Community Detection using Locality Statistics

by

Heng Wang

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

November, 2015

Abstract

The goal of community detection is to identify clusters and groups of vertices that share common properties or play similar roles in a graph, using only the information encoded in the graph. Our work analyzes two methods of identifying an anomalous community in temporal graphs and another method of identifying active communities in a static massive graph. All methods are based on locality statistics.

In [50], an anomalous community is detected that shows growing connectivities in a time series of graphs. We formulate the task as a hypothesis-testing problem in stochastic block model time series. We derive the limiting properties and power characteristics of two competing test statistics built on distinct underlying locality statistics. In addition, we provide applicable implementations of two competing test statistics and detailed experimental results for a neural imaging application in [36].

In [51], active communities are detected in a static massive graph on which many community detection algorithms scale poorly. We propose a novel framework for detecting active communities that consist of the most active vertices. Our framework utilizes a parallelizable trimming algorithm based on a locality statistic to filter out inactive vertices, and then clusters the remaining active vertices via spectral decomposition of their

similarity matrix. The framework is applicable to graphs consisting of billions of vertices and hundreds of billions of edges.

In summary, this work provides developments in community detection, in both temporal graphs and static massive graphs, by employing locality statistics.

ABSTRACT

Advisor: Carey E. Priebe

Primary Reader: Carey E. Priebe

Secondary Reader: Amitabh Basu

Acknowledgments

I would never have been able to finish my Ph.D. journey without the guidance of my advisor, help from research collaborators, and support from my family and friends.

First, it is my pleasure to sincerely thank my advisor, Carey E. Priebe, for his support, caring, and excellent guidance during my graduate studies. He introduced me to this fascinating topic, encouraged me to explore methodologies in the field, and also mentored me to become an independent researcher. In the past five years, Dr. Priebe has not only provided insightful suggestions to help me increase expertise on this research topic but also continued to support my personal progress from all sides. For everything you have done for me, I truly appreciate all of it. As an old Chinese saying says: "One day as your teacher, like a father for a lifetime."

I would also like to acknowledge Dr. Youngser Park, Dr. Minh Tang, and Da Zheng for their guidance and help in my research. As collaborators and friends, they always enthusiastic about helping me develop computational skills, presentation skills, critical thinking, and other computer science domain knowledge. Special thanks to Dr. Basu, who agreed to be on my final defense committee and gave valuable comments regarding my dissertation thesis as well. My research would not have been possible without their

help.

Finally, I want to thank my parents, my girlfriend, and all my soccer teammates. Their encouragement and love have made my Ph.D. life full of enjoyable, memorable, and pleasant moments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problems

In the analysis of network graphs, community detection often refers to identifying, in an unsupervised fashion, a subset of vertices that have excessive "cohesiveness" among themselves, and at the same time are relatively well separated from the remaining vertices. Interest in community detection has increased because communities in a large graph often imply noteworthy group structures in a graph-represented real system. For example, as summarized in [10], in a World Wide Web graph, communities are more likely to be groups of web pages associated with similar topics; in a protein-protein interaction graph, communities are formed by proteins having the same functionality within a cell; in a scientific citation graph, communities are identified as research collaborators or potential collaborators. These valuable findings could further lead to concrete applications in business insights, security enhancement, recommendation systems, and so on.

In this work, we conduct community detection in two specific scenarios. In the first scenario, we assume that the network is temporal, and the objective is to find the emergence of an anomalous community in a time series of networks. The *anomalous community* is a local region in the network with a significantly growing number of communications occurring at some unknown time stamp. This time stamp is also called the *change-point* in a time series of graphs. In the second scenario, we consider a static massive graph where the number of vertices and edges is on the scale of a billion. This is an interesting situation because such a graph is too large to be processed using its full topology. The goal is to detect potential communities consisting of the most active vertices in a network by ignoring insignificant vertices of a network. The communities we obtain in this scenario are called *active communities*.

First, we discuss previous work done in the area of the change-point detection problem. The change-point detection problem in a dynamic network is becoming increasingly prevalent in many applications of the emerging discipline of dynamic graph mining. Dynamic network data are often readily observed, with vertices denoting entities and time-evolving edges signifying relationships between entities, and thus considered as a time series of graphs, which is a natural framework for investigation. An anomalous signal is broadly interpreted as constituting a deviation from some normal network pattern, e.g., a model-based characterization such as large scan statistics [38] or non-model-based notions such as a community structure change, while a change-point is the time window during which the anomaly appears.

Recently, many tailor-made approaches based on different models, aiming for change-point detection in graphs, have been proposed in a growing body of literature. In [17],

a two-stage Bayesian anomaly detection method for social dynamic graphs is designed. Both its model and parallelization in computation are built on the assumption that the communication between each pair of individuals independently follows a counting process. In [30], an algorithm called NetSpot was created to find arbitrary but evolutionary anomalies that are maintained over a spatial or time window; i.e., the anomalous signal does not appear and then disappears instantaneously. In [49], the subgraph anomaly detection problem in graphs was analyzed through likelihood ratio tests under a Poisson random graph model. Finally, in [29], the L_1 norm of the eigenvectors of the modularity matrix was used to detect an (anomalous) small dense subgraph embedded inside a large, sparser graph.

Second, we review varied algorithms proposed to locate communities in massive graphs. Let n denote the number of vertices of a graph and m denote the number of edges. A traditional graph partitioning approach, the Kernighan-Lin algorithm [21], is still widely used today and has a complexity of $O(n^2 \log n)$ and $O(n^2)$ on sparse graphs. A hierarchically agglomerative clustering approach, embedding all vertices in space to employ a similarity measure, results in a complexity of $O(n^2)$ for single linkage and $O(n^2 \log n)$ for a complete and average linkage scheme [10]. A hierarchically divisive clustering algorithm proposed by Girvan and Newman [14] [33] iteratively partitions a graph by removing edges with low similarity, takes $O(nm^2)$, and gains popularity in countless applications [10]. In contrast, spectral clustering such as that used in [8] and [53] has a much lower asymptotic computational complexity. Their most expensive cost is the computation of the dominant Laplacian eigenvectors, which has a complexity of $O(m)$ in each iteration but may require a large number of iterations [5].

Another prominent approach is a group of modularity-based methods, developed from the stopping criterion of the Girvan and Newman algorithm in [14]. A greedy modularity optimization algorithm [6] allows the analysis of large graphs with up to $n = 10^6$ vertices and a running time $O(n \log^2 n)$ and is improved by [46] to handle graphs of up to $n = 10^7$. In the past few years, the modularity-based technique [4] known as Louvain clustering has been in vogue because it can analyze graph sizes of up to $m = 10^9$ in a reasonable time. The phase of attaining local modularity maxima in Louvain clustering requires multiple iterations, and each iteration has a complexity of $O(m)$. The downside is that the number of iterations is unknown and the convergence speed is influenced by the order of sequential sweeps over all vertices.

In this dissertation, we will investigate the above two community detection problems by making use of a locality statistic. The locality statistic is a foundation for the work in this dissertation. In Chapter 2, we approach the dynamic anomalous community/change-point detection problem through the use of locality-based scan statistics. In Chapter 3, we propose an active community detection framework through the use of a locality-based trimming algorithm.

1.2 Graph Preliminary and Notation

The term "network" often refers to a collection of individuals and inter-relations among these individuals. Its mathematical structure can be well-represented by terminologies from a subfield of mathematics – graph theory. Specifically, vertices denote individuals in a network, and edges denote interactions between individuals. In this section, we review some basic concepts from graph theory and introduce the corresponding notation that will be used throughout the dissertation.

Generally, a graph is denoted by $G = (V, E)$, with the vertex set $V = V(G)$ and edge set $E = E(G)$. The number of vertices of a graph is usually denoted by n (or $|V|$), and the number of edges is denoted by m (or $|E|$). They are also sometimes called the *order* and *size* of the graph G, respectively. We denote by A the adjacency matrix of a finite graph G where A is an $n \times n$ matrix such that $A_{u,v}$ is the number of edges (edge weight) from vertex u to vertex v in an unweighted (weighted) graph. For a graph G on n vertices, the vertex set is usually taken to correspond to the set $[n] = \{v_1, v_2, \ldots, v_n\}$. In our subsequent discussion, we might also partition V into subsets, or blocks. If V is partitioned into B blocks of size $n_1, n_2, \ldots, n_B$ vertices, then, with a slight abuse of notation, we shall denote by $[n_i]$ the vertices in block i.

Let G be a graph. For any $u, v \in V$, we write (u, v) if there exists an edge between u and v in G. A vertex $v \in V$ is incident on an edge $e \in E$ if v is one of the end points of e. The list of *incident* edges of a vertex v is denoted by $E[v]$. A graph G is *directed* if each edge in E has an ordering to its vertices such that (u, v) is different from (v, u) for $u, v \in V$. We write $d(u, v)$ for the shortest path distance between u and v in G if

G is undirected. If G is directed, with abuse of notation, $d(u, v)$ stands for the shortest path distance between u and v on the underlying undirected graph of G by removing orientations of all edges. A graph is *weighted* if the strength of relations between vertices, i.e., edge weights, are considered in the model. Note that in the next two chapters, we consider only unweighted graphs without self-loops. Some extensions to weighted graphs are discussed in Chapter 4.

A graph $G' = (V', E')$ is a *subgraph* of another graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. For $V' \subset V$, $G' = (V', E')$ is an *induced subgraph* of $G = (V, E)$ by V' if E' is a collection of edges in E, both of whose end points are in V'. For $v \in V$, we denote by $N_k[v; G]$ the set of vertices u at a distance of at most k from v, i.e., $N_k[v; G] = \{u \in V : d(u, v) \leq k\}$ and denote by $\Omega(V', G)$ the subgraph of G induced by V'. Thus, $\Omega(N_k[v; G], G)$ is the subgraph of G induced by vertices at a distance of at most k from v.

As an example to illustrate the above concepts and notation, we consider a simple and small simulated graph below. Figure 1.1 is an unweighted and directed graph $G_{toy} = (V, E)$ with $n = 10$ and $m = 26$. Based on this small graph, in Table 1.1, we summarize the above concepts, their corresponding notation, and example quantity values in G_{toy} to improve the reader's understanding of these concepts.

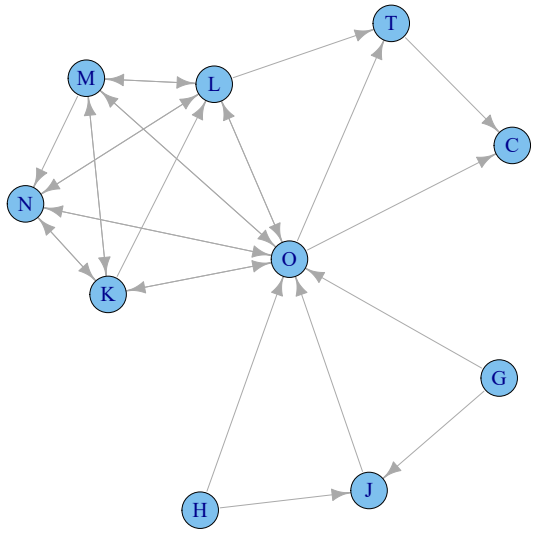


Figure 1.1: A toy example graph G_{toy} to illustrate basic concepts and notation

1.3 Graph Data

In this section, we briefly examine some contexts in which networks arise so that we can establish an initial sense of how the graphs of interest are constructed according to data from the real world. By following [22] and [32], we categorize most networks familiar to us into four classes: technological, social, biological, and informational. This classification is not strict, and it is not uncommon for many networks to fall into more than one category. More specifically, we briefly present in this section three fascinating datasets that will be used in the following chapters for real data experiments. As we will see, each of the three datasets belongs to at least two of the above categories.

Notation	**Concept**	**Value in Figure 1.1**
n (or $\|V\|$)	order (i.e., the number of vertices)	10
m (or $\|E\|$)	size (i.e., the number of edges)	26
B	number of blocks or partitions	e.g., 2
$[n_i]$	vertices in block i	e.g., $[n_1] = \{C, G, H, J, T\}$, $[n_2] = \{K, L, M, N, O\}$
(u, v)	an edge from u to v	e.g., (H, O)
$E[v]$	incident edges of v	e.g., $E[J] = \{(H, J), (J, O), (G, J)\}$
$d(u, v)$	shortest path distance between u and v	e.g., $d(G, N) = 2$
$N_k[v; G]$	set of vertices at a distance of at most k from v	e.g., $N_1[J; G_{toy}] = \{G, H, J, O\}$
$\Omega(N_k[v; G], G)$	subgraph of G induced by $N_k[v; G]$	e.g., $\Omega(N_1[J; G_{toy}], G_{toy}) = (V', E')$ where $V' = \{G, H, J, O\}$ and $E' = \{(H, O), (H, J), (J, O), (G, O), (G, J)\}$

Table 1.1: Summary of basic concepts based on the toy example graph G_{toy} in Figure 1.1.

1.3.1 Technological Networks

A technological network is often a network of human-made systems that functions collectively to produce flows of products and services between entities. To represent a technological network in the form of a graph, we let each entity play the role of a vertex and each flow of products and services play the role of a single (weighted) edge. For example, the airplane network can be a graph in which vertices are airport sites and edges are airline routes between pairs of sites. In the Internet network, individual digital devices (e.g., personal laptops) communicate over wired and wireless connections using Internet traffic

packets. Currently, there are over 3×10^9 Internet users per day across the world, and a massive global Internet graph can be created if we map digital devices to vertices and Internet traffic flows to edges. In general, the topologies of many technological networks are available to us, such as highways between two cities and wireless connections between cell phones and local switches. Thus, for such networks, many investigations have focused on the optimization of flows to accommodate limited physical resources or financial profit maximization.

1.3.2 Social Networks

Social networks link people or groups of people. The vertices of social graphs typically consist of humans, and edges are social interactions among people. Examples of social interactions include friendships among people, personal contacts in a social group, and cooperation to reach a common objective. Traditionally, collecting social network data has been difficult, requiring extensive surveys in the social group being studied. However, the merging of technological networks and social networks presently enables us to quantify interactions in an online setting. Celebrated social sites such as Facebook, LinkedIn, and YouTube are recording the net surfing behaviors of their users at any time. Interactions can now be observed and quantified at different levels of scale and resolution.

Using graph analysis, one typical aim in social network mining is to model social structures, monitor possible structure changes, and unearth hidden social interactions that drive structural change. One of the questions of interest in this dissertation is whether there is an emergence of anomalous subgroups of social actors showing excessive so-

cial interactions among themselves in a dynamic network. We will use an Enron email dataset [39] in Chapter 2 as an example. Email communication networks fall into both the technological network and social network categories. This dataset consists of $|V| = 184$ unique email users at Enron Company and the collection of their emails over a period of 189 weeks from 1998 to 2002. For each week $t = 1, \ldots, 189$, there is a directed graph $G_t = (V, E_t)$ where $(u, v) \in E_t$ denotes that u sends at least one email to v during the t-th week. Investigating the time series of graphs $\{G_t\}_{t=1}^{184}$, we will show in Chapter 2 the emergence of an anomalous community of users and how change points indicated by our methodology are related to the timeline of Enron scandal news.

1.3.3 Biological Networks

A biological network is any network that applies to biological systems. Not surprisingly, the graph data collected on biological networks and the manner in which they are generated and analyzed vary widely with the nature of underlying biological systems. For instance, protein-protein interaction networks are represented by graphs in which proteins are vertices and their interactions are edges. A food web is a graph in which living species, i.e., vertices of the network, are connected to each other through predator-prey interactions. Additionally, there is another rule for graph edge construction such that two vertices are connected if they achieve a sufficient level of "associations". This type of graph construction is called an *association network* and is frequently used in biological networks. There are, of course, a number of choices for mathematical definitions of "association" in practice, such as Pearson correlation, partial correlation, etc. According to different definitions of "association", different graph models have been proposed, such

as correlation graphs, partial correlation graphs, Gaussian graphical model graphs, and others.

In Chapter 2, we also apply our anomalous community detection methodology on a time series of association networks of zebrafish neuronal activity. The raw dataset is a $D = n \times T$ multiple time series where $n(= 5,379)$ is the number of zebrafish neurons and $T(= 5,000)$ is the number of total time steps. $D_{i,t}$ records the activity level of neuron i at time stamp t. After data munging and aggregation, we discretize the data into 99 time chunks and construct a thresholded correlation graph in each time chunk. Details about the construction of graphs is also given in [36]. For this time series of Pearson correlation graphs, it is of interest to see whether the alarmed detection time provided by our methodology matches well with the time of the real olfactoric stimulus given by the lab scientist.

1.3.4 Information Networks

Information networks are networks describing relationships among elements of information. The relationship between elements of information includes citations between academic publications, co-authorships between academic researchers, and so on. The most celebrated example is the World Wide Web, which can be seen as a graph by representing web pages as vertices and the referencing of one page by another page as edges. Community detection on web graphs is of interest for many purposes. One goal is to identify clusters created by link farms, discouraging unfair competition on search engine ranks [10].

However, most community detection algorithms scale poorly on massive web graphs, particularly given the billions of vertices and edges characteristic of modern systems. In Chapter 3, we will introduce an active community detection framework by focusing only on significant web pages and communities formed by significant web pages. To demonstrate the practicality and efficiency of our method, we will conduct experiments on the largest public real-world hyperlink graph to date [28]. The page-level hyperlink graph covers 3.5 billion web pages and 128 billion hyperlinks between pages. According to our findings, we can answer questions such as "How does the content of important web pages induce clustering on the WWW?" and "Is any suspicious link farm found on the WWW?". To the best of our knowledge, this is the first community detection algorithm applied to a real information network dataset at this scale.

1.4 Random Graph Models

A random graph is a graph with $|V|$ fixed vertices and $|E|$ edges generated at random, where the edge set E satisfies some distribution over all possible edge sets. Mathematically, a random graph $A : \Omega \mapsto \mathcal{A}$ is a map from the probability sample space to the space of all adjacency matrices on n vertices. Different random graph models produce different probability distributions on graphs. For instance, the most well-known is the *Erdős Rényi graph*, denoted by $G(n, p)$, in which every possible edge occurs independently with a probability of $0 < p < 1$. That is, for any realized $A^\star \in \mathcal{A}$ and given p,

$$P[A = A^\star] = \prod_{u<v} p^{A^\star_{u,v}}(1-p)^{(1-A^\star_{u,v})}.$$

In this section, we briefly summarize several generalized models of the *Erdős Rényi graph*: the latent position model of [18], the dot product model of [54], and the stochastic blockmodel of [19] and [52]. This is because, in Chapter 2, we formulate anomalous community/change-point detection as a hypothesis testing problem in terms of a generative latent position model, focusing on the special case of the stochastic block model time series, and in Chapter 3, we demonstrate the validity of our method with synthetic stochastic block model graphs. Note that we consider only undirected graphs for the introduction of these models below.

1.4.1 Latent Position Model

The *latent position model* (LPM) is motivated by the assumption that each vertex v is associated with a K-dimensional latent random vector X_v. For any pair of vertices

u and v, conditioned on the two latent positions X_u and X_v, the existence of an edge between u and v is independently determined by a Bernoulli trial with a probability of $f(X_u, X_v)$ where f is a symmetric kernel function $f : \mathbb{R}^K \times \mathbb{R}^K \to [0, 1]$. Namely, $A_{u,v}|(X_u, X_v) \overset{ind}{\sim} Bernoulli(f(X_u, X_v))$. Thus, for any latent position graph with latent positions $\{X_v\}_{v \in V}$, kernel function f, and realized $A^\star \in \mathcal{A}$, we have

$$P[A = A^\star | \{X_i\}_{i \in V}] = \prod_{u<v} f(X_u, X_v)^{A^\star_{u,v}} (1 - f(X_u, X_v))^{(1-A^\star_{u,v})}.$$

1.4.2 Random Dot Product Model

The *random dot product graph model* (RDPM) [54] is a special case of the latent position model. In the random dot product graph model, the kernel function f is specified as the Euclidean inner product, i.e., $f(X_u, X_v) = \langle X_u, X_v \rangle$. In addition, for each vertex v, the latent random vector X_v takes its values in the unit simplex $\mathscr{S}$ so that $0 \leq \langle X_u, X_v \rangle \leq 1$ where $\mathscr{S} = \{x \in [0, 1]^K : \sum_{k=1}^K x_k \leq 1\}$. Thus, for any latent position graph with latent positions $\{X_v\}_{v \in V} \in \mathscr{S}$ and realized $A^\star \in \mathcal{A}$, we have

$$P[A = A^\star | \{X_i\}_{i \in V}] = \prod_{u<v} < X_u, X_v >^{A^\star_{u,v}} (1 - < X_u, X_v >)^{(1-A^\star_{u,v})}.$$

Hence, $X = (X_1, X_2, \ldots, X_{|V|})$ parametrizes the random dot product model. In this case, an adjacency matrix $A \sim RDPM(X)$.

1.4.3 Stochastic Block Model

The *stochastic block model* (SBM) of [19,52] is a random graph model in which each vertex is randomly assigned a block membership among $\{1, \ldots, B\}$, according to a membership

assignment function $\kappa : [n] \mapsto \{1, \ldots, B\}$, where B is the number of blocks. Given block memberships, the connectivity probabilities among all vertices are characterized by a $B \times B$ symmetric block connectivity matrix $\mathbf{P}$ where $\mathbf{P}_{j,k}$ denotes the block connectivity probability between blocks j and k. Namely, $A_{u,v} \overset{ind}{\sim} Bernoulli(\mathbf{P}_{j,k})$ given $u \in [n_j]$ and $v \in [n_k]$. Thus, for any stochastic block model with a block membership function κ, a block connectivity probability matrix B, and a realized $A^\star \in \mathcal{A}$, we have

$$P[A = A^\star | \mathbf{P}] = \prod_{u<v} \mathbf{P}_{\kappa(u),\kappa(v)}^{A^\star_{u,v}} (1 - \mathbf{P}_{\kappa(u),\kappa(v)})^{(1-A^\star_{u,v})}.$$

In the latent position model setting, if we add a block membership function $\kappa : [n] \mapsto \{1, \ldots, B\}$ and $\{\xi_i\}_{i=1}^B \in \mathcal{X}$ such that, for any $v \in V$, there exists $\xi_{\kappa(v)} \in \{\xi_1, \ldots, \xi_B\}$ with $X_v = \xi_{\kappa(v)}$, the model is then accommodated to a stochastic block model with $A_{u,v} \overset{ind}{\sim} Bernoulli(\mathbf{P}_{j,k})$ where $\mathbf{P}_{j,k} = f(\xi_{\kappa(u)}, \xi_{\kappa(v)})$. Similarly, under the same setting, a random dot product model can be represented as a stochastic block model with $A_{u,v} \overset{ind}{\sim} Bernoulli(\mathbf{P}_{j,k})$ where $\mathbf{P}_{j,k} =< \xi_{\kappa(u)}, \xi_{\kappa(v)} >$.

In the next chapter, we will assume that the time series of random graphs $\{G_t\}$ are generated according to a stochastic block model where the block membership of the vertices are randomly assigned at the initial time t_0. Then, at each subsequent time, G_t follows a SBM with a $B \times B$ probability matrix $\mathbf{P}_t$, conditioned on the initial block membership at time t_0. Under this model, the graphs are conditionally independent over time, the conditioning being on the block membership of the vertices. This assumption leads to a time series of graphs where the graphs are "weakly" dependent; i.e., they are dependent only on the block membership of the vertices at the initial time t_0.

1.5 Overview of Contributions

In summary, this dissertation contributes by presenting the following two research developments of community detection using locality statistics.

1. Using locality statistics, we can perform anomalous community detection in a time series of graphs. We formulate the task as a hypothesis-testing problem in terms of a generative latent position model, focusing on the special case of the stochastic block model time series. We analyze two classes of scan statistics, based on distinct underlying locality statistics presented in the literature. Our main contribution is the derivation of the limiting properties and power characteristics of the competing scan statistics. Performance is compared theoretically and on synthetic data. We demonstrate that both statistics are admissible in one simple setting, while one of the statistics is inadmissible in a second setting. In addition, practicality is demonstrated via application on an Enron email corpus dataset and a zebrafish neuronal activity dataset.

2. Using locality statistics, we can perform active community detection on a static but massive graph on which many community detection algorithms scale poorly. We propose a novel framework for detecting active communities that consist of the most active vertices in massive graphs. This framework is applicable to graphs consisting of billions of vertices and hundreds of billions of edges. Our framework utilizes a parallelizable trimming algorithm based on a locality statistic to filter out inactive vertices, and then clusters the remaining active vertices via spectral decomposition on their similarity matrix. We demonstrate the validity of our method

with synthetic stochastic block model graphs, using the Adjusted Rand Index as the performance metric. We further demonstrate its practicality and efficiency on a real-world hyperlink web graph consisting of over 3.5 billion vertices and 128 billion edges.

Chapter 2

Anomalous Community Detection in a Time Series of Graphs

In this chapter, we approach the dynamic anomalous community/change-point detection problem through the use of locality-based scan statistics. Scan statistics are commonly used in signal processing to detect a local signal in an instantiation of some random field [15, 23]. The idea is to scan over a small time or spatial window of the data and calculate some locality statistic for each window. The maximum of these locality statistics is known as the scan statistic. Large values of the scan statistic suggests the existence of non-homogeneity, such as a local region with significantly excessive communications. We refer to this local region as a *anomalous community* at the change-point. Under some homogeneity hypothesis, change-point detection can then be reduced to statistical hypotheses testing (c.f. § 2.2) using scan statistics. For example, [1] built a simple testing framework with the null hypothesis being Erdös-Rényi and the alternative hypothesis being a graph containing an unusually dense subgraph. In the static graph setting,

detection boundaries and conditions are given in [1] such that the scan statistics they specified for the testing were non-negligibly powerful. To capture anomalies (e.g., hacker attacks) in computer networks, [31] employed scan statistics through two shapes of locality statistics: 'star' and 'k-path'. The power properties of 'star' as a locality measure will be further explored in § 2.4.1 here.

In this chapter, we identify excessive communication activity in an anomalous community of a dynamic network by employing the scan statistics $S_{\tau,\ell,k}(t;\cdot)$ defined in § 2.3, with τ denoting the number of vertex-standardization steps, ℓ denoting the number of temporal-normalization steps, and k denoting local neighborhood distance. We consider two variations of $S_{\tau,\ell,k}(t;\cdot)$, namely $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$, where Ψ and Φ are two related but distinct locality statistics. The use of the locality statistics Ψ and Φ is based upon earlier the work of [38] and [48]. In particular, Ψ was introduced in [38] to detect the emergence of local excessive activities in a time series of Enron graphs whereas Φ was proposed in [48] to detect communication pattern changes in their departmental email network. Using the locality statistic Ψ, [35] constructed fusion statistics of graphs for anomaly detection, while [40] presented an analysis of the Enron data-set to illustrate statistical inference for attributed random graphs. However, all these cited works are mostly empirical in nature and do not provide much theoretical analysis of these locality-based scan statistics. Under the assumption that the time series of graphs is stationary before a change point, we demonstrate in the following that, for $\tau = 1$ and $\ell = 0$, the limiting $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ are the maximum of random variables which, under proper normalizations, follow a standard Gumbel $\mathcal{G}(0,1)$ distribution in the limit. Through these limiting properties, a comparative power analysis between $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ for $\tau = 1$ and $\ell = 0$ is performed. We demonstrate that both Ψ and Φ are admissible if

$k = 0$, while Ψ is inadmissible if $k = 1$.

This chapter is structured as follows. We discuss a generative model for a time series of graphs in § 2.1. The problem of anomalous community change-point detection is formulated in § 2.2. The formulation associates a change-point in the time series with changes in the underlying generative model. We introduce in § 2.3 two closely related notions of the locality statistic, Ψ and Φ, and their corresponding scan statistics $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$. Note that the two proposed scan statistics are motivated by our change-point problem formulation, but not limited to use under the model in § 2.1. In fact, practitioners can utilize the scan statistics in § 2.3 to detect the emergence of a "chatty" subgroup in any time series of graphs, not necessarily having to model the time series of graphs into a stochastic block model beforehand. As a side product of the proposed scan statistics, we then present how to locate the anomalous community by employing scan statistics at the change-point. As a key contribution in this dissertation, under the model in § 2.1, the limiting properties and power characteristics for some representative instances of $S_{\tau,\ell,k}(t;\cdot)$ are given in § 2.4.1 and 2.4.2, while some experimental results regarding locality-based statistics on synthetic data are also included.

2.1 Time Series of Random Graphs from Stochastic Block Model

In this section, we discuss a generative model, based on stochastic block partitioning, for time series of graphs. We shall assume that the time series of random graphs $\{G_t\}$ are generated according to a stochastic block model where the block membership of the vertices are fixed across time while the connectivity probabilities matrix $\mathbf{P} = \mathbf{P}_t$ may vary with time (c.f. our formulation of the change-point detection problem in § 2.2). That is to say, at some initial time, say $t_0 = 0$, we randomly assign each vertex to a block membership among $\{1, 2, \ldots, B\}$. Then at each subsequent time $t \geq t_0$, G_t follows a SBM with a $B \times B$ probability matrix $\mathbf{P}_t$, conditioned on the initial block membership at time t_0. Under this model, the graphs are *conditionally* independent over time, the conditioning being on the block membership of the vertices. This assumption on the generative model for the $\{G_t\}$ leads to a time series of graphs where the graphs are "weakly" dependent, i.e., they are dependent only on the block membership of the vertices at the initial time t_0. If, instead, for each time t, we resample the vertices' block membership for G_t then the resulting time series of graphs is independent.

Our construction of a time series of graphs in terms of the SBM as outlined above is a limiting case of the following model constructed using the random dot product graphs[1] .

[1]The Dirichlet distribution is a multivariate generalization of the beta distribution and corresponds to a distribution of points in the unit simplex. The Dirichlet distribution, Dirichlet($\vec{\alpha}$), $\vec{\alpha} = (\alpha_1, \ldots, \alpha_K), \alpha_j > 0, 1 \leq j \leq K$, has density $f_{\vec{\alpha}}(x_1, \ldots, x_K) = \frac{\Gamma(\sum_{j=1}^K \alpha_j)}{\prod_{j=1}^K \Gamma(\alpha_j)} \prod_{j=1}^K x_j^{\alpha_j - 1}, 0 < x_j < 1, \sum_{j=1}^K x_j = 1$.

1. For each $v \in [n]$ and $t \in \mathbb{N}$,

$$X_v(t) \sim \text{Dirichlet}(r_v \vec{\alpha}_v + \vec{1}).$$

2. For each $t \in \mathbb{N}$ and pair of vertices (u, v),

$$P(A_{u,v} = 1 | \, \mathbf{X}(t)) = \langle X_u(t), X_v(t) \rangle.$$

where $\vec{\alpha}_v \in \mathscr{S}$ is a fixed location parameter for the Dirichlet distribution and r_v is the concentration parameter that will be explained now.

It is worthwhile to note that $r_v = 0$ for all $v \in [n]$ means all vertices follow the same probabilistic behavior (uniform on the simplex) and $\min_{v \in V} r_v \to \infty$ implies that $X_v(t)$ has a point mass distribution at $\vec{\alpha}_v$ for each vertex. In the case $\min_{v \in V} r_v \to \infty$, the random dot product model can be further reduced to the stochastic block model (SBM) by letting vertices sharing the same $\vec{\alpha}_v$ share the same block membership. Next, we re-denote by $\vec{\alpha}_i$ the common Dirichlet location parameter corresponding to block $[n_i]$ and V is partitioned into B distinct blocks $[n_1], \ldots, [n_B]$ if there are B distinct $\vec{\alpha}_i$'s in total. Accordingly, as $\min r_i \to \infty$, $P(A_{u,v} = 1 | \, u \in [n_j], v \in [n_k]) \to \langle \vec{\alpha}_j, \vec{\alpha}_k \rangle$. We note that the above Dirichlet can be viewed as generating a time-series of graphs where the graphs are also "weakly" dependent, e.g., dependency between graphs at time t and t' being on the location and concentration parameters $\{(\alpha_v, r_v)\}$ for the vertices. Other generalizations of the above construction for generating time series of graphs are also possible. See, e.g., [24, 25] for examples of constructions where the time series of graphs depends on some underlying latent stochastic processes.

2.2 Anomalous Community Detection Problem In Stochastic Block Model Formulation

An important inference task in time series analysis is the problem of anomaly or change-point detection. An anomaly is broadly interpreted to mean deviation from a "normal" pattern and a *change-point* is the time-window during which the anomalous deviation occurs. For example, in social networks, we usually represent a time-evolving collection of emails, phone calls, web pages visits, etc. as a time series of graphs $\{G_t\}$ and we want to infer, from $\{G_t\}$, if there exists anomalous activities e.g., excessive phone calls among a subgroup in the network. After identifying a change-point, we will locate the anomalous subgroup having excessive communications at the change-point. We refer to the subgroup as an *anomalous community* at the change-point. In the detection problem described below in § 2.2 and its theoretical analysis presented in § 2.4.1 and § 2.4.2, we shall implicitly assume, for ease of exposition, that the $\{G_t\}$ are independent. As we pointed out in our discussion of the generative model for time-series of graphs in § 2.1, this independence corresponds to conditioning on the right parameters. In the setup of our theoretical analysis in this chapter, this corresponds to conditioning on the block membership of the vertices, which are fixed in time. Related discussions in the context of the latent process models of [24, 25] are given in § 5.

Statistically speaking, we want to test, for an unknown but non-random $t \in \mathbb{N}$, the null hypothesis H_0 that t is not a change-point against the alternative hypothesis H_A that

t is a change-point. There are many different ways to formulate the notion that t is a change-point. The following formulation, in the context of our discussion, is reasonable and sufficiently general and forms the basis of our subsequent investigation.

We say that t^* is a change-point for $\{G_t\}$ if there exists distinct choices of $\mathbf{P}^0$, $\mathbf{P}^A$ independent of t such that

$$H_A : G_t \sim \begin{cases} \text{SBM}(\mathbf{P}^0, \{[n_i]\}) & \text{for } t \leq t^* - 1 \\ \\ \text{SBM}(\mathbf{P}^A, \{[n_i]\}) & \text{for } t \geq t^* \end{cases},$$

where $\text{SBM}(\mathbf{P}, \{[n_i]\})$ denote the stochastic blockmodel with block connectivity probabilities $\mathbf{P}$ and unknown, but fixed in time, block memberships $\{[n_i]\}$. In contrast, the null hypothesis, i.e., the nonexistence of change-point, is

$$H_0 : G_t \sim \text{SBM}(\mathbf{P}^0, \{[n_i]\}) \text{ for all } t.$$

That is to say, under the alternative, at time t^*, a subset of the vertices change their behavior. The vertices whose behavior changes correspond to the vertices with block memberships whose corresponding rows in the connectivity matrix changes, i.e., from $\mathbf{P}^0$ to $\mathbf{P}^A$. As permutation of the vertex block labels does not affect our subsequent analysis, we will refer to $(t^*, \{[n_i]\}, \mathbf{P}^0, \mathbf{P}^A)$ as the change parameters. As a convention, if $t^* = \infty$, we assume all vertices follow their original dynamics for all t.

In the following, we discuss a specific form for $\mathbf{P}^0$ and $\mathbf{P}^A$, illustrating, albeit in an exaggerated manner, the chatter anomaly, i.e., a subset of vertices with altered communication behavior in an otherwise stationary setting.

$$\mathbf{P}^0 = \begin{pmatrix} p_1 & p_{1,2} & \cdots & \cdots & p_{1,B} \\ p_{2,1} & h_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_{B-1} & p_{B-1,B} \\ p_{B,1} & \cdots & \cdots & p_{B,B-1} & p_B \end{pmatrix}, \tag{2.2.1}$$

$$\mathbf{P}^A = \begin{pmatrix} p_1 & p_{1,2} & \cdots & \cdots & p_{1,B} \\ p_{2,1} & h_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_{B-1} & p_{B-1,B} \\ p_{B,1} & \cdots & \cdots & p_{B,B-1} & p_B + \delta \end{pmatrix}, \tag{2.2.2}$$

for some $\delta > 0$, with $n_1, n_2, \ldots, n_B$ being of size

$$(n_1, n_2, \ldots, n_B) = (\Theta(n), O(n), \ldots, O(n)).$$

For this form of $\mathbf{P}^0$ and $\mathbf{P}^A$, the blocks have their own (possibly distinct) self-connectivity probabilities which are diagonal entries of matrices. In other words, before the change-point, each of the blocks $i = 2$ up to $B-1$ have self-connectivity probability h_i. The block $i = 1$ is of size $\Theta(n)$ with self-connectivity probability p_1, representing the probabilistic behaviors of the vast majority of actors in a very large network. The case where $h_2 > p_1, \ldots, h_{B-1} > p_1$ is of interest because we can consider each of the $[n_i]$ as representing a "chatty" group for time $t \leq t^* - 1$, and at t^*, the previously non-chatty group $[n_B]$ becomes chatty if $p_B = p_1$, or the previously chatty group $[n_B]$ becomes even more chatty if $p_B > p_1$. See Figure 2.1 for a notional depiction of $\mathbf{P}^0$ and $\mathbf{P}^A$ for the case of $B = 3$

blocks with $p_1 = p_3 = p_{1,2} = p_{1,3} = p_{2,3} = p$. The detection of this transition for the vertices in $[n_B]$ is one of the main reasons behind the locality statistics that will be introduced in § 2.3.

Figure 2.1: Notional depiction of $\mathbf{P}^0$ and corresponding $\mathbf{P}^A$. $\mathbf{P}^0$: all vertices connect with probability p except that the self-connectivity probability of $[n_2]$ is h; $\mathbf{P}^A$: the self-connectivity probability of $[n_3]$ transitions from p to $p + \delta$ while $[n_2]$ retains its previous behavior.

2.3 Locality Statistics and Scan Statistics

In this section, we introduce two closely related notions of locality statistic, Ψ and Φ, and their corresponding scan statistics $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$. Large values of the smoothed scan statistic indicates a change-point where there is an excessive increase in communications among an anomalous community. By employing values of scan statistics at the identified change-point, we then present the way to locate the anomalous community.

2.3.1 Locality Statistics

Suppose we are given a time series of graphs $\{G_t\}_{t\geq 1}$ where $V(G_t)$ is independent of t, i.e., the graphs G_t are constructed on the same vertex set V. We now define two different but related locality statistics on $\{G_t\}$. For a given t, let $\Psi_{t;k}(v)$ be defined for all $k \geq 1$ and $v \in V$ by

$$\Psi_{t;k}(v) = |E(\Omega(N_k(v;G_t);G_t))|. \tag{2.3.1}$$

$\Psi_{t;k}(v)$ counts the number of edges in the subgraph of G_t induced by $N_k(v;G_t)$, the set of vertices u at a distance at most k from v in G_t. In a slight abuse of notation, we let $\Psi_{t;0}(v)$ denote the degree of v in G_t. The statistic Ψ_t was first introduced in [38]. [39] investigated the use of Ψ_t in analyzing the Enron data corpus.

Let t and t' be given, with $t' \leq t$. Now define $\Phi_{t,t';k}(v)$ for all $k \geq 1$ and $v \in V$ by

$$\Phi_{t,t';k}(v) = |E(\Omega(N_k(v;G_t);G_{t'}))|. \tag{2.3.2}$$

The statistic $\Phi_{t,t';k}(v)$ counts the number of edges in the subgraph of $G_{t'}$ induced by

$N_k(v; G_t)$.

Once again, with a slight abuse of notation, we let $\Phi_{t,t';0}(v)$ denote the degree of v in $G_t \cap G_{t'}$, where $G \cap G'$ for G and G' with $V(G) = V(G')$ denotes the graph $(V(G), E(G) \cap E(G'))$. The statistic $\Phi_{t,t';k}(v)$ is motivated by a statistic named the permanent window metric introduced in [47]. The permanent window metric was meant to capture events involving not just a single individual but the whole community. As the community at time t is assumed to be approximated by $N_k(v; G_t)$, the statistic $\Phi_{t,t';k}(v)$ uses the community structure at time t in its computation of the locality statistic at time $t' \leq t$. Through this measure, a community structure shift of v can be captured even when the connectivity level of v remains unchanged across time, i.e., when the Ψ_t stays mostly constant as t changes in some interval. With the purpose of determining whether t is a change-point, two kinds of normalizations based on past Ψ and Φ locality statistics and their corresponding normalized scan statistics are introduced in the next subsection.

2.3.2 Temporally-normalized statistics

Let $J_{t,t';k}$ be either the locality statistic $\Psi_{t';k}$ in Eq. (2.3.1) or $\Phi_{t,t';k}$ in Eq. (2.3.2), where for ease of exposition the index t is a dummy index when $J_{t,t';k} = \Psi_{t';k}$. We now define two normalized statistics for $J_{t,t';k}$, a vertex-dependent normalization and a temporal normalization. These normalizations and their use in the change-point detection problem are depicted in Figure 2.2.

For a given integer $\tau \geq 0$ and $v \in V$, we define the vertex-dependent normalization

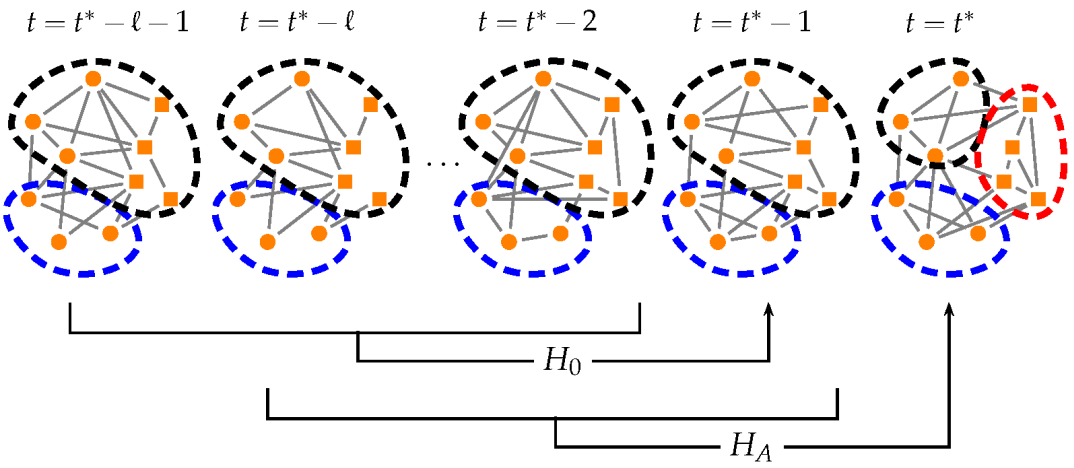


Figure 2.2: Temporal standardization: when testing for change at time t, the recent past graphs $G_t, G_{t-1}, \ldots$ are used to standardize the invariants.

$\widetilde{J}_{t,\tau;k}(v)$ of $J_{t,t';k}(v)$ by

$$\widetilde{J}_{t;\tau,k}(v) = \begin{cases} J_{t,t;k}(v) & \tau = 0 \\ J_{t,t;k}(v) - \hat{\mu}_{t;\tau,k}(v) & \tau = 1 \\ (J_{t,t;k}(v) - \hat{\mu}_{t;\tau,k}(v))/\hat{\sigma}_{t;\tau,k} & \tau > 1 \end{cases}, \tag{2.3.3}$$

where $\mu_{t;\tau,k}$ and $\sigma_{t;\tau,k}$ are defined as

$$\hat{\mu}_{t;\tau,k}(v) = \frac{1}{\tau} \sum_{s=1}^{\tau} J_{t,t-s;k}(v), \tag{2.3.4}$$

$$\hat{\sigma}_{t;\tau,k}(v) = \sqrt{\frac{1}{\tau - 1} \sum_{s=1}^{\tau} (J_{t,t-s;k}(v) - \hat{\mu}_{t;\tau,k}(v))^2}. \tag{2.3.5}$$

We then consider the maximum of these vertex-dependent normalizations for all $v \in V$, i.e., we define a $M_{\tau,k}(t)$ by

$$M_{\tau,k}(t) = \max_v (\widetilde{J}_{t,\tau;k}(v)). \tag{2.3.6}$$

We shall refer to $M_{\tau,0}(t)$ as the standardized max-degree and to $M_{\tau,1}$ as the standardized scan statistics. From Eq. (2.3.3), we see that the motivation behind vertex-dependent normalization is to standardize the scales of the raw locality statistics $J_{t,t';k}(v)$. Otherwise,

in Eq. (2.3.6), a noiseless vertex in the past who has dramatically increasing communications at the current time t would be inconspicuous because there might exist a talkative vertex who keeps an even higher but unchanged communication level throughout time.

Finally, for a given integer $\ell \geq 0$, we define the temporal normalization of $M_{\tau,k}(t)$ by

$$S_{\tau,\ell,k}(t) = \begin{cases} M_{\tau,k}(t) & \ell = 0 \\ M_{\tau,k}(t) - \tilde{\mu}_{\tau,\ell,k}(t) & \ell = 1 \\ (M_{\tau,k}(t) - \tilde{\mu}_{\tau,\ell,k}(t))/\tilde{\sigma}_{\tau,\ell,k}(t) & \ell > 1 \end{cases}, \tag{2.3.7}$$

where $\tilde{\mu}_{\tau,\ell,k}$ and $\tilde{\sigma}_{\tau,\ell,k}$ are defined as

$$\tilde{\mu}_{\tau,\ell,k}(t) = \frac{1}{\ell} \sum_{s=1}^{\ell} M_{\tau,k}(t-s), \tag{2.3.8}$$

$$\tilde{\sigma}_{\tau,\ell,k}(t) = \sqrt{\frac{1}{\ell - 1} \sum_{s=1}^{\ell} (M_{\tau,k}(t-s) - \tilde{\mu}_{\tau,\ell,k}(t))^2}. \tag{2.3.9}$$

The motivation behind temporal normalization, based on recent ℓ time steps, is to perform smoothing for the statistics $M_{\tau,k}$, similar to how smoothing is performed in time series analysis. Large values of the smoothed statistic indicates an anomaly where there is an excessive increase in communications among a subset of vertices. We will use these $S_{\tau,\ell,k}$ as the test statistics for the change-point detection problem described in § 2.2.

We note that because $\Psi_{t;k}(v) = \Phi_{t,t;k}(v)$ for $M_{\tau,k}$ when $\tau = 0$, the choice of locality statistic for $J_{t,t';k}$ does not matter when $\tau = 0$. For convenience of notation, since $S_{\tau,\ell,k}(t)$ is essentially a function of the $J_{t,t';k}$, we denote by $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ the $S_{\tau,\ell,k}(t)$ when the underlying statistic $J_{t,t';k}$ is $\Psi_{t';k}$ and $\Phi_{t,t';k}$, respectively.

After the above introduction of the temporally-normalized statistics $S_{\tau,\ell,k}(t;\cdot)$ with three parameters τ, ℓ, k , we now present a simple toy example to illustrate a key step in the

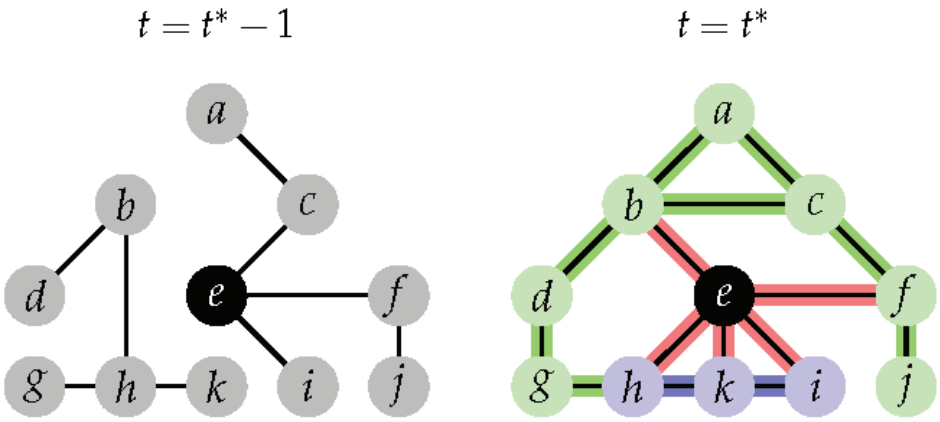
calculation of $S_{\tau,\ell,k}(t;\cdot)$, namely the calculation of the vertex-dependent normalization $\widetilde{J}_{t;\tau,k}(v)$ presented in Eq. (2.3.3). In Figure 2.3, the second table calculates $\widetilde{J}_{t^*;\tau,k}(v)$, when $\tau = 1$ and $v = e$, for different underlying statistics $J_{t,t';k}$ and different values of k. More concretely, because $\tau = 1$, $\widetilde{J}_{t^*;1,k}(e) = \Psi_{t^*;k}(e) - \Psi_{t^*-1;k}(e)$ if the underlying statistic is $\Psi_{t;k}(e)$ and $\widetilde{J}_{t^*;1,k}(e) = \Phi_{t^*,t^*;k}(e) - \Phi_{t^*,t^*-1;k}(e)$ if the underlying statistic is $\Phi_{t,t';k}(e)$.

2.3.3 Anomalous Community Identification

This section presents procedures to identify the anomalous community, according to a change-point $t^\star$ and scan statistics. Since $S_{\tau,\ell,k}(t;\cdot)$ is the running test statistic, $\{S_{\tau,\ell,k}(t;\cdot)\}_{t=1}^{\infty}$ is going to be a univariate time series recorded by the anomaly monitoring system. The system captures $t^\star$ as a change-point if $S_{\tau,\ell,k}(t^\star;\cdot)$ has a significant increment. After identifying the change-point $t^\star$, we return to Eq.(2.3.6), determining that

$$v^\star = \arg\max_v(\widetilde{J}_{t^\star,\tau;k}(v)). \tag{2.3.10}$$

Through the motivation behind the construction of temporally-normalized statistics, we know that a significant increase of $S_{\tau,\ell,k}(t^\star;\cdot)$ foreshadows a dramatic rise of the raw locality statistic value of $J_{t^*,t^*;k}(v^\star)$. $v^\star$ is the center of an anomalous subgroup of vertices, $N_k[v^\star; G_{t^\star}]$, that shows an excessive increase in communications at the change-point $t^\star$. In other words, it is the dramatic increase of communications in $N_k[v^\star; G_t]$ that gives rise to the upsurge of $S_{\tau,\ell,k}(t;\cdot)$ at $t^\star$ regardless of the selection of an underlying locality statistic. Thus, $N_k[v^\star; G_{t^\star}]$ is identified as an *anomalous community* in our detection framework.



	$\Psi_{t^*-1;k}(e)$	$\Psi_{t^*;k}$ (e)	$\Phi_{t^*,t^*-1;k}(e)$
$k=0$	3	5	2
$k=1$	3	7	4
$k=2$	5	5	9

	$J^k_{t^*,t^*}(e)$		$\widehat{\mu}^k_{t^*,\tau}(e)$		$\widetilde{J}^k_{t^*,\tau}(e)$	
	$k=0$	$k=1$	$k=0$	$k=1$	$k=0$	$k=1$
$\Psi_{t';k}$	5	7	3	3	2	4
$\Phi_{t,t';k}$	5	7	2	4	3	3

Figure 2.3: An example to differentiate the calculation of $\widetilde{J}_{t^*;\tau,k}(v)$ with varying underlying statistics ($\Psi_{t;k}$ or $\Phi_{t,t';k}$) and order distances ($k = 0$ or $k = 1$). In the right graph G_{t^*}, note that the red edges are $E(\Omega(N_{k=0}[e; G_{t^*}], G_{t^*}))$; the red and blue edges are $E(\Omega(N_{k=1}[e; G_{t^*}], G_{t^*}))$; the red, blue, and green edges are $E(\Omega(N_{k=2}[e; G_{t^*}], G_{t^*}))$. For instance, the magenta-marked number 3 is $\Psi_{t^*-1;0}$ where $\Psi_{t^*-1;0}(e) = |E(\Omega(N_0(e; G_{t^*-1}); G_{t^*-1}))|$ and $E(\Omega(N_0(e; G_{t^*-1}); G_{t^*-1})) = \{e \sim c, e \sim f, e \sim i\}$ in G_{t^*-1}.; the orange-marked number 4 is $\Phi_{t^*,t^*-1;1}(e)$ where $\Phi_{t^*,t^*-1;1}(e) = |E(\Omega(N_1(e; G_{t^*}); G_{t^*-1}))|$; and $E(\Omega(N_1(e; G_{t^*}); G_{t^*-1})) = \{h \sim k, b \sim h, e \sim i, e \sim f\}$ in G_{t^*-1}.

In summary, to capture an anomalous community in a time series of graphs $\{G_t\}_{t=1}^T$, we should follow the four steps below:

(i) Select an underlying locality statistic (Ψ or Φ) and parameters τ, ℓ, k.

(ii) Employ locality-based scan statistics $S_{\tau,\ell,k}(t^\star;\cdot)$ to detect a change-point $t^\star$ of $\{G_t\}_{t=1}^T$.

(iii) Determine $v^\star = \arg\max_v(\widetilde{J}_{t^\star,\tau;k}(v))$.

(iv) Identify the anomalous community $N_k[v^\star; G_{t^\star}]$.

2.4 Limiting properties and Power characteristics of Scan Statistics

2.4.1 Power Estimate of $S_{\tau=1,\ell=0,k=0}(t;\cdot)$

For algebraic simplicity, in Section 2.4.1 and 2.4.2, we consider a particularly simple form of $\mathbf{P}^0$ and $\mathbf{P}^A$ where

$$\mathbf{P}^0 = \begin{pmatrix} p & p & \dots & \dots & p \\ p & h_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_{B-1} & p \\ p & \dots & \dots & p & p \end{pmatrix}, \tag{2.4.1}$$

$$\mathbf{P}^A = \begin{pmatrix} p & p & \dots & \dots & p \\ p & h_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_{B-1} & p \\ p & \dots & \dots & p & p+\delta \end{pmatrix}. \tag{2.4.2}$$

With above form of $\mathbf{P}^0$ and $\mathbf{P}^A$, in this section, we will derive the limiting properties of $S_{1,0,0}(t;\Psi)$ and $S_{1,0,0}(t;\Phi)$ where $S_{1,0,0}(t;\Psi) = \max_v(\Psi_{t;0}(v) - \Psi_{t-1;0}(v))$ and $S_{1,0,0}(t;\Phi) = \max_v(\Phi_{t,t;0}(v) - \Phi_{t,t-1;0}(v))$. Theorem 1 below shows that in the limit $S_{\tau,\ell,k}(t;\cdot)$ is the maximum of random variables that converge to the standard Gumbel distributions $\mathcal{G}(0,1)$ under proper normalizations.

Theorem 1. *Let $\{G_t\}$ be a time series of random graphs according to the alternative H_A detailed in § 2.2. In particular, $G_t \sim SBM(\mathbf{P}^0, \{[n_i]\}_{i=1}^B)$ for $t \leq t^* - 1$ and $G_t \sim SBM(\mathbf{P}^A, \{[n_i]\}_{i=1}^B)$ for $t \geq t^*$ with $\mathbf{P}^0$ and $\mathbf{P}^A$ being of the form in Eq. (2.4.1) and Eq. (2.4.2), respectively. Let $S_{1,0,0}(t;\Psi)$ denote the statistic $S_{\tau,l,k}(t;\Psi)$ with $\tau = 1$, $l = 0$, and $k = 0$. Let $\mathcal{G}(\alpha,\gamma)$ denote the Gumbel distribution with location parameter α and scale parameter γ. For a given $n \in \mathbb{N}$, let a_n and b_n be given by*

$$a_n = \sqrt{2\log n}\Big(1 - \frac{\log\log n + \log 4\pi}{4\log n}\Big),$$
$$b_n = \frac{1}{\sqrt{2\log n}}.$$

Then as $n = \sum n_i \to \infty$, $S_{1,0,0}(t;\Psi)$ has following properties:

$$S_{1,0,0}(t;\Psi) = \max_{1\leq i\leq B} W_0(n_i;\Psi) \quad t < t^*, \tag{2.4.3}$$

$$S_{1,0,0}(t;\Psi) = \max_{1\leq i\leq B} W_A(n_i;\Psi) \quad t = t^*, \tag{2.4.4}$$

where

$$\frac{W_0(n_i;\Psi) - \mu_0(n_i;\Psi)}{\gamma_0(n_i;\Psi)} \xrightarrow{d} \mathcal{G}(0,1)$$
$$\frac{W_A(n_i;\Psi) - \mu_A(n_i;\Psi)}{\gamma_A(n_i;\Psi)} \xrightarrow{d} \mathcal{G}(0,1)$$

and the $\mu_0, \mu_A, \gamma_0, \gamma_A$ are given by

$$\mu_0(n_i;\Psi) = a_{n_i}\sqrt{Cnp(1-p)}$$
$$\gamma_0(n_i;\Psi) = b_{n_i}\sqrt{Cnp(1-p)}$$
$$\mu_A(n_i;\Psi) = \mu_0(n_i;\Psi) + \mathbf{1}_{\{i=B\}} n_B \delta$$
$$\gamma_A(n_i;\Psi) = \gamma_0(n_i;\Psi).$$

C is some explicit, computable constant.

Similarly, let $S_{1,0,0}(t;\Phi)$ *denote* $S_{\tau,l,k}(t;\Phi)$ *with* $\tau = 1$, $l = 0$, *and* $k = 0$. *Then as* $n = \sum n_i \to \infty$,

$$S_{1,0,0}(t;\Phi) = \max_{1\le i\le B} W_0(n_i;\Phi) \quad t < t^*, \tag{2.4.5}$$

$$S_{1,0,0}(t;\Phi) = \max_{1\le i\le B} W_A(n_i;\Phi) \quad t = t^*, \tag{2.4.6}$$

where

$$\frac{W_0(n_i;\Phi) - \mu_0(n_i;\Phi)}{\gamma_0(n_i;\Phi)} \xrightarrow{d} \mathcal{G}(0,1)$$
$$\frac{W_A(n_i;\Phi) - \mu_A(n_i;\Phi)}{\gamma_A(n_i;\Phi)} \xrightarrow{d} \mathcal{G}(0,1)$$

and the $\mu_0, \mu_A, \gamma_0, \gamma_A$ *in this case are*

$$\kappa(p) = p(1-p)(1-p(1-p))$$

$$\xi_0(n_i;\Phi) = \mathbf{1}_{\{i\notin\{1,B\}\}} n_i(h_i(1-h_i) - p(1-p))$$

$$\mu_0(n_i;\Phi) = a_{n_i}\sqrt{Cn\kappa(p)} + np(1-p) + \xi_0(n_i;\Phi)$$

$$\gamma_0(n_i;\Phi) = b_{n_i}\sqrt{Cn\kappa(p)}$$

$$\mu_A(n_i;\Phi) = \mu_0(n_i;\Phi) + \mathbf{1}_{\{i=B\}} n_B\delta(1-p)$$

$$\gamma_A(n_i;\Phi) = \gamma_0(n_i;\Phi).$$

Proof. Firstly, we investigate the case that the underlying locality statistic is Ψ. We will derive the limiting property of $S_{1,0,0}(t;\Psi)$ for $t = t^*$ in some detail. The property of $S_{1,0,0}(t;\Psi)$ when $t < t^*$ can be derived in a similar manner.

As $\tau = 1$ and $\ell = 0$, for any t, we have $\widetilde{\Psi}_{t;1,0}(v) = \Psi_{t;0}(v) - \Psi_{t-1;0}(v)$ from Eq. (2.3.3)

and Eq. (2.3.4). Without loss of generality, let us assume $v \in [n_i]$ and divide $\Psi_{t;0}(v)$ into two parts with $t = t^*$ and $t = t^* - 1$:

$$\Psi_{t^*;0}(v) = X_1 + X_2$$

where $X_1 \sim Bin(n - n_i, p), X_2 \sim Bin(n_i - 1, \mathbf{P}^A_{i,i})$;

$$\Psi_{t^*-1;0}(v) = X_3 + X_4$$

where $X_3 \sim Bin(n - n_i, p), X_4 \sim Bin(n_i - 1, \mathbf{P}^0_{i,i})$.

Since G_{t^*-1} and G_{t^*} are independent, we have

$$\begin{aligned}
&\frac{\widetilde{\Psi}_{t^*;1,0}(v) - (n_i - 1)(\mathbf{P}^A_{i,i} - \mathbf{P}^0_{i,i})}{\sqrt{np(1-p)}} \\
=&\frac{\Psi_{t^*;0}(v) - [(n - n_i)p + (n_i - 1)\mathbf{P}^A_{i,i}]}{\sqrt{np(1-p)}} - \frac{\Psi_{t^*-1;0}(v) - [(n - n_i)p + (n_i - 1)\mathbf{P}^0_{i,i}]}{\sqrt{np(1-p)}} \\
=&\frac{X_1 - (n - n_i)p}{\sqrt{(n - n_i)p(1-p)}} \cdot \frac{\sqrt{(n - n_i)p(1-p)}}{\sqrt{np(1-p)}} - \frac{X_3 - (n - n_i)p}{\sqrt{(n - n_i)p(1-p)}} \cdot \frac{\sqrt{(n - n_i)p(1-p)}}{\sqrt{np(1-p)}} \\
&+ \frac{X_2 - (n_i - 1)\mathbf{P}^A_{i,i}}{\sqrt{(n_i - 1)\mathbf{P}^A_{i,i}(1 - \mathbf{P}^A_{i,i})}} \cdot \frac{\sqrt{(n_i - 1)\mathbf{P}^A_{i,i}(1 - \mathbf{P}^A_{i,i})}}{\sqrt{np(1-p)}} \\
&- \frac{X_4 - (n_i - 1)\mathbf{P}^0_{i,i}}{\sqrt{(n_i - 1)\mathbf{P}^0_{i,i}(1 - \mathbf{P}^0_{i,i})}} \cdot \frac{\sqrt{(n_i - 1)\mathbf{P}^0_{i,i}(1 - \mathbf{P}^0_{i,i})}}{\sqrt{np(1-p)}} \\
\xrightarrow{d}& \mathcal{N}(0,1) \cdot C_1 - \mathcal{N}(0,1) \cdot C_2 + \mathcal{N}(0,1) \cdot C_3 - \mathcal{N}(0,1) \cdot C_4 \\
\xrightarrow{d}& \mathcal{N}(0, C)
\end{aligned} \tag{2.4.7}$$

where

$$C_1 = C_2 = \lim_{n\to\infty} \sqrt{\frac{n-n_i}{n}},$$
$$C_3 = \frac{\sqrt{(n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^A_{i,i})}}{\sqrt{np(1-p)}},$$
$$C_4 = \frac{\sqrt{(n_i-1)\mathbf{P}^0_{i,i}(1-\mathbf{P}^0_{i,i})}}{\sqrt{np(1-p)}},$$
$$C = \sum_{i=1}^{i=4} C_i^2$$

Next, plug in $\mathbf{P}^A_{i,i}$ and $\mathbf{P}^0_{i,i}$ into Eq. (2.4.7), we obtain

$$\frac{\widetilde{\Psi}_{t^*;1,0}(v) - \mathbf{1}_{\{i=B\}} n_B \delta}{\sqrt{Cnp(1-p)}} \xrightarrow{d} \mathcal{N}(0,1),\ v \in [n_i]$$

We can show that the dependency among the$\{\widetilde{\Psi}_{t^*;1,0}(v)\}_{v\in V(G_t)}$ is negligible by showing that the correlation between any two of the $\widetilde{\Psi}_{t^*;1,0}(v)$ goes to 0 sufficiently fast as $n \to \infty$. For u and v in block $[n_i]$,

$$corr(\widetilde{\Psi}_{t^*;1,0}(u), \widetilde{\Psi}_{t^*;1,0}(v)) \leq \frac{1}{Cnp(1-p)} = O(\frac{1}{n})$$

Hence, the sample maximum of $\{Y_v\}_{v\in[n_i]}$ converges to the sample maximum of n_i i.i.d $\mathcal{N}(0,1)$ random variables where $Y_v = \dfrac{\widetilde{\Psi}_{t^*;1,0}(v) - \mathbf{1}_{\{i=B\}} n_B \delta}{\sqrt{Cnp(1-p)}}$ ([3], Theorem 3.1). Also, it is known that the sample maximum of i.i.d $\mathcal{N}(0,1)$ random variables weakly converges to the Gumbel distribution ([12], § 2.3). One then verifies that the composition of above weak convergences still holds (see e.g., proof of Proposition 5 in [45]) and we thus have

$$\frac{W_A(n_i;\Psi) - \mu_A(n_i;\Psi)}{\gamma_A(n_i;\Psi)} \xrightarrow{d} \mathcal{G}(0,1).$$

Eq.(2.3.6) and Eq. (2.3.7) then implies that

$$S_{1,0,0}(t^*;\Psi) = \max_{v\in[n]} \widetilde{\Psi}_{t^*;1,0}(v) = \max_{1\leq i\leq B} \{W_A(n_i;\Psi)\}.$$

That is, the maximum of $\widetilde{\Psi}_{t^*;1,0}(v)$ over all n vertices is equivalent to the maximum of $W_A(n_i;\Psi)$ over all B blocks where $W_A(n_i;\Psi)$ converges to $\mathcal{G}(0,1)$ under proper normalization.

Similarly, the case when $t < t^*$ can be derived through the same approaches above. The limiting property of $S_{1,0,0}(t;\Psi)$ with $t < t^*$ then has the form in Eq. (2.4.3) with variations of $\mu_0(n_i;\Psi)$ and $\gamma_0(n_i;\Psi)$ for the normalization of $W_0(n_i;\Psi)$.

We now consider the case where the underlying locality statistic being Φ. The derivation of limiting property of $S_{1,0,0}(t;\Phi)$ for $t = t^*$ is given below. The derivation of the limiting property of $S_{1,0,0}(t;\Phi)$ for $t < t^*$ is similar and can be obtained with minor changes.

Let's assume $v \in [n_i]$, from Eq.(2.3.2) to (2.3.4),

$$\Phi_{t^*,t^*;0}(v) = X_1 + X_2$$

where $X_1 \sim Bin(n - n_i, p), X_2 \sim Bin(n_i - 1, \mathbf{P}^A_{i,i})$ and

$$\Phi_{t^*,t^*-1;0}(v)|G_{t^*} = X_3 + X_4$$

where $X_3 \sim Bin(X_1, p), X_4 \sim Bin(X_2, \mathbf{P}^0_{i,i})$.

Because $\widetilde{\Phi}_{t^*;1,0}(v) = \Phi_{t^*,t^*;0}(v) - \Phi_{t^*,t^*-1;0}(v)$, $\widetilde{\Phi}_{t^*;1,0}(v)$ counts the number of edges, for vertex v, appearing in G_{t^*} but disappearing in G_{t^*-1}. Accordingly, the edge is independently counted with probability $\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})$ to neighbors in $[n_i]$ and $p(1-p)$ to neighbors in $[n]\backslash[n_i]$ respectively. That is,

$$\widetilde{\Phi}_{t^*;1,0}(v) = B_3 + B_4$$

where $B_3 \sim Bin(n - n_i, p(1-p)), B_4 \sim Bin(n_i - 1, \mathbf{P}^A_{i,i}(1 - \mathbf{P}^0_{i,i}))$.

By the central limit theorem, we have

$$\begin{aligned}
&\frac{\widetilde{\Phi}_{t^*;1,0}(v) - [(n-n_i)p(1-p) + (n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})]}{\sqrt{np(1-p)[1-p(1-p)]}} \\
=&\frac{B_3 - (n-n_i)p(1-p)}{\sqrt{(n-n_i)p(1-p)[1-p(1-p)]}} \cdot \frac{\sqrt{(n-n_i)p(1-p)[1-p(1-p)]}}{\sqrt{np(1-p)[1-p(1-p)]}} \\
&+ \frac{B_4 - (n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})}{\sqrt{(n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})[1-\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})]}} \\
&\cdot \frac{\sqrt{(n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})[1-\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})]}}{\sqrt{np(1-p)[1-p(1-p)]}} \\
\xrightarrow{d}&\mathcal{N}(0,1)\cdot C_1 + \mathcal{N}(0,1)\cdot C_2 \\
\xrightarrow{d}&\mathcal{N}(0,C)
\end{aligned} \tag{2.4.8}$$

where

$$\begin{aligned}
C_1 &= \lim_{n\to\infty}\sqrt{\frac{n-n_i}{n}}, \\
C_2 &= \lim_{n\to\infty}\frac{\sqrt{(n_i-1)\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})[1-\mathbf{P}^A_{i,i}(1-\mathbf{P}^0_{i,i})]}}{\sqrt{np(1-p)[1-p(1-p)]}}, \\
C &= \sum_{i=1}^{i=2} C_i^2.
\end{aligned}$$

Similarly, after plugging $\mathbf{P}^0_{i,i}$ and $\mathbf{P}^A_{i,i}$ into Eq. (2.4.8), we obtain

$$\frac{\widetilde{\Phi}_{t^*;1,0}(v) - np(1-p) - \xi_0(n_i;\Phi) - \mathbf{1}_{\{i=B\}}n_B\delta(1-p)}{\sqrt{Cnp(1-p)[1-p(1-p)]}} \xrightarrow{d} \mathcal{N}(0,1).$$

For locality statistic Φ, the dependency among $\{\widetilde{\Phi}_{t^*;1,0}(v)\}_{v\in[n]}$ is also negligible because

$$corr(\widetilde{\Phi}_{t^*;1,0}(u), \widetilde{\Phi}_{t^*;1,0}(v)) = \frac{cov(\widetilde{\Phi}_{t^*;1,0}(u), \widetilde{\Phi}_{t^*;1,0}(v))}{Cnp(1-p)[1-p(1-p)]} \leq \frac{1}{Cnp(1-p)[1-p(1-p)]} = O(\frac{1}{n})$$

Therefore by following the same procedures of reasoning the limiting distribution of $W_A(n_i;\Psi)$, we can also obtain

$$\frac{W_A(n_i;\Phi) - \mu_A(n_i;\Phi)}{\gamma_A(n_i;\Phi)} \xrightarrow{d} \mathcal{G}(0,1)$$

where $W_A(n_i;\Phi) = \max_{v\in[n_i]} \widetilde{\Phi}_{t^*;1,0}(v)$.

Thus, $S_{1,0,0}(t^*;\Phi)$ is the maximum of $W_A(n_i;\Phi)$ over B blocks as desired.

□

We note the following corollary to Theorem 1 for the case of $B = 3$ blocks.

Corollary 2. *Assume the setting in Theorem 1 with $B = 3$. Let $\alpha > 0$ be given. Let β_Φ be the power of the test statistic $S_{1,0,0}(t;\Phi)$ when $t = t^*$ for testing the hypothesis that t is a change point at a significance level of α. Similarly, let β_Ψ be the power of the test statistic $S_{1,0,0}(t;\Psi)$ when $t = t^*$ for testing the same hypothesis at the same significance level of α. Then, as $(n_1, n_2, n_3) = (\Theta(n), O(n), O(n))$, β_Φ, β_Ψ and α have the following relationship* [2]*:*

1. $n_3 = o(\sqrt{n})$ *implies* $\beta_\Phi = \alpha, \beta_\Psi = \alpha$.

2. $n_3 = \Omega(\sqrt{n})$ *implies* $\beta_\Psi > \alpha$.

3. $n_3 = \Theta(\sqrt{n}) = \Theta(n_2)$ *implies* $\beta_\Phi > \alpha$.

4. $n_3 = \omega(\sqrt{n}) = \Theta(n_2)$ *implies*

$$\begin{aligned} &\beta_\Phi = \alpha && \text{if } \lim_{n\to\infty} \tfrac{n_2(h(1-h)-p(1-p))}{n_3\delta(1-p)} > 1, \\ &\beta_\Phi > \alpha && \text{if } \lim_{n\to\infty} \tfrac{n_2(h(1-h)-p(1-p))}{n_3\delta(1-p)} \leq 1. \end{aligned}$$

5. $n_3 = \Omega(\sqrt{n}) = \omega(n_2)$ *implies* $\beta_\Phi > \alpha$.

[2] the significance level α in Corollary 2 and Proposition 4 represents the Type I error rate of the hypothesis testing. It is the probability of incorrectly rejecting the nonexistence of change-point.

6. $n_3 = \Omega(\sqrt{n}) = o(n_2)$ *implies*

$$\beta_\Phi = \alpha \text{ if } h + p < 1,$$

$$\beta_\Phi > \alpha \text{ if } h + p \geq 1.$$

Proof. The limiting distributions of $\widetilde{\Psi}_{t^*-1;1,0}(v)$ and $\widetilde{\Psi}_{t^*;1,0}(v)$ derived in the proof **Theorem 1** provides that, under H_0,

$$\frac{\widetilde{\Psi}_{t^*-1;1,0}(v) - 0}{\sqrt{Cnp(1-p)}} \xrightarrow{d} \mathcal{N}(0,1),\ v \in [n_i]$$

and, under H_A,

$$\frac{\widetilde{\Psi}_{t^*;1,0}(v) - \mathbf{1}_{\{i=3\}} n_3 \delta}{\sqrt{Cnp(1-p)}} \xrightarrow{d} \mathcal{N}(0,1),\ v \in [n_i]$$

Accordingly, the ratio of the shift in the mean, from null to alternative, over the standard deviation of $\widetilde{\Psi}_{t;1,0}(v)$ for each vertex would be $\dfrac{\mathbf{1}_{\{i=3\}} n_3 \delta}{\sqrt{Cnp(1-p)}}$. We obtain two relationships between β_Ψ and α on the basis of the order of n_3:

1. if $n_3 = o(\sqrt{n})$, the ratio approaches to 0 and thus implies $\beta_\Psi = \alpha$.

2. if $n_3 = \Omega(\sqrt{n})$, then $\exists k > 0$ such that $\dfrac{\mathbf{1}_{\{i=3\}} n_3 \delta}{\sqrt{Cnp(1-p)}} \geq k > 0$ as $n \to \infty$ which implies $\beta_\Psi > \alpha$.

Likewise, from **Theorem 1**, the limiting distributions of $\widetilde{\Phi}_{t;1,0}(v)$ under null and alternative respectively are

$$\frac{\widetilde{\Phi}_{t^*-1;1,0}(v) - np(1-p) - \xi_0(n_i;\Phi)}{\sqrt{Cnp(1-p)[1-p(1-p)]}} \xrightarrow{d} \mathcal{N}(0,1).$$

$$\frac{\widetilde{\Phi}_{t^*;1,0}(v) - np(1-p) - \xi_0(n_i;\Phi) - \mathbf{1}_{\{i=3\}} n_3 \delta(1-p)}{\sqrt{Cnp(1-p)[1-p(1-p)]}} \xrightarrow{d} \mathcal{N}(0,1).$$

The relationship between β_Φ and α is more involved when $\xi_0(n_2;\Phi)$ are included. In order to clarify the order dominance relationship between $\xi_0(n_2;\Phi)$ and $n_3\delta(1-p)$, there are five separate cases to be considered:

1. if $n_3 = o(\sqrt{n})$, as $n \to \infty$, $\widetilde{\Phi}_{t;1,0}(v)$ share the same mean and variance under both H_0 and H_A, thus $\beta_\Phi = \alpha$.

2. if $n_3 = \Theta(\sqrt{n}) = \Theta(n_2)$, $\dfrac{n_3\delta(1-p)}{\sqrt{Cnp(1-p)[1-p(1-p)]}}$ and $\dfrac{\xi_0(n_2;\Phi)}{\sqrt{Cnp(1-p)[1-p(1-p)]}}$ have the same order $\Theta(1)$ so that the increment, $\dfrac{n_3\delta(1-p)}{\sqrt{Cnp(1-p)[1-p(1-p)]}} = \Theta(1)$, is not negligible and implies $\beta_\Phi > \alpha$.

3. if $n_3 = \omega(\sqrt{n}) = \Theta(n_2)$, whether $\beta_\Phi > \alpha$ is determined by if $P(\arg\max \widetilde{\Phi}_{t;1,0}(v) \in [n_3])$ under H_A is larger than under H_0. In fact, if $\dfrac{\xi_0(n_2;\Phi)}{n_3\delta(1-p)} > 1$, $P(\arg\max \widetilde{\Phi}_{t;1,0}(v) \in [n_2]) = 1$ as $n \to \infty$ under both H_0 and H_A, hence $\beta_\Phi = \alpha$. Otherwise, $n_3\delta(1-p)$ in $[n_3]$ contributes to the power increment.

4. if $n_3 = \Omega(\sqrt{n}) = \omega(n_2)$, $n_3\delta(1-p)$ dominates $\xi_0(n_2;\Phi)$ in the limit thereby the location shift in block $[n_3]$ results in $P(\arg\max \widetilde{\Phi}_{t^*;1,0}(v) \in [n_3]) = 1$ and thus $\beta_\Phi > \alpha$.

5. if $n_3 = \Omega(\sqrt{n}) = o(n_2)$, whether $n_3\delta(1-p)$ leads to a power increment depends on the sign of $\xi_0(n_2;\Phi)$. If $h + p < 1$ such that $\xi_0(n_2;\Phi)$ being positive, $P(\arg\max \widetilde{\Phi}_{t;1,0}(v) \in [n_2]) = 1$ under both H_0 and H_A as $n \to \infty$ because $n_3 = o(n_2)$. On the contrary, if $h+p \geq 1$, $\xi_0(n_2;\Phi) < 0$ enables $P(\arg\max \widetilde{\Phi}_{t;1,0}(v) \in [n_3])$ to increase from H_0 to H_A. Thus, we have $\beta_\Phi = \alpha$ if $h + p < 1$; $\beta_\Phi > \alpha$ if $h + p \geq 1$.

□

From Corollary 2, an unanswered question is whether there exists a dominance between $S_{1,0,0}(t;\Psi)$ and $S_{1,0,0}(t;\Phi)$. By using Theorem 1, we now present an example to show that both statistics are admissible if we restrict the test statistic space to only two elements- $S_{1,0,0}(t;\Psi)$ and $S_{1,0,0}(t;\Phi)$. That is, neither statistic has a statistical power dominance.

Our setup is as follows. Let $p = 0.43$. For each pair $(h, p+\delta)$ satisfying $p < h < 1$ and $p < p+\delta < 1$, we generate a null and alternative hypothesis pair H_0 and H_A according to the model in § 2.2 with $B = 3$ blocks, i.e.,

$$\mathbf{P}^0 = \begin{pmatrix} 0.43 & 0.43 & 0.43 \\ 0.43 & h & 0.43 \\ 0.43 & 0.43 & 0.43 \end{pmatrix}, \mathbf{P}^A = \begin{pmatrix} 0.43 & 0.43 & 0.43 \\ 0.43 & h & 0.43 \\ 0.43 & 0.43 & p+\delta \end{pmatrix}.$$

with $n = n_1 + n_2 + n_3 = 1000$ and n_1, n_2, n_3 being functions of n, h and δ ($n_2 = n_3 = c_{p,h,\delta}\sqrt{n \log n}$ where the constant $c_{p,h,\delta}$ is dependent on p, h and δ). In order to compare sensitivities of $S_{1,0,0}(t;\Psi)$ and $S_{1,0,0}(t;\Phi)$ in detection, we then calculate $\beta_\Psi - \beta_\Phi$ by deriving the limiting property of $S_{1,0,0}(t;\Psi)$ using Eqs. (2.4.3) and (2.4.4) and the limiting property of $S_{1,0,0}(t;\Phi)$ using Eqs. (2.4.5) and (2.4.6). The result is illustrated in Figure 2.4 where we have plotted $\beta_\Psi - \beta_\Phi$ for different combinations of h and $q(= p+\delta)$. Figure 2.4 indicates that the two statistics $S_{1,0,0}(\cdot;\Psi)$ and $S_{1,0,0}(\cdot;\Phi)$ are both admissible because $S_{1,0,0}(t;\Phi)$ achieves a larger statistical power in the blue-colored region but a smaller power in the red-colored region.

We now analyze the use of Theorem 1 as a large-sample approximation to $S_{1,0,0}(t;\Phi)$ and $S_{1,0,0}(t;\Psi)$. From Figure 2.4 with $p = 0.43$, we choose a $(h, p+\delta)$ pair, with $\beta_\Psi - \beta_\Phi > 0.05$, namely $h = 0.95$ and $p + \delta = 0.98$. We then estimate the power of β_Φ and β_Ψ by repeated sampling of graphs from stochastic blockmodel with parameters, $(\mathbf{P}^0, n_1, n_2, n_3)$

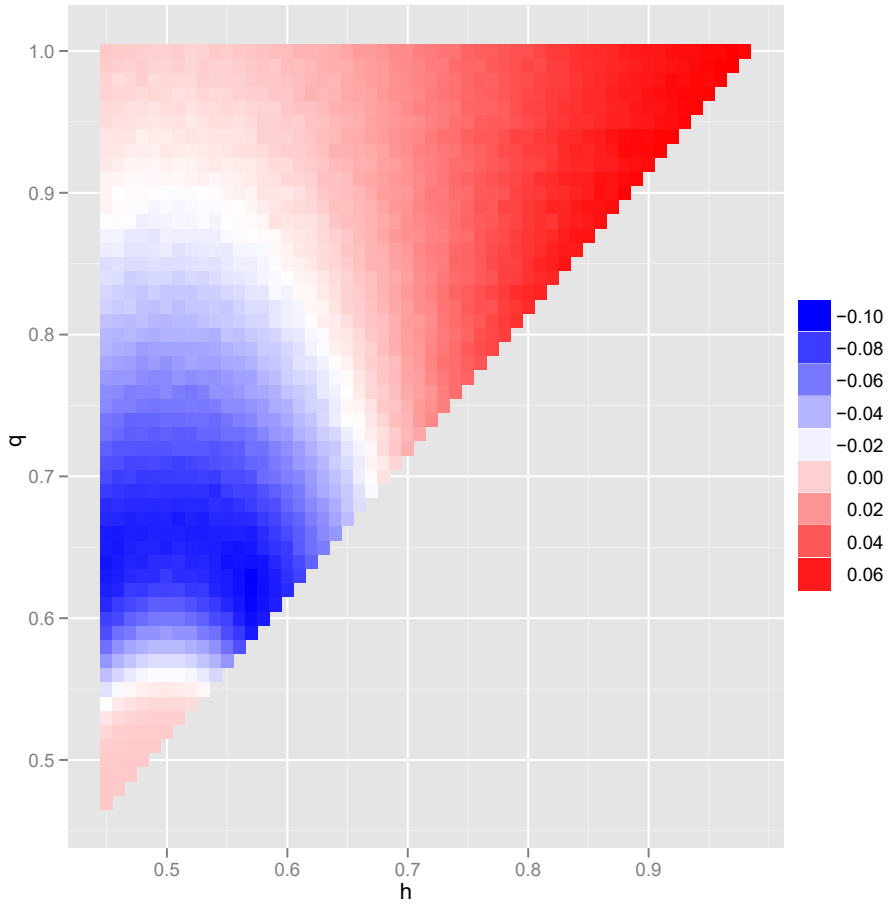


Figure 2.4: A comparison using the limiting properties of $S_{1,0,0}(t;\Psi)$ and $S_{1,0,0}(t;\Phi)$, of $\beta_\Psi - \beta_\Phi$ for different null and alternative hypotheses pairs as parameterized by h and $q(=p+\delta)$. The blue-colored region corresponds to values of h and $q(=p+\delta)$ for which $\beta_\Psi < \beta_\Phi$, while the red-colored region corresponds to values of h and $p+\delta$ with $\beta_\Psi > \beta_\Phi$.

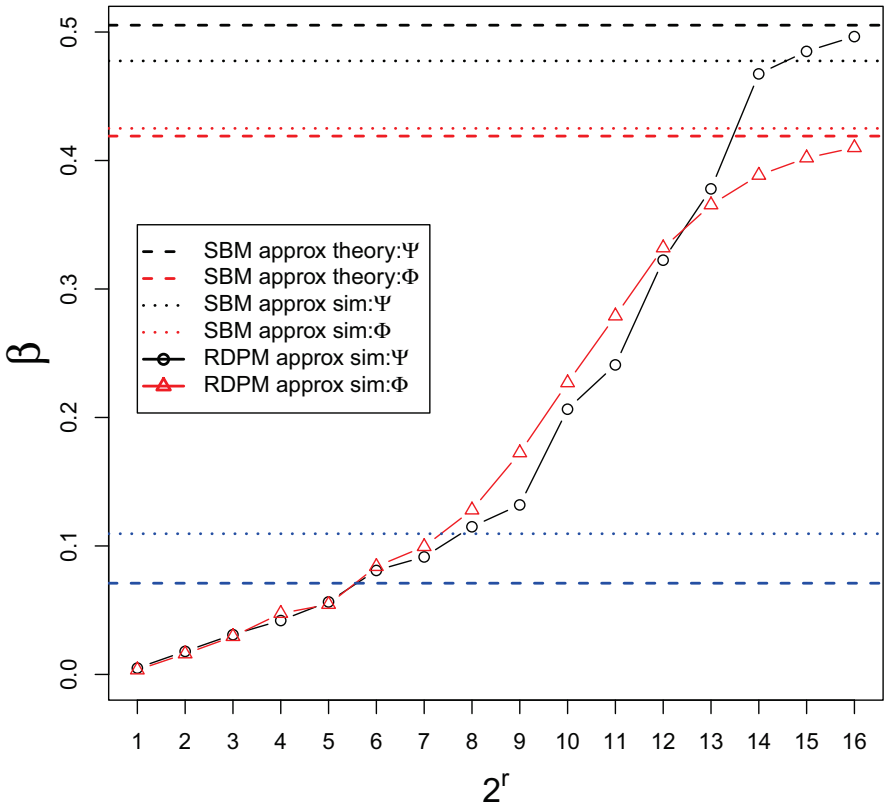


Figure 2.5: Power estimates β_Ψ against β_Φ using Monte Carlo simulation on random graphs from the stochastic blockmodel, Monte Carlo simulation on random graphs from the random dot product model, and large-sample approximation for the stochastic blockmodel. r is the concentration parameter. Dashed blue line: power estimate of large-sample approximation to $S_{0,0,0}(t;\Psi)$; dotted blue line: power estimate of SBM Monte Carlo simulation to $S_{0,0,0}(t;\Psi)$.

for the null distribution and $(\mathbf{P}^A, n_1, n_2, n_3)$ for the alternative distribution. The result is presented in Figure 2.5.

We see that the large-sample approximation obtained via Theorem 1 matches well with sampling from the stochastic blockmodel (SBM). Figure 2.5 also includes power estimates for the random dot product model (RDPM) with varying concentration parameter r and predetermined location parameters $\vec{\alpha}_1, \vec{\alpha}_2, \vec{\alpha}_3$. Specifically, $\vec{\alpha}_1, \vec{\alpha}_2, \vec{\alpha}_3$ are carefully chosen such that their Euclidean inner products match corresponding block connectivity

probabilities i.e., (p, h, q) specified above. We see that, as r increases, the power estimates for the random dot product model matches well with those of the stochastic blockmodel and large-sample approximation. Finally Figure 2.5 also includes power estimates for the locality statistics based on Φ and Ψ for $\tau = 0$, i.e., no vertex-dependent normalization and is equivalent to the use of the max degree statistic to test H_0 against H_A. These are represented as dashed and dot blue lines, corresponding to large-sample approximation and Monte Carlo simulations, respectively. Clearly, vertex-dependent normalization leads to better performance for this H_0 and H_A pair.

2.4.2 Power Estimate of $S_{\tau=1,\ell=0,k=1}(t;\cdot)$

In this section, we provide investigations of $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ with a larger scale parameter $k = 1$ instead of $k = 0$. We keep $\tau = 1$ and $\ell = 0$ the same as before and derive the limiting properties of $\max_v(\Psi_{t;1}(v) - \Psi_{t-1;1}(v))$ and $\max_v(\Phi_{t,t;1}(v) - \Phi_{t,t-1;1}(v))$. To make conclusions concise and presentable, firstly, we delve into the limiting properties in the model presented in § 2.2 with number of blocks $B = 3$.

Proposition 3. *Assume the same setting in Theorem 1 with $B = 3$. As $(n_1, n_2, n_3) = (\Theta(n), o(n), o(n))$ and $n \to \infty$, $S_{1,0,1}(t;\Psi)$ has the following properties:*

$$S_{1,0,1}(t;\Psi) = \max_{1\le i\le 3} W_0'(n_i;\Psi) \quad t < t^*,$$

$$S_{1,0,1}(t;\Psi) = \max_{1\le i\le 3} W_A'(n_i;\Psi) \quad t = t^*,$$

where

$$\frac{W_0'(n_i;\Psi) - \mu_0'(n_i;\Psi)}{\gamma_0'(n_i;\Psi)} \xrightarrow{d} \mathcal{G}(0,1)$$
$$\frac{W_A'(n_i;\Psi) - \mu_A'(n_i;\Psi)}{\gamma_A'(n_i;\Psi)} \xrightarrow{d} \mathcal{G}(0,1)$$

and the $\mu_0', \mu_A', \gamma_0', \gamma_A'$ *are given by*

$$\kappa'(n, p, n_2, h, i) = np^2 + 1 + \mathbf{1}_{\{i=2\}} n_2 p(h - p)$$

$$\mu_0'(n_i; \Psi) = \mu_0(n_i; \Psi)\kappa'(n, p, n_2, h, i)$$

$$\gamma_0'(n_i; \Psi) = \gamma_0(n_i; \Psi)\kappa'(n, p, n_2, h, i)$$

$$\zeta(n_3, p, \delta, i) = \frac{\delta}{2}[n_3^2(\mathbf{1}_{\{i\neq 3\}}p^2 + \mathbf{1}_{\{i=3\}}(p+\delta)^2) + n_3(\mathbf{1}_{\{i\neq 3\}}p(1-p) + \mathbf{1}_{\{i=3\}}(p+\delta)(1-p-\delta))]$$

$$\mu_A'(n_i; \Psi) = \mu_A(n_i; \Psi)[\kappa'(n, p, n_2, h, i) + \frac{\mathbf{1}_{\{i=3\}} n_3 p\delta}{2}] + \zeta(n_3, p, \delta, i)$$

$$\gamma_A'(n_i; \Psi) = \gamma_A(n_i; \Psi)[\kappa'(n, p, n_2, h, i) + \frac{\mathbf{1}_{\{i=3\}} n_3 p\delta}{2}].$$

Likewise,

$$S_{1,0,1}(t; \Phi) = \max_{1\leq i\leq 3} W_0'(n_i; \Phi)\ t < t^*,$$

$$S_{1,0,1}(t; \Phi) = \max_{1\leq i\leq 3} W_A'(n_i; \Phi)\ t = t^*,$$

where

$$\frac{W_0'(n_i; \Phi) - \mu_0'(n_i; \Phi)}{\gamma_0'(n_i; \Phi)} \xrightarrow{d} \mathcal{G}(0, 1)$$
$$\frac{W_A'(n_i; \Phi) - \mu_A'(n_i; \Phi)}{\gamma_A'(n_i; \Phi)} \xrightarrow{d} \mathcal{G}(0, 1)$$

and the $\mu_0', \mu_A', \gamma_0', \gamma_A'$ *are given by*

$$\eta(p) = p^3(1-p)$$

$$\xi_0(n_i; \Phi) = \mathbf{1}_{\{i=2\}} n_2(h(1-h) - p(1-p))$$

$$\mu_0'(n_i; \Phi) = a_{n_i}\sqrt{Cn^2\eta(p)} + np(1-p) + \xi_0(n_i; \Phi)$$

$$\gamma_0'(n_i; \Phi) = b_{n_i}\sqrt{Cn^2\eta(p)}$$

$$\zeta(n_3, p, \delta, i) = \frac{\delta}{2}[n_3^2(\mathbf{1}_{\{i\neq 3\}}p^2 + \mathbf{1}_{\{i=3\}}(p+\delta)^2) + n_3(\mathbf{1}_{\{i\neq 3\}}p(1-p) + \mathbf{1}_{\{i=3\}}(p+\delta)(1-p-\delta))]$$

$$\mu'_A(n_i;\Phi) = \mu'_0(n_i;\Phi) + \mathbf{1}_{\{i=3\}}n_3\delta(1-p) + \zeta(n_3, p, \delta, i)$$

$$\gamma'_A(n_i;\Phi) = \gamma'_0(n_i;\Phi)$$

Proof. We present a sketch of the proof based on arguments from [42] for the case where the underlying locality statistic is Ψ. The case where the underlying locality statistic is Φ follows from the proof of Theorem 5.

Let $v \in [n_i](i \in \{1,2,3\})$, locality statistics $\Psi_{t^*,t^*;1}(v)$ and $\Psi_{t^*,t^*-1;1}(v)$ are respectively decomposed as follows

$$\Psi_{t^*,t^*;1}(v) = X_i + \sum_{j\neq i} X_j + \sum_{j=1}^{3} Y_j + \sum_{1\leq j<k\leq 3} Z_{jk},\ v \in [n_i]$$

where

$$X_i \sim Bin(n_i - 1, \mathbf{P}^A_{i,i}),$$

$$X_j \sim Bin(n_j, \mathbf{P}^A_{i,j}),$$

$$Y_j|X_j \sim Bin(\binom{X_j}{2}, \mathbf{P}^A_{j,j}),$$

$$Z_{jk}|X_j, X_k \sim Bin(X_jX_k, \mathbf{P}^A_{j,k}).$$

and

$$\Psi_{t^*,t^*-1;1}(v) = X'_i + \sum_{j\neq i} X'_j + \sum_{j=1}^{3} Y'_j + \sum_{1\leq j<k\leq 3} Z'_{jk},\ v \in [n_i]$$

where

$$X'_i \sim Bin(n_i - 1, \mathbf{P}^0_{i,i}),$$

$$X'_j \sim Bin(n_j, \mathbf{P}^0_{i,j}),$$

$$Y'_j | X'_j \sim Bin(\binom{X'_j}{2}, \mathbf{P}^0_{j,j}),$$

$$Z'_{jk} | X'_j, X'_k \sim Bin(X'_j X'_k, \mathbf{P}^0_{j,k}).$$

Hence, when $\mathbf{P}^0$ and $\mathbf{P}^A$ are substituted, we have

$$\begin{aligned}
&\widetilde{\Psi}_{t^*;1,1}(v) \\
=&\Psi_{t^*,t^*;1}(v) - \Psi_{t^*,t^*-1;1}(v) \\
=&[(X_i + \sum_{j\neq i} X_j) - (X'_i + \sum_{j\neq i} X'_j)] + [(\sum_{j=1}^{3} Y_j + \sum_{1\leq j<k\leq 3} Z_{jk}) - (\sum_{j=1}^{3} Y'_j + \sum_{1\leq j<k\leq 3} Z'_{jk})] \\
=&[(X_i + \sum_{j\neq i} X_j) - (X'_i + \sum_{j\neq i} X'_j)] \cdot \Big[1 + \frac{p}{2}[(X'_i + \sum_{j\neq i} X'_j) + (X_i + \sum_{j\neq i} X_j)]\Big] \\
&+ \frac{h-p}{2}(X_2^2 - {X'_2}^2) + \frac{\delta}{2} X_3^2 \\
=&\widetilde{\Psi}_{t^*;1,0}(v) \cdot \Big[1 + \frac{p}{2}[(X'_i + \sum_{j\neq i} X'_j) + (X_i + \sum_{j\neq i} X_j)]\Big] + \frac{h-p}{2}(X_2^2 - {X'_2}^2) + \frac{\delta}{2} X_3^2
\end{aligned}$$

Thus, by using similar approaches given in the proof of lemma 3.2 and lemma 3.3 from [42], we obtain, as $n \to \infty$,

$$\arg max_{v\in[n_i]} \widetilde{\Psi}_{t^*;1,0}(v) = \arg max_{v\in[n_i]} \widetilde{\Psi}_{t^*;1,1}(v)$$

and

$$\lim P(W'_A(n_i;\Psi) > \mu'_A(n_i;\Psi)) = \lim P(W_A(n_i;\Psi) > \mu_A(n_i;\Psi))$$

where $W'_A(n_i;\Psi) = \max_{v\in[n_i]} \widetilde{\Psi}_{t^*;1,1}(v)$ and $W_A(n_i;\Psi) = \max_{v\in[n_i]} \widetilde{\Psi}_{t^*;1,0}(v)$. This leads to the fact that $(W'_A(n_i;\Psi) - \mu'_A(n_i;\Psi))/\beta'_A(n_i;\Psi)$ follows standard Gumbel distribution

$\mathcal{G}(0,1)$ and $S_{1,0,1}(t^*;\Phi) = \max_{1\le i\le 3} W'_A(n_i;\Phi)$. Similar arguments apply to $S_{1,0,1}(t^*-1;\Psi)$. □

Naturally, the limiting properties of $S_{1,0,1}(t;\Psi)$ and $S_{1,0,1}(t;\Phi)$ as given above offer the following power comparison result.

Proposition 4. *In the model shown in Figure2.1, Let $\alpha > 0$ be given, β'_Φ be the power of the test statistic $S_{1,0,1}(t;\Phi)$ when $t = t^*$ for testing the hypothesis that t is change point at a significance level of α and β'_Ψ be the power of the test statistic $S_{1,0,1}(t;\Psi)$ when $t = t^*$ for testing the same hypothesis at the same significance level of α. As $n \to \infty$, β'_Φ, β'_Ψ and α have the following relationship:*

1. *$n_3 = o(\sqrt{n})$ implies $\beta'_\Phi = \beta'_\Psi = \alpha$.*

2. *$n_3 = \Omega(\sqrt{n})$ implies $\beta'_\Phi \ge \beta'_\Psi > \alpha$.*

Consequently, Proposition 4 leads to the conclusion that the performance of $S_{1,0,1}(t;\Phi)$ dominates $S_{1,0,1}(t;\Psi)$ in the 3-block model. Moreover, this superiority can be generalized to the case with any given number of blocks $B \ge 3$. This is because each block $[n_i]$ with $1 < i < B$ in B-blocks model follows a similar probabilistic behavior as block $[n_2]$ in 3-blocks model while the power of hypothesis testing is otherwise determined by the change of probabilistic behavior of block $[n_B]$. In the limiting condition with $n \to \infty$, both β'_Φ and β'_Ψ in B-blocks model can be characterized as a function of p, δ, n_B only. In other words, though $h_2 > p, \ldots, h_{B-1} > p$, the "chatty" groups $[n_2], \ldots, [n_{B-1}]$ do not make any contribution on β'_Φ or β'_Ψ. Hence, the number of "chatty groups", namely $B-2$, is independent of the fact of dominance of $S_{1,0,1}(t;\Phi)$. Due to the superiority of $S_{1,0,1}(t;\Phi)$, only the limiting properties of $S_{1,0,1}(t;\Phi)$ in the general B-block model is given below.

Theorem 5. *Let $\{G_t\}$ be a time series of random graphs according to the alternative H_A detailed in § 2.2. In particular, $G_t \sim SBM(\mathbf{P}^0, \{[n_i]_{i=1}^B\})$ for $t < t^*$ and $G_t \sim SBM(\mathbf{P}^A, \{[n_i]_{i=1}^B\})$ for $t \geq t^*$ with $\mathbf{P}^0$ and $\mathbf{P}^A$ being of the form in Eq. (2.2.1) and Eq. (2.2.2), respectively. Let $S_{1,0,1}(t;\Phi)$ denote the statistic $S_{\tau,l,k}(t;\Phi)$ with $\tau = 1$, $l = 0$, and $k = 1$.*

Then as $n = \sum n_i \to \infty$, $S_{1,0,1}(t;\Phi)$ has the following properties:

$$S_{1,0,1}(t;\Phi) = \max_{1\leq i\leq B} W_0'(n_i;\Phi) \quad t < t^*,$$

$$S_{1,0,1}(t;\Phi) = \max_{1\leq i\leq B} W_A'(n_i;\Phi) \quad t = t^*,$$

where

$$\frac{W_0'(n_i;\Phi) - \mu_0'(n_i;\Phi)}{\gamma_0'(n_i;\Phi)} \xrightarrow{d} \mathcal{G}(0,1)$$
$$\frac{W_A'(n_i;\Phi) - \mu_A'(n_i;\Phi)}{\gamma_A'(n_i;\Phi)} \xrightarrow{d} \mathcal{G}(0,1)$$

and the $\mu_0', \mu_A', \gamma_0', \gamma_A'$ are given by

$$\eta(p) = p^3(1-p)$$

$$\xi_0(n_i;\Phi) = \mathbf{1}_{\{i\notin\{1,B\}\}} n_i(h_i(1-h_i) - p(1-p))$$

$$\mu_0'(n_i;\Phi) = a_{n_i}\sqrt{Cn^2\eta(p)} + np(1-p) + \xi_0(n_i;\Phi)$$

$$\gamma_0'(n_i;\Phi) = b_{n_i}\sqrt{Cn^2\eta(p)}$$

$$\zeta(n_B,p,\delta,i) = \frac{\delta}{2}[n_B^2(\mathbf{1}_{\{i\neq B\}}p^2 + \mathbf{1}_{\{i=B\}}(p+\delta)^2) + n_B(\mathbf{1}_{\{i\neq B\}}p(1-p) + \mathbf{1}_{\{i=B\}}(p+\delta)(1-p-\delta))]$$

$$\mu_A'(n_i;\Phi) = \mu_0'(n_i;\Phi) + \mathbf{1}_{\{i=B\}} n_B\delta(1-p) + \zeta(n_B,p,\delta,i)$$

$$\gamma_A'(n_i;\Phi) = \gamma_0'(n_i;\Phi)$$

Before proving Theorem 5, we state and prove a technical lemma on the correlations among the $\{\widetilde{\Phi}_{t;1,1}(v)\}$.

Lemma 6. *Let G_{t-1} and G_t be two independent Erdös-Rényi graphs with connectivity probability p, i.e., $G_{t-1} \sim G(n,p)$ and $G_t \sim G(n,p)$. For each v, $\widetilde{\Phi}_{t;1,1}(v)$ is defined according to Eq. (2.3.3). Then for any pair of vertices u and v, the correlation between $\widetilde{\Phi}_{t;1,1}(u)$ and $\widetilde{\Phi}_{t;1,1}(v)$ is of order $O(\frac{1}{n})$ for $n \to \infty$.*

Proof. From Eq. (2.3.3) and (2.3.4), for any pair of vertices (u,v),

$$\begin{aligned} & cov(\widetilde{\Phi}_{t;1,1}(u), \widetilde{\Phi}_{t;1,1}(v)) \\ =& cov(\Phi_{t,t;1}(u), \Phi_{t,t;1}(v)) - cov(\Phi_{t,t;1}(u), \Phi_{t,t-1;1}(v)) - \\ & cov(\Phi_{t,t-1;1}(u), \Phi_{t,t;1}(v)) + cov(\Phi_{t,t-1;1}(u), \Phi_{t,t-1;1}(v)) \end{aligned} \tag{2.4.9}$$

We then consider to decompose $\Phi_{t,t;1}(u)$ into two parts representing the cardinalities of two disjoint sets of edges.

$$\Phi_{t,t;1}(u) = X_t(u) + Y_t(u)$$

where the intuitive interpretations behind two terms are listed below:

$$X_t(u) = |\{(u,w) : (u,w) \in E(G_t) \text{ and } w \in N_1(u;G_t) \backslash \{u\}\}|$$

$$Y_t(u) = |\{(w_1,w_2) : (w_1,w_2) \in E(G_t), w_1 < w_2 \text{ and } w_1, w_2 \in N_1(u;G_t) \backslash \{u\}\}|$$

Also, $\Phi_{t,t-1;1}(u)$ is decomposed into two terms as well.

$$\Phi_{t,t-1;1}(u) = X_{t-1}(u) + Y_{t-1}(u)$$

where the intuitive interpretations behind two terms are listed below:

$$X_{t-1}(u) = |\{(u,w) : (u,w) \in E(G_t) \cap E(G_{t-1}) \text{ and } w \in N_1(u;G_t) \backslash \{u\}|$$

$$Y_{t-1}(u) = |\{(w_1,w_2) : (w_1,w_2) \in E(G_{t-1}), w_1 < w_2 \text{ and } w_1, w_2 \in N_1(u;G_t) \backslash \{u\}\}|$$

Table 2.1: Decomposition of the covariance terms in $cov(\widetilde{\Phi}_{t;1,1}(u), \widetilde{\Phi}_{t;1,1}(v))$

$cov(\cdot,\cdot)$	$X_t(v)$	$X_{t-1}(v)$	$Y_t(v)$	$Y_{t-1}(v)$
$X_t(u)$	$+$	$-$	$+$	$-$
$X_{t-1}(u)$	$-$	$+$	$-$	$+$
$Y_t(u)$	$+^{\dagger}$	$-^{\ddagger}$	$+^{\text{II}}$	$-$
$Y_{t-1}(u)$	$-^{\dagger}$	$+^{\ddagger}$	$-^{\text{II}}$	$+$

Similarly, $\Phi_{t,t;1}(v)$ and $\Phi_{t,t-1;1}(v)$ are decomposed with the same structure. By expanding above decompositions into Eq. (2.4.9), we have the following Table 2.1 recording 16 terms and their signs in (2.4.9).

Given Table 2.1, we have that all off-diagonal terms earning the same color (blue, green or magenta) and same positive/negative sign are symmetric. Additionally, the terms having the same mark ($\dagger, \ddagger$ or II) are canceled out due to the fact $Y_t(\cdot)|X_t(\cdot) \overset{iid}{\sim} Y_{t-1}(\cdot)|X_t(\cdot)$. More concretely, for example, for four terms marked by blue, we have $cov(X_t(u), Y_t(v)) = cov(Y_t(u), X_t(v)) = cov(Y_{t-1}(u), X_t(v)) = cov(X_t(u), Y_{t-1}(v))$. The first and third equality are guaranteed by symmetry property. The second equality holds because $Y_t(u)$ and $Y_{t-1}(u)$ share the same conditional distribution, $Bin(\binom{X_t(u)}{2}, p)$, given $X_t(u)$. That is, $cov(Y_t(u), X_t(v)|X_t(u)) = cov(Y_{t-1}(u), X_t(v)|X_t(u))$ and hence $cov(Y_t(u), X_t(v)) = cov(Y_{t-1}(u), X_t(v))$ with application of law of total covariance.

We now return to Eq. (2.4.9). The above reasoning gives

$$
\begin{aligned}
&cov(\widetilde{\Phi}_{t;1,1}(u), \widetilde{\Phi}_{t;1,1}(v)) \\
=&cov(X_t(u), X_t(v)) - cov(X_t(u), X_{t-1}(v)) - cov(X_{t-1}(u), X_t(v)) + cov(X_{t-1}(u), X_{t-1}(v)) \\
=&O(n).
\end{aligned}
$$

The last equality holds because the Cauchy-Schwarz inequality guarantees each of fours term are $O(n)$ where $X_t(\cdot) \sim Bin(n-1,p)$ and $X_{t-1}(\cdot) \sim Bin(n-1,p^2)$.

In the following, to compute $var(\widetilde{\Phi}_{t;1,1}(u))$, $\widetilde{\Phi}_{t;1,1}(u)$ is decomposed as

$$\widetilde{\Phi}_{t;1,1}(u) = X_t + Y_t - X_{t-1} - Y_{t-1}$$

where

$$\begin{aligned} X_t &\sim Bin(n-1,p), \\ Y_t|X_t &\sim Bin(\binom{X_t}{2},p), \\ X_{t-1}|X_t &\sim Bin(X_t,p), \\ Y_{t-1}|X_t &\sim Bin(\binom{X_t}{2},p), \\ Y_t|X_t &\perp Y_{t-1}|X_t. \end{aligned}$$

By applying law of total variance, we reach the following variance order estimation

$$\begin{aligned} & var(\widetilde{\Phi}_{t;1,1}(u)) \\ =&\Theta(var(Y_t - Y_{t-1})) \\ =&\Theta(E[var(Y_t - Y_{t-1}|X_t)] + var[E(Y_t - Y_{t-1}|X_t)]) \\ =&\Theta(E[2\binom{X_t}{2}p(1-p)] + var[0]) \\ =&\Theta(n^2p^3(1-p)) \end{aligned}$$

Therefore, it follows that

$$corr(\widetilde{\Phi}_{t;1,1}(u), \widetilde{\Phi}_{t;1,1}(v)) = O(\frac{1}{n})$$

as desired. □

Now we can prove Theorem 5 with aid of the above lemma.

Proof(Theorem5). Again, to avoid redundant arguments, we only provide derivations of limiting distribution of $\widetilde{\Phi}_{t^*;1,1}(v)$ and the case $t < t^*$ can be achieved in the same approach. Let $v \in [n_i]$, locality statistics $\Phi_{t^*,t^*;1}(v)$ and $\Phi_{t^*,t^*-1;1}(v)$ are respectively decomposed as follows:

$$\Phi_{t^*,t^*;1}(v) = X_i + \sum_{j \neq i} X_j + \sum_{j=1}^{B} Y_j + \sum_{1 \leq j < k \leq B} Z_{jk}, \ v \in [n_i] \tag{2.4.10}$$

where

$$\begin{aligned} X_i &\sim Bin(n_i - 1, \mathbf{P}^A_{i,i}), \\ X_j &\sim Bin(n_j, \mathbf{P}^A_{i,j}), \\ Y_j | X_j &\sim Bin(\binom{X_j}{2}, \mathbf{P}^A_{j,j}), \\ Z_{jk} | X_j, X_k &\sim Bin(X_j X_k, \mathbf{P}^A_{j,k}). \end{aligned}$$

and

$$\Phi_{t^*,t^*-1;1}(v) = X'_i + \sum_{j \neq i} X'_j + \sum_{j=1}^{B} Y'_j + \sum_{1 \leq j < k \leq B} Z'_{jk}, \ v \in [n_i] \tag{2.4.11}$$

where

$$\begin{aligned} X'_i | X_i &\sim Bin(X_i, \mathbf{P}^0_{i,i}), \\ X'_j | X_j &\sim Bin(X_j, \mathbf{P}^0_{i,j}), \\ Y'_j | X_j &\sim Bin(\binom{X_j}{2}, \mathbf{P}^0_{j,j}), \\ Z_{jk} | X_j, X_k &\sim Bin(X_j X_k, \mathbf{P}^0_{j,k}). \end{aligned}$$

Accordingly, the mean of $\widetilde{\Phi}_{t^*;1,1}(v)$ is estimated as follows

$$
\begin{aligned}
&E(\widetilde{\Phi}_{t^*;1,1}(v)) \\
=&E(\Phi_{t^*,t^*;1}(v) - \Phi_{t^*,t^*-1;1}(v)) \\
=&E(X_i + \sum_{j\neq i} X_j - X_i' - \sum_{j\neq i} X_j') + E(\sum_{j=1}^{B} Y_j + \sum_{1\leq j<k\leq B} Z_{jk} - \sum_{j=1}^{B} Y_j' - \sum_{1\leq j<k\leq B} Z_{jk}') \\
=&E(\widetilde{\Phi}_{t^*;1,0}(v)) + E(\sum_{j=1}^{B} Y_j + \sum_{1\leq j<k\leq B} Z_{jk} - \sum_{j=1}^{B} Y_j' - \sum_{1\leq j<k\leq B} Z_{jk}') \\
=&E(\widetilde{\Phi}_{t^*;1,0}(v)) + E(Y_B - Y_B') + o(n) \\
=&np(1-p) + \xi_0(n_i;\Phi) + \mathbf{1}_{\{i=B\}} n_B \delta(1-p) + \zeta(n_i,p,\delta,i) + o(n).
\end{aligned}
$$

Under our setting of $\mathbf{P}^0$ and $\mathbf{P}^A$, the penultimate equality is obtained easily because Z_{jk} and Z_{jk}' share the same distribution and Y_j share the same distribution with Y_j' except $j = B$.

Now let's consider the estimation of $var(\widetilde{\Phi}_{t^*;1,1}(v))$ since the exact derivation of $var(\widetilde{\Phi}_{t^*;1,1}(v))$, through the use of law of total variance, is tedious. Due to the assumption $[n_1, n_2, \ldots, n_B] = [\Theta(n), o(n), \ldots, o(n)]$ and decompositions in Eq.(2.4.10) and Eq.(2.4.11), instead we express variance of $\widetilde{\Phi}_{t^*;1,1}(v)$ as

$$
\begin{aligned}
&var(\widetilde{\Phi}_{t^*;1,1}(v)) = var(\Phi_{t^*,t^*;1}(v) - \Phi_{t^*,t^*-1;1}(v)) \\
=&var(Y_1 - Y_1') + O(n^{2-\epsilon}) = Cn^2p^3(1-p) + O(n^{2-\epsilon})
\end{aligned}
$$

Thus, the central limit theorem leads to

$$
\frac{\widetilde{\Phi}_{t^*;1,1}(v) - E(\widetilde{\Phi}_{t^*;1,1}(v))}{\sqrt{Cn^2p^3(1-p)}} \xrightarrow{d} \mathcal{N}(0,1)
$$

According to Lemma 6, dependencies among $\{\widetilde{\Phi}_{t^*;1,1}(v)\}_{v\in[n_i]}$ are negligible and thus

$$
\frac{max_{v\in[n_i]}\widetilde{\Phi}_{t^*;1,1}(v) - \mu_A'(n_i;\Phi)}{\gamma_A'(n_i;\Phi)} = \frac{W_A'(n_i;\Phi) - \mu_A'(n_i;\Phi)}{\gamma_A'(n_i;\Phi)} \xrightarrow{\mathrm{d}} \mathcal{G}(0,1).
$$

Through similar arguments as in Theorem 1, we can show that $S_{1,0,1}(t^*;\Phi) = \max_{1\leq i\leq B} W_A'(n_i;\Phi)$.

□

Corollary 7. *Assume the setting in Theorem 5. Let β'_Φ be the power of the test statistic $S_{1,0,1}(t;\Phi)$ for $t = t^*$ and β'_Ψ be the power of the test statistic $S_{1,0,1}(t;\Psi)$ for $t = t^*$. Then, as $(n_1, n_2, \ldots, n_B) = (\Theta(n), o(n), \ldots, o(n))$ and $n \to \infty$, $\beta'_\Phi \geq \beta'_\Psi$ and thus $S_{1,0,1}(t;\Psi)$ is inadmissible.*

Proof. This corollary is a generalization of Proposition 3 and Proposition 4. The underlying idea is as follows. In the model presented at the beginnning of § 2.4.1, the variation of number of chatty blocks before $t^* - 1$ makes no difference on the sensitivity of statistics $S_{1,0,1}(t;\Psi)$ and $S_{1,0,1}(t;\Phi)$ as long as the orders of chatty blocks are $o(n)$. Namely, in the limiting case β'_Φ and β'_Ψ are functions of n_B and independent of $\{n_2, n_3, \ldots n_{B-1}\}$. We can then extend the power comparison conclusion from Proposition 4 for $B = 3$ to the general case. The details are somewhat tedious and are omitted. □

Chapter 3

Applications

In this section, for interested practitioners and engineers, § 3.1 provides implementations of two scan statistics in language R and a hands-on example for illustration. Next, § 3.2 presents applications of two locality-based scan statistics on Enron email corpus dataset and Zebrafish neuronal activities dataset [36].

3.1 Code Implementation

In this section, we present implementations of above detection procedures using R and its `igraph` package. Firstly, we generate a time series of graphs with a change-point as a running example, generated from Stochastic Block Model introduced in § 2.2. Secondly, we introduce the implementation of two locality statistics, i.e., `local.scan`, in `igraph` package and implementations of temporal normalization steps. Lastly, we demonstrate the potency of our detection methodology introduced in § 2.3 by showing that the change-point reported by proposed procedures matches well with the underlying change-point of synthetic data.

3.1.1 Generating Time Series of Graphs with Change-point under SBM

To provide a running example, we first create an artificial time series of graphs $\{G_t\}_{t=1}^{20}$, each sampled from a stochastic block model with $|V| = 20, B = 3$. The block membership of the vertices is fixed over time while the connectivity probabilities matrix $\mathbf{P} = \mathbf{P}^t$ changes in the last time step, i.e., $\mathbf{P}^t = \mathbf{P}^0$ for $t = 1, \ldots, 19$,and $\mathbf{P}^{20} = \mathbf{P}^A$, where $\mathbf{P}^0 \neq \mathbf{P}^A$

$$\mathbf{P}^0 = \begin{pmatrix} p & p & p \\ p & h & p \\ p & p & p \end{pmatrix}, \qquad \mathbf{P}^A = \begin{pmatrix} p & p & p \\ p & h & p \\ p & p & q \end{pmatrix} \tag{3.1.1}$$

The blocks contain $[n_1, n_2, n_3] = [10, 5, 5]$ vertices and $[p, h, q] = [0.2, 0.5, 0.8]$.

```
# Generate a time series of graphs under SBM
# Input parameters:maxTime=20, [n1,n2,n3]=[10,5,5],
# change-point-time=20, [p,h,q]=[0.2,0.5,0.8]
library(igraph)
library(Matrix)
set.seed(123456)              # set seed to make tsg reproducible
maxTime <- 20                 # number of total time steps
n1 <- 10; n2 <- 5; n3 <- 5    # number of vertices in each block
changingTime <- 20            # change point time stamp
p <- 0.2; h <- 0.5; q <- 0.8 # distinct elements in block matrices

# construct P0 and PA
P0 <- matrix(p,3,3)
P0[2,2] <- h
PA <- P0
PA[3,3] <- q
nVertex <- n1+n2+n3

# create graph time series before changeTime with P0
tsg_normal <- lapply(1:(changingTime-1), function(x) {
                  g <- sbm.game(nVertex,P0,c(n1,n2,n3),
                                directed=T);
                  V(g)$name <- LETTERS[1:20];
                  V(g)$label <- V(g)$name; return(g);
              })

# create graph time series at changeTime with PA
tsg_anomaly <- lapply(changingTime:maxTime,  function(x) {
                   g <- sbm.game(nVertex,PA,c(n1,n2,n3),
                                 directed=T);
                   V(g)$name <- LETTERS[1:20];
                   V(g)$label <- V(g)$name; return(g);
                })

igraph_tsg <- c(tsg_normal, tsg_anomaly)
```

Let's plot in Figure 3.1 the graphs at most recent four timestamps of `igraph_tsg` and peek the existence of anomolous (red) community at change-point, that is, $t = 20$ whose excessive communications is new at the change-point. This authenticates that the time series of graphs `igraph_tsg` is an well-generated example to motivate our proposed graph invariants in § 2.3, i.e., scan statistics, though the identities of anomalous community members are unknown to observers.

```
set.seed(123456) # set seed to make plot layout reproducible
layout1 <- layout.fruchterman.reingold(igraph_tsg[[20]])
par(mfrow=c(2,2))
for(i in (maxTime-3):maxTime)
{
    g <- igraph_tsg[[i]]
    # set distinct vertex colors to diffrent blocks
    V(g)$label.color <- c(rep(rgb(0,0,0,1),n1),
                          rep(rgb(1,1,0,1),n2),
                          rep(rgb(1,0,0,1),n3))
    plot(g,layout=layout1,main=paste0('t=',i))
}
```

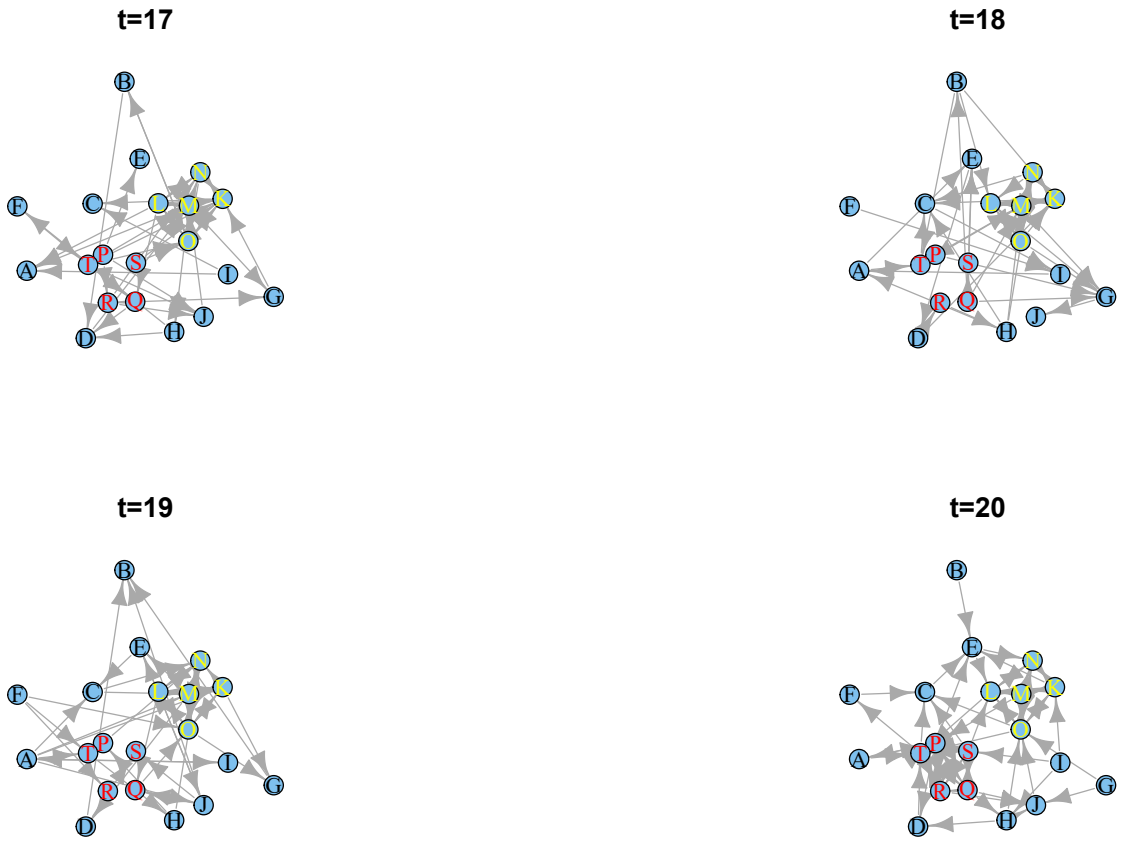


Figure 3.1: A generated time series of graphs under SBM with change-point at $t = 20$. V is partitioned into three blocks where black vertices are from $[n_1]$, yellow from $[n_2]$, and red from $[n_3]$. The subgroup $[n_3]$ exhibits the change of community frequency at the pre-determined change-point and hence becomes the target community for detection.

3.1.2 Two Locality Statistics and `local.scan` in `igraph`

In this section, we look into two proposed locality statistics using the running synthetic example `igraph_tsg` and introduce their implementation `local.scan` function in `igraph` package.

Given a generated time series of directed graphs `igraph_tsg`, firstly, we make an another example to illustrate the concept of locality statistic $\Psi_{t;k}(v)$, e.g., Eq, (2.3.1), using `igraph`. For instance, we let $t = 20, k = 1, v = O$ and trace k-order neighborhoods through both directions of edges (argument `mode='all'`). As a side note, in a directed graph, `mode='in'` and `mode='out'` correspond to tracing neighborhoods through "in-edge" and "out-edge" respectively. We can plot $\Omega(N_{k=1}(v = O; G_{t=20}); G_{t=20})$ in Figure 3.2 by

```
par(mfrow=c(1,1))
t <- 20; k <- 1; vertex_name <- 'O'
g <- igraph_tsg[[t]]

# find the subgraph induced by k-hop distance neighborhoods
# from node 'O'
sub_g <- graph.neighborhood(graph=g, order=k,
                            nodes=vertex_name, mode='all')
plot(sub_g[[1]]);
```

and count the number of edges in the subgraph shown in Figure 3.2

$$\Psi_{t=20;k=1}(v = O) = |E(\Omega(N_{k=1}(v = O; G_{t=20}); G_{t=20}))| = 27$$

by

```
length(E(sub_g[[1]])) # same as ecount(sub_g[[1]])
[1] 27
```

Moreover, if we want to calculate $\{\Psi_{t;k}(v)\}_{v \in V}$ over all vertices in a graph, which is a nec-

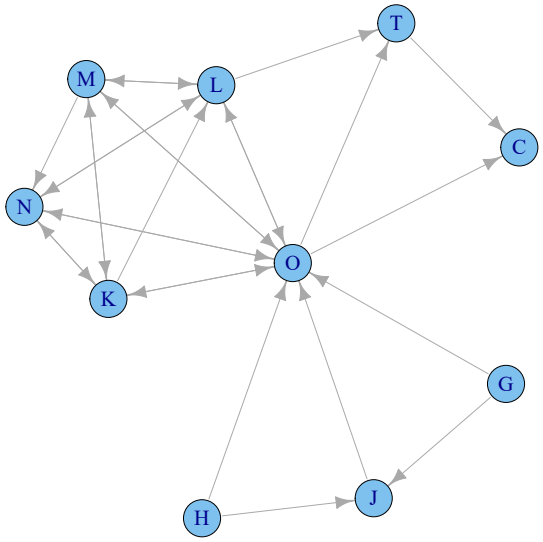


Figure 3.2: Induced subgraph $\Omega(N_{k=1}(v = O; G_{t=20}); G_{t=20})$ of the last graph G_{20} from $v = O$ with $k = 1$ order neighborhood.

essary intermediate step of computing scan statistics introduced in § 2.3, `local.scan` function in `igraph` implements these calculations and outputs an $|V|$-dimensional vector representing $\Psi_{t;k}(v)$ on each vertex. In our previous setting $t = 20, k = 1$, the whole 20 locality statistics $\{\Psi_{t;k}(v)\}_{v='A'}^{'T'}$ are

```
local.scan(graph.us = igraph_tsg[[t]], graph.them = NULL,
           k = 1, weighted = F, mode = 'all')
[1]  5  1  9  5 11  3  3  7  3  9 22 26 19 21 27 26 23 21 24 34
```

Secondly, we take a look at the other locality statistic $\Phi_{t,t';k}(v) = |E(\Omega(N_k(v; G_t); G_{t'}))|$ in Eq. (2.3.2) . Let us get back to our `igraph` example – the time series of graphs `igraph_tsg`. Assume $t = 20$, $t' = 19$, $k = 1$ and $v = O$, the single locality statistic $\Phi_{t,t';k}(v)$ is 25 where Figure 3.3 shows $G_{t'=19}$ and $\Omega(N_{k=1}[v = O; G_{t=20}], G_{t'=19})$. Note that $N_{k=1}[v = O; G_{t=20}]$ could be found in Figure 3.2.

```
t <- 20; t_prime <- 19; k <- 1; vertex_name <- 'O'
```

```
# find the k-hop distance neighborhoods from node 'O' in graph G_t
nbrs <- unlist(neighborhood(graph=igraph_tsg[[t]], order=k,
                nodes=vertex_name,mode='all'))
# find the induced subgraph with above nbrs in graph G_t'
g <- induced.subgraph(graph = igraph_tsg[[t_prime]], vids = nbrs)
length(E(g))
[1] 25
```

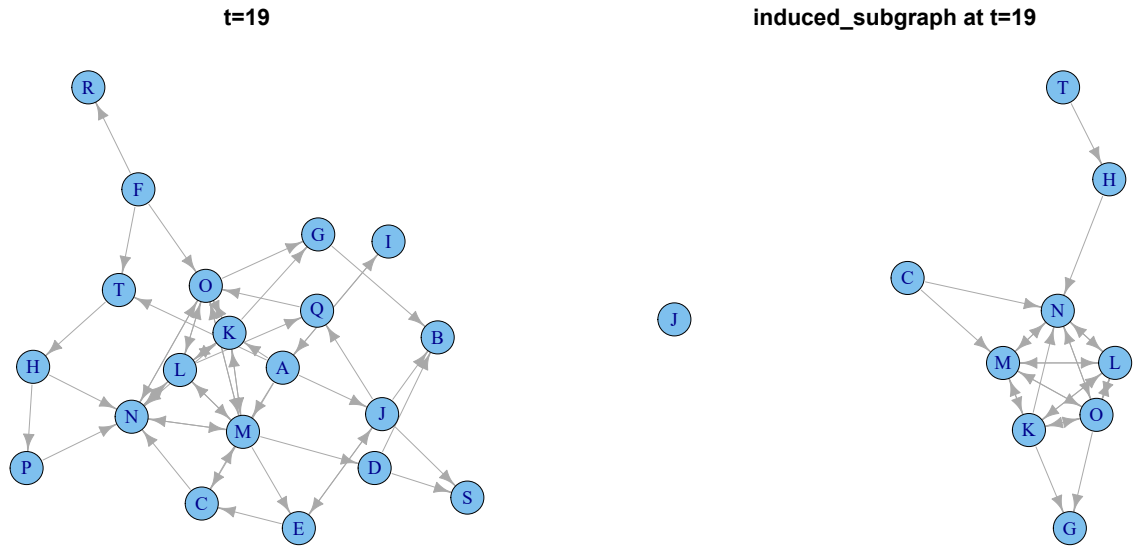


Figure 3.3: The left figure is the graph $G_{t'=19}$, i.e., the graph at time stamp 19 in the generated time series of graphs. The right figure is $\Omega(N_{k=1}[v = O; G_{t=20}], G_{t'=19})$, i.e., induced subgraph in $G_{t'=19}$ by vertex set $N_{k=1}[v = O; G_{t=20}]$ where $N_{k=1}[v = O; G_{t=20}]$ is shown in Figure 3.2.

Likewise, given t, t', k, if we want to calculate $\{\Phi_{t,t';k}(v)\}_{v \in V}$ over all vertices in a graph, which is a necessary intermediate step of computing scan statistics introduced in § 2.3, `local.scan` function in `igraph` implements these calculations and outputs an $|V|$-dimensional vector representing $\{\Phi_{t,t';k}\}_{v=1}^{|V|}$ on each vertex. In our previous setting $t = 20$, $t' = 19$, $k = 1$, the whole 20 locality statistics $\{\Phi_{t,t';k}(v)\}_{v='A'}^{'T'}$ are

```
local.scan(graph.us = igraph_tsg[[t]],
           graph.them = igraph_tsg[[t_prime]],
           k = 1, weighted = F, mode = 'all')
[1]  1  0  3  2  8  1  1  2  0  3 20 20 20 20 25  5  4  2  1 10
```

3.1.3 Temporally-normalization Implementation

Using `igraph_tsg` as a test example , the eventual goal is to calculate $S_{\tau,\ell,k}(t;\Psi)$ (or $S_{\tau,\ell,k}(t;\Phi)$) on `igraph_tsg` and uncover whether a significant increment arises at change-point $t = 20$. Having the time series `tsg,k,tau,ell` and the selection of underlying locality statistic (Ψ or Φ) as input arguments, the main function `scanstat` below enables to calculate the $S_{\tau,\ell,k}(t;\cdot)$ at all time steps. The function `local.scan` introduced in § 3.1.2 is embedded in `scanstat`. It is worthwhile to note that $S_{\tau,\ell,k}(t;\cdot)$ is only well-defined when $t > \tau + \ell$. Hence, we leave $S_{\tau,\ell,k}(t;\cdot) = 0$ when $1 \leq t \leq \tau + \ell$ such that the ouput of function `scanstat` is a complete $1 \times$ `maxTime` vector starting from time stamp 1 and ending at `maxTime` .

```
scanstat <- function(tsg, k, tau, ell, locality,mode) {
  # determine weighted/unweighed directed/undirected
  isWeighted = is.weighted(tsg[[1]])
  isDirected = is.directed(tsg[[1]])

  # number of time steps and number of vertices
  maxTime = length(tsg)
  nVertex = vcount(tsg[[1]])

  # Underlying locality stat is \Psi
  if (locality == 'Psi') {
    lstatPsi <- matrix(0,nrow=nVertex,ncol=maxTime)
    for (i in 1:maxTime) {
      # graph at time i
      g <- tsg[[i]]
      # locality statistics \Psi over all vertices
      # at t=i in Eq (1)
      lstatPsi[,i] <- local.scan(graph.us=g,graph.them=NULL,
                                 k=k,mode=mode,
                                 weighted=isWeighted)
    }
    lstat <- lstatPsi
  }

  # Underlying locality stat is \Phi
  else if (locality == 'Phi') {
```

```
    lstatPhi <- array(0,dim=c(nVertex,(tau+1),maxTime))
    for (i in 1:maxTime) {
      if (i>tau) {
        # graph to trace k-th order neighborhood
        g <- tsg[[i]]
        for(j in 0:tau) {
          # graph to construct induced subgraph on which
          # counting edges
          g_prime <- tsg[[i-tau+j]]
          # locality statistics \Phi over all vertices
          # with t=i and t'=i-tau+j in Eq (2)
          lstatPhi[,(j+1),i] <- local.scan(graph.us=g,
                                           graph.them=g_prime,
                                           k=k,mode=mode,
                                           weighted=isWeighted)
        }
      }
    }
    lstat <- lstatPhi
  }

  # vertex-dependent normalization of Eq (3)
  nlstat <- vertex.norm(lstat,tau)
  # temporal normalization of Eq (7)
  scanstat <- temp.norm(nlstat,tau,ell)
  (scanstat)
}
```

where subfunction `vertex.norm` comprehensively implements Eq. (2.3.3)-(2.3.5)

```
vertex.norm <-function (input_stat, tau = 1)
{
  if (is.matrix(input_stat)) {
      n <- nrow(input_stat)
      nbins <- ncol(input_stat)
      nstat <- matrix(0, n, nbins)
      for (i in 1:nbins) {
          if (i > tau) {
              if (tau==0)
                  nstat[,i]=input_stat[,i]
              else {
           muv <- apply(as.matrix(input_stat[,(i-tau):(i-1)]),1,mean)
           sdv <- apply(as.matrix(input_stat[,(i-tau):(i-1)]),1,sd)
           sdv[is.na(sdv)] <- 1
           nstat[, i] <- (input_stat[,i] - muv)/pmax(sdv, 1)
              }
          }
```

```
      }
  }
  else {
    dd <- dim(input_stat)
    n <- dd[1]
    nbins <- dd[3]
    nstat <- matrix(0, n, nbins)
    for (i in 1:nbins) {
      if (i > tau) {
        if (tau==0)
          nstat[, i]=input_stat[,(tau+1),i]
        else {
 muv <- apply(as.matrix(input_stat[,(1:tau),i]),1,mean)
 sdv <- apply(as.matrix(input_stat[,(1:tau),i]),1,sd)
          sdv[is.na(sdv)] <- 1
          nstat[, i] <- (input_stat[,(tau+1),i] - muv)/pmax(sdv, 1)
        }
      }
    }
  }
  return(nstat)
}
```

and subfunction `temporal.norm` comprehensively implements Eq. (2.3.6)-(2.3.9)

```
temp.norm<-function (stat, tau = 1, ell = 0)
{
    maxTime <- ncol(stat)
    Mtilde <- apply(stat, 2, max)
    argmaxV <- apply(stat, 2, which.max)
    if (ell == 0) {
        return(list(sstat=Mtilde,argmaxV=argmaxV))
    }
    else if(ell ==1 ) {
        return(list(sstat=Mtilde-c(0,Mtilde[-maxTime]),
                    argmaxV=argmaxV))
    }
    else {
        muMtilde <- rep(0, maxTime)
        sdMtilde <- rep(1, maxTime)
        for (i in (ell + 1):maxTime) {
            muMtilde[i] <- mean(Mtilde[(i - ell):(i - 1)])
            sdMtilde[i] <- sd(Mtilde[(i - ell):(i - 1)])
        }
        sstat <- (Mtilde - muMtilde)/pmax(sdMtilde, 1)
        sstat[1:(tau + ell)] <- 0
        argmaxV[1:(tau + ell)] <- NA
```

```
        return(list(sstat=sstat, argmaxV=argmaxV))
    }
}
```

After sourcing above three functions into R, we can calculate the final scan statistic $S_{\tau,\ell,k}(t;\Psi)$ (or $S_{\tau,\ell,k}(t;\Phi)$) by calling `scanstat(tsg, k, tau, ell, locality,mode)`. For instance, if Ψ is selected as the underlying locality statistic, $\{S_{\tau,\ell,k}(t;\Psi)\}_{t=1}^{maxTime}$ of `igraph_tsg` is obtained as below.

```
# set tau, ell, k
tau <- 4; ell <- 3; k <-1;
# calculate scan statistics on igraph_tsg with locality 'Psi'
scanstat(tsg=igraph_tsg, k=k, tau=tau, ell=ell,
         locality = 'Psi', mode= 'all')

$sstat
 [1]  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
 [6]  0.0000000  0.0000000  -0.4463684 1.7714957 -0.8157701
 [11]-0.5724556 -0.2357372  2.5132600 -0.2241908  3.0586020
 [16]-1.7545396 -0.5428413 -0.8836799  0.3208085  6.4385630

$argmaxV
 [1] NA NA NA NA NA NA NA  7  2  4 19  1  5  7  6  8 20  3  1 16
```

For `igraph_tsg` example, we can see that `sstat,i.e.,`$\{S_{\tau,\ell,k}(t;\Psi)\}_{t=1}^{maxTime}$, is a vector of length 20 and first 7 values are 0 due to $\{S_{\tau,\ell,k}(t;\Psi)=0\}_{t=1}^{\tau+\ell}$ by our convention. Furthermore, at the true-but-unknown change-point, `changingTime=20`, $S_{\tau,\ell,k}(t=20;\Psi)$ is significantly larger than $\{S_{\tau,\ell,k}(t;\Psi)\}_{t=1}^{19}$ of previous time stamps. Thus, the inconsistency, particularly surprising increment of $S_{\tau,\ell,k}(t=20;\Psi)$, implies an emergence of an anomalous community at time 20. This conjecture matches our expectation well, foreshadowing correctness and practicality of $S_{\tau,\ell,k}(t;\Psi)$.

Sometimes observers are also interested in unearthing the center actor of the arising anomalous community, i.e., v that achieves the maximum in Eq (2.3.6). They can resort

to the argmaxV whose elements represent $\arg\max_v \widetilde{J}_{t,\tau;k}(v)$ at each time point. For example, in `igraph_tsg` at $t = 20$, the center of anomalous community is

```
V(igraph_tsg[[20]])[16]
Vertex sequence:
[1] "P"
```

In other words, the vertex P is an individual whose behavior deserves further digging in the network. Through tracing the k-th order neighborhood of P in $G_{t=20}$, we locate individuals in the community $N_k(v = P; G_{t=20})$ as anomalous group on which more investigation is to conduct afterwards.

```
# find the anomalous community by tracing k-th (k=1 in this example)
# order neighborhoods.
V(igraph_tsg[[20]])[neighborhood(igraph_tsg[[20]],1,"P")[[1]]]
Vertex sequence:
[1] "P" "A" "D" "E" "M" "Q" "R" "S" "T"
```

Similarly, above procedures can be replicated if the underlying locality statistic is Φ and we obtain

```
tau <- 4; ell <- 3; k <-1;
# calculate scan statistics on igraph_tsg with locality 'Phi'
scanstat(tsg=igraph_tsg, k=k, tau=tau, ell=ell, locality = 'Phi',
         mode= 'all')
$sstat
[1] 0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
[6] 0.000000000  0.000000000 -0.455884852 -0.455884852  0.786908912
[11]0.399728001 -0.005263018 -0.648717474  0.351282526  3.583333333
[16]-0.039193090-0.783523371 -1.22425526   0.000000000  6.387437324

$argmaxV
 [1] NA NA NA NA NA NA NA  8  6 19  2  1  3  2 19  8 10  8  3 20

V(igraph_tsg[[20]])[20]
Vertex sequence:
[1] "T"
# find the community by tracing k-th (k=1 in this example)
# order neighborhoods.
V(igraph_tsg[[20]])[neighborhood(igraph_tsg[[20]],1,"T")[[1]]]
Vertex sequence:
```

```
 [1] "T" "A" "C" "D" "F" "L" "O" "P" "Q" "R" "S"
```

It's trivial to see that both $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ are capable of detecting the latent change-point for this synthetic example as there exist large deviations from the normal pattern in time series of scan statistics. Also, both methods report vertices P,Q,R,S,T to be members of anomalous community. This is consistent with their probabilistic characterizations in the generating process. In summary, $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ are connectivity-based graph invariants to track the upsurge of anomalous community and simultaneously unbury the center of dense community.

To achieve a better visualization of $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ of `igraph_tsg` with varying $k = 0, 1, 2$, we plot the Figure 3.4 using the following R snippet

```
tau <- 4; ell <- 3; tmax <- length(igraph_tsg)
phi0 <- scanstat(igraph_tsg, k=0, tau=tau, ell=ell, 'Phi', 'all')
psi0 <- scanstat(igraph_tsg, k=0, tau=tau, ell=ell, 'Psi', 'all')
phi1 <- scanstat(igraph_tsg, k=1, tau=tau, ell=ell, 'Phi', 'all')
psi1 <- scanstat(igraph_tsg, k=1, tau=tau, ell=ell, 'Psi', 'all')
phi2 <- scanstat(igraph_tsg, k=2, tau=tau, ell=ell, 'Phi', 'all')
psi2 <- scanstat(igraph_tsg, k=2, tau=tau, ell=ell, 'Psi', 'all')
dat2 <- rbind(phi0$sstat,phi1$sstat,phi2$sstat)
dat1 <- rbind(psi0$sstat,psi1$sstat,psi2$sstat)

psd <- 5
pos <- seq((tau+ell+1),tmax,4)
xlabs <- 1:tmax
rownames(dat1) <- c("k=0","k=1","k=2")
rownames(dat2) <- c("k=0","k=1","k=2")

require(ggplot2)
require(reshape2)
require(plyr)
dat3 <- melt(t(dat1))
dat4 <- melt(t(dat2))
dat <- rbind(dat3,dat4)
dat <- cbind(dat,c(rep("Psi",nrow(dat3)),rep("Phi",nrow(dat4))))
colnames(dat) <- c("time","group","stat","method")
dat$sd <- rep(psd,nrow(dat))
```

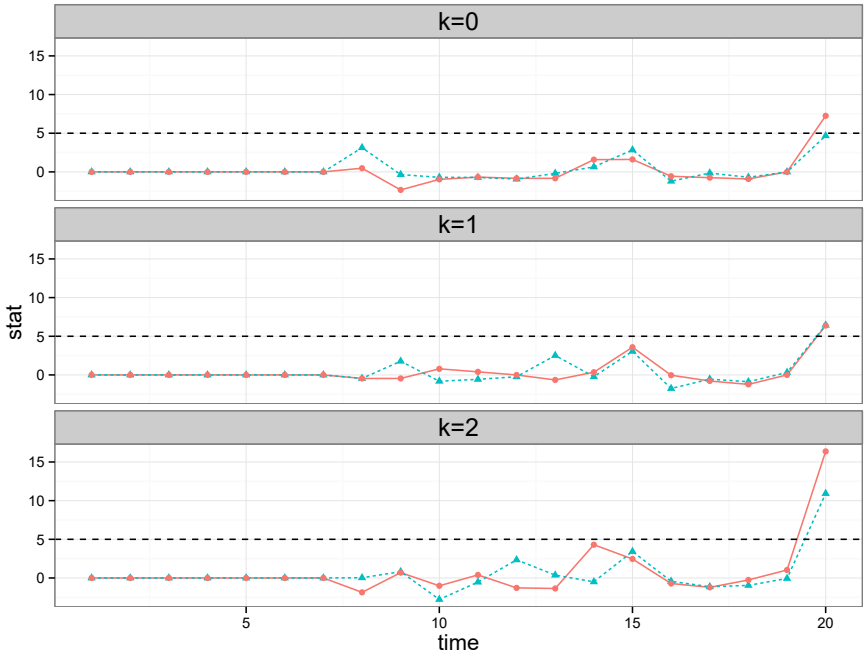


Figure 3.4: $S_{\tau,\ell,k}(t;\Psi)$(sea green) and $S_{\tau,\ell,k}(t;\Phi)$(orange), the temporally-normalized standardized scan statistics using $\tau = 4, \ell = 3$ in time series of graphs. Top: $k = 0$; Middle: $k = 1$; Bottom: $k = 2$.

```
fsize <- 15
lsize <- 1.4
csize <- 1.7
p <- ggplot(data=dat, aes(x=time,y=stat))
 + facet_wrap(~group, nrow=3)
 + geom_line(aes(color=method, linetype=method))
 + geom_point(aes(color=method,shape=method))
 + scale_colour_hue(guide="none")
 + scale_fill_hue(guide="none")
 + geom_hline(aes(yintercept=sd),linetype="dashed")
 + theme_bw()
 + theme(axis.title.x=element_text(size=fsize))
 + theme(axis.title.y=element_text(size=fsize))
 + theme(strip.text=element_text(size=rel(lsize)))
 + ylim(range(dat$stat))
 + theme(legend.position="none")
p
```

Figure 3.4 depicts $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$, the temporally-normalized standardized scan statistics for various $k = \{0, 1, 2\}$ using $\tau = 4, \ell = 3$. The horizontal dashed line indicated 5 standard deviation, so any scan statistic exceeding that threshold can be

considered as an anomaly. According to different application domains, the criterion of anomaly alert is determined by the observer. The higher deviation in terms of number of standard deviations is selected, the lower false positive rate can be achieved; the lower deviation is selected, the higher true positive rate can be achieved. In this example, both $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ detect anomaly ($t = 20$) with $k = 1, 2$, only $S_{\tau,\ell,k}(t;\Phi)$ detects anomaly with $k = 0$.

3.2 Enron Emails dataset

In this section, we apply previously proposed anomalous community detection technique, for analyzing two real datasets: Enron email dataset and Zebrafish dataset. They are briefly introduced in § 1.3. For each dataset, we identify change-points and further investigate anomalous communities by referring real event information associated with the time-line. Both experiments demonstrate the efficacy of locality-based scan statistics in practice. We use the Enron email data used in [39] in this experiment. It consists of time series of graphs $\{G_t\}$ with $|V| = 184$ vertices for each week $t = 1, \ldots, 189$, where we draw an unweighted edge when vertex v sends at least one email to vertex w during a one week period.

After truncating first 40 weeks for vertex-standardized and temporal normalizations, Figure 3.5 depicts $S_{\tau,\ell,k}(t;\Psi)$ (sea-green) and $S_{\tau,\ell,k}(t;\Phi)$ (orange) in the remaining 149 weeks from August 1999 to June 2002. In this experiment, we choose both $\tau = \ell = 20$, used in [39], to keep the comparisons between the two papers meaningful. As indicated in [39], detections are defined as weeks t such that $S_{\tau,\ell,k} > 5$. Hence, from Figure 3.5 we

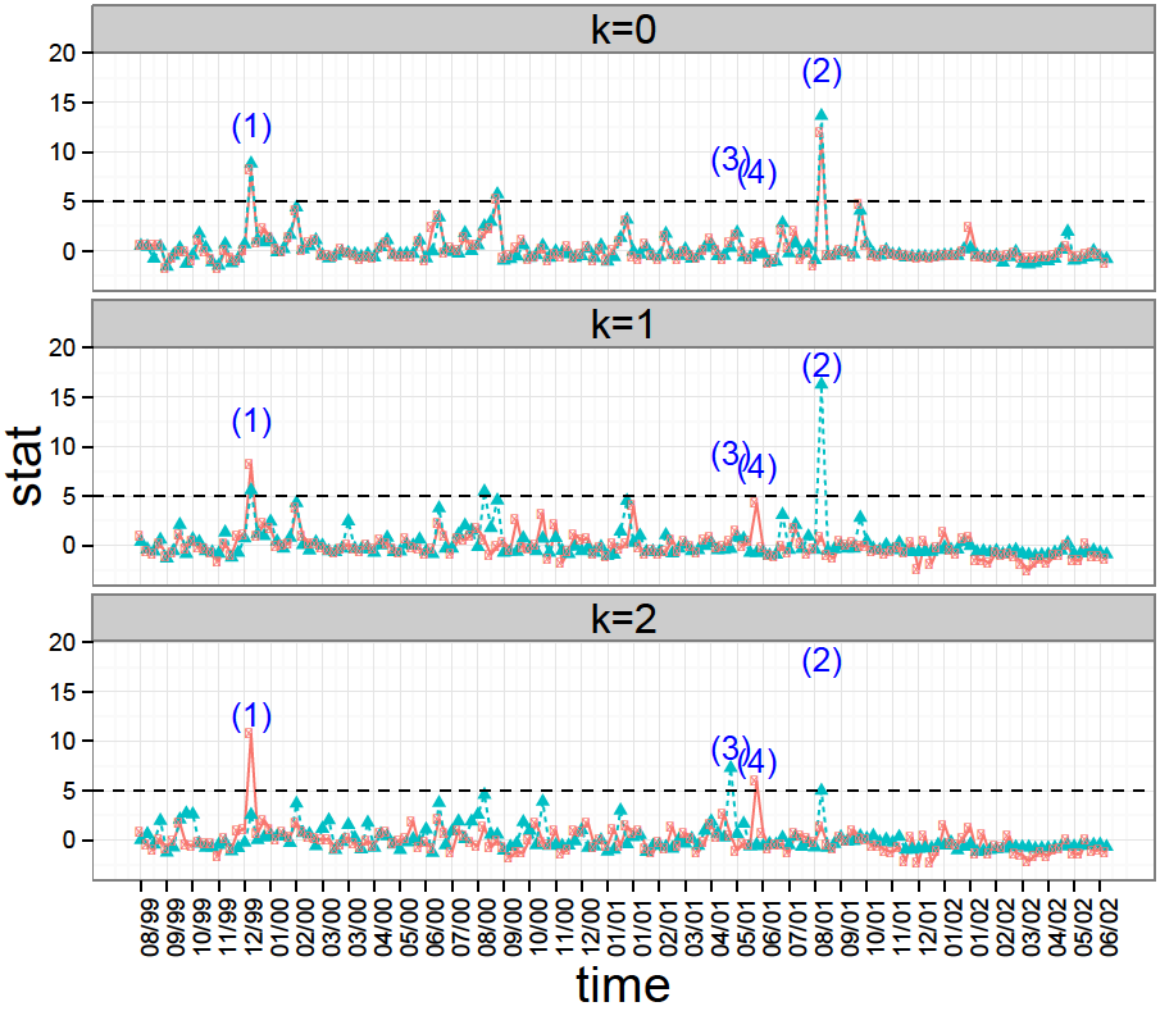


Figure 3.5: $S_{\tau,\ell,k}(t;\Psi)$(sea green) and $S_{\tau,\ell,k}(t;\Phi)$(orange), the temporally-normalized standardized scan statistics using $\tau = \ell = 20$, in time series of Enron email-graphs from August 1999 to June 2002. Top: $k = 0$; Middle: $k = 1$; Bottom: $k = 2$. In the case $k = 0$, both $S_{20,20,0}(t;\Psi)$ and $S_{20,20,0}(t;\Phi)$ show detections ($S_{\tau,\ell,k}(t;\cdot) > 5$) at observation mark (1) and (2); in the case $k = 1$, both $S_{20,20,1}(t\Psi)$ and $S_{20,20,1}(t;\Phi)$ show detections at observation mark (1), $S_{20,20,1}(t;\Psi)$ also indicates an anomaly at observation mark (2); in the case $k = 2$, $S_{20,20,2}(t;\Psi)$ detects anomalies at observation mark (2) and (3), but $S_{20,20,2}(t;\Phi)$ captures anomalies at observation mark (1) and (4). Detailed analyses on each observation [(1) - (4)] are provided in §3.2 respectively.

have following observations and reasonings.

1. $S_{20,20,0}(t;\Psi), S_{20,20,0}(t;\Phi), S_{20,20,1}(t;\Psi), S_{20,20,1}(t;\Phi)$ and $S_{20,20,2}(t;\Phi)$ indicate a clear anomaly at $t^* = 58$ in December 1999. This coincides with the happening of Enron's tentative sham energy deal with Merrill Lynch to meet profit expectations

and boost stock price [13]. The center of suspicious community-employee v_{154} is identified by all five statistics.

2. $S_{20,20,0}(t;\Psi)$, $S_{20,20,0}(t;\Phi)$, $S_{20,20,1}(t;\Psi)$ and $S_{20,20,2}(t;\Psi)$ capture an anomaly at $t^* = 146$ in the mid-August 2001. This is the period that Enron CEO Skilling made a resignation announcement when the company was surrounded by public criticisms shown in [13]. The center of suspicious community-employee v_{95} is identified by these four statistics.

3. $S_{20,20,2}(t;\Psi)$ signifies an anomaly at $t^* = 132$ in late April 2001 where $S_{20,20,k}(t;\Phi)$ fails to alert for any $k \in \{0,1,2\}$. This phenomenon occurs because $S_{20,20,2}(t;\Psi)$ captures the employee v_{90} whose second-order neighborhood $N_2(v_{90};G_{132})$ contains 116 emails at $t^* = 132$ but 0 email in his second-order neighborhoods of previous 20 weeks. That is, the time-dependent second-order neighborhood $N_2(v_{90};G_t)$ had no communication in the period from $t = 112$ to $t = 131$. On the other hand, this behavior cannot be monitored by $S_{20,20,2}(132;\Phi)$ because the change of communication frequency in a fixed second-order neighborhood $N_2(v_{90};G_{132})$, measured by locality statistics Φ, is not so significant. More concretely, the number of emails in the unchanged $N_2(v_{90};G_{132})$ has a mean 45.5 and a standard deviation 14.9 from $t = 112$ to $t = 131$. In [13], this anomaly appears after the Enron Quaterly Conference Call in which a Wall Street analyst Richard Grubman questioned Skilling on the company's refusal of releasing balance sheet but then got insulted by Skilling.

4. $S_{20,20,2}(t;\Phi)$ shows a detection on v_{135} at $t^* = 136$ before June 2001 over $S_{20,20,2}(t;\Psi)$. This comes from the fact that the fixed second-order neighborhood of employee v_{135} at $t^* = 136$, i.e., $N_2(v_{135};G_{136})$, has a small standard deviation 1.08 in previous 20

weeks while the communications in time-dependent neighborhoods $\{N_2(v_{90}; G_t)\}_{t=116}^{135}$ has a large standard deviation 10.04. Practically speaking, in this case, a dramatic increment of email contacts in the certain community $N_2(v_{135}; G_{136})$ could be captured by $S_{20,20,2}(t; \Phi)$ but ignored by $S_{20,20,2}(t; \Psi)$ because unstable communication patterns in $\{N_2(v_{90}; G_t)\}_{t=116}^{135}$ offsets the sensitivity of signal. According to [13], this anomaly corresponds to the formal notice of closure and termination of Enron's single largest foreign investment, the Dabhol Power Company in India.

In summary, observations 1 and 2 demonstrate that in some cases both $S_{\tau,\ell,k}(t; \Psi)$ and $S_{\tau,\ell,k}(t; \Phi)$ are capable of capturing the same community which has a significant increment of connectivity. Besides, in some situations shown in observations 3 and 4, $S_{\tau,\ell,k}(t; \Psi)$ and $S_{\tau,\ell,k}(t; \Phi)$ achieve different detections due to its adaptability.

3.3 Zebrafish dataset

3.3.1 Data Description

The original zebrafish dataset is a simultaneous whole-brain neuronal activity data at near single cell resolution obtained using Light-Field Deconvolution Microscopy (LFDM) combined with GCaMP as a calcium reporter [37]. The data consist of periods of spontaneous neuronal activity and sequences of different types of olfactory stimulations. More concretely, the raw data set consists of a multivariate time series D of dimension of $n \times m$ where $n(= 5,379)$ is the number of zebrafish neurons and $m(= 5,000)$ is the number of total time frames across 250 seconds. Each single time frame is approximately $\frac{1}{20}$ second. $D_{i,j}$ records a measure of activity level of the neuron i at time step j by computing fluorescence traces of spatial filters divided by its mean.

During the data collection process over time, a lab scientist creates an underlying change-point occuring at the 16th second, by giving an olfactoric stimulus to the zebrafish. This stimulus lasts about 2 seconds. Additionally, the lab scientist observed that there is an eye movement of the zebrafish happening at the 59th second, and tentative tail movements of the zebrafish happening at the 78th and 218th seconds. These four time stamps are the ground truths of anomalies in this dataset of which we are presently aware. Additionally, during data preprocessing, some spurious edge neurons were removed. The cleaned data used in the following experiments is a multivariate time series D where $n = 5,105$ and $m = 5,000$.

3.3.2 Construction of Time Series of Association Graphs

Using the multivariate time series D and two parameters, window size Δ and edge threshold θ, we construct a time series of unweighted graphs $\{G_t\}_{t=1}^{T}$ with a coarser resolution by following three steps below:

1. Firstly, we split the data across time into chunks and let each chunk contain data across Δ time steps out of $m(= 5,000)$ steps. Note that we also let adjacent chunks share overlapping $\frac{\Delta}{2}$ steps so that contiguous chunks are dependent. Hence, there are $t = 1, \ldots, \frac{2m}{\Delta} - 1$ chunks in total, and each chunk represents $\frac{\Delta}{20}$ seconds in real-time duration (i.e., Δ time steps in the resolution of D). For example, if $\Delta = 50$, the first chunk is $[1, 50]$, the second chunk is $[26, 75]$, the third chunk is $[51, 100]$, and so on. The interval of the t-th time chunk can be formulated as $[\frac{(t-1)\Delta}{2} + 1, \frac{(t-1)\Delta}{2} + \Delta]$ in original m time steps.

2. For each time stamp t, each neuron then has Δ data samples of measurements of activity levels between $[\frac{(t-1)\Delta}{2} + 1, \frac{(t-1)\Delta}{2} + \Delta]$ in original D. Based on Δ samples in $[\frac{(t-1)\Delta}{2} + 1, \frac{(t-1)\Delta}{2} + \Delta]$, we construct an association matrix $M^{(t)}$ for time stamp t where $M_{i,j}^{(t)}$ denotes the absolute value of the sample's Pearson correlation coefficient between neuron i and neuron j.

$$M_{i,j}^{(t)} = \frac{|\sum_{k=(t-1)\Delta/2+1}^{(t-1)\Delta/2+\Delta}(D_{i,k} - \bar{D}_i)(D_{j,k} - \bar{D}_j)|}{\sqrt{\sum_{k=(t-1)\Delta/2+1}^{(t-1)\Delta/2+\Delta}(D_{i,k} - \bar{D}_i)^2}\sqrt{\sum_{k=(t-1)\Delta/2+1}^{(t-1)\Delta/2+\Delta}(D_{j,k} - \bar{D}_j)^2}}$$

   where $\bar{D}_i = \frac{1}{100}\sum_{k=(t-1)\Delta/2+1}^{(t-1)\Delta/2+\Delta} D_{i,k}$.

3. So far, $\{M^{(t)}\}_{t=1}^{T}$ can be seen as a time series of weighted adjacency matrices where $T = \frac{2m}{\Delta} - 1$. However, we consider only unweighted graphs in this section and

thus set a threshold θ on all entries of M to convert a weighted adjacency matrix to an unweighted adjacency matrix. Specifically, for any pairs of vertices u and v in temporal graphs $\{G_t\}_{t=1}^T$, (u, v) is connected if and only if $M_{u,v}^{(t)} > \theta$, i.e., $(u, v) \iff M_{u,v}^{(t)} > \theta$.

3.3.3 Scan Statistics and Anomalous Community Identification

After the construction of temporal graphs $\{G_t\}_{t=1}^T$, we are able to apply our anomalous community detection technique on $\{G_t\}_{t=1}^T$. The goal is to find change-points at which a subgroup of neurons show an excessive increase of associations. In our settings, we need to specify the type of locality statistic and five parameters $(\theta, \tau, \ell, k, \Delta)$ as inputs of the detection algorithm.

In all experiments below, we select $k = 1$ because the graph order ($n = 5,105$) is in a moderate scale. Without loss of generality, we also select $\Delta = 50$, but all methodologies and analyses below can be trivially adapted to other Δ values. After truncating the first $\tau + \ell$ time steps for vertex-standardized and temporal normalizations, Figure 3.6 depicts $S_{\tau,\ell,k}(t; \Psi)$ (sea-green) and $S_{\tau,\ell,k}(t; \Phi)$ (orange) given $(\theta, \tau, \ell) = (0.8, 10, 10)$; and Figure 3.7 depicts $S_{\tau,\ell,k}(t; \Psi)$ (sea-green) and $S_{\tau,\ell,k}(t; \Phi)$ (orange) given $(\theta, \tau, \ell) = (0.8, 5, 5)$. Based on Figure 3.6 and 3.7, we select time stamps $t^\star = \{16, 44, 59, 78, 129, 218, 238\}$ (seconds), which are alarmed as change-points and pink-marked with arrows, for further investigations. For each $t^\star$, the anomalous community $N_1[v^\star; G_{t^\star}]$ is plotted in Figure 3.8 when the locality

statistic is Ψ and in Figure 3.9 when the locality statistic is Φ. For comparison, $N_1[v^\star; G_{t^\star-1}]$ and $N_1[v^\star; G_{t^\star+1}]$ are also given to visualize the increment or shift of neighbors of $v^\star$.

According to Figures 3.6, 3.7, 3.8, and 3.9, our findings are summarized in the following Table 3.1. This table indicates that change-points $t^\star = \{78, 129, 218\}$ can be detected by both $S_{\tau,\ell,k}(t; \Psi)$ and $S_{\tau,\ell,k}(t; \Phi)$. On the other hand, due to the difference between detection mechanisms using different locality statistics, $S_{\tau,\ell,k}(t; \Phi)$ is able to detect change-points $t^\star = \{44, 238\}$, while $S_{\tau,\ell,k}(t; \Phi)$ failed; $S_{\tau,\ell,k}(t; \Psi)$ is able to detect change-points $t^\star = 59$, while $S_{\tau,\ell,k}(t; \Phi)$ failed. Note that four ground truths of change-points observed by a lab scientist, i.e., $t^\star = \{16, 59, 78, 218\}$, are all correctly alarmed by either $S_{\tau,\ell,k}(t; \Psi)$ or $S_{\tau,\ell,k}(t; \Phi)$. This demonstrates the efficacy and practicality of our methodology. Furthermore, based on the last column of Table 3.1, and Figures 3.8 and 3.9, we demonstrate that at all selected change-points $S_{\tau,\ell,k}(t; \Psi)$ and $S_{\tau,\ell,k}(t; \Phi)$ capture different anomalous communities.

3.3.4 Detection Persistence Analysis

In the previous section, both $S_{\tau,\ell,1}(t; \Psi)$ and $S_{\tau,\ell,k}(t; \Phi)$ performed well on anomaly detection of neuron associations when $(\tau, \ell, \theta) = (10, 10, 0.8)$ or $(\tau, \ell, \theta) = (5, 5, 0.8)$. Under the above two particular settings of (τ, ℓ, θ), the practicality of $S_{\tau,\ell,1}(t; \Psi)$ and $S_{\tau,\ell,1}(t; \Phi)$ are validated by ground truths – identified anomalies are in fact triggered by tentative zebrafish eye or tail movements. However, there is a question remaining unanswered: How do $S_{\tau,\ell,1}(t; \Psi)$ and $S_{\tau,\ell,1}(t; \Phi)$ perform with other selections of

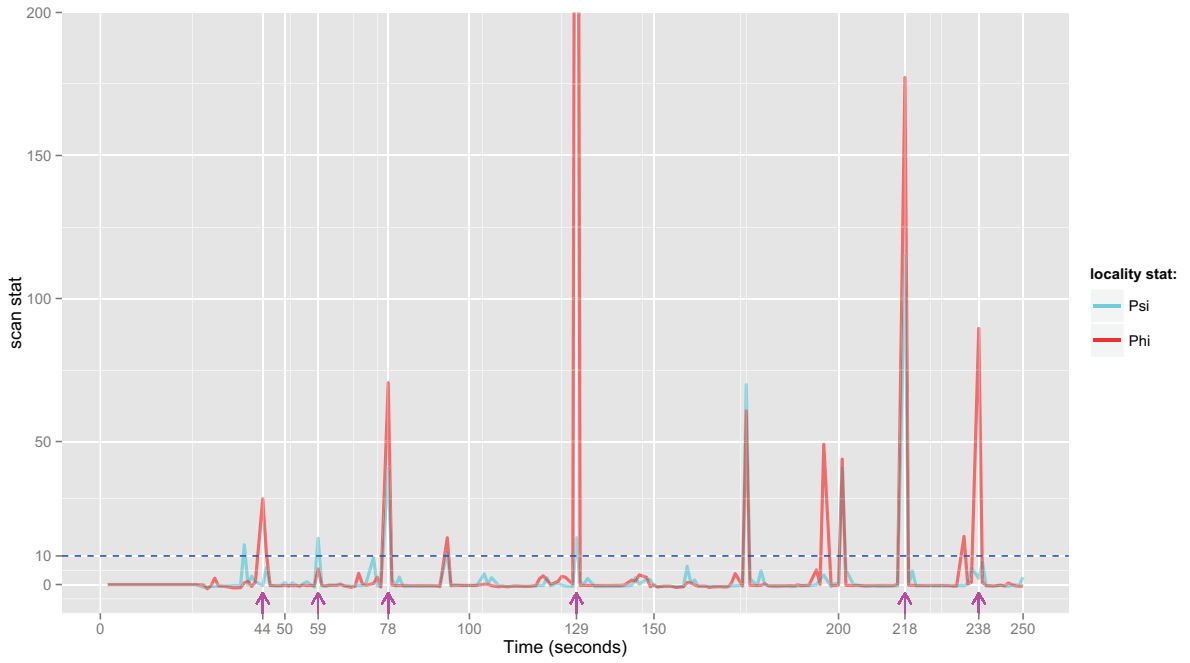


Figure 3.6: $S_{\tau,\ell,k}(t;\Psi)$(sea green) and $S_{\tau,\ell,k}(t;\Phi)$(orange), the temporally-normalized standardized scan statistics using $(\tau,\ell,\theta) = (10,10,0.8)$, in time series of zebrafish association-graphs across 250 seconds. Anomaly detection is indicated if $S_{\tau,\ell,k}(t;\cdot) > 10$ (blue dashed line). $t = 59th, 78th, 218th$ seconds are underlying change-points caused by zebrafish eye movement or tail movements. A summary of selected change-points (pink-marked with arrows) is provided in Table 3.1 and identifications of anomalous communities at selected change-points are provided in Figures 3.8 and 3.9.

(τ,ℓ,θ), and do detections still persist with varying parameters?

In this section, to eliminate the effect of parameter selection on performance evaluation, we investigate the persistence of detections by calculating $\{S_{\tau,\ell,1}(t;\cdot)\}_{t=1}^{T}$, fixing all but one parameters and varying the remaining target parameter. A persistent plot can be obtained where the x- axis is time (in seconds) and the y-axis presents the continuous values of the target parameter. The darkness/color at (x,y) entry is proportional to the scale of values of $S_{\tau,\ell,k}(t;\cdot)$ with $t = x$ and target parameter $= y$. If all values of $S_{\tau,\ell,k}(t;\cdot)$ across the target parameter (i.e., y-axis) are large at some $t = t^\star$, we claim that detection at $t^\star$ is *persistent* with respect to the target parameter when $S_{\tau,\ell,k}(t;\cdot)$ is used as test statistic. An ideal scenario would be that all

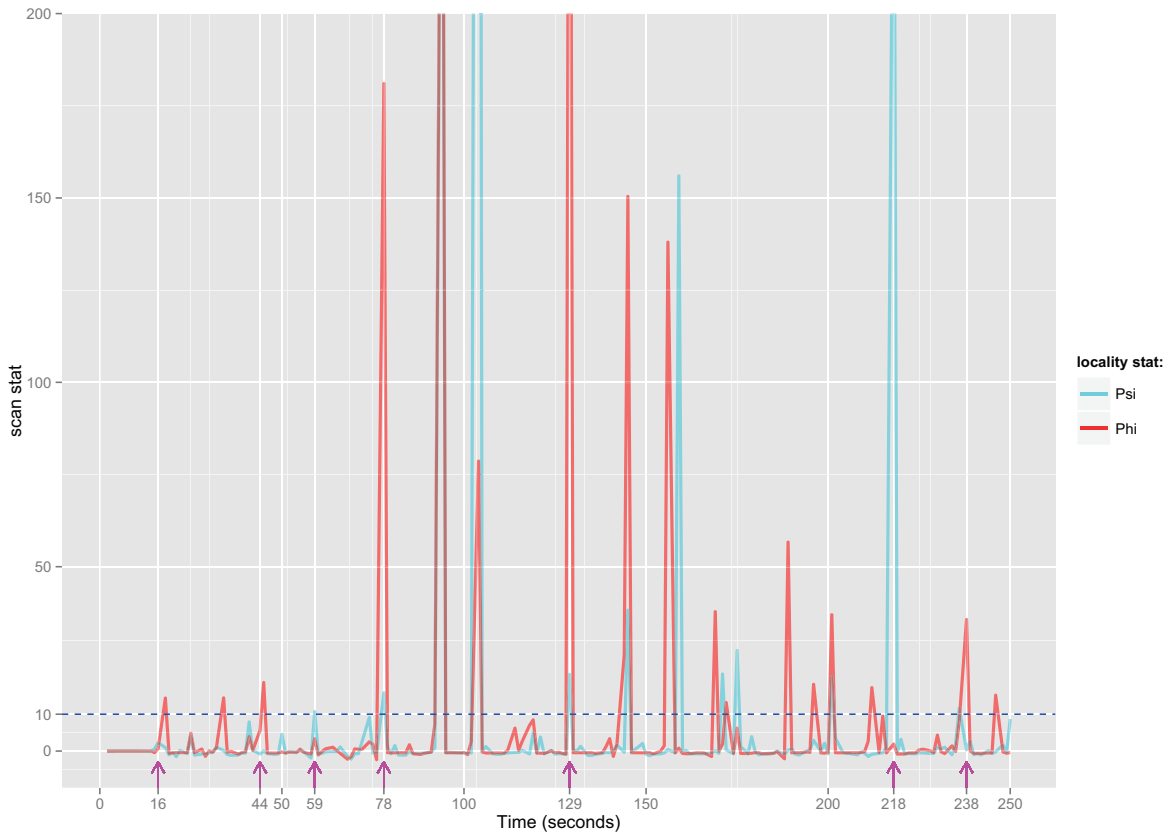


Figure 3.7: $S_{\tau,\ell,k}(t;\Psi)$(sea green) and $S_{\tau,\ell,k}(t;\Phi)$(orange), the temporally-normalized standardized scan statistics using $(\tau,\ell,\theta) = (5,5,0.8)$, in time series of zebrafish association-graphs across 250 seconds. Anomaly detection is indicated if $S_{\tau,\ell,k}(t;\cdot) > 10$ (blue dashed line). $t = 16th$ second is an underlying change-point at which the zebrafish is given a odor stimulus, and this stimulus lasts for 2 seconds. $t = 59th, 78th, 218th$ seconds are underlying change-points caused by zebrafish eye movement or tail movements. The Summary of selected change-points (pink-marked with arrows) is provided in Table 3.1, and identifications of anomalous communities at selected change-points are provided in Figures 3.8 and 3.9.

detections alarmed in previous section $t^\star = \{16, 44, 59, 78, 129, 218, 238\}$ are persistent with respect to θ, τ and ℓ, respectively.

Figure 3.10 is a persistent plot with respect to θ by fixing $(\tau,\ell) = (5,5)$ and letting θ range from 0.5 to 0.9 with step size 0.01. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths, we blue-mark another other top 8 persistent detections at the top time axis. The superiority of persistence at a particular $t^\star$ here is quantified by $\sum_{\theta=0.5}^{0.9} \mathbf{1}_{\{S_{\tau=5,\ell=5,k=1}(t=t^\star;\cdot)>10\}}$, the cumulative counts of

$t^\star$	$S_{10,10,1}(t;\Psi)$	$S_{10,10,1}(t;\Phi)$	$S_{5,5,1}(t;\Psi)$	$S_{5,5,1}(t;\Phi)$	$N_1[v^\star;G_{t^\star};\Psi]=N_1[v^\star;G_{t^\star};\Phi]$
16	NA	NA	×	✓	no
44	×	✓	×	✓	no
59	✓	×	✓	×	no
78	✓	✓	✓	✓	no
129	✓	✓	✓	✓	no
218	✓	✓	✓	×	no
238	×	✓	×	✓	no

Table 3.1: Summary results of anomaly detection on $\{G_t\}_{t=1}^T$ by employing $S_{\tau,\ell,1}(t;\cdot)$. Anomaly is indicated if $S_{\tau,\ell,1}(t;\cdot) > 10$. '✓' and '×' denote the success and failure of detection, respectively. NA is applicable in the case that $t^\star \leq \tau + \ell$, while $\{G_t\}_{t=1}^{\tau+\ell}$ are truncated for vertex standardization and temporal normalization. The last column $N_1[v^\star;G_{t^\star};\Psi] = N_1[v^\star;G_{t^\star};\Phi]$ tests if identified anomalous communities are the same when using different underlying locality statistic Ψ and Φ.

alarmed detections across varying θ. We can see that, in general, $S_{\tau=5,\ell=5,k=1}(t;\Phi)$ is more persistent than $S_{\tau=5,\ell=5,k=1}(t^\star;\Psi)$ at $t^\star = \{16, 45, 238\}$ and achieves earlier detections at $t^\star = 212$ instead of $t^\star = 218$. However, all other detections at $t^\star = \{59, 78, 129\}$, discovered in the previous section, show clear persistences with respect to θ using both underlying locality statistics.

Similarly, Figures 3.11 and 3.12 are persistent plots with respect to τ with underlying scan statistics $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$, respectively. The superiority of persistence at a particular $t^\star$ is quantified by $\sum_{\tau=2}^{10} \mathbf{1}_{\{S_{\tau,\ell,k=1}(t=t^\star;\cdot)>10\}}$, the cumulative counts of alarmed detections across varying τ. We can see that detections at $t^\star = \{78, 129, 218\}$ are persistent with respect to τ regardless of the selections of ℓ and the type of locality statistic. On the other hand, $S_{\tau,\ell,k=1}(t;\Phi)$ is more persistent than $S_{\tau,\ell,k=1}(t^\star;\Psi)$ at $t^\star = \{44, 238\}$, but $S_{\tau,\ell,k=1}(t^\star;\Psi)$ is more persistent at

$t^\star = 59$. This finding matches well with our conclusion from Table 3.1.

Figures 3.13 and 3.14 are persistent plots with respect to ℓ with underlying scan statistics $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$, respectively. The superiority of persistence at a particular $t^\star$ is quantified by $\sum_{\ell=2}^{10} \mathbf{1}_{\{S_{\tau,\ell,k=1}(t=t^\star;\cdot)>10\}}$, the cumulative counts of alarmed detections across varying τ. Comparing Figures 3.13 and 3.14, we are able to obtain the same conclusion shown in Figures 3.11 and 3.12. Furthermore, note that as ℓ increases, the signal or consistence of detection, quantified through values of scan statistics $S_{\tau,\ell,k}(t;\cdot)$, is often weakened in both Figures 3.13 and 3.14. This is reasonable because the signal is going to be smoothed out if there is a large number of temporal normalizations. Thus, this observation will not have influence on our conclusion about the persistent detections with respect to ℓ.

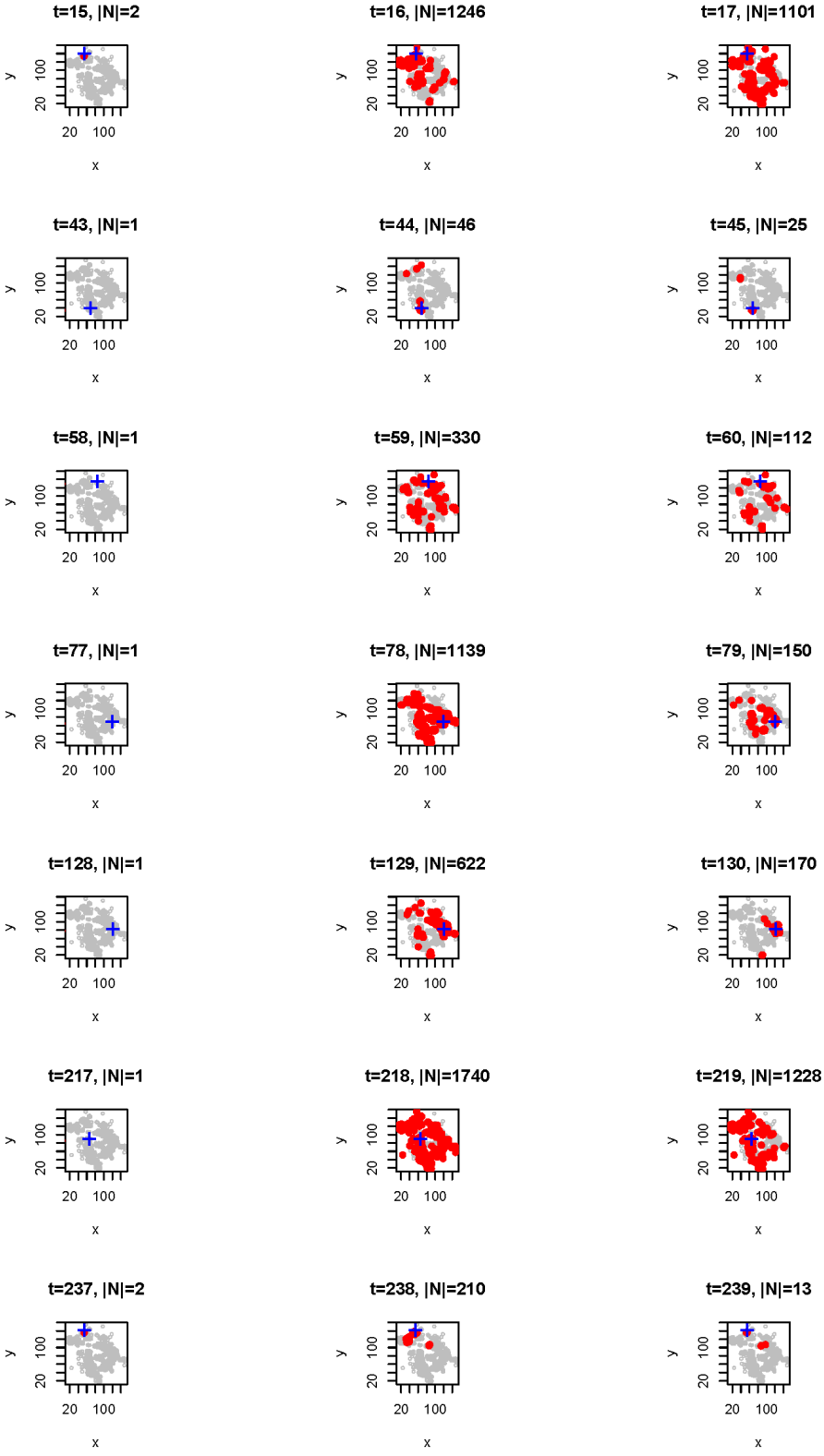


Figure 3.8: For each $t^\star \in \{16, 44, 59, 78, 129, 218, 238\}$, the members of anomalous community $N_1[v^\star; G_{t^\star}]$ are visualized in red when $S_{\tau,\ell,1}(t; \Psi)$ is employed for detection with $(\tau, \ell, \theta) = (5, 5, 0.8)$. All neurons are spatially located according to their (x,y) coordinates. "+" denotes $v^\star = \arg\max_v(\widetilde{J}_{t^\star,\tau;k}(v))$, the center of the anomalous community. "$|N|$" denotes the cardinality of $N_1[v^\star; G_{t^\star}]$. For example, when $t^\star = 59$, there are $|N| = 330$ neurons in $N_1[v^\star; G_{t^\star}]$. For comparison, $N_1[v^\star; G_{t^\star-1}]$ and $N_1[v^\star; G_{t^\star+1}]$ are also included at the left and right of each row.

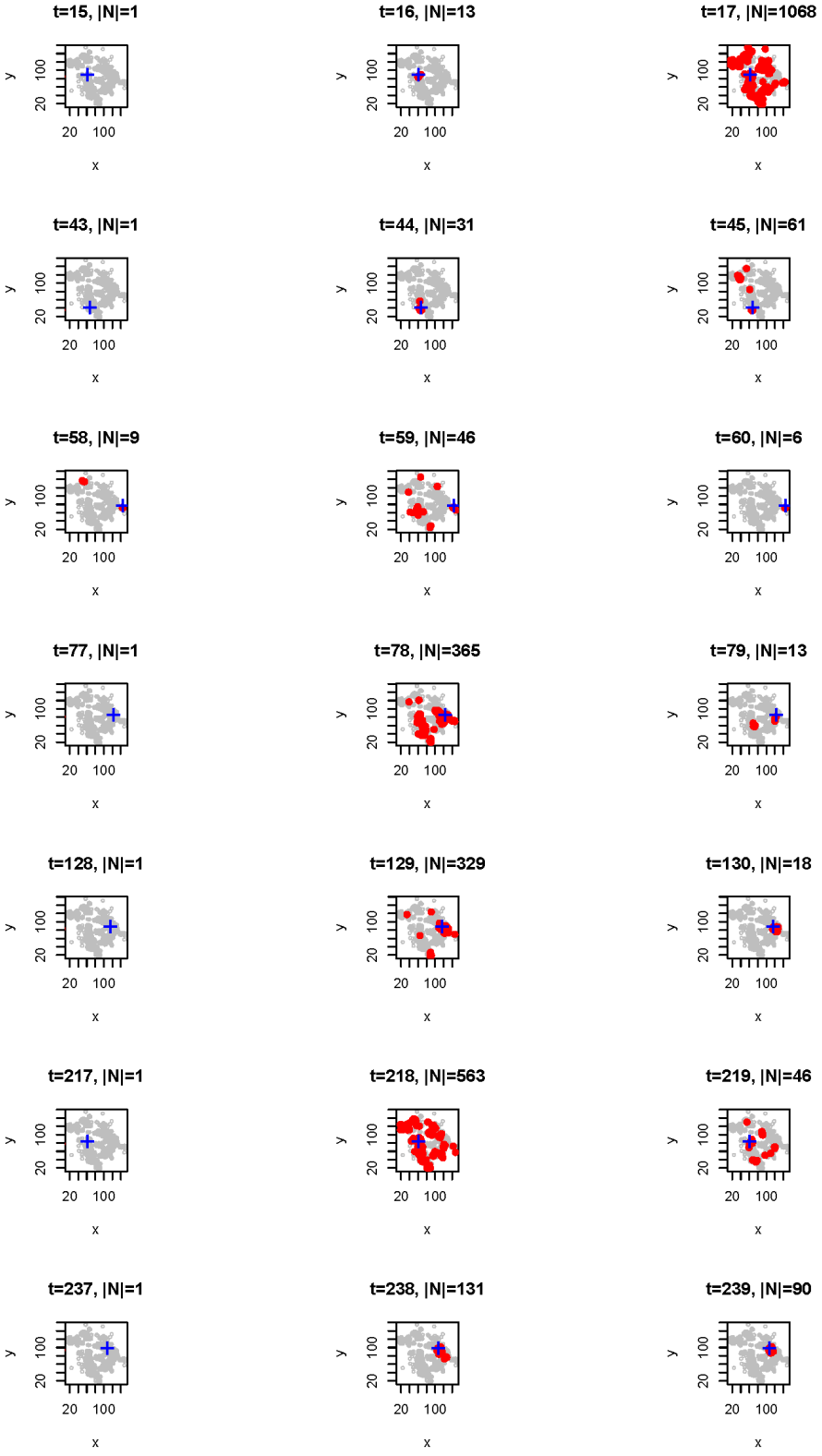


Figure 3.9: For each $t^\star \in \{16, 44, 59, 78, 129, 218, 238\}$, the members of anomalous community $N_1[v^\star; G_{t^\star}]$ are visualized in red when $S_{\tau,\ell,1}(t; \Phi)$ is employed for detection with $(\tau, \ell, \theta) = (5, 5, 0.8)$. All neurons are spatially located according to their (x,y) coordinates. "+" denotes $v^\star = \arg\max_v(\widetilde{J}_{t^\star,\tau;k}(v))$, the center of the anomalous community. "$|N|$" denotes the cardinality of $N_1[v^\star; G_{t^\star}]$. For example, when $t^\star = 59$, there are $|N| = 330$ neurons in $N_1[v^\star; G_{t^\star}]$. For comparison, $N_1[v^\star; G_{t^\star-1}]$ and $N_1[v^\star; G_{t^\star+1}]$ are also included at the left and right of each row.

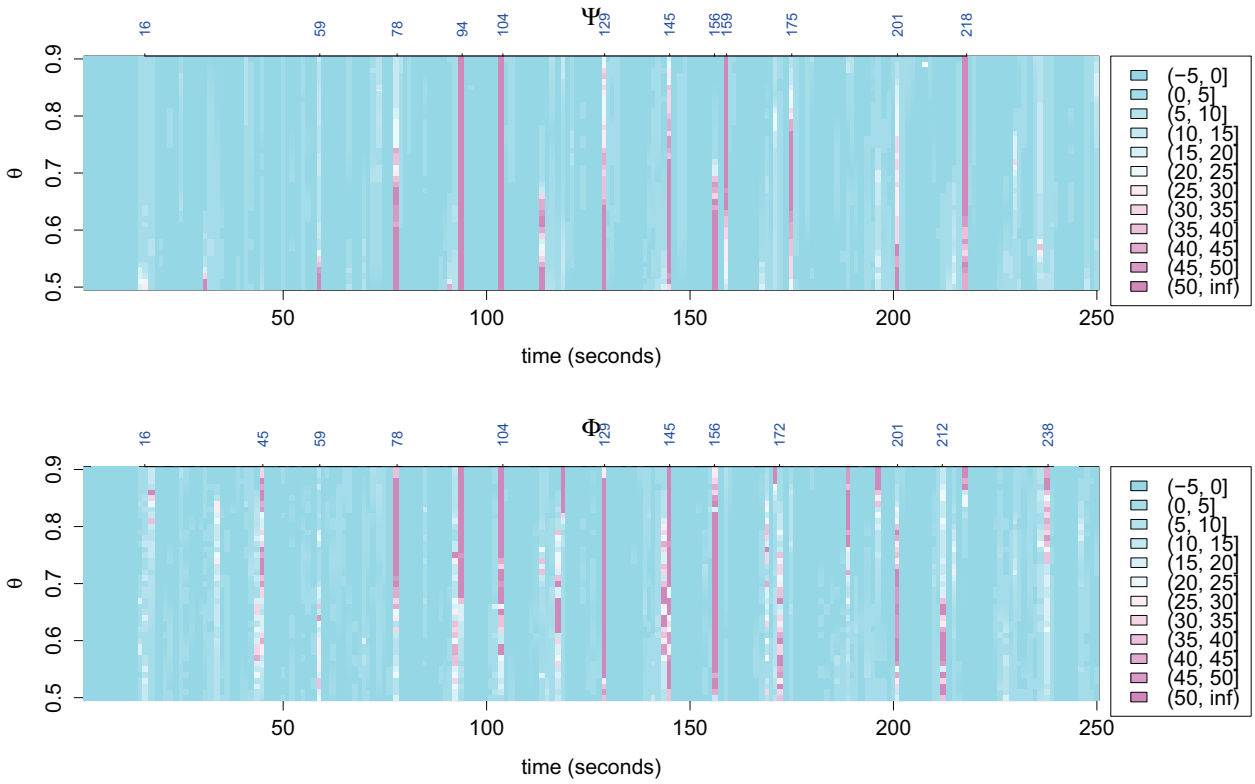


Figure 3.10: Persistent plot with respect to θ by fixing $(\tau, \ell) = (5, 5)$ and allowing θ to range from 0.5 to 0.9 with step size 0.01. Upper and lower subfigures correspond to the test statistics used, $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$, respectively. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths (eye/tail movements), other top 8 persistent detections are also blue-marked at the top time axis. Values of $S_{\tau,\ell,k}(t;\cdot)$ at all entries are quantitatively displayed by colors in legend.

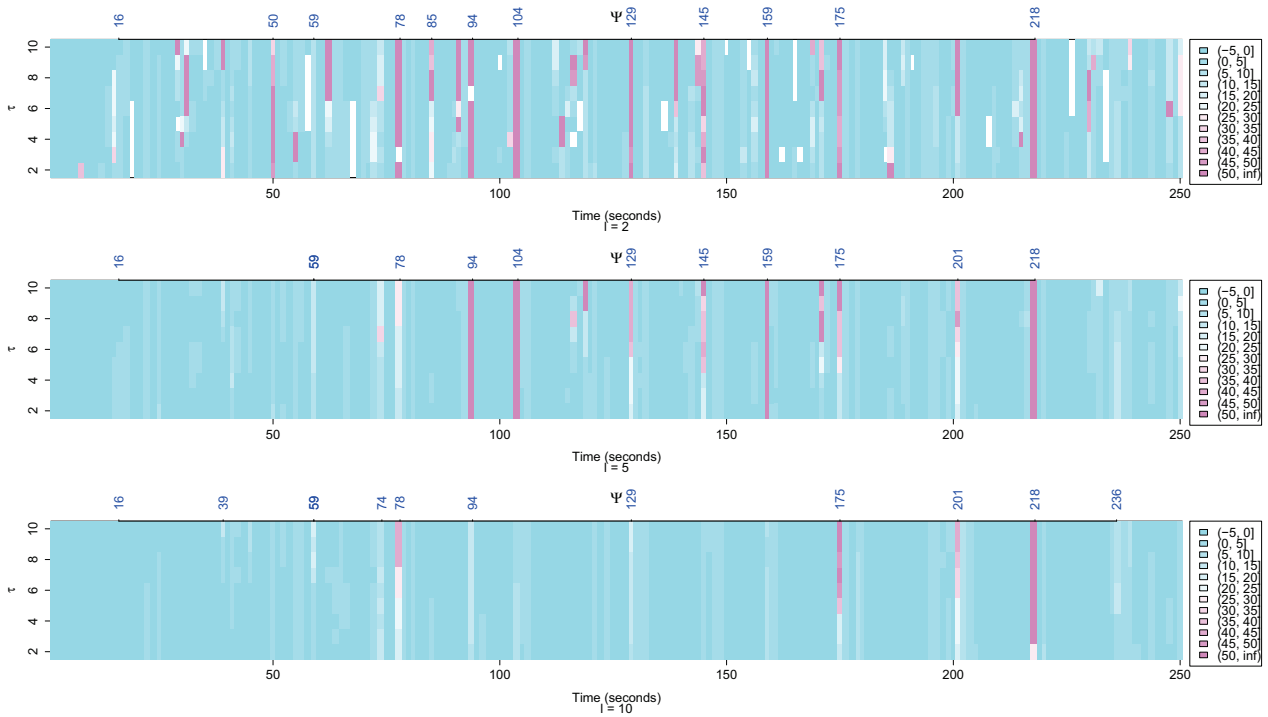


Figure 3.11: Persistent plot with respect to τ by fixing $(\theta, \ell) = (0.8, 2)$(upper), $(\theta, \ell) = (0.8, 5)$(middle), $(\theta, \ell) = (0.8, 10)$(lower) and allowing τ to range from 2 to 10 with step size 1. The test statistic used is $S_{\tau,\ell,k}(t; \Psi)$. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths (eye/tail movements), other top 8 persistent detections are also blue-marked at the top time axis. Values of $S_{\tau,\ell,k}(t; \cdot)$ at all entries are quantitatively displayed by colors in legend.

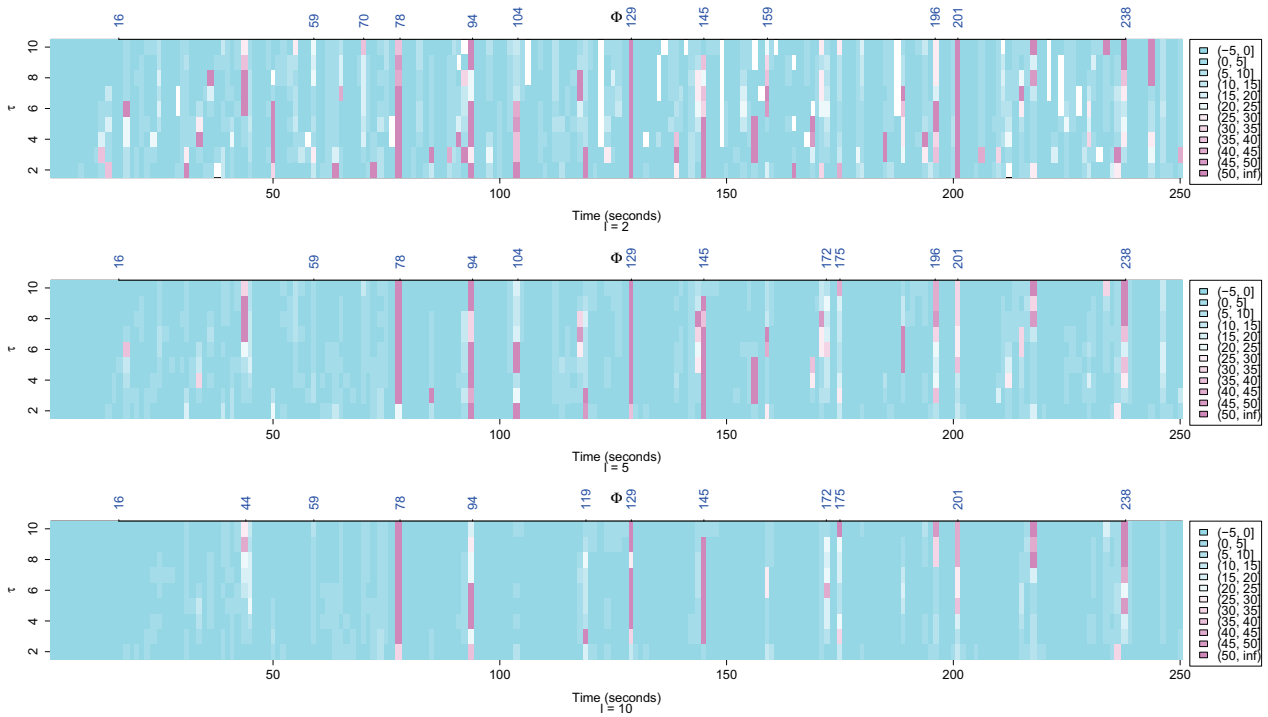


Figure 3.12: Persistent plot with respect to τ by fixing $(\theta, \ell) = (0.8, 2)$(upper), $(\theta, \ell) = (0.8, 5)$(middle), $(\theta, \ell) = (0.8, 10)$(lower) and allowing τ to range from 2 to 10 with step size 1. The test statistic used is $S_{\tau,\ell,k}(t; \Phi)$. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths (eye/tail movements), other top 8 persistent detections are also blue-marked at the top time axis. Values of $S_{\tau,\ell,k}(t; \cdot)$ at all entries are quantitatively displayed by colors in legend.

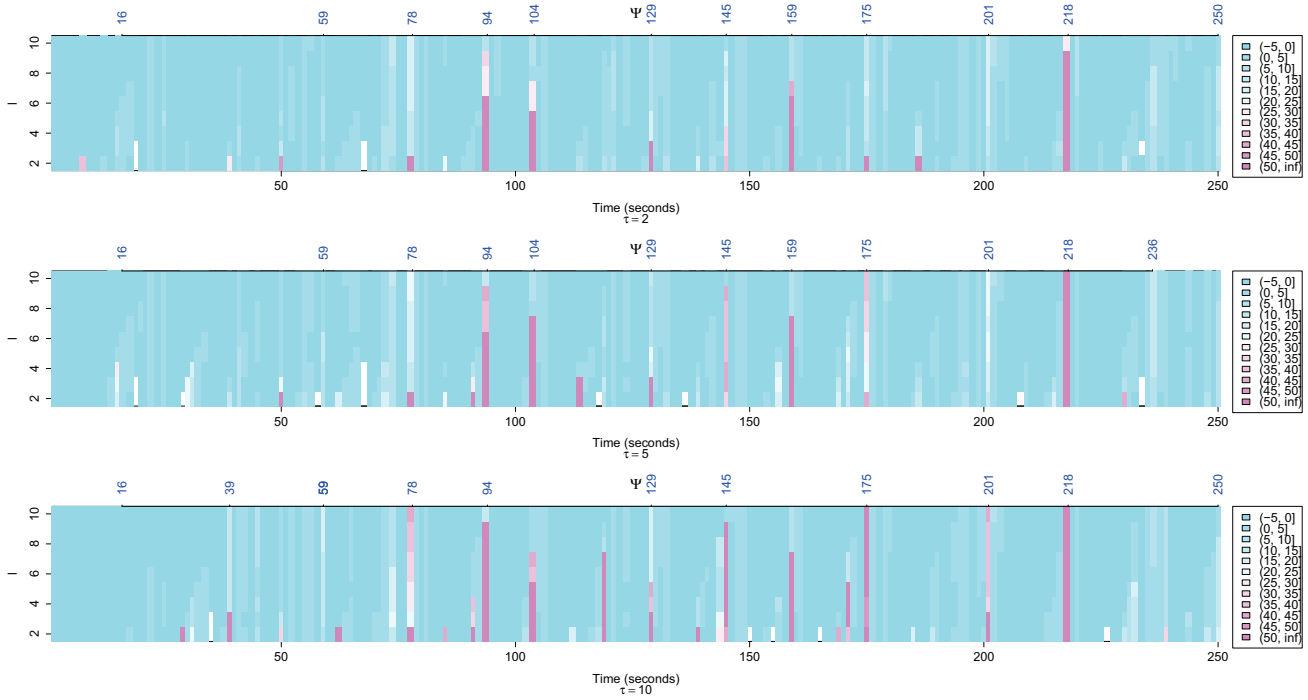


Figure 3.13: Persistent plot with respect to ℓ by fixing $(\theta, \tau) = (0.8, 2)$(upper), $(\theta, \tau) = (0.8, 5)$(middle), $(\theta, \tau) = (0.8, 10)$(lower) and allowing ℓ to range from 2 to 10 with step size 1. The test statistic used is $S_{\tau,\ell,k}(t; \Psi)$. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths (eye/tail movements), other top 8 persistent detections are also blue-marked at the top time axis. Values of $S_{\tau,\ell,k}(t; \cdot)$ at all entries are quantitatively displayed by colors in legend.

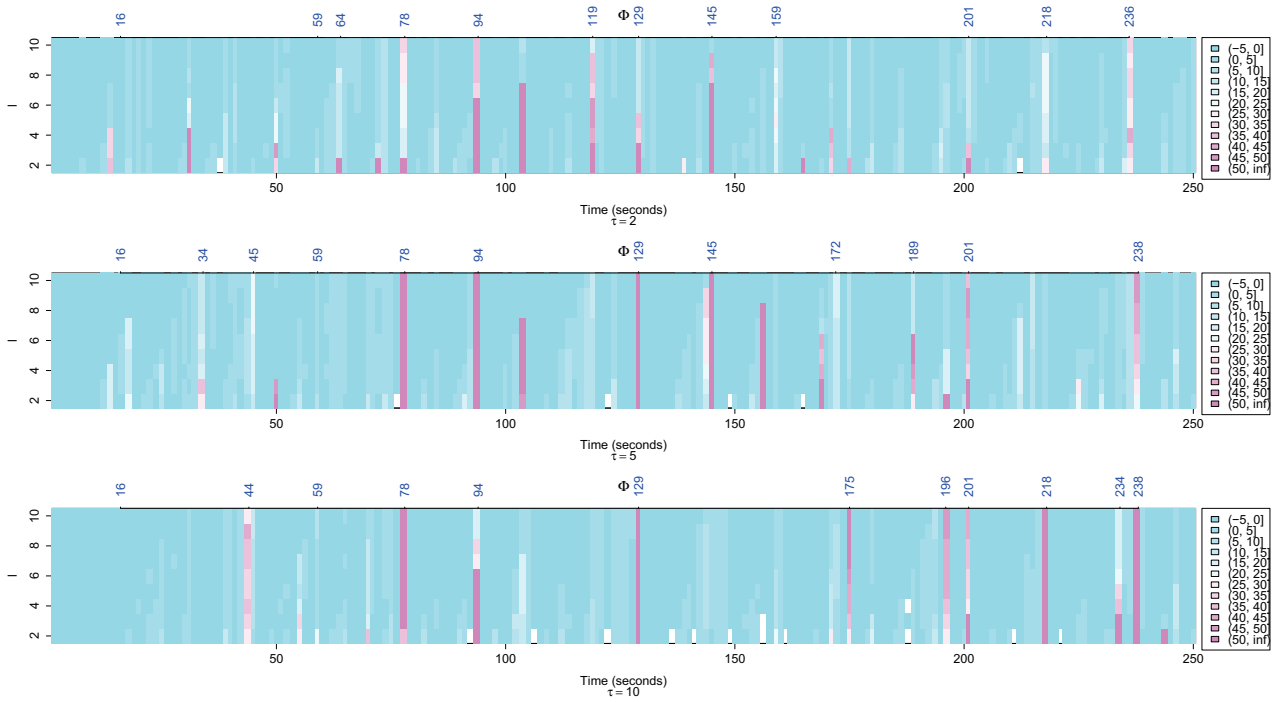


Figure 3.14: Persistent plot with respect to ℓ by fixing $(\theta, \tau) = (0.8, 2)$(upper), $(\theta, \tau) = (0.8, 5)$(middle), $(\theta, \tau) = (0.8, 10)$(lower) and allowing ℓ to range from 2 to 10 with step size 1. The test statistic used is $S_{\tau,\ell,k}(t; \Phi)$. Besides four time stamps $t^\star = \{16, 59, 78, 218\}$ having ground truths (eye/tail movements), other top 8 persistent detections are also blue-marked at the top time axis. Values of $S_{\tau,\ell,k}(t; \cdot)$ at all entries are quantitatively displayed by colors in legend.

Chapter 4

Active Community Detection in Massive Graphs

A canonical problem in graph mining is the detection of dense communities. This problem is exacerbated for a graph with a large order and size - with respect to the number of vertices and edges - as many community detection algorithms scale poorly. In this chapter, we propose a novel framework for detecting active communities that consists of the most active vertices in massive graphs. The framework is applicable to graphs having billions of vertices and hundreds of billions of edges. Our framework utilizes a parallelizable trimming algorithm based on a locality statistic to filter out inactive vertices, and then clusters the remaining active vertices via spectral decomposition on their similarity matrix. We demonstrate the validity of our method with synthetic stochastic block model graphs, using the Adjusted Rand Index as the performance metric. We further demonstrate its practicality and efficiency on a real-world hyperlink Web graph consisting of over 3.5 billion vertices and 128 billion edges.

An outline of this chapter is given as follows. § 4.1 provides the motivation of our concentration solely on active members in a massive network and summarizes the main contributions of our framework. One of the locality statistics proposed in § 2.3 will be revised in § 4.2 to suit a static graph setting. § 4.3 presents the procedures in our active community detection algorithm framework. § 4.4 describes a parallelizable trimming algorithm that cost-effectively skips actual computation on the majority of vertices. In § 4.5, our detection algorithm is empirically validated on graphs with true and known community structures. For the real data experiment in § 4.6, we apply the proposed algorithm on the massive hyperlink graph collected recently in [28].

4.1 Motivation

To the best of our knowledge, in the field of community detection on static graphs, almost all popular partitioning or clustering procedures are computed from the full topology of a graph and thus have high computation complexity. Their objective function values for optimizing clusters are determined by cluster labels of all vertices. That is, all vertices are involved in each step of these clustering algorithms, such as a modularity-based algorithm, spectral decomposition-based algorithm, etc. The problem is that it is challenging to run these algorithms on a billion-scale graph. For example, the most recent Hyperlink Graph has 3.5 billion and 128 billion edges [28], the largest graph available to the public. Even growing at $O(m)$ in each iteration, Louvain clustering and spectral clustering potentially require many iterations to converge, which is computationally challenging to work at the billion scale, let alone algorithms with the complexity of $O(n^2 \log n)$ or $O(nm^2)$ introduced in § 1.1. Thus, it is important to consider the situation where a graph is too large to be processed on its full topology.

Moreover, sometimes it is only dense and comparatively active groups of vertices that we are concerned with in graph analysis. Dense clusters consisting of only inactive vertices in a giant network, e.g., small cliques incorporating only insignificant websites in the Hyperlink Graph, are unimportant for observers. In this scenario, investigations solely on active vertices are sufficient to detect potential communities consisting of the most active vertices. In this work, we propose to use a locality statistic [50] to measure the activity level of a vertex. The communities that consist of the most active vertices are referred to as "active communities". For example, some link farms in web graphs are "active communities".

The contribution mainly has two facets. Firstly, we propose an alternative community detection framework because it is unattainable to cluster on an entire massive graph due to the large graph order or size and it is only active vertices that are important in many networks. The framework identifies the most *active vertices*, i.e., the ones of the largest locality statistic values, builds a smaller graph over active vertices and then assigns the most active vertices into communities through typical clustering methods. Secondly, to unearth the most active vertices in a network, we provide a highly parallelizable trimming algorithm to screen out inactive vertices. The number of discovered active vertices is much smaller than graph order n. We apply our methodology on the famous Hyperlink Graph [28] to identify active communities. To the best our knowledge, this is the first community detection algorithm applied to a real graph dataset at this scale. As a note, in this chapter we consider only directed and unweighted graphs without self-loops. All procedures can be easily adapted to undirected or weighted graphs if necessary (§ 5)

4.2 Locality Statistic $\Psi_k(v)$

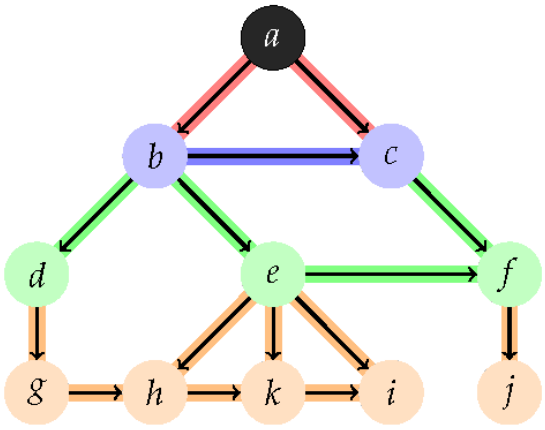
The locality statistic $\Psi_{t;k}(v)$ has been introduced in § 2.3 Eq.(2.3.1), used in temporal graph mining to detect a local region in the graph with significantly excessive intra-region connections [50]. If t is fixed, the locality statistic $\Psi_{t;k}(v)$ is the number of edges within the k-th order neighborhood of v in G_t. In a massive graph, k can be seen as the implicit and limited horizon that a vertex often reaches within the network. A vertex has a limited horizon because it is more likely to only interact with a subset of vertices in a massive graph and knows nothing about other parts of the graph [10]. As we know, a large locality statistic $\Psi_{t;k}(v)$ foreshadows the existence of a dense k-th order neighborhood centering at v. Hence, the locality statistic $\Psi_{t;k}(v)$ becomes a measure of activity level of the vertex v in the network G_t. Since our focus is a static directed graph below, we naturally refine $\Psi_{t;k}(v)$ to be $\Psi_k(v)$ by dropping unnecessary time information t.

Formally, let G be a graph. The locality statistic $\Psi_k(v)$ for all $k \geq 1$ and $v \in V$ on G is defined as

$$\Psi_k(v) = |E(\Omega(N_k[v; G], G))|. \tag{4.2.1}$$

For simplification, we re-denote $N_k[v; G]$ as $N_k[v]$ here because there is only a single G to consider in this chapter. Since G is unweighted, $\Psi_k(v)$ counts the number of edges in the subgraph of G induced by $N_k[v] = \{u \in V : d(u, v) \leq k\}$, a local territory of v where all elements are at a distance at most k from v in G. Note that, in a directed graph G, $d(u, v)$ stands for the shortest path distance between u and v on the underlying undirected graph of G by removing orientations of all edges.

More specifically, if $k = 1$, the case thoroughly investigated below, $\Psi_1(v)$ counts the



k	$E(\Omega(N_k[\mathbf{a}; G]))$	$\Psi_k(\mathbf{a})$
0	⊖	2
1	⊖ + ⊖	3
2	⊖ + ⊖ + ⊖	7
3	⊖ + ⊖ + ⊖ + ⊖	15

	d	Ψ_1	Ψ_2	Ψ_3
a	2	3	7	15
b	4	5	14	15
c	3	4	8	15
d	2	2	8	14
e	5	7	15	15
f	3	3	12	15
g	2	2	7	14
h	3	4	10	15
i	2	3	7	15
j	1	1	3	12
k	3	5	8	15

Figure 4.1: A toy example to illustrate calculations of $\Psi_k(a)$ with various $k = 0, 1, 2, 3$, on the directed G. For example, if $k = 2$, $N_2[a] = \{u \in V : d(u, a) \leq 2\} = \{a, b, c, d, e, f\}$, and thus $E(\Omega(N_2[a], G))$ contains edges colored in red, blue, and green.

number of edges either incident to v or involved in triangles containing v. Large locality statistic implies a dense region whose members are all inclined to be "friends" with each other, and such a region is not necessarily limited to a clique. If we use a clique to locate a dense region, a subgraph $N_k[v]$ such as the one with all possible internal links except one is undervalued even though it is an extremely cohesive region. In a slight abuse of notation, we let $\Psi_0(v)$ be the sum of in-degree and out-degree of v. A simple toy example (Figure 4.1) illustrates calculations of $\Psi_k(a)$ with $k = 0, 1, 2, 3$, on the directed graph G.

4.3 Detection Framework

The proposed detection framework is mainly composed of three steps and the first step is novel. We elaborate the rational behind each step. Given a graph, our detection framework is summarized as follows:

(i) Find the set of the top Q most active vertices $\mathcal{C}$, i.e., the ones of the Q largest locality statistic values $\Psi_k(v)$; we re-denote them as $\mathcal{C} = \{v_1, v_2, \ldots, v_Q\}$.

(ii) Construct a similarity matrix $\mathcal{S}$ for vertices in $\mathcal{C}$. $\mathcal{S}$ is a $Q \times Q$ matrix where $\mathcal{S}_{i,j}$ measures similarity between vertex v_i and v_j.

(iii) Run a clustering algorithm on the similarity matrix $\mathcal{S}$ and report clusters as active communities.

Step (i) employs the locality statistic as a quantity to identify the Q largest hubs in a network whose activity level is evaluated in the k-th order neighborhood. Similar criteria of defining local activity level are proposed in [2] [9]: [2] uses the L shell method to agglomeratively find a community for each vertex, which has extremely heavy computational burden in large graphs; [9] distinguishes vertices based on high clustering coefficients, which may trigger false alarms on small cliques such as triangles in our problem. It is computationally expensive to compute locality statistic on all vertices. For example, if $k = 1$, the computation on all vertices has an equivalent complexity of triangle counting in the same graph [44]. The complexity is $O(md_{max})$ where d_{max} is the largest vertex degree in the graph. Thus, we deploy a trimming algorithm to obtain the top Q largest locality statistic values, shown in § 4.4 for the case $k = 1$. As Q is a user-defined input, it

determines the computational burden of Step (i) and is explored in §4.5. Also, note that in certain applications some of the most active vertices might be considered as outliers if they are not connected to the rest of top Q vertices. These outliers are unnecessarily taken into account for community identification. In that case, a larger Q should be used and the largest outlying vertices should be trimmed. We do not use this assumption in this work but all methodologies and arguments can be easily adapted for this assumption.

After identifying the most active vertices, our framework uses the spectral clustering approach instead of popular modularity-based methods for several reasons. First of all, modularity optimization requires information of the whole graph so that information extracted solely from active vertices is less likely to be applicable to modularity-based methods. Furthermore, modularity maximum does not necessarily mean that a graph has a community structure and also high proximate modularities can fail to be similar partitions [10]. These reasons lead us to construct a similarity matrix on $\{v_1, \ldots, v_Q\}$.

In Step (ii), the problem of quantifying similarities between active vertices has received significant attention. Two main types of similarities have been studied in a large number of application domains: vertex feature based and network structure based. The former quantifies similarities resting on attribute values of each vertex such as [54]. The latter focuses only on graph topologies: a pair of vertices achieve a high degree of similarity if they share many neighbors. Some classic measures can be used, such as *Jaccard Index*

$$S_{ij} = \frac{|N_k[v_i] \cap N_k[v_j]|}{|N_k[v_i] \cup N_k[v_j]|}$$

where $N_k[v]$ is the set of vertices at distance at most k from v in original graph G,

Cosine Similarity [43]

$$S_{ij} = \frac{|N_k[v_i] \cap N_k[v_j]|}{|N_k[v_i]||N_k[v_j]|}$$

where $N_k[v]$ is the set of vertices at distance at most k from v in original graph G, and another normalized overlap [41]

$$S_{ij} = \frac{|N_k[v_i] \cap N_k[v_j]|}{\min(|N_k[v_i]|, |N_k[v_j]|)}.$$

Alternatively, an algebraic approach of iteratively making use of the adjacency matrix is also proposed in [26] to construct a similarity matrix. Assessment of similarities between active vertices $\{v_1, v_2, \ldots, v_Q\}$ is not the main aim of this work. Whether to select a feature based or network structure based approach or which classic measure to be used is application domain dependent.

Once the similarity matrix $\mathcal{S}$ is available, we can cluster the Q vertices through a large number of standard clustering algorithms such as Hierarchical Clustering, Gaussian Mixture Model Clustering, Self-organizing maps, Graph Spectral Clustering, etc. In our approach, we prefer to cluster the Q vertices, in a space obtained from eigenvectors, through spectral clustering [34] on $\mathcal{S}$ because the representation induced by eigenvectors enables the clustering distinctness of initial data points to be more evident [10]. With the spectral decomposition above, the large gaps between consecutive eigenvalues suggest the number of clusters in a graph. Additionally, if the computation of all Q eigenvectors, whose complexity grows at $O(Q^3)$, is unattainable, the Lanczos algorithm [5] is recommended to compute leading eigenvectors of $\mathcal{S}$ as a few leading eigenvectors suffice to achieve good partitions.

4.4 Framework Implementation

As a key distribution, this section presents a trimming algorithm (§ 4.4.1) that efficiently identifies the most active vertices in a graph. Its implementation is discussed in § 4.4.2 and § 4.4.3 from both shared memory and external memory prospectives respectively.

4.4.1 Trimming Algorithm

In the first step of our framework, we need to identify the vertices of the largest locality statistic values in a massive graph. It is inefficient to compute locality statistic values of all vertices, while we only need to identify the largest ones. As shown by *local_stat* in Figure 4.2, $\Psi_1(v)$ counts the number of edges, of which adjacent vertices are both in $N_1[v]$, in the collection of incident edges of vertices in $N_1[v]$. That is, $\Psi_1(v) = \frac{1}{2}\sum_{u \in N_1[v]} \sum_{e \in E[u]} \mathbf{1}_{\{S[e] \in N_1[v] \wedge D[e] \in N_1[v]\}}$, where $\mathbf{1}_{\{\cdot\}}$ is an indicator function. The complexity of computing $\Psi_1(v)$ of all vertices is $O(md_{max})$ and becomes especially intensive if v has more than millions of neighbors..

Therefore, we deploy a cost-effective trimming algorithm to safely skip the computation of $\Psi_1(v)$ on the vertices with small locality statistic, while still being able to identify the vertices with the Q largest locality statistic values. The trimming algorithm skips the wasteful computation based on the upper bound of the locality statistic of a vertex. The tighter upper bound we achieve, the more vertices on which we can skip computation. The procedures in the rest of the section describe the trimming algorithm that works for the first-order neighborhood.

```
1: function local_stat(v)
2:     lstat ← 0
3:     for all u ∈ N1[v] do
4:         for all e ∈ E[u] do
5:             if S[e] ∈ N1[v] and D[e] ∈ N1[v] then
6:                 lstat ← lstat + 1
7:     return lstat/2
```

Figure 4.2: $local\_stat(v)$ computes $\Psi_1(v)$. $S[e]$ denotes the source vertex of an edge e and $D[e]$ denotes the destination vertex of an edge e.

We develop two upper bounds of $\Psi_1(v)$ in our trimming optimization, shown by $est\_lstat1(v)$ and $est\_lstat2(v)$ in Figure 4.3. $est\_lstat1(v) = \Psi_0(v)^2 + \Psi_0(v)$, is a very loose but computationally efficient upper bound. Because v has at most $\Psi_0(v)$ neighbors, $\Psi_1(v) \leq \Psi_0(v)^2 + \Psi_0(v)$ and the equality holds when all neighbors of v are fully connected. $est\_lstat2(v)$ computes a much tighter upper bound and is also more computationally expensive. We denote by $contr_v(u)$ the amount of potential contribution of $u \in N_1[v]$ to $\Psi_1(v)$. The amount of contribution of u is measured by the number of edges incident to u and also counted in $\Psi_1(v)$. $\Psi_1(v)$ is upper bounded by the sum of $contr_v(u)$ over all neighbors in $N_1[v]$, i.e., $\Psi_1(v) \leq \sum_{u \in N_1[v]} contr_v(u)$. $contr_v(u)$ meets two inequalities: $contr_v(u) \leq \Psi_0(u)$ and $contr_v(u) \leq 2 \times |N_1[v]|$, because the number of distinct directed triangles incorporating both u and v is upper bounded by $\Psi_0(u)$ and $2|N_1[v]|$. Since $\sum_{u \in N_1[v]} contr_v(u)$ counts each potential edge twice, we divide the sum by two. Although $\frac{1}{2} \sum_{u \in N_1[v]} min(\Psi_0(u), |N_1[v]| \times 2)$ is not the tightest bound, it is sufficiently accurate to eliminate computation of locality statistic on most vertices.

Having upper bounds est_lstat1 and est_lstat2, we now describe our procedure of finding

```
1: function est_lstat1(v)
2:     return Ψ0(v)^2 + Ψ0(v)

1: function est_lstat2(v)
2:     est ← 0
3:     for all u ∈ N1[v] do
4:         est ← est + min(Ψ0(u), |N1[v]| × 2)
5:     return est/2
```

Figure 4.3: $est\_lstat1(v)$ and $est\_lstat2(v)$ compute the upper bound of $\Psi_1(v)$. $est\_lstat2(v)$ computes a much tighter upper bound but requires more expensive computation.

$\arg\max_{v \in V} \Psi_1(v)$ over any set of vertices V, illustrated by *top_lstat* in Figure 4.4. The idea is to maintain the largest locality statistic discovered so far (*curr_max*) and skip expensive computation on the vertices whose upper bound of locality statistic is smaller than *curr_max*. Since *est_lstat*2 requires more computation than *est_lstat*1, we compute *est_lstat*1 first and only compute *est_lstat*2 if *est_lstat*1 is greater than *curr_max*. To reach $\arg\max_{v \in V} \Psi_1(v)$ early, the procedure starts from the vertices with the largest degree with an assumption that a larger-degree vertex is more likely to have a larger locality statistic. To accelerate finding top Q vertices, *top_lstat* returns not only $\arg\max_{v \in V} \Psi_1(v)$ but also all of the vertices whose locality statistic has been computed during the process of finding $\arg\max_{v \in V} \Psi_1(v)$.

By utilizing *top_lstat*, *topQ_lstat* (Figure 4.5) finds vertices with the Q largest locality statistic values. *topQ_lstat* takes two stages to look for vertices with the largest locality statistic. In the first stage, we find at least Q vertices of large locality statistic by repeatedly invoking *top_lstat* on the remaining vertices in the graph whose locality statistic is

```
1: function top_lstat(V, curr_max)
2:     sort V s.t. degree(V) in DESC
3:     V' ← {}
4:     for all v ∈ V do
5:         est ← est_lstat1(v)
6:         if est ≥ curr_max then
7:             est ← est_lstat2(v)
8:         if est ≥ curr_max then
9:             lstat ← local_stat(v)
10:            V' ← V' ∪ {v}
11:            curr_max ← max(lstat, curr_max)
12:    return V'
```

Figure 4.4: *top_lstat* computes the largest locality statistic among a set of vertices V.

1: **function** $topQ\_lstat(V, Q)$
2: $curr\_max \leftarrow 0$
3: $knownV \leftarrow \{\}$
4: **while** $|knownV| < Q$ **do**
5: $V' \leftarrow top\_lstat(V, 0)$
6: $V \leftarrow V \setminus V'$
7: $knownV \leftarrow knownV \cup V'$
8: sort $knownV$ s.t. $local\_stat(V)$ in DESC
9: $kth\_lstat \leftarrow 0$
10: **while** $kth\_lstat \neq local\_stat(knownV[Q])$ **do**
11: $kth\_lstat \leftarrow local\_stat(knownV[Q])$
12: $V' \leftarrow top\_lstat(V, kth\_lstat)$
13: $V \leftarrow V \setminus V'$
14: $knownV \leftarrow knownV \cup V'$
15: sort $knownV$ s.t. $local\_stat(V)$ in DESC

Figure 4.5: *topQ_lstat* finds the vertices of Q largest locality statistic values among V.

unknown. In the second stage, we use *top_lstat* to continue searching for vertices with the largest locality statistic among the remaining vertices in the graph whose locality statistic is unknown. The procedure stops when *top_lstat* can no longer discover a vertex whose locality statistic is larger than the current Qth largest locality statistic.

The complexity of computing top Q locality statistic values depends on both graph structures and the parameter Q. Theoretically, a very loose upper bound of the complexity is $O(md_{max})$, the complexity of computing locality statistic on all vertices. However, its

complexity in practice is much smaller when $Q \ll n$ because the trimming algorithm skips computation on the majority of the vertices in a graph. For example, if $Q = 100,000$, our algorithm only needs to compute locality statistic on $163,409$ vertices in the Hyperlink graph, which account for 0.0047% of vertices in the graph. The complexity of running est_lstat1 on all vertices is $O(n)$ and running est_lstat2 on all vertices is $O(m)$. Therefore, the complexity of the trimming algorithm throughout the entire computation is between $O(n)$ and $O(m)$ where the constant factor here is 1.

4.4.2 Shared-memory Parallel Implementation

In this section, we describe the parallel implementation of our algorithm in shared memory. Although trimming skips unnecessary computation on many vertices to speed up computation, a parallel implementation is still necessary for a graph with billions of vertices, especially in the era of multi-core processors. We implement our algorithm in FlashGraph [57], a programming framework for large-scale graph analysis. The implementation is written in C++.

We parallelize our implementation by parallelizing the function *top_lstat* since its computation on each vertex is independent. We split the vertices in a graph into multiple partitions and create a thread for each partition to process the vertices in the input set of *top_lstat* in parallel. Once a thread completes all vertices in its own partition, it steals vertices from other partitions and processes these stolen vertices.

However, a naive parallel implementation of the algorithm may have highly skewed workloads among threads due to the power-law distribution of vertex degree in many real-world

graphs. Our algorithm only needs to perform intensive computation (*local_stat* in Figure 4.2) on few vertices, which dominates the entire computation in *top_lstat*. Furthermore, the time of computing *local_stat* on different vertices varies significantly. Therefore, the naive load balancing scheme, which moves the computation of an entire vertex to another thread, is insufficient to evenly distribute the most intensive computation among threads.

Therefore, we further split computation of $\Psi_1(v)$ for better load balancing by splitting $N_1[v]$ into j parts $N_{1,1}[v], N_{1,2}[v], ..., N_{1,j}[v]$. Each part $N_{1,i}[v]$ is only responsible for computing the contribution to $\Psi_1(v)$ from its own part, i.e., computing the cardinality of the intersection of $N_1[v]$ and $N_1[u]$, for all $u \in N_{1,i}[v]$. When load balancing is triggered, the computation of $N_{1,i}[v]$ can be moved to another thread. Since there are many splits, each of which contains a small amount of computation, it is much easier to distribute computation evenly among threads.

An additional issue in the parallel implementation is to maintain the maximal locality statistic discovered currently in *top_lstat* without much locking overhead. Given the fact that the maximal locality statistic is updated very infrequently and the value increases monotonically, we always compare a new locality statistic with the current maximal value without locking before updating the maximal value with locking. As such, we avoid most locking for updating the maximal locality statistic. We do not lock when we read the maximal locality statistic. Even though we might read a stale value in a very rare case, it does not affect the correctness of our implementation. The worst case is that we need to compute locality statistic on slightly more vertices.

4.4.3 External-memory implementation

Given a graph with billions of vertices and hundreds of billions of edges, we can no longer store the entire graph in RAM in a single machine. With the advance of solid state drives (SSD) in hardware [11] and software [56], SSDs can now perform over one million I/Os per second. This makes SSDs a natural extension of RAM in large-scale data analysis, as illustrated by FlashGraph [57]. FlashGraph stores algorithmic vertex state in RAM and edge lists on SSDs. In order to scale, FlashGraph requires the size of vertex state to be a small constant.

We use a very compact data structure for our algorithm to store vertex state, which only occupies eight bytes per vertex. The eight bytes can be used to store the locality statistic of a vertex, the upper bound of the locality statistic, or a pointer to the neighbor list of a vertex. We keep the neighbor list of a vertex in memory only when we perform the expensive computation *local_stat* on the vertex. Therefore, we only maintain a small number of neighbor lists in RAM at a time. Furthermore, we read the edge lists of neighbor vertices from SSDs only when they are required. As a result, our implementation has a small memory footprint, compared with the graph storage size, which allows us to process graphs with billions of vertices in a single commodity machine.

4.5 Validation on Synthetic Graphs

This section looks into the performance of our active community detection methodology on a synthetic graph whose underlying probabilistic behaviors of network participants and true active community structures are known. To test the proposed framework on the synthetic graph, the behavior of Receiver Operating Characteristic (ROC) and Adjusted Rand Index(ARI) [27] [55] [10] are observed under three scenarios, where $k = 0, 1, 2$, to quantitatively evaluate how similar the partitions delivered by the framework are to the true partitions.

The performance of our detection framework is evaluated through synthetic experiments because the underlying randomness that governs a real network is usually unknown. The artificial graphs used in the synthetic experiments are generated from Stochastic Block Model (SBM) introduced in § 2.1. Note that SBM, a more generic version of Planted Partition Model [16] [7], is widely used as a testbed for community detection algorithms today [20] [10] [55] and gains the reputation of a standard benchmark. In a stochastic block model containing blocks $\{1, \ldots, B\}$, V is randomly partitioned into B distinct blocks $[n_1], \ldots [n_B]$, where $[n_i]$ denotes the vertices in block i. That is, each vertex is associated with one block membership between 1 to B. The connectivity probabilities among all vertices are characterized by a $B \times B$ symmetric Bernoulli rate matrix $\mathbf{P}$, where $\mathbf{P}_{i,j}$ denotes the block connectivity probability between blocks i and j.

In order to preserve sparsity, degree heterogeneity and built-in active community structures in a SBM graph, we select the following SBM parameter settings:

$$B = 4, n_1 = 940, n_2 = n_3 = n_4 = 20$$

and

$$\mathbf{P} = \begin{pmatrix} 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.2 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.3 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.4 \end{pmatrix},$$

Given the parameters above, G is a graph having 4 blocks where the majority block $[n_1]$ involves 94% actors of the network. All actors in $[n_1]$ connects with any other actor with success probability 0.01. Other blocks have their own distinct self-connectivity probabilities which are diagonal entries of matrix $\mathbf{P}$. Each of the blocks $i = 2$ up to B has self-connectivity probability $\mathbf{P}_{i,i} = 0.1 \times i$. The case where $\mathbf{P}_{4,4} > \mathbf{P}_{3,3} > \mathbf{P}_{2,2} \gg \mathbf{P}_{1,1}$ is of interest because we can consider $[n_2], [n_3], [n_4]$ as three built-in active communities whose inner connectivity level is anomalously high.

Figure 4.6 is a sample plot of graph adjacency matrix generated by using above setting in which black dots represent established edges. The three grids, 20×20 of each, with high intensities at the right bottom serve as three active communities to detect.

Figure 4.7 shows a sample graph configuration of G where the size of each vertex v is proportional to underlying $\Psi_{k=1}(v)$. For a pleasant visualization, only one-tenth of total edges are uniformly sampled and incorporated in the figure as $m = 10358$. Also, White (no label), yellow (label 2), red (label 3) and green (label 4) clusters stand for blocks $[n_1], [n_2], [n_3]$ and $[n_4]$ respectively. We observe that sizes of vertices belonging to colored clusters are more likely to be larger than the ones in the majority white block. This phenomena foreshadows the rationale of using top Q locality statistic values to cut off a massive number of negligible vertices which are unlikely to be in active communities.

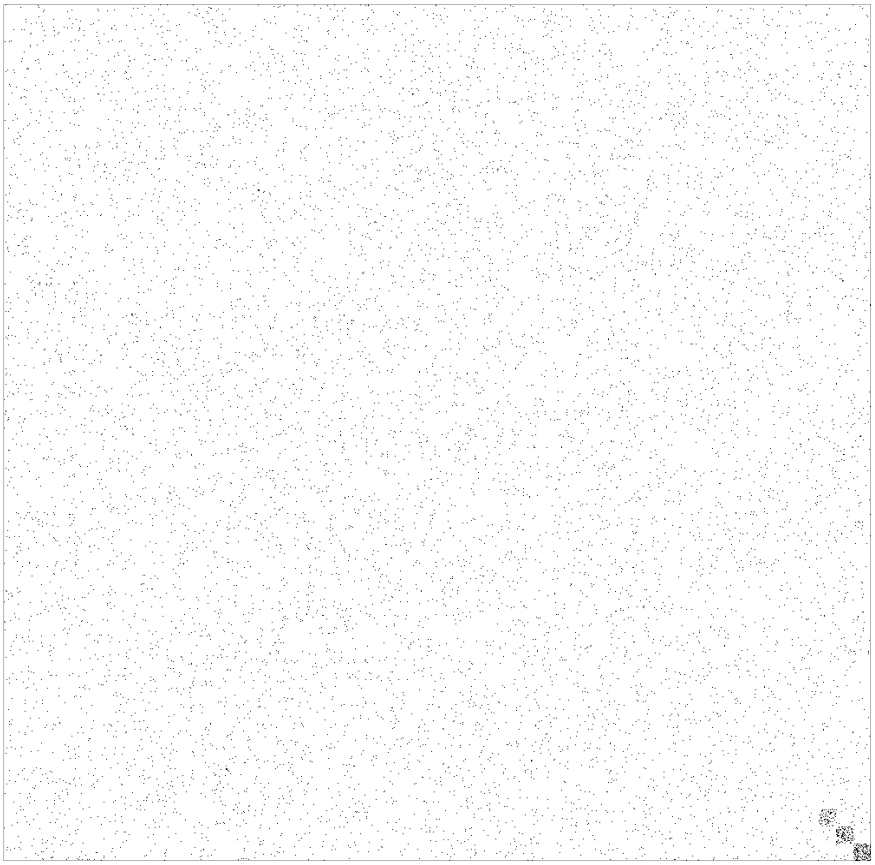


Figure 4.6: The adjacency matrix configuration of one sampled graph G generated through the Stochastic Block Model. The SBM parameters are: $B = 4, n_1 = 940, n_2 = n_3 = n_4 = 20$, and block connectivity matrix is given in $\mathbf{P}$. Three blocks $[n_2], [n_3], [n_4]$ at the bottom right, having significantly higher intensities, are three unknown but true active communities.

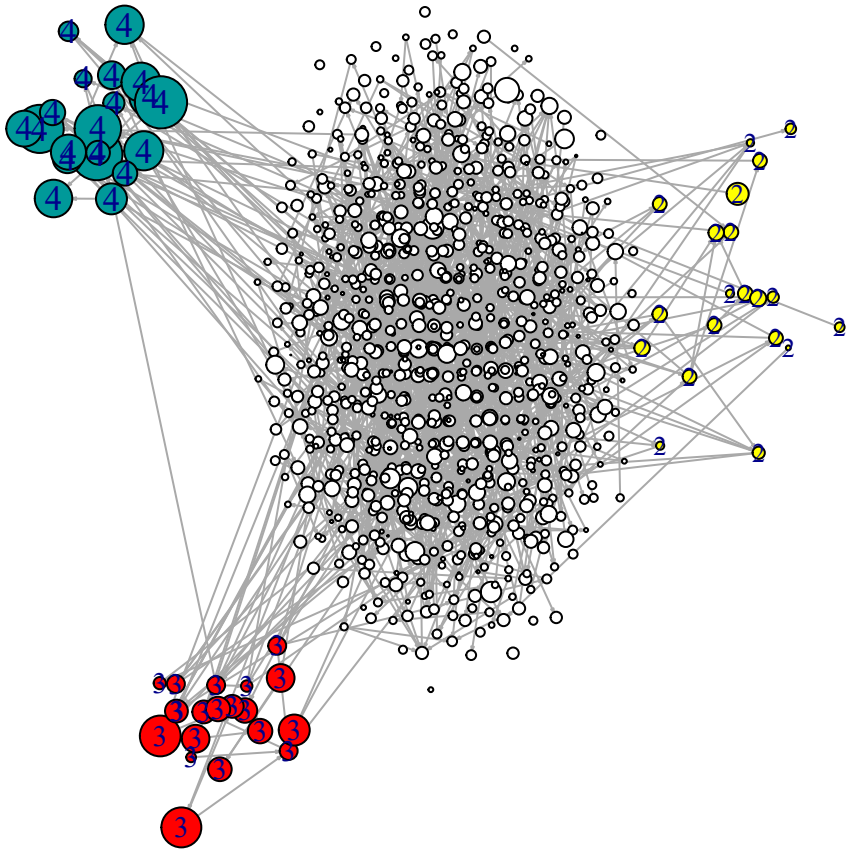


Figure 4.7: One sample graph G with $n = 1000, m = 10358$. One-tenth of uniformly sampled edges are incorporated in the figure. White (no label), yellow (label 2), red (label 3), and green (label 4) clusters represent blocks $[n_1], [n_2], [n_3]$, and $[n_4]$, respectively. Sizes of all vertices are proportional to locality statistic $\{\Psi_{k=1}(v)\}_{v=1}^{n}$.

The performance of separating built-in active vertices from inactive vertices by top Q locality statistic values in SBM graphs is evaluated as follows. Selection of Q vertices with the largest locality statistic values to form $\mathcal{C} = \{v_1, v_2, \ldots, v_Q\}$ can induce false alarms because it is likely that only a subset of $\mathcal{C}$ are built-in active community members in SBM random realizations. We can treat the Step (i) as a binary classification task, where $[n_2] \cup [n_3] \cup [n_4]$ are underlying positive labels and $[n_1]$ are negative ones, by using the Q-th largest locality statistic as a decision boundary. Next, the performance of the classifier is empirically evaluated through Receiver Operating Characteristic (ROC) curve and Area Under Curve (AUC). The empirical ROC curve is built through Monte Carlo simulations. Specifically, we repeatedly generate stochastic block model graphs and run Step (i) by varying Q from 1 to n for each graph. Accordingly, we calculate true and false positive rates according to true labels for each Q in each run.

Figure 4.8 shows the ROC mean curve of the classifiers with different k based on 4000 Monte Carlos. All three classifiers achieve AUC over 0.9 in this scenario, which demonstrates the usefulness of applying the Q-th largest locality statistic as a classifier boundary. It is also interesting to note that $\Psi_{k=1}(v)$ outperforms $\Psi_{k=0}(v), \Psi_{k=2}(v)$ in this moderate scale graph. In a graph at this scale, compared with $\Psi_{k=0}(v)$, $\Psi_{k=1}(v)$ aggregates more edges in a larger neighborhood to outclass itself from other vertices if v is in an active community. Compared with $\Psi_{k=2}(v)$, $\Psi_{k=1}(v)$ dominates in this experiment because $N_2(v)$ are more likely indistinguishable and highly overlapped between vertices from majority groups and active groups.

Next, after pinning down the Q most active vertices, we construct their similarity matrix $\mathcal{S}$ through the Jaccard Index in § 4.3, and perform a classic spectral clustering algorithm

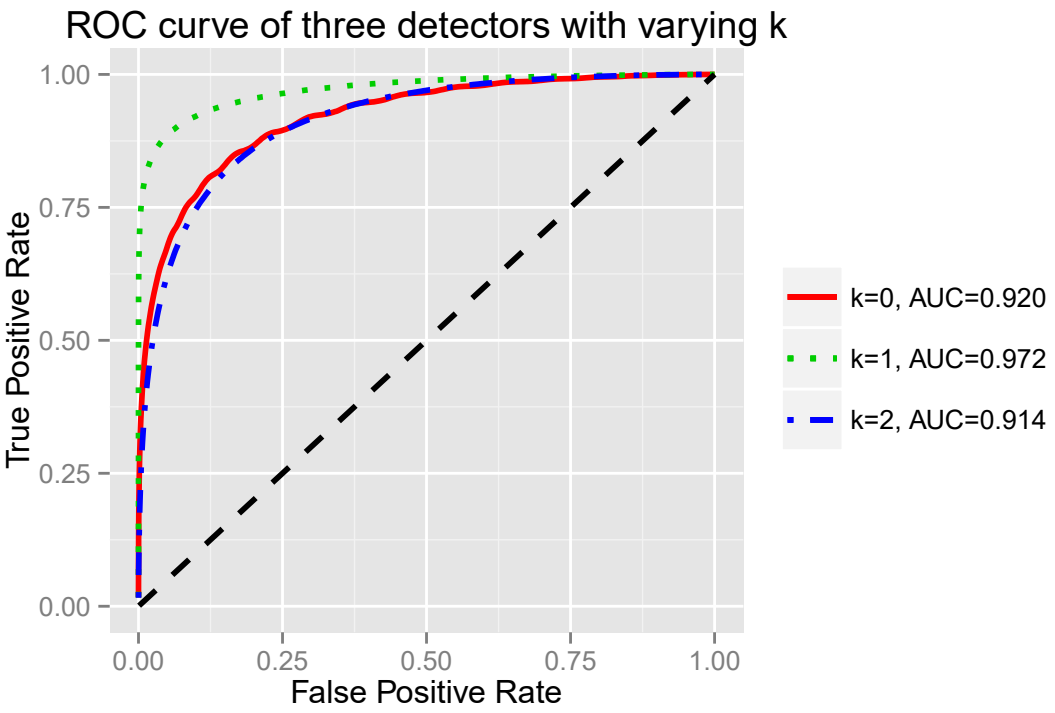


Figure 4.8: Receiver operating characteristic (ROC) mean curves and corresponding Area Under Curves (AUCs) of classifying active vertices using Q-th largest $\Psi_k(v)$ as decision boundary. The curve is built on 4,000 Monte Carlo simulations where each run generates an stochastic block model graph and calculate one discrete ROC curve by enlarging Q to increase false positive rate.

with Radial Basis Function (RBF) Kernel on $\mathcal{S}$ to cluster the Q vertices. This is a clustering task so that Adjusted Rand Index (ARI), recommended in [27] [55] [10], is an appropriate ad-hoc assessment of detection accuracy because the underlying cluster labels of the Q vertices are known.

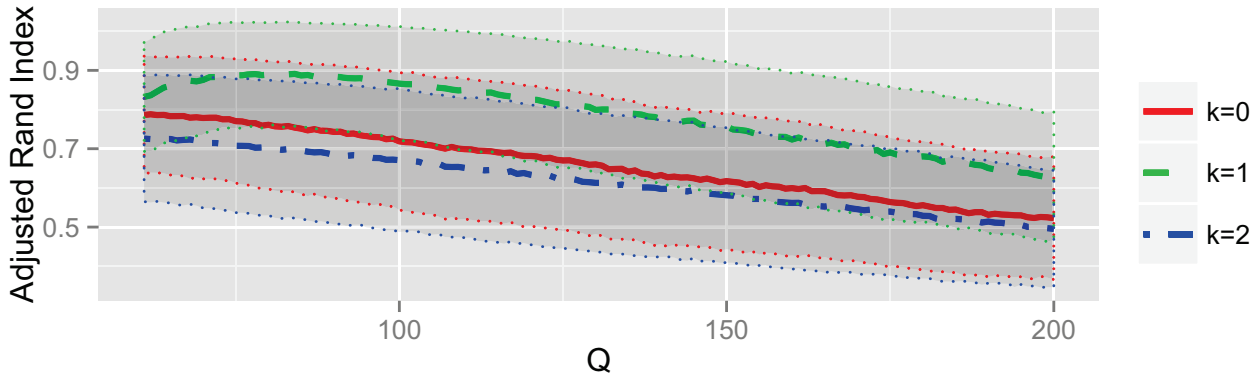


Figure 4.9: Adjusted Rand Index curves against Q, based on $4,000$ Monte Carlo simulations, between spectral clustering results and true clusterings of top Q vertices.

Figure 4.9 shows the ARI curves against Q, based on 4000 Monte Carlos, between our spectral clustering results and true clusterings of the top Q vertices. The horizontal axis starts from $Q = 61$ to guarantee that the top Q vertices precisely come from 4 distinct clusters $[n_1], [n_2], [n_3], [n_4]$. The bold curves are mean values and dot curves are mean curves plus (or minus) one standard deviation. It is clear that all mean ARI curves are still greater than 0.5 even when one-fifth of V are classified as active community members in step (i). In fact, if Q is well specified by a user, e.g., $Q < 75$, ARI values of clustering based on all three locality statistics are greater than 0.7. The results here suggest the satisfying accuracy of our detection framework.

4.6 Application

In this section, we evaluate our framework on the Hyperlink graph from August 2012 Common Crawl Corpus [28], the largest real-world graph dataset publicly available so far. The Hyperlink graph provides three different levels of aggregations on the graph. In this work, we use the Page-level version of the Hyperlink graph, where each vertex is a single web page, to verify the scalability of our detection framework. The Hyperlink graph is an unweighted and directed graph with $3,563,602,789$ vertices and $128,736,914,167$ edges. It is infeasible to perform any community detection algorithms with the complexity of $O(nm)$ or $O(n^2)$ on this graph. Furthermore, in the web graph society, a typical motivation of investigating community detection is to identify link farms which are deliberately created to increase search engine ranks [10]. With this motivation, observers are concerned only with communities consisting of active hyperlinks. These two constraints are the obstacles of deploying other algorithms but bypassed by our detection framework.

4.6.1 Active Communities of Hyperlink Graph

We run our detection framework on the Hyperlink graph to determine its effectiveness on the massive graph. In our experiment, we select $k = 1$ and run the trimming algorithm to identify the top Q vertices of the largest locality statistic values, where $Q = 2000$. In Step (ii) of the detection framework, Jaccard Index is selected to construct the similarity matrix $\mathcal{S}$ among the top 2000 vertices. Next, to cluster pinpointed websites into active communities, we use the same spectral clustering method with RBF kernel in §4.5. The number of clusters is suggested by the spectral gaps of S.

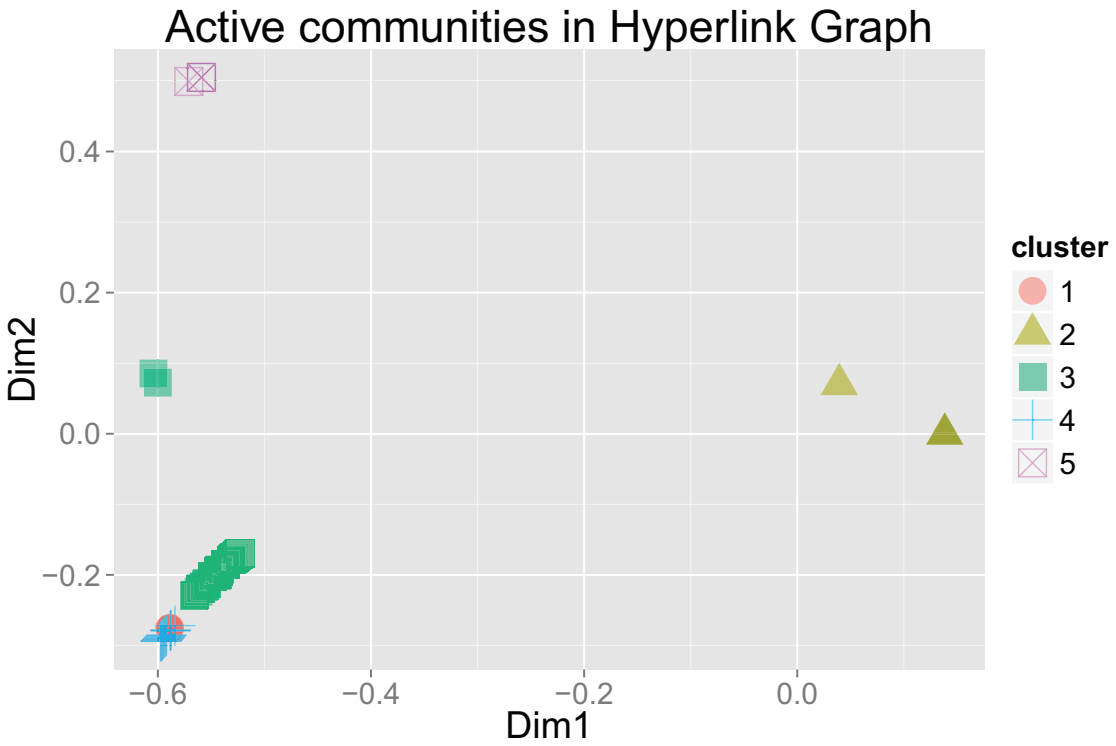


Figure 4.10: Five active communities in HyperLink graph. Top $Q = 2,000$ vertices projected into first two dimensions of classic multidimensional scaling of $\mathcal{S}$. 5 communities are colored separately where community index is consistent with Table 4.1. The sizes of Active community 1 to 5 are $n_1 = 35$, $n_2 = 1603$, $n_3 = 199$, $n_4 = 42$, and $n_5 = 121$, respectively.

Community	Selected URLs
1	http://www.families.com/ , http://www.eromance.com/ http://www.freecoupons.com/ , http://www.networkmedia.com/ http://www.younger.com/ , http://www.familytree.com/
2	all in this pattern: http://www.alphateenies.com/movies/*
3	http://wordpress.org/ , http://www.youtube.com/ http://www.google.com/ , http://www.flickr.com/ http://www.yahoo.com/ , http://www.facebook.com/ http://twitter.com/
4	http://www.amazon.com/ , http://www.zappos.com/ http://www.abebooks.com/ , http://www.myhabit.com/ http://www.woot.com/ , http://www.fabric.com/ http://www.diapers.com/
5	http://www.acidmovies.com/ , http://www.azimuthmovies.com/ http://www.asteroidmovies.com/ , http://www.croakmovies.com/ http://www.drymovies.com/ , http://www.btwmovies.com/ http://www.finishmovies.com/

Table 4.1: Table of selected URLs from active communities in Hyperlink Graph provided by our detection framework. URLs of similar topics are clustered in the same active communities. Community 1 are URLs maintained and developed by **networkmedia** company; Communities 2 and 5 are collections of adult websites; Community 3 consists of popular social media sites. Community 4 is composed of online shopping sites.

The procedure above detects five colored active communities decomposed from 2000 vertices (Figure 4.10 and Table 4.1). In Figure 4.10, the top 2000 vertices are projected into a two-dimensional space through classical multidimensional scaling (MDS) on the similarity matrix $\mathcal{S}$. Five active communities obtained from our detection framework are colored separately. The sizes of community 1 to 5 are $n_1 = 35$, $n_2 = 1603$, $n_3 = 199$, $n_4 = 42$ and $n_5 = 121$ respectively. Table 4.1 lists five selected web URLs from each cluster for further illustration of detected communities.

Out of 2000 vertices, there are 1603 vertices forming the community 2 whose members are all hyperlinks extracted from a single Pay-level-domain adult website (i.e., `http://www.alphateenies.com`). Community 1 is a collection of websites that are all developed, sold or to be sold by an Internet media company **networkmedia**, such as `http://www.families.com/`, `http://www.familytree.com/` and `http://www.freecoupons.com/`. Community 4 consists of websites related to online shopping such as the shopping giant Amazon and the bookseller AbeBooks. Community 5 is another collection of 121 adult web pages where each web page comes from a different Pay-level-domain in this cluster. In the community 3, most links are social media websites and often used in our daily life such as WordPress, Facebook, Twitter, Flicker and Google. In summary, top 5 active communities in the Hyperlink Graph are grouped with high topical similarities, which is consistent with findings in [10]. Therefore, these noteworthy clusters produced by our detection framework not only imply its applicability on a massive graph but also practicality on real World Wide Web graphs.

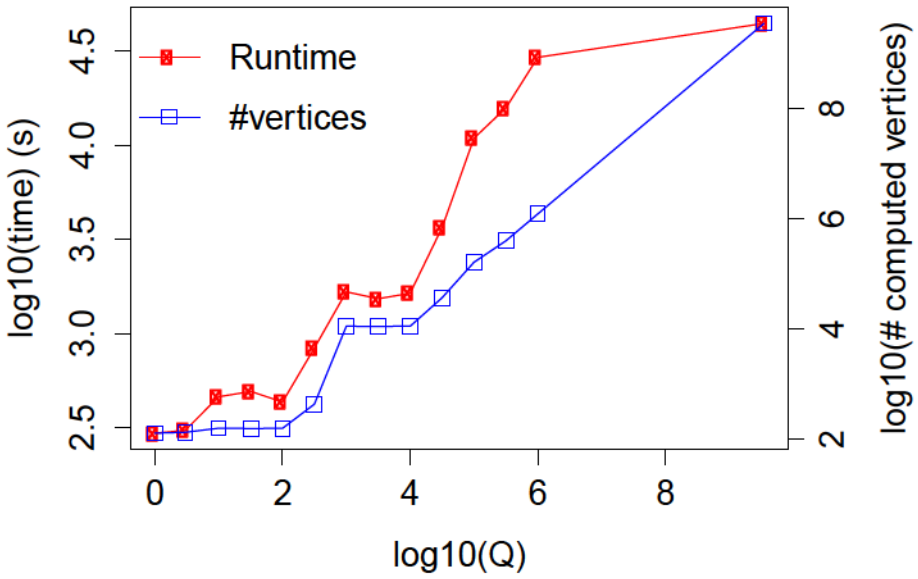


Figure 4.11: Log-log plot of time consumption and the number of locality statistic-computed vertices against Q of trimming algorithm. The log base is 10, and Q ranges from 1 to n. In the Hyperlink graph, the running time of trimming algorithm $T(Q) = O(\sqrt{Q})$ and computing top $Q = 10^4$ locality statistic values only takes 3.7% time consumption on all locality statistic values

4.6.2 Time-saving Trimming Algorithm

We evaluate the time saving achieved by the trimming algorithm (Section §4.4) on the massive Hyperlink graph. The computing environment of conducting the trimming experiment is a machine with four Intel Xeon E5-4620 processors, clocked at 2.2 GHz, and 512 GB memory of DDR3-1333. Each processor has eight cores with hyperthreading enabled, resulting in 16 logical cores. The machine has three LSI SAS 9207-8e host bus adapters (HBA) connected to a SuperMicro storage chassis, in which 12 OCZ Vertex 4 SSDs are installed. We conduct an experiment to show the relation of time consumption against Q and the number of locality statistic values computed against Q in trimming. The log-log plot with base 10 is given in Figure 4.11.

This experiment demonstrates that the trimming algorithm winnows active vertices efficiently even if Q is large. Figure 4.11 shows that the running time of computing top

Q locality statistic values on the Hyperlink graph is sub-linear against Q and could be upper bounded by $T(K) = O(\sqrt{K})$. For example, the ratio $T(Q = 10^4)/T(Q = n) = 0.03689694$ implies that computation on top $Q = 10^4$ vertices only takes 3.7% time of computation on all vertices because our algorithm only needs to compute locality statistic on 0.00032% vertices in the graph to find top 10^4 vertices.

Chapter 5

Conclusions and Discussion

This chapter concludes our current work and presents several directions that can be pursued in future related research.

5.1 Conclusion

This dissertation has presented methodologies of community detection using locality statistics in both temporal and static graph settings.

For temporal graphs, this work has summarized a generative latent position model for a time series of graphs and set up the anomalous community detection problem in a time series of graphs in terms of stochastic block models. We have proposed a method of handling with anomalous community detection through the use of scan statistics $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ constructed from two different locality statistics, Ψ and Φ, respectively. We derived the limiting properties for four representative instances of locality-based scan statistics $S_{1,0,0}(t;\Psi), S_{1,0,0}(t;\Phi), S_{1,0,1}(t;\Psi)$, and $S_{1,0,1}(t;\Phi)$. The limiting properties were then used to derive estimates for the power of the tests. The simulation experiments indicate that the analytic power estimates, even when they are limited in scope, are useful in answering some important questions about the locality statistics. In particular, it was shown that neither Ψ nor Φ dominates the other when $\tau = 1; \ell = 0; k = 0$, while Ψ is dominated by Φ when $\tau = 1; \ell = 0; k = 1$.

For a static massive graph, we propose a novel framework for detecting active communities that scales to a billion-node graph. Our framework consists of two parts: trimming of inactive vertices and clustering on selected active vertices. In the trimming step, we employ the locality statistic Ψ and present a parallelizable algorithm to distribute computation. In the clustering step, we use the spectral clustering approach, but other approaches are also applicable based on the context. The results on synthetic SBM graphs indicate that our framework performs well and yields reasonable active communities. A

general strength of our method is that, unlike most other approaches, it is scalable to extremely massive graphs. Its application to the hyperlink graph with billions of vertices discovers meaningful communities in the real World Wide Web graph dataset.

5.2 Future Work

Anomaly detection in temporal or massive graphs has applications in diverse areas, e.g., predicting the emergence of subgroups within an organization, monitoring disease spread in public networks, and detecting modules of cancer and metastasis communities in protein-protein interaction (PPI) networks. We envision that these and many other applications will benefit from the type of investigation outlined in this work. However, much remains to be done, both mathematically and computationally. We list here some future research avenues related to this work that have not been (sufficiently) explored so far.

5.2.1 Weighted Graphs

In our assumptions in Chapter 2 and 3, we focused only on unweighted graphs where the locality statistic $\Psi_k(v)$ (or $\Phi_k(v)$) counts the number of edges in the induced subgraph. For future work, if G is weighted, $\Psi_k(v)$ can be extended to the sum of the edge weights in the induced subgraph. Specifically, let us denote a time series of weighted digraphs by $\{G_t\}$, the edge from i to j by (i,j), and its corresponding weight by w_{ij}. The weight-included locality statistics $\Psi^w_{t;k}(v)$ corresponding to Eq. (2.3.1) ($\Psi^w_k(v)$ corresponding to Eq. (4.2.1)) and $\Phi^w_{t,t';k}(v)$ corresponding to Eq. (2.3.6) are defined as

$$\Psi^w_{t;k}(v) = \sum_{(i,j)} w_{i,j} \mathbf{1}_{\{(i,j)\in E(\Omega(N_k(v;G_t);G_t))\}} \tag{5.2.1}$$

$$\Phi^w_{t,t';k}(v) = \sum_{(i,j)} w_{i,j} \mathbf{1}_{\{(i,j)\in E(\Omega(N_k(v;G_t);G_{t'}))\}} \tag{5.2.2}$$

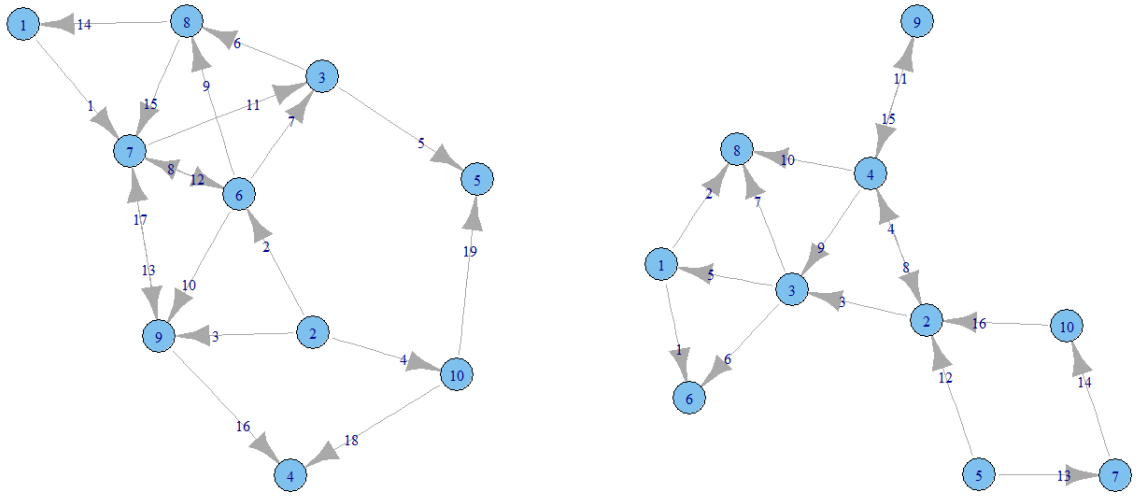


Figure 5.1: A two-step time series of weighted digraphs. left: G_1, right: G_2

.

where $N_k[v; G]$ remains as the set of vertices at a distance at most k from v by ignoring the direction of the edges. It is straightforward to see that the definitions in Eq. (2.3.1) and (2.3.6) are, respectively, special cases of Eq. (5.2.1) and Eq. (5.2.2) if $w_{i,j} = 1$ for any pair of (i, j).

With the aid of Figure 5.1, which shows a two-step time series of graphs $\{G_t\}_{t=1}^{2}$, we can illustrate the above definitions with a simple example. For instance, let $v = 3$; then

$\Psi^{w}_{t=2;k=1}(v=3)$ is computed as shown below.

$$
\begin{aligned}
&\Psi^{w}_{t=2;k=1}(v=3)\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in E(\Omega(N_k(v;G_{t=2});G_{t=2}))\}}\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in E(\Omega(\{1,2,3,4,6,8\};G_2))\}}\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in\{(1,6),(1,8),(3,1),(3,6),(3,8),(4,3),(2,3),(4,8),(2,4),(4,2)\}\}}\\
=&1+2+5+6+7+9+3+10+4+8=55;
\end{aligned}
$$

Similarly, if $v=6$, then $\Phi^{w}_{t=2,t'=1;k=2}(v=6)$ is computed as shown below.

$$
\begin{aligned}
&\Phi^{w}_{t=2,t'=1;k=2}(v=6)\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in E(\Omega(N_2(v=6;G_2);G_1))\}}\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in E(\Omega(\{1,2,3,4,6,8\};G_1))\}}\\
=&\sum_{(i,j)} w_{i,j}\mathbf{1}_{\{(i,j)\in\{(8,1),(2,6),(3,8),(6,3),(6,8)\}\}}\\
=&14+2+6+7+9=38
\end{aligned}
$$

Using new definitions in Eq. (5.2.1) and Eq. (5.2.2) to respectively replace Eq. (2.3.1)(Eq. (4.2.1) in the static massive graph setting) and Eq. (2.3.6), we can still conduct our proposed community detection methods on weighted graphs. Note that all other procedures introduced in Chapters 2 and 3 remain unchanged when they are applied to weighted graphs. However, theoretical investigations, such as limiting properties of scan statistics when the underlying locality statistic Ψ or Φ is adapted to include edge weights, are currently unavailable.

5.2.2 Streaming Graphs

The community detection on streaming graphs is in its infancy because even the optimal solution of clustering on a single static graph remains controversial. To identify how a dynamic community emerges, evolves, or dies, researchers often choose to represent time-stamped graph data in the form of a time series of graphs. This is also the environment in which our scan statistics perform detections. One drawback of the algorithm is that it is limited to a time series of static graphs, and is not suitable for streaming graphs, especially when the underlying locality statistic is Φ. If $\tau = 1, \ell = 0$ and $k = 1$, it is worth noting that Ψ, compared with Φ, is inadmissible but computationally inexpensive. For instance, to complete the τ-step vertex-dependent normalization calculation presented in Eq.(2.3.3), we have to record previous τ-step graphs to calculate $J_{t,t';k}(v)$ if the underlying locality statistic is Φ. This is certainly formidable and undesired in a streaming graph algorithm. However, if the underlying locality statistic is Ψ, graph storage is not necessary, and recording only the previous τ-step statistics $\Psi_{t';k}(v)$ is sufficient to calculate $J_{t,t';k}(v)$. Locality statistics based on Ψ can be readily computed in a real-time streaming data environment, in contrast to those based on Φ. Thus, the adaption or approximation of locality statistics based on Φ for streaming environments is of interest.

Figure 5.2.2 presents a way to maintain $\{\Psi_{t,k=1}(w)\}_{w=1}^{n}$ in a real-time streaming data environment. In a data stream at time stamp t, we assume that there is only one edge (u_t, v_t) that is either inserted or deleted in G_t. The procedures of update $\{\Psi_{t+1,k=1}(w)\}_{w=1}^{n}$ are given below, and the time cost of updating $\Psi_{t,k=1}(u_t)$ and $\Psi_{t,k=1}(v_t)$ is $O(|N_1[u_t; G_t] \cap N_1[v_t; G_t]|)$.

Require: An initial graph G_0, a stream of inserted or deleted edges $\{(u_t, v_t)\}_{t=1}^{\infty}$ ▷

Return updated locality statistics $\{\Psi_{t+1,k=1}(w)\}_{w=1}^{n}$

Initialization: record $\{N_1[v]\}_{v=1}^{n}$, compute $\{\Psi_{0,k=1}(v)\}_{v=1}^{n}$

while (u_t, v_t) is inserted or deleted **do**

  $S \leftarrow N_1[u_t] \cap N_1[v_t]$

  $\delta = \mathbf{1}_{\{(u_t,v_t) \text{ is inserted}\}} - \mathbf{1}_{\{(u_t,v_t) \text{ is deleted}\}}$

  if (u_t, v_t) inserted **then**

    $N_1[u_t] \leftarrow N_1[u_t] \cup \{v_t\}$; $N_1[v_t] \leftarrow N_1[v_t] \cup \{u_t\}$;

  if (u_t, v_t) deleted **then**

    $N_1[u_t] \leftarrow N_1[u_t] - \{v_t\}$; $N_1[v_t] \leftarrow N_1[v_t] - \{u_t\}$;

  if $|S| = 0$ **then**

    $\Psi_{t+1,k=1}(w) \leftarrow \Psi_{t,k=1}(w) + \delta \quad w \in \{u_t, v_t\}$;

  else

    $\Psi_{t+1,k=1}(w) \leftarrow \Psi_{t,k=1}(w) + \delta(1 + |S|) \quad w \in \{u_t, v_t\}$;

    $\Psi_{t+1,k=1}(w) \leftarrow \Psi_{t,k=1}(w) + \delta \quad w \in S$;

  $\Psi_{t+1,k=1}(w) \leftarrow \Psi_{t,k=1}(w) \quad w \in [n] - \{u_t, v_t\} - S$;

Figure 5.2: Fast update rules for $\{\Psi_{t,k=1}(w)\}_{w=1}^{n}$ in a data stream of edge insertions and deletions.

5.2.3 Beyond Stochastic Block Model Graphs

The investigations presented in this work do not take into account attributes of the edges. The incorporation of edge attributes into the current work is, however, straightforward. For example, [45] handles attributes by linear fusion, and many of the results there can be adapted. In particular, one can define fused locality statistics for attributed graphs. For active community detection in massive graphs, we can find active vertices by concentrating on vertices with large values of fused locality statistics. In the process of constructing a similarity matrix, similarities between active vertices can be measured based on the attributes of vertices instead of their structural connectivities. For anomalous community detection in temporal graphs, power estimates for these fused locality statistics can be derived in a manner similar to those presented in Chapter 2. Other considerations, e.g., optimal fusion parameters, can also be investigated. However, the statistics considered in [45] are only temporally normalized and do not contain a vertex-dependent normalization. Thus, the derivation of their limiting properties is much less involved. In addition, as the experimental results in Figure 2.5 show, the vertex-dependent normalization does lead to improved statistical power in many situations of interest.

Furthermore, the power estimates in Chapter 2 are also useful for reasoning about the behavior of more complicated models without the $\{G_t\}$ independency assumption, such as the latent process model proposed in [24]. In [24], a latent process model was developed for a time series of attributed graphs based on a random dot process model. Having n vertices governed by n individual continuous-time finite-state stochastic processes, this model generates a time series of dependent attributed random graphs, or, equivalently, conditioning on the sample paths of the stochastic processes, the graphs are

independent. This source also provides two approximations to the exact latent process model. The first-order approximation is the stochastic block model that gives rise to a time series of independent random graphs with independent edges. The second-order approximation corresponds to the random dot product model that gives rise to a time series of independent random dot product graphs. Both of these approximations are presented in § 2.1.

5.2.4 Parameter Selection

Another research direction is to study an optimal combination of input parameters for our proposed algorithms and understand the interplay between them. Ideally, we would like to determine the best approach to select a combination of (k, τ, ℓ) in Chapter 2 and a combination of $(k, Q, \text{similarity measure}, \text{clustering method})$ in Chapter 3.

In inference of the time series of graphs, we hope that the following experiment will help to motivate subsequent work in understanding the interplay between locality statistics, vertex and temporal normalizations, and power estimates. In § 2.4.1 and § 2.4.2, for simplicity in analytic investigations, we theoretically obtain power estimates of $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ under the restrictions of $\tau = 1$ and $\ell = 0$. In addition to analytic investigations, we empirically study the power performances of $S_{\tau,\ell,k}(t;\Psi)$ and $S_{\tau,\ell,k}(t;\Phi)$ with other (τ, ℓ) combinations via Monte Carlo simulations. In this experiment, we let τ range from 0 to 10 and ℓ range from 0 to 10. In each Monte Carlo replicate, a time series of random graphs based on the SBM is considered in §2.4.1, where $(n_1, n_2, n_3) = (870, 65, 65), (p, h, q) = (0.43, 0.95, 0.98)$, is sampled. Next, $S_{\tau,\ell,k}(t^* - 1;\Psi)$,$S_{\tau,\ell,k}(t^* -$

	$\max_{(\tau,\ell)} \beta$	(τ^*, ℓ^*)
$S_{\tau,\ell,0}(t;\Psi)$	0.483	$(1,0)$
$S_{\tau,\ell,0}(t;\Phi)$	0.384	$(1,10)$
$S_{\tau,\ell,1}(t;\Psi)$	0.571	(1,10)
$S_{\tau,\ell,1}(t;\Phi)$	0.758	(1,9)

Table 5.1: The optimal τ and ℓ in an experiment comparing the statistical power of $S_{\tau,\ell,k}$ for $k = 0, 1$ and locality statistics Φ and Ψ. We vary $\tau, \ell \in \{0, 1, \ldots, 10\}$ and compare the statistical power for each choice of τ and ℓ through a Monte Carlo experiment with $2,000$ replicates.

$1;\Phi)$,$S_{\tau,\ell,k}(t^*;\Psi)$ and $S_{\tau,\ell,k}(t^*;\Phi)$ are calculated individually according to specific (τ,ℓ,k). After $2,000$ replicates, for each test statistic, the largest empirical power (denoted by $\max_{(\tau,\ell)} \beta$) and the corresponding optimal choice of (τ,ℓ)(denoted by (τ^*,ℓ^*)) is obtained and summarized in Table 5.2.4.

The empirical results in Table 5.2.4 demonstrate the potential value of extending the theoretical investigations in §2.4.1 and §2.4.2 to cases of $\tau \geq 1$ and $\ell \geq 1$, although this extension appears significantly more challenging than the case $(\tau,\ell) = (1,0)$. In [45], $S_{\tau,\ell,k}(t;\Psi)$ was also investigated for cases of $\tau = 0$, $\ell \to \infty$, and $k \leq 1$ under the SBM setting. However, power estimates for other, more complex locality-based scan statistics, such as $S_{\tau,\ell,k}(t;\cdot)$ for $1 < \tau < \infty$, $0 < \ell < \infty$ and $k \geq 2$, remain to be investigated.

In active community detection in a massive graph, we notice that $\Psi_{k=1}(v)$ outperforms $\Psi_{k=0}(v), \Psi_{k=2}(v)$ in a moderate-scale graph in Figure 4.8. In fact, which detector obtains the dominating performance depends on the graph scale and structure. For instance, $\Psi_{k=0}(v)$ provides even better performance on smaller networks because it is more likely

that $N_1[v]$, let alone $N_2[v]$, are indistinguishable and highly overlapped between vertices from the majority group and the anomalous group. By the same argument, $\Psi_{k=2}(v)$ performs the best on a larger-scale network if v is in an active community since it aggregates a greater number of edges in a larger neighborhood size to outclass itself from other vertices. Hence, it is reasonable to study the optimal choice of k, as varying k may yield different trimming results. Moreover, we should also explore the trade-off between heavier trimming computational burden and trimming performance. Finally, although our current experiment uses a combination of the Jaccard index and spectral clustering to perform clustering, it might be interesting to determine whether an alternative combination dominates our current approach.

Bibliography

[1] Ery Arias-Castro and Nicolas Verzelen. Community detection in random networks. *arXiv preprint arXiv:1302.7099*, 2013.

[2] James P. Bagrow and Erik M. Bollt. Local method for detecting communities. *Physical Review E*, 72(4):046108, 2005.

[3] Simeon M. Berman. Limit theorems for the maximum term in stationary sequences. *Annals of Mathematical Statistics*, 35(2):502–516, 1964.

[4] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):10008, 2008.

[5] Daniela Calvetti, L Reichel, and Danny Chris Sorensen. An implicitly restarted lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, 2(1):21, 1994.

[6] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[7] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[8] Luca Donetti and Miguel A. Muñoz. Improved spectral algorithm for the detection of network communities. *arXiv preprint physics/0504059*, 2005.

[9] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the national academy of sciences*, 99(9):5825–5829, 2002.

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

[11] Fusion-IO ioDrive Octal. `http://www.fusionio.com/platforms/iodrive-octal/`, Accessed 4/18/2014.

[12] Janos Galambos. *The Asymptotic Theory of Extreme Order Statistics*. John Wiley & Sons, 1987.

[13] James P. Galasyn. Enron chronology. `http://www.desdemonadespair.net/2010/09/bushenron-chronology.html`, Accessed 12/2013.

[14] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[15] Joseph Glaz, Joseph Naus, and Sylvan Wallenstein. *Scan Statistics*. Springer, 2001.

[16] Matthew B. Hastings. Community detection as an inference problem. *Phys. Rev. E*, 74:035102, 2006.

[17] Nicholas A. Heard, David J. Weston, Kiriaki Platanioti, and David J. Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645–662, 2010.

[18] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[19] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.

[20] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[21] Brian W. Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[22] Eric D. Kolaczyk. *Statistical analysis of network data: methods and models*. Springer Science & Business Media, 2009.

[23] M. Kulldorff. A spatial scan statistic. *Communications in Statistics- Theory and Methods*, 26(6):1481–1496, 1997.

[24] Nam H. Lee and Carey E. Priebe. A latent process model for time series of attributed random graphs. *Statistical Inference for Stochastic Processes*, 14:231–253, 2011.

[25] Nam H. Lee, Jordan Yoder, Minh Tang, and Carey E. Priebe. On latent position

inference from doubly stochastic messaging activities. *Multiscale Modeling and Simulation*, 11(3):683–718, 2013.

[26] E. A. Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2), 2006.

[27] Marina Meilă. Comparing clusteringsan information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[28] Robert Meusel, Oliver Lehmberg, Christian Bizer, and Sebastiano Vigna. Web Data Commons - Hyperlink Graph. `http://webdatacommons.org/hyperlinkgraph/`.

[29] Benjamin Miller, Nadya Bliss, and Patrick J. Wolfe. Subgraph detection using eigenvector L1 norms. In *Neural Information Processing Systems Foundation*, pages 1633–1641, 2010.

[30] Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Ambuj K. Singh, Evangelos E. Papalexakis, and Christos Faloutsos. Netspot: Spotting significant anomalous regions on dynamic networks. In *SIAM International Conference on Data Mining*, pages 727–738, May 2013.

[31] Joshua C. Neil. *Scan Statistics for the Online Discovery of Locally Anomalous Subgraphs*. PhD thesis, The University of New Mexico, May 2011.

[32] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[33] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[34] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:845–856, 2002.

[35] Youngser Park, Carey E. Priebe, and Abdou Youssef. Anomaly detection in time series of graphs using fusion of graph invariants. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):67–75, Feb 2013.

[36] Youngser Park, Heng Wang, Töbias Nobauer, Alipasha Vaziri, and Carey E. Priebe. Anomaly detection on whole-brain functional imaging of neuronal activity using graph scan statistics. In *21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Workshop on Outlier Detection, and Description*, 2015.

[37] Robert Prevedel, Young-Gyu Yoon, Maximilian Hoffmann, Nikita Pak, Gordon Wetzstein, Saul Kato, Tina Schrödel, Ramesh Raskar, Manuel Zimmer, Edward S Boyden, et al. Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. *Nature methods*, 11(7):727–730, 2014.

[38] Carey E. Priebe. Scan statistics on graphs. Technical Report 650, Johns Hopkins University, 2004.

[39] Carey E. Priebe, John M. Conroy, David J. Marchette, and Youngser Park. Scan statistics on Enron graphs. *Computational and Mathematical Organization Theory*, 11:229–247, 2005.

[40] Carey E. Priebe, Youngser Park, David J. Marchette, John M. Conroy, John Grothendieck, and Allen L. Gorin. Statistical inference on attributed random graphs: Fusion of graph features and content: An experiment on time series of enron graphs. *Computational Statistics and Data Analysis*, 54:1766–1776, 2010.

[41] Erzsébet Ravasz, Anna Lisa Somera, Dale A. Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

[42] Andrey Rukhin and Carey E. Priebe. On the limiting distribution of a graph scan statistic. *Communications in statistics-Theory and Methods*, 41(7):1151–1170, 2012.

[43] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[44] Thomas Schank and Dorothea Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. In *Experimental and Efficient Algorithms*, pages 606–609. Springer, 2005.

[45] Minh Tang, Youngser Park, Nam H. Lee, and Carey E. Priebe. Attribute fusion in a latent process model for time series of graphs. *IEEE Transactions on Signal Processing*, 61(7):1721–1732, April 2013.

[46] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks:[extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276. ACM, 2007.

[47] Xiaomeng Wan, Jeannette Janssen, Nauzer Kalyaniwalla, and Evangelos Milios. Statistical analysis of dynamic graphs. In *Proceedings of AISB06: Adapation in Artificial and Biological Systems*, pages 176–179, 2006.

[48] Xiaomeng Wan and Nauzer Kalyaniwalla. Capturing causality in communications graphs. In *DIMACS/DyDAn workshop on computational methods for dynamics interaction*, 2007.

[49] Bei Wang, Jeff M. Phillips, Robert Schreiber, Dennis Wilkinson, Nina Mishra, and Robert Tarjan. Spatial scan statistics for graph clustering. In *SIAM International Conference on Data Mining*, pages 727–738, 2008.

[50] Heng Wang, Minh Tang, Youngser Park, and Carey E. Priebe. Locality statistics for anomaly detection in time series of graphs. *IEEE Transactions on Signal Processing*, 62(3):703–717, 2014.

[51] Heng Wang, Da Zheng, Randal Burns, and Carey E. Priebe. Active community detection in massive graphs. In *SIAM International Conference on Data Mining: The second workshop on Mining Networks and Graphs: A Big Data Analytic Challenge*, Vancouver, Canada, 2015.

[52] Yuchung J. Wang and George Y. Wong. Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.

[53] Bo Yang and Jiming Liu. Discovering global network communities based on local centralities. *ACM Transactions on the Web (TWEB)*, 2(1):9, 2008.

[54] Stephen J. Young and Edward R. Scheinerman. Random dot product models for

social networks. In *Proceedings of the 5th international conference on algorithms and models for the web-graph*, pages 138–149, 2007.

[55] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

[56] Da Zheng, Randal Burns, and Alexander S. Szalay. Toward millions of file system IOPS on low-cost, commodity hardware. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2013.

[57] Da Zheng, Disa Mhembere, Randal Burns, Joshua Vogelstein, Carey E. Priebe, and Alexander S. Szalay. FlashGraph: Processing billion-node graphs on an array of commodity SSDs. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, pages 45–58, Santa Clara, CA, 2015.

Vita

Heng Wang was born on January 7, 1988 in Wuhan, P.R.China. He received the B. S. degree of Mathematics from Nanjing University and enrolled in the Applied Mathematics and Statistics Ph.D. program at Johns Hopkins University in 2010. During his study at Johns Hopkins University, he obtained M.S.E of Applied Mathematics and Statistics in 2012 and M.S.E of Computer Science in 2014. His research interests include anomaly detection in time series of graphs, community detection on large-scale graphs and network mining.