stub sampling is shown by example in Figure 2.3. In the figure, ensemble averages for the CC_g for ten different power-law distributed degree sequences calculated using different sampling methods are compared. The sequences each have $N = 1000$ nodes. Results for SIS sampling using three different freedom choices are shown. For each sequence, 1000 samples were generated, and weighted averages were calculated. Error bar for each sequence is 95% confidence interval of the weighted average, calculated using bootstrapping method. [64, 44]

To provide a comparison for the SIS results, we also used link-swap MCMC [149] to calculate the CC_g of the sequences. Simulation runs consisting of $1000 \times 2M$ link-swaps were performed for each sequence. Each run began with a Havel-Hakemi graph [71, 69], but the use of other types of initial graphs was explored and found to be statistically irrelevant to the sampling results. Graphs were sampled at intervals of $2M$ link-swaps during the runs, producing 1000 samples for each sequence. The

mean and its standard error of the CC_g for the 1000 were then calculated and are shown in the figure.

In theory, all SIS methods will converge to same, correct result as long as there are enough samples [46]. However, in practice, the number of samples that can be generated is limited. The method that can give the most accurate result for a given number of samples is, thus, preferred. Assuming that MCMC gives the correct result, Figure 2.3 indicates that node sampling tends to underestimate the CC_g, while stub sampling tends to overestimate it. Efficient stub sampling, however, gives results that are consistent with those of MCMC. The bias that the different sampling methods demonstrate may, of course, be different for another quantity, but, at least for the CC_g, efficient stub sampling appears to give unbiased results that allow accurate estimation for sequences as large as several thousand nodes.

In order to more accurately estimate the size of graphs that can be feasibly sampled with the different SIS methods, the distribution of the standard deviation of the log-weights for different degree sequence lengths must be known. For each SIS method and sequence length, we generated 1000 random sequences and 1000 graph samples for each sequence. From these results, the standard deviation of the log-weights σ was calculated for each sequence, and the distribution of σ for each N and SIS method compiled. Figure 2.4 shows the results. Generally, as expected, for all N, efficient stub sampling performs best, followed by stub sampling, and then node sampling worst. The dashed grey line indicates the value of $\sigma = 2.6$ for which 1000 weighted samples have an effective sample size of unity, according to Eq. 2.5 where the effective sample size is $E[n_{eff}] = n/e^{\sigma^2} = 1000/e^{2.6^2} \approx 1$. This line provides a rough estimation of the limits of the feasibility of SIS sampling. For node sampling, the maximum length of sequences that can feasibly be sampled is only about 30. With stub sampling,

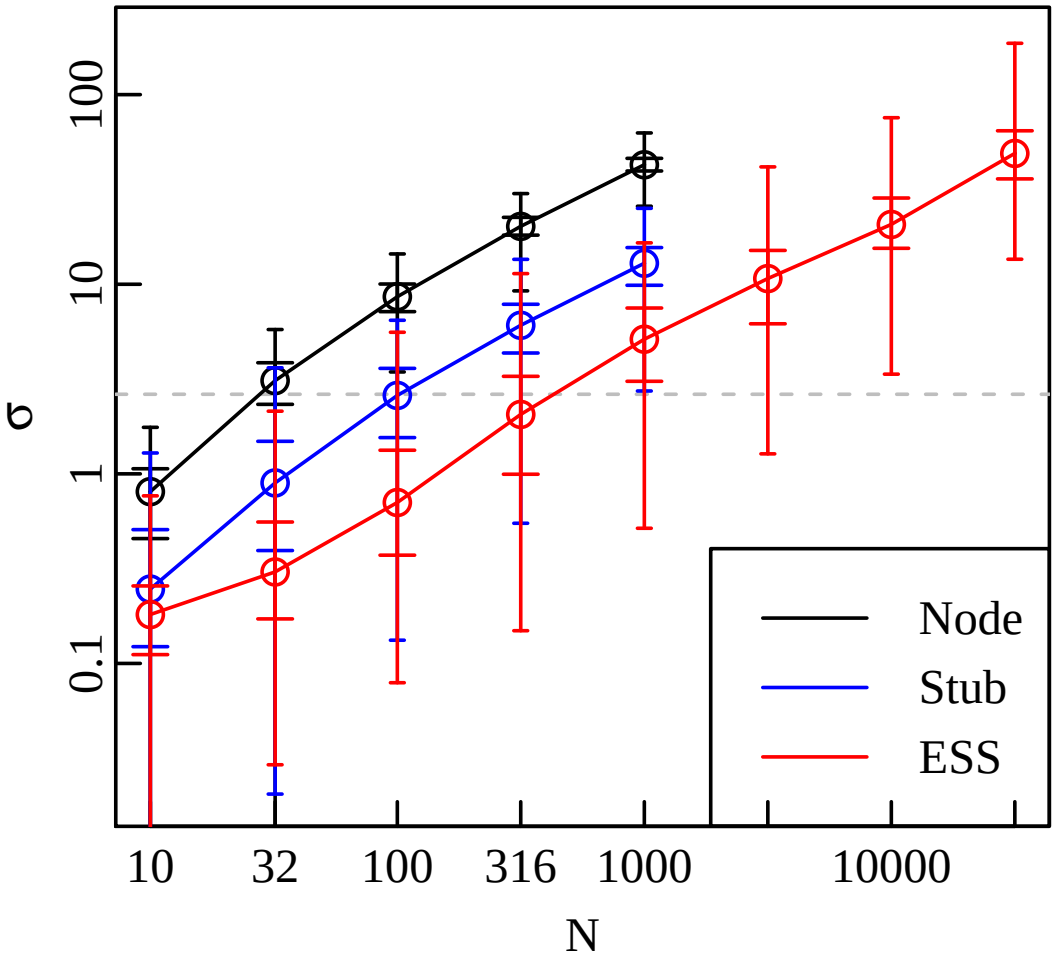


Figure 2.4: Distribution range of the standard deviation of log-weight for SIS using different freedom choices. Black is for node sampling, blue is for stub sampling, and red is for efficient stub sampling. For each sequence length N and for each method, the minimum (bottom of bar), 25% quantile (lower wide error bar), median (circle), 75% quantile (upper wide error bar) and maximum (top of bar) of distribution is shown. The dashed grey line indicates where 1000 samples would produce one effective unweighted sample.

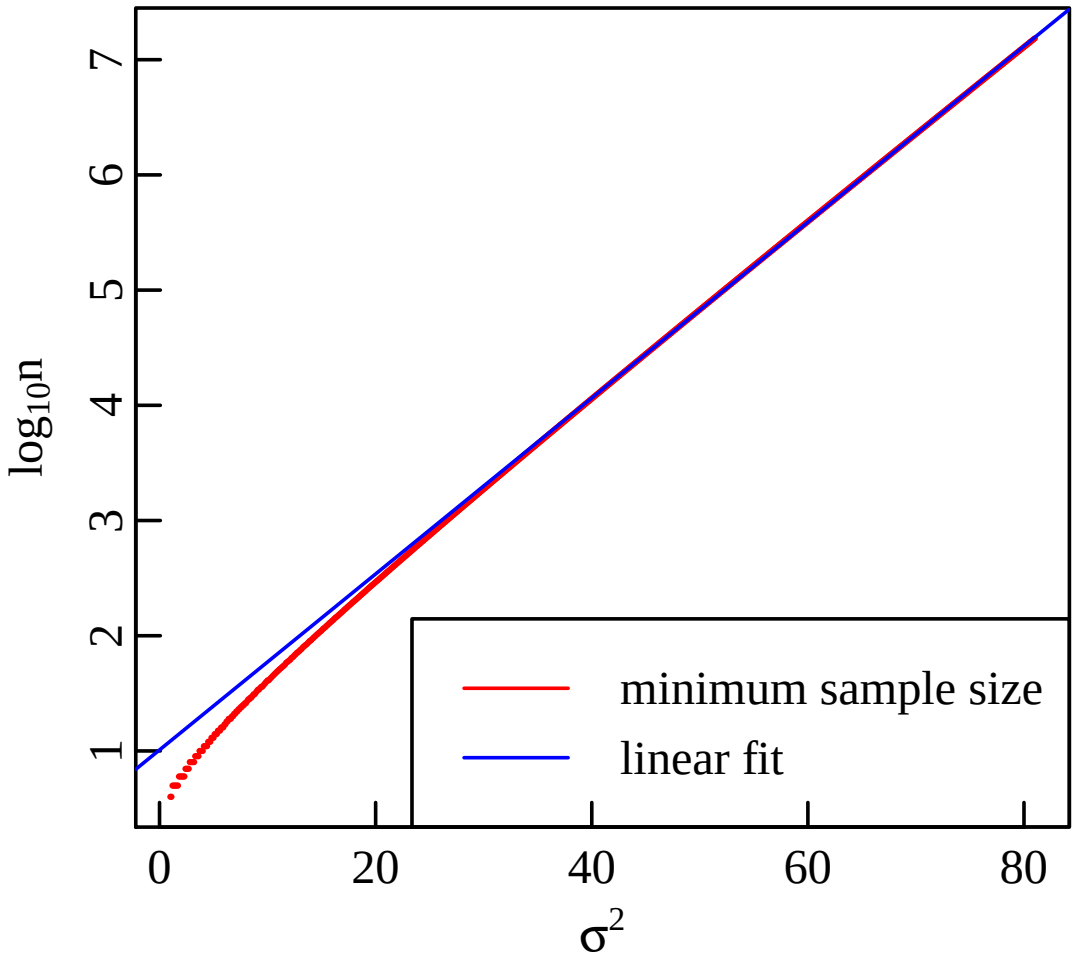


Figure 2.5: Minimum sample size n that the largest weight no longer dominates. (sample size that the expected largest weight equals to half the expected total weight.) Red dots are the minimum sample sizes. Blue line is a linear fit with formula $\log_{10} n = 0.076\sigma^2 + 1.009$.

perhaps sequences with $N \leq 100$ can be sampled. While for efficient stub sampling, maybe sequences with $N \approx 300$ can be feasibly sampled. Note that this estimate is smaller than what the results in Figure 2.3 for the CC_g indicate. Perhaps for certain measurables, estimates can be feasibly made for longer degree sequences. However, as Figure 2.4 shows, the distribution of σ is quite broad and can be very large for a given sequence, even for smaller N. This suggests that caution must be used when using SIS regardless of sequence length.

2.4 Sampling large graphs

Despite the significant improvement in the size of graphs that can be feasibly sampled when efficient stub sampling is used, it is still not possible to use it to directly study large graphs. This is due to the slow decay of the distribution of the sample weights. The weight of a sample is inversely proportional to the relative probability of generating a particular graph, Eq. 2.1, which due to the random multiplicative nature of the graph construction process, according to the central limit theorem [19], is generally expected to have a log-normal distribution in the limit of large graphs [46]. The logarithm of the weights are normally distributed. As argued in Section 2.2, an exponentially large number of samples is therefore required for direct weighted averages of measurable quantities to reliably converge to their ensemble averages. This makes using weighted sample averages to estimate ensemble averages impossible for large graphs. A completely different approach is required to calculate ensemble averages for large graphs.

Here we show that the knowledge that the sample weight distribution has a log-normal form can be used advantageously to calculate ensemble averages for large

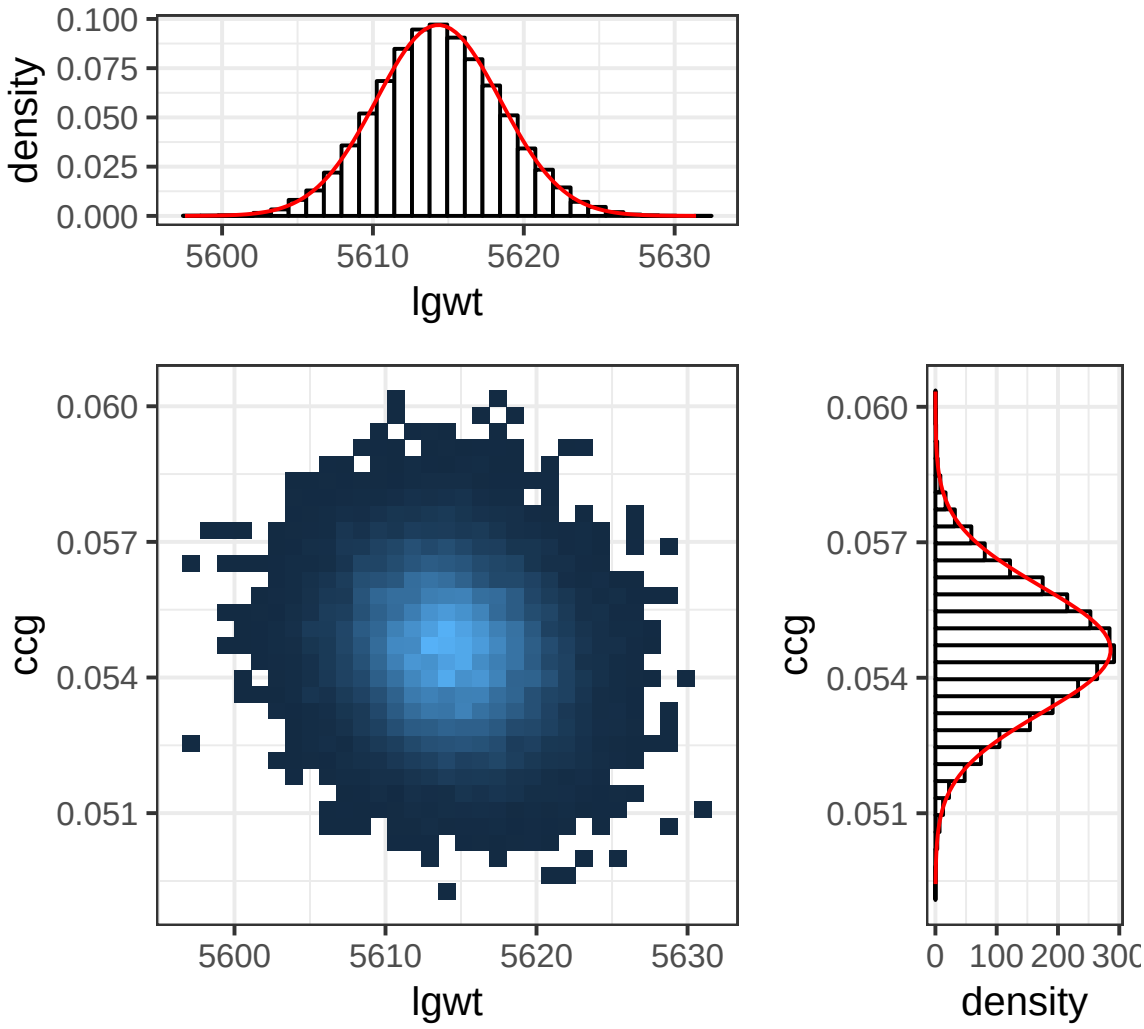


Figure 2.6: Joint distribution of the logarithm of the sample weights (log-weights) and the CC_g of an ensemble for a prescribed degree sequence with $N = 1000$. Note the approximate bivariate normal form of the joint distribution shown in the central plot and the approximate normal form of the marginal distributions of the log-weights and of the CC_g shown as projections at the edges of the figure.

graphs. That is, when the graph is large enough to *assume* that the sample weight distribution is log-normal and that the joint distribution of the sample log-weights and a measurable quantity of interest Q is a bivariate normal distribution, ensemble averages can be estimated reliably indirectly, by using the graph sampling to first estimate the parameters of the multivariate distribution and then using the estimated distribution to make the ensemble average estimates. The validity of the bivariate normal distribution assumption can be seen by example in Figure 2.6. The figure shows the joint distribution of the sample log-weights and the CC_g for the first degree sequence studied in Figure 2.3. Although the degree sequence has only $N = 1000$, the joint distribution has an approximate bivariate normal form. This can be clearly seen by the form of the marginal distributions for the log-weights and the CC_g that are also shown in the figure.

Of course, the central limit only applies in the limit of large sequences. For a finite length sequence, the joint probability distribution of sample log-weight and Q need not have a bivariate normal form. However, as N increases, the likelihood that it does have a bivariate normal distribution increases. Figure 2.7 shows this is true when $Q = CC_g$. In the figure, for each value of N, 1000 different power-law distributed degree sequences were considered. For each sequence, 1000 sample graphs were generated and the joint probability distribution (JPD) of the sample log-weight and the CC_g were calculated. The resulting JPDs were then tested to determine if they had a bivariate normal form using Henze-Zirkler test [72], Royston test [135] and Mardia test [105]. All tests were applied using a significance threshold of 0.05 for deviation from bivariate normal form. Error bars correspond to one σ standard statistical error. As anticipated, the fraction of sequences with bivariate normal JPDs increases toward unity with N. It should be noted, however, that for a given

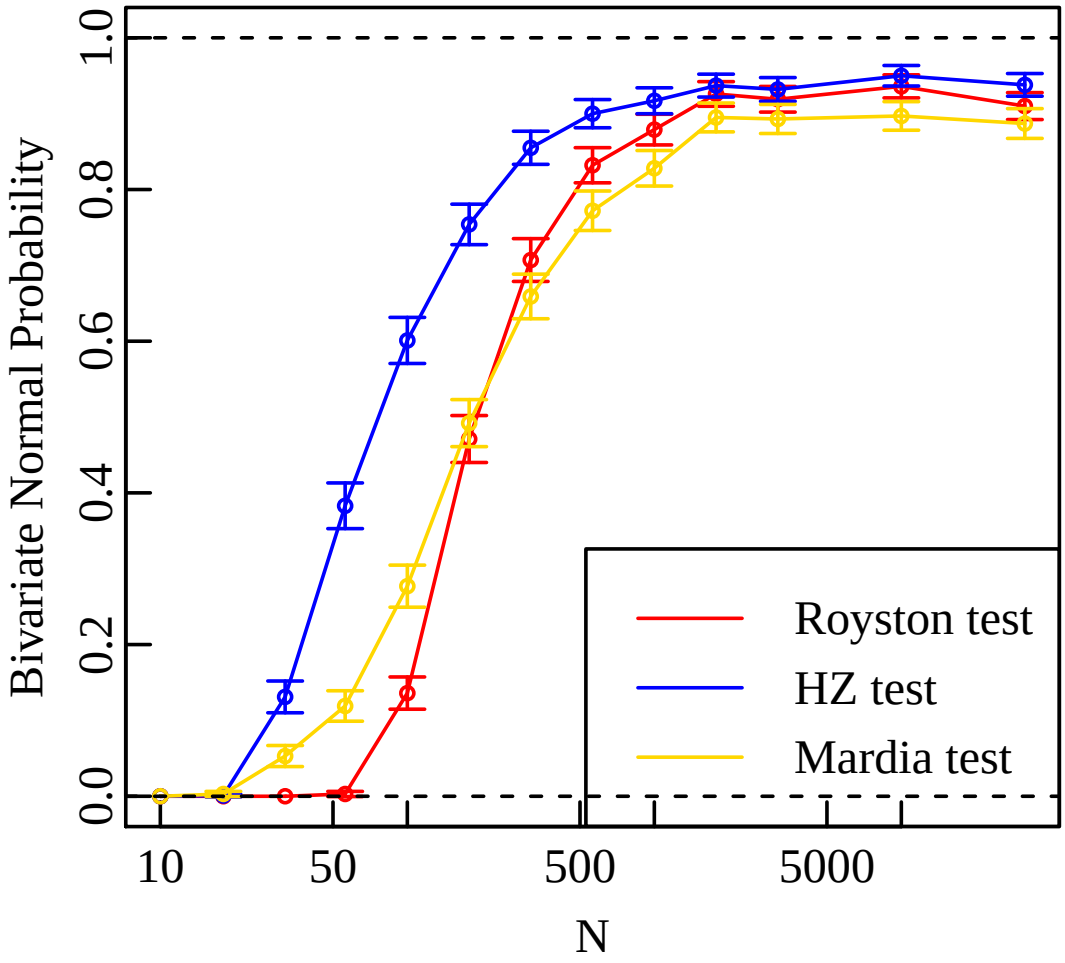


Figure 2.7: Probability that the joint probability distribution of sample log-weight and CC_g has a bivariate normal form as a function of prescribed degree sequence length. Fraction of sequences satisfying the Henze-Zirkler test [72], the Royston test [135] and Mardia test [105] are shown in blue, red, and gold respectively. (Error bars show 95% confidence interval. For each system size N 1000 degree sequences are tested and 1000 graphs per sequence are generated.)

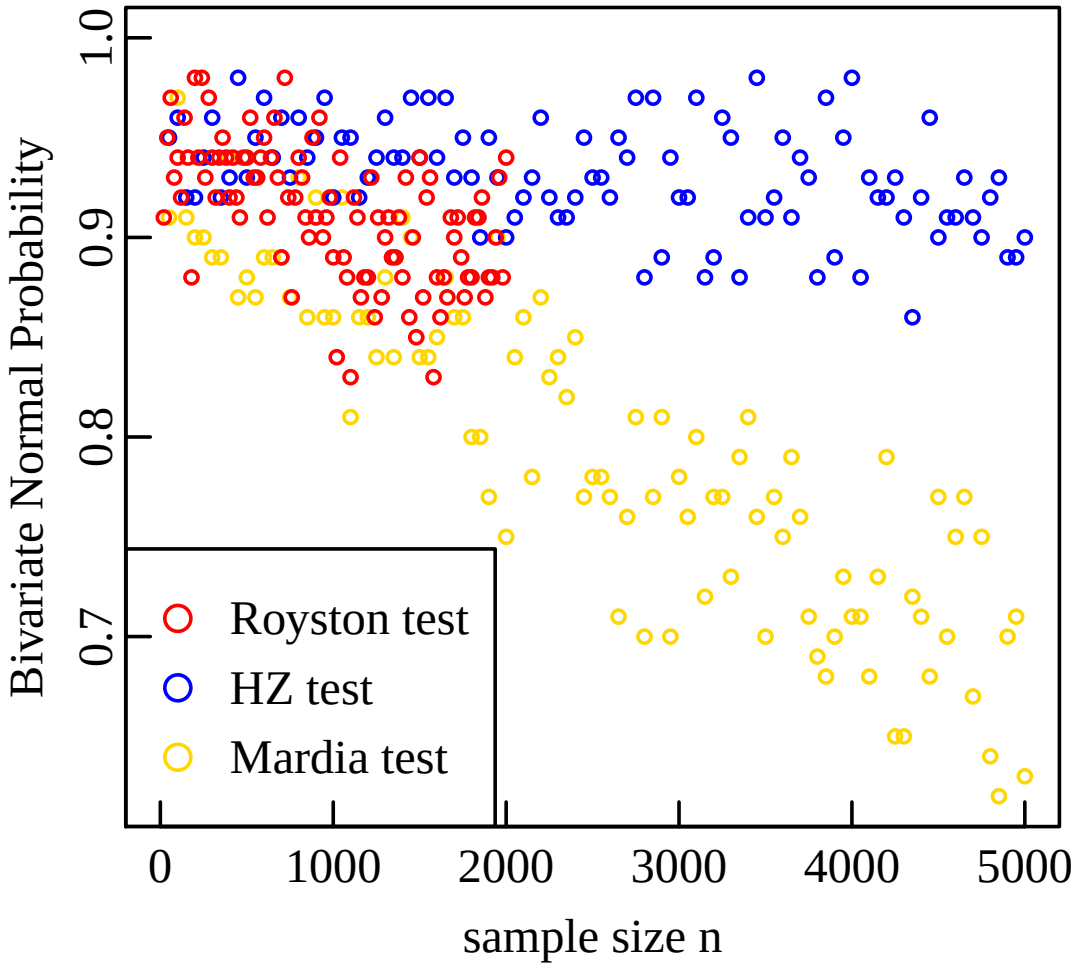


Figure 2.8: Bivariate normal probability of a sequence with different sample size. Blue for Henze-Zirkler test, red for Royston Test, and gold for Mardia test. It seems that as the sample size increases, it is less and less likely that the joint distribution passes the bivariate normal test. (For this specific degree sequence with 1000 nodes, a pool of 10^6 graphs are generated and for each sample size n we resampled 100 times from this pool.)

prescribed degree sequence, as the number of samples increases, the probability that its JPD will be bivariate normal will decrease as the precision of the measurement of the JPD begins to reveal deviation from bivariate normal form. JPDs are thus typically found to be bivariate normal for a moderate number of samples. Whether or not particular JPD is approximately bivariate normal should be tested.

If the JPD is assumed to be bivariate normal, then it is completely characterized by five independent parameters: the mean $\bar{Q}$ and variance σ_Q^2 of sampled value of the measurable Q, the mean $\bar{y}$ and variance σ_y^2 of the sample log-weight $y \equiv \ln w$, and V_{Qy} the covariance of Q and y. Each of these parameters can be estimated by simple, unweighted sample averages:

$$\bar{Q} = \frac{1}{n} \sum_{i=1}^{n} Q(s_i) \ , \tag{2.7}$$

$$\sigma_Q^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left[Q(s_i) - \bar{Q} \right]^2 \ , \tag{2.8}$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} \ln w_{s_i} \ , \tag{2.9}$$

$$\sigma_y^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(\ln w_{s_i} - \bar{y} \right)^2 \ , \tag{2.10}$$

and

$$V_{Qy} = \frac{1}{n-1} \sum_{i=1}^{n} \left[Q(s_i) - \bar{Q} \right] \left[\ln w_{s_i} - \bar{y} \right] \ . \tag{2.11}$$

In terms of these parameters, the JPD can be written [87]

$$P(Q, y) = \frac{1}{2\pi \sigma_Q \sigma_y \sqrt{1-\rho^2}} e^{-z/[2(1-\rho^2)]} \ , \tag{2.12}$$

where

$$z = \frac{(Q-\bar{Q})^2}{\sigma_Q^2} - \frac{2\rho (Q-\bar{Q})(y-\bar{y})}{\sigma_Q \ \sigma_y} + \frac{(y-\bar{y})^2}{\sigma_y^2} \ , \tag{2.13}$$

and correlation

$$\rho = \mathrm{cor}(Q, y) = \frac{V_{Qy}}{\sigma_Q \ \sigma_y}. \tag{2.14}$$

From estimates of these parameters, an estimate of $\langle Q \rangle$, the ensemble average of Q, can be made. To do so note that $\langle Q \rangle$ is the weighted average of Q. The weight of a point (Q, y) is e^y, where y is a log-weight. Thus, the weighted distribution of Q is

$$P_w(Q) = \frac{\int P(Q, y) e^y dy}{\int \int P(Q, y) e^y dQ dy} \ , \tag{2.15}$$

in which

$$\int P(Q, y) e^y dy = e^{\mu_y + \sigma_y^2/2} \frac{1}{\sqrt{2\pi}\sigma_Q} e^{-(Q - (\bar{Q} + \rho\sigma_Q \ \sigma_y))^2/(2\sigma_Q^2)} \ , \tag{2.16}$$

and

$$\int \int P(Q, y) e^y dQ dy = e^{\mu_y + \sigma_y^2/2} \ , \tag{2.17}$$

thus,

$$P_w(Q) = \frac{1}{\sqrt{2\pi}\sigma_Q} e^{-(Q - (\bar{Q} + \rho\sigma_Q \ \sigma_y))^2/(2\sigma_Q^2)} \ . \tag{2.18}$$

Thus, the weighted distribution of Q is a normal distribution. Its mean is the ensemble average

$$\langle Q \rangle = \bar{Q} + \rho \ \sigma_Q \ \sigma_y = \bar{Q} + V_{Qy}. \tag{2.19}$$

The ensemble average of Q is thus a function of not only the unweighted average of Q but also of the covariance of Q and y. To determine the statistical error in the estimation of $\langle Q \rangle$, note that since $\bar{Q}$ and V_{Qy} are independent

$$Var(\langle Q \rangle) = Var(\bar{Q}) + Var(V_{Qy}) \ , \tag{2.20}$$

where

$$Var(V_{Qy}) = \sigma_Q^2 \ \sigma_y^2 \ (1 + \rho^2) \ , \tag{2.21}$$

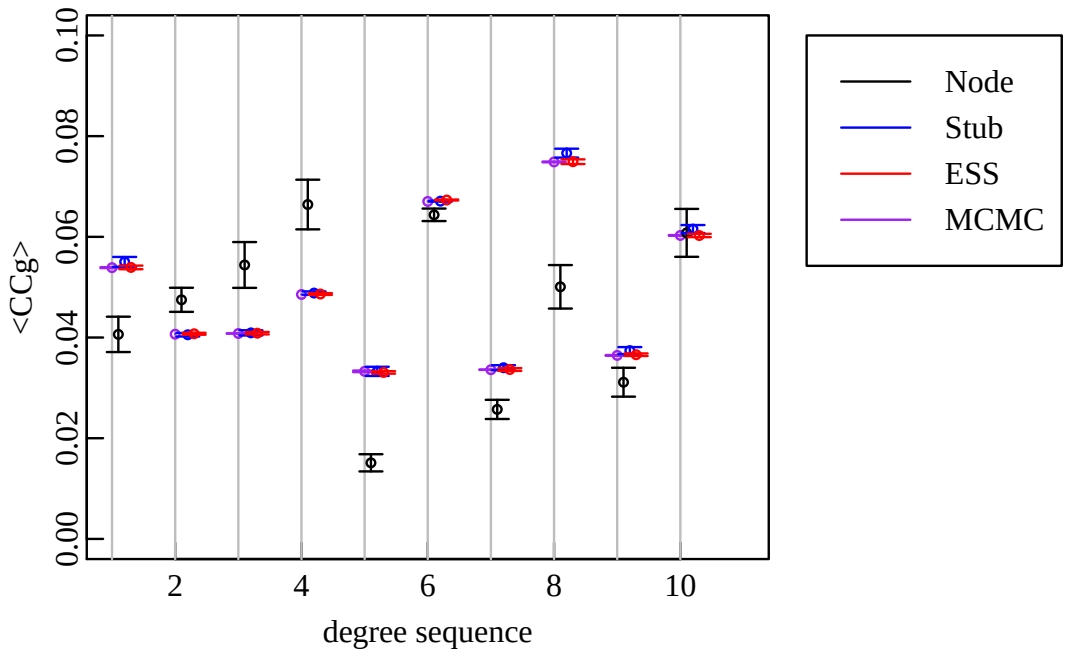


Figure 2.9: Weighted average of the global clustering coefficient CC_g for the same 10 random power-law distributed degree sequences with $N = 1000$ nodes considered in Figure 2.3, calculated by assuming a bivariate normal form of the CC_g-log-weight joint probability distribution JPD and estimating the parameters of the JPD by using different sampling methods. Results are shown in black for node sampling with smallest nodes as hubs, blue for stub sampling with largest nodes as hubs, and red for efficient Stub sampling with largest nodes as hubs. Purple shows the results for MCMC. (Error bars show 95% confidence interval.)

since covariance matrices of multivariate normal distributions have Wishart distributions [160]. Thus, the standard error in estimating $\langle Q \rangle$ is

$$SE(\langle Q \rangle) = \sqrt{\frac{\sigma_Q^2 + \sigma_Q^2 \, \sigma_y^2 \, (1+\rho^2)}{n}} \; . \tag{2.22}$$

Since typically $\sigma_y \gg 1$, the relative error

$$z = \frac{SE(\langle Q \rangle)}{\sigma_Q} = \sqrt{\frac{1 + \sigma_y^2 \, (1+\rho^2)}{n}} \sim \frac{\sigma_y}{\sqrt{n}} \; . \tag{2.23}$$

Thus, given z, the number of samples required will be $n \sim \frac{\sigma_y^2}{z^2}$, which is more scalable than $e^{\sigma_y^2}$ in Eq. 2.5.

All of the independent parameters that characterize the JPD depend most strongly on the center part of the distribution, and not on its tails. Thus, they can be accurately estimated with a relatively small number of samples. Accurate estimates of ensemble averages and their statistical error can then be made using Eqs. 2.19 and 2.22. Figure 2.9 shows results for ensemble average of the CC_g calculated in this way for the same 10 prescribed sequences studied in Figure 2.3. Again, results for SIS sampling using three different freedom choices are shown. For each sequence and each freedom choice, the same $1,000$ samples used for Figure 2.3 were reanalyzed, and the mean and 95% confidence interval are shown. For each sequence improved results are shown. The JPD for every set of samples, except those for sequence 6, were found to be bivariate normal using both the Henze-Zirkler test and the Royston test.

The difference between calculating an ensemble average by a direct weighted average and by estimating the parameters of the bivariate normal JPD are shown in Figure 2.10. The figure shows the running ensemble average CC_g, calculated with both methods, in twenty different runs for sequence 1 in Figure 2.3 and Figure 2.9. Efficient stub sampling was used in all runs. The figure shows that the

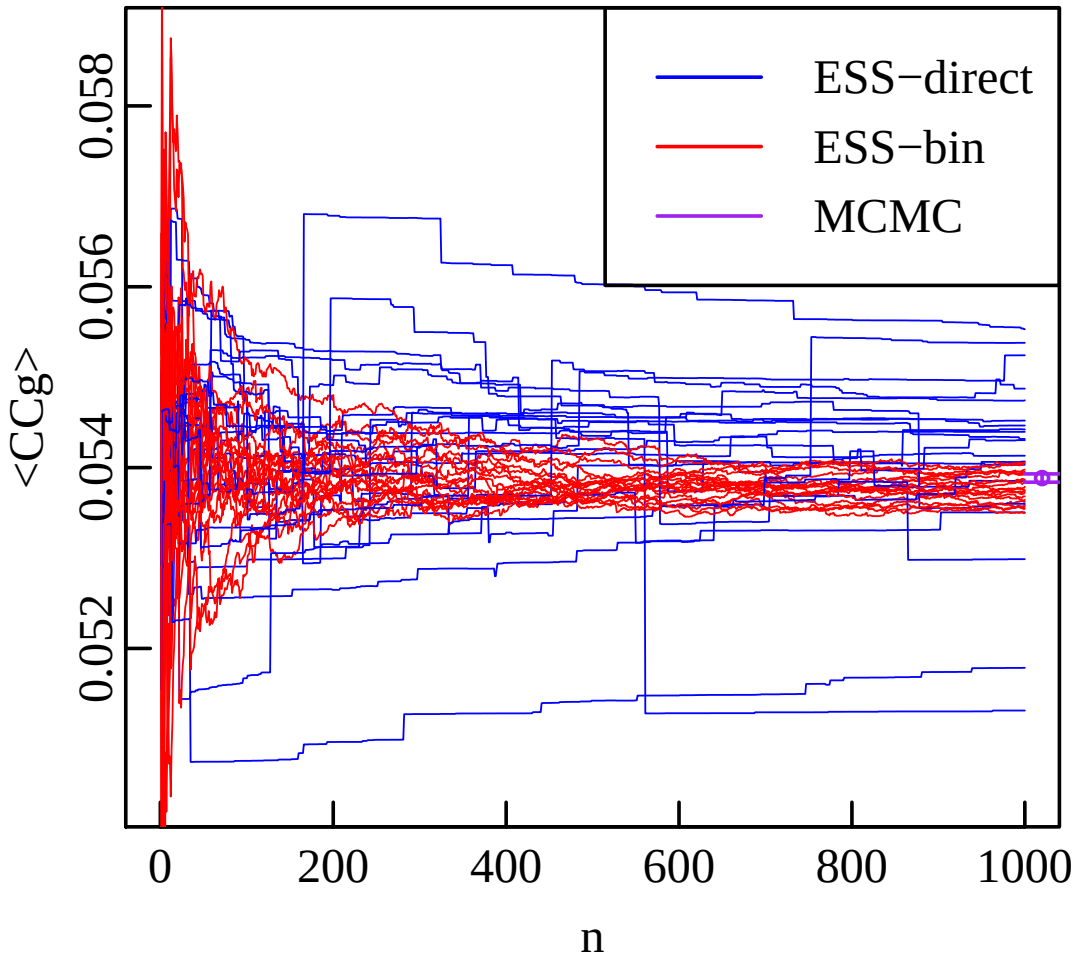


Figure 2.10: Running ensemble average of the CC_g for a prescribed random power-law distributed sequence. Results for twenty independent runs of $n = 1000$ samples are shown. Direct weighted averages are shown in blue and distribution estimation averages are shown in red. MCMC results are shown in purple at the right edge for comparison.

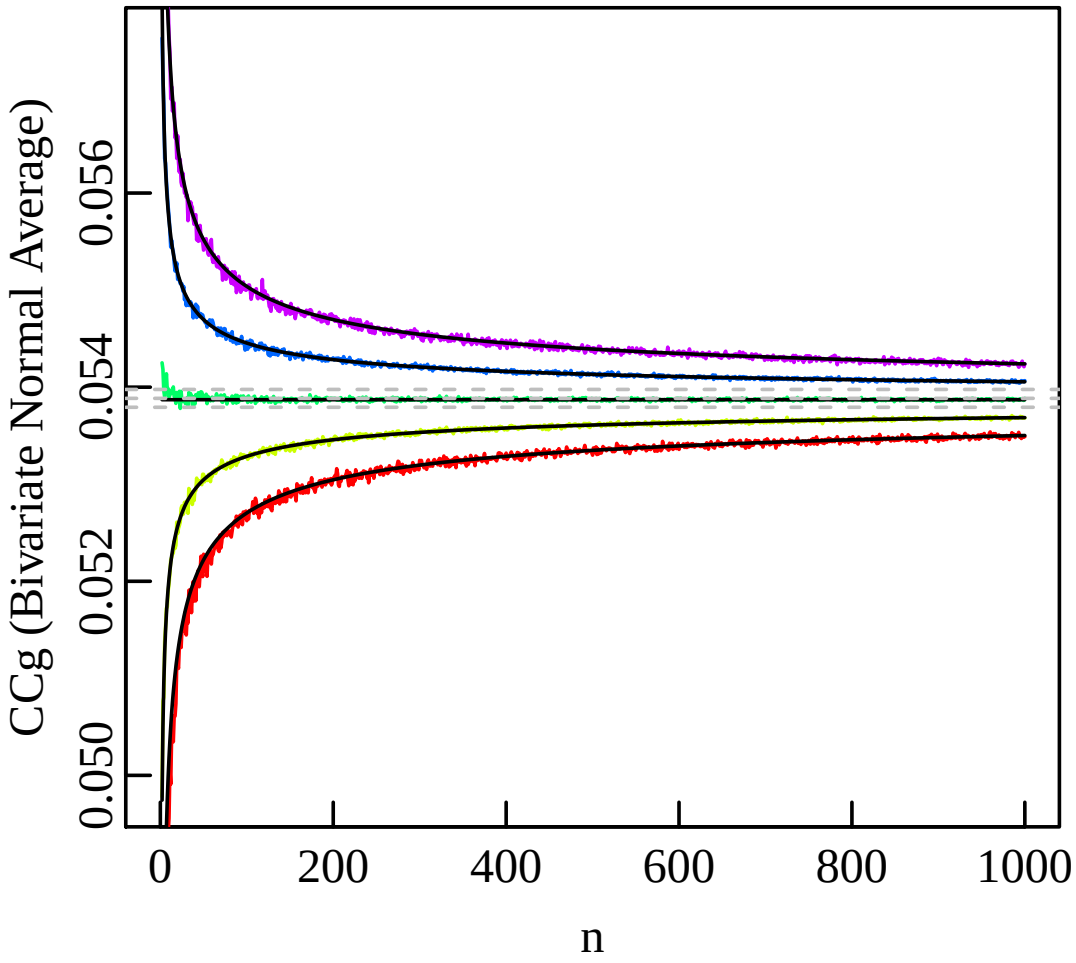


Figure 2.11: Quantile of ensemble average for different sample size using bivariate normal assumption. Black lines are theoretical results. Grey dash lines are MCMC result with 95% confidence interval. The quantiles are the same as quantiles for standard normal distribution with {-2, -1, 0, 1, 2} standard deviations, i.e. approximately {0.02, 0.16, 0.5, 0.84, and 0.98}.

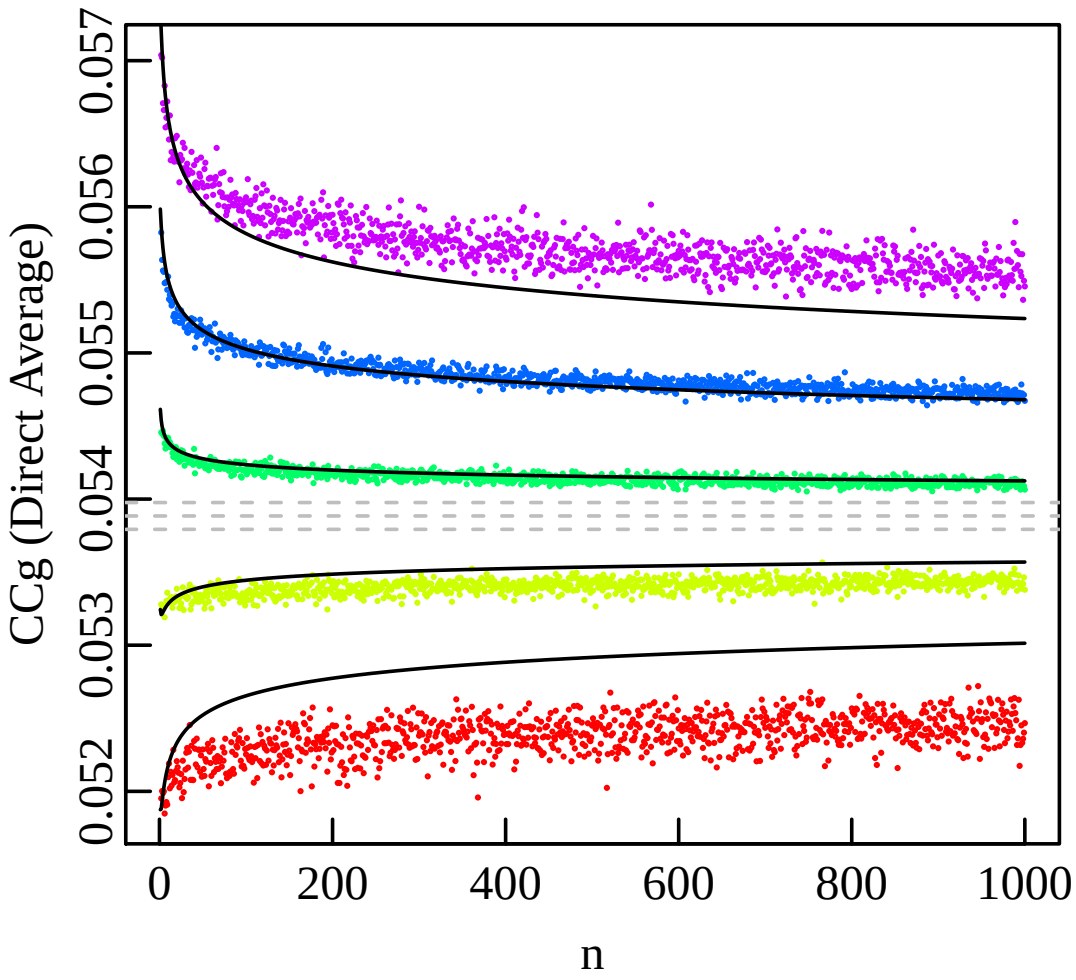


Figure 2.12: Quantile of ensemble average for different sample size using direct average when the underlying joint distribution is actually bivariate normal. Black lines are theoretical results. Grey dash lines are MCMC result with 95% confidence interval. The quantiles are the same as quantiles for standard normal distribution with {-2, -1, 0, 1, 2} standard deviations, i.e. approximately {0.02, 0.16, 0.5, 0.84, and 0.98}.

direct weighted running averages have many step-like jumps as sample size increases and converge slowly, while the bivariate normal JPD parameter estimation running averages converge quickly around the MCMC value. The mean value of the twenty runs after $n = 1000$ samples and the standard error of those twenty independent measurements is $5.407 \times 10^{-2} \pm 1.07 \times 10^{-3}$ for direct weighted averages and is $5.378 \times 10^{-2} \pm 1.7 \times 10^{-4}$ for averages using JPD parameter estimation. Using MCMC the result was $5.388 \times 10^{-2} \pm 5 \times 10^{-5}$.

2.5 An example with 10^6 nodes

In order to demonstrate the usefulness of our sampling methods in a practical real-world problem involving a large graph, consider the Youtube user friendship network (YoutubeNet) [96, 110]. Is the structure of this network somehow special, perhaps due to some self-organizing process? Or, is it random? YoutubeNet is an undirected graph with 1,134,890 nodes and 2,987,624 edges. As Figure 2.13 shows, it has a degree distribution that decays approximately as a power-law with exponent $\gamma \approx 2.3$. The global clustering coefficient the graph is $CC_g = 0.00622$. How random is the clustering of the YoutubeNet? This is partially answered by determining how unusual it is for a graph to have the CC_g value that YoutubeNet has within the ensemble of graphs that have the same degree sequence.

Using efficient stub sampling we constructed 1000 random graphs with the same degree sequence as the YoutubeNet. We used the degree sequence of YoutubeNet to generate random graphs using efficient stub sampling and link-swap method. Then we analyze the data of ESS using directed weighted average and bivariate normal assumption after verifying the (CCg, log-weight) JPD is really bivariate normal. Since

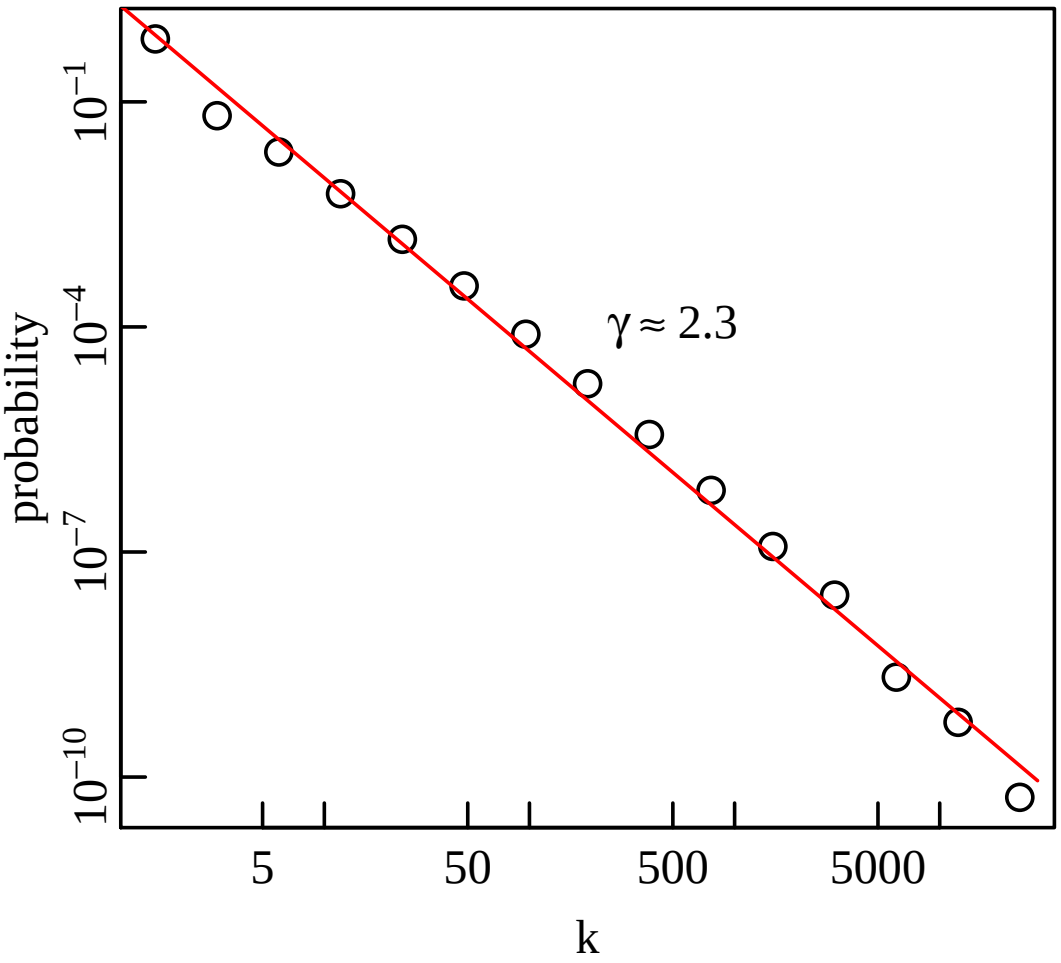


Figure 2.13: Degree distribution of the YoutubeNet. The red line is a decaying power-law function with exponent 2.3.

the standard deviation of log-weight for this 1000 random graphs is around 9, according to Eq. 2.23, $z \sim 9/\sqrt{1000} \approx 0.3$. Thus, with 1000 samples, we can expect estimation using bivariate normal assumption reasonably close to the actual result.

Figure 2.14 shows the CCg distribution. Note that the direct weighted distribution has sharp peaks. This may be caused by outliers with extremely large weights. On the other hand, the weighted distribution using bivariate normal assumption is closer to the distribution using link-swap.

Table 2.1: Statistics for different methods and corresponding z-score of YoutubeNet

method	mean CCg	sd of CCg	z-score
ESS-direct	$3.69(1) \times 10^{-3}$	$3(3) \times 10^{-6}$	791
ESS-normal	$3.702(6) \times 10^{-3}$	$2.04(4) \times 10^{-5}$	123
Link-swap	$3.708(1) \times 10^{-3}$	$2.01(4) \times 10^{-5}$	125

The number in the parentheses shows uncertainty of one standard error and applies to least significant digit. In order to be consistent, uncertainty for different methods are all calculated using bootstrapping method.

As shown in Table 2.1, all three methods give similar mean. This suggests that they all converge to the real value. Efficient stub sampling with bivariate normal assumption gives similar variance as link-swap. However, variance using direct weighted method is much smaller than the other two methods and has very large uncertainty. We believe this is because the largest weight dominates the distribution. Also note that the z-score is very large in all three cases, which means the structure of YoutubeNet cannot be explained by its degree sequence only.

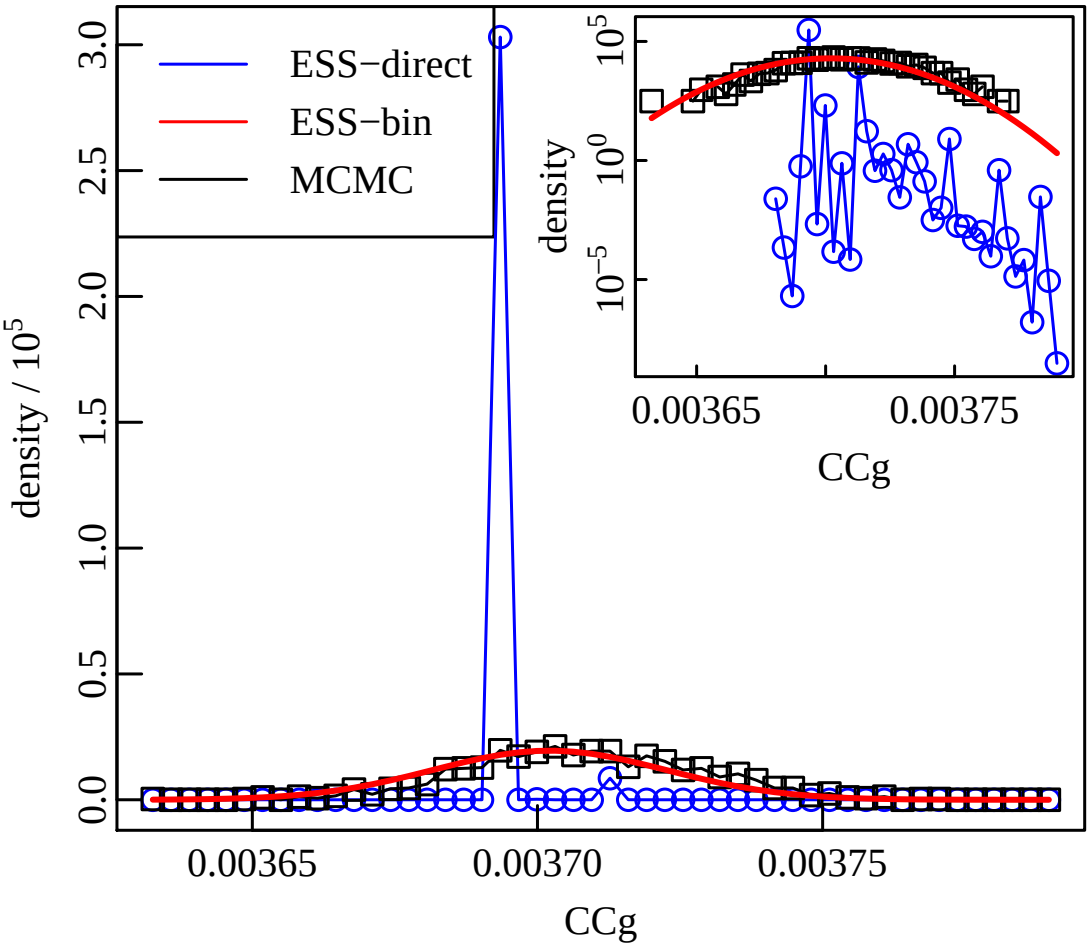


Figure 2.14: Weighted distribution of the global clustering coefficient $P_w(CC_g)$ for graphs with the degree sequence of the YoutubeNet. Graphs were sampled using efficient stub sampling, and then analyzed directly as a weighted sum (blue circles) and with distribution estimation (red line). The distribution was also calculated for link-swap, analyzing the results directly as an unweighted sum (black squares). The line connecting the blue circles and the one connecting the black squares are simply guides to the eye. The red line is a Gaussian function. Inset shows the same data, but plots density in log-scale.

2.6 Discussion

In this paper, we found that using efficient stub sampling in Eq. 2.6 with large hub first is an optimized way to assign probabilities to nodes in the allowed set for SIS [46] algorithm.

If we further assume the measured property and log-weight JPD to be bivariate normal, we can express the weighted average of this property in JPD parameters using Eq. 2.19 and its standard error using Eq. 2.22. For log-normal weight distribution, estimating distribution parameters is more stable than doing weighted average directly on weights with large fluctuation. Thus, by estimating JPD parameters, the sample size required to estimate the weighted average would grow linearly as system size grows, instead of exponentially when doing weighted average directly.

We also compared our method with MCMC. The fact that two independent methods converge to the same result indicates that both methods work, and the converged value should be close to the real value.

Some ways might improve the current method. One is to get a more accurate estimation of the number of possible graphs for a given residual degree sequence with star constraint. This may help reduce the variance of log-weight [63]. Another way is to run many simulations at the same time and use resampling at every hub-choosing step. This may help control the variance of log-weight, preventing it from growing too fast [109].

Also, it might be useful to use SIS and MCMC together. For example, we can use SIS with different order to choose hub and different probability to connect nodes in the Allowed Set to generate different initial graphs for MCMC [124]. Then we can

use MCMC to randomize those graphs. In this way, we can explore different parts of the configuration space. And we can be more confident if MCMC results from all different initial graphs converge.

In this paper we concentrate on degree sequence. But there is still a lot to explore for the graph sampling problem in general. For example, it might be interesting to sample random geometric graphs satisfying some constraint, or random graphs satisfying some global constraint like spectrum [123].

2.7 Acknowledgements

This work was supported by the NSF through grant DMR-1507371. Simulation in this project was run on the uHPC cluster managed by the University of Houston and acquired through NFS Award Number 1531814. We are grateful for the support of the Center for Advanced Computing and Data Science at the University of Houston for assistance with the calculations carried out in this work.

2.8 Derivation of Eq. 2.6

We want to solve the Lagrange multipliers $\{\beta_i\}$ in Eq. 1.20. Since these equations are difficult to solve analytically in general, we need to make some assumptions to simplify the equations and get some idea of the relation between d_i and β_i.

In the simplest case, assuming β_i has a sharp distribution centered at $\bar{\beta}$, then using $\bar{\beta}$ to represent all β_j where $j \neq i$, we have

$$d_i \approx \frac{N-1}{1+e^{-(\beta_i+\bar{\beta})}} \ , \tag{2.24}$$

thus,

$$\beta_i \approx \ln(\frac{d_i}{N-1-d_i}) - \bar{\beta} \ . \tag{2.25}$$

Then if we define $\bar{\beta}$ as the arithmetic mean of β_i, we have

$$\bar{\beta} = \frac{1}{N}\sum_i \beta_i \approx \frac{\sum_i \ln \frac{d_i}{N-1-d_i}}{N} - \frac{\sum_i \bar{\beta}}{N} = \frac{\sum_i \ln \frac{d_i}{N-1-d_i}}{N} - \bar{\beta} \ , \tag{2.26}$$

$$\bar{\beta} \approx \frac{\sum_i \ln \frac{d_i}{N-1-d_i}}{2N} = \frac{1}{2}\ln(\prod_i \frac{d_i}{N-1-d_i})^{1/N} \ , \tag{2.27}$$

$$e^{2\bar{\beta}} \approx (\prod_i \frac{d_i}{N-1-d_i})^{1/N} \ , \tag{2.28}$$

where the right hand side is geometric mean of $\frac{d_i}{N-1-d_i}$.

In sequential importance sampling, it's easier to update arithmetic mean degree $\bar{d} \equiv \sum_i d_i/N$ than to update geometric mean Eq. 2.28, so let's assume

$$e^{2\bar{\beta}} \approx \frac{\bar{d}}{N-1-\bar{d}} \ . \tag{2.29}$$

In ERGM, the probability of connecting a pair of nodes i and j is 1.18. Substituting Eq. 2.25 and Eq. 2.29 into Eq. 1.18, we have

$$p_{ij} \approx \Big(1 + \frac{\bar{d}(N-1-d_i)(N-1-d_j)}{(N-1-\bar{d})d_i d_j}\Big)^{-1} \ . \tag{2.30}$$

Now let's consider a few example cases:

- For regular graph, $\forall i$, $d_i = d$, thus, $\bar{d} = d$,

$$p_{ij} = \Big(1 + \frac{d(N-1-d)^2}{(N-1-d)d^2}\Big)^{-1} = \frac{d}{N-1} \ , \tag{2.31}$$

  which gives us the correct probability.

- For very small $d_i, d_j, \bar{d} \ll N$, $\frac{d}{N-1-d} \ll 1$ for all d_i, d_j and $\bar{d}$,

$$p_{ij} \approx \Big(\frac{\bar{d}(N-1-d_i)(N-1-d_j)}{(N-1-\bar{d})d_i d_j}\Big)^{-1} \approx \frac{d_i d_j}{(N-1)\bar{d}} \ , \tag{2.32}$$

  which is the same as Chung-Lu model [35, 36].

- For very large $d_i, d_j, \bar{d} \sim N$, $\frac{d}{N-1-d} \gg 1$ for all d_i, d_j and $\bar{d}$,

$$p_{ij} \approx 1, \ p_{ij} < 1 \ , \tag{2.33}$$

  which solves the Chung-Lu model's problem that connection probability for a pair of nodes can exceed 1 if the degrees of the nodes are too large.

Thus, Eq. 2.30 is a qualitatively reasonable probability of connecting a pair of nodes.

If we change the variables in Eq. 2.30 to residual degree so that it is compatible with sequential importance sampling, we can get Eq. 2.6.

Chapter 3

Comparison between hard and soft constraint methods

3.1 Introduction

Complex systems are often modeled as networks. Networks contain a set of connections (edges) linking a set of points (nodes), where degree measures the number of edges per node [114, 4, 113]. The number of edges which link to a node is known as its degree. The empirical study of networks is confounded by incomplete knowledge, which results from the network's: size, non-static nature, or even privacy concerns [97]. Perhaps only one metric of the system, typically the degree, can be easily measured. However, the way in which the network is connected remains unknown. In graph enumeration problem, it is known that the number of configurations for regular random graph with n nodes and k neighbors for each node is asymptotically $\Omega(n,k) = \frac{(nk)!}{(nk/2)!2^{nk/2}(k!)^n} \exp(-\frac{k^2-1}{4} - \frac{k^3}{12n} + O(k^2/n))$ if $k = o(n^{1/2})$

[107]. Using this formula, a reasonably small regular graph with $N = 30$ and $k = 3$ has 2×10^{44} configurations. Thus, in general it's not practical to generate all possible configurations, and sampling method must be used. Sampling possible configurations from this set can be accomplished using network modeling methods (null models) [114, 111, 46, 1, 159, 33, 88]. Methods including Sequential Importance Sampling (SIS) [46, 13], Degree-Preserving Link-Swap (link-swap) [46], Chung-Lu [1, 139, 159], and the Exponential Random Graph Model (ERGM) [114, 32] are widely used in graph sampling and fall into two main categories: soft-constraint methods (SC) and hard-constraint methods (HC). Both use known information about the system as guidelines (constraints) to generate graphs. HC meets these constraints with each graph generated [132], while SC meets constraints on average over an ensemble of graphs [45, 114, 1].

Despite the variance of individual graphs, SC are preferred for their speed and simplicity for theoretical treatment as HC can be slow and difficult to analyze due to their discrete nature [46, 159, 137]. HC's main strength is they produce only graphs which reflect possible configurations of the original system, while ensemble bias introduced by SC is not well understood. Known problems with SC include degeneracy in which the prescribed value is met on average, but values are not distributed around the mean, and instead separate into two or more clusters of micro-states none of which reflect the expected value [76, 62, 114]. A well-known example is the star triangle problem [114, 76]

Our research focuses on network sampling methods on systems with scale-free distributions. Scale-free network's degree distribution conforms to $P(k) \sim k^{-\gamma}$. Examples of such degree distribution include: some social networks, protein-protein interactions and disease transmission [4, 114, 139]. Several major scale-free systems

notably the Internet have $2 < \gamma < 3$ [116]. In this region, scale-free networks have several known graphical and structural constraints [45, 116, 77].

It was previously unclear what structural bias SC methods exhibit when used to generate scale-free networks in the range $2 < \gamma < 3$. Previous research indicates that many SC methods do not capture community structure well [139]. We observe a bias toward higher transitivity when utilizing SC for scale-free sequences. Further study of SC on regular graphs as the degree k approaches its maximum possible value $N-1$ hints that SC may not perform as expected on many other types of graphs.

3.2 Methods

The main SC method used, ERGM, allows for great flexibility in the application of constraints to the set of possible graphs [114, 150]. In the case where a prescribed degree sequence k_i is given, the constraints on the ensemble become,

$$\langle k_i \rangle = \sum_G P(G) k_i(G) \ . \tag{3.1}$$

Maximizing the entropy of the set of possible graphs minimizes the bias in the resulting ensemble. The derivation of this minimization is explained by Newman [114]. The derivation includes a set of Lagrange multipliers, β_i, for each constraint applied to the system. Utilizing a prescribed degree sequence k_i as a constraint, and restricting the model to simple, undirected networks with no self-links, gives a non-linear system of size N with the form,

$$\langle k_i \rangle = \sum_{j \neq i} \frac{1}{1 + e^{-(\beta_i + \beta_j)}} \ . \tag{3.2}$$

For a network whose largest degree is small than $O(N^{1/2})$,

$$\langle k_i \rangle = \sum_j e^{\beta_i} e^{\beta_j}, \ if: \ e^{-(\beta_i+\beta_j)} \gg 1 \ , \tag{3.3}$$

enabling the β values to be estimated using,

$$\beta_i = \ln(\frac{k_i}{\sqrt{\sum_j k_j}}) \ . \tag{3.4}$$

Here, the Chung-Lu model could be used as a simplification to avoid solving the non-linear system of equations involving β_i [114]. However,Chung-Lu is incapable of producing networks where $2 < \gamma < 3$ [22, 31] if we do not set an artificial cutoff for maximum degree [113, 1, 159].

For ERGM, using the degree sequence as the constraint, the exact solutions can be numerically evaluated; in many situations with more complex constraints these values must be estimated using maximum likelihood estimation using Markov chain Monte Carlo [79, 141, 73], although new methods are available that outperform Markov chain methods [29].

Figure 3.1 shows the approximations in Eq. 3.4 and the numerical solutions to Eq. 3.2. For large power-law exponents Eq. 3.4 can be used to estimate the Lagrange multipliers, however for $2 < \gamma < 3$ they must be numerically solved for the correct solution.

The HC method used in this paper is Markov chain Monte Carlo method (MCMC), i.e. degree-preserving link-swap method and sequential importance sampling (SIS) [46, 164]. Before we decided to choose those methods, we also tested the configuration model (CM) [114], which connects pairs of stubs, i.e. half-edges randomly. Theoretically, CM can sample independent graphs uniformly. However, in practice, CM is not

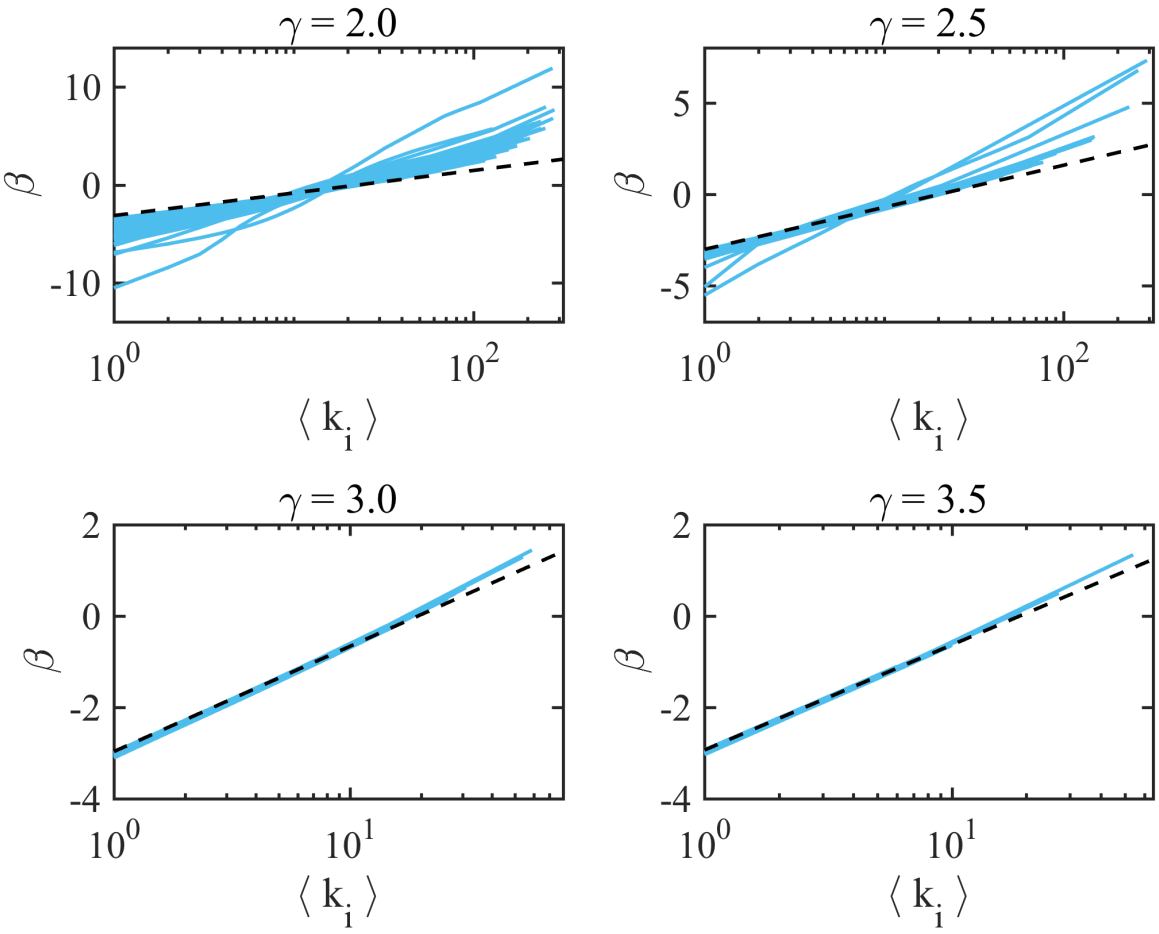


Figure 3.1: Approximations and numerical solutions of Lagrange multipliers β_i for exponential random graph model for scale-free degree sequences from various exponents γ. Network size is 316. The panels show the approximate values in Eq. 3.4 (black dashed) and the numerical solutions of Eq. 3.2 (blue) at different γ values.

efficient when generating scale-free simple graphs with $\gamma < 3$ due to self-loops and multi-edges.

In order to use MCMC [40] or link-swap in simulation, deciding how many steps we need to run for each sample is an important factor. That is, knowing the equilibrium time and correlation time is useful. However, this is a difficult theoretical question, and we were unable to find any useful theoretical upper bound of correlation time for our problem. So we make the step number to be the total degree for a degree sequence and tried different hard-constraint methods (CM, SIS, and link-swap with different initial configuration) on test sequences. We find that distributions from the different methods converge, which suggests that the step number we choose for link-swap is enough for our problem.

3.3 Results for scale-free networks

We examine the effectiveness of SC methods at reproducing the structural characteristics of simple undirected scale-free graphs by comparing the results to HC methods. Global clustering coefficient, also known as transitivity [116, 113, 114] , a feature of much interest in many graphs, is the measurement we examine. Transitivity is a measure of the connectivity of a graph, or a probability that if person X is friends with persons Y and Z that Y and Z are likewise friends. The transitivity of a graph may be measured as,

$$CC_g = \frac{number\ of\ closed\ triplets}{number\ of\ connected\ triplets} = \frac{3 \times (\ number\ of\ triangles)}{number\ of\ connected\ triples}\ . \qquad (3.5)$$

HC methods are used as a benchmark for transitivity because the graphs they produce accurately reflect possible configurations of the given constraints [46, 13].

We choose scale-free graphs $P(k) \propto k^{-\gamma}$ due to structural transitions at $\gamma = 3$ and graphical transition at $\gamma = 2$ [45]. Following [114],

$$CC_g = \frac{1}{N} \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3} \ . \tag{3.6}$$

Since for scale-free degree sequence,

$$1 \approx \int_1^N P(k)dk \Rightarrow P(k) \approx \frac{\gamma - 1}{1 - N^{1-\gamma}} k^{-\gamma} \ , \tag{3.7}$$

$$\langle k \rangle \approx \int_1^N kP(k)dk \approx \frac{\gamma - 1}{\gamma - 2} \frac{1 - N^{2-\gamma}}{1 - N^{1-\gamma}} \ , \tag{3.8}$$

$$\langle k^2 \rangle \approx \int_1^N k^2 P(k)dk \approx \frac{\gamma - 1}{\gamma - 3} \frac{1 - N^{3-\gamma}}{1 - N^{1-\gamma}} \ . \tag{3.9}$$

For graphs where $\gamma \geq 3$, $\frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3}$ is finite, $CC_g \sim O(N^{-1})$, thus, asymptotically no clustering is expected [114] and the network is connected in a locally tree-like structure. However, it was previously unclear what effect structural correlations, specifically those where $2 < \gamma < 3$, would have on CC_g.

We show that for two common graph types, SC methods are biased toward higher transitivity and spread than is expected by HC results.

Figure 3.2 shows the SC and HC transitivity cumulative density functions. We observe that for $\gamma = 2.0$ *or* 2.5, SC is biased toward much higher transitivity as much as 18% higher (an order of 10^{-2} larger) than the HC value. As $\gamma = 3.0$ *or* 3.5, this bias drops significantly to an order of 10^{-3} as the network becomes uncorrelated. Although the exact transition point for structural correlation occurs at $\gamma = 3.0$, this is only true in the limit of large network size [45]. For our results, a network size of

316 was too small to observe this transition cleanly, and we expect that in the limit of large N and γ that SC will agree more closely with HC.

The probability density functions for the scale-free results are shown in Figure 3.3 and further displays the differences between SC and HC transitivity distributions. For each γ the distributions between SC and HC differ, for larger values of γ transitivity values are small, and a large majority of values are 0. This trend, which is expected for scale-free networks will be more pronounced for larger γ and bigger networks. Although the mean values for HC and SC appear to converge to zero, the HC and SC distributions are still quite different in the shape of the tail for $\gamma = 3.5$, indicating that SC may never exactly match the distribution of HC. Of particular interest in both Figure 3.2 and Figure 3.3 is that for $\gamma = 2.0$ *and* 2.5 the SC distribution has a larger spread than HC, however the opposite is true where $\gamma = 3.0$ *and* 3.5. $\gamma = 3.0$ appears to be the transition point where the spread of HC and SC are equal. Why this occurs is unclear and should be the topic of further research.

As shown in Figure 3.4, when measuring the bias using z-score defined as $z = (\mu^{soft}_{CC_g} - \mu^{hard}_{CC_g})/\sigma^{hard}_{CC_g}$, the bias remains as system size increases and gets worse as $\gamma \to 2$.

In Figure 3.5, we also measured the z-score $z_{jk} = (p^{hard}_{jk} - p^{soft}_{jk})/\sigma^{soft}_{jk}$ of degree mixing matrix, which shows the probability of an edge connecting two nodes with degree d_i and d_j. As Figure 3.5 shows, the connecting patterns between very large degree and very small degree are quite different between HC and SC.

A qualitative explanation why SC overestimates CC_g is that the large number of nodes with small expected degree in scale-free sequences may have 0 actual degree.

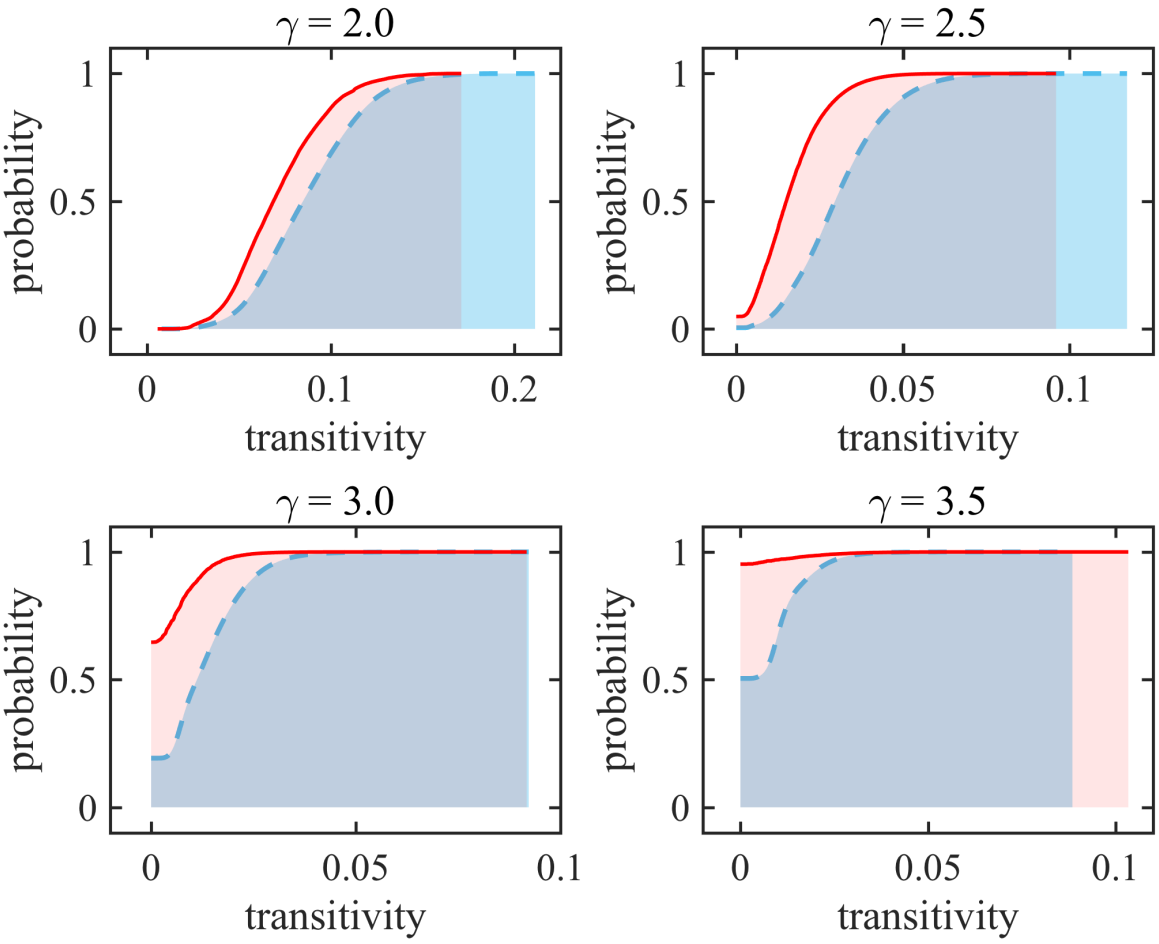


Figure 3.2: Cumulative distribution function (CDF) of transitivity for scale-free graphs with various exponents,γ and network size 316. The figures show transitivity distributions generated by SC (blue dashed) and predicted by HC (red) for different γ values.

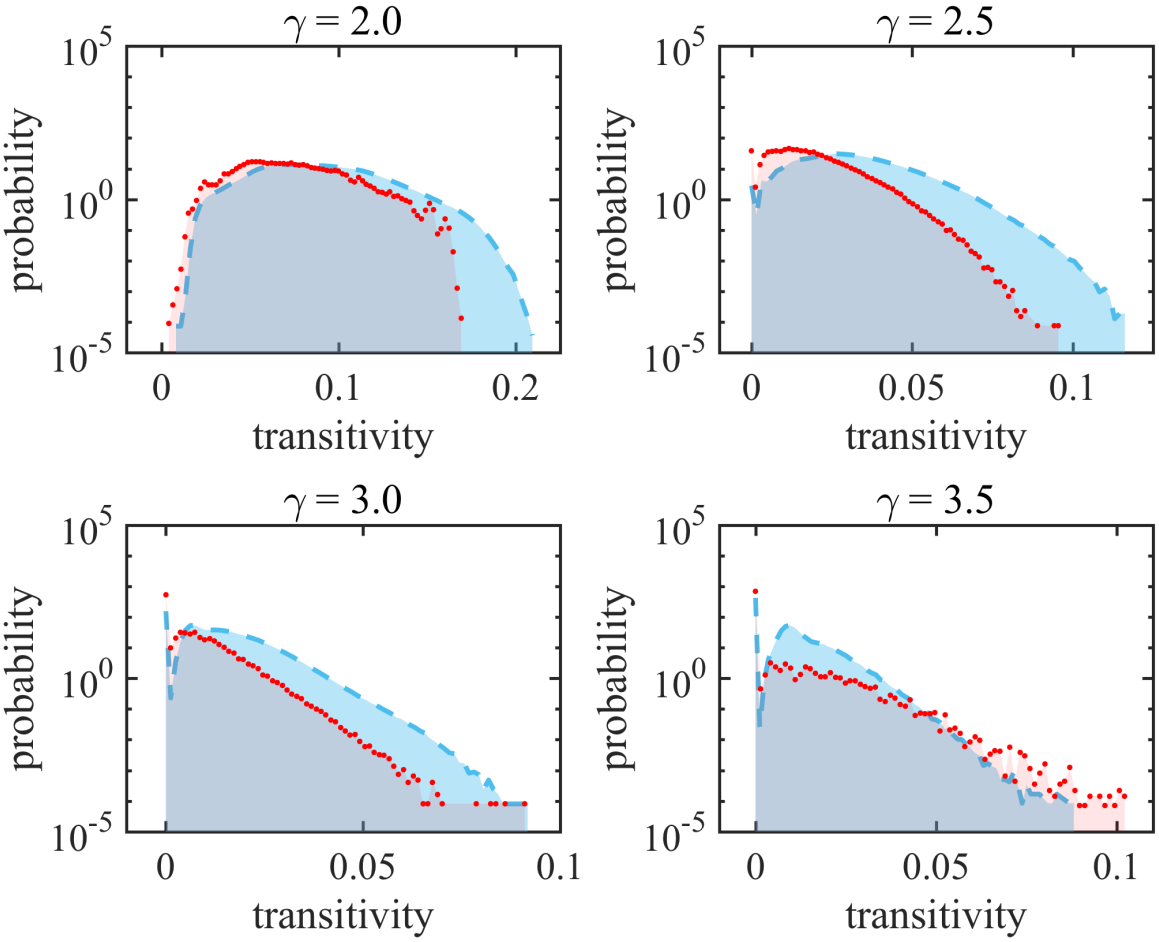


Figure 3.3: Probability density function (PDF) of transitivity for scale-free graphs with various exponents,γ and network size 316. The figures show transitivity distributions generated by SC (blue dashed) and HC (red points) for different γ values.

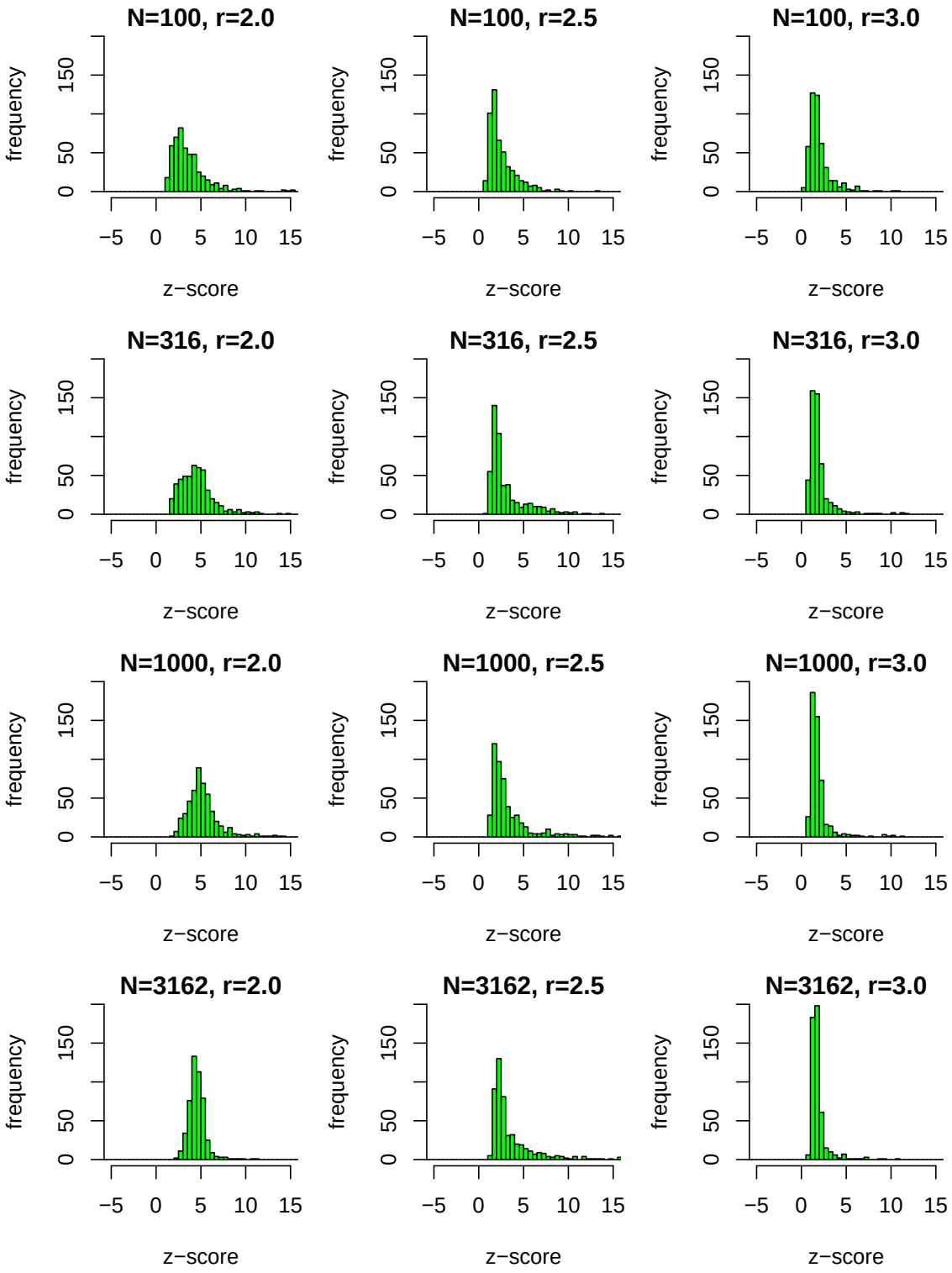


Figure 3.4: z-score distribution for different system size and exponents. Here $z = (\mu^{soft}_{CC_g} - \mu^{hard}_{CC_g})/\sigma^{hard}_{CC_g}$. In order to compare different parameters in same scale, distribution is truncated so that only bulk part is shown.

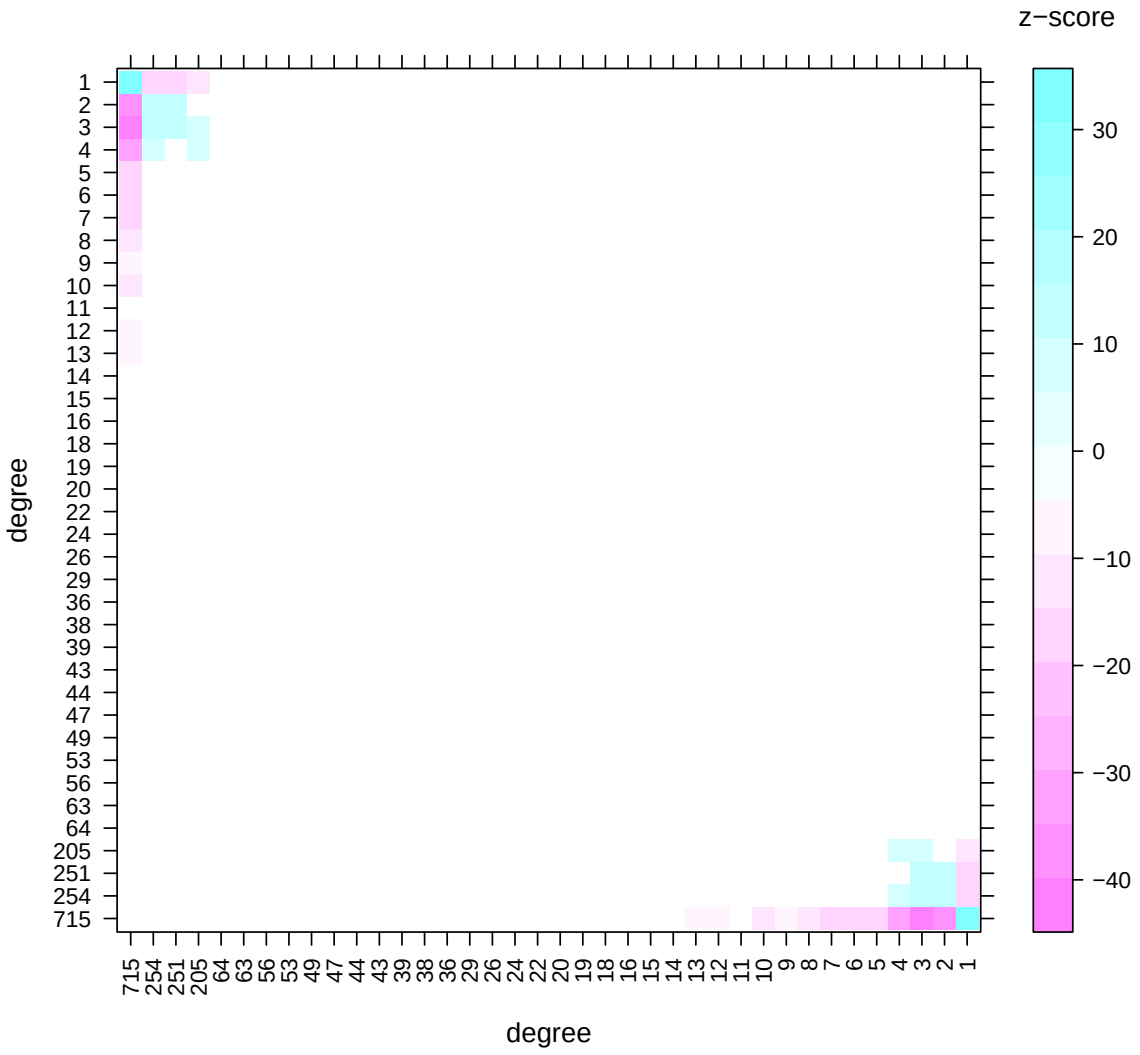
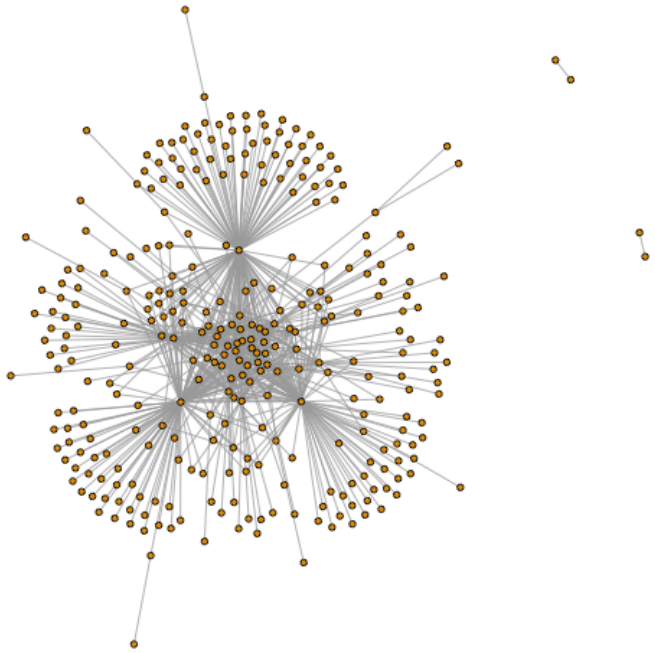
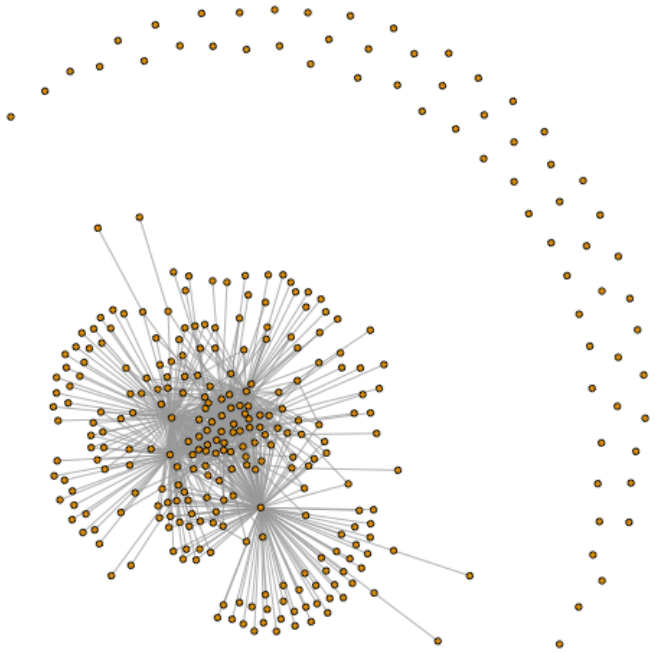


Figure 3.5: z-score of degree mixing matrix. $z_{jk} = (p_{jk}^{hard} - p_{jk}^{soft})/\sigma_{jk}^{soft}$ To reduce noise, only elements with $|z| > 6$ are shown. $N = 1000$, $\gamma = 2.0$

Because SC does not fix the prescribed constraint for every graph, a node with expected degree 1 could take on an actual degree of 0,1,2 or more. If the node has actual degree 0, then the edge must be distributed elsewhere. The remaining part of the graph must become denser, thus, increasing the clustering of the whole graph, as shown in Figure 3.6. To determine whether this explains the difference between SC and HC, we examine graphs whose prescribed degree is much larger than 1 and therefore avoid this "pitfall". As we show in regular random graphs, SC still predicts higher transitivity for graphs where the prescribed degree is much larger than 1.



(a) Rewire



(b) ERGM

Figure 3.6: Graphs with same expected degree sequence but generated with different methods. $N = 316$, $\gamma = 2.0$

3.4 Result for regular random graphs

Simple undirected k-regular graph requires all nodes have same prescribed degree d. This provides a model where the prescribed degree can be set to much larger than 1. We limit our study to networks of size 316. As Figure 3.7 shows, the distributions of transitivity for various prescribed degrees differ greatly for SC and HC. While SC has a large spread, HC has a much lower standard deviation, especially for larger prescribed degrees. The two methods also are not centered around the same value with SC predicting higher transitivity.

Figure 3.8 shows the mean and standard deviation of CC_g from SC (blue) and HC (red) methods, and their predicted values for a random graph of size 316 with prescribed degree d using Table 3.1.

Table 3.1: Comparison between hard and soft constraint methods on regular random graphs

	hard constraint	soft constraint
μ_Δ	$\frac{N(N-1)}{(N-2)^2}\frac{(d-1)^3}{6}$	$N(N-1)(N-2)p^3/6$
σ^2_Δ	$\frac{(d-1)^3(N-2-d)^3}{6(N-4)^3}*$	$\sim N^4p^5(1-p)/2$
μ_V	$Nd(d-1)/2$	$N(N-1)(N-2)p^2/2$
σ^2_V	0	$\sim 2N^4p^3(1-p)$
ρ	–	observation: 1 for large p
μ_{CC_g}	$\frac{N-1}{(N-2)^2}\frac{(d-1)^2}{d}$	p
$\sigma^2_{CC_g}$	$\frac{3^2\sigma^2_\Delta}{\mu^2_V}$	$\frac{\sigma^2_{x/y}}{\mu^2_{x/y}} = \frac{\sigma^2_x}{\mu^2_x} + \frac{\sigma^2_y}{\mu^2_y} - 2\rho\frac{\sigma_x}{\mu_x}\frac{\sigma_y}{\mu_y}$

Here we use a HC which produces identical results to link-swap, known as sequential importance sampling [46, 164] because for dense graphs link-swap becomes prohibitively slow. Using link-swap, we were able to reach $k = 273$, and results were identical to sequential importance sampling up to this point. The standard deviations

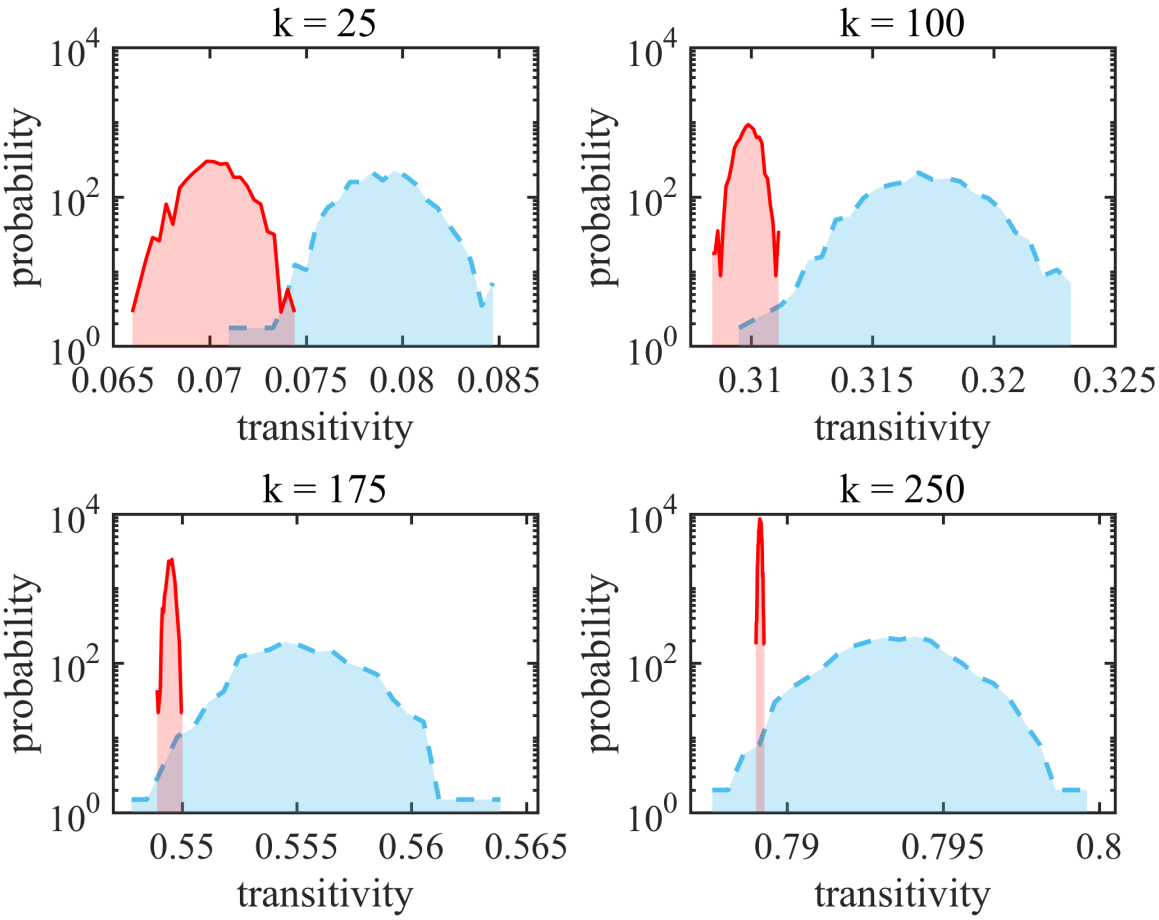


Figure 3.7: PDF for transitivity measured on regular random graphs with degree, k, and size 316. Figures display various k, and the resulting SC (blue dashed) and HC (red) distributions.

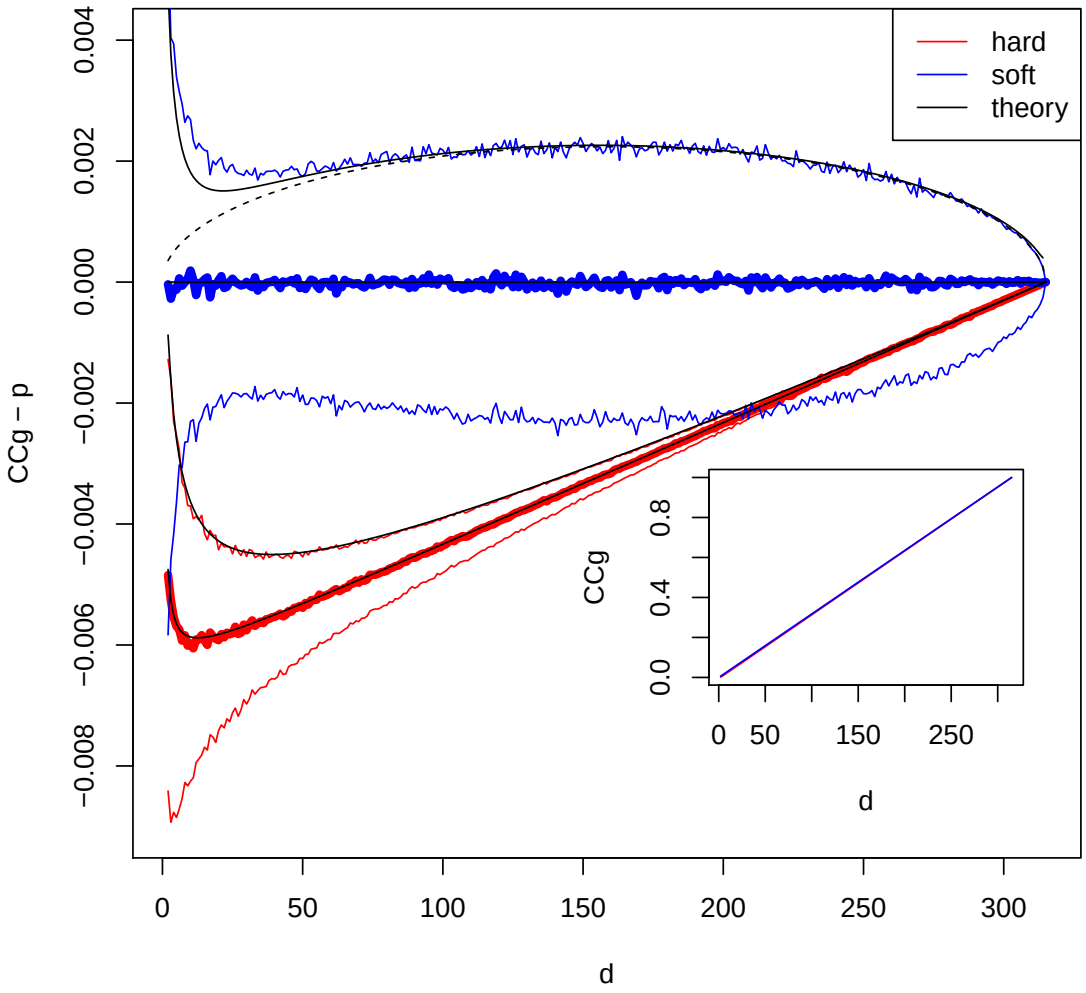


Figure 3.8: CC_g minus baseline $p = d/(N-1)$ for regular random graph with number of nodes $N = 316$ and degree d using hard (red) and soft (blue) constraint methods. Thick colored lines are the mean value. Thin colored lines show standard deviation of the CC_g distribution. Black continuous lines show theoretical prediction using Table 3.1. Black dashed line is from Eq. 3.25. The inset figure shows mean CC_g instead of $CC_g - p$.

for each k are also shown.

As shown in Figure 3.8, theoretical results in Table 3.1 agree with the data pretty well. In next section we'll explain the theory and argument behind Table 3.1.

3.5 Theoretical explanation of Table 3.1 for regular random graphs

For regular graph with number of nodes N and degree of each node $d \geq 2$, define $p = d/(N-1)$. For number of triangles N_Δ and number of connected triples N_V, we want to estimate their mean μ, variance σ^2 and correlation ρ, and use those distribution parameters to calculate the mean and variance of global clustering coefficient CC_g.

3.5.1 Theory for hard constraint methods

According to Corollary 2.19 in [23], the number of i-cycles in a random graph with node number N and degree d are asymptotically independent Poisson random variables with mean and variance $\lambda_i = (d-1)^i/(2i)$.

For $i = 3$, $\lambda_3 = (d-1)^3/6$. If $d = N-1$, $\lambda_3 = (N-2)^3/6$.

However, for complete graph $d = N-1$ the number of triangles $\mu_\Delta(N-1) = \binom{N}{3} = N(N-1)(N-2)/6 \neq (N-2)^3/6 = \lambda_3$.

In order to resolve this inconsistency, we can try to multiply λ_3 by a finite size correction factor $N(N-1)/(N-2)^2$.

Thus, the mean of number of triangles μ_Δ is

$$\mu_\Delta(d) = \frac{N(N-1)}{(N-2)^2}\frac{(d-1)^3}{6} \ . \tag{3.10}$$

To estimate the variance of number of triangles $\sigma^2_\Delta(d)$, we need to make use of the symmetry that $\sigma^2_\Delta(d) = \sigma^2_\Delta(N-1-d)$.

Based on the expression of λ_3, $\sigma^2_\Delta(d) \propto (d-3)^3$. Using the symmetry above, $\sigma^2_\Delta(d) \propto (d-1)^3(N-1-d-1)^3 = (d-1)^3(N-2-d)^3$. Define $\sigma^2_\Delta(d) = (d-1)^3(N-2-d)^3B$ where B is an unknown factor.

For small d, $\sigma^2_\Delta(d)$ and $\lambda_3(d)$ shouldn't be too different. Assuming for $d = 2$, $\sigma^2_\Delta(d) = \lambda_3(d)$, we have $(2-1)^3(N-2-2)^3B = (2-1)^3/6$. Thus, $B = \frac{1}{6(N-4)^3}$ and the variance of number of triangles:

$$\sigma^2_\Delta(d) = \frac{(d-1)^3(N-2-d)^3}{6(N-4)^3} \ , \tag{3.11}$$

where $d < N-1$.

For regular random graph with degree d the same for every node, the number of connected triples is $N\binom{d}{2} = Nd(d-1)/2$. Thus, the mean $\mu_V(d)$ and variance $\sigma^2_V(d)$ are

$$\mu_V(d) = Nd(d-1)/2 \ , \tag{3.12}$$

and

$$\sigma^2_V(d) = 0 \ . \tag{3.13}$$

Since $\sigma^2_V(d) = 0$, the correlation ρ between number of triangles N_Δ and number of connected triples N_V is undefined.

Since the global clustering coefficient is defined as $CC_g = 3N_\Delta/N_V$, the mean of

global clustering coefficient μ_{CC_g} is

$$\mu_{CC_g}(d) = \frac{3\mu_\Delta(d)}{\mu_V(d)} = 3\frac{N(N-1)}{(N-2)^2}\frac{(d-1)^3}{6}\frac{2}{Nd(d-1)} = \frac{N-1}{(N-2)^2}\frac{(d-1)^2}{d} \ . \quad (3.14)$$

Since $\sigma_V^2 = 0$, the variance of global clustering coefficient $\sigma^2_{CC_g}$ can be calculated using

$$\sigma^2_{CC_g}(d) = \frac{3^2\sigma^2_\Delta(d)}{\mu_V(d)^2} = \frac{9\times 4}{N^2d^2(d-1)^2}\frac{(d-1)^3(N-2-d)^3}{6(N-4)^3} = \frac{6(d-1)(N-2-d)^3}{d^2N^2(N-4)^3} \ . \quad (3.15)$$

3.5.2 Theory for soft constraint methods

Generating random regular graphs using soft constraint method is the same as generating random graphs using Erdos-Renyi model, where each pair of nodes is independently connected with identical probability $p = d/(N-1)$.

For any three nodes i, j, k randomly chosen from all $\binom{N}{3}$ combinations, in order for i, j, k to form a triangle, the elements of adjacency matrix $A_{ij} = 1, A_{jk} = 1, A_{ki} = 1$. Since they are connected independently, the probability for all three edges to be connected is $P(A_{ij} = 1, A_{jk} = 1, A_{ki} = 1) = p^3$. Thus, the expected value, or mean of number of triangles $\mu_\Delta(p)$ is

$$\mu_\Delta(p) = \binom{N}{3}p^3 = N(N-1)(N-2)p^3/6 \ . \quad (3.16)$$

From now on we'll skip the (p) notation and simply write μ_Δ, σ^2_Δ, etc. Just remember those are functions of p.

Following the method in [74, 136], define $Y_{ijk} = 1$ if i, j, k forms a triangle and $Y_{ijk} = 0$ if they don't. Then the number of triangles $N_\Delta = \sum_{i,j,k} Y_{ijk}$. The

variance $\sigma_\Delta^2 = E(N_\Delta^2) - \mu_\Delta^2$, where $E(N_\Delta^2)$ is the expected value of N_Δ^2. $N_\Delta^2 = \sum_{ijk,i'j'k'} Y_{ijk}Y_{i'j'k'}$.

To compute the expected value of $Y_{ijk}Y_{i'j'k'}$, we need to consider the probability $P(Y_{ijk} = 1, Y_{i'j'k'} = 1)$. But since ijk and $i'j'k'$ are not necessarily different, we need to consider different situations.

If ijk and $i'j'k'$ share 3 nodes, then $P(Y_{ijk} = 1, Y_{i'j'k'} = 1) = p^3$, and there is $\binom{N}{3}\binom{3}{3} = 1\binom{N}{3}$ case.

If ijk and $i'j'k'$ share 2 nodes, then $P(Y_{ijk} = 1, Y_{i'j'k'} = 1) = p^5$, and there are $\binom{N}{4}\binom{4}{2}\binom{2}{1}\binom{1}{1} = 12\binom{N}{4}$ cases.

If ijk and $i'j'k'$ share 1 node, then $P(Y_{ijk} = 1, Y_{i'j'k'} = 1) = p^6$, and there are $\binom{N}{5}\binom{5}{2}\binom{3}{2}\binom{1}{1} = 30\binom{N}{5}$ cases.

If ijk and $i'j'k'$ share 0 nodes, then $P(Y_{ijk} = 1, Y_{i'j'k'} = 1) = p^6$, and there are $\binom{N}{6}\binom{6}{3}\binom{3}{3} = 20\binom{N}{6}$ cases.

Thus, the variance of number of triangles

$$\sigma_\Delta^2 = E(N_\Delta^2) - \mu_\Delta^2 = \binom{N}{3}p^3 + 12\binom{N}{4}p^5 + 30\binom{N}{5}p^6 + 20\binom{N}{6}p^6 - \binom{N}{3}^2 p^6 \ . \quad (3.17)$$

Since

$$
\begin{aligned}
&30\binom{N}{5} + 20\binom{N}{6} - \binom{N}{3}^2 \\
=&N(N-1)(N-2)(N-3)(N-4)/4 \\
&+ N(N-1)(N-2)(N-3)(N-4)(N-5)/36 \\
&- N^2(N-1)^2(N-2)^2/36 \\
=&N^5/4 - (1+2+3+4)N^4/4 + O(N^3) \\
&+ N^6/36 - (1+2+3+4+5)N^5/36 \\
&+ (2+3+4+5+6+8+10+12+15+20)N^4/36 + O(N^3) \\
&- N^2(N^2-3N+2)^2/36 \\
=&N^5/4 - 5N^4/2 + N^6/36 - 15N^5/36 + 85N^4/36 + O(N^3) \\
&- N^2(N^4+9N^2-6N^3+4N^2+O(N))/36 \\
=&N^5/4 - 5N^4/2 + N^6/36 - 15N^5/36 + 85N^4/36 - N^6/36 + N^5/6 - 13N^4/36 + O(N^3) \\
=&-N^4/2 + O(N^3)\ ,
\end{aligned}
$$

for large N and not too small p, ignoring $O(N^3)$ terms, σ_Δ^2 can be simplified as

$$\sigma_\Delta^2 \approx N^4(p^5 - p^6)/2 = N^4 p^5(1-p)/2\ . \tag{3.18}$$

Using same method, we can calculate the number of connected triples N_V. Define L_{ijk} the number of connected triples formed by 3 randomly chosen nodes i, j, k. Then $N_V = \sum_{i,j,k} L_{ijk}$. L can take values 0, 1 and 3. The expected value $E(L_{ijk}) = \binom{3}{2}p^2(1-p) + 3\binom{3}{3}p^3 = 3p^2$.

Thus, the mean of number of connected triples:

$$\mu_V = \binom{N}{3} E(L) = N(N-1)(N-2)p^2/2\ . \tag{3.19}$$

To calculate the variance $\sigma_V^2 = E(N_V^2) - \mu_V^2$, where $N_V^2 = \sum_{ijk,i'j'k'} L_{ijk}L_{i'j'k'}$, we need to consider the probability $P(L_{ijk} = 1, L_{i'j'k'} = 1)$.

If ijk and $i'j'k'$ share 3 nodes i, j, k, then $E(L_{ijk}) = 3p^2$.

If ijk and $i'j'k'$ share 2 nodes i, j, then

$$\begin{aligned} E(L_{ijk}L_{ijk'}) =& P(A_{ij} = 1)E(L_{ijk}|A_{ij} = 1)^2 + P(A_{ij} = 0)E(L_{ijk}|A_{ij} = 0)^2 \\ =& p(2p(1-p) + 3p^2)^2 + (1-p)(p^2)^2 \\ =& p^3(p^2 + 4p + 4) + p^4 - p^5 \\ =& 5p^4 + 4p^3 \ . \end{aligned}$$

If ijk and $i'j'k'$ share 1 node i, then $E(L_{ijk}L_{ij'k'}) = E(L_{ijk})E(L_{ij'k'}) = 9p^4$.

If ijk and $i'j'k'$ share 0 nodes, then $E(L_{ijk}L_{i'j'k'}) = E(L_{ijk})E(L_{i'j'k'}) = 9p^4$.

Thus, the variance of number of connected triples:

$$\sigma_V^2 = E(N_V^2) - \mu_V^2 = \binom{N}{3}3p^2 + 12\binom{N}{4}(5p^4 + 4p^3) + (30\binom{N}{5} + 20\binom{N}{6} - \binom{N}{3}^2)9p^4 \ . \tag{3.20}$$

For large N and not too small p,

$$\sigma_V^2 \approx N^4(5p^4 + 4p^3 - 9p^4)/2 = 2N^4p^3(1-p) \ . \tag{3.21}$$

We observed that correlation ρ between number of triangles N_Δ and number of connected triples N_V approaches 1 for sufficiently large p.

Thus, the mean of global clustering coefficient

$$\mu_{CC_g} = \frac{3\mu_\Delta}{\mu_V} = \frac{3N(N-1)(N-2)p^3/6}{N(N-1)(N-2)p^2/2} = p \ , \tag{3.22}$$

and the variance of global clustering coefficient can be calculated using

$$\frac{\sigma_{x/y}^2}{\mu_{x/y}^2} = \frac{\sigma_x^2}{\mu_x^2} + \frac{\sigma_y^2}{\mu_y^2} - 2\rho\frac{\sigma_x}{\mu_x}\frac{\sigma_y}{\mu_y} \ . \tag{3.23}$$

For large N and not too small p, we can use the approximations $\mu_\Delta \approx N^3p^3/6$, $\sigma^2_\Delta \approx N^4p^5(1-p)/2$, $\mu_V \approx N^3p^2/2$, $\sigma^2_V \approx 2N^4p^3(1-p)$, and $\rho \approx 1$.

Then

$$\begin{aligned}\frac{\sigma^2_{\Delta/V}}{\mu^2_{\Delta/V}} =& \frac{\sigma^2_\Delta}{\mu^2_\Delta} + \frac{\sigma^2_V}{\mu^2_V} - 2\rho\frac{\sigma_\Delta}{\mu_\Delta}\frac{\sigma_V}{\mu_V} \\ \approx& \frac{N^4p^5(1-p)/2}{N^6p^6/36} + \frac{2N^4p^3(1-p)}{N^6p^4/4} - 2\frac{\sqrt{N^4p^5(1-p)/2 \times 2N^4p^3(1-p)}}{N^3p^3/6 \times N^3p^2/2} \\ =& \frac{18(1-p)}{N^2p} + \frac{8(1-p)}{N^2p} - \frac{24(1-p)}{N^2p} = \frac{2(1-p)}{N^2p} \ .\end{aligned}$$

Thus,

$$\sigma^2_{\Delta/V} \approx \frac{2(1-p)}{N^2p}\mu^2_{\Delta/V} = \frac{2(1-p)}{N^2p}(\frac{N^3p^3/6}{N^3p^2/2})^2 = \frac{2p(1-p)}{9N^2} \ , \tag{3.24}$$

$$\sigma^2_{CC_g} = \sigma^2_{3\Delta/V} = 3^2\sigma^2_{\Delta/V} \approx \frac{2p(1-p)}{N^2} \ . \tag{3.25}$$

This expression successfully captures the behaviour of $\sigma^2_{CC_g}$ at large p but fails at small p. So it is recommended to use the complete expression instead of the approximation.

3.6 Conclusion

In conclusion, we show that the SC methods are biased toward networks with higher transitivity on structurally constrained graphs when compared to HC methods. By comparing the transitivity of graphs generated from scale-free degree sequences using the exponential random graph model and link-swap; we observe divergence in the structural characteristics. For scale-free degree sequences this divergence occurs when

$2 < \gamma < 3$, a region where known structural constraints come into play [45]. Similar results on k-regular graphs show that SC systematically overestimates transitivity compared with HC. In conjunction with other known disadvantages of using soft-constrained methods notably, degeneracy [114, 76, 62], we urge extreme caution be exercised when using ERGM and other soft-constrained methods.

Chapter 4

Preferred degree extreme dynamics

4.1 Introduction

In previous sections, we discussed different methods to generate ensemble of graphs having a certain degree sequence. However, we didn't talk about why and how the group of nodes are connected in a certain way. This is exactly what we expected from a null model: making good use of known information while making as few assumptions as possible of the unknown.

On the other hand, real-world networks can be understood as emergent phenomena coming from some dynamics or rules followed by each node, like preferential attachment [4]. Generating graphs using some rules or dynamics means we are making assumptions not directly observed in the data. When doing this, we lose the ability to generate all possible graphs with uniform probability. But we might find interesting phenomena coming from the dynamics.

To generate graphs with prescribed degree sequence, we can use the "preferred

degree extreme dynamics" [168]:

- Each node i has a preferred degree κ_i.
- At every time step a random node i is chosen.
- If its actual degree $k_i > \kappa_i$, it randomly chooses one of its neighbors and cut the link.
- If $k_i < \kappa_i$, it randomly chooses one of the nodes not yet connected to it and connect.
- If $k_i = \kappa_i$, it does nothing.

This model is extreme in the sense that any node, when given a chance, will try to make its actual degree closer to its preferred degree, regardless of how different they are, and regardless of the effect of this move on other nodes. This is different from simulated annealing, where the whole system accepts a move with probability determined by a global cost function.

This extreme dynamics, in this general form, is difficult to study analytically. To simplify the problem, we introduce the following two models:

- eXtreme Introverts and Extroverts (XIE): two types of nodes, N_I introverts with $\kappa_I = 0$ and N_E extroverts with $\kappa_E = N_I + N_E - 1$
- Generalized Introverts and Extroverts (GIE): $0 \leq \kappa_I \leq \kappa_E \leq N_I + N_E - 1$

Previous study shows that XIE has mixed order phase transition [14], while GIE with non-integer preferred degree shows non-equilibrium effect [58].

4.2 XIE will reach equilibrium

In XIE dynamics, starting from any initial condition and letting the system evolve, finally the system will reach and stay in a subset of configurations, or states, where all introverts are disconnected from other introverts, and all extrovert are connected with other introverts. Only the cross-link between introverts and extroverts will change. Those states can be represented by their incidence matrices I.

Previous results show that XIE will reach equilibrium [14]. Below is a detailed proof for reader's convenience.

4.2.1 Ergodicity

Since for any element I_{ij} in the incidence matrix, if $I_{ij} = 1$, then there is a probability $1/N_I \neq 0$ that introvert i is chosen, and probability $1/k_i \neq 0$ that introvert i cut this link, making $I_{ij} = 0$. Similarly, if $I_{ij} = 0$, then there is a probability $1/N_E \neq 0$ that extrovert j is chosen, and probability $1/p_j \neq 0$ that extrovert j add this link, making $I_{ij} = 1$. Here $p_j = N_I + N_E - k_j$ means the number of nodes that are not connected with j. Thus, for any element in the incidence matrix, there is a non-zero probability for it to flip its value.

The configuration space of the incidence matrix is a hypercube with $N_I N_E$ dimensions. Since all directed edges of this hypercube have non-zero transition probability, the system is ergodic.

4.2.2 Detailed balance

4.2.2.1 General idea

In order to prove detailed balance, we can use Kolmogorov's criterion [86], which says that detailed balance is satisfied if and only if any loop of any length is reversible.

We first show that a combination of loops is reversible if all the components are reversible and have finite product of probabilities.

We know [14] that in XIE any loop with length 4 is reversible.

If we assume any loop with length n is reversible, then for any loop with length $n+2$, by considering the subspace of an active element in the state vector, the loop can be written as a combination of a loop with length n and a loop that is a combination of loops with length 4. Thus, any loop with length $n+2$ is also reversible.

Thus, any loop is reversible.

Thus, XIE satisfies detailed balance.

4.2.2.2 Definition

Consider a set of states $s_i \in S$. The transition probability from s_i to s_j is $p_{s_i s_j}$. A loop $s_1 s_2 \cdots s_n s_1$ is reversible if and only if $p_{s_1 s_2} p_{s_2 s_3} \cdots p_{s_n s_1} = p_{s_1 s_n} p_{s_n s_{n-1}} \cdots p_{s_2 s_1}$.

4.2.2.3 Combination of loops

Consider two reversible loops sharing a path. Loop A $s_1^C \cdots s_m^C s_1^A \cdots s_p^A s_1^C$ and loop B $s_1^C \cdots s_m^C s_1^B \cdots s_q^B s_1^C$ have common path $s_1^C \cdots s_m^C$.

Since the two loops are reversible,

$$p_{s_1^C s_2^C} \cdots p_{s_m^C s_1^A} p_{s_1^A s_2^A} \cdots p_{s_p^A s_1^C} = p_{s_1^C s_p^A} \cdots p_{s_2^A s_1^A} p_{s_1^A s_m^C} \cdots p_{s_2^C s_1^C} \ , \tag{4.1}$$

$$p_{s_1^C s_q^B} \cdots p_{s_2^B s_1^B} p_{s_1^B s_m^C} \cdots p_{s_2^C s_1^C} = p_{s_1^C s_2^C} \cdots p_{s_m^C s_1^B} p_{s_1^B s_2^B} \cdots p_{s_q^B s_1^C} \ . \tag{4.2}$$

Multiply Eq.4.1 with Eq.4.2 and divide both sides by the common factor

$$p_{s_m^C s_{m-1}^C} \cdots p_{s_2^C s_1^C} p_{s_1^C s_2^C} \cdots p_{s_{m-1}^C s_m^C} \ , \tag{4.3}$$

as long as it is not zero, we have

$$p_{s_m^C s_1^A} p_{s_1^A s_2^A} \cdots p_{s_p^A s_1^C} p_{s_1^C s_q^B} \cdots p_{s_2^B s_1^B} p_{s_1^B s_m^C} \tag{4.4}$$

$$= p_{s_1^C s_p^A} \cdots p_{s_2^A s_1^A} p_{s_1^A s_m^C} p_{s_m^C s_1^B} p_{s_1^B s_2^B} \cdots p_{s_q^B s_1^C} \ . \tag{4.5}$$

Thus, loop $s_m^C s_1^A \cdots s_p^A s_1^C s_q^B \cdots s_1^B s_m^C$ is reversible.

Thus, a combination of loops is reversible if all the individual loops are reversible, and there is no zero-transition-probability on any path.

4.2.2.4 XIE basic loops

We can describe an XIE state by its incidence matrix. Let's write the incidence matrix in one column and call it state vector. Since neighboring states are different by one element, the state vectors form a hypercube.

Consider loops with 2 moving elements and 4 states.

In XIE, whether the 2 elements are chosen from (1) same row, (2) same column or (3) different rows and columns from the incidence matrix, we can prove [14] from the transition probabilities that all those loops are reversible.

4.2.2.5 Induction

We know that any loop with length 4 is reversible.

Assume any loop with length n is reversible.

We want to prove that any loop with length $n+2$ is also reversible. (For hypercube, length of a loop must be even number.)

For an arbitrary loop $s_0 s_1 \cdots s_{n+1} s_0$ with length $n+2$, there must exist at least 1 element in the state vector that changes its value as we go around the loop. Without loss of generality, let's call this active element a and let $a = 0$ for state s_0 and $a = 1$ for s_1. Then we can go along the loop until the first time a changes from 1 back to 0. Let's say $a = 1$ for s_m and $a = 0$ for s_{m+1}. Thus, $a = 1$ for s_1, s_2, $\cdots$, s_m.

Now let's consider new states s'_1, s'_2, $\cdots$, s'_m, which are the same as s_1, s_2, $\cdots$, s_m except that $a = 0$. Obviously $s'_1 = s_0$ and $s'_m = s_{m+1}$.

Now consider two loops

$$A : s'_1 s'_2 \cdots s'_m s_{m+2} s_{m+3} \cdots s_{n+1} s'_1 \ , \tag{4.6}$$

$$B : s_1 s_2 \cdots s_m s'_m s'_{m-1} \cdots s'_1 s_1 \ . \tag{4.7}$$

Loop A has n states, thus, from the induction condition, loop A is reversible.

Consider loops $s_1 s_2 s'_2 s'_1$, $s_2 s_3 s'_3 s'_2$, $\cdots$, $s_{m-1} s_m s'_m s'_{m-1}$. Those loops are all of length 4, thus reversible.

$$p_{s_1 s_2} p_{s_2 s'_2} p_{s'_2 s'_1} p_{s'_1 s_1} = p_{s_1 s'_1} p_{s'_1 s'_2} p_{s'_2 s_2} p_{s_2 s_1} \ , \tag{4.8}$$

$$p_{s_2 s_3} p_{s_3 s'_3} p_{s'_3 s'_2} p_{s'_2 s_2} = p_{s_2 s'_2} p_{s'_2 s'_3} p_{s'_3 s_3} p_{s_3 s_2} \ , \tag{4.9}$$

...

$$p_{s_{m-1}s_m}p_{s_ms'_m}p_{s'_ms'_{m-1}}p_{s'_{m-1}s_{m-1}} = p_{s_{m-1}s'_{m-1}}p_{s'_{m-1}s'_m}p_{s'_ms_m}ps_ms_{m-1} \ . \tag{4.10}$$

Multiply both sides and divide by common factor,

$$p_{s_2s'_2}p_{s'_2s_2}p_{s_3s'_3}p_{s'_3s_3}\cdots p_{s_{m-1}s'_{m-1}}p_{s'_{m-1}s_{m-1}} \ , \tag{4.11}$$

we have,

$$p_{s_1s_2}p_{s_2s_3}\cdots p_{s_{m-1}s_m}p_{s_ms'_m}p_{s'_ms'_{m-1}}\cdots p_{s'_2s'_1}p_{s'_1s_1} \tag{4.12}$$

$$= p_{s'_1s'_2}p_{s'_2s'_3}\cdots p_{s'_{m-1}s'_m}p_{s'_ms_m}p_{s_ms_{m-1}}\cdots p_{s_2s_1}p_{s_1s'_1} \ . \tag{4.13}$$

Which is the probability product of loop B. Thus, loop B is reversible.

Now that both loop A and loop B are reversible, their combination, which is the original loop with length $n+2$, is also reversible.

This completes the inductive step.

Thus, for any loop with any length, the loop will be reversible. ■

4.3 XIE: degree distribution, cross-link distribution and correlation.

Previous studies use a mean field approach to estimate XIE degree distribution, which works fine when $N_I \neq N_E$. To get a more accurate result for the case $N_I = N_E$, we define Fixed cross-link XIE (fXIE) ensemble, which is a "cross-section" of XIE at a fixed number of cross-links. Using a self-consistent mean field approach, we can get the fXIE degree distribution. Then from fXIE degree distribution we can get XIE degree distribution, cross-link distribution, and correlation.

4.3.1 fXIE degree distribution is truncated Poisson

Using same argument as in [14], we can show that fXIE degree distribution is truncated Poisson.

For degree $k \geq 0$, the relationship between degree density $\rho_I(k)$ and transition probability $R(k \to k+1)$ satisfies the detailed balance:

$$\rho_I(k)R(k \to k+1) = \rho_I(k+1)R(k+1 \to k) \ . \tag{4.14}$$

Given X, N_I, N_E: In order for a certain introvert with degree $k+1$ to cut one link, it only needs to be chosen (with probability $\frac{1}{N_I+N_E}$):

$$R(k+1 \to k) = \frac{1}{N_I + N_E} \ . \tag{4.15}$$

For a certain introvert with degree k to add one link, one of the extroverts which haven't link to this introvert must be chosen (with probability $\frac{N_E-k}{N_I+N_E}$), and it must add one link with this introvert (with probability $< \frac{1}{p_E} >$)

$$R(k \to k+1) = \frac{N_E - k}{N_I + N_E} < \frac{1}{p_E} > \ , \tag{4.16}$$

where p_E is hole degree of extrovert and $< \frac{1}{p_E} >$ is the average of reciprocal of p_E.

Thus,

$$\frac{\rho_I(k+1)}{\rho_I(k)} = \frac{R(k \to k+1)}{R(k+1 \to k)} = (N_E - k) < \frac{1}{p_E} > \ . \tag{4.17}$$

Since

$$\frac{\rho_I(k)}{\rho_I(0)} = \frac{\rho_I(k)}{\rho_I(k-1)} \frac{\rho_I(k-1)}{\rho_I(k-2)} \cdots \frac{\rho_I(1)}{\rho_I(0)} \ , \tag{4.18}$$

we have

$$\begin{aligned}\frac{\rho_I(k)}{\rho_I(0)} &= (N_E-k+1)(N_E-k+2)\cdots(N_E) < \frac{1}{p_E} >^k \\ &= \frac{N_E!}{(N_E-k)!} < \frac{1}{p_E} >^k \ .\end{aligned} \tag{4.19}$$

Thus,

$$\rho_I(k) = \frac{1}{Z(X)}\frac{N_E!}{(N_E-k)!} < \frac{1}{p_E} >^k \ . \tag{4.20}$$

where

$$\begin{aligned}Z(X) &= \sum_{k=0}^{N_E} \frac{N_E!}{(N_E-k)!} < \frac{1}{p_E} >^k \\ &= N_E! < \frac{1}{p_E} >^{N_E} \sum_{k=0}^{N_E} \frac{1}{(N_E-k)!} < \frac{1}{p_E} >^{k-N_E} \\ &= N_E! < \frac{1}{p_E} >^{N_E} \sum_{k=0}^{N_E} \frac{1}{(N_E-k)!} < \frac{1}{p_E} >^{-(N_E-k)} \\ &= N_E! < \frac{1}{p_E} >^{N_E} \sum_{q=0}^{N_E} \frac{1}{q!} < \frac{1}{p_E} >^{-q} \ .\end{aligned} \tag{4.21}$$

Let us define $f = \frac{X}{N_I N_E}$ and assume p_E is a constant with value $p_E = N_I - \frac{X}{N_E} = N_I(1-\frac{X}{N_I N_E}) = N_I(1-f)$. If we further assume $N_I = N_E = L$, then $p_E = L - \frac{X}{L} = L(1-f)$.

Thus,

$$\begin{aligned}\rho_I(k) &= \frac{1}{Z(X)}\frac{L!}{(L-k)!}(\frac{1}{L-X/L})^k \\ &= \frac{1}{Z(X)}\frac{L!}{(L-k)!}(L-X/L)^{-k} \ ,\end{aligned} \tag{4.22}$$

and

$$\begin{aligned}Z(X) &= L!(\frac{1}{L(1-f)})^L \sum_{q=0}^{L} \frac{1}{q!}(\frac{1}{L(1-f)})^{-q} \\ &= \frac{L!}{[L(1-f)]^L} \sum_{q=0}^{L} \frac{[L(1-f)]^q}{q!} \ .\end{aligned} \tag{4.23}$$

This interpretation of the parameter in truncated Poisson distribution works fine as long as f is neither too large nor too small, so that the probability of introverts having 0 degree (happy introverts) and the probability of extroverts having 0 hole degree (happy extroverts) can be safely ignored. Otherwise, we should solve the parameter in truncated Poisson distribution numerically in a self-consistent way.

4.3.2 Correlation between cross links in fXIE

4.3.2.1 Truncated Poisson degree distribution

For fixed crosslink XIE model where $N_I = N_E = N$ and crosslink X, define $f = \frac{X}{N^2}$, the introvert degree distribution is a truncated Poisson distribution

$$\rho(k) = \frac{1}{Z}\frac{\lambda^{N-k}}{(N-k)!} , \tag{4.24}$$

where

$$Z = \sum_0^N \frac{\lambda^{N-k}}{(N-k)!} , \tag{4.25}$$

and

$$\frac{X}{N} = fN = \bar{k} = \sum_0^N k\rho(k) . \tag{4.26}$$

Mean of truncated Poisson distribution can be calculated as follows:

Change variable $p = N - k$, $\zeta(p) = \rho(k) = \frac{1}{Z}\frac{\lambda^p}{p!}$, then $Z = \sum_0^N \frac{\lambda^p}{p!}$ remains the same.

$$\bar{p} = \sum_0^N p\zeta(p) = \sum_0^N \frac{p}{Z}\frac{\lambda^p}{p!} = \sum_1^N \frac{\lambda}{Z}\frac{\lambda^{p-1}}{(p-1)!} = \frac{\lambda}{Z}\sum_0^{N-1}\frac{\lambda^p}{p!} = \frac{\lambda}{Z}(Z - \frac{\lambda^N}{N!}) . \tag{4.27}$$

Since $\rho(0) = \frac{1}{Z}\frac{\lambda^N}{N!}$, $\bar{p} = \lambda(1 - \frac{1}{Z}\frac{\lambda^N}{N!}) = \lambda(1 - \rho(0))$. Thus,

$$\lambda = \frac{\bar{p}}{1-\rho(0)} = \frac{N-\bar{k}}{1-\rho(0)} = \frac{N - \frac{X}{N}}{1-\rho(0)} . \tag{4.28}$$

So the parameter λ can be represented by happy introverts instead of happy extroverts.

Variance of truncated Poisson distribution can be calculated as follows.

First calculate

$$\begin{aligned}\overline{p(p-1)} &= \sum_0^N p(p-1)\zeta(p) = \sum_0^N \frac{p(p-1)}{Z}\frac{\lambda^p}{p!} = \sum_2^N \frac{\lambda^2}{Z}\frac{\lambda^{p-2}}{(p-2)!} \\ &= \lambda^2 \sum_0^{N-2} \frac{1}{Z}\frac{\lambda^p}{p!} = \lambda^2(1-\rho(0)-\rho(1)) \ .\end{aligned} \tag{4.29}$$

Since $\rho(1) = \frac{1}{Z}\frac{\lambda^{N-1}}{(N-1)!} = \frac{N}{\lambda}\rho(0)$, $\overline{p(p-1)} = \lambda^2(1-\rho(0)-\frac{N}{\lambda}\rho(0))$. Since $\bar{p} = \lambda(1-\rho(0))$, $\rho(0) = 1 - \frac{\bar{p}}{\lambda}$. Thus,

$$\overline{p(p-1)} = \lambda^2(1-(1-\frac{\bar{p}}{\lambda}) - \frac{N}{\lambda}(1-\frac{\bar{p}}{\lambda})) = \bar{p}\lambda - N\lambda + N\bar{p} \ . \tag{4.30}$$

Second moment

$$\overline{p^2} = \overline{p(p-1)} + \bar{p} = \bar{p}\lambda - N\lambda + N\bar{p} + \bar{p} \ . \tag{4.31}$$

Variance

$$var(p) = \overline{p^2} - \bar{p}^2 = \bar{p}\lambda - N\lambda + N\bar{p} + \bar{p} - \bar{p}^2 \ . \tag{4.32}$$

4.3.2.2 fXIE cross-link correlation

Given degree distribution, the correlation between two entries in the same row of the incidence matrix is

$$\chi^*_{EE} = \frac{\overline{k^2} - \bar{k}}{N(N-1)} - \frac{\bar{k}^2}{N^2} = \frac{\overline{k^2}}{N(N-1)} - \frac{f}{N-1} - f^2 \ , \tag{4.33}$$

$$\chi_{EE} = \frac{\chi^*_{EE}}{f(1-f)} \ . \tag{4.34}$$

Since $p = N - k$, $var(k) = var(p)$. Thus,

$$\overline{k^2} = var(k) + \bar{k}^2 = \bar{p}\lambda - N\lambda + N\bar{p} + \bar{p} - \bar{p}^2 + \bar{k}^2 \ . \tag{4.35}$$

Since $\bar{k} = Nf$, $\bar{p} = N - \bar{k} = N - Nf = N(1-f)$,

$$\overline{k^2} = N(1-f)\lambda - N\lambda + N^2(1-f) + N(1-f) - N^2(1-f)^2 + N^2 f^2 \ . \tag{4.36}$$

$$\begin{aligned}
\chi^*_{EE} &= \frac{\overline{k^2}}{N(N-1)} - \frac{f}{N-1} - f^2 \\
&= \frac{1}{N-1}(\frac{\overline{k^2}}{N} - f - (N-1)f^2) \\
&= \frac{1}{N-1}(\lambda - f\lambda - \lambda + N - Nf + 1 - f - N - Nf^2 + 2Nf + Nf^2 - f - Nf^2 + f^2) \\
&= \frac{1}{N-1}(f^2 - 2f + 1 + Nf - Nf^2 - \lambda f) \\
&= \frac{1}{N-1}((1-f)^2 + Nf(1-f) - \lambda f) \ .
\end{aligned} \tag{4.37}$$

Since $\lambda = \frac{\bar{p}}{1-\rho(0)} = \frac{N(1-f)}{1-\rho(0)}$,

$$\begin{aligned}
\chi^*_{EE} &= \frac{1}{N-1}((1-f)^2 + Nf(1-f)(1 - \frac{1}{1-\rho(0)})) \\
&= \frac{1}{N-1}((1-f)^2 - Nf(1-f)\frac{\rho(0)}{1-\rho(0)}) \ .
\end{aligned} \tag{4.38}$$

$$\chi_{EE} = \frac{\chi^*_{EE}}{f(1-f)} = \frac{1}{N-1}\frac{1-f}{f} - \frac{N}{N-1}\frac{\rho(0)}{1-\rho(0)} \ . \tag{4.39}$$

4.3.2.3 Asymptotic Behaviour when $f \to 1$, $\rho(0) \to 0$

Though both terms in Eq.4.39 are small and approaches 0, the second term vanishes earlier. For sufficiently large f, $\rho(0)$ is effectively 0, the second term can be ignored. Thus,

$$\chi_{EE} \to \frac{1}{N-1}\frac{1-f}{f} = \frac{1}{N-1}(\frac{1}{f} - 1) \ . \tag{4.40}$$

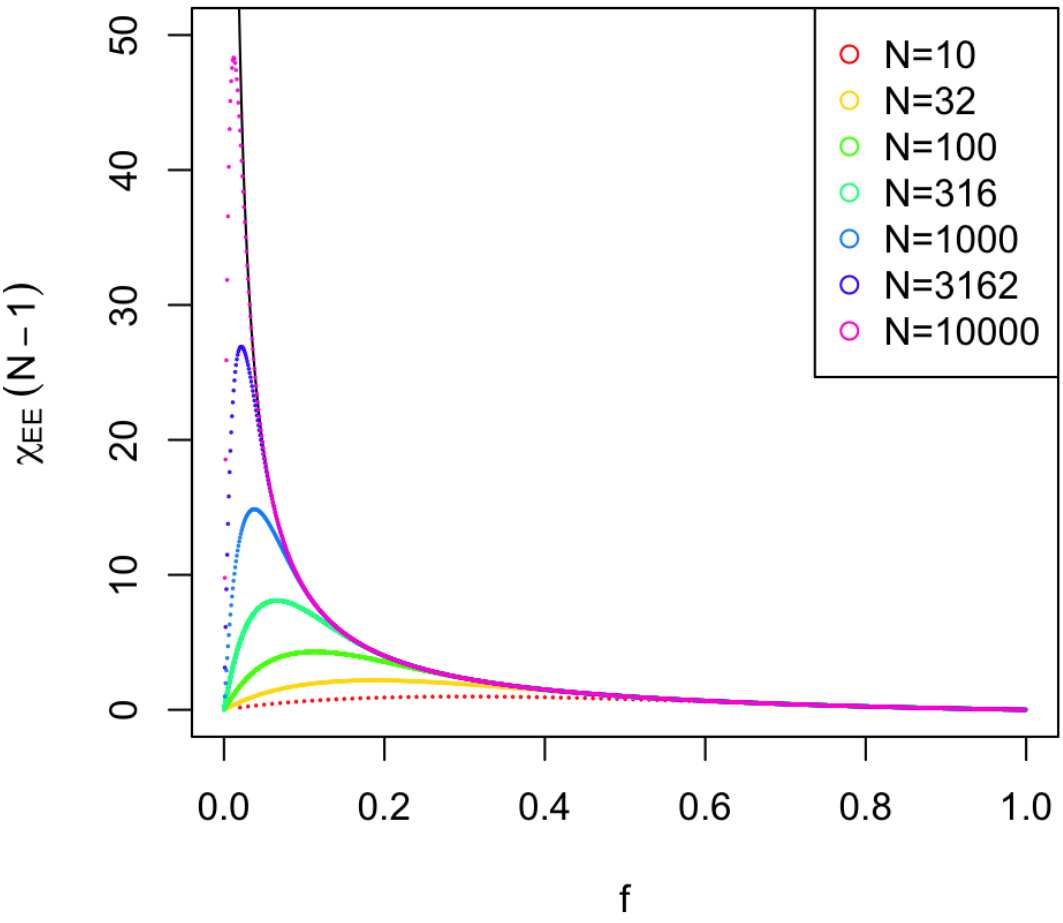


Figure 4.1: Scaling behaviour of χ_{EE} for different N when f is large. Black line is asymptotic result using Eq.4.40

.

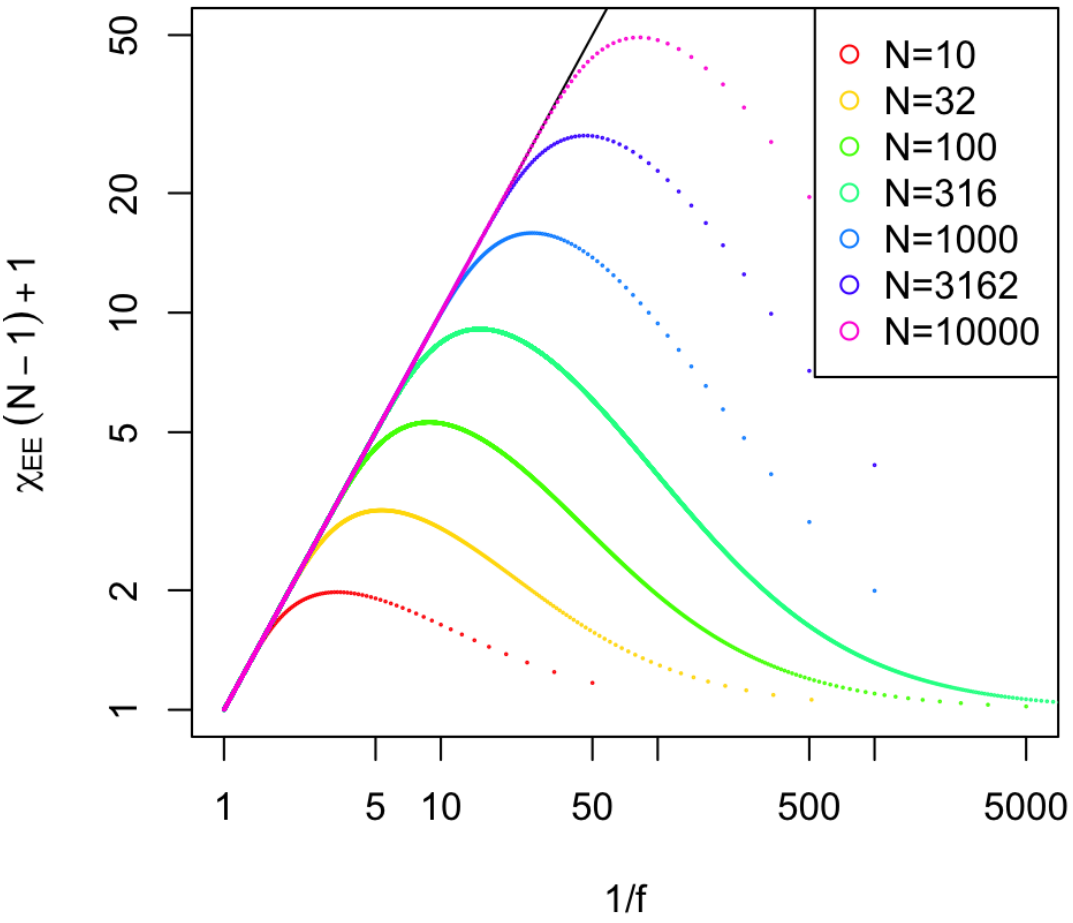


Figure 4.2: Scaling behaviour of χ_{EE} for different N when f is large. Black line is asymptotic result using Eq.4.40

.

4.3.2.4 Asymptotic Behaviour when $f \to 0$, $\rho(0) \to 1$

Now that both terms in Eq.4.39 go to infinity, the difference between the two is highly sensitive to the estimation of $\rho(0)$. Rather than using Eq.4.39, it's easier to make use of the sparsity of the incidence matrix and directly calculate χ_{EE} using the truncated Poisson distribution Eq.4.24.

Using Eq.4.24, we have

$$\rho(1) = \rho(0)\frac{N}{\lambda} \ , \ \rho(2) = \rho(0)\frac{N(N-1)}{\lambda^2} \ , \cdots \tag{4.41}$$

Define $\nu = \frac{1}{\lambda}$, we have

$$\rho(1) = \rho(0)N\nu \ , \ \rho(2) = \rho(0)N(N-1)\nu^2 \ , \ \cdots \tag{4.42}$$

Since $f \to 0$, the incidence matrix is sparse, the probability of $k \geq 3$ can be ignored. So the degree distribution can be written as

$$\rho(0) = \frac{1}{W} \ , \ \rho(1) = \frac{N\nu}{W} \ , \ \rho(2) = \frac{N(N-1)\nu^2}{W} \ , \tag{4.43}$$

where $W = 1 + N\nu + N(N-1)\nu^2$ is just the normalization factor.

The average degree

$$\bar{k} = fN = \sum k\rho(k) = \frac{0 \times 1 + 1 \times N\nu + 2 \times N(N-1)\nu^2}{1 + N\nu + N(N-1)\nu^2} \ . \tag{4.44}$$

Thus,

$$f = \frac{\nu + 2(N-1)\nu^2}{1 + N\nu + N(N-1)\nu^2} \ . \tag{4.45}$$

For $f \to 0$, $\rho(0) \gg \rho(1) \gg \rho(2)$. The right hand side of Eq.4.45 is just ν. Thus,

$$\nu \approx f \ , \tag{4.46}$$

and

$$W \approx 1 \ . \tag{4.47}$$

Note that in this case, $\rho(2) \approx N(N-1)f^2$, which is different from the binomial distribution $\binom{N}{2} f^2(1-f)^{N-2} \approx \frac{N(N-1)}{2} f^2$.

Since only degree $k = 2$ has non-zero contribution to $\overline{k(k-1)}$,

$$\chi^*_{EE} = \frac{\overline{k(k-1)}}{N(N-1)} - f^2 = \frac{2(2-1)\rho(2)}{N(N-1)} - f^2 \approx \frac{2N(N-1)f^2}{N(N-1)} - f^2 = f^2 \ . \tag{4.48}$$

Thus,

$$\chi_{EE} \approx \frac{f^2}{f(1-f)} \approx f \ . \tag{4.49}$$

4.3.2.5 Cross point

We can see where the two asymptotic curves Eq.4.49 and Eq.4.40 cross f^* by solving them together.

$$f^* = \frac{1}{N-1} \frac{1-f^*}{f^*} \ , \tag{4.50}$$

$$(N-1)(f^*)^2 + f^* - 1 = 0 \ , \tag{4.51}$$

$$f^* = \frac{-1 \pm \sqrt{1 - 4(N-1)(-1)}}{2(N-1)} = \frac{-1 \pm \sqrt{4N-3}}{2(N-1)} \ . \tag{4.52}$$

Since $f^* > 0$, $f^* = \frac{-1+\sqrt{4N-3}}{2(N-1)}$. For large $N \to \infty$, we have

$$f^* \to \frac{\sqrt{4N}}{2N} = \frac{1}{\sqrt{N}} \ . \tag{4.53}$$

Since for $N_I = N_E = N$, the bulk part (plateau) of $P(f)$ distribution is on the right side of $\frac{1}{\sqrt{N}}$ [170], for large N we can safely say that for a practical realization of fXIE, $\chi_{EE} \approx \frac{1}{N-1}(\frac{1}{f} - 1)$.

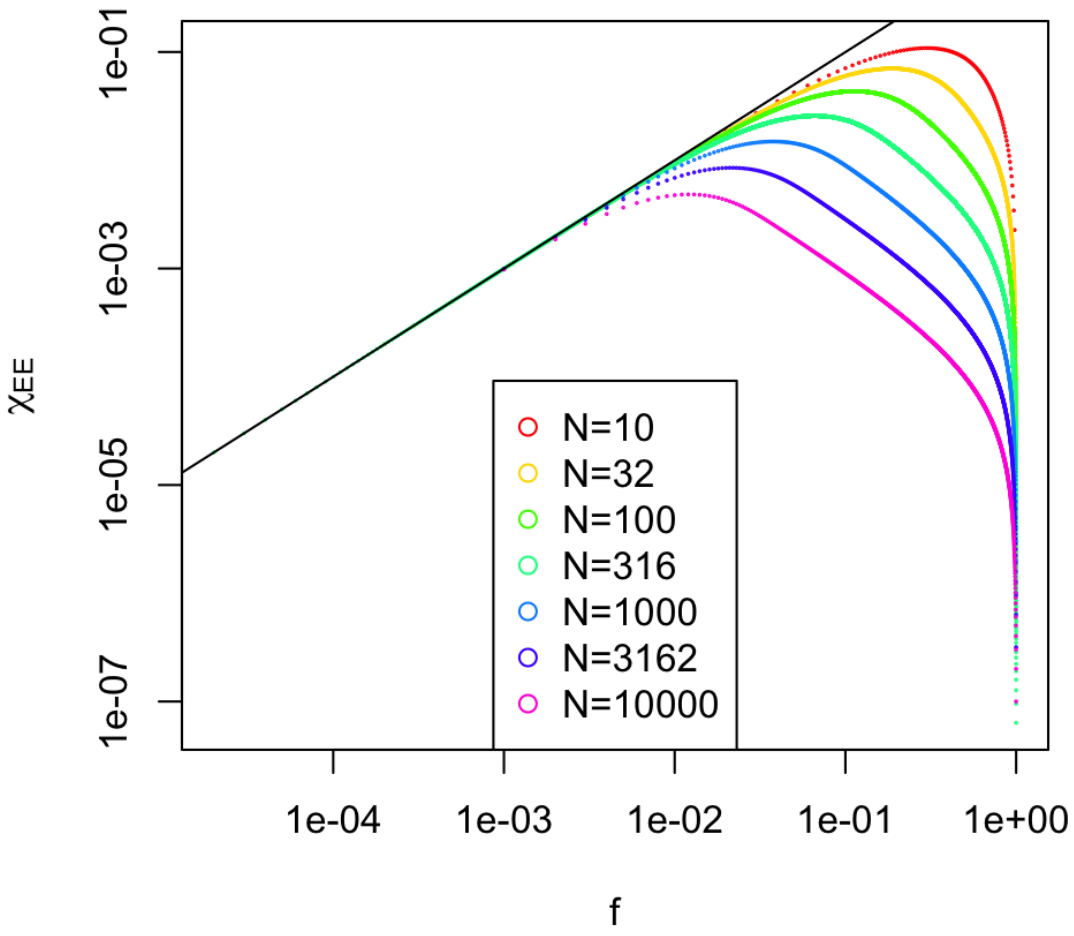


Figure 4.3: Behaviour of χ_{EE} for different N when f is small. Black line is asymptotic result in Eq.4.49

.

4.3.3 From fXIE to XIE

Once we have the fXIE degree distribution, using detailed balance

$$P(X)N_E(1-\zeta_X(0)) = P(X+1)N_I(1-\rho_{X+1}(0)) \ , \tag{4.54}$$

we can get the cross link distribution.

XIE degree distribution is just an average of fXIE degree distribution weighted by cross-link distribution

$$\rho(k) = \sum_X P(X)\rho_X(k) \ . \tag{4.55}$$

From degree distribution we can calculate cross-link correlation and other properties.

4.4 A special case of GIE similar to XIE

Inspired by Erdos-Gallai theorem [56], we can create a special case of GIE which is very similar to XIE:

$$N_E\kappa_E - N_E(N_E-1) > N_I\kappa_I \ . \tag{4.56}$$

The condition above means extroverts' preferred degrees are so large that even after connecting with all other extroverts and used up all the preferred degree of introverts together, they still want more connections. Thus, all extroverts are connected with each other, all introverts are disconnected from each other, all extroverts only want to add links with introverts, and all introverts only want to cut links with extroverts.

4.5 All loops of length 4 are reversible if preferred degrees are integers

4.5.1 Proof that all loops of length 4 are reversible

Now let's consider a case where the preferred degrees $\hat{d}_i$ are integers, which means it's possible for a node to be satisfied without initiating any move when it comes to its turn.

Following the extreme dynamics, the probabilities for an element of the adjacency matrix A_{ij} to flip its value are

$$P(A_{ij}:0\to 1)=\begin{cases}\frac{1}{N-1-d_i}, & d_i<\hat{d}_i, d_j\geq\hat{d}_j\\ \frac{1}{N-1-d_j}, & d_i\geq\hat{d}_i, d_j<\hat{d}_j\\ \frac{1}{N-1-d_i}+\frac{1}{N-1-d_j}, & d_i<\hat{d}_i, d_j<\hat{d}_j\\ 0, & d_i\geq\hat{d}_i, d_j\geq\hat{d}_j\ ,\end{cases} \tag{4.57}$$

and

$$P(A_{ij}:1\to 0)=\begin{cases}\frac{1}{d_i}, & d_i>\hat{d}_i, d_j\leq\hat{d}_j\\ \frac{1}{d_j}, & d_i\leq\hat{d}_i, d_j>\hat{d}_j\\ \frac{1}{d_i}+\frac{1}{d_j}, & d_i>\hat{d}_i, d_j>\hat{d}_j\\ 0, & d_i\leq\hat{d}_i, d_j\leq\hat{d}_j\ .\end{cases} \tag{4.58}$$

To get a loop of length 4, consider 2 elements of the adjacency matrix (A_{ij}, A_{kl}) take values $(0,0)\to(1,0)\to(1,1)\to(0,1)\to(0,0)$.

If i,j and k,l are all different, then their dynamics don't affect each other. Thus,

the product of forwarding probabilities

$$\prod(P \rightarrow) = P(A_{ij} : 0 \rightarrow 1)P(A_{kl} : 0 \rightarrow 1)P(A_{ij} : 1 \rightarrow 0)P(A_{kl} : 1 \rightarrow 0) \ , \quad (4.59)$$

and the product of backwarding probabilities

$$\prod(P \leftarrow) = P(A_{kl} : 0 \rightarrow 1)P(A_{ij} : 0 \rightarrow 1)P(A_{kl} : 1 \rightarrow 0)P(A_{ij} : 1 \rightarrow 0) \ . \quad (4.60)$$

Since $\prod(P \rightarrow) = \prod(P \leftarrow)$, this loop is reversible.

Now consider a case where i, j and k, l has 1 common index. Without loss of generality, let $k = i$. Thus, we have 2 elements of the adjacency matrix A_{ij} and A_{il} at the same row.

Given preferred degree $\hat{d}_i$, $\hat{d}_j$, $\hat{d}_l$, we can enumerate all the possible initial conditions of d_i, d_j, d_l in the following tables. Here $deg(i)$ represents the degree of node i at current state of the loop.

Table 4.1: $d_i \leq \hat{d}_i - 2$, $d_j \leq \hat{d}_j - 1$, $d_l \leq \hat{d}_l - 1$

$P \uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P \downarrow$
	(0,0)	d_i	d_j	d_l	
0					$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_j}$
	(1,0)	$d_i + 1$	$d_j + 1$	d_l	
0					$\frac{1}{N-1-(d_i+1)} + \frac{1}{N-1-d_l}$
	(1,1)	$d_i + 2$	$d_j + 1$	$d_l + 1$	
$\frac{1}{N-1-(d_i+1)} + \frac{1}{N-1-d_j}$					0
	(0,1)	$d_i + 1$	d_j	$d_l + 1$	
$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_l}$					0
	(0,0)	d_i	d_j	d_l	

$\prod(P \uparrow) = \prod(P \downarrow) = 0$, reversible.

Table 4.2: $d_i = \hat{d}_i - 1$, $d_j \leq \hat{d}_j - 1$, $d_l \leq \hat{d}_l - 1$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
0					$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_j}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2}$					$\frac{1}{N-1-d_l}$
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-d_j}$					$\frac{1}{d_i+2}$
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_l}$					0
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Table 4.3: $d_i \geq \hat{d}_i$, $d_j \leq \hat{d}_j - 1$, $d_l \leq \hat{d}_l - 1$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
$\frac{1}{d_i+1}$					$\frac{1}{N-1-d_j}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2}$					$\frac{1}{N-1-d_l}$
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-d_j}$					$\frac{1}{d_i+2}$
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_l}$					$\frac{1}{d_i+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = \frac{1}{(N-1-d_j)(N-1-d_l)(d_i+1)(d_i+2)}$, reversible.

Table 4.4: $d_i \leq \hat{d}_i - 2$, $d_j \leq \hat{d}_j - 1$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
0					$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_j}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_l+1}$					$\frac{1}{N-1-(d_i+1)}$
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-(d_i+1)} + \frac{1}{N-1-d_j}$					0
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_j}$					$\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Table 4.5: $d_i = \hat{d}_i - 1$, $d_j \leq \hat{d}_j - 1$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
0					$\frac{1}{N-1-d_i} + \frac{1}{N-1-d_j}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2} + \frac{1}{d_l+1}$					0
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-d_j}$					$\frac{1}{d_i+2}$
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_i}$					$\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Table 4.6: $d_i \geq \hat{d}_i$, $d_j \leq \hat{d}_j - 1$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
$\frac{1}{d_i+1}$					$\frac{1}{N-1-d_j}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2}+\frac{1}{d_l+1}$					0
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-d_j}$					$\frac{1}{d_i+2}$
	(0,1)	d_i+1	d_j	d_l+1	
0					$\frac{1}{d_i+1}+\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Table 4.7: $d_i \leq \hat{d}_i - 2$, $d_j \geq \hat{d}_j$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
$\frac{1}{d_j+1}$					$\frac{1}{N-1-d_i}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_l+1}$					$\frac{1}{N-1-(d_i+1)}$
	(1,1)	d_i+2	d_j+1	d_l+1	
$\frac{1}{N-1-(d_i+1)}$					$\frac{1}{d_i+1}$
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_i}$					$\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = \frac{1}{(N-1-d_i)(N-1-(d_i+1))(d_j+1)(d_l+1)}$, reversible.

Table 4.8: $d_i = \hat{d}_i - 1$, $d_j \geq \hat{d}_j$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
$\frac{1}{d_j+1}$					$\frac{1}{N-1-d_i}$
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2}+\frac{1}{d_l+1}$					0
	(1,1)	d_i+2	d_j+1	d_l+1	
0					$\frac{1}{d_i+2}+\frac{1}{d_j+1}$
	(0,1)	d_i+1	d_j	d_l+1	
$\frac{1}{N-1-d_i}$					$\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Table 4.9: $d_i \geq \hat{d}_i$, $d_j \geq \hat{d}_j$, $d_l \geq \hat{d}_l$

$P\uparrow$	(A_{ij}, A_{il})	$deg(i)$	$deg(j)$	$deg(l)$	$P\downarrow$
	(0,0)	d_i	d_j	d_l	
$\frac{1}{d_i+1}+\frac{1}{d_j+1}$					0
	(1,0)	d_i+1	d_j+1	d_l	
$\frac{1}{d_i+2}+\frac{1}{d_l+1}$					0
	(1,1)	d_i+2	d_j+1	d_l+1	
0					$\frac{1}{d_i+2}+\frac{1}{d_j+1}$
	(0,1)	d_i+1	d_j	d_l+1	
0					$\frac{1}{d_i+1}+\frac{1}{d_l+1}$
	(0,0)	d_i	d_j	d_l	

$\prod(P\uparrow) = \prod(P\downarrow) = 0$, reversible.

Thus, all loops of length 4 are reversible.

However, this doesn't necessarily mean that "all loops with any length are reversible". A counterexample is shown in Figure 4.4.

Also, note that depending on its actual degree, a node can sometimes behave as an introvert, disconnecting with an existing neighbor; and sometimes an extrovert, connecting to a new neighbor. An example preferred degree sequence is $\{2, 1, 1, 1\}$.

4.5.2 Configurations for preferred degree sequence can be not ergodic

The simplest degree sequence that has topologically different configurations is $\{2, 2, 2, 1, 1\}$. Let's call those nodes modest nodes.

We can add one extreme extrovert (XE) connecting all of the 5 modest nodes and let the 5 modest nodes to be happy about it. Thus, the preferred degree sequence becomes $\{5, 3, 3, 3, 2, 2\}$.

Then we can add one extreme introvert (XI) not connecting to any of those 6 nodes. Thus, the preferred sequence becomes $\{6, 3, 3, 3, 2, 2, 0\}$.

Now while running extreme dynamics, the graph will end up in one of the two topologically different configurations for the modest nodes. XE will connect to all the 5 modest nodes. XI will not connect to any of the 5 modest nodes. The 5 modest nodes are happy. The dynamics only happen between XI and XE. If XI and XE are connected and we pick up XI, it will cut its only link with XE. If XI and XE are not connected and we pick up XE, it can only connect to XI because all other links between XE and modest nodes are already there. Thus, once the 5 modest nodes are

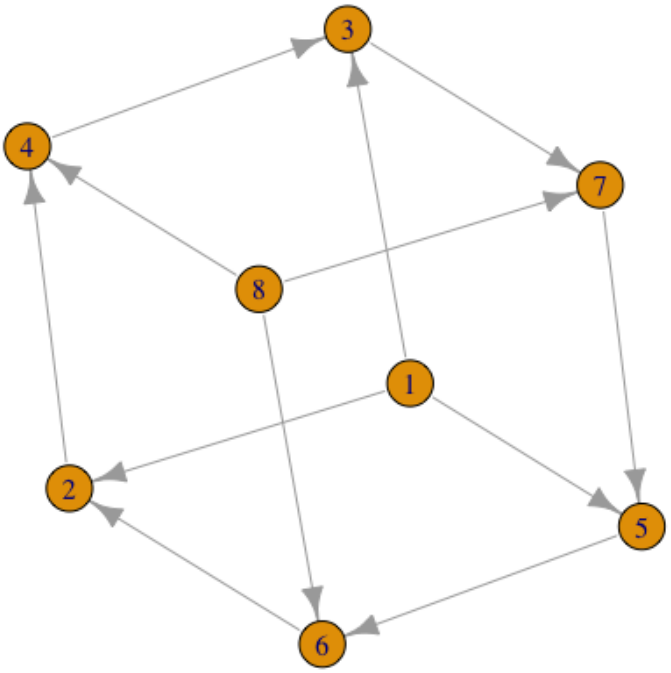
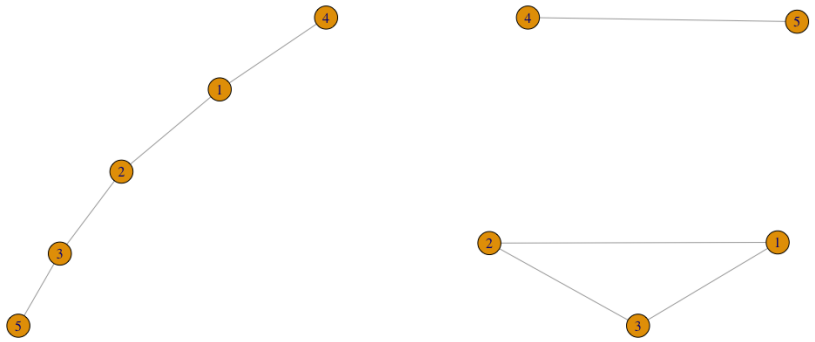


Figure 4.4: A counterexample showing that there exist irreversible loops even if all loops of length 4 are reversible. Each node represents a configuration. The transition probability is 1 along the directed edge and 0 otherwise. For any loop of length 4, the product of forward probabilities and backward probabilities are equally 0. Thus, any loop of length 4 is reversible. However, following the loop $2 \to 4 \to 3 \to 7 \to 5 \to 6 \to 2$, the product of forward probabilities is $1^6 = 1$ but the product of backward probabilities is $0^6 = 0$. Thus, this loop of length 6 is irreversible.

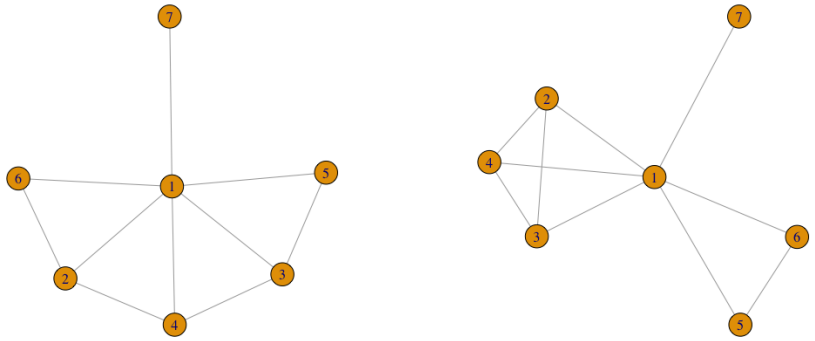
happy, they will not change the connections between them. However, since there are two topologically different configurations if we just consider the 5 modest nodes, those two configurations can not go from one to the other in stationary distribution. Thus, it is possible for extreme dynamics on preferred degree sequence to be not ergodic.

4.5.3 Conjecture that the system will reach equilibrium.

Though we haven't proven ergodicity and detailed balance in the general case, we conjecture that the system following extreme dynamics will finally reach equilibrium as long as the preferred degrees are integers and ergodicity is satisfied.



(a) (b)



(c) (d)

Figure 4.5: (a) and (b) are two topologically different configurations for degree sequence $\{2, 2, 2, 1, 1\}$. (c) and (d) are two disjoint basins for degree sequence $\{6, 3, 3, 3, 2, 2, 0\}$ where XI (Node 7) and XE (Node 1) are connected. In each basin of configurations, the connection between XI and XE can be changed. However, once reaches one of the two basins, the system cannot jump to the other basin.

4.6 Summary

In this section we use fXIE to help us understand XIE. We also conjecture that the system will reach equilibrium when the preferred degree are integers.

A graph from XIE in equilibrium belongs to a class of graphs called split graph, which is the topic of next chapter.

Chapter 5

Split graph and deeply nested network

5.1 Introduction

5.1.1 Split graph

A split graph [152, 59] is a undirected simple graph in which the vertices can be partitioned into a clique U, in which every node is connected with every other node, and an independent set W, in which there is no connection between any node. An example is shown in Figure 5.1.

If we order the nodes so that the first $|U|$ nodes are from the clique, and the rest $|W|$ nodes are from the independent set, then the adjacency matrix has the form

$$\begin{bmatrix} 1 - I_{|U|\times|U|} & X_{|U|\times|W|} \\ X^T_{|W|\times|U|} & 0_{|W|\times|W|} \end{bmatrix} \tag{5.1}$$

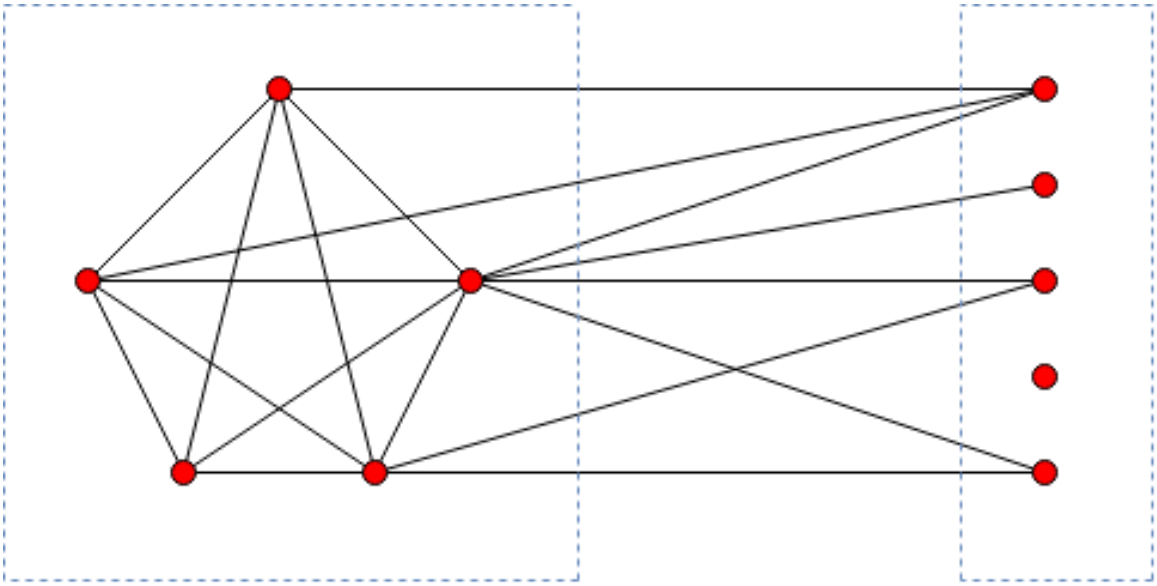


Figure 5.1: An example of split graph. The 5 nodes on the left form a clique, a.k.a. a complete graph, where every node is connected with every other node. The 5 nodes on the right form an independent set, where there's no connection between any pair of nodes.

Here I is the identity matrix. The upper left corner is $1 - I$ because for simple graph we excluded the self-loops.

In a split graph, the only edge pairs that can be used for degree-preserving link-swaps are those between U and W [55]. Thus, if we can decompose a graph into a series of split graphs, this decomposition can be used to improve the efficiency of MCMC.

5.1.2 Graph decomposition

We can try to decompose [55] a graph into a clique U, an independent set W and a set of nodes V, s.t. every node in V is connected to every node in U, but no node in V is connected to any node in W, as shown in Figure 5.2.

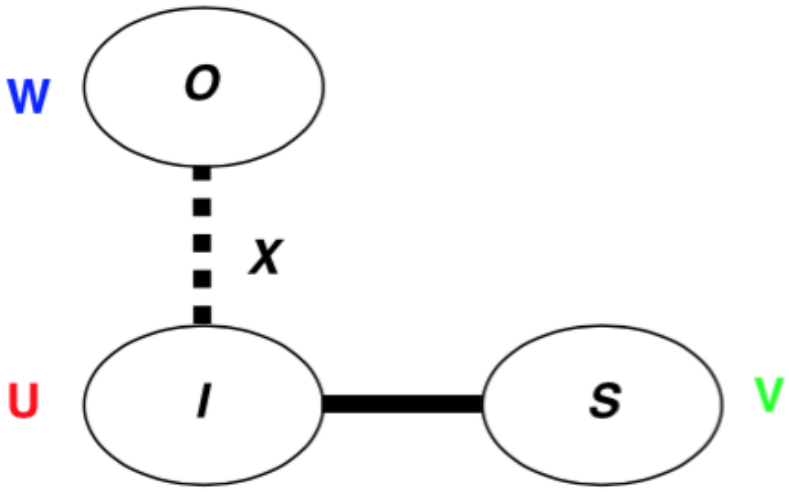


Figure 5.2: Graph decomposition. U is a clique, thus, $U \times U$ part of the adjacency matrix will be $1 - I$. W is an independent set, thus, the $W \times W$ part of the adjacency matrix will be 0. All nodes in V are connected with all nodes in U, thus, the $U \times V$ part of the adjacency matrix will be 1. No node in V is connected to any node in W, thus, the $V \times W$ part of the adjacency matrix will be 0. The connections between U and W are represented by matrix X. And the connections within V are represented by matrix S.

When the nodes are sorted in the order of $\{U, V, W\}$, the adjacency matrix can be written as

$$\begin{bmatrix} 1 - I_{|U|\times|U|} & 1_{|U|\times|V|} & X_{|U|\times|W|} \\ 1_{|V|\times|U|} & S_{|V|\times|V|} & 0_{|V|\times|W|} \\ X^T_{|W|\times|U|} & 0_{|W|\times|V|} & 0_{|W|\times|W|} \end{bmatrix} \tag{5.2}$$

While a graph may be decomposed in different ways, we want to find a way that extract the most information out of the degree sequence and leave least degree of freedom. Thus, we have canonical decomposition.

5.1.3 Canonical decomposition

Theorem 1. *Theorem 2* in [151]

1. *The graph G with non-increasing degree n-sequence is decomposable iff $\exists p, q$ non-negative integers s.t.*

$$0 < p + q < n \ , \quad \sum_{i=1}^{p} d_i = p(n - q - 1) + \sum_{i=n-q+1}^{n} d_i \ . \tag{5.3}$$

2. *Call a pair (p, q) satisfying condition (5.3) good. To every good pair $(p; q)$ we can associate the decomposition $(\langle U, W\rangle; E) \circ H = G$ where $(d_1, \ldots, d_p); (d_{p+1}, \ldots, d_{n-q})$ and $(d_{n-q+1}, \ldots, d_n)$ are the degree sequences in $U, V(H)$ and W respectively. Moreover, every such decomposition is associated with some good pair.*

3. *Let p_0 be the minimum first component of the good pairs. Let $q_0 = |\{i : d_i < p\}|$ if $p_0 \neq 0$ and $q_0 = 1$ otherwise. Then $(\langle U, W\rangle; E)$ is indecomposable if and only if the associated good pair is (p_0, q_0).*

Theorem 2. *Corollary 3.4* in [151]

- *Every graph G can be uniquely decomposed (up to isomorphism) into the form*

$$G = (\langle U_1, W_1\rangle; E_1) \circ \cdots \circ (\langle U_l, W_l\rangle; E_l) \circ G_0 \ , \tag{5.4}$$

  where each split graph and the non-split simple graph G_0 (if it exists) are indecomposable. The composition operation is associative but not commutative.

5.1.4 Graph composition

Given a split graph $(\langle U, W\rangle; E)$ and another graph G, we can compose a new graph $(\langle U, W\rangle; E) \circ G$ by adding connections between every node in G and every node in U.

5.1.5 Deeply nested network

We can try to perform canonical decomposition on any graph G. If G can be decomposed into a long series of split graphs, then we can roughly say that this graph is deeply nested. Figure 5.3 shows a constructed deeply nested network. The building block is a split graph $\langle u, w \rangle$ with degree sequence $\{4, 2; 1, 1, 1, 1\}$. The number of iterations $N = 10$.

If the decomposed components have the fast-mixing property when doing MCMC link-swap, then the original graph is also fast-mixing [55].

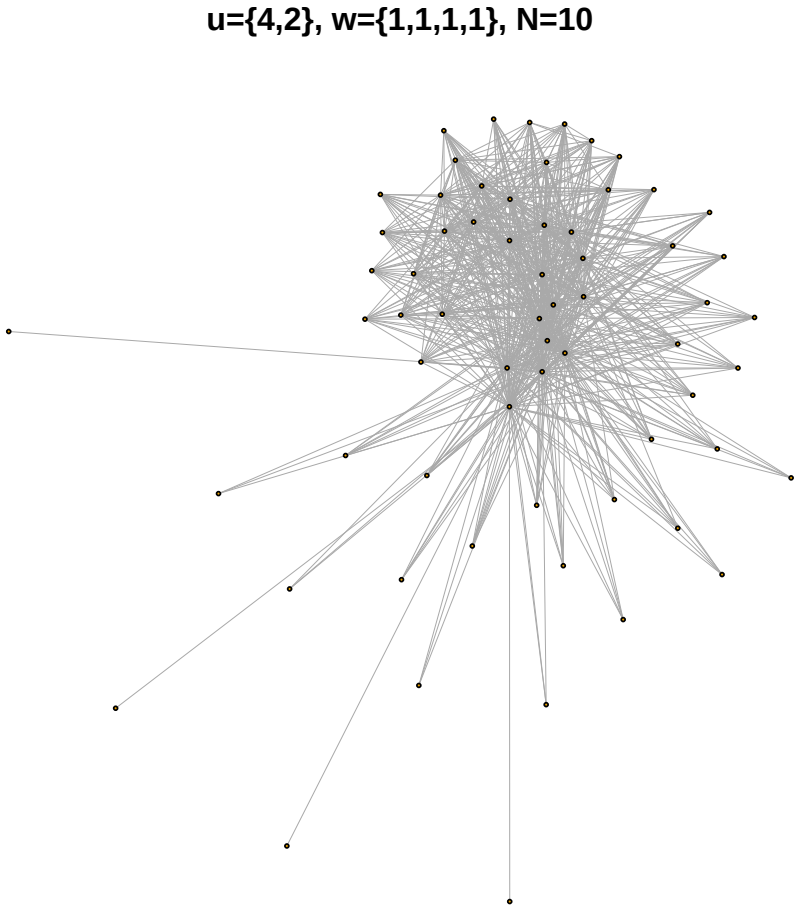


Figure 5.3: An example of deeply nested network. The building block is a split graph $\langle u, w \rangle$ with degree sequence $\{4, 2; 1, 1, 1, 1\}$. The number of iterations $N = 10$.

5.2 Nodes with the same degree separate together in canonical decomposition

Previous results [151] have shown that in order to perform canonical decomposition, all we need is the degree sequence. Here we proved that nodes with the same degree would be put into the same decomposition component.

Theorem 3. *All nodes in a degree class separate together in a canonical split graph decomposition, except when the separating split graphs have an empty U or W set.*

Proof. A) Split graph $\{u, w\}$ with $|u| > 0$, $|w| > 0$ and $d_{|u|} = |u| - 1$ is decomposable. proof: Consider a split graph $\{u, w\}$ containing clique u and independent set w, $|u| + |w| = n$, assuming $|u| > 0$, $|w| > 0$, $d_{|u|} = |u| - 1 = n - |w| - 1$. Let's use Tyshkevich's theorem 2.1. Since the graph is already a split graph, if $p = |u|$, $q = |w|$, then $0 < p + q = n$, $S_p = p(n - q - 1) + S_q$, so the first condition is violated but the second condition holds. Let's try $p' = |u| - 1$ and $q' = |w|$: $p' = p - 1$, $q' = q$, $0 < p'+q' = p-1+q = n-1 < n, S_{p'} = S_p-(n-|w|-1)$, $p'(n-q'-1) = (p-1)(n-|w|-1)$, $S_{q'} = S_q$, $S_{p'} - p'(n - q' - 1) - S_{q'} = S_p - (n - |w| - 1) - (p - 1)(n - |w| - 1) - S_q = S_p - p(n - |w| - 1) - S_q = 0$. So $0 < p' + q' < n$, $S_{p'} = p'(n - q' - 1) + S_{q'}$, both conditions are satisfied. So split graph with $d_{|u|} = |u| - 1$ is decomposable.

B) For nontrivial canonical decomposition, $d_p > n - q - 1$. proof: During the canonical decomposition, $d_p \geq n - q - 1$. If $d_p = p - 1 + n - p - q = n - q - 1$, then after decomposition the clique u has $d_{min} = p - 1$, then using A), the split graph $\{u, w\}$ is decomposable. This is in contradict to the definition of canonical decomposition that each split graph is indecomposable, So $d_p > n - q - 1$.

C) For nontrivial canonical decomposition, $d_p \neq d_{p+1}$. proof: Using B), $d_p >$

$n-q-1$. By definition, $d_{p+1} \leq n-p-q-1+p = n-q-1$. So $d_p \neq d_{p+1}$.

D) For nontrivial canonical decomposition, all decomposition put nodes with the same degree to the same group. proof: Using C), $d_p \neq d_{p+1}$. By definition, $d_{n-q} \neq d_{n-q+1}$. So the decomposition doesn't separate nodes with the same degree. □

5.3 Canonical decomposition algorithm with linear complexity

5.3.1 Idea

We want an efficient algorithm with which to find the canonical split graph decomposition of a simple graph. Such a decomposition consists of series of indecomposable split graphs, and perhaps one non-decomposable non-split graph.

A graph $\mathcal{G}$ is decomposable if there is a split graph $(\langle U, W\rangle; E)$ consisting of a set of completely connected nodes U and a set of completely disconnected nodes W such that $(\langle U, W\rangle; E) \circ \mathcal{H} = \mathcal{G}$. Assume that the graph $\mathcal{H}$ consists of vertex set V.

This algorithm relies on the idea that all nodes in a degree class separate together, except when the separating split graphs have either U or W empty. Note that for decomposed clique U with size p, every node in the independent set W with size q has degree smaller than p, and every node in the not yet decomposed part has degree at least p. Using this fact, we can use p to determine q, and increase p to find the first valid decomposition as canonical decomposition. This leads us to the following algorithm.

5.3.2 Algorithm

Consider a non-increasing degree sequence of length n: $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$. Note that nodes can be completely disconnected, i.e. $d_n = 0$ is possible.

Assume that the number of nodes with degree d is s_d, i.e. s_d is the size of degree class d which is the set of all nodes with degree d. Then

$$\sum_{j=0}^{n-1} s_j = n \ . \tag{5.5}$$

Define the number of nodes in the residual, not yet decomposed, sequence n_0, and initially set $n_0 = n$. Let k_0 and j_0 be the minimum and maximum degree class that is not yet decomposed, and initially set $k_0 = 0$ and $j_0 = n - 1$. Note that k_0 always points to the zero residual degree of the residual sequence.

1. If $s_{k_0} \neq 0$, then decompose a nested set of s_{k_0} split graphs, each of which consist of a single node in the W set. Set $n_0 = n_0 - s_{k_0}$. If $n_0 = 0$, then terminate program, decomposition is complete, else continue.

2. If $s_{j_0} = 0$, then set $j_0 = j_0 - 1$ and repeat this step. Else continue.

3. If $j_0 = n_0 - 1 + k_0$ and $s_{j_0} \neq 0$, then decompose a nested set of s_{j_0} split graphs:

   (a) If $s_{j_0} \neq n_0$ then each of the split graphs will consist of single node in the U set. Set $n_0 = n_0 - s_{j_0}$, $k_0 = k_0 + s_{j_0}$ $j_0 = j_0 - 1$, and return to step 1.

   (b) Else the first $s_{j_0} - 1$ split graphs will consist of a single node in the U set and the last one of which consists of a single node in the W set, then terminate program, decomposition is complete.

   Else continue.

4. Set $i = 0$,

$$p = s_{j_0} \ , \tag{5.6}$$

$$m_p = (j_0 - k_0)\, s_{j_0} \ , \tag{5.7}$$

and

$$q = \sum_{k=k_0+1}^{k_0+p-1} s_k \ , \tag{5.8}$$

$$m_q = \sum_{k=k_0+1}^{k_0+p-1} (k - k_0)\, s_k \ . \tag{5.9}$$

5. (a) If $p + q > n_0$, then all remaining n_0 nodes go into a final, non-split graph, and program terminates.

   (b) Else if

$$m_p - p(n_0 - 1 - q) - m_q = 0 \ , \tag{5.10}$$

   then decompose a split graph with $|U| = p$ and $|W| = q$, set $n_0 = n_0 - p - q$, $k_0 = k_0 + p$ and $j_0 = j_0 - 1 - i$, and then

      i. If $n_0 = 0$, then terminate program, decomposition is complete,

      ii. Else return to step 1.

   (c) Else set $i = i + 1$ and continue.

6. Set $k_1 = k_0 + p$,

$$p = p + s_{j_0 - i} \ , \tag{5.11}$$

$$m_p = m_p + (j_0 - i - k_0)\, s_{j_0 - i} \ , \tag{5.12}$$

and

$$q = q + \sum_{k=k_1}^{k_0+p-1} s_k \ , \tag{5.13}$$

$$m_q = m_q + \sum_{k=k_1}^{k_0+p-1} (k - k_0)\, s_k \ . \tag{5.14}$$

Return to step 5.

5.3.3 Computational complexity

The complexity of this algorithm is $\mathcal{O}(n)$. Note that each degree class gets visited just once before deciding about separation.

5.4 Theoretical results on graph composition

5.4.1 Degree distribution of composed graph

We can compose a degree sequence V for deeply nested graph using the degree sequence $\{u, w\}$ of a small indecomposable split graph as unit.

We define the notation as follows. u, V, and w are degree sequences, thus vectors. $|V|$ is the length of V, which is a scalar. The $+$ in $u + |V|$ is defined as adding the same scalar $|V|$ to every element in vector u. V_n is the degree sequence after n-th composition.

- For every new unit added, $V_n = \{u + |V_{n-1}|, V_{n-1} + |u|, w\}$.
- Using induction, $V_n = \{u + (n-1)|u| + |V_0| + (n-1)|w|, u + (n-1)|u| + |V_0| + (n-2)|w|, ..., u + (n-1)|u| + |V_0| + |w|, u + (n-1)|u| + |V_0|; V_0 + n|u|; w + (n-1)|u|, w + (n-2)|u|, ..., w + |u|, w\}$.
- So the degree distribution for large n is:
  1. w-block, $width = n|u|$, $density \sim |w|/|u|$;

2. u-block, $width = n|w|$, $density \sim |u|/|w|$;

Thus, the block or step structure of degree distribution only depends on the size of u and w, as shown in Figure 5.4, and the exact degree sequence u and w only matters when we look at the details of each block with small enough bin size.

5.4.2 Composed graphs are dense.

For large number of composition $t \gg 1$, we can calculate the number of edges $|E|$ of the composed graph up to the leading order:

- $|V_t| = |V_0| + (|u| + |w|)t \sim (|u| + |w|)t$,
- $t \sim \frac{|V_t|}{|u|+|w|}$,
- $|E_{t+1}| = |E_t| + |u||V_t| + C_1$, $C_1 = |E_u| + |E_{uw}|$,
- $\Delta|E_t| = |E_{t+1}| - |E_t| = |u||V_t| + C_1$,
- $\Delta|E_t| \sim |u|(|u| + |w|)t + C_1 \sim |u|(|u| + |w|)t$,
- $|E_t| = |E_0| + \sum_{\tau=0}^{t-1} \Delta|E_\tau|$,
- $|E_t| \sim |E_0| + \sum_{\tau=0}^{t-1} |u|(|u| + |w|)\tau \sim \frac{|u|(|u|+|w|)}{2} t^2$,
- $|E| \sim \frac{|u|(|u|+|w|)}{2}(\frac{|V_t|}{|u|+|w|})^2 = \frac{|u|}{|u|+|w|}\frac{|V|^2}{2} \sim O(|V|^2)$.

Since $|E| \sim O(|V|^2)$, the composed graph is dense.

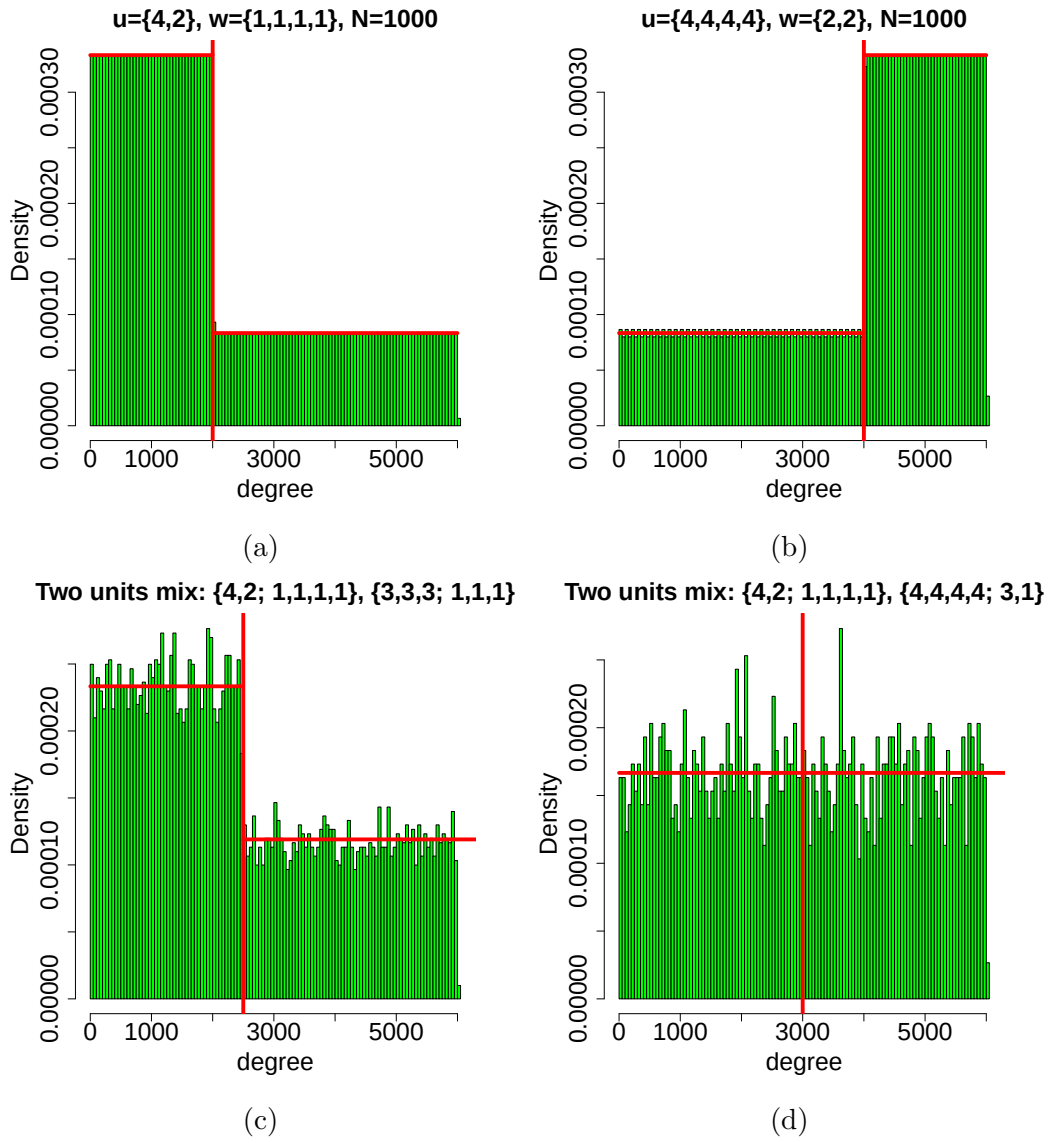
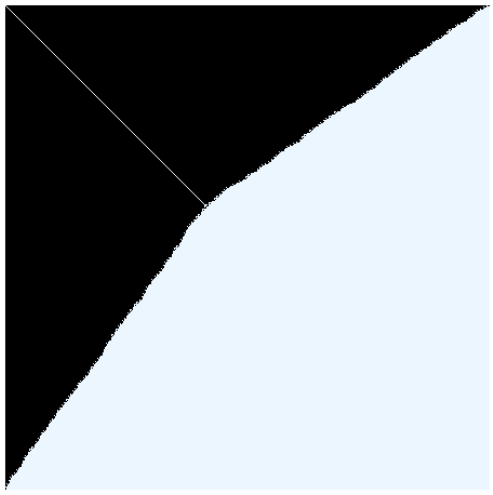
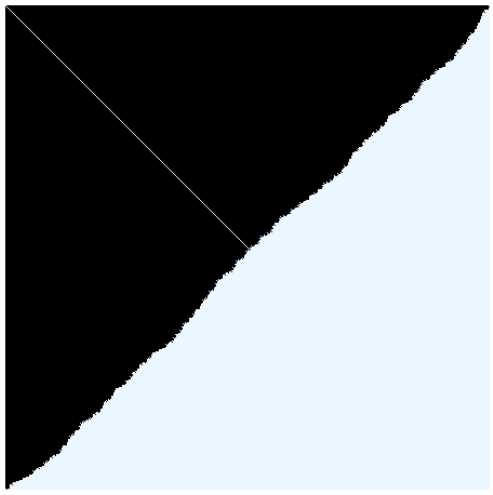


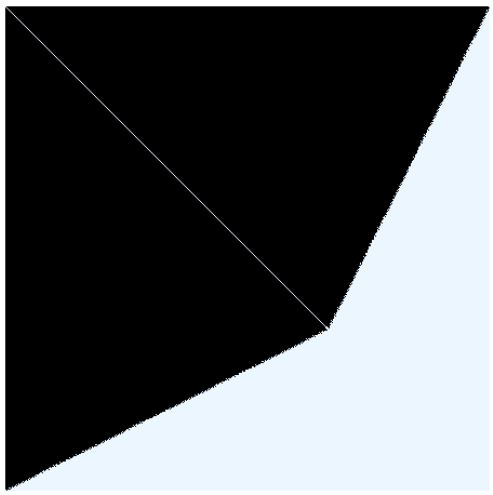
Figure 5.4: Degree distribution of composed graphs. (a) Degree distribution of a composed graph from a single unit split graph with $|u| = 2$ and $|w| = 4$, iterated 1000 times. (b) Single unit with $|u| = 4$ and $|w| = 2$. (c) Random mixture of two units with average $|\bar{u}| = 2.5$ and $|\bar{w}| = 3.5$. (d) Random mixture with $|\bar{u}| = 3$ and $|\bar{w}| = 3$. Red lines are theoretical predictions.



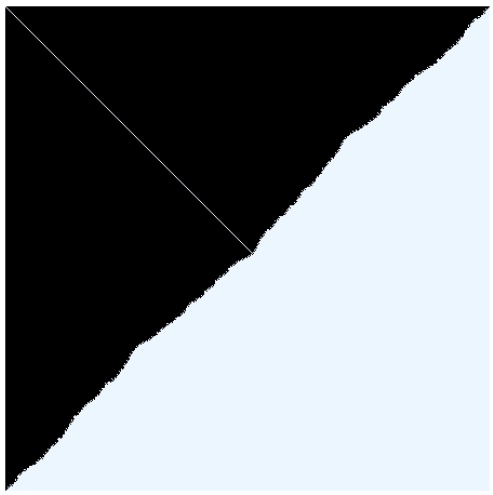
(a) Adjacency matrix, two units mix: {4,2; 1,1,1,1}, {3,3,3; 1,1,1}



(b) Adjacency matrix, two units mix: {4,2; 1,1,1,1}, {4,4,4,4; 3,1}

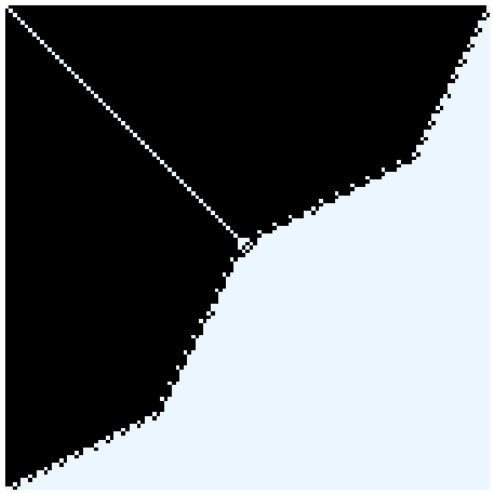


(c) Adjacency matrix, two units mix: {4,4,4,4; 3,1}, {4,4,4,4; 2,2}

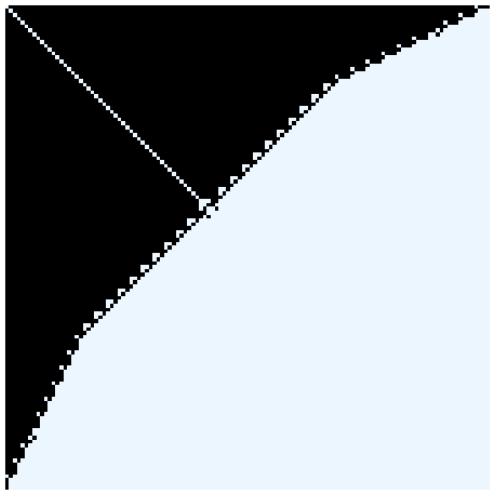


(d) Adjacency matrix, all units mix.

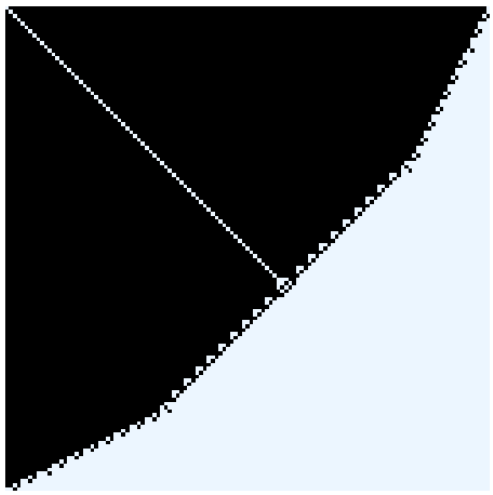
Figure 5.5: Adjacency matrices of composed graphs. Nodes are ordered by degree. Number of iterations $N = 100$. At each time step a random unit split graph is chosen. (a) $|\bar{u}| = 2.5$, $|\bar{w}| = 3.5$. (b) $|\bar{u}| = 3$, $|\bar{w}| = 3$. (c) $|\bar{u}| = 4$, $|\bar{w}| = 2$. (d) $|\bar{u}| = 3$, $|\bar{w}| = 3$. The overall wedge shape is only decided by $|\bar{u}|$ and $|\bar{w}|$. The degree sequence of unit split graph only affects the detailed shape at the boundary.



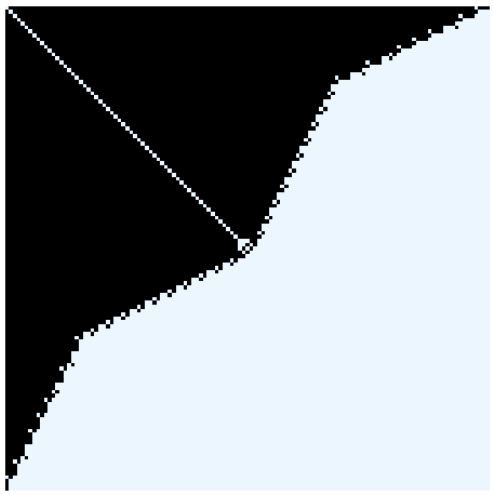
(a) Adjacency matrix, two units combined: {4,2; 1,1,1,1}, {4,4,4,4; 3,1}



(b) Adjacency matrix, two units combined: {3,3,3; 1,1,1}, {4,2; 1,1,1,1}



(c) Adjacency matrix, two units combined: {3,3,3; 1,1,1}, {4,4,4,4; 3,1}



(d) Adjacency matrix, two units combined: {4,4,4,4; 3,1}, {4,2; 1,1,1,1}

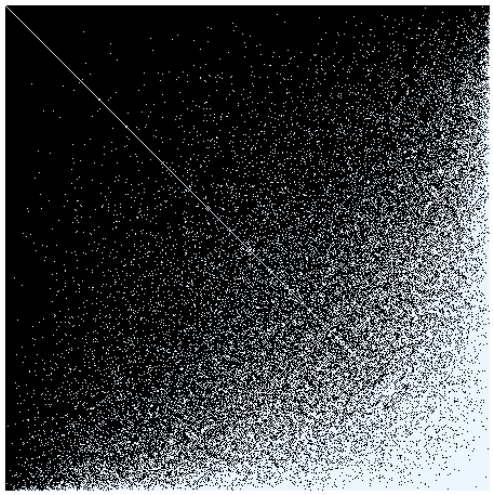
Figure 5.6: Adjacency matrices of composed graphs. Nodes are ordered by degree. The graph is composed by first using the first unit split graph for N times and then using the second unit split graph for another N times. Here we choose a small number of iterations $N = 10$ to see the details at the boundary.

5.5 Random power-law graphs are not deeply nested

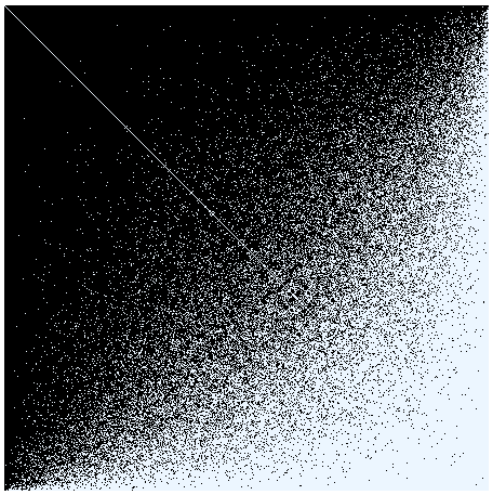
We generated random power-law degree sequences $P(k) \propto k^{-\gamma}$ with exponent $\gamma \in \{-2, -1, 0, 2\}$, and generated random graphs from those degree sequences. Figure 5.7 gives a few examples of their adjacency matrices.

When we try to perform canonical decomposition, we found that random power law sequences are not very decomposable. Only a few decomposition steps are performed, and a large portion of the graph left is not decomposable, as shown in Figure 5.8.

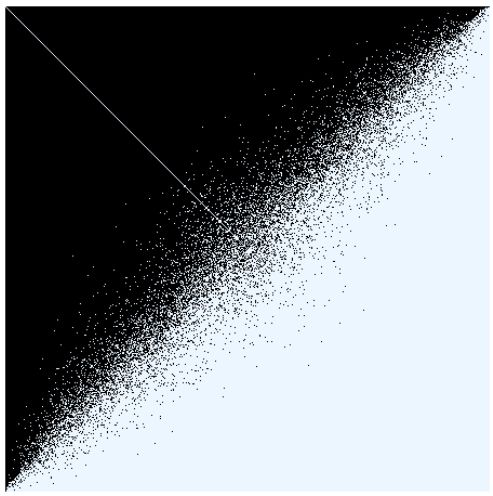
This suggests that if we find any graph that is deeply nested, it is very unlikely that this nestedness comes from random connection. The graph must be designed or evolved into this nested state.



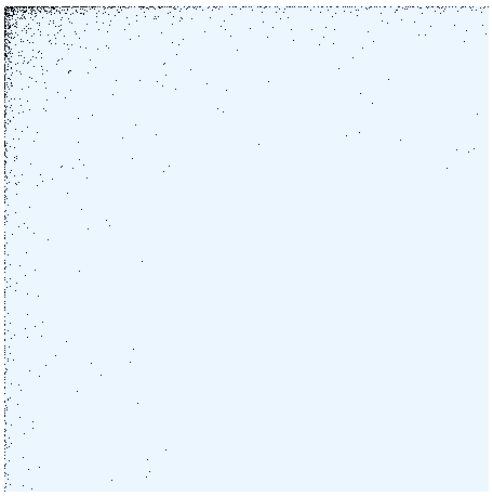
(a) Adjacency matrix of power-law degree sequence with $N = 600$ and $\gamma = -2$



(b) Adjacency matrix of power-law degree sequence with $N = 600$ and $\gamma = -1$

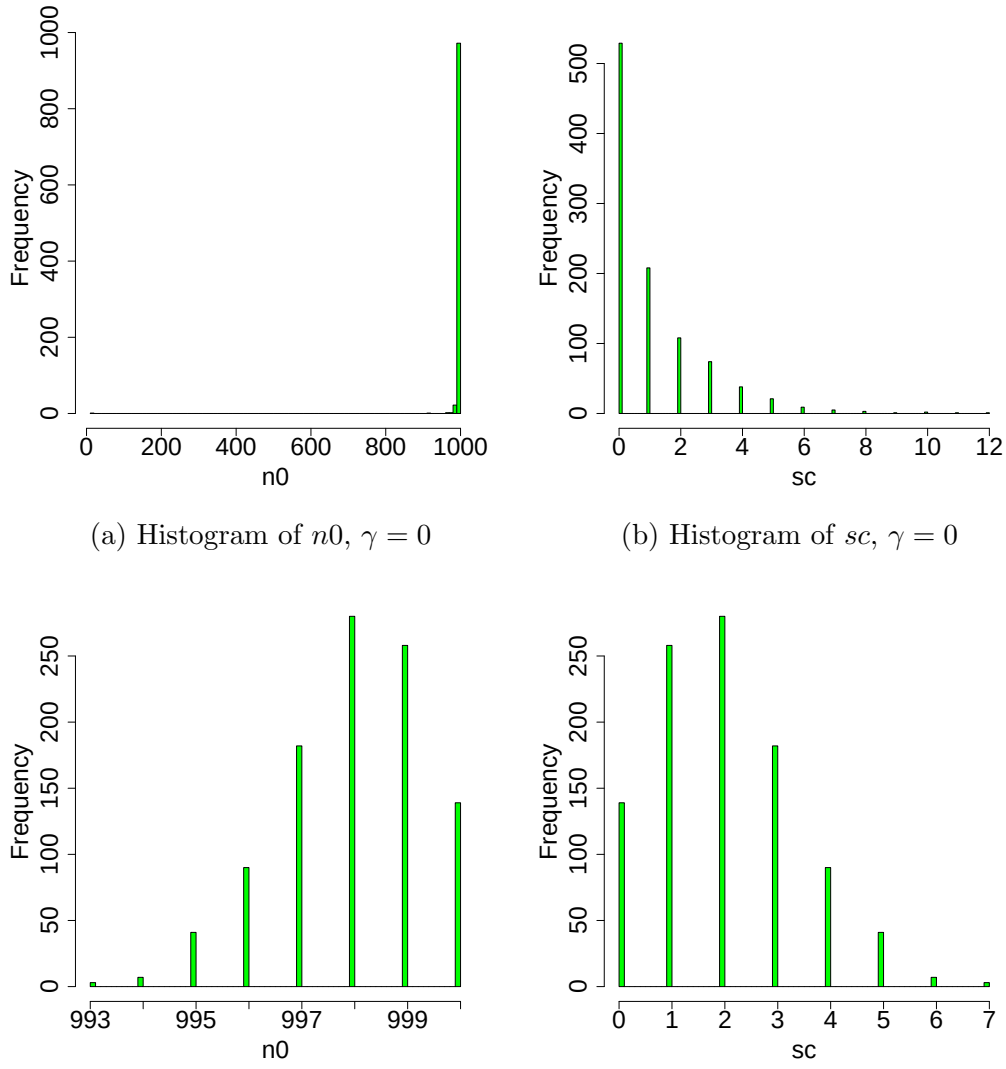


(c) Adjacency matrix of power-law degree sequence with $N = 600$ and $\gamma = 0$



(d) Adjacency matrix of power-law degree sequence with $N = 600$ and $\gamma = 2$

Figure 5.7: Adjacency matrices of graph from power-law degree distribution $P(k) \propto k^{-\gamma}$. Nodes are ordered by degree. Number of nodes is 600. (a) $\gamma = -2$. (b) $\gamma = -1$. (c) $\gamma = 0$. (d) $\gamma = 2$.



(a) Histogram of $n0$, $\gamma = 0$

(b) Histogram of sc, $\gamma = 0$

(c) Histogram of $n0$, $\gamma = -1$

(d) Histogram of sc, $\gamma = -1$

Figure 5.8: (a) Histogram of number of nodes left in the non-split graph $n0$ after canonical decomposition. Number of nodes $N = 1000$. Degree sequence follows power law $P(k) \propto k^{-\gamma}$ with exponent $\gamma = 0$. 1000 degree sequences are generated for better statistics. (b) Histogram of number of canonical decomposition steps performed sc for $\gamma = 0$. (c) Histogram of $n0$ for $\gamma = -1$ with same system size and number of degree sequences. (d) Histogram of sc for $\gamma = -1$.

5.6 Summary

In this chapter we studied the way to decomposed any graph into a series of split graphs. We first proved that all nodes with the same degree separate together in canonical decomposition. Utilizing this theorem, we developed a canonical decomposition algorithm with linear complexity.

We also studied the inverse problem: composing a graph using split graph. We found the degree sequence of composed graph, and showed that composed graphs are dense.

Finally, we applied our decomposition algorithm to random power-law degree sequences, and found that they are in general not deeply nested.

Chapter 6

Conclusion

In this dissertation, we discussed different ways to generate random graphs with prescribed degree sequence. We developed efficient stub sampling with a more stable weighted average, and compared the hard constraint method with soft constraint method. Then we studied the preferred degree extreme dynamics model, and a broader class of graph called split graph. This chapter summarizes those results and discusses the potential direction of research in the future.

In Chapter 2, we optimized the previously developed Sequential Importance Sampling (SIS) method, which is a hard constraint method, using our knowledge on Exponential Random Graph Model (ERGM), which is a soft constraint method. Here we use a mean-field approximation to estimate the probability to connect a pair of nodes in ERGM, and use this probability to decide the connection between Hub and nodes in the Allowed Set in SIS. This helps reduce the variance of log-weight in SIS. Moreover, we developed a way to calculate the weighted average of graph property

by measuring the distribution parameter of the property and log-weight joint probability distribution as long as this distribution is bivariate normal. We can get more stable weighted average using this bivariate normal assumption. Using efficient stub sampling with bivariate normal assumption, we studied a real-world social network with 1 million nodes, and concluded that it's very unlikely that this specific network is formed by chance.

In Chapter 3, we studied the difference of ensembles generated using soft and hard constraint methods developed in Chapter 2. We showed, both theoretically and through simulation, that soft constraint method significantly overestimates the global clustering coefficient. Thus, we need to be cautious about the null models we use when doing statistical inference.

In Chapter 4, we analyzed a different model that tries to generate graphs with preferred degree: the preferred degree extreme dynamics model, and its special cases eXtreme Introverts and Extroverts (XIE) model, and Generalized Introverts and Extroverts (GIE) model. We solved the degree distribution, cross-link distribution and correlation in XIE model. We also conjectured that the system following extreme dynamics would reach an equilibrium as long as the preferred degrees are integers.

In Chapter 5, we explored the properties of split graphs, which were seen in Chapter 4. We proved that during canonical decomposition, nodes with the same degree would be decomposed into the same component. Using this theorem, we developed a linear complexity algorithm to perform canonical decomposition. We also studied the inverse problem: graph composition. Here we found the degree distribution of composed graph, and proved that composed graphs are dense. Finally, we tried to decompose random power-law degree sequences, but found that most

random power-law degree sequences are not deeply nested. This indicates that any deeply nested graphs we observed must be either designed or evolved into that state.

Though we made some progress in this dissertation, there are still a lot of questions to be answered. In SIS, while we tried to reduce the variance of log-weight, it still increases approximately linearly as the system size goes larger. It might be promising to use resampling method to control the range of weight, or even get rid of the weight altogether. In ERGM, whether there exists a simple but more accurate approximation of connection probability for all possible degree sequences is an important question both theoretically and in practice. In GIE, other studies showed that the system is in non-equilibrium when preferred degrees are non-integers. The physical meaning of this phenomena and its potential application are yet to be found. For split graphs, the definition of split graph and canonical decomposition might be too strict for real-world graphs. Instead of asking whether or not a graph is decomposable, we might want to ask to what extent a decomposition describes the structure in the original graph. A "softer" definition of decomposition may give us a less accurate but more coarse-grained description of real-world graphs.

Bibliography

[1] W. Aiello, F. Chung, and L. Lu. A Random Graph Model for Massive Graphs. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 171–180, New York, NY, USA, 2000. ACM.

[2] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the North American power grid. *Physical Review E*, 69(2):025103, pages 1–4, Feb. 2004.

[3] R. Albert and A.-L. Barabási. Topology of Evolving Networks: Local Events and Universality. *Physical Review Letters*, 85(24):5234–5237, Dec. 2000.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan. 2002.

[5] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the World-Wide Web. *Nature*, 401(6749):130–131, Sept. 1999.

[6] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, June 2007.

[7] C. J. Anderson, S. Wasserman, and B. Crouch. A p* primer: logit models for social networks. *Social Networks*, 21(1):37–66, Jan. 1999.

[8] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, Oct. 1999.

[9] A.-L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, Feb. 2004.

[10] A. Barrat and M. Weigt. On the properties of small-world network models. *The European Physical Journal B - Condensed Matter and Complex Systems*, 13(3):547–560, Feb. 2000.

[11] J. Bascompte. Structure and Dynamics of Ecological Networks. *Science*, 329(5993):765–766, Aug. 2010.

[12] K. E. Bassler, D. Dhar, and R. K. P. Zia. Networks with preferred degree: a mini-review and some new results. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(7):P07013, pages 1–38, July 2015.

[13] K. E. Bassler, C. I. D. Genio, P. L. Erdős, I. Miklós, and Z. Toroczkai. Exact sampling of graphs with prescribed degree correlations. *New Journal of Physics*, 17(8):083052, pages 1–18, Aug. 2015.

[14] K. E. Bassler, W. Liu, B. Schmittmann, and R. K. P. Zia. Extreme Thouless effect in a minimal model of dynamic social networks. *Physical Review E*, 91(4):042102, pages 1–10, Apr. 2015.

[15] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261 [cs, stat]*, June 2018.

[16] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, May 1978.

[17] G. Bianconi. The entropy of randomized network ensembles. *Europhysics Letters*, 81(2):28005, pages 1–6, Dec. 2007.

[18] G. Bianconi. Entropy of network ensembles. *Physical Review E*, 79(3):036114, pages 1–10, Mar. 2009.

[19] P. Billingsley. *Probability and Measure*. Wiley-Interscience, New York, 3 edition edition, May 1995.

[20] J. Blitzstein and P. Diaconis. A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees. *Internet Mathematics*, 6(4):489–522, 2010.

[21] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175–308, Feb. 2006.

[22] M. Boguñá, R. Pastor-Satorras, and A. Vespignani. Cut-offs and finite size effects in scale-free networks. *The European Physical Journal B*, 38(2):205–209, Mar. 2004.

[23] B. Bollobás. *Random Graphs: Second Edition*. Cambridge University Press, Cambridge ; New York, 2 edition, Oct. 2001.

[24] B. Bollobas. *Modern Graph Theory*. Springer, New York, corrected edition, Aug. 2002.

[25] B. Bollobás and O. Riordan. Random Graphs and Branching Processes. In B. Bollobás, R. Kozma, and D. Miklós, editors, *Handbook of Large-Scale Random Networks*, Bolyai Society Mathematical Studies, pages 15–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[26] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca. Network Analysis in the Social Sciences. *Science*, 323(5916):892–895, Feb. 2009.

[27] M. Boss, H. Elsinger, M. Summer, and S. Thurner 4. Network topology of the interbank market. *Quantitative Finance*, 4(6):677–684, Dec. 2004.

[28] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen. Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, May 2003.

[29] C. T. Butts. A perfect sampling method for exponential family random graph models. *The Journal of Mathematical Sociology*, 42(1):17–36, Jan. 2018.

[30] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, May 2009.

[31] M. Catanzaro, M. Boguñá, and R. Pastor-Satorras. Generation of uncorrelated random scale-free networks. *Physical Review E*, 71(2):027103, pages 1–4, Feb. 2005.

[32] S. Chatterjee and P. Diaconis. Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461, Oct. 2013.

[33] S. Chatterjee, P. Diaconis, and A. Sly. Random graphs with a given degree sequence. *The Annals of Applied Probability*, 21(4):1400–1435, Aug. 2011.

[34] T. Chen, P. Singh, and K. E. Bassler. Network community detection using modularity density measures. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(5):053406, pages 1–15, May 2018.

[35] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, Dec. 2002.

[36] F. Chung and L. Lu. Connected Components in Random Graphs with Given Expected Degree Sequences. *Annals of Combinatorics*, 6(2):125–145, Nov. 2002.

[37] F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, Providence, RI, Aug. 2006.

[38] S. Ciliberti, O. C. Martin, and A. Wagner. Innovation and robustness in complex regulatory gene networks. *Proceedings of the National Academy of Sciences*, 104(34):13591–13596, Aug. 2007.

[39] P. Colomer-de Simon and M. Boguñá. Clustering of random scale-free networks. *Physical Review E*, 86(2):026120, pages 1–5, Aug. 2012.

[40] A. C. C. Coolen, A. D. Martino, and A. Annibale. Constrained Markovian Dynamics of Random Graphs. *Journal of Statistical Physics*, 136(6):1035–1067, Sept. 2009.

[41] C. Cooper, M. Dyer, and C. Greenhill. Sampling Regular Graphs and a Peer-to-Peer Network. *Combinatorics, Probability and Computing*, 16(4):557–593, July 2007.

[42] L. d. F. Costa, O. N. O. Jr, G. Travieso, F. A. Rodrigues, P. R. V. Boas, L. Antiqueira, M. P. Viana, and L. E. C. Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, June 2011.

[43] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, Jan. 2007.

[44] A. C. Davison and D. V. Hinkley. *Bootstrap Methods And Their Application*. Cambridge University Press, Cambridge ; New York, NY, USA, 1 edition, Oct. 1997.

[45] C. I. Del Genio, T. Gross, and K. E. Bassler. All Scale-Free Networks Are Sparse. *Physical Review Letters*, 107(17):178701, pages 1–4, Oct. 2011.

[46] C. I. Del Genio, H. Kim, Z. Toroczkai, and K. E. Bassler. Efficient and Exact Sampling of Simple Graphs with Given Arbitrary Degree Sequence. *PLoS ONE*, 5(4):e10012, pages 1–7, Apr. 2010.

[47] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4):1275–1335, Oct. 2008.

[48] S. N. Dorogovtsev and J. F. F. Mendes. Scaling behaviour of developing and decaying networks. *Europhysics Letters*, 52(1):33–39, Oct. 2000.

[49] S. N. Dorogovtsev and J. F. F. Mendes. Effect of the accelerating growth of communications networks on their structure. *Physical Review E*, 63(2):025101, pages 1–4, Jan. 2001.

[50] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, June 2002.

[51] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of Growing Networks with Preferential Linking. *Physical Review Letters*, 85(21):4633–4636, Nov. 2000.

[52] A. Doucet, N. d. Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer-Verlag, New York, 2001.

[53] J. C. Doyle, D. L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger. The "robust yet fragile" nature of the Internet. *Proceedings of the National Academy of Sciences*, 102(41):14497–14502, Oct. 2005.

[54] P. Erdős and A. Rényi. On the Evolution of Random Graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[55] P. L. Erdős, I. Miklós, and Z. Toroczkai. New Classes of Degree Sequences with Fast Mixing Swap Markov Chain Sampling. *Combinatorics, Probability and Computing*, 27(2):186–207, Mar. 2018.

[56] P. Erdős and T. Gallai. Graphs with prescribed degrees of vertices [hungarian]. *Matematikai Lapok*, 11:264–274, 1960.

[57] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[58] M. Ezzatabadipour. *Non-Equilibrium Statistical Mechanics of a Mixed Order Phase Transition in a Dynamical Network Model*. PhD thesis, University of Houston, 2019.

[59] S. Foldes and P. L. Hammer. Split Graphs Having Dilworth Number Two. *Canadian Journal of Mathematics*, 29(3):666–672, June 1977.

[60] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, Feb. 2010.

[61] S. Fortunato and D. Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, Nov. 2016.

[62] A. Fronczak. Exponential Random Graph Models. In R. Alhajj and J. Rokne, editors, *Encyclopedia of Social Network Analysis and Mining*, pages 1–18. Springer, New York, 2017.

[63] P. Gao and N. Wormald. Enumeration of graphs with a heavy-tailed degree sequence. *Advances in Mathematics*, 287:412–450, Jan. 2016.

[64] D. F. Gatz and L. Smith. The standard error of a weighted mean concentration—I. Bootstrapping vs other methods. *Atmospheric Environment*, 29(11):1185–1193, June 1995.

[65] J. Gómez-Gardeñes and Y. Moreno. Local versus global knowledge in the Barabási-Albert scale-free network model. *Physical Review E*, 69(3):037103, pages 1–4, Mar. 2004.

[66] C. Greenhill and M. Sfragara. The switch Markov chain for sampling irregular graphs and digraphs. *Theoretical Computer Science*, 719:1–20, Apr. 2018.

[67] J.-L. Guillaume and M. Latapy. Bipartite structure of all complex networks. *Information Processing Letters*, 90(5):215–221, June 2004.

[68] R. Guimerà, S. Mossa, A. Turtschi, and L. a. N. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799, May 2005.

[69] S. Hakimi. On Realizability of a Set of Integers as Degrees of the Vertices of a Linear Graph. I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, Sept. 1962.

[70] B. Hall, A. Jaffe, and M. Trajtenberg. *The NBER patent citations data file: Lessons, insights and methodological tools*. 2001.

[71] V. Havel. Poznámka o existenci konečných grafů. *Časopis pro pěstování matematiky*, 080(4):477–480, 1955.

[72] N. Henze and B. Zirkler. A class of invariant consistent tests for multivariate normality. *Communications in Statistics - Theory and Methods*, 19(10):3595–3617, Jan. 1990.

[73] S. M. Herbert Robbins. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 1951.

[74] P. W. Holland and S. Leinhardt. A Method for Detecting Structure in Sociometric Data. *American Journal of Sociology*, 76(3):492–513, 1970.

[75] P. Holme and B. J. Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, pages 1–4, Jan. 2002.

[76] S. Horvát, Czabarka, and Z. Toroczkai. Reducing Degeneracy in Maximum Entropy Models of Networks. *Physical Review Letters*, 114(15):158701, pages 1–5, Apr. 2015.

[77] H.-B. Hu and X.-F. Wang. Disassortative mixing in online social networks. *Europhysics Letters*, 86(1):18003, pages 1–6, Apr. 2009.

[78] P. Hu and W. C. Lau. A Survey and Taxonomy of Graph Sampling. *arXiv preprint arXiv:1308.5865 [cs, math, stat]*, Aug. 2013.

[79] D. R. Hunter and M. S. Handcock. Inference in Curved Exponential Family Models for Networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, Sept. 2006.

[80] E. T. Jaynes. Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630, May 1957.

[81] E. T. Jaynes. Information Theory and Statistical Mechanics. II. *Physical Review*, 108(2):171–190, Oct. 1957.

[82] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Vol. 1.* Wiley-Interscience, New York, 2 edition edition, Oct. 1994.

[83] V. Kalapala, V. Sanwalani, A. Clauset, and C. Moore. Scale invariance in road networks. *Physical Review E*, 73(2):026130, pages 1–6, Feb. 2006.

[84] Kaluza Pablo, Kölzsch Andrea, Gastner Michael T., and Blasius Bernd. The complex network of global cargo ship movements. *Journal of The Royal Society Interface*, 7(48):1093–1103, July 2010.

[85] R. Kannan, P. Tetali, and S. Vempala. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. *Random Structures & Algorithms*, 14(4):293–308, 1999.

[86] F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, Cambridge ; New York, revised ed. edition edition, Aug. 2011.

[87] J. F. Kenney and E. S. Keeping. *Mathematics of statistics. Part 2 ,second edition*. D. Van Nostrand, Princeton, N.J., 1951. OCLC: 122291120.

[88] H. Kim, C. I. D. Genio, K. E. Bassler, and Z. Toroczkai. Constructing and sampling directed graphs with given degree sequences. *New Journal of Physics*, 14(2):023012, pages 1–23, Feb. 2012.

[89] H. Kim, Z. Toroczkai, P. L. Erdős, I. Miklós, and L. A. Székely. Degree-based graph construction. *Journal of Physics A: Mathematical and Theoretical*, 42(39):392001, pages 1–10, Sept. 2009.

[90] L. Kish. *Survey sampling*. Wiley, Jan. 1965.

[91] H. Klein-Hennig and A. K. Hartmann. Bias in generation of random graphs. *Physical Review E*, 85(2):026101, pages 1–7, Feb. 2012.

[92] K. Klemm and V. M. Eguíluz. Highly clustered scale-free networks. *Physical Review E*, 65(3):036123, pages 1–5, Feb. 2002.

[93] B. Klimt and Y. Yang. Introducing the enron corpus. In *First Conference on Email and Anti-Spam (CEAS) Proceedings*, 2004.

[94] P. L. Krapivsky, S. Redner, and F. Leyvraz. Connectivity of Growing Random Networks. *Physical Review Letters*, 85(21):4629–4632, Nov. 2000.

[95] J. Leskovec and C. Faloutsos. Sampling from Large Graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 631–636, New York, NY, USA, 2006. ACM.

[96] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[97] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, June 2001.

[98] W. Liu, B. Schmittmann, and R. K. P. Zia. Extraordinary variability and sharp transitions in a maximally frustrated dynamic network. *Europhysics Letters*, 100(6):66007, Dec. 2012.

[99] Y. Liu, T. Safavi, A. Dighe, and D. Koutra. Graph Summarization Methods and Applications: A Survey. *ACM Comput. Surv.*, 51(3):62:1–62:34, June 2018.

[100] Y.-Y. Liu and A.-L. Barabási. Control principles of complex systems. *Reviews of Modern Physics*, 88(3):035006, pages 1–58, Sept. 2016.

[101] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou. Vital nodes identification in complex networks. *Physics Reports*, 650:1–63, Sept. 2016.

[102] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou. Recommender systems. *Physics Reports*, 519(1):1–49, Oct. 2012.

[103] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, Mar. 2011.

[104] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142, Dec. 2013.

[105] K. V. Mardia. Measures of Multivariate Skewness and Kurtosis with Applications. *Biometrika*, 57(3):519–530, 1970.

[106] S. Maslov, K. Sneppen, and A. Zaliznyak. Detection of topological patterns in complex networks: correlation profile of the internet. *Physica A: Statistical Mechanics and its Applications*, 333:529–540, Feb. 2004.

[107] B. D. McKay and N. C. Wormald. Asymptotic enumeration by degree sequence of graphs with degreeso(n1/2). *Combinatorica*, 11(4):369–382, Dec. 1991.

[108] G. A. Miller. *WordNet: An Electronic Lexical Database*. A Bradford Book, Cambridge, Mass, May 1998.

[109] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint arXiv:cond-mat/0312028*, Dec. 2003.

[110] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

[111] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 1995.

[112] M. Molloy and B. Reed. The Size of the Giant Component of a Random Graph with a Given Degree Sequence. *Combinatorics, Probability and Computing*, 7(3):295–305, Sept. 1998.

[113] M. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, Jan. 2003.

[114] M. Newman. *Networks: An Introduction*. Oxford University Press, Oxford ; New York, 1 edition, May 2010.

[115] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, Jan. 2001.

[116] M. E. J. Newman. Assortative Mixing in Networks. *Physical Review Letters*, 89(20):208701, pages 1–4, Oct. 2002.

[117] M. E. J. Newman. Random graphs as models of networks. In S. Bornholdt and H. G. Schuster, editors, *Handbook of Graphs and Networks*, pages 35–68. Wiley-VCH Verlag GmbH & Co. KGaA, 2002. DOI: 10.1002/3527602755.ch2.

[118] M. E. J. Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25(1):83–95, Jan. 2003.

[119] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, Jan. 2012.

[120] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, pages 1–17, July 2001.

[121] M. E. J. Newman and D. J. Watts. Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4):341–346, Dec. 1999.

[122] K. Norlen, G. Lucas, M. Gebbie, and J. Chuang. EVA: Extraction, Visualization and Analysis of the Telecommunications and Media Ownership Network. In *Proceedings of International Telecommunications Society 14th Biennial Conference (ITS2002), Seoul Korea*, pages 27–129.

[123] A. Nyberg, T. Gross, and K. E. Bassler. Mesoscopic structures and the Laplacian spectra of random geometric graphs. *Journal of Complex Networks*, 3(4):543–551, Dec. 2015.

[124] D. Obradović and M. Danisch. Direct generation of random graphs exactly realising a prescribed degree sequence. In *2014 6th International Conference on Computational Aspects of Social Networks*, pages 1–6, July 2014.

[125] C. Orsini, M. M. Dankulov, P. Colomer-de Simón, A. Jamakovic, P. Mahadevan, A. Vahdat, K. E. Bassler, Z. Toroczkai, M. Boguñá, G. Caldarelli, S. Fortunato, and D. Krioukov. Quantifying randomness in real networks. *Nature Communications*, 6:8627, pages 1–10, Oct. 2015.

[126] G. A. Pagani and M. Aiello. The Power Grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, June 2013.

[127] A. Paranjape, A. R. Benson, and J. Leskovec. Motifs in Temporal Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 601–610, New York, NY, USA, 2017. ACM.

[128] D. D. S. Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306, 1976.

[129] D. J. D. S. Price. Networks of Scientific Papers. *Science*, 149(3683):510–515, 1965.

[130] M. Pujari and R. Kanawati. Supervised Rank Aggregation Approach for Link Prediction in Complex Networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1189–1196, New York, NY, USA, 2012. ACM.

[131] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555, Aug. 2002.

[132] E. S. Roberts, A. Annibale, and A. C. C. Coolen. Controlled Markovian Dynamics of Graphs: Unbiased Generation of Random Graphs with Prescribed Topological Properties. In C. Grácio, D. Fournier-Prunaret, T. Ueta, and Y. Nishio, editors, *Nonlinear Maps and their Applications*, Springer Proceedings in Mathematics & Statistics, pages 25–34. Springer New York, 2014.

[133] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29(2):173–191, May 2007.

[134] G. Rotundo and A. M. D'Arcangelis. Ownership and control in shareholding networks. *Journal of Economic Interaction and Coordination*, 5(2):191–219, Dec. 2010.

[135] P. Royston. Approximating the Shapiro-Wilk W-test for non-normality. *Statistics and Computing*, 2(3):117–119, Sept. 1992.

[136] R. Ryder. probability - Distribution and Variance of Count of Triangles in Random Graph, Apr. 2018.

[137] W. E. Schlauch, E. Á. Horvát, and K. A. Zweig. Different flavors of randomness: comparing random graph models with fixed degree sequences. *Social Network Analysis and Mining*, 5(1):36, pages 1–14, July 2015.

[138] P. Sen, S. Dasgupta, A. Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna. Small-world properties of the Indian railway network. *Physical Review E*, 67(3):036106, pages 1–5, Mar. 2003.

[139] C. Seshadhri, T. G. Kolda, and A. Pinar. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E*, 85(5):056109, pages 1–9, May 2012.

[140] H. A. Simon. On a Class of Skew Distribution Functions. *Biometrika*, 42(3/4):425–440, 1955.

[141] T. A. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40, April 2002.

[142] R. V. Solé, R. Pastor-Satorras, E. Smith, and T. B. Kepler. A model of large-scale proteome evolution. *Advances in Complex Systems*, 05(01):43–54, Mar. 2002.

[143] O. Sporns. The human connectome: a complex network. *Annals of the New York Academy of Sciences*, 1224(1):109–125, 2011.

[144] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–425, Sept. 2004.

[145] D. Strauss. On a General Class of Models for Interaction. *SIAM Review*, 28(4):513–527, Dec. 1986.

[146] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, Mar. 2001.

[147] G. Strona, D. Nappo, F. Boccacci, S. Fattorini, and J. San-Miguel-Ayanz. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, 5:4114, pages 1–7, June 2014.

[148] W. M. Tam, F. C. M. Lau, and C. K. Tse. Complex-Network Modeling of a Call Network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(2):416–429, Feb. 2009.

[149] R. Taylor. Contrained switchings in graphs. In K. L. McAvaney, editor, *Combinatorial Mathematics VIII*, Lecture Notes in Mathematics, pages 314–336. Springer Berlin Heidelberg, 1981.

[150] Tom A. B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. New Specifications for Exponential Random Graph Models. *Sociological Methodology*, 36(1):99–153, Aug. 2006.

[151] R. Tyshkevich. Decomposition of graphical sequences and unigraphs. *Discrete Mathematics*, 220(1):201–238, June 2000.

[152] R. I. Tyškevič and A. A. Černjak. Canonical decomposition of a graph determined by the degrees of its vertices. *Vescī Akadèmīī Navuk BSSR. Seryja Fīzīka-Matèmatyčnyh Navuk*, 5(5):14–26, 138, 1979.

[153] A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of Protein Interaction Networks. *Complexus*, 1(1):38–44, 2003.

[154] N. D. Verhelst. An Efficient MCMC Algorithm to Sample Binary Matrices with Fixed Marginals. *Psychometrika*, 73(4):705–728, Apr. 2008.

[155] F. Viger and M. Latapy. Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence. In L. Wang, editor, *Computing and Combinatorics*, Lecture Notes in Computer Science, pages 440–449. Springer Berlin Heidelberg, 2005.

[156] S. Vitali, J. B. Glattfelder, and S. Battiston. The Network of Global Corporate Control. *PLoS ONE*, 6(10):e25995, pages 1–6, Oct. 2011.

[157] X. F. Wang and G. Chen. Complex networks: small-world, scale-free and beyond. *IEEE Circuits and Systems Magazine*, 3(1):6–20, 2003.

[158] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[159] M. Winlaw, H. DeSterck, and G. Sanders. An In-Depth Analysis of the Chung-Lu Model. Technical Report LLNL-TR-678729, Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), Oct. 2015.

[160] J. Wishart. The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population. *Biometrika*, 20A(1/2):32–52, 1928.

[161] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A Comprehensive Survey on Graph Neural Networks. *arXiv preprint arXiv:1901.00596 [cs, stat]*, Jan. 2019.

[162] Y. Xia, C. K. Tse, W. M. Tam, F. C. M. Lau, and M. Small. Scale-free user-network approach to telephone network traffic analysis. *Physical Review E*, 72(2):026116, pages 1–7, Aug. 2005.

[163] J. Zhang and Y. Chen. Sampling for Conditional Inference on Network Data. *Journal of the American Statistical Association*, 108(504):1295–1307, Dec. 2013.

[164] W. Zhang and K. E. Bassler. Efficient sampling of ensembles of large graphs with prescribed degrees. (in preparation for publication).

[165] W. Zhang, E. McMillan, and K. E. Bassler. Graphs sampled from soft degree-sequence-constraint methods are more clustered. (in preparation for publication).

[166] Z. Zhang, P. Cui, and W. Zhu. Deep Learning on Graphs: A Survey. *arXiv preprint arXiv:1812.04202 [cs, stat]*, Dec. 2018.

[167] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. Graph Neural Networks: A Review of Methods and Applications. *arXiv preprint arXiv:1812.08434 [cs, stat]*, Dec. 2018.

[168] R. K. P. Zia, W. Liu, S. Jolad, and B. Schmittmann. Studies of adaptive networks with preferred degree. *Physics Procedia*, 15:102–105, Jan. 2011.

[169] R. K. P. Zia, W. Liu, and B. Schmittmann. An Extraordinary Transition in a Minimal Adaptive Network of Introverts and Extroverts. *Physics Procedia*, 34:124–127, Jan. 2012.

[170] R. K. P. Zia, W. Zhang, M. Ezzatabadipour, and K. E. Bassler. Exact results for the extreme Thouless effect in a model of network dynamics. *Europhysics Letters*, 124(6):60008, pages 1–6, Jan. 2019.