Stability-optimization algorithms for the detection of community structure in networks

Marianne McKenzie

Keble College

University of Oxford

A thesis submitted for the degree of

MSc in Mathematics and the Foundations of Computer Science

September 2012

Abstract

Community detection is an active field of research across the sciences due to its potential value in practical applications. Many different paradigms are devoted to the development of algorithmic solutions which reveal these mesoscopic, modular subnetworks. Much of the work in the field has been devoted to detection using the structure of the network. However, an alternate approach to the problem is to use dynamics on the network to reveal the underlying mesoscopic structures.

A new quality function known as *stability* uses the normalize Laplacian dynamics to reveal the underlying community structure. Stability can be generalized to reveal ties to the popular quality function *modularity.* It can also be utilized to perform community detection on a set of structurally complex network known as *multislice networks*, which are composed of network slices coupled together.

We use different normalize Laplacian dynamics with a teleportation step to define a new form of stability from the PageRank algorithm. A careful examination of the teleportation step for these dynamics reveals the best form of teleportation to use for stability. The introduction of the PageRank form of stability allows for stability to be defined for directed single, static networks or directed multislice networks where directionality is observed. A stability-optimization algorithm can then be employted to identify community structures. A few real world example are examined using the popular Zacharys Karate Club and a some artificially-constructed, simple benchmark networks.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Mason Porter for his guidance, encouragement, and patience. His willingness to sacrifice his time and talents to guide my work was truly appreciated. More importantly, he proved to be an invaluable ally when overcoming the difficulties that my exploration uncovered. I am also grateful to Professors Colin McDairmid and Oliver Riordan for their lectures in Graph Theory and Combinatorics that have proven to be invaluable sources of knowledge and inspiration in my work. I am also indebted to my Supervisors Alexander Scott and Stephen Cameron and my Senior Tutor Marc Brodie for their guidance through the intricate halls and traditions of the University of Oxford. Finally, I would not have been granted this opportunity if not for the support of my family, friends, and course mates. A special thank you to Pat and Daniese Mc̲Kenzie for instilling in me a love knowledge and to Michael Timbs, the Brothers Mc̲Kenzie, and Sharon Mc̲Kenzie for unending encouragement, support, and inspiration.

Contents

Chapter 1

Background

The study of networks has received a large amount of attention in the scientific community as networks provide a way to model many systems of interest, such as social networks, the Internet, computer networks, and metabolic and regulatory networks [1, 2, 3]. The network models of these systems have been found to divide naturally into mesoscopic modular structures known as *communities* [1, 2]. However, detecting and characterizing these community structures is an outstanding problem and continues to receive attention in both the social and physical sciences. The problem of community detection is complicated by the complexities of the real world systems modeled, such as time dependent systems or systems containing objects with multi-faceted connections between them. Currently, networks that model these more structurally complex systems are not fully encompassed by the available algorithms. However, despite the complex structures that can be constructed and modeled, the basic structure of a network is easily defined and understood.

1.1 The Basics

Networks are composed of two primary building blocks: connecting lines known as *edges*, *links*, or *arcs* and connecting points for the edges known as *nodes* or *vertices.* For this work, we will use the terms edges and nodes. An edge in a network can have an associated *weight*, a value which designates the strength of the interaction the edge represents. Similarly, each node in a network has an associated *strength* given by the sum of the weights of all adjacent edges. Nodes also have a *degree* given by the number of adjacent edges.

The above definitions are for *undirected networks*, where an edge represents a pairwise binary connection which is either present or not. However, networks often have *directed edges*, in which the edges are oriented to point from one vertex to

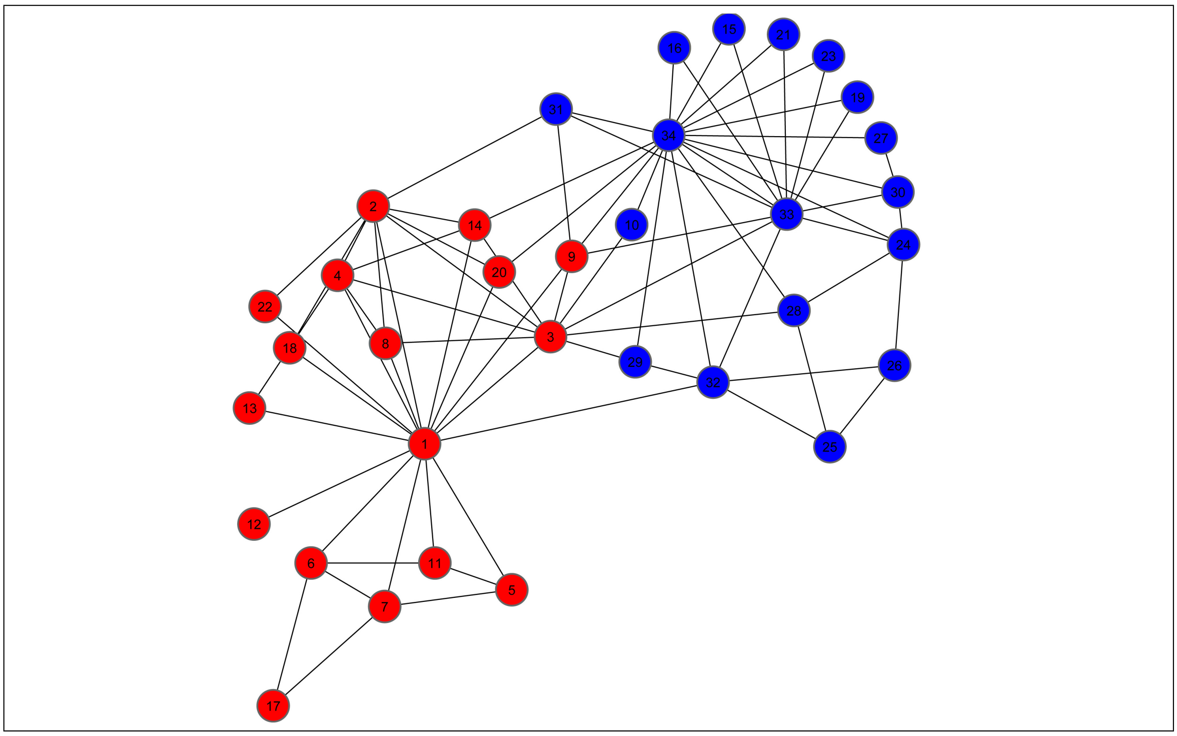

Figure 1.1: The Zacharys Karate Club network is a famous network depicting members of a university karate club in the 1970s [4]. The members are represented as nodes and their social connections are represented by edges. A conflict within the club led to a split, with the post-split affiliations designated by color. The ability for a community detection algorithm to reproduce these affiliations is a basic benchmark by which community-detection algorithms can be tested. This figure was generated using Cytoscape 2.7.0.

another. Using this idea of orientation, we define an *in-degree* as the number of adjacent edges that are oriented so that they end at the node, and an *out-degree* as the number of adjacent edges beginning at the node. *In-strength* and *out-strength* are defined similarly.

The simplicity and versatility of the building blocks of networks make them popular choice for modeling artificial and real world systems by representing objects in the system as nodes and connections between objects as edges.These definitions reveal that a network shares some commonality with a graph, which may provide some intuition for the definitions and basic properties from analogous problems in graph theory. However, the term network and graph are not always synonymous as networks allow for more complicated structures. An illustration of this fact can be found in biological networks, where biological processes may occur between more than two objects, requiring edges to be between more than two nodes in a network [5].

Networks can often be decomposed into mesoscopic structures known as *commu-*

nities. Such structures help identify structural and topological features of large-scale networks and sometimes can represent functional units [6]. A large body of literature has been devoted to exploring algorithms for detection of community structures. However, no definitive definition for a community exists. This makes formulating a sharp analytical definition of community impossible, so instead we use a general idea of a group of nodes that are densely connected to each other and sparsely connected to other densely connected groups of nodes [7, 1, 8, 9, 2, 6]. The range of applications and the variety of ways of defining a community have resulted in a large number of community detection algorithms [1, 2].

1.2 Community Detection in Networks

The diverse notions of what defines a community has resulted in community detection algorithms which use a variety of techniques to discover community structures. Some of the most popular approaches rely on identifying communities through hierarchical clustering or partition optimization. A hierarchical clustering approach uses clusters of nodes to model communities and typically divides or merges these community clusters by considering an appropriately defined measure such as a distance or similarity between them. In contrast, partition optimization approaches repeatedly imposes the grouping of nodes into network partitions according to a given criterion until an optimal partition is found. In both kinds of approaches, a decision must be made whether to focus on *local network properties* or *global network properties.* Local properties are defined by the topology and structure of a neighborhood around a node or a subgraph regardless of the structure of the rest of the network. For example, identifying complete subgraphs of k-nodes, known as *k-cliques* requires examining local properties. In contrast, global properties consider the entire network at once, and may be defined from large-scale statistical properties of the network. An example of global approach is presented in the following subsection on modularity optimization.

1.2.1 Modularity Optimization

A popular community-detection algorithm is *modularity optimization.* Modularity is a quality function which was developed by Newman [7] to measure the quality, or goodness, of a network partitioning.

Definition 1. *Network Partition* For a network A of size N, a partition is an assignment function $\mathcal{P} : 1, ..., N \to 1, ..., p$ where $\mathcal{P}(i)$ represents the class assignment for node i.

For modularity, the partitions represent communities within a network. Therefore, finding a network partition with high modularity means a strong community structure has been found. Modularity measures if edges are more abundant within a community than one would expect if edges were randomly distributed.

Q = (fraction of edges within communities) − (expected fraction of edges with random distribution).

The idea of random edge distribution is given by a *null model*, a random network defined in such a way that it shares some structural features with the original network [2].

Definition 2. Null Model The null model is used as a baseline for comparison to a network. Consider a network A with N nodes. The null model with respect to A is a random network P on N nodes with probability P_{ij} that an edge (i, j) will occur. The restrictions for generating P are problem dependent.

Using this idea, an expression for modularity can be defined.

Definition 3. *Modularity* Consider an weighted, undirected network A with network partition $\mathcal{P}$. Let A_{ij} be a standard adjaceny matrix which represents the weight of the edge between nodes i and j. Since the network is undirected, it is true that $A_{ij} = A_{ji}$.The strength of node i is $k_i \equiv \sum_j A_{ij}$ and the total weight of the network is $m \equiv \sum_{i,j} \frac{A_{ij}}{2}$. The modularity function Q is

$$Q = \frac{1}{2m} \sum_{C \in \mathcal{P}} \sum_{i,j \in C} [A_{ij} - P_{ij}], \tag{1.1}$$

where $i,\ j \in C$ is a summation over pairs of nodes i and j belonging to the same community C of $\mathcal{P}$ and P_{ij} is an appropriately defined null model.

One of the most popular methods for modularity was introduced by Newman and Girvan [7] and used the null model $P_{ij} = \frac{k_i k_j}{2m}$ so that, although the edges are randomly placed, the degree distribution is the same as the original network A.

Definition 4. *Newman-Girvan Modularity* Consider an weighted, undirected network A with network partition $\mathcal{P}$. The Newman-Girvan modularity function Q_{NG} is

$$Q_{NG} = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}). \tag{1.2}$$

Despite its popularity, modularity suffers from a few drawbacks. It is a known NP-Hard problem, meaning it is impractical to test all partitions to determine the maximal modularity, and thus best solution [10]. Therefore, in general modularity optimization algorithms are used to approximate the best solution. It also suffers from a *resolution limit*, meaning that it can not distinguish communities below a certain size which may result in incorrect partitions [11]. This resolution limit has been circumvented in a few ways [2] and we will present one such approach from Lambiotte [9] and a more thorough exploration of the resolution limit in Chapter 2.

1.2.2 Synchronization

Thus far, we have considered community detection algorithms which rely directly upon a network's structure to identify communities. However, it is also true that the information about mesoscopic structure can be revealed by the behavior of dynamics on the network [9]. Arenas et al. [12] sought to employ this idea to identify a communities modular topology using *synchronization*. A brief outline of this method is useful for motivation the use of dynamics on a network to identify community structures. Synchronization phenomena may occur in systems of interacting units and has been studied across many disciplines relating to nature, society and technology. One of the first and most successful models for understanding synchronization phenomena was introduced by Kuramoto [13] who utilized a model of phase oscillators coupled through the sine of their phase differences. The Kuramoto model consists of a population of N coupled phase oscillators where the phase of the i-th unit, $\theta_i(t)$, evolves according to the following dynamics

$$\frac{d\theta_i}{dt} = \omega_i + \sum_j K_{ij} \sin(\theta_j - \theta_i), i = 1, ..., N, \tag{1.3}$$

where ω_i is the natural frequency, and K_{ij} expresses the coupling between units of the system. This model displays a large variety of synchronization patterns and can be adapted to many different contexts [12].

It is typically true that densely interconnected sets of oscillators synchronize more easily than those with sparse connections [12]. This scenario suggests that highl-interconnected units forming local clusters will synchronize first and that larger and larger structures also will follow until the final state, where the whole population should have the same phase. If a community structure exists on the network, then it is reasonable to expect that this will occur at different time scales [12]. Thus, different topological structures, which we presume represent communities, are revealed by following the dynamical route towards the global attractor. In Chapter 2 and 3, we will explore in detail another dynamical approach using a normalized Laplacian dynamics.This approach will be used to define stability, a new quality function. We will generalize this formalization to include several classes of networks including multislice networks, which have multiple associated adjacency matrices coupled together. In Chapter 4, we will use the approach given in Chapter 2 to define an alternate form of stability based on the PageRank algorithm and use it to explore community detection for networks with directed edges.

Chapter 2

Stability-optimization

Much of the previous work done in community detection has focused upon structural properties to discover communities [1, 9]. As we saw with the Kuramoto model, dynamics on a networks also provide empirical results by which mesoscopic structures can be uncovered. Another approach that uses dynamics on a network to reveal the underlying structure, rather than examining topological or structural properties directly, is *stability*. Stability was introduced by Lambiotte et al. [9] as a quality function defined in terms of the statistical properties of a dynamical process taking place on the network. Using this dynamics based approach standard modularity functions, such as Newman-Girvan, can be re-derived [9]. Furthermore, stability is quite robust and can be generalized to encompass several specialized forms of networks, a topic which we will explore in Chapter 3.

2.1 Laplacian dynamics and stability

Stability is a quality function introduced by Lambiotte et al. [9] which uses stochastic processes applied on a network to determine the quality of a partition. One of the motivating concepts behind stability is that the flow of probability on a network will remain trapped within a community for a long period of time [14]. We will express the stability, in terms of the quality of a partition $\mathcal{P}$, as an autocovariance function of an ergodic Markov process $\mathcal{M}(t)$ on the network. We define the stability $R_{\mathcal{M}(t)}$ as follows

$$R_{\mathcal{M}(t)} = \sum_{C \in \mathcal{P}} P(C,t) - P(C,\infty), \tag{2.1}$$

where $P(C,t)$ is the probability for a walker to be within a community C both initially and at time t. Due to the ergodicity of the dynamics, the second expression is also the the probability for two independent walker to be in C as the initial condition is lost at infinity. We can then express this autocovariance function in a way which can be easily understood

$R_{\mathcal{M}(t)}$ = (Probability for a random walker to be in the same community initially and at time t) - (Probability for two independent random walkers to be in the same community).

The idea captured by the right hand side is the null model for this approach. The selection of an appropriate null model is crucial as they serve as the baseline by which the goodness of a partition is measured [15]. Additionally, the null model must fit the network model well so that it can accurately reproduce the appropriate network structures and laws that are present in the original network.

2.1.1 Normalized laplacian dynamics

Lambioette et al. use normalized Laplacian dynamics as the model for an unbiased random walker on the network. The discrete-time expression for the density of random walkers on a node i of an undirected, weighted network evolves according to

$$p_{i,n+1} = \sum_j \frac{1}{k_j} A_{ij} p_{j,n}, \tag{2.2}$$

where A_{ij} is a standard adjacency matrix and $k_i = k_i^{out} = \sum_i A_{ij}$ the out-strength of node i. We can define a continuous-time process for the density of walkers on a node i by assuming that there are independent, identical homogeneous Poisson processes defined on each node on the graph [9],

$$\dot{p}_i = \sum_j \frac{1}{k_j} A_{ij} p_j - pi. \tag{2.3}$$

Note that the continuous-time density equation is driven by the $A_{ij}/k_j - \delta_{ij}$, which is the negative of the Laplacian operator. The steady states for both the discrete-time and continuous-time dynamics is given by $p_j^* = \frac{k_j}{2m}$, where $2m = \sum_i k_i = \sum_{ij} A_{ij}$ is the total strength in the network.

2.1.2 Stability

First examining the continuous-time normalized Laplacian, we arrive at the following expression for the stability of a partition for an undirected, weighted network.

$$R(t) = \sum_{c} \sum_{i,j \in C} [(e^{tL})_{ij} \frac{k_j}{2m} - \frac{k_i}{2m} \frac{k_j}{2m}] \tag{2.4}$$

where $L_{ij} = \delta_{ij} - A_{ij}/k_j$ is the negative of the Laplacian operator. The first expression is the probability that a walker is in a community C at two successive time steps and the second is the multiplication of the probabilities that two independent walker are both contained in the community.

Note that 2.4 is a time-dependent expression. Therefore, different time scales result in different measures for stability. It is of note that this means a single optimal partitioning may not exist, but rather there will be a sequence of optimal partitions. Lambiotte et al.[9] show that time actually acts as a resolution parameter.

2.2 Time as a resolution parameter

It was discovered in 2.4 that stability possessed a time dependency. If we linearize the exponential expression, we can derive the following form of stability:

$$R(t) = \sum_{c} \frac{1}{2m} \sum_{i,j \in C} \left[(A_{ij} t - \frac{k_i k_j}{2m} \right] . \tag{2.5}$$

In this form, it is clear that at time $t = 1$, stability reduces to a familiar expression, Newman-Girvan modularity.

$$R(1) = \sum_{c} \sum_{i,j \in C} [\frac{A_{ij}}{k_j} \frac{k_j}{2m} - \frac{k_i}{2m} \frac{k_j}{2m}] = Q_{NG}. \tag{2.6}$$

Lambiotte et al. [9] prove that time can be treated as an intrinsic resolution parameter. If we let $\gamma = \frac{1}{t}$ then stability with resolution parameter γ is

$$R(t) = \sum_{c} \frac{1}{2m} \sum_{i,j \in C} \left[(A_{ij} - \gamma \frac{k_i k_j}{2m} \right] . \tag{2.7}$$

However, we must ask if dividing by t effects the optima for a specified t. Lambiotte et al. [9] prove that this does not occur. Consider when t = 0, then $R_(0) = 1 - sum_c \frac{1}{2m} \sum_{i,j \in C} \frac{k_i k_j}{2m}$. By attempting to maximize this expression, we discover that the optimal partition will be when each node is contained in its own network. In contrast, if $t \to \infty$, then eigenvalue decomposition can be used to prove that $R(t)$ is typically maximized by a partition into two communities. However, the details of the full proof if this fact are not particularly instructive for our understanding of stability an thus has been omitted.

Optimizing the stability function at each time step gives us an optimal stable partition, where the resolution parameter allows the characteristic size of the community to be adjusted. This new form quality function, therefore, shows itself to be a versatile while still being generalizable to standard Newman-Girvan modularity. However, we will show in the following chapter that stability can be generalized to several classes of models, including some with complicated structural properties that have thus far been unsuccessfully handled by community detection algorithms.

Chapter 3

Multislice networks

Thus far the focus for community detection methods have been on single, static networks. However, sometimes the real-world systems possess structures that can not be accurately modeled on such networks. For example, a social network may have social connections which evolve over time or the connections between individuals may have different meaning, such as business relationship versus a friendship. The networks examined thus far do not offer structures which can accurately reflect such complicated connections on a single network. Several types of network have been proposed which can model more complicated systems, such as:

- *Dynamic networks* that have nodes or edges which change over time.
- *Hierarchical or Multiscale Networks* where the hierarchical structures allow for groups of nodes to be repeatedly divide into smaller sub-networks and such divisions can be made over multiple scales.
- *Multiplex networks* that have multiple edge between nodes such that each edge represents a different type of connection.

Recently, Mucha et al. [8] propose a new form of network model which is capable of modeling all of these forms.

Definition 5. Multislice Networks A network with s slices. Each slice has an associated adjacency matrix such that A_{ijs} represents inter-slice connection between nodes (i, s) and (j, r). Additionally, intra-slice couplings are represented by a matrix C such that C_{jrs} connects a node j to itself in two slices, specifically it couples (j, s) to (j, r). We will introduce ω as a parameter to control coupling between slices. An example of a multislice network is represented in 3.1.

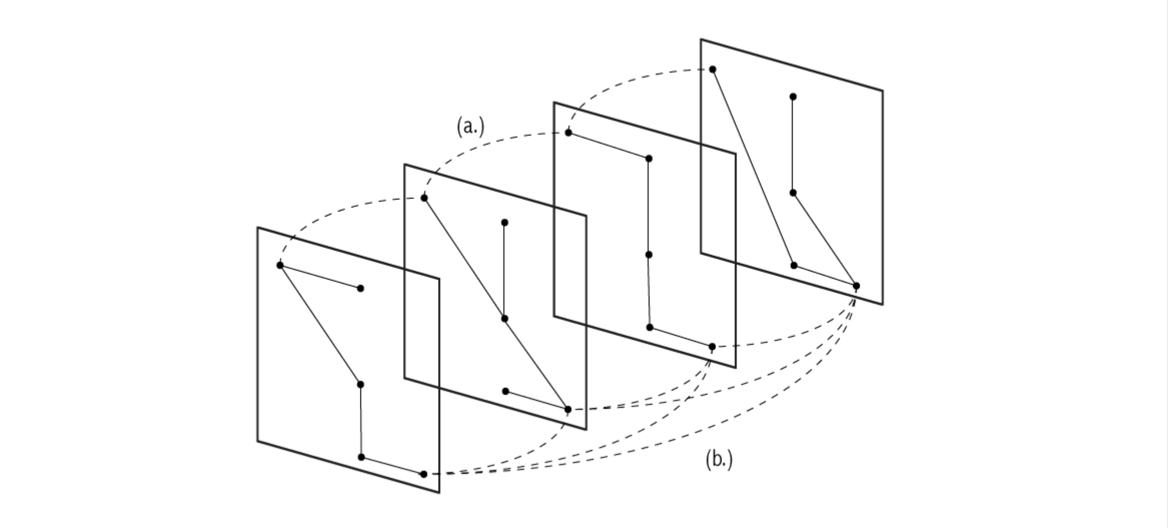

Figure 3.1: A multislice network with four slices. Within each slice, the adjacencies A_{ijs} encode *intra-slice* connections represented by the internal solid lines. The dashed lines between slices are the *inter-slice* connections that are encoded by C_{jrs}. Inter-slice connections connect a node j in slice s with itself in slice r. The connections shown are (a.) *Ordered* connections such that a node is couple only with neighboring slices; and (b.) *All-to-all* connections such that a node is coupled with itself any every other slice. Alternate forms of couplings may also be used.

Multislice networks are versatile. They can allow for dynamic networks by modeling a different time-step on each slice or multiplex networks where a slice could contain only one type of connection. The ability to accurately model hierarchical networks is less obvious. However, the following discussion in 3.1.2 will show how stability defined for multislice networks can allow for different resolution parameters to used for different slices.

For the moment, we will restrict our considerations to undirected, unipartite network slices so that $A_{ijs} = A_{jis}$. We will do the same for couplings so that $C_{jrs} = C_{jsr}$.

Definition 6. Multislice Strength The strength for each node consists of two forms of strength.The slice strength $k_{js} = \sum_i A_{ijs}$ and the coupling strength $c_{js} = \sum_r C_{jsr}$. The total multislice strength $\kappa_{jr} = c_{jr} + k_{jr}$

Directed edges may be introduce to the multislice model in much the same way as they would for single, unipartite networks.

3.1 Stability for multislice networks

Mucha et al. [8] were able to successfully able to extend stability to multislice network. In order to do so, three key generalizations have to be made. Although the stability equation for multislice networks can be understood without full exposure to these generalizations, it is instructive to see them first so that the reasons for the multislice stability formulation are clear. We will show that the three generalizations made are capable of recovering the appropriate null models for a specific category of network.

3.1.1 Generalizations and null models for bipartite, signed, and directed networks

Consider that multislice networks have two types of connections, intra-slice and inter-slice, and the type of connection traversed matters when considering a random walker on the network. Therefore, when considering the probability of a random walker remaining within the same community after time t in the statistically steady state it will be necessary to restrict the independent contribution to be conditional on the type of connection necessary to step between nodes. So, we can replace the independent contribution $p_i^* p_j^*$ in 2.4 with a conditional independent contribution $\rho^*_{i|j}\rho^*_j$ which takes into account the type of edge being traversed. This is done through $\rho^*_{i|j}$ a conditional probability of jumping from node$i$ from node j along a specific edge type. The edge type that is allowed at a given step will be specified depending on the category of networks.

Bipartite networks are a particular category of graphs which would employ such a generalization as they have two types of nodes based on which side of the bipartite partition the node falls. By definition, every edge must connect a node of one kind to a node of the other kind. For an undirected, bipartite network the adjacency matrix A in the Laplacian operator L_{ij} defined previously will have a specific form due to the bipartite partition. This results in the same steady state solution for all nodes, $p^* = k_j/2m$ where $2m$ is defined as before. The bipartite conditional probability $\rho_{is|jr}$ of visiting node (i, s) is conditional on whether there is a structure that will allow the traversal from (j, r) to (i, s). Thus, the conditional probability is $\rho_{is|jr} = \frac{b_{ij}k_i}{m}$ where b_{ij} as a binary indicator function for bipartiteness, meaning it is 1 if nodes i and j are of different types and 0 otherwise. Note that the probability of stepping to a given node i is now conditional on the information of the partition assignment, doubling the probability, thus the denominator of $\rho_{is|jr}$ is m.

We can now state the stability for bipartite networks.

$$R_{bipartite} = \frac{1}{2m}\sum_{ij}\{\left(A_{ij} - \gamma b_{ij}\frac{k_i k_j}{m}\right)\delta(g_i, g_j). \tag{3.1}$$

This is the generalization of the Barber bipartite null model [16] when the resolution parameter γ is incorporated.

Next, consider that dynamics on multislice networks must allow for motion along both types of edges. So, we will generalize the Laplacian dynamics to allow for multiple types of connections. We will do so by first considering directed networks.

A directed networks have two types of strength, $k_{in} = \sum_j A_{ij}$ and $k_{out} = \sum_i A_{ij}$. We will define total strength to be the sum of the two strengths. We will consider incoming and outgoing edges as two separate types of edges and define the Laplacian dynamics to include motion along both types, ignoring the directionality of the edges. Thus, the continuous-time normalized Laplacian dynamics will be $cdotp_{is} = \sum_{jr} \left(A_{ij} + A_{ji}\right) p_j/k_j - p_i$. The steady state for the dynamics is $p^*_{jr} = k_j/2m$ with $2m = \sum_j kj$. The conditional probability $\rho_{is|jr}$ for the null model must now consider the type of edge being traverse as it may in-going or out-going. Thus

$$\rho_{is|jr} = \left(\frac{k_i^{in}}{m}\frac{k_j^{out}}{k_j} + \frac{k_j^{in}}{m}\frac{k_i^{out}}{k_j}\right)\frac{k_j}{2m} = \frac{k_i^{in}k_j^{out} + k_i^{out}k_j^{in}}{2m^2}. \tag{3.2}$$

We can now state stability for directed edges.

$$R_{directed} = \frac{1}{m}\sum_{ij}\{\left(A_{ij} - \gamma\frac{k_i^{in}k_j^{out}}{m}\right)\delta(g_i, g_j). \tag{3.3}$$

This recovers the generalization for the standard directed null model [17] with an incorporated resolution parameter γ.

The next form of network to be consider is an undirected, signed network such that an edge may have a positive or negative link weight. Clearly, edges with differing signs will be treated separately as one contributes positively and the other negatively, however we define both $A^-_{ij} \geq 0$ and $A^+_{ij} \geq 0$. Similar to the way we define two forms of strength for incoming and outgoing edges, we will define the strength to be $k_j = k_j^+ + k_j^-$. Thus, the continuous-time normalized Laplacian dynamics will be $dotp_{is} = \sum_{jr}\left(A^+_{ij} + A^-_{ij}\right)p_j/k_j - p_i$. This is nearly identical as the Laplacian dynamics for directed networks and the steady state is again given by $p^*_{jr} = k_j/2m$, but now $m = m^+ + m^-$. The conditional probability is also nearly identical. However, we will give $A^-_{ij} \geq 0$ and k^-_j a negative contribution to the expression as they are penalizations to the stability. So, using the form of 3.3 we arrive at

$$\rho_{is|jr} = \left(\frac{k_i^+}{2m^+}\frac{k_j^+}{k_j} - \frac{k_i^-}{2m^-}\frac{k_j^-}{k_j}\right)\frac{k_j}{2m} = \frac{1}{2m}\left(\frac{k_i^+k_j^+}{2m^+} + \frac{k_i^-k_j^-}{2m^-}\right). \tag{3.4}$$

The stability is related to the directed case, but it does not simplify as nicely: $R_{signed} = \frac{1}{2m}\sum_{ij}\{\left(A^+_{ij} - A^-_{ij} - \gamma\left(\frac{k^+_i k^+_j}{2m^+} - \frac{k^-_i k^-_j}{2m^-}\right)\right)\delta(g_i, g_j)$.

Before we consider the null model for R_{signed}, we will examine the third and final generalization. Mutislice networks have different spreading weights for the different types of edges, so we need a generalization which accommodates this fact. For signed networks, it can be useful to consider reweighted conditional probabilities at stationarity using some factor other than the relative strengths of the different edges at node j. This generalization will then allow us to give the following final form for R_{signed},

$$R_{signed} = \frac{1}{2m}\sum_{ij}\{\left(A^+_{ij} - A^-_{ij} - \gamma^+\left(\frac{k^+_i k^+_j}{2m^+}\right) - \gamma^-\left(\frac{k^-_i k^-_j}{2m^-}\right)\right)\delta(g_i, g_j). \tag{3.5}$$

The null model that is obtained from the signed networks is the undirected version of a general form of null model for signed networks [18] .

3.1.2 Multislice stability

The three key generalizations can be applied to derive the null model for multislice networks. Note that we must consider additional parameters introduced by the connections between slices. The steady-state probability distribution will be given by $p^*_{jr} = \kappa_{jr}/2\mu$ where $\mu = \sum_{jr}\kappa_{jr}$. The multislice null model will be given in terms of the conditional probability $\rho_{is|jr}$ of visiting node (i, s) conditional on whether there is a structure that will allow the traversal from (j, r) to (i, s).

$$\rho_{is|jr} = \left[\frac{k_{is}}{2m_s}\frac{k_{jr}}{\kappa_{jr}}\delta_{sr} + \frac{C_{jsr}}{c_{jr}}\frac{c_{jr}}{\kappa_{jr}}\delta_{ij}\frac{\kappa_{jr}}{2\mu}\right]. \tag{3.6}$$

Note that the second term within the brackets expresses the conditional probability for motion between two slices and makes careful use of the definition of C_{jrs}. Specifically, movement from (j, r) to (i, s) along an inter-slice coupling allowed only between the same node in different slices. Furthermore, the probability of selecting a inter-slice link between the two nodes is proportional to $\frac{C_{jsr}}{c_{jr}}$ which is precisely the probability of selecting a precise inter-slice link that connects to slice s.

The continuous time Laplacian dynamics on a multislice network must be altered to consider the contribution from both the inter- and intra-slice edges. Thus, it will be given by $\dot{p}_{is} = \sum_{jr}\left(A_{ijs}\delta_{sr} + \delta_{is}C_{jrs}\right)p_{jr}/\kappa_{jr} - p_{is}$.

We can now take the difference of the exponential solution to the Laplacian dynamics and the conditional probability just defined.

$$R_{multislice} = \frac{1}{2\mu}\sum_{ijsr}\{\left(A_{ijs} - \gamma_s \frac{k_{is}k_{js}}{2m_s}\right)\delta_{sr} + \delta_{ij}C_{jsr}\}\delta(g_{is}, g_{jr}) \tag{3.7}$$

Note, the conditional probabilities have been reweighted using γ_s in much the same way as in standard stability. This allows different resolution parameter for each slice. This allows the ability to examine hierarchical network over multiple scales at the same time. The inter-slice coupling parameter ω has been absorbed into C_{jsr}. We will assume that C_{jsr} takes on binary values of $\{0, \omega\}$ to indicate the presence of a coupling (ω) or the lack of a coupling (0).

Therefore, we have obtained a quality function, $R_{multislice}$, which measures the quality of a partition on a multislice network using an appropriately defined null model. This introduces the necessary quality function by which to define a stability-optimization algorithm that allows for community detection to be performed on many types of networks, including multiplex and dynamic, that have previously been neglected by the available community detection algorithms.

Chapter 4

PageRank and Smart Teleportation

The Laplacian dynamics were able to reveal community structures for standard multislice networks once an appropriately defined null model was developed. However, although we are able to generalize the stability equation to directed multislice networks, a major drawback of the generalization was that the directionality of an edge is ignored to ensure an ergodic solution. Even for the single, static network, the directed case must be defined carefully, such as by assuming total connectivity, to ensure an ergodic solution. One way to resolve these problems is to consider different dynamics which may have ergodic solutions for directed networks.*PageRank* is a popular dynamic that has resolved the problem of finding ergodic solutions in directed networks. It was developed as a link-analysis algorithm to rank the importance of webpage results for web keyword searches [19, 20]. PageRank achieves an ergodic solution by introducing teleportation into a random walk.

Definition 7. *Random Walk with Teleportation.* A random walk with teleportation is a standard random walk with the addition of a constant probability of teleporting to a random node at each step.

Random walks with teleportation are necessary for PageRank as it was designed to perform on directed networks that model webpage links, which are one-directional by definition [19]. However, directed networks are not guaranteed to be ergodic because a random walker on a directed network may become trapped at a *dangling node.*

Definition 8. *Dangling Node.* A dangling node is a node with zero out-degree.

Consider a standard random walk governed by normalized Laplacian dynamics, the walker will make a decision about the destination of its next step based in part upon the local topology. Thus, if the next step leads to a dangling node, or in the vicinity of a dangling node, there is a possibility the walker will become stranded there at a future time step. The introduction of teleportation allows the walker to, with some probability α, ignore the local network topology and escape to a random node via teleportation. This ensures an ergodic solution by preventing entrapment, while still allowing directed edges. Unfortunately, this approach is not with out its drawbacks. Previously, a random walker was influenced by the topological properties of the network and thus so was the flow of dynamics on the network. The implementation of an artificial teleportation process disrupts this flow as it also plays a role in the behavior a random walker [3]. An examination of the teleportation step in PageRank provides some useful insight into how teleportation alters a standard random walk.

Teleportation on a network is controlled by a preference vector with elements, v_i, which represents the frequency with which node i is selected as the destination of the teleportation step. Note that this definition of the preference vector assumes that teleportation occurs between two nodes as this is the form of teleportation used by the standard PageRank algorithm.

Definition 9. *Node Teleportation.* Teleportation with a uniform preference vector so that a node is selected as the destination of the teleportation step with uniform frequency.

However, non-standard teleportation can utilize different structural features, such as degree or strength, to define the preference vector in a different way and thus model other forms of teleportation.

Let us consider a weighted, directed network with the adjacency matrix A defined in the standard way and the strength of a node $k_i = k_i^{out} = \sum_j A_{ij}$. Since the network is directed

Definition 10. *PageRank* A damping factor, α, is introduced to a standard random walk. This damping factor represents the probability of continuing a random walk uninterrupted and $1 - \alpha$ is the probability of teleporting to a random node. The expected density of random walkers on a node will be

$$p_{i;t} = \alpha \sum_j T_{ji} p_{j;t-1} + (1 - \alpha + \alpha d_i) v_i) \tag{4.1}$$

Where v_i is a component of the preference vector which represents the frequency with which a node i is selected as the destination of the teleportation step and $d_i = 1$ only if node i is a dangling node and 0 otherwise.

Note two items about this definition, the order of the indices of the matrix T matter as the network is directed and if $\alpha = 1$ then we have derived a standard random walk. The associated continuous time dynamics are as follows.

$$\dot{p}_i = \alpha \sum_j T_{ji} p_j + (1 - \alpha + \alpha d_i) v_i - p_i \tag{4.2}$$

The steady state for this dynamic is $p_i^* = (1 - \alpha + \alpha a_i d_i) \sum_j (I - \alpha T)_{ji}^{-1} v_j$.

In order to ensure an ergodic solution we must prove that $(I - \alpha T)$ is always invertible. This result follows from the following theorem.

Theorem 1. *If u is a vector and M is a transition matrix, then $||Mv||_1 \leq ||v||_1$ where $||\cdot||_1$ is the l_1 norm.*

Proof. This proof is not instructive for our discussion and thus can be found in Appendix I. □

This leads to the following corollary.

Corollary 1. *If M is a transition matrix, then $I - sM$ is invertible for all $s < 1$.*

Proof. This proof is not instructive for our discussion and thus can be found in Appendix I. □

The introduction of teleportation requires that we re-examine the assumptions behind our stability definition. For example, is it still true that random walker remains trapped within a community for a long period of time? Teleportation is non-local in nature so there is a chance this assumption no longer holds. Instead of remaining constrained by the local topology, at each time step there is some probability that the walker will teleport away, possibly out of the local community. In fact, Lambiotte and Rosvall [3] showed that by using Taylor expansion on p_i^* with regards to α we can explicitly see the non-local nature of this form of teleportation.

$$p_i^* = v_i + \sum_{k=1}^{k=\infty} \alpha^k \sum_j (T_{ji}^k - T_{ji}^{k-1}) v_j. \tag{4.3}$$

An examination reveals that each term is associated with paths of length k supporting our intuition about the potential problems caused by teleportation. Lambiotte and Rosvall [3] proposed that one solution to this problem is to employ forms of *smart teleportation* to reduce the non-local nature of teleportation and, for detecting modular structures, they suggest two forms of smart teleportation be combined.

Definition 11. *Link Teleportation.* Link teleportation uses a random walk with teleportation where the elements of the preference vector, v_i, are proportional to the strength of the node i.

Definition 12. *Unrecorded Teleportation* A random walk with smart teleportation where the teleportation step is not recorded, meaning that teleportation step do not contribute towards the density of walkers on a node.

Unrecorded link teleportation can be achieved by defining the elements v_i of the preference vector to be proportional to the out-degree of the node i. Essentially, in a standard teleportation step a random walkers arrival at a node is treated the same regardless of it method of arrival. Therefore, a teleportation step may introduce an artificial connection between two communities.If instead the teleportation step is not recorded, then , since the frequency is proportional to the out-degree i, the teleportation step is as if the random walker teleported to a random edge. So, when considering the density of walkers $\dot{p}_i$ at node i the contributions from teleportation steps are not recorded, and thus not consider in any measure of the partition. This results in the steady-state solution $p_i^* = (1-\alpha)\sum_l T_{li} \sum_j (I - \alpha T)_{jl}^{-1} v_j$. Notice this is the steady-state solution for standard PageRank with an additional step for the random walker at the end. Again, we use Taylor expansion on the steady-state solution.

$$p_i^* = \sum_l T_{li} \left(v_i + \sum_{k=1}^{k=\infty} \alpha^k \sum_j (T_{ji}^k - T_{ji}^{k-1}) v_j \right). \tag{4.4}$$

Now the dominant component of p_{is}^* is v_i, which has been defined as proportional to the out-strength of the node i. Therefore, by imposing unrecorded link teleportation we regain some of the local nature of a random walk. We will use unrecorded link teleportation in the following discussions.

Previously, a random walk with smart teleportation was employed by Lambiotte to identify modular structures in a network using a clustering method known as InfoMap [3]. I sought to extend the application of smart teleportation for identifying community structures by defining a stability function using the PageRank algorithm with unrecorded, link teleportation.

The first step is to solve the original dynamical system $\dot{p}_i = \alpha \sum_j T_{ji} p_j + (1 - \alpha + \alpha d_i) v_i - p_i$.

$$p_i = (1-\alpha) \int_0^\tau (e^{-(I-\alpha T)\tau})_{ij} v_i T_{ij} d\tau. \tag{4.5}$$

Integrating and linearizing the solution we arrive at the following expression.

$$p_i = \sum_j (\delta_{ij} + t(\alpha T - I)_{ij}) p_i^*. \tag{4.6}$$

We now have all the expressions necessary to define stability for a standard network under PageRank with unrecorded link teleportation. Let us define the resolution parameter $gamma = \frac{1}{t}$ so that we arrive at the following final expression for stability.

$$R_{UR} = \sum_{ij} \left(\alpha(T_{ij}) p_i^* - \gamma p_i^* p_j^* \right). \tag{4.7}$$

Note that the same null that was used for standard stability was employed in this approach as well.

4.0.3 Stability-optimization and PageRank

In order to determine how the introduction of teleportation affects community detection, a few experiments were undertaken. The first was to define special benchmark directed networks that are designed to trap a random walker in a community using the directionality of edges. The simplest way to ensure a trapped random walker is to form a circle so that there is a single entrance and a single exit node. Once a walker enters the circle, it must traverse the directed edges around the circle to an exit node at the opposite side. A simple example of such a network with 5 communities composed of 6 nodes each is displayed in 4.1. Fixing the resolution parameter at $\gamma = 0.1$ the teleportation parameter was varied $\alpha = 0.05, 0.1, ..., 0.95$ and PageRank stability-optimization was undertaken using a general Louvain method [22] to determine the resulting communities.

Next, the Zacharys Karate Club was explored in a similar manner. A few interesting resolution parameters are given in **??**. The introduction of teleportation prevents the original community partition from be discovered. Instead, at least 4 communities are found under each resolution parameter when $\alpha = 0.95$ and for any resolution parameter greater that 1.5, exclusively disjoint communities are found.

For these simple results, the indication is that the teleportation rate should be kept low for accurate results. However, further experiments should be performed to test how other types of networks, such as hierarchical networks or those with larger degree distribution, are affected by the introduction of smart teleportation.

Lambioette [3] found that

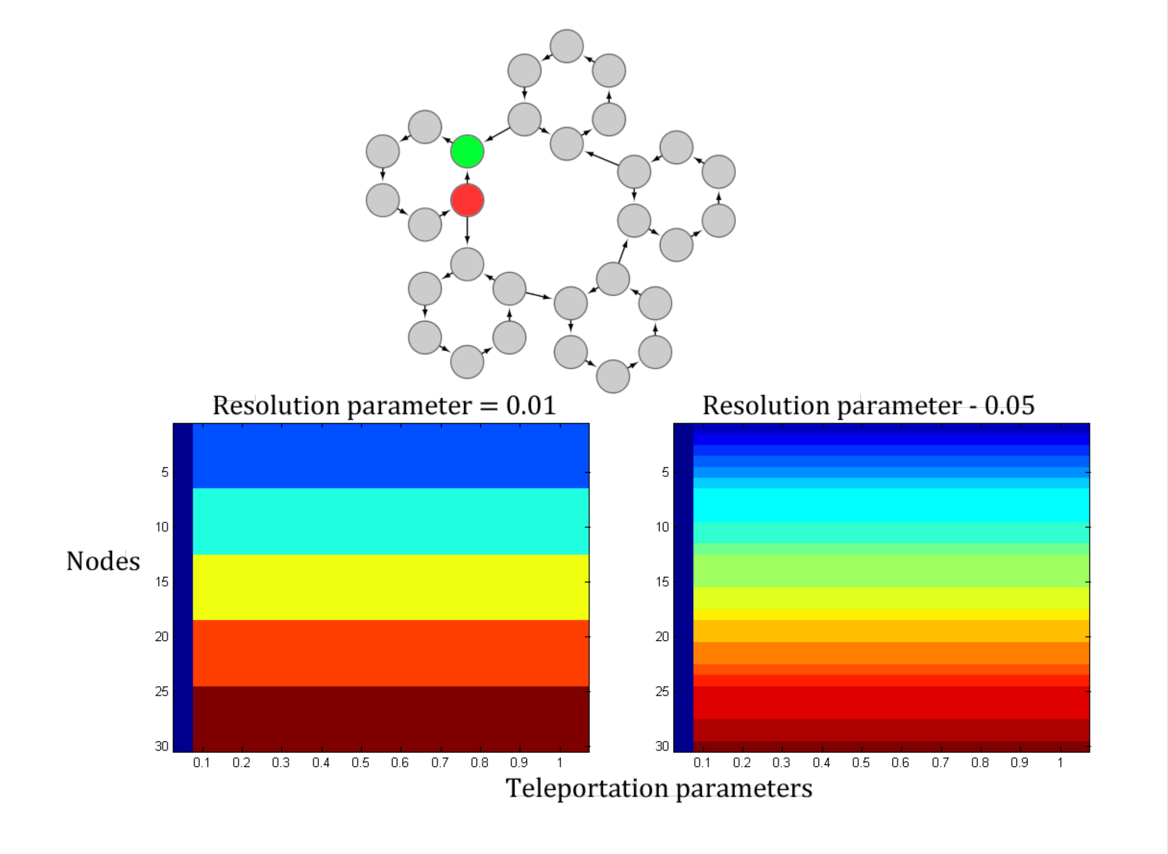

Figure 4.1: The directed, benchmark network is displayed above. The first subnetwork on the left has the entrance and exit nodes highlighted in green and red respectively. The two heat maps display the communities discovered at two separate resolution parameters. Color indicates the assigned community. At an appropriate resolution, the intended community structures are recovered regardless of the teleportation parameter. In fact,the teleportation parameter has very have any affect on the final community assignment for this simple benchmark network due to the unrecorded, link teleportation being used.This figure was generated using the Matlab Heatmap toolbox [21].

4.0.4 PageRank on a multislice network

We can generalize the approach in the previous section to Multislice networks using an approach similar to that found in 3. Just as we had to differentiate between traversing an intra-slice and inter-slice edge, in a multislice network, we must consider two forms of teleportation.

Definition 13. *Inter-slice Teleportation* Teleportation from a node (i, s) to a node (j, r) such that $s \neq r$.

Definition 14. *Intra-slice Teleportation* Teleportation from a node (i, s) to a node (j, s).

As a result there are two possible approaches to defining the frequency vector for the multislice network. We can define a total frequency v_{is} which represents the total frequency with which a node (i, s) is selected. Alternately, we can define two separate frequency vectors which represent the frequency with which a node is selected given that the appropriate form of inter- or intra-slice is occurring at that time step. We will focus on the first form of teleportation and briefly consider the latter in the following section.

Therefore, we will consider a random walk with teleportation such that α is the damping factor. We will again consider an undirected, weighted network.Unfortunately the former notation for multi-slice networks proves unwieldy in the following discussion, therefore we will instead introduce a transition matrix $T_{(is)(jr)}$ encodes the probability of moving from node (i, s) to node (j, r). In terms of our former notations,

$$T_{(is)(jr)} = (A_{ijs}\delta_{sr} + \delta_{ij}C_{jrs})\kappa_{jr}$$

. If $\kappa_{jr} = 0$ then $T_{(is)(jr)} = 0$. It is important to note that in a directed network $T_{(is)(jr)} \neq T_{(is)(jr)}$. We will define $k_{is} = k^{out}_{is} = \sum_j A_{ijs}$ and $c_{js} = c^{out}_{js} = \sum_r C_{jsr}$. Thus, the strength for each node is $\kappa = \sum_{jr} A_{ijs} + C_{jsr} = \sum_{jr} T_{(is)(jr)}$. It is also useful to impose teleportation for dangling nodes (j, r) by replacing the associated column in the transition matrix with the teleportation frequency vector v.

To define link teleportation on a multislice network, consider the preference vector $v_{is} = \frac{1}{\kappa_{is}}$. The dynamics for the density of random walkers on a node can now be expressed.

$$\dot{p}_{is} = \alpha \sum_{jr} T_{(jr)(is)} p_{jr} + (1-\alpha)v_{is} - p_{is}. \tag{4.8}$$

The steady-state solution is $p^*_{is} = (1-\alpha)\sum_{lm} T_{(lm)(is)} \sum_{jr}(I - \alpha T)^{-1}_{(jr)(lm)} v_{jr})$.

The conditional probability $\rho_{is|jr}$ is defined as in the standard multislice network but with the additional probability that a walker will teleport from the node (j, r) to (i, s).

$$\rho_{is|jr} = \alpha \left[(\frac{k_{is}}{2m_s}\frac{k_{jr}}{\kappa_{jr}})\delta_{sr} + \frac{C_{jrs}}{c_{jr}}\frac{c_{jr}}{\kappa_{jr}}\delta_{ij} \right] + (1-\alpha)v_{is}. \tag{4.9}$$

Linearizing the exponential terms in the solution p_{is}, our final stability for a random walk withe teleportation on a multislice network is revealed to be

$$R_{Telep.} = \sum_{ijsr} \alpha T_{(jr)(is)} p^*_{jr} - \gamma_s \left[\alpha(\frac{k_{is}}{2m_s}\frac{k_{jr}}{\kappa_{jr}})\delta_{sr} + \alpha T_{(is)(jr)}\delta_{is} + (1-\alpha)v_{is} \right] p^*_{jr}. \tag{4.10}$$

where $gamma = \frac{1}{t}$ is the resolution parameter.

PageRank on a directed network ensures an ergodic solution and it allows us to incorporate directionality of the network into our stability. However, it does comes with the same drawbacks as those that occurred in a singular, static network. Even though unrecorded link teleportation helps to eliminate the noise introduced by teleportation by artificially creating edges, it is still true that the random walker's decisions are no longer as strongly connected to the network topology. Despite its drawbacks, the stability equation for a directed, weighted multislice networks is a key step in generalizing the work of Lambiotte et al. [9] and Mucha et al. [1] to a networks with directed edges.

4.0.5 Intra-slice and inter-slice restricted teleportation

In special cases it seems natural to seek to further reduce the problem of noise caused by teleportation. In the previous subsection, we considered *total teleportation* as it allowed teleportation to occur between nodes in any location of the multislice network. However, for multislice networks with undirected intra-slice edges but directed inter-slice edges the teleportation could be restricted to inter-slice jumps. Such teleportation is not strictly necessary to ensure an ergodic solution, the undirected nature of the inter-slice edges always prevent a walker from becoming trapped. However, for the purpose of identifying communities they are necessary. Without teleportation, a walker may become trapped in the same slice which negates the benefits from the multislice structure as the long-term behavior of the walker will be equivalent to a random walk in that single slice. The alternate approach is to allow directionality and use the unrecorded link teleportation for inter-slice teleportation. By restricting the teleportation to inter-slice jumps, we avoid the problem of becoming stranded in a slice while reducing the disruption from teleportation as it is less likely that the random walker will teleport to an entirely different community. Such an approaches result in dynamical systems of the following form, where $u_{is} = \frac{1}{c_{is}}$ is the preference vector

$$\dot{p}_{is} = \sum_{jr} \alpha(A_{ijs}\delta_{sr} + C_{jrs}\delta_{ij})\frac{p_{jr}}{\kappa_{jr}} + (1-\alpha)u_{is} - p_{is} \tag{4.11}$$

On examination, it is clear that this is the same dynamic as with total teleportation, but with a different preference vector. Therefore, we easily find the the steady-state solution $p^*_{is} = (1-\alpha)\sum_{lm} T_{(lm)(is)} \sum_{jr}(I-\alpha T)^{-1}_{(jr)(lm)} u_{jr})$.

However, the new form of teleportation can be used to better define our conditional probability $\rho is|jr$

$$\rho_{is|jr} = (\alpha \frac{k_{is}}{2m_s} \frac{k_{jr}}{\kappa_{jr}} + (1-\alpha)u_{is})\delta_{sr} + \alpha \frac{C_{jrs}}{c_{jr}} \frac{c_{jr}}{\kappa_{jr}} \delta_{ij}. \tag{4.12}$$

Unlike in total teleportation, there is no contribution from the teleportation guaranteed at each step and the δ_{jr} ensures that the teleportation step only contributes if the nodes are both in the same slice. Therefore, our final stability for a random walk with only inter-slice teleportation on a multislice network is revealed to be

$$R_{IS} = \sum_{ijsr} \frac{\alpha}{\kappa_{jr}} (A_{ijs}\delta_{sr} + \delta_{ij}C_{jsr})p^*_{is}p^*_{jr} - \gamma_s[\alpha(\frac{k_{is}}{2m_s}\frac{k_{jr}}{\kappa_{jr}})\delta_{sr} + (1-\alpha)u_{is}\delta_{sr} + \alpha \frac{C_{jsr}}{\kappa_{jr}}\delta_{ij}]p^*_{jr}. \tag{4.13}$$

The definition of stability for directed inter-slice edges and undirected intra-slice edges by restricting the teleportation to intra-slice jumps is derived in a nearly identical fashion and thus has been omitted.

4.0.6 Community detection using PageRank stability on multislice networks

Consider, the Zacharys Karate Club defined as in the standrad multislice example but with the additional restriction that intraslice edges connection slices with different resolution parameters are directed.

The test for the stability equation for a random walk with teleportation was on the Zacharys Karate Club (ZKC) benchmark networks (See Fig. 1). The same 34-node unweighted adjacency matrix was maintained across 16 slices (so that $A_{ijs} = A_{ij}$ for all s). Simultaneous community detection across 16 resolution parameters was performed with the following sequence of resolution parameters, $\gamma_s = 0.25, 0, 5, 0.75, ..., 4$. The selection of these parameters was due to the ability to compare results with previous multislice results on the ZKC network [1]. The results for an interesting parameter is displayed in 4.3, the intraslice strength parameters varied was $\omega = 0.1$, the teleportation parameter was $\alpha = 0.85$ and the resolution parameter varied over $\gamma = 0.25, 0.5, ..., 4$. These results were generated using stability-optimization with a generalization of the Louvain algorithm [22] and a process known as the Kernighan-Lin (KL) node-swapping steps [23].

As expected, small resolution parameters are needed to detect communities. Additionally, it is quite clear from examining the left hand side of each diagram that when $1 - \alpha$, the probability of taking a teleportation step, increased above 0.5 that the modular structure of the graph is no longer distinguishable using the PageRank dynamics to define stability. Above a resolution parameter of 2, and values near it, each node is seen as its own community. For small values of the teleportation parameter, again, each node is seen as it's own community as it is impossible to completely eliminate the non-local contributions that teleportation introduces.

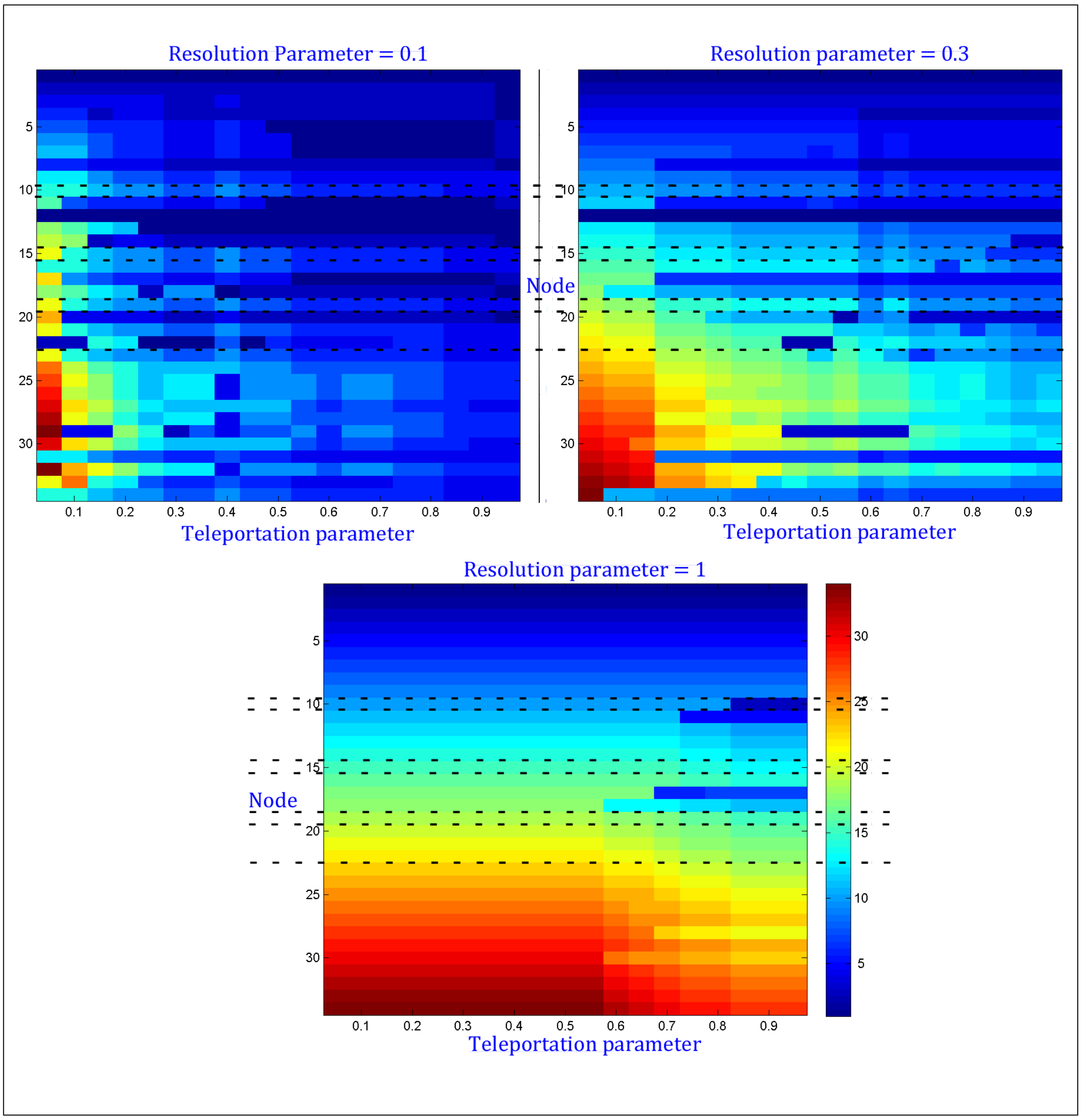

Figure 4.2: The Zacharys Karate Club [4] was analyzed using the PageRank stability for a single, static network. Colors depict community assignments of the 34 nodes and each column is an independent assignment of the nodes to communities. The resolution parameter γ was held constant and the teleportation parameter α was varied over $\{0.05, 0.1, 0.15, ..., 0.95\}$. The dashed lines represent the original community division as portrayed in 1.1. As the teleportation parameter α increases, the chance of teleportation decreases, and thus the underlying community structures are more closely preserved. However, even the introduction of a small amount of teleportation results in a minimum of 4 communities, where as without teleportation only two communities are observed at the lowest resolution parameter [9]. As expected, the community structures observed change under different resolution parameters and as the resolution increase the communities structure of the underlying network is no longer distinguishable and each node begins to be placed in its own community. This figure was generated using the Matlab Heatmap toolbox [21].

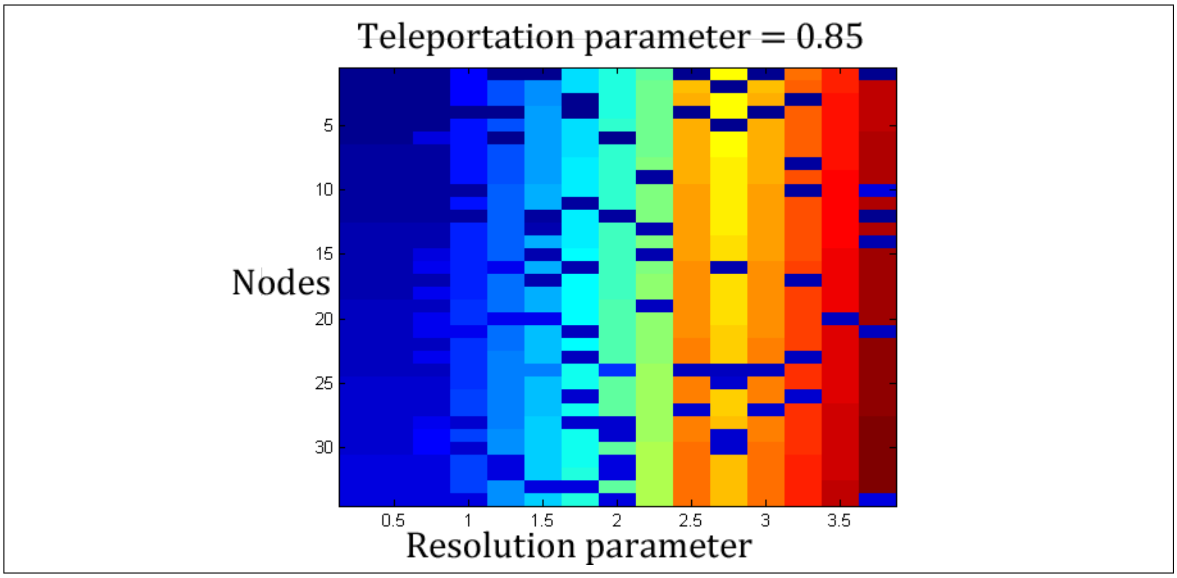

Figure 4.3: The Zacharys Karate Club [4] was analyzed using the PageRank stability for mutlislice networks. Colors depict community assignments of the 34 nodes and each column is an independent assignment of the nodes to communities. The resolution parameter γ was varied and the teleportation parameter α was held constants at $\alpha = 0.85$. As the resolution parameter α increases, the teleportation parameter appears to have introduced noise into the community assignment. However, for smaler values thus the underlying community structures are not too far removed from the expected community distribution for the Zacharys Karate Club. In contrast when the teleportation parameter α was allowed to be near .50 and $\omega = 0$ the communities deteriorated into nearly every community consisting of a single node.This figure was generated using the Matlab Heatmap toolbox [21].

Chapter 5

Conclusions and Future Work

The work outlined in this dissertation is still ongoing and two open areas of work are outlined below. The first, is a non-Laplacian dynamics approach to defining stability. The second, benchmark models which have shown some promise for testing community detection algorithms for multi-scale networks..

5.1 Non-laplacian dynamics and stability

One form of dynamic which has yet to be generalized for stability is a non-Laplacian dynamic.

5.1.1 SIS-Model

One epidemic model known as Susceptible-Infected-Susceptible (SIS) provides some clues for successfully generalizing stability for a non-Laplacian dynamic. However, time constraints have prevented a full exploration of this area. I have included the motivation behind this exploration in hopes that it will give some insight into future directions for research.

SIS is one of the simplest virus infection models, in which nodes in a network are in one of two states: Healthy, but susceptible to infection or Infected by the virus and infectious to adjacent nodes. For our purpose, we will use the terms "infection" and virus loosely to express the possibility that some "virus is transferred from an infected individual to the local network surroundings. Rather than considering a virus, suppose and item is being passes from node to node.

Definition 15. *Susceptible-Infected-Susceptible (SIS) model* In the SIS model [24], nodes have two states, infected or susceptible. For an infected node, there is probability β for each adjacent edge that the infection will spread to a neighbor. Once infected, a node may recover from the infection with some probability δ. Therefore, the effective infection rate $\tau = \frac{\beta}{\delta}$.

The importance of this is that unlike many other infection models, as time goes to infinity we do not necessarily expect to see the entire system become solely infected or healed.

Recall that stability was based on the idea of a random walker being trapped in a community for a long period of time. In this model, unlike a random walk where the event that a walker is at node i and at node j are mutually exclusive, the events that node i and j are infected may occur at the same time. However the idea of passing an item from one node to another is quite similar to the idea of random walker. The main difference is that the SIS model at each time step a node has a probability of splitting the item into pieces and passing them to its neighbors. In a similar way, the recovery probability can be thought of as the probability that the node destroys the item. As we assumed with a single random walker, these multiple items should each remain within in a community, assuming a low disappearance rate, for a long period of time due to the modular structure of a sommunity. Alternately, this expresses that the spread of an infection will remain localized within a community for long period of time, a concept that is already used when modeling the spread of disease [25].

Therefore, although we would move away from the familiar Laplacian dynamics the motivated our formulation, the idea which motivated stability remains largely intact. However, since this is a *filling model*, careful consideration must be given to whether stability would still be defined as with random walks for such a model.

Current research has a discrete-time SIS-model introduced by Wang et al. [24] with a threshold for a steady-state solution that can be used for any network. However, this steady-state turns out to be intractable with real networks due to the size of the Markov process, thus a mean-field approximation to the solution known as *N-intertwined SIS* was introduced by [26] is a possible area to explore. The *N-intertwined SIS* gives a continuous time approximation to the system. However, the resulting dynamical system is non-linear and the solution is not easily found in order to predict the probability that an infection is contained within a community initially and at time t. In order to find an expression for stability, an approximation for the model would need to be made. One possible idea is to use the fact that dynamical systems behaves in a nearly linear fashion near equilibrium points. Thus we could impose a restriction on τ to ensure that the system is approximately linear. The discrete-time model for SIS is presented below. Notice the non-linear form of the dynamics.

5.1.1.1 Discrete-time SIS

Wang et. al. [24] express the spread of disease through a network using $p_{i,t}$ the probability that a node is infected at time t. To define this expression, consider $U_{i,t}$ the probability that a node is not infected by a neighbor at time t. Although this seems to be the opposite of out intended goal of finding the probability of infection for a node, we utilize this approach as $1 - pi, t$, the probability a node is not infected at time t, is more intuitive to express.

$$U_{i,t} = \prod_{j,j \in Nb} (1-\beta)p_{j,t-1} + (1 - p_{j,t-1}) = \prod_{j,j \in Nb} 1 - \beta p_{j,t-1} \tag{5.1}$$

Consider that a node i is susceptible at a given time step under two conditions:

- At the previous step, i was healthy and in this time step did not receive an infection from any infected neighbors.
- The node i was infected at a previous time step and was cured at the current time step. This can occur in two ways.
  - Node i receives an infection from a neighbor at the current time step as it is being cured, in this case the "new" infection is ignored.
  - Node i does not receives an infection from a neighbor at the current time step and is cured.

The assumption is made that for an infected node, the probability that a parallel infection from a neighbor and cure at a time step occurs roughly 50

Therefore, we can now define the probability that a node is healthy.

$$1 - p_{i,t} = (1 - p_{i,t-1})U_{i,t} + \delta p_{i,t-1}U_{i,t} + \frac{\delta}{2} p_{i,t-1}(1 - U_{i,t}) \tag{5.2}$$

The first term in this expression is the probability that node i was not infected at the previous time step but is infected at this time step.

Therefore, we define the infection density to be

$$\dot{p_i} = 1 - (1 - p_{i,t-1})U_{i,t} + \delta p_{i,t-1}U_{i,t} + \frac{\delta}{2} p_{i,t-1}(1 - U_{i,t}) - pi; t \tag{5.3}$$

The steady state solution is the discrete-time expression $p_i^* = \frac{1 - C_{i,t}}{\delta C_{i,t} - C_{i,t} + \frac{\delta}{2}(1 - C_{i,t}) + 1}$. Wang, et al. [24] proved the following.

Theorem 2. *SIS-Model Epidemic Threshold If an epidemic dies out on a network A, then the the effective infection rate* $\frac{\beta}{\delta} < \tau = \frac{1}{\delta_{1,A}}$*, where* β *is a constant birth rate, at* δ *the recovery rate, and* $\delta_{1,A}$ *is the largest eigenvalue of the network's adjacency matrix A.*

Proof. The proof of this theorem is rather long and adds little to the immediate discussion, therefore, interested readers are referred to [24]. □

Equivalently, above this threshold τ, a persistent number of nodes will remain infected and thus a steady-state is guaranteed. Note that this is a Markov process for a network A with N nodes so that every node has two possible states. The state of the entire network at time t is S(t) and will be defined by every possible configuration of state on the nodes that can be achieved at time t.Therefore the state space consists of 2^N states .This makes this model computationally intractable for real world networks [27]. However, a reasonable approximation based on mean field theory known as the *N-intertwined SIS-model* was recently introduced by Van Mieghem et. al [27, 26] and seems to be a possible area of exploration for finding a tractable solution for the steady state needed to define stability.

This model does not fit perfectly with the previous definitions designed for stability with normalized Laplacian matrices. Thus, additional work is required in order to better understand how the a filling model dynamic will change the ability to detect underlying structures in a network and if nonlinear systems are impractical due to the inability to solve the system without imposing strict restrictions on the parameters of the model.

5.2 Benchmarks for Multislice Community Detection

There are no benchmarks specifically designed for multislice networks. It is possible to use certain standard benchmark networks such as the popular LFR benchmarks [28] to define a multislice network by placing instances of the benchmark in each slice. This is similar to what was done with the multislice Zacharys Karate Club network. However, although this can provide some insight, it does not specifically address the complicated structures that a multislice network can allow.

For example, LFR benchmarks [28] allow for communities of varying size and degree distribution. If an LFR benchmark network is generated and copied to each slice the slice are coupled with some weight ω, then multislice community detection can be performed over several resolution parameters to find hierarchical communities in the networks. The use of LFR benchmarks to test the ability for stability-optimization to find hierarchical structures can thus be tested in this way. IT exploits the ability to use multiple resolution parameters. However, an alternate approach would be to put a separately generated LFR benchmark in each slice and couple these slices together. In this case the special construction of multislice networks is not explicitly considered when determining how the model should be constructed. Furthermore, LFR benchmarks are constructed with specific degree requirements, it would seem that the contribution of intra-slice connections should be considered in a model if the degree distribution is to be carefully controlled.

Ideally, a set of benchmark network are needed which consider the special structure of multislice networks. This may require multiple types of benchmarks, as a benchmark to model directed or time dependent networks may require consideration not necessary for detecting hierarchical community structures or multiplex networks. Once benchmarks have been developed, then a rigorous test of the multislice stability formulations with multiple parameters to determine the effect of varying the parameters should be undertaken.

5.3 Conclusion

There is much work that can still be done to explore both the generalizations for stability to other forms of dynamics and further extensions of multislice networks. I have presented two of unfinished problems that I found most interesting and which show the most promise for successful future exploration. Further applications and verifications of the PageRank model would be useful to help determine it's limits and to analyze some real-world models with time dependencies to see is current community detection algorithms have missed out on key communities due to inefficient models for directed edges.

Theorems, Corollaries, and Proofs This appendix includes proofs to the theorems and corollaries given in the main text.

Theorem 3. *If u is a vector and M is a transition matrix, then $||Mv||_1 \leq ||v||_1$ where $||\cdot||_1$ is the l_1 norm.*

Proof. The definition of the l_1 norm for a vector is $||u||_1 = \sum_i |u_i|$. Applying this definition directly to our expression.

$$||Mv||_1 = \sum_i sum_j |M_{ij}u_j| \leq \sum M_{jk}|v_k| \leq ||v||_1$$

Using the triangle inequality, $|a + B| \leq |a| + |b|$ and the fact that each entry M_{jk} in a transition matrix is positive so $M_{jk} = |M_{jk}|$. A transition matrix has column sum 1 for each column. Therefore, if we sum over k we get our desired result.

$$||Mv||_1 \leq \sum_k |v_k| = ||v||_1$$

□

This theorem leads to the following interim corollary concerning the eigenvalues of a transition matrix that was omitted form the main text but proves helpful in proving other results.

Corollary 2. *If M is a transition matrix, then all for all eigenvalues λ of M it is true that $\lambda \leq 1$*

Proof. Let v be a be an eigenvector if M with eigenvalue λ, so $\lambda v = Mv$. Use the definition of the l_1 norm and apply it to both sides.

$$|\lambda| ||v||_1 = ||Mv||_1$$

Now, from the previous proof we know that $|\lambda| ||v||_1 \leq ||v||_1$, and thus we have achieved our result $|\lambda| < 1$. □

Corollary 3. *If M is a transition matrix, then $I - \gamma M$ is invertible for all $\gamma < 1$.*

Proof. This follows from the previous corollary. We will prove the result by contradiction. Suppose that $I - \gamma M$ is not invertible, then for some non-trivial vector v it is true that $(I - \gamma M)v = 0$. However, this means that $Mv = \frac{1}{\gamma}v$, implying that M has an eigenvalue $\frac{1}{\gamma} > 1$, but this is prohibited by the previous theorem. Thus, a contradiction. □

Bibliography

[1] M. A. Porter, J.-P. Onnela, and P. J. Mucha, “Communities in networks,” *Notices of the American Mathematical Society*, vol. 56, p. 1082, 2009.

[2] S. Fortunato, “Community detection in graphs,” *Physics Report*, vol. 486(3-5), pp. 75–184, 2010.

[3] R. Lambiotte and M. Rosvall, “Ranking and clustering of nodes in networks with smart teleportation,” *Phys. Rev. E*, vol. 85, p. 056107, 2012.

[4] W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[5] S. Klamt, U. Haus, and F. Theis, “Hypergraphs and cellular netowrks,” *PLoS Computational Biology*, vol. 5(5), p. e1000385., 2009.

[6] X. S. Zhang, R. S. Wang, Y. Wang, J. Wang, Y. Qiu, L. Wang, and L. Chen, “Modularity optimization in community detection of complex networks,” *Euro-Physics Letters*, vol. 87, no. 3, p. 38002, 2009.

[7] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 026113, 2004.

[8] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” *Science*, vol. 328, pp. 876–878, 2010.

[9] R. Lambiotte, J.-C. Delvenne, and M. Barahona, “Laplacian dynamics and multiscale modular structure in networks,” *arXiv:0812.1770v3*, 2009.

[10] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner, “On modularity clustering,” *IEEE transactions on knowledge and data engineering*, vol. 20(2), pp. 172–188, 2008.

[11] S. Fortunato and M. Barthélemy, “Resolution limit in community detection,” *Proceedings of the National Academy of Sciences USA*, vol. 104.

[12] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, “Synchronization in complex networks,” *Physics Reports*, vol. 469, pp. 93–153, 2008.

[13] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence.* Springer-Verlag, Berlin, Germany, 1984.

[14] R. Lambiotte, “Multi-scale modularity in complex networks,” *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, vol. 2010 Proceedings of the 8th International Symposium on, pp. 546–553, 2010.

[15] P. O. Perry and P. J. Wolfe, “Null models for network data,” *arXiv:1201.5871v1*, 2012.

[16] M. J. Barber, “Community structure in directed networks,” *Phys. Rev. E*, vol. 100, p. 118703, 2008.

[17] E. A. Leicht and M. E. J. Newman, “Community structure in directed networks.,” *Phys. Rev. Letters*, vol. 76, p. 066102, 2007.

[18] V. A. Traag and J. Bruggeman, “Community detection in networks with positive and negative links,” *Phys. Rev. E*, vol. 80, p. 036115, 2009.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.,” Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[20] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in *Seventh international world-wide web conference*, 1998.

[21] A. Deoras, *Customizable Heat Maps.* Natick, Massachusetts: The MathWorks Inc., 2011.

[22] P. de Meo, E. Ferrara, G. Fiumara, and A. Provetti, “Generalized louvain method for community detection in large networks,” *11th International Conference On Intelligent Systems Design And Applications*, pp. 1–6, 2011.

[23] B. Kernighan and S.Lin, “An efficient heuristic procedure for partitioning graphs,” *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[24] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsosy, “Epidemic spreading in real networks: an eigenvalue viewpoint,” *22nd international Symposium on Reliable Distributed Systems (SRDS03).*

[25] M. Salathe and J. H. Jones, “Dynamics and control of diseases in networks with community structure,” *PLoS Computational Biology*, vol. 6(4), p. e1000736., 2010.

[26] P. V. Mieghem, “The n-intertwined SIS epidemic network model,” *Computing*, vol. 93, p. 147.

[27] P. V. Mieghem, J. Omic, and R. E. Kooij, “Virus spread in networks,” *IEEE/ACM Transaction on Networking*, vol. 17, no. 1, pp. 1–14, 2009.

[28] L. A, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Phys. Rev E*, vol. 78(4), p. 046110, 2008.