Hitting Time of the Von Neumann Entropy for Networks Undergoing Rewiring

By

Zichao Li

Senior Honors Thesis
Department of Mathematics
University of North Carolina at Chapel Hill

April 15, 2016

Approved:

Peter Mucha, Thesis Advisor

David Adalsteinsson, Reader

Katherine Newhall, Reader

Abstract

A random graph model defines a distribution over graphs, and this distribution induces a distribution over certain measurements of graphs, such as the Von Neumann entropy. Interestingly, the Von Neumann entropy of an empirical network typically has a very small probability to be drawn from the distribution over the Von Neumann entropy of graphs generated by the Erdős-Rényi random graph model with the same number of vertices and edges. It has been shown that Erdős-Rényi random graph model may be inappropriate for modeling certain properties of real-world networks such as the small-world property [22] and scale-free property [2], and the Von Neumann entropy provides yet another, complementary way to measure how real-world networks differ from Erdős-Rényi random graphs. Subjecting the network to a random rewiring process offers another approach to measure how far it is from an Erdős-Rényi random graph. In particular, it can be shown using Markov chain theory that the ensemble of networks after many non-degree-preserving rewirings limits to the ensemble of Erdős-Rényi random graphs. In this paper, we develop a connection between these two approaches by studying the Von Neumann entropy of networks undergoing rewiring. More specifically, we are interested in the number of rewiring times needed until the Von Neumann entropy of the rewired graph is larger than some quantile of the distribution over the Von Neumann entropy of Erdős-Rényi random networks, as it can be used to quantify the difference between any given network and networks generated by the Erdős-Rényi random graph model. However, performing a large number of simulations to compute the expected number of rewiring times is not computationally efficient, and we apply matrix perturbation methods to derive an estimation by using a small perturbation to the adjacency matrix to approximate one random rewiring of a graph. The estimation can be computed directly for a given network without performing any numerical simulation.

Acknowledgement

First, I would like to thank my thesis advisor Prof. Peter Mucha, who provided me with the opportunity to do research on networks. He sparked my interest in networks analysis, and without his constant encouragement and mentorship I would not have been able to learn about networks and complete this thesis.

I would also like to thank Dr. Dane Taylor, who provided detailed guidance in both developing the outline of this research project and deriving the equations using matrix perturbation theory. I sincerely appreciate the time and effort he invested to discuss with me about this research project during our weekly meetings.

Additional thanks must also be given to other group members in Prof. Mucha's research group for asking questions and giving suggestions when I gave oral presentations of my research project. In addition, I would like to thank Dr. Saray Shai for helping me learn how to run MATLAB programs on UNC's research computing systems Kure and Killdevil, which greatly reduced the runtime of some numerical experiments.

Finally, I must express my gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of doing research and writing this thesis. This accomplishment would not have been possible without them.

Contents

1 Introduction

In recent years, networks have gained increasing attention as a tool for modeling complex systems, including social, biological, technological, and information networks. Networks can be considered as graphs in which vertices represent individuals and edges represent interactions between individuals, and they can be represented by adjacency matrices. Given a simple graph, the adjacency matrix A is defined as

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertex } i \text{ and vertex } j \\ 0 & \text{otherwise} \end{cases}$$

and the degree matrix D is defined as

$$D_{ij} = \begin{cases} \sum_k A_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Spectral graph theory is the study of properties related to the eigenvalues and eigenvectors of matrices associated to a graph, and one important matrix is the Laplacian matrix. Given a simple graph, the Laplacian matrix L is defined as

$$L = D - A$$

Hence the elements of the Laplacian matrix L are given by

$$L_{ij} = \begin{cases} D_{ij} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } A_{ij} = A_{ji} = 1 \\ 0 & \text{otherwise} \end{cases}$$

(There are several different ways to define a Laplacian matrix, and this version is often called either the combinatorial or unnormalized Laplacian matrix.) The concept of the Laplacian matrix of a graph is used in a wide range of applications such as analyzing diffusion and random walks on networks [4, 19, 20], calculating the number of spanning trees for a graph [16, 18, 5], and performing community detection [17, 11].

In this paper, we will study the Von Neumann entropy [8, 18, 5] of networks undergoing random rewiring process. The Von Neumann entropy of a graph is a concept that allows one to measure the similarity of networks based on the spectra of the Laplacian matrix L. More specifically, given a network, we are interested in the expected number of rewiring times needed until the Von Neumann entropy of the network undergoing random rewiring

is "close enough" to the distribution of Von Neumann entropy of networks generated by a random graph model with the same number of vertices and edges. The Von Neumann entropy of a graph is closely related to the spectral properties of its Laplacian matrix, and we will use matrix perturbation techniques to give a highly correlated estimation of the expected number of rewiring that are required for a given network to be similar to networks within a random graph ensemble.

The remainder of this thesis is organized as follows. In section 2, we introduce two types of random graph models and two corresponding types of random rewiring processes. In section 3, we use the concept of the Von Neumann entropy of a graph to define a hitting time for non-degree-preserving random rewiring process. In section 4, we apply matrix perturbation methods to derive an estimation of the expected hitting time and present numerical simulation results. In section 5, we provide a concluding discussion.

2 Random Graph Models and Random Rewiring Processes

In this section, we will introduce two types of random graph models (the Erdős-Rényi model [10] and the configuration model [3]), and two corresponding types of random rewiring processes (non-degree-preserving rewiring and degree-preserving rewiring). Each random graph model defines a distribution over graphs on a certain space of graphs, and each random rewiring process defines a discrete-time Markov chain with a finite number of states representing graphs in the corresponding space of graphs. Both the Erdős-Rényi model and the configuration model can be used as null models to study the structural properties of networks, and both non-degree-preserving rewiring process and degree-preserving rewiring process can be used to model networks that change in time, but they differ in the specific structural and dynamical properties of networks that are modeled [1].

2.1 The Erdős-Rényi Model and Non-Degree-Preserving Rewiring

The most basic and classic random graph model is the Erdős-Rényi model. In fact, there are two closely related but slightly different models for generating random graphs that are called the Erdős-Rényi model.

The first model $G(N, M)$ generates a random graph with N vertices and M edges by choosing M edges uniformly at random from the $\binom{N}{2} = N(N-1)/2$ possible edges for a graph with N vertices. In total there are

$\binom{N(N-1)/2}{M}$ different graphs with N vertices and M edges, and all of them have the same probability of being generated by this model.

The second model $G(N,p)$ generates a random graph with N vertices by connecting each pair of vertices independently with equal probability p. Every graph with N vertices is possible to be generated by this model, and the number of edges of a graph with N vertices follows a binomial distribution $B(N(N-1)/2,p)$. Therefore the probability of generating a graph with N vertices and M edges is

$$p^M(1-p)^{N(N-1)/2-M}$$

In this paper, we will focus on $G(N,M)$.

Given a graph with N vertices and M edges, we can define a *non-degree-preserving random rewiring process* by simultaneously choosing an edge to remove from the existing edges uniformly at random and an edge to add from the pairs of vertices that are not connected uniformly at random. This non-degree-preserving rewiring process defines a time-homogenous discrete-time Markov chain on a finite state space with $\binom{N(N-1)/2}{M}$ states, where each state represents a graph with N vertices and M edges. This finite state discrete-time Markov chain is irreducible (it is possible to get to any state from any state) and aperiodic (returns to any state can occur at irregular times), therefore it is positive recurrent (expected return time of any state is finite) and has a unique limiting distribution, and it converges to this limiting distribution regardless of its initial state. In addition, each state i in the Markov chain is connected to $k = M[N(N-1)/2 - M]$ other states, which implies that the Markov chain describes a k-regular graph. By writing down the balance equations

$$\pi_j = \sum_i \pi_i P_{ij} \text{ for any } j$$

where P_{ij} is row stochastic "transition matrix" such that $\sum_j P_{ij} = 1$ and the normalization equation

$$\sum_j \pi_j = 1$$

we can see that the unique positive solution is $\pi_j = 1/\binom{N(N-1)/2}{M}$, which means that the limiting distribution of this Markov chain is the uniform distribution on the space of graphs with N vertices and M edges. We note that this distribution over graphs is the same as the distribution defined by the first Erdős-Rényi model $G(N,M)$.

2.2 The Configuration Model and Degree-Preserving Rewiring

The configuration model $G(N, \vec{k})$ generates a random graph with N vertices and a fixed degree sequence $\vec{k}$ uniformly at random. The degree sequence $\vec{k}$ is defined as an $N \times 1$ vector in which k_i denote the degree of vertex i, and it can be chosen by drawing k_i i.i.d. from some distribution such as the Poisson distribution or as the precise degree sequence of an empirical network.

Given a graph with N vertices and a degree sequence $\vec{k}$, we can define a *degree-preserving random rewiring process* by simultaneously choosing two different edges (a, b) and (c, d) connecting four different vertices to remove from the existing edges uniformly at random and then adding two new edges (a, c) and (b, d) so that the degree sequence is preserved. This degree-preserving rewiring process defines a time-homogenous discrete-time Markov chain on a finite state space, where each state represents a graph with N vertices and the degree sequence $\vec{k}$. This finite state discrete-time Markov chain is also irreducible and aperiodic, therefore it is positive recurrent and has a unique limiting distribution, and it converges to this limiting distribution regardless of its initial state.

3 The Von Neumann Entropy and the Expected Hitting Time of Random Rewiring Processes

The Von Neumann entropy is initially defined as a function of a density matrix in quantum mechanics to measure the departure of a quantum system from its pure state, but recently it has been generalized to be defined as a function of a graph [5, 8, 18]. Given a graph G with N vertices and M edges, the Von Neumann entropy h of a graph G is defined as

$$h = -\operatorname{Tr}(\mathcal{L}_G \log_2 \mathcal{L}_G)$$

where $\mathcal{L}_G = \frac{1}{2M} L$ is the Laplacian matrix associated to the graph G rescaled by $\frac{1}{2M}$. Since $\mathcal{L}_G$ is positive semi-definite and $\operatorname{Tr}(\mathcal{L}_G) = 1$, it can be shown that h can be written in terms of the set $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ of eigenvalues of $\mathcal{L}_G$ as

$$h = -\sum_{i=1}^{N} \lambda_i \log_2(\lambda_i)$$

where by convention we have $0 \log_2(0) = 0$ [8]. The Von Neumann entropy encodes information about the spectral properties of a graph, which in turn is closely related to topological structure of a graph [6]. It can be interpreted as

a measure of regularity of a graph, since in general regular graphs have higher Von Neumann entropy when the number of edges is fixed [18]. Moreover, it is also shown that the Von Neumann entropy are smaller for graphs in which the vertices form a highly connected cluster when the number of edges is fixed [18]. In addition, the Von Neumann entropy can be used to define a notion of "distance" between networks, and one can perform clustering on networks using this notion of distance [8].

As we have previously stated, a random graph model defines a distribution over graphs on a certain space of graphs. Specifically, the Erdős-Rényi random graph model defines a uniform distribution over graphs with N vertices and M edges. This distribution over graphs induces a distribution over the Von Neumann entropy of graphs, and we can run numerical simulations for a large number of times to obtain an empirical distribution over Von Neumann entropy of graphs to approximate the theoretical distribution over the Von Neumann entropy of Erdős-Rényi random graphs of same sizes. For this numerical simulation, we generate $T = 10000$ samples of Erdős-Rényi random networks and plot the estimated probability density function of the Von Neumann entropy of Erdős-Rényi random network samples.

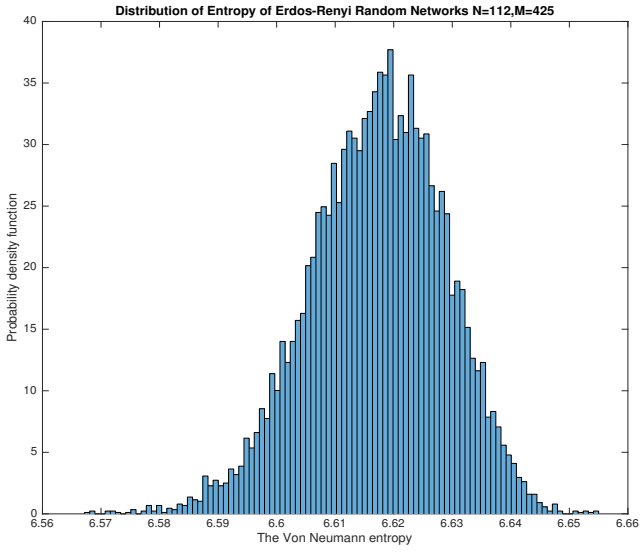

Figure 1: The estimated probability density function of the Von Neumann entropy of Erdős-Rényi random networks

As we can see, in Figure 1, the Von Neumann entropy of Erdős-Rényi random networks with $N = 112$ vertices and $M = 425$ edges are mostly in the range of $[6.57, 6.66]$. However, as we will see in the next numerical simulation, the Von Neumann entropy of an empirical network is usually smaller than almost all the Von Neumann entropy of networks generated by the Erdős-Rényi random graph model. In other words, the Von Neumann entropy of an empirical network typically has a small probability to be drawn from the distribution over Von Neumann entropy of Erdős-Rényi random graphs with the same number of vertices and edges. This indicates that the structure of an empirical network is typically different from the structure of networks generated by the Erdős-Rényi random graph model. To quantify this difference, we consider the expected hitting time of the Markov chain defined by the non-degree-preserving rewiring process. More specifically, for a given network, we are interested in the expected number of rewiring times needed until the Von Neumann entropy of the network undergoing non-degree-preserving rewiring falls within some range of the distribution over Von Neumann entropy of Erdős-Rényi random graphs of same sizes.

Now we present some numerical simulation results to visualize this process. The graph we use is the adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens [17]. Vertices represent the most commonly occurring adjectives and nouns in the book, and edges represent any pair of words that occur in adjacent to one another in the book. The network has $N = 112$ vertices and $M = 425$ edges. For each simulation, we perform non-degree-preserving random rewiring for $K = 1500$ time steps, and we run $T = 100$ independent simulations. The blue, red and yellow curves represent the 5% quantile, median and 95% quantile of the empirical distribution over the Von Neumann entropy of rewired networks, respectively. The red and pink horizontal lines represent the median and 5% quantile, 95% quantile of the empirical distribution over the Von Neumann entropy of Erdős-Rényi random networks of same size and density.

As we can see, in Figure 2, the Von Neumann entropy of the original graph is much smaller than the 5% quantile of the empirical distribution over the Von Neumann entropy of Erdős-Rényi random graphs of same size and density. However, as the number of rewiring times increases, the empirical distribution over the Von Neumann entropy of the rewired graphs is becoming more and more "close" to the the empirical distribution over the Von Neumann entropy of the Erdős-Rényi random graphs. After randomly rewiring the original graph for about 1200 times, the distribution over the Von Neumann entropy of the rewired graph becomes almost identical to that of the Erdős-Rényi random graphs. In other words, by studying the Von

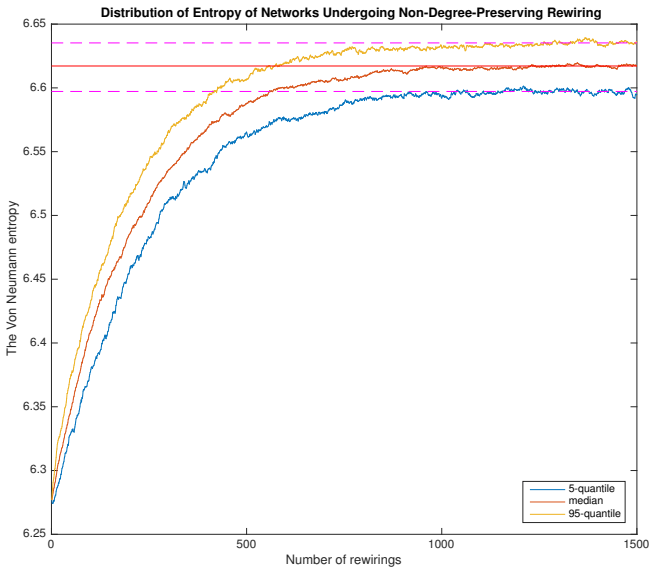

Figure 2: The Von Neumann entropy of networks undergoing random rewiring

Neumann entropy of networks, we numerically observe that the distribution of rewired networks converges to the Erdős-Rényi random graph ensemble. We note that such convergence would be otherwise difficult to observe.

4 A Matrix Perturbation Method to Estimate the Expected Hitting Time

In the previous section, we performed numerical experiments to study the hitting time of non-degree-preserving rewiring process. However, this approach is generally not computationally efficient, especially when the graph is large. Therefore, for the remainder of this research we will seek a way to estimate the expected hitting time given a graph G without performing actual numerical simulations. In other words, we would like to find a function $f(G)$ of a graph that approximates the empirical expected hitting time of the graph G.

To this end, in this section we conduct a perturbation analysis of the Von Neumann entropy to approximate how it changes when an edge is randomly

rewired. To do so, we first develop perturbation analyses for the Laplacian matrix (Sec. 4.1) and its eigenvalues (Sec. 4.2) and eigenvectors (Sec. 4.3). We present a perturbation analysis of the Von Neumann entropy in Sec. 4.4, and introduce a function $f(G)$ to approximate the expected hitting time in Sec. 4.5.

Before continuing, we define some notations and constraints for our perturbation analysis. Given an undirected, unweighted graph G with N vertices and M edges, let A denote the adjacency matrix of G, and D denote the degree matrix of G. The Laplacian matrix L of the graph G is defined as $L = D - A$. Let L denote the Laplacian matrix before rewiring an edge (p, q) to (r, s), and L' denote the Laplacian matrix after rewiring an edge (p, q) to (r, s). Then $L' = L + \Delta L$. We assume the edges are not identical before and after one rewiring, namely $p \neq q \neq r \neq s$.

4.1 The Expected Value of the Difference Between Laplacian Matrices

The process of randomly rewiring an edge (p, q) to (r, s) can be decomposed into two steps. The first step is removing an edge (p, q) from the original graph $G^{(0)}$, resulting in an intermediate graph $G^{(1)}$. The second step is adding an edge (r, s) to the graph $G^{(1)}$, resulting in the rewired graph $G^{(2)}$. Let $L^{(0)}$ denote the Laplacian matrix of the original graph $G^{(0)}$, $L^{(1)}$ denote the Laplacian matrix of the intermediate graph $G^{(1)}$, and $L^{(2)}$ denote the Laplacian matrix of the rewired graph $G^{(2)}$, then we have $L^{(1)} = L^{(0)} + \Delta L^{(0)}$, $L^{(2)} = L^{(1)} + \Delta L^{(1)}$. In terms of our previous notations, we have

$$L = L^{(0)} \ , \ L' = L^{(2)} \ , \ \Delta L = \Delta L^{(0)} + \Delta L^{(1)}$$

4.1.1 Removing an edge

In this section we study how the Laplacian matrix changes due to the removal of an edge (p, q).

As we have stated previously, the elements of the Laplacian matrix L are given by

$$L_{ij} = \begin{cases} D_{ii} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } A_{ij} = A_{ji} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Since removing an edge (p, q) means $A_{pq} = A_{qp}$ changes from 1 to 0, the

elements of $\Delta L_{ij}^{(0)}$ are given by

$$\Delta L_{ij}^{(0)} = \begin{cases} -1 & \text{if } i = j \in \{p, q\} \\ 1 & \text{if } i \in \{p, q\} \text{ and } j \in \{p, q\} \setminus i \\ 0 & \text{otherwise} \end{cases}$$

Hence the expected value of $\Delta L_{ij}^{(0)}$ is given by

$$\mathbf{E}[\Delta L_{ij}^{(0)}] = \begin{cases} P(p = i \text{ or } q = i) \times (-1) & \text{if } i = j \\ P((p = i \text{ and } q = j) \text{ or } (p = j \text{ and } q = i)) \times 1 & \text{if } i \neq j \end{cases}$$

Since there are M edges in total, and we can only remove an edge when $A_{ij} = A_{ji} = 1$, we can write down the probabilities as

$$P(p = i \text{ or } q = i) = \frac{d_i}{M}$$

and

$$P((p = i \text{ and } q = j) \text{ or } (p = j \text{ and } q = i)) = \frac{A_{ij}}{M}$$

After substituting these probabilities into the previous equation, we have

$$\mathbf{E}[\Delta L_{ij}^{(0)}] = \begin{cases} -\frac{d_i}{M} & \text{if } i = j \\ \frac{A_{ij}}{M} & \text{if } i \neq j \end{cases} \tag{1}$$

4.1.2 Adding an edge

In this section we study how the Laplacian matrix changes due to the addition of an edge (r, s).

Since adding an edge (r, s) means $A_{rs} = A_{sr}$ changes from 0 to 1, the elements of $\Delta L_{ij}^{(1)}$ are given by

$$\Delta L_{ij}^{(1)} = \begin{cases} 1 & \text{if } i = j \in \{r, s\} \\ -1 & \text{if } i \in \{r, s\} \text{ and } j \in \{r, s\} \setminus i \\ 0 & \text{otherwise} \end{cases}$$

Hence the expected value of $\Delta L_{ij}^{(1)}$ is given by

$$\mathbf{E}[\Delta L_{ij}^{(1)}] = \begin{cases} P(r = i \text{ or } s = i) \times 1 & \text{if } i = j \\ P((r = i \text{ and } s = j) \text{ or } (r = j \text{ and } s = i)) \times (-1) & \text{if } i \neq j \end{cases}$$

Since there are $\frac{N(N-1)}{2}$ possible edges in total for a graph with N vertices, and we can only add an edge when $A_{ij} = A_{ji} = 0$, and $\{i, j\} \neq \{p, q\}$. Therefore, there are $\frac{N(N-1)}{2} - M$ possible edges to add. Let $R = \frac{N(N-1)}{2} - M$, then we can write down the probabilities as

$$P(r = i \text{ or } s = i) = \frac{N - 1 - d_i}{R}$$

and

$$P((r = i \text{ and } s = j) \text{ or } (r = j \text{ and } s = i)) = \frac{1 - A_{ij}}{R}$$

After substituting these probabilities into the previous equation, we have

$$\mathbf{E}[\Delta L_{ij}^{(1)}] = \begin{cases} \frac{N-1-d_i}{R} & \text{if } i = j \\ -\frac{1-A_{ij}}{R} & \text{if } i \neq j \end{cases} \tag{2}$$

4.1.3 Rewiring an edge

In this section we study how the Laplacian matrix changes due to the rewiring of an edge (p, q) to (r, s). Since rewiring an edge can be decomposed into removing an edge and adding an edge, we can simply combine the results in previous sections and write down a formula for $\mathbf{E}[\Delta L_{ij}]$.

By the linearity of expectation, we have

$$\mathbf{E}[\Delta L] = \mathbf{E}[\Delta L^{(0)}] + \mathbf{E}[\Delta L^{(1)}] \tag{3}$$

We substitute Eqs. 1 and 2 into 3 to obtain

$$\mathbf{E}[\Delta L_{ij}] = \begin{cases} \frac{N-1-d_i}{\frac{N(N-1)}{2}-M} - \frac{d_i}{M} & \text{if } i = j \\ \frac{A_{ij}}{M} - \frac{1-A_{ij}}{\frac{N(N-1)}{2}-M} & \text{if } i \neq j \end{cases} \tag{4}$$

4.1.4 Linear approximation of rewiring multiple times

In previous sections we derived the formula to compute the expected value of the difference between Laplacian matrices before and after rewiring one time. We can use linear approximation to approximate the expected value of the difference between Laplacian matrices before and after rewiring multiple times. Let L_0 denote the original Laplacian matrix before any rewiring, and L_n denote the Laplacian matrix after rewiring n times, then

$$\mathbf{E}[(L_n - L_0)_{ij}] \approx \mathbf{E}[n\Delta L_{ij}] = \begin{cases} n(\frac{N-1-d_i}{\frac{N(N-1)}{2}-M} - \frac{d_i}{M}) & \text{if } i = j \\ n(\frac{A_{ij}}{M} - \frac{1-A_{ij}}{\frac{N(N-1)}{2}-M}) & \text{if } i \neq j \end{cases} \tag{5}$$

Now we present some numerical simulation results to support our theoretical results. The graph we use is the same as the graph used in previous numerical simulations. It has $N = 112$ vertices and $M = 425$ edges.

First, we ran numerical simulations to compute $\mathbf{E}[\Delta L_{ij}]$ by choosing three specific entries (i, j). The first entry $(51, 51)$ is on the diagonal, the second entry $(22, 19)$ is not on the diagonal and corresponds to an edge, and the third entry $(88, 59)$ is not on the diagonal and does not correspond to an edge. For each choice for the number of rewires, n, we computed the mean across $T = 10000$ simulations. We plot the results for this experiment in Figure 3 and Figure 4. The straight lines represent the theoretical prediction according to Eq. (5), and the symbols and error bars represent the mean values and standard errors that we observe, respectively.

Next we ran numerical simulations to compute $||\mathbf{E}[\Delta L]||_F$, where $||\cdot||_F$ is the matrix Frobenius norm. For each choice of the number of rewires, n, we computed the mean across $T = 10000$ simulations, and we plot the results in Figure 5. The straight line represents the theoretical prediction according to Eq. (5), and the X symbols represent the observed values.

4.2 The Expected Value of the Difference Between Eigenvalues of Laplacian Matrices

In this section we will use matrix perturbation theory to derive a relationship between $\mathbf{E}[\Delta L]$, the expected difference between Laplacian matrices, and $\mathbf{E}[\Delta\lambda_i]$, the expected difference between eigenvalues of Laplacian matrices. The central idea is to use a small perturbation to the adjacency matrix of the graph to approximate the process of randomly rewiring an edge of a graph. However, whether such approximation will be sufficiently accurate or not depends on the size and density of the graph, and intuitively we would expect that the approximation will be more accurate for large dense graphs.

Throughout this section we will use ΔL, $\Delta\lambda_i$, and Δv_i to denote the actual difference between the Laplacian matrices, eigenvalues of the Laplacian matrices, and eigenvectors of the Laplacian matrices before and after one rewiring, and we will use δL, $\delta\lambda_i$, and δv_i to denote the difference between the Laplacian matrices, eigenvalues of the Laplacian matrices, and eigenvectors of the Laplacian matrices before and after one small perturbation.

Let λ_i and v_i denote the i-th eigenvalue and the associated unit eigenvector of a Laplacian matrix L, then we have

$$Lv_i = \lambda_i v_i$$

Let λ_i' and v_i' denote the corresponding eigenvalue and the associated unit

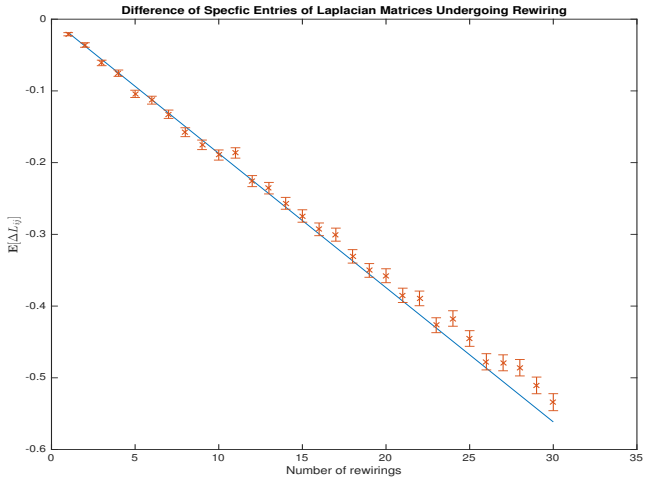

Figure 3: $\mathbf{E}[\Delta L_{ij}]$ for $(i, j) = (51, 51)$

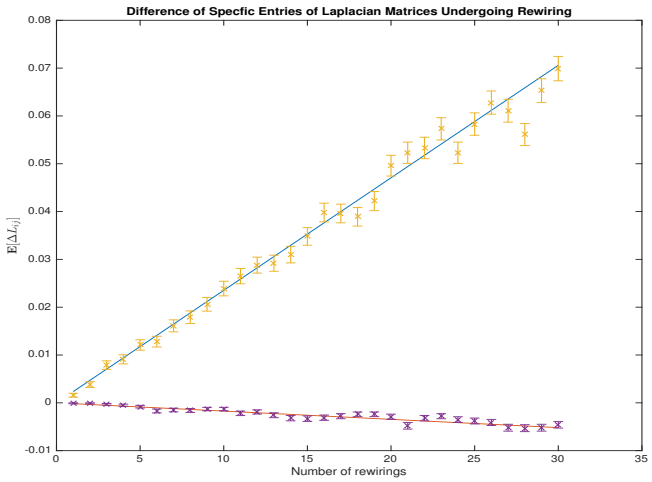

Figure 4: $\mathbf{E}[\Delta L_{ij}]$ for $(i, j) = (22, 19)$ and $(88, 59)$

eigenvector of the Laplacian matrix L' after perturbation, then we also have

$$L'v_i' = \lambda_i' v_i'$$

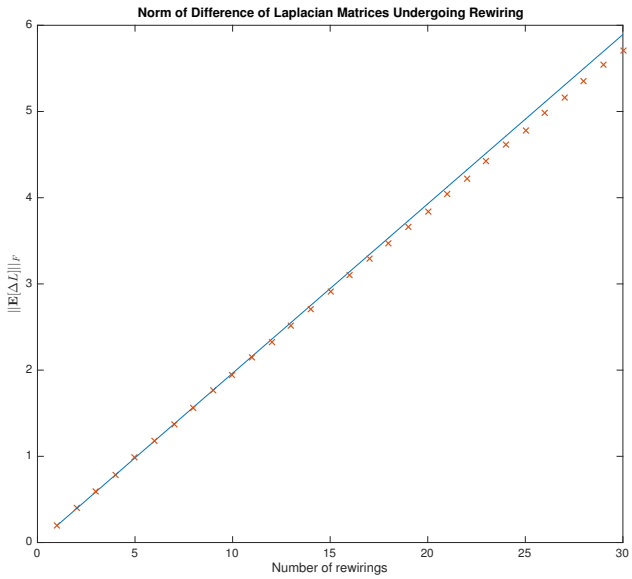

Figure 5: $||\mathbf{E}[\Delta L]||_F$ for different number of rewirings

Let $\delta L = L' - L$, $\delta\lambda_i = \lambda_i' - \lambda_i$, and $\delta v_i = v_i' - v_i$, then the previous equation is equivalent to

$$(L + \delta L)(v_i + \delta v_i) = (\lambda_i + \delta\lambda_i)(v_i + \delta v_i)$$

After expanding both sides of the equation, we have

$$Lv_i + L\delta v_i + \delta L v_i + \delta L \delta v_i = \lambda_i v_i + \lambda_i \delta v_i + \delta\lambda_i v_i + \delta\lambda_i \delta v_i$$

Since $Lv_i = \lambda_i v_i$, and we can discard higher order delta terms $\delta L \delta v_i$ and $\delta\lambda_i \delta v_i$. the previous equation reduces to

$$L\delta v_i + \delta L v_i = \lambda_i \delta v_i + \delta\lambda_i v_i$$

Multiplying v_i^T on both sides, we have

$$v_i^T(L\delta v_i + \delta L v_i) = v_i^T(\lambda_i \delta v_i + \delta\lambda_i v_i) \tag{6}$$

Since L is symmetric, we have $v_i^T L = (Lv_i)^T = (\lambda_i v_i)^T = v_i^T \lambda_i$. Hence

$$v_i^T(L\delta v_i + \delta L v_i) = v_i^T \lambda_i \delta v_i + v_i^T \delta L v_i \tag{7}$$

Since v_i is a unit vector, we have

$$v_i^T(\lambda_i \delta v_i + \delta\lambda_i v_i) = \lambda v_i^T \delta v_i + \delta\lambda_i v_i^T v_i = \lambda_i v_i^T \delta v_i + \delta\lambda_i \tag{8}$$

Plugging equations 7 and 8 into equation 6, we have

$$\delta\lambda_i = v_i^T \delta L v_i$$

Since $\Delta\lambda_i \approx \delta\lambda_i$, $\Delta L \approx \delta L$, we have

$$\Delta\lambda_i \approx v_i^T \Delta L v_i$$

By the linearity of expectation, we have

$$\mathbf{E}[\Delta\lambda_i] \approx \mathbf{E}[v_i^T \Delta L v_i] = v_i^T \mathbf{E}[\Delta L] v_i$$

Now we present some numerical simulation results. The graph we use is the same as the graph used in previous simulations, and it is a relatively small graph with $N = 112$ vertices and $M = 425$ edges. As we have stated before, small perturbation to the adjacency matrix is only an approximation of the process of randomly rewiring an edge of a graph, and there are mainly two ways to account for this fact and test if our estimation becomes more and more accurate as the process of randomly rewiring an edge becomes more and more "close" to a small perturbation. The first way is to run simulations on graphs of different sizes, and ideally we would like to see our estimation becomes more and more accurate as the size and density of the graph becomes larger and larger. However, this method requires computing the eigenvalues of large matrices, which is time-consuming if we want to run simulations for a large number of times. The second way is to introduce a constant ϵ to control the magnitude of the perturbation, and it is the way we will use. Specifically, given the original adjacency matrix $A^{(0)}$, first we randomly rewire an edge and obtain a new adjacency matrix $A^{(1)}$. Then we manually shrink $\Delta A = A^{(1)} - A^{(0)}$ by a factor of ϵ by setting $A^{(2)} = A^{(0)} + \epsilon(A^{(1)} - A^{(0)})$. We consider $A^{(2)}$ as the resulting adjacency matrix after a small perturbation is applied to $A^{(0)}$, and by linearity of expectation we can write down the expected value of the difference between Laplacian matrices as

$$\mathbf{E}[L^{(2)} - L^{(0)}] = \epsilon\mathbf{E}[L^{(1)} - L^{(0)}] = \begin{cases} \epsilon\left(\frac{N-1-d_i}{\frac{N(N-1)}{2}-M} - \frac{d_i}{M}\right) & \text{if } i = j \\ \epsilon\left(\frac{A_{ij}}{M} - \frac{1-A_{ij}}{\frac{N(N-1)}{2}-M}\right) & \text{if } i \neq j \end{cases}$$

If we let $\epsilon = 1$, then the result is the same as randomly rewiring an edge. However, we can also let $\epsilon < 1$, such as $\epsilon = 0.1$, and we would expect our estimation to become more accurate, because a smaller ϵ means a smaller perturbation.

In our numerical simulations, we try three different values of ϵ, namely $\epsilon = 1$, $\epsilon = 0.1$ and $\epsilon = 0.01$, and we compare the results to see if our estimation becomes more accurate as ϵ decreases from $\epsilon = 1$ to $\epsilon = 0.01$. For each simulation, we compute $\Delta\lambda_i$ for every entry i, and we run the simulation for $T = 10000$ times. The blue line represents the theoretical values $\mathbf{E}[\Delta\lambda_i]$, and the symbols and error bars represent the mean values and standard errors of the simulation results.

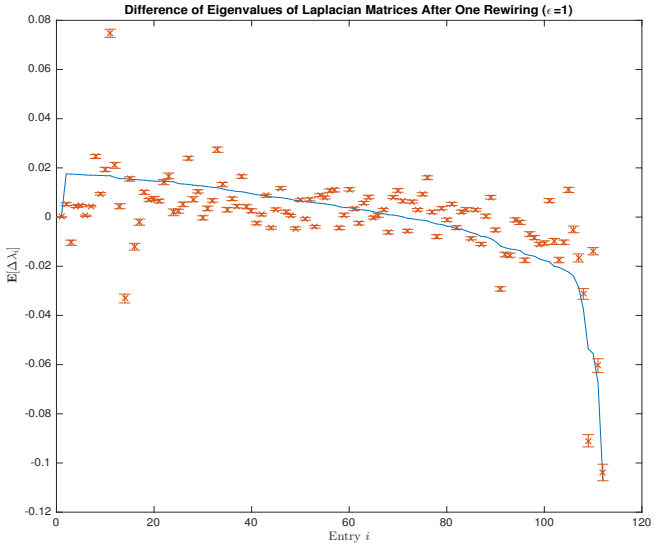

Figure 6: $\mathbf{E}[\Delta\lambda_i]$ for all entry i when $\epsilon = 1$

As we can see, in Figure 6, when $\epsilon = 1$, our estimation of $\mathbf{E}[\Delta\lambda_i]$ based on matrix perturbation method is not very accurate. One possible reason is that the size of the graph is not large enough, and therefore randomly rewiring an edge of the graph cannot be accurately modeled as a "small perturbation". However, as shown in Figure 7 and 8, when $\epsilon = 0.1$ and $\epsilon = 0.01$, our estimation becomes more and more accurate, and this result confirms our expectation that the smaller the perturbation, the more accurate our estimation of $\mathbf{E}[\Delta\lambda_i]$ becomes.

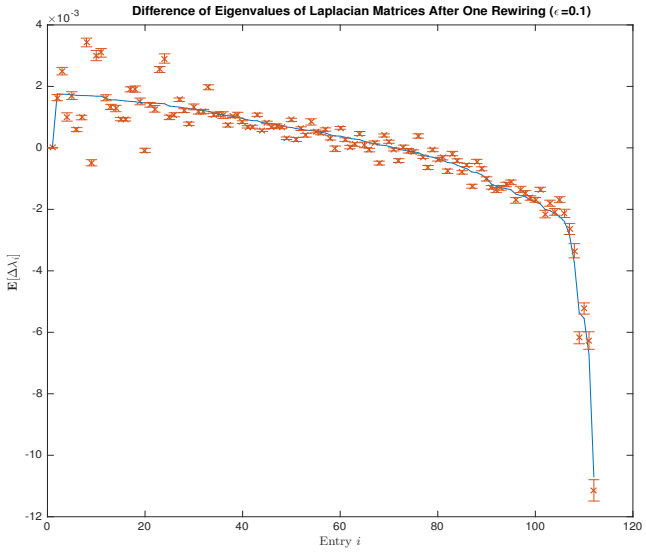

Figure 7: $\mathbf{E}[\Delta\lambda_i]$ for all entry i when $\epsilon = 0.1$

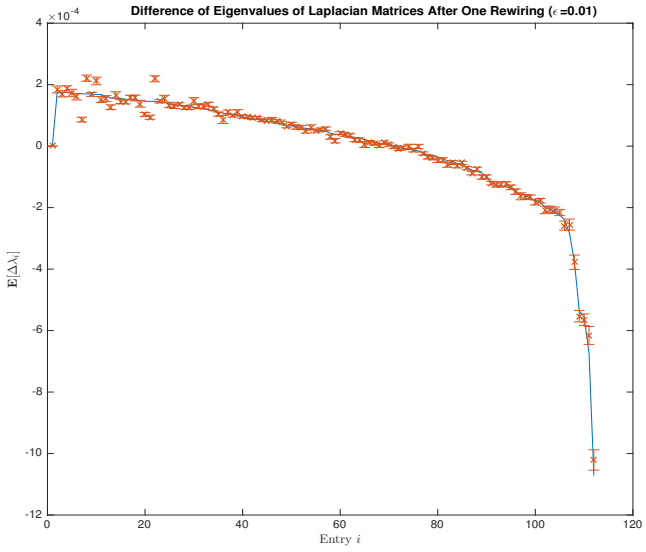

Figure 8: $\mathbf{E}[\Delta\lambda_i]$ for all entry i when $\epsilon = 0.01$

4.3 The Expected Value of the Difference Between Eigenvectors of Laplacian Matrices

In this section we will continue using matrix perturbation theory to derive a relationship between ΔL, the expected value of the difference between Laplacian matrices, $\Delta\lambda_i$, the expected value of the difference between eigenvalues of Laplacian matrices, and Δv_i the expected value of the difference between eigenvectors of Laplacian matrices.

Using the notations defined in the previous section, we have

$$L\delta v_i + \delta L v_i = \lambda_i \delta v_i + \delta\lambda_i v_i$$

Hence

$$(L - \lambda_i I)\delta v_i = (\delta\lambda_i I - \delta L)v_i \tag{9}$$

Now we consider the singular matrix $L - \lambda_i I$. Let $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$ and $V = [v_1 \cdots v_n]$ where each v_i is a column unit eigenvector, then by the spectral theorem, since L is symmetric, we can decompose L as

$$L = V\Lambda V^T = \sum_{j=1}^{n} \lambda_j v_j v_j^T$$

Since $V^T V = VV^T = I$, we have

$$L - \lambda_i I = V(\Lambda - \lambda_i I)V^T = \sum_{j\neq i}(\lambda_j - \lambda_i)v_j v_j^T$$

Define pseudoinverse of the matrix $L - \lambda_i I$ as

$$(L - \lambda_i I)^+ = \sum_{j\neq i} \frac{1}{(\lambda_j - \lambda_i)} v_j v_j^T$$

Then after left multiplying equation 9 by $(L - \lambda_i I)^+$, we have

$$\delta v_i = (L - \lambda_i I)^+(\delta\lambda_i I - \delta L)v_i$$

Since $\Delta v_i \approx \delta v_i$, $\Delta\lambda_i \approx \delta\lambda_i$, $\Delta L \approx \delta L$, we have

$$\Delta v_i \approx (L - \lambda_i I)^+(\Delta\lambda_i I - \Delta L)v_i$$

By the linearity of expectation, we have

$$\begin{aligned}
\mathbf{E}[\Delta v_i] &= (L - \lambda_i I)^+ \mathbf{E}[\Delta\lambda_i I - \Delta L]v_i \\
&= \sum_{j\neq i} \frac{1}{(\lambda_j - \lambda_i)} v_j v_j^T (\mathbf{E}[\Delta\lambda_i] I - \mathbf{E}[\Delta L])v_i \\
&= \sum_{j\neq i} \frac{v_j^T \mathbf{E}[\Delta L] v_i}{(\lambda_i - \lambda_j)} v_j
\end{aligned} \tag{10}$$

Although this result is not used in deriving $\mathbf{E}[\Delta h]$, we note that it can be useful for deriving equations for other measurements of a graph.

4.4 The Expected Value of the Difference Between Von Neumann Entropies

In this section we will use the results in previous sections to drive a relationship between Δh, the difference between the Von Neumann entropy of the graphs, and $\Delta\lambda_i$, the difference between eigenvalues of Laplacian matrices. Note that in previous sections when we derive the formula for the Von Neumann entropy h of a graph G as

$$h = -\sum_{i=1}^{N} \lambda_i \log_2(\lambda_i)$$

the λ_i are the eigenvalues of the rescaled Laplacian matrix $\frac{1}{2M}L$. If we use the notations defined in previous matrix perturbation sections, we would have

$$h = -\sum_{i=1}^{N} \frac{\lambda_i}{2M} \log_2(\frac{\lambda_i}{2M})$$

where λ_i are the eigenvalues of the Laplacian matrix L.

Let h and h' denote the Von Neumann entropy of a graph before and after one small perturbation, and using the notations defined in the previous sections, we have

$$h = -\sum_{i=1}^{N} \frac{\lambda_i}{2M} \log_2(\frac{\lambda_i}{2M})$$

and

$$h' = -\sum_{i=1}^{N} \left(\frac{\lambda_i}{2M} + \frac{\delta\lambda_i}{2M}\right) \log_2 \left(\frac{\lambda_i}{2M} + \frac{\delta\lambda_i}{2M}\right)$$

Using first order Taylor series approximation, we have

$$\left(\frac{\lambda_i}{2M} + \frac{\delta\lambda_i}{2M}\right) \log_2 \left(\frac{\lambda_i}{2M} + \frac{\delta\lambda_i}{2M}\right) \approx \frac{\lambda_i}{2M} \log_2(\frac{\lambda_i}{2M}) + \frac{\delta\lambda_i}{2M} \left(\log_2(\frac{\lambda_i}{2M}) + \frac{1}{\ln(2)}\right)$$

Since $\delta h = h' - h$, we have

$$\delta h \approx -\sum_{i=1}^{N} \frac{\delta\lambda_i}{2M} \left(\log_2(\frac{\lambda_i}{2M}) + \frac{1}{\ln(2)}\right)$$

Since $\Delta\lambda \approx \delta\lambda$, $\delta h \approx \Delta h$, we have

$$\Delta h \approx -\sum_{i=1}^{N} \frac{\Delta\lambda_i}{2M}\left(\log_2(\frac{\lambda_i}{2M}) + \frac{1}{\ln(2)}\right)$$

By the linearity of expectation, we have

$$\mathbf{E}[\Delta h] \approx -\sum_{i=1}^{N} \frac{\mathbf{E}[\Delta\lambda_i]}{2M}\left(\log_2(\frac{\lambda_i}{2M}) + \frac{1}{\ln(2)}\right) \tag{11}$$

Now we present some numerical simulation results. The graph we use is the same as the graph used in previous simulations, and it has $N = 112$ vertices and $M = 425$ edges. Similar to what we did in previous numerical simulations, we use a constant ϵ to control the perturbation magnitude, and we try different values of ϵ to see if our estimations become more and more accurate as ϵ decreases. For each value of ϵ, we run the simulation for $T = 10000$ times. The blue line represents the theoretical values $\mathbf{E}[\Delta h]$, and the red symbols and error bars represent the mean values and standard errors of the simulation results.

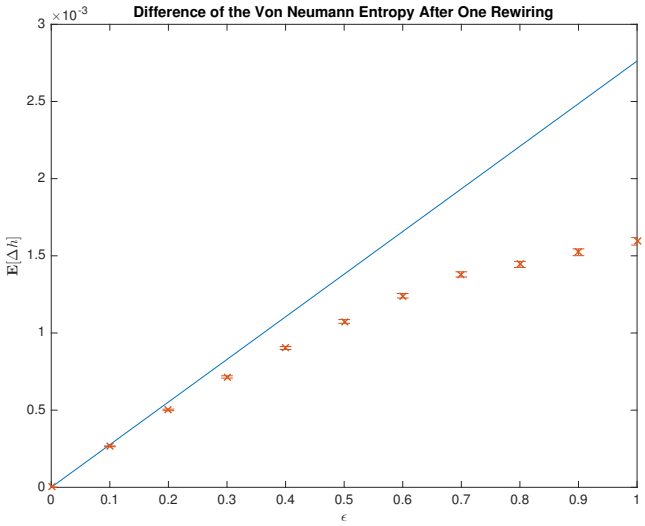

Figure 9: $\mathbf{E}[\Delta h]$ for different values of ϵ

As we can see, in Figure 9, when $\epsilon = 1$, our estimation of $\mathbf{E}[\Delta h]$ based on matrix perturbation method and first order Taylor series approximation

is not very accurate. One possible reason is that the size of the graph is not large enough, and therefore randomly rewiring an edge of the graph cannot be accurately modeled as a "small perturbation". However, when ϵ gradually decreases, our estimation becomes more and more accurate, and this result confirms our expectation that the smaller the perturbation, the more accurate our estimation of $\mathbf{E}[\Delta h]$ becomes.

4.5 Estimation of the Expected Hitting Time

In this section we use the results from previous sections to estimate the expected hitting time. For a given graph with N vertices and M edges, let h denote its Von Neumann entropy, and let $h^{(.05)}$ denote the 5% quantile of the empirical distribution over Von Neumann entropy of Erdős-Rényi random graphs with N vertices and M edges. We are interested in the expected number of non-degree preserving rewiring times needed until the Von Neumann entropy of the rewired graph is no less than $h^{(.05)}$, and one natural estimation would be

$$f(G) = \frac{h^{(.05)} - h}{\mathbf{E}[\Delta h]}$$

The intuition is that the Von Neumann entropy of a graph is expected to increase by $\mathbf{E}[\Delta h]$ after one random rewiring, and if the Von Neumann entropy of a graph is constantly increasing at this rate, then the number of non-degree preserving rewiring times needed until the Von Neumann entropy of the rewired graph is no less than $h^{(.05)}$ is simply the difference between the target Von Neumann entropy and the initial Von Neumann entropy divided by the increasing rate.

However, we would expect that our estimation of the expected hitting time would be smaller than the actual value obtained from numerical simulation, because as we can see from Figure 2, the increasing rate of the Von Neumann entropy $\mathbf{E}[\Delta h]$ of the network undergoing rewiring is decreasing as the number of rewiring times increases.

We perform numerical simulations to compare our estimation of the expected hitting time with the mean of the actual hitting time on multiple empirical networks of different sizes. For each network, we perform $T = 100$ independent simulations, and in each simulation, we start from the empirical network and keep performing non-degree-preserving rewiring until the Von Neumann entropy of the rewired network is no less than $h^{(.05)}$, obtaining 100 different samples of hitting time for each network. We note that we do not actually compute the Von Neumann entropy of the graph every time it is

Network	N	M	Reference
adjnoun	112	425	[17]
airport	500	2980	[7]
neural	297	2148	[22]
metabolic	453	2025	[9]
dolphins	62	159	[15]
lesmis	77	254	[14]
jazz	198	2742	[12]
email	1133	5451	[13]

Table 1: Basic summary statistics for the networks that we use for this numerical simulation. The number of vertices is denoted by N, and the number of edges is denoted by M.

rewired, because computing the Von Neumann entropy of a graph requires computing all the eigenvalues of a graph, and the runtime of performing this operation for millions of times is too large. Instead, we compute the Von Neumann entropy of the rewiring graph for every $10\lceil N/100)\rceil$ rewirings. This trick greatly reduces the runtime of the numerical simulation, because the number of times of computing all the eigenvalues of a graph is reduced by a factor of $10\lceil N/100)\rceil$. Although it also increases the variance of the sample mean estimator, the standard error of the sample mean divided by sample mean is no larger than 0.05, which indicates that the sample mean should be close enough to the true expected hitting time.

As we can see from Figure 10, the dotted least squares regression line indicates that there is a linear relationship between our estimation and the actual expected hitting time obtained from our numerical simulation. The correlation coefficient is 0.9941. This numerical simulation demonstrates that our estimation of the expected hitting time is strongly correlated with the actual expected hitting time. However, the runtime of computing our estimation of the expected hitting time $f(G) = \frac{h^{(.05)}-h}{\mathbf{E}[\Delta h]}$ is much smaller than computing the expected hitting time by performing a large number independent random rewiring simulations, each rewiring for a large number of times, because the only extra term that needs to be computed by our estimation is $\mathbf{E}[\Delta h]$, which can be computed directly for a given graph G with N nodes and M edges using our previous results.

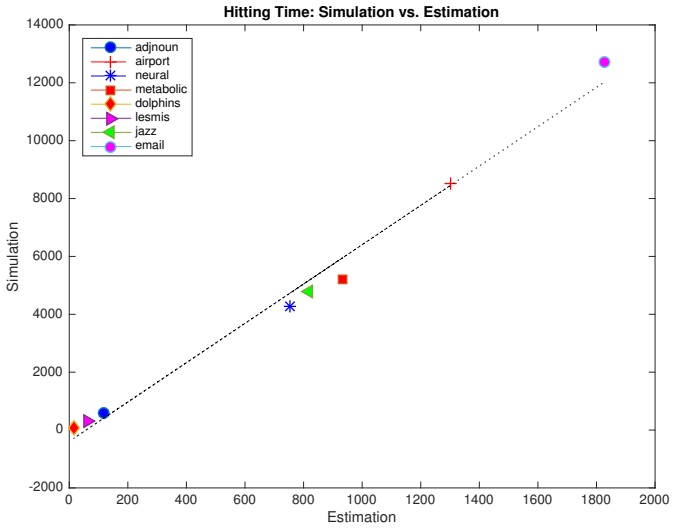

Figure 10: Comparison of the sample mean of actual hitting times and our estimation $\frac{h^{(.05)}-h}{\mathbf{E}[\Delta h]}$ for networks of different sizes

5 Conclusion

In this paper, we used the concept of the Von Neumann entropy of a graph to define a hitting time for non-degree-preserving random rewiring process, which is the number of rewiring times needed until the Von Neumann entropy for the rewired graph is larger than 5% quantile of the distribution over the Von Neumann entropy of Erdős-Rényi random networks. Since the purpose of defining the hitting time is to quantify the difference between a given network and networks generated by the Erdős-Rényi random graph model with the same number of vertices and edges, we do not really need to find an approximation of the actual value of the expected hitting time. Instead, we only need to give an estimation that is positively correlated with the expected hitting time, and we used

$$f(G) = \frac{h^{(.05)} - h}{\mathbf{E}[\Delta h]}$$

where h and $h^{(.05)}$ are respectively the Von Neumann entropies of graph G and the 5% quantile for the ensemble of Erdős-Rényi random graphs, and $E[\Delta h]$ is the expected change of the Von Neumann entropy due to one ran-

dom rewiring as given by Eq. (11). The intuition behind our estimation $f(G)$ is a simple linear approximation, and we expect that a more accurate estimation can be derived by using more complicated models such as an exponential model to model the trajectories of the Von Neumann entropy of networks undergoing rewiring. Experimental results show that our estimation is highly correlated with the sample mean of the actual hitting time obtained from numerical simulation, and the advantage of using our estimation is that it can be computed directly without performing a large number of simulations and taking the sample mean of the actual hitting times.

The central part of the estimation is $\mathbf{E}[\Delta h]$, which is obtained by using a small perturbation to the adjacency matrix of the graph to approximate one random rewiring of a graph, and then applying matrix perturbation techniques to derive equations for $\mathbf{E}[\Delta L]$, $\mathbf{E}[\Delta \lambda_i]$, and $\mathbf{E}[\Delta h]$. However, whether such approximation would be sufficiently accurate or not depends on the size and density of the graph, and as we have seen in our numerical simulations, if the graph is relatively small, then matrix perturbation approximation might not be accurate. However, we would expect that as the size and density of the graph increases, matrix perturbation approximation would become more and more accurate.

Bibliography

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] András Békéssy, P Bekessy, and János Komlós. Asymptotic enumeration of regular matrices. *Stud. Sci. Math. Hungar*, 7:343–353, 1972.

[4] Béla Bollobás. *Modern Graph Theory*, volume 184. Springer Science & Business Media, 2013.

[5] Samuel L Braunstein, Sibasish Ghosh, and Simone Severini. The laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. *Annals of Combinatorics*, 10(3):291–317, 2006.

[6] Fan RK Chung. *Spectral Graph Theory*, volume 92. American Mathematical Soc., 1997.

[7] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3(4):276–282, 2007.

[8] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature Communications*, 6, 2015.

[9] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, 2005.

[10] Paul Erd6s and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

[11] Miroslav Fiedler. Laplacian of graphs and algebraic connectivity. *Banach Center Publications*, 25(1):57–70, 1989.

[12] Pablo M Gleiser and Leon Danon. Community structure in jazz. *Advances in Complex Systems*, 6(04):565–573, 2003.

[13] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68(6):065103, 2003.

[14] Donald Ervin Knuth, Donald Ervin Knuth, and Donald Ervin Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*, volume 37. Addison-Wesley Reading, 1993.

[15] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[16] Stephen B Maurer. Matrix generalizations of some theorems on trees, cycles and cocycles in graphs. *SIAM Journal on Applied Mathematics*, 30(1):143–148, 1976.

[17] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[18] Filippo Passerini and Simone Severini. The von neumann entropy of networks. *Available at SSRN 1382662*, 2008.

[19] Louis M Pecora and Thomas L Carroll. Master stability functions for synchronized coupled systems. *Physical Review Letters*, 80(10):2109, 1998.

[20] Per Sebastian Skardal, Dane Taylor, and Jie Sun. Optimal synchronization of complex networks. *Physical Review Letters*, 113(14):144101, 2014.

[21] Gilbert W. Stewart and Ji guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[22] Duncan J Watts and Steven H Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393(6684):440–442, 1998.