Noname manuscript No.
(will be inserted by the editor)

A Multi-Level Anomaly Detection Algorithm for Time-Varying Graph Data with Interactive Visualization

Robert A. Bridges · John Collins · Erik M. Ferragut · Jason Laska · Blair D. Sullivan

Received: December 18, 2015 / Accepted: date

Abstract This work presents a modeling and analysis framework for graph sequences which addresses the challenge of detecting and contextualizing anomalies in streaming graph data. Our goal is to detect changes at multiple levels of granularity, thereby identifying specific nodes and subgraphs causing a graph to appear anomalously. In particular, the framework detects changes in community membership, density, and node degree in a sequence of graphs where these are relatively stable. In route to this end, we introduce a new graph model, a generalization of the BTER model of Seshadhri et al., by adding

This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (`http://energy.gov/downloads/doe-public-access-plan`).
Blair D. Sullivan supported in part by DARPA GRAPHS/SPAWAR Grant N66001-14-1-4063, the Gordon & Betty Moore Foundation, and the National Consortium for Data Science. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of DOE, DARPA, SSC Pacific, the Moore Foundation, or the NCDS.

Robert A. Bridges (lead author), John Collins, Erik M. Ferragut, Jason Laska
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 32831
Tel.: 1-865-241-0319
E-mail: {bridgesra, ferragutem, laskaja}@ornl.gov, jparcoll@gmail.com

Blair D. Sullivan
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
E-mail: blair_sullivan@ncsu.edu

flexibility to community structure, and use this model to perform multi-scale graph anomaly detection. This technique provides insight into a graph's structure and internal context that may shed light on a detected event. Additionally, this multi-scale analysis facilitates intuitive visualizations by allowing users to narrow focus from an anomalous graph to particular subgraphs or nodes causing the anomaly. For evaluation, two hierarchical anomaly detectors are tested against a baseline Gaussian method on a series of sampled graphs. We demonstrate that our graph statistics-based approach outperforms both a distribution-based detector and the baseline in a labeled setting with community structure, and it accurately detects anomalies in synthetic and real-world datasets at the node, subgraph, and graph levels. To illustrate the accessibility of information made possible via this technique, the anomaly detector and an associated interactive visualization tool are tested on NCAA football data, where teams and conferences that moved within the league are identified with perfect recall, and precision greater than 0.786.

Keywords Anomaly Detection · Graph Sequence · Visualization

1 Introduction

Social networks play an increasingly important role in today's society, yet extracting domain insights from their analysis and visualization remains challenging—in large part due to their transient nature and the inherent complexity of many graph algorithms. Many social graphs naturally have (i) labeled nodes representing individuals or entities, and (ii) an edge set that changes over time, creating a sequence or time-series of individual snapshots of the network. A key task in understanding this data is the ability to identify patterns and aberrations across snapshots—specifically in a way that can pinpoint areas of interest, and provide context for results. The importance of context in anomaly detection is easily exemplified in a cyber-security setting, where observing an unanticipated connection (edge) between an internal IP and an external host might warrant alarm. However, providing the context that many similar IPs (i.e. nodes in a common community) regularly contact that host could save an unnecessary investigation. Although this time-varying labeled scenario is natural in many domains, most existing techniques for anomaly detection are either limited to static graphs or unable to "zoom in" on the reason a graph is identified as non-standard (see Section 7.2).

Here we address the problem of identifying and contextualizing anomalies at multiple levels of granularity in the sequential graph setting. This problem is neither a special case nor an extension of the more commonly studied scenario of finding anomalous parts of a single (static) graph, since the availability of common node labels provides information not available in single-graph or unlabeled graph ensemble problems, necessitating the development of new methods. We build on our preliminary work (Bridges et al. [3]) in which we propose and test a novel method for anomaly detection in time-varying graph data using hierarchically related distributions to detect related abnormalities

at three increasingly fine levels of granularity (i.e., at graph, subgraph, and node levels). We assume that we have observations of a sequence of graphs, and seek to model and detect anomalies in community density, community membership and node degrees. Consequently, there is an implicit assumption that we can identify nodes across graphs, that the communities (i.e., relatively dense subgraphs) will have relatively persistent members and density across graphs. Our approach is a multi-scale technique that gives insight into exactly what caused the anomaly, and allows one to focus attention on the specific subgraphs involved. This extended version provides new testing on real data, discusses and exhibits preliminary visualizations which are informed by the detection setup, and adds depth to previous discussions.

The probabilistic multi-scale detection relies on comparison with an underlying graph model, and we use an extension (described in Section 3) of the recent BTER model [20] that enables improved prescription of community structure. To fit an instance of the model to observed graphs, we give methods for detecting communities and estimating parameters (see Section 4). Section 5 defines the probability calculations for two new multi-scale detectors, as well as a baseline detector similar to that of [17] (which is limited to detecting anomalies at the graph-level). Finally, to test a newly observed graph for anomalous structure, we compute hierarchically-related probabilities from the tuned model and their associated p-values using a Monte-Carlo simulation. Our workflow is a streaming detection framework, where parameters are learned from previous observations, the detector is applied to new data, then the parameters are updated to include the new graph in the observations.

We note that performing anomaly detection using a graph's probability—as given by the model from which it was sampled—will often result in an inaccurate detector when node labels are known. This is a consequence of the likelihood of an unlabeled graph being shared by isomorphic copies distinguished by these labels and is discussed in Section 5. We illustrate this phenomenon and provide empirical evidence that modeling a set of statistics indicative of node/subgraph interactions provides more accurate detection in two experiments described in Section 6. In Section 7, we apply our detector to NCAA Football data, establishing its accuracy in detecting variations in teams' schedules as imposed by changes in conference membership. This application naturally allows the detector to find abnormal interactions at the node (team), community (conference), and full graph (season) levels. Finally, we describe and show sample screenshots of applying our interactive anomaly visualization tool, which leverages the multi-scale analysis to enable users to easily focus their attention on the most critical changes in the data.

In summary, this work gives generalizes a known graph model, the BTER model to more accurately model communities. This is a stepping stone to the main goal, which is an anomaly detection capability for sequential graph data. By exploiting knowledge of node labels, this technique identifies multiple, related abnormalities each at increasing levels of granularity to facilitate greater understanding and more meaningful visualizations.

2 Related Work

In this paper, we focus on identifying anomalous instances in a sequence of graphs with common node labels. We note this problem is distinct from finding anomalies in a static graph; specifically, to characterize normal behavior, we leverage observations of nodes and communities from previous graphs in the sequence—information not available in a single graph or unlabeled graph ensemble problems—and new methods are required to fully exploit this information. There is limited work transforming a graph sequence to a single instance; e.g., Eberle et al. consider the disjoint union of subgraphs from each data instance as a single graph in [6]. For a survey on graph anomaly detection, we refer the readers to Akoglu et al. [1].

Common techniques for finding anomalies in graphs can broadly be categorized as using compression techniques or a form of hypothesis testing with graph statistics. For example, Eberle & Holder use a compression algorithm relying on minimum description length to detect repetitive subgraphs and identify slight deviations as anomalies [6]. Because this technique searches for subgraphs almost isomorphic to a found normative pattern, it is a much more rigid detection framework than ours. Hypothesis testing has a broader set of prior work, including papers of Miller et al. which use statistics based on the residual matrix [15,16]. Much of this work is geared towards detecting abnormally dense communities seeded into an R-MAT graph. In [15], the techniques are extended to accommodate the dynamic graph setting and include methods for identifying highly connected regions. Our detectors are designed to identify anomalies caused not only by abnormal density, but changes in the interactions within or between communities. A more recent hypothesis testing approach of Neville & Moreno fits Gaussian distributions to three statistics [17]. Due to similarities with our workflow (using a p-value estimated by a Monte-Carlo simulation from a graph model to decide anomalies), we test our method against a baseline detector using similar Gaussian estimates, although we note that [17] focused on Kronecker graphs, not the GBTER model that we use in this work and introduce in Section 3.Written concurrently with this work is that of Peel & Clauset [18] which addresses the problem of change detection for time-varying network sequences. Like Peel & Clauset, we use a hierarchical generative graph model and Bayesian hypothesis testing. Our work differs in that it introduces a new graph model (Section 3) and seeks related anomalies at different scales (as opposed to a "shock" that changes the overall graph structure).

To the authors' knowledge, using multiple related detectors that respect the structure of the graph is a new technique, introduced in the prelimary version of this paper, [3]. This has lead to work of Harshaw et al. [11], which borrowed the multiscale anomaly detection idea, but uses graphlet and automorphism orbit counts in sequential graph data to detect anomalies and multiple scales. The difference in techniques follows the difference in goals between the two works; specifically, our work seeks to model and detect anomalies in commu-

nity membership, community density, and node degrees, while the method of Harshaw et al. seeks changes in the graph's local topology over time.

By design, the multi-scale analysis informs an interactive tool for exploring the nature of abnormalities in each graph, a relatively unstudied aspect of graph visualization. Wong et al. [22] present a multi-scale tool for exploring large graphs, informed by a clustering algorithm especially tuned to detecting star-burst patterns. In contrast, we detect dense regions as communities and integrate multi-scale visual-analytics with anomaly scores.

A common question for any anomaly detection system is "How to choose an appropriate threshold for alerts?". Often α is tuned in light of labeled results to find an acceptable balance of false vs. true positives. Given a particular application and the presence of labeled data, such a technique would give insight into setting the threshold for the proposed graph anomaly detection method. Mathematical developments in Ferragut et al. [8] provide a theorem allowing users to regulate the expected alert rate à priori by setting α. The theorem assumes low p-value events are defined as anomalous. Hence it depends on the probabilistic framework for anomaly detection, but the theorem holds independent of the distribution used. This is good news for streaming anomaly detection, in which the probability model changes upon receipt of new data. In particular, operators can choose a threshold to regulate the alert rate a priori. For our work this gives an additional method for setting the alert threshold—operators can specify an expected number of alerts per time unit (day, month, etc.), and the theorem of Ferragut et al. gives a method for setting the threshold.

3 The Generalized BTER Model (GBTER)

In order to perform probabilistic anomaly detection, we need a randomized generative graph model that enables computation of probabilities for various graph configurations while accurately modeling a graph's community structure and degree sequence. Significant prior work has been devoted to developing such models and validating the importance of capturing both of these aspects of a real-world data set (e.g., [2, 4, 5, 13, 20]). A broad survey of graph models and common graph characteristics is given by Chakrabarti et al. [4].

More specifically, motivated by social and cyber settings, we require a generative model that can accommodate observed hierarchical structure; more specifically, we seek to model node degree, community density, and community membership. A natural candidate is a Stochastic Block Model, first introduced in [12], which defines community membership and generates intra-community edges with an Erdös-Rényi (ER) [7] model and inter-community edges with a probability that depends on the membership of their endpoints. This achieves flexible community membership and density, but the expected degree of each node is implicitly determined by the community structure and parameters—if nodes i and j reside in the same community, they must have the same expected degree.

To improve adherence to degree distribution, one could use the Block Two-Level Erdös-Rényi (BTER) of Seshadhri et al. [13,20]. This model allows prescription of the expected degree distribution and expected clustering coefficients of each degree, but has a consequence that community structure is implicitly determined. (Further explanation of clustering coefficients are provided in the Appendix 9.2.) Inputs to the model are the (1) number of nodes of each degree and (2) the clustering coefficient of each degree. Nodes are then implicitly grouped into communities by putting degree d nodes into groups of size $d+1$, and assigning the density of these degree-d communities to be $c_d^{1/3}$, here c_d is the given clustering coefficient of degree d nodes. Next, BTER edge generation occurs in two steps, with an ER model[1] used for intra-community edges, followed by a Chung-Lu (CL) process [5] to match a specified expected degree distribution. While the BTER model has been designed and shown to model both degree distribution and clustering coefficients of each degree, the inflexibility of community structure is very limiting; for example, communities are necessarily comprised of nodes with the same degree (barring a few low frequency exceptions), hence communities with nodes of different degree are in general not possible. Furthermore, communities with different densities but similar degree members are also not possible. These restrictions follow from the desire to model the clustering coefficient of each degree.

To address these challenges, we define and use a generalization of BTER that mimics its two-step edge generation process, but allows explicit prescription of the communities' size, membership, and approximate density. The remainder of this section describes this generalized version and compares it to the original BTER model.

Our new model, the Generalized Block Two-level Erdös-Rényi (GBTER) model takes as input (1) the expected degree of each node, (2) community assignments of the nodes, i.e., a partition of the vertex set into disjoint subsets, $\{C_j\}$, and (3) an edge probability p_j for edges within each community C_j. In the first stage of edge generation, within-community edges are sampled from an Erdös-Rényi [7] model, $\mathrm{ER}(|C_j|, p_j)$, for each community C_j. Note the expected degree of a node within C_j is $p_j(|C_j|-1)$ after the first stage. In the second stage, we define the *excess expected degree* of a node i, denoted ε_i, to be the difference between the input expected degree λ_i and the expected degree after stage one. Formally, $\varepsilon_i := \max(0, \lambda_i - p_j(|C_j|-1))$ for node i in community C_j. We then apply a Chung-Lu style model [5] on the excess expected degree-sequence, $[\varepsilon_i]_{i\in V}$. Specifically, the probability of adding the edge (i,j), is

$$P(i,j \mid \varepsilon) = \frac{\varepsilon_i \varepsilon_j}{\sum_k \varepsilon_k}. \tag{1}$$

Note that the second stage can generate both inter- and intra-community edges. It is necessary that Chung-Lu inputs, $\{\varepsilon_i\}$, satisfy $\varepsilon_i\varepsilon_j \le \sum_k \varepsilon_k$ for Equation 1 to define a probability. A calculation shows that d_i (the user's

[1] denoted here as $\mathrm{ER}(n,p)$, in which n nodes are fixed and edges occur independently with probability p

input expected degree for node i) is indeed the expected degree of node i under this model whenever the CL model is well defined, i.e., whenever $d_i \geq p_j(|C_j| - 1)$ (the expected degree from the first-stage edges does not exceed the total expected degree of any node).

To calculate the probability of edge (i, j), we condition on whether i and j share a community. Recall, our communities partition the set of nodes, so each i is in exactly one community. If i and j are assigned to the same community, C, let p denote the internal edge probability of C, and we see

$$P(i, j \mid i, j \in C) = p + (1 - p)\frac{\varepsilon_i \varepsilon_j}{\sum \varepsilon_k}. \tag{2}$$

If i and j are assigned to different communities, the edge probability is as given in Equation 1.

GBTER differs from the original BTER model by allowing greater flexibility and assignment of community membership, size, and internal edge density (p). As indicated in [13], the expected clustering coefficient for an ER(n, p) graph is p^3. This implies that GBTER also allows pre-specification of each community's approximate clustering coefficient. Note that GBTER assumes node labels allowing specification of community assignments, while BTER only depends on the number of nodes of each expected degree, and BTER community assignment is not known/specified à priori. Finally, since edge and graph probabilities depend on community assignments (in both models), such calculations are complicated and expensive in the BTER model (as all possible community assignments must be considered), which inhibits its use for anomaly detection. Knowing the community assignments makes computation of edge probabilities straightforward with GBTER, as discussed in the next section.

4 Fitting Model Parameters

We now describe how to fit the GBTER model to observations. Our assumption is that we have a sequence of observed graphs with common node labels (i.e., node i in graph G_n is identified as node i also in G_{n+1}). Because we expect to detect changes in community density, membership, and node degree, the detection capability relies on an implicit assumption that these are fairly stable as the graph sequence progresses. Below we discuss the community partitioning algorithm used in our experiments, and present Bayesian techniques for inferring the within-community edge densities and the expected node degrees. Once a specific instance of the model is deduced, probabilistic anomaly detectors are constructed, as detailed in Section 5.

4.1 Community Assignments

We begin by noting that the GBTER model and ensuring detection capabilities require and are defined on any partition of the nodes into communities.

As a degenerate case, using each node as a singleton community reduces the GBTER model to the well-known Chung-Lu model, and the detection capability will only regard node degree in this case. While the GBTER model and the detection capabilities are defined for any partition of the nodes, reasonable results (e.g., results that do not detect a majority of graphs, communities, or nodes as anomalous) rely on a partition of each graph in the sequence into communities that are relatively stable. Specifically, we expect that the communities are disjoint subgraphs with relatively high density of edges and that the densities of the communities and the memberships do not change dramatically from G_n and G_{n+1} often.

In this work, a partition of the vertex set into communities is learned using the Markov Clustering (MC) algorithm of Van Dongen [21] and a brief description of this clustering algorithm is given in Appendix 9.1. We chose MC as it can scale well, is easy to implement, and is known to partition the graph into relatively dense subgraphs. To apply MC, a weighted graph is constructed from observed graphs; specifically, for the experiments in Section 6, the weighted graph is constructed by counting the occurrence of each edge, and for the application in Section 7 exponential weights are used to down weight older observation of edges (i.e., given observed adjacency matrices $A_{-N}, ..., A_{-1}, A_0$ we apply MC to $A = \sum_{i=0}^{N} A_{-i} s^{i+1}$ for $0 < s < 1$). We note that one possibility is to use communities known from the application domain (e.g., grouping nodes by a known affiliation), which can obviate the partitioning of nodes into communities and may provide more insightful results in a real-world setting. Finally, for more information on community detection algorithms see the survey of Fortunato, [9].

4.2 Community Densities

Given community assignments, the within-community edge densities are estimated. Each community, C, is modeled internally by an Erdös-Rényi random graph, $\mathrm{ER}(|C|, p)$, and we seek to estimate p. Letting k denote the number of edges within the subgraph C, it follows that $k \sim \mathrm{Binomial}(\binom{|C|}{2}, p)$. In order to use Bayesian inference, we assume $p \sim \mathrm{Beta}(\alpha, \beta)$, with prior parameters $\alpha > 0, \beta > 0$, and then use the maximum á posterior likelihood estimation (MAP). The reason for this is that in the presence of only a few observations, non-representative distributions can be realized when using simply the maximum likelihood estimation (MLE), yet, in the presence of a large number of observations, the MAP and MLE are nearly equal and should converge to the (assumed) underlying distribution. Specifically, $(p|k_1, \ldots, k_N) \sim \mathrm{Beta}(\hat{\alpha}, \hat{\beta})$ with posterior parameters

$$\hat{\alpha} = \alpha + \sum_i k_i, \text{ and } \hat{\beta} = \beta + N\binom{|C|}{2} - \sum_i k_i$$

where k_i denotes the number of edges internal to C observed in the i-th graph, G_i, for $i = 1, \ldots, N$. MAP gives $p := (\hat{\alpha} - 1)/(\hat{\alpha} + \hat{\beta} - 2)$, the mode of the posterior.

4.3 Expected Degree Sequence

Lastly, the expected degree sequence must be estimated from the data. For a fixed node, we assume its degree, d, is Poisson distributed with expected degree λ, i.e. $d \sim \text{Poisson}(\lambda)$. We use the conjugate prior, $\lambda \sim \text{Gamma}(\alpha, \beta)$ with prior parameters $\alpha > 1, , \beta > 1$. This yields the posterior distribution, $(\lambda | d_1, \ldots, d_N) \sim \text{Gamma}(\hat{\alpha}, \hat{\beta})$ with posterior parameters $\hat{\alpha} = \alpha + \sum_i d_i$, and $\hat{\beta} = \hat{\beta} + N$, where d_i denotes the observed degree of the node in G_i. For each node, MAP gives its expected degree, $\lambda := (\hat{\alpha} - 1)/\hat{\beta}$, the mode of the posterior Gamma.

5 Anomaly Detectors

Given an instance of a GBTER model, which defines a probability distribution on graphs, one can leverage the distribution to detect anomalies at the graph, subgraph, and node level. This section defines two multi-scale detectors, one which uses the GBTER distribution directly, and one which leverages statistics inherent to the GBTER model. The Multi-Scale Probability Detector naturally uses the graph probability as determined by the GBTER model for detection, which is then decomposed into probabilities of subgraphs and nodes for hierarchical information. Although intuitive, this detector suffers from a few limitations, discussed below, which informs construction of the Multi-Scale Statistics Detector. This second detector builds from the bottom up defining the probability of a node based on the likelihood of its internal and external degree. Subgraph probabilities are determined by those of its member nodes, so multi-scale analysis is facilitated by both models. Lastly, a baseline method for detecting anomalous graphs by fitting Gaussian distributions to graph statistics, is described. We note that the Gaussian Baseline is only used for identifying anomalous graphs, as it cannot discriminate anomalies at the subgraph or node level. Section 6 gives results of testing the three methods on synthetic (seeded) data and Section 7 on NCAA football data.

5.1 Multi-Scale Probability Detector

Our first anomaly detector uses the graph probability, as given by the GBTER model, for anomaly detection. Specifically, given a graph $G = (V, E)$ with vertices V and edges E, the probability of G is

$$P(G) = \prod_{(i,j) \in E} P(i,j) \prod_{(i,j) \notin E} (1 - P(i,j)), \tag{3}$$

where $P(i,j)$ is the probability of the edge (i,j) under the GBTER model, as derived in Section 3. In practice, given a graph G, we compute it's probability using Equation 3, then use Monte-Carlo simulation to estimate its p-value.

In order to detect anomalies at different scales, the probability of a graph is decomposed into a product of subgraph probabilities. Specifically, we define the probability of node i_0 as

$$P(i_0) := \prod_{j:(i_0,j)\in E} P(i_0,j) \prod_{j:(i_0,j)\notin E} (1-P(i_0,j)).$$

It follows that $P(G) = \prod_i P(i)^{1/2}$. Similarly, the probability of a subgraph $G' = (V', E')$ is $\prod P(i)^{1/2}$, with the product over $i \in V'$. Hence, given a partition of V into communities, $\{C_i\}$, the probability of G also breaks into a product of community probabilities, i.e., $P(G) = \prod_i P(C_i)$. This formulation allows anomaly detection of any fixed subgraph, in particular at the node, community, and graph level.

The probability of sampling a graph under a given generative model is an intuitive choice for anomaly detection. Upon further examination, this technique will yield poor results in models where the mode of the distribution varies depending on whether or not labels are regarded. As an illustrative example, consider the ER model on three labeled nodes, $V = \{1,2,3\}$ with $p = 1/3$. The most probable unlabeled graph under this distribution has exactly one edge, and occurs with probability $\binom{3}{1}(1/3)(2/3)^2 = 4/9$. Now labeling nodes, there are three different but isomorphic graphs with one edge each, namely, with edge $(1,2)$ or $(2,3)$ or $(1,3)$ only. But the probability of each of these one-edge graphs is $(1/3)(2/3)^2 = 4/27$, while the probability of the empty graph is $(2/3)^3 = 8/27$. Hence when labels are regarded, the mode of the distribution is the empty graph, not the one-edge graphs as in the unlabeled case; consequently, in this case the Multi-Scale Probability Model will view the expected graphs as more anomalous than the less likely empty graph! Now consider the GBTER model used in the experiment above. Because the probability of a within-community edge is greater than $1/2$ and inter-community edge is less than $1/2$ with the given parameters, the labeled-node mode of the distribution is the graph with every community as a clique and no other edges. Although this graph is unlikely to be sampled, the Multi-Scale Probability Model will regard it as the most "normal" possible graph. The conclusion of this reasoning is that using the graph's probability will produce unwarranted results, yet modeling characterizing statistics of the graph (e.g., inter- and intra-community node degrees) gives accurate detection capabilities. This is exhibited in our empirical results, and motivates the second detector.

5.2 Multi-Scale Statistics Detector

Our second detector is based on observing and modeling intra- and inter-community node degrees (after learning GBTER parameters). Fix a node $i_0 \in V$, and let C denote i_0's community, p denote C's intra-community edge

probability, and λ the expected degree of node i_0 (all as learned from fitting the GBTER model to our observations). We set $d_{in} := |\{(i_0, j) \in E : j \in C\}| =$ i_0's internal degree, and $d_{ex} := |\{(i_0, j) \in E : j \notin C\}| = i_0$'s external degree. Following the ER($|C|, p$) assumption, we assume $d_{in} \sim$ Binomial($|C| - 1, p$), and $d_{ex} \sim$Poisson(ε), where $\varepsilon = \max(0, \lambda - p(|C| - 1))$, is the excess expected degree of i_0 (see Section 3). For the Multi-Scale Statistics anomaly detector, the probability of node i_0 is defined as the joint probability of its degrees, $P(i_0) := P(d_{in}, d_{ex})$. We assume the two degrees are independent and obtain,

$$\begin{aligned} P(i_0) :&= P(d_{in}, d_{ex}) \\ &= \binom{|C| - 1}{d_{in}} p^{d_{in}} (1 - p)^{|C|-1-d_{in}} \frac{e^{-\varepsilon} \varepsilon^{d_{ex}}}{d_{ex}!}. \end{aligned}$$

Given a subgraph $G' = (V', E')$ we set $P(G') := \prod_{V'} P(i)$. Hence anomaly detection of any subgraph is made possible.

Note that since GBTER allows both internal and external edges to be created by the second stage of the process, the model above inflates internal degree d_{in} and deflates d_{ex} compared to GBTER. Additionally, as the range of a Poisson variable is unbounded, degrees exceeding $|V| - 1$ (an impossibility) are assigned positive probability by this model. To circumvent this possibility, the truncated Poisson can be used for sampling. In our experiments, the expected degree (λ) and expected excess degree (ε) are sufficiently smaller than $|V| - 1$, which implies the $P(\deg(i) > |V| - 1|)$ is negligible. Testing with and without the truncation exhibited similar results.

To use either of the multi-scale detectors, we set thresholds at each level, and any node/subgraph/graph with p-value below the respective threshold is detected. The model parameters are updated upon receipt and detection of each graph. A discussion of previous work addressing how to set the alert rate is given in Section 2. To summarize, methods for choosing the threshold by bounding the alert rate are possible as well as less sophisticated techniques of checking results on a labeled subset of data and choosing an acceptable true- versus false-positive rate.

5.3 Gaussian Baseline Detector

Our baseline method fits univariate Gaussian distributions to graph statistics and uses the product of the p-values for detection. From each observed graph three statistics are obtained: average node degree (X_1), average clustering coefficient (X_2), and the spectral norm (X_3). For the definition of spectral norm see Appendix 9.3. Calculating X_1 and X_2 from a given graph is straightforward. In order to calculate X_3, the GBTER model is used with parameters estimated as described above to produce the expected adjacency matrix $E(A)$, in which $E(A)_{i,j}$ gives the probability of an edge between nodes i and j. The spectral norm is defined as the maximum modulus eigenvalue of the residual

Table 1: Community assignments for GBTER Experiment

	C_1	C_2	C_3	C_4	...	C_{10}
M_r	[0,1,2,3]	[4,5,6,7]	[8,9,10,11]	[12,...,15]	...	[36,...,39]
M_a	[0, *11*, 2, *4*]	[*3*, 5, 6, *8*]	[*7*, 9, 10, *1*]	[12,...,15]	...	[36,...,39]

Note: The seeded-anomaly model M_a is obtained from M_r by switching the position of 2 nodes from each of the first 3 communities. Anomalous nodes shown in italicized red print, and anomalous communities are circled.

matrix $A - E(A)$. After computing the observed statistics, independent univariate Gaussian distributions ($\mathcal{N}(\mu_i, \sigma_i)$) are fit to each of the three statistics. Lastly, given a newly observed graph, G, with statistics x_1, x_2, x_3, we assign

$$p\text{-value}(G) := \prod_{i=1}^{3} P(X_i \leq x_i | \mathcal{N}(\mu_i, \sigma_i)).$$

As before, p-values falling below a given threshold, α, are labeled anomalous, and the three normal distributions are updated upon receipt of each new graph.

This follows the approach of Moreno and Neville [17], although their work is based on mixed Kronecker Product graphs and uses average geodesic distance instead of the spectral norm we employ for X_3. Since the average geodesic distance is undefined for disconnected graphs, we selected the spectral norm based on prior use in network hypothesis testing and strong results for similar tests involving Chung-Lu random graphs [16]. While we consider this baseline a natural adaptation of [17], the disparity in use between their and our application inhibits direct comparison.

6 Synthetic Graph Experiment

In order to test the anomaly detection capabilities, two hidden GBTER models are used to generate labeled data, (1) a "regular" model, M_r, for sampling non-anomalous graphs, and (2) a seeded-anomaly model, M_a, with slightly perturbed inputs to generate anomalous graphs. To begin the experiment, 100 non-anomalous graphs are sampled from M_r, and the anomaly detectors are fit to the data, as described in Section 5. Prior distributions Beta(1,1), Gamma(2,2) were used. To test the streaming anomaly detection, 500 graphs are iteratively generated and observed with every fifth graph from the seeded anomaly model. Upon sampling a new graph, its p-value according to each anomaly detector is computed (and it is labeled as anomalous if it falls below a given threshold). Similarly, the hierarchical detectors label each node and community depending on its respective p-value. An more explicit account of this streaming experiment algorithm is given in the Appendix 9. For comparison of the models, we choose the threshold, α, which optimizes F1 score.[2]

[2] F1 is defined as the harmonic average of Precision, P, and Recall, R. Specifically, F1:= $(\text{average}(P^{-1},R^{-1}))^{-1} = 2PR/(P+R)$.

Lastly, each anomaly detector's GBTER parameters are updated to include observation of the new graph.

We conduct two experiments, both using networks of 40 nodes divided into ten equally-sized communities. For the "regular" model, each community is assigned a within-edge probability of $p = .8$, and the expected degrees of nodes vary in the range of five to eight according to a truncated power-law. To create the seeded-anomaly model for the first experiment, two nodes from each of the first three communities are interchanged resulting in six (of 40) anomalous nodes and three (of ten) anomalous communities per anomalous graph (see Table 1). For the second experiment, community assignments are held constant, but the within-community density (p) of the first four communities is changed from 0.8 to 0.4 in the seeded-anomaly model, and the expected degree of the nodes in these four communities is increased by two. This will decrease intra-community, and increase extra-community interaction for these four communities. All together the second experiment has four (of ten) anomalous communities, and 16 (of 40) anomalous nodes per anomalous graph.

To evaluate the detectors' performance, the Receiver Operator Characteristic (ROC) curve, and area under the ROC curve (AUC) are displayed in Figure 1. Results of the experiments are given in Table 2, which includes the AUC as well as Precision, Recall, and F1 for each detector at the threshold α maximizing its F1 score. Recall that the Gaussian Baseline is only for graph level detection and thus does not contribute to the community or node level results. For the full-graph tests, the Gaussian Baseline is far inferior to the new models with the Multi-Scale Statistics Detector as the clear winner. Further, the results at all levels provide evidence that the Multi-Scale Statistics Detector is the superior method, as expected after the à priori analysis given in Section 5.1.

7 NCAA Football Data Experiment

To illustrate the insight given by multi-scale anomaly detection on real-world data, the Graph Statistics Model is applied to NCAA Football data [14]. For comparison, we also run the Gaussian Baseline Detector. Each season is represented as a graph with a node for each Division I team and an edge for each game played. Seasons 2008 and 2009 are used to fit parameters of the models initially, and the streaming detection is performed on years 2010-2012. That is, after fitting parameters on previously observed years, the detectors give p-values for the newly observed season. Then, the parameters are updated to include the newly observed data (and the detectors are applied to the next year). This dataset was chosen for two reasons, (1) NCAA conferences give a ground-truth community structure to the graph, and (2) conference membership was relatively constant in the 2008-2010 seasons but experienced changes in 2011 and 2012. Because teams play most of their schedule within their

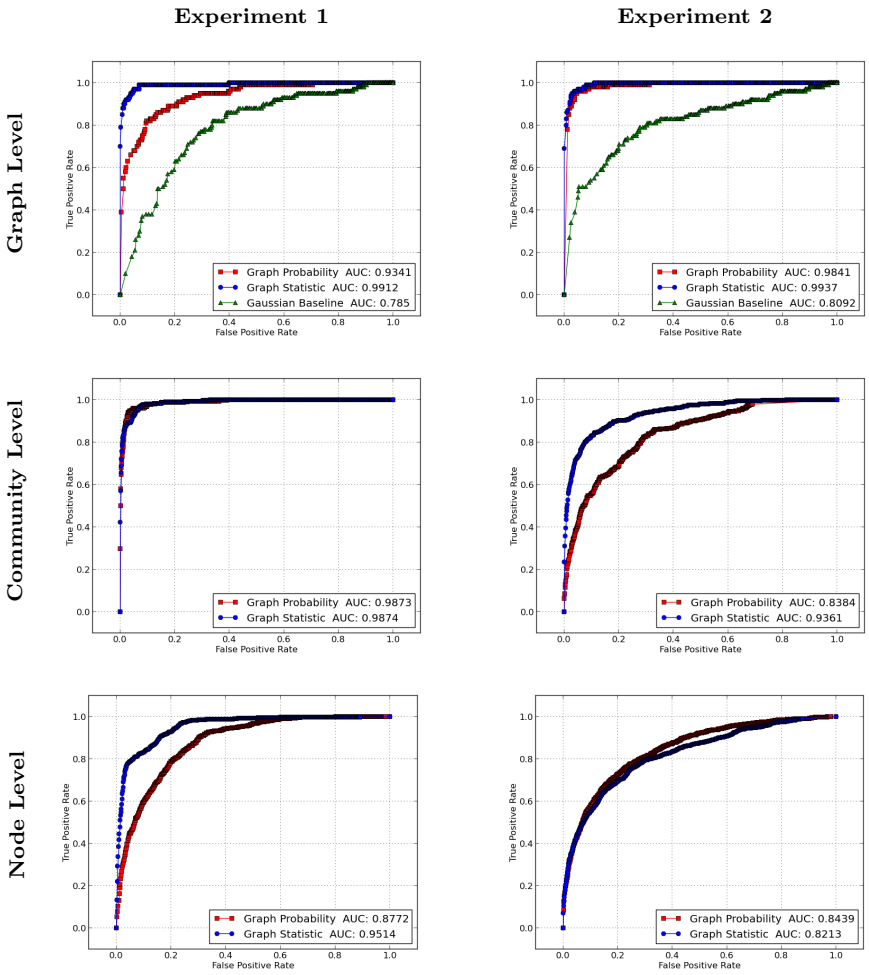
Table 2: GBTER Synthetic Data Experiment Results

Method	α	F1	Precision	Recall	AUC
Experiment 1					
Graph Level					
Graph Probability	0.020	0.742	0.678	0.820	0.934
Graph Statistic	0.009	**0.919**	**0.929**	**0.910**	**0.991**
Gaussian Baseline	0.029	0.526	0.418	0.710	0.785
Community Level					
Graph Probability	0.019	0.810	0.745	**0.887**	**0.987**
Graph Statistic	0.009	**0.830**	**0.840**	0.820	**0.987**
Node Level					
Graph Probability	0.020	0.298	0.239	0.393	0.877
Graph Statistic	0.017	**0.547**	**0.453**	**0.690**	**0.951**
Experiment 2					
Graph Level					
Graph Probability	0.007	0.895	0.855	**0.940**	0.984
Graph Statistic	0.011	**0.922**	**0.904**	**0.940**	**0.993**
Gaussian Baseline	0.006	0.590	0.697	0.510	0.809
Community Level					
Graph Probability	0.062	0.436	0.390	0.495	0.838
Graph Statistic	0.028	**0.654**	**0.620**	**0.693**	**0.936**
Node Level					
Graph Probability	0.053	**0.436**	0.368	**0.533**	**0.894**
Graph Statistic	0.047	0.434	**0.427**	0.442	0.821

Note: This table gives results from the synthetic experiment. In both experiments graphs are sampled from a GBTER graph model. To produce ground-truth anomalies, in experiment 1 graphs are occasionally sampled from the same model with a few nodes changing community membership, while in experiment 2 community membership is held fixed but community density is turned down. Streaming anomaly detection is performed to test the algorithm. This table presents results using α chosen to maximize F1 score.

conference, these changes are reflected in the season's graph and community structure. Our expectation is that the 2010 graph should produce a relatively higher p-value (be less anomalous) than the next two years. Furthermore, we expect our multi-scale detector to pinpoint the conferences and teams that experienced change. For the experiment, the parameters are learned as discussed in Section 5. Communities are detected using Markov clustering as before but with exponential down-weighting of previous years' edges as in Section 3. With appropriate configuration of Markov clustering parameters, the communities identified match almost identically with actual conferences, and we à posteriori label/refer to communities by the corresponding conference name for ease of discussion.

Fig. 1: ROC Curves from Synthetic Data Experiments



7.1 Football Data Results

Both the Gaussian Baseline Detector and the new Graph Statistics Detector accurately classified the full graphs (seasons), identifying 2010 as non-anomalous, and 2011 and 2012 as anomalous graphs. More specifically, the Gaussian Baseline reported scores of $13 * 10^{-5}$ for 2010, numerical 0 for 2011, and $5.2 * 10^{-10}$ for 2012—recall this is the product of three Gaussian p-values attained from their CDFs. A threshold between 10^{-10} and 10^{-5} will give accurate classification. Our Graph Statistics Detector reported scores of 1.0 for 2010, and 0.0 for 2011-12, indicating that no graph sampled in the Monte Carlo simulation was more probable than the 2010 graph, and none were less probable than the 2011 (or 2012) graph.

Table 3: Each season's conferences ranked by anomalousness displayed with $p-$value and number of membership changes. Blue entries are true-positives while red entries are false-positives with threshold $\alpha = 10^{-4}$.

2010			**2011**			**2012**		
Conf.	$p-$**value**	n	**Conf.**	$p-$**value**	n	**Conf.**	$p-$**value**	n
ACC	0.000	0	MWC	0.000	3	WAC	0.000	5
WAC	0.001	0	PAC-10	0.000	2	Big-12	0.000	4
CUSA	0.090	0	Big-12	0.000	2	MWC	0.000	4
PAC-10	0.272	0	WAC	0.000	1	Big-East	0.000	2
Big-12	0.295	0	CUSA	0.000	0	SEC	0.000	2
MWC	0.433	0	SEC	0.178	0	MAC	0.000	2
Sun-Belt	0.455	0	MAC	0.211	0	Sun-Belt	0.000	1
MAC	0.551	0	ACC	0.287	0	PAC-10	0.000	0
SEC	0.639	0	Big-East	0.324	0	CUSA	0.002	0
Big-East	0.646	0	Big-10	0.513	1	ACC	0.965	0
Big-10	0.721	0	Sun-Belt	0.537	0	Big-10	0.973	0

In addition to identifying the seasons that are/are not anomalous, our method detects the conference from the graph structure, and gives p-values for the conferences and individual teams that are causing the anomaly. Table 3 ranks the most anomalous conferences detected each year by the Graph Statistics Detector. Each conference experiencing a change in membership is detected as maximally anomalous, with $p-$value equal to 0. Across all three years, there were a total of three false-positives, and no false-negatives. At the conference level this gives precision of $11/14 \approxeq .786$, and perfect recall (11/11).

The results for the Graph Statistics Detector at the node level are given in Table 4, which details the ten most anomalous teams from each season in decreasing order along with their $p-$value and ground-truth conference memberships for the previous and current season. We notice that a threshold of $\alpha = 10^{-6}$ gives perfect classification, identifying exactly which teams changed conferences as anomalous. In short, this method tells not only which graphs are anomalous, but with high accuracy can pinpoint the nodes and communities causing the anomaly.

7.2 Interactive Data Visualization

By design, the multi-scale detector allows users to focus attention on noteworthy communities and nodes and facilitates an interactive visualization tool for easily accessing the fine-grained structure of anomalous areas of the graph. Figure 2 illustrates the benefits of this approach in screenshots from a prototype visualization. While the 2011 graph (Figure 2.a), consists of only $\sim$130 nodes and has well-defined community structure, an unprocessed visualization provides little insight into the anomalous sections of the graph. Alternatively, coarsening and displaying only "super"-nodes representing communities and using darker shades to indicate increased anomalousness, obviates the communities of interest (Figure 2.b). In addition, our tool allows conference names and p-values to be automatically displayed so contextual information from

Table 4: Ten Most Anomalous Teams for years 2010, 2011, & 2012 as given by Graph Statistics Detector. Threshold $\alpha = 10^{-6}$ gives perfect classification.

Team	$p-$**value**	**2009 Conf.**	**2010 Conf.**
Wake Forest	4.84e-4	ACC	ACC
Washington	2.17e-3	PAC-10	PAC-10
San Jose St.	2.78e-3	WAC	WAC
Utah St.	3.05e-3	WAC	WAC
Tulsa	4.32e-3	CUSA	CUSA
Toledo	1.25e-2	MAC	MAC
San Diego St.	1.50e-2	MWC	MWC
Maryland	2.21e-2	ACC	ACC
North Carolina	2.21e-2	ACC	ACC
North Carolina St.	2.63e-2	ACC	ACC

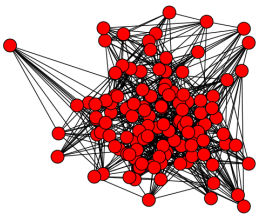
Team	$p-$**value**	**2010 Conf.**	**2011 Conf.**
Boise St.	2.81e-18	WAC	MWC
Utah	1.59e-12	MWC	PAC-10
Brigham Young	6.33e-10	MWC	-
Colorado	1.70e-09	Big-12	PAC-10
Nebraska	3.53e-08	Big-12	Big-10
Washington St.	1.29e-06	PAC-10	PAC-10
Washington	2.28e-06	PAC-10	PAC-10
Arizona St.	3.28e-06	PAC-10	PAC-10
San Jose St.	2.26e-05	WAC	WAC
Utah St.	2.31e-06	WAC	WAC

Team	$p-$**value**	**2011 Conf.**	**2012 Conf.**
Missouri	3.95e-16	Big-12	SEC
W. Virginia	7.70e-14	Big-East	Big-12
Texas A&M	4.03e-13	Big-12	SEC
Texas Christian	7.66e-10	MWC	Big-12
Temple	1.28e-09	MAC	Big-East
Nevada	2.55e-07	WAC	MWC
Fresno St.	3.08e-07	WAC	MWC
Hawaii	4.19e-07	WAC	MWC
Texas	4.10e-05	Big-12	Big-12
Mississippi	1.04e-04	SEC	SEC

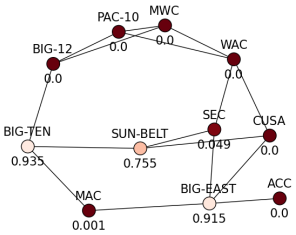
the analysis and the domain are easily absorbed by a user. Conference nodes are clickable, and selection displays the inter-conference subgraph, again with nodes shaded to indicate anomalousness of the teams they represent. This setup facilitates interactive exploration of anomalies, and the contexts in which they occur. For example, clicking on the Mountain West Conference (MWC) node displays the graph in Figure 2.c, from which it is immediately apparent that, while Utah and Brigham Young were previously members of that community, they cease to participate in the MWC. The PAC-10 Conference subgraph (Figure 2.d) exhibits high density, but each node is very anomalous. This indicates that the interaction outside the conference has changed and referencing the tables confirms that new teams, namely Utah and Colorado, are now in this conference. Altogether, the framework for multi-scale detection yields analytic results that are readily input into an interactive visualization.

Fig. 2: 2011 Season Graph interactive visualizations screenshots (produced by MatPlotLib using NetworkX [10] spring force directed layout) illustrate discovery of anomalies at each level. Team and conference labels were applied in post-analysis along with p-values in (b)-(d). Darker colors correspond to more anomalous communities/nodes.

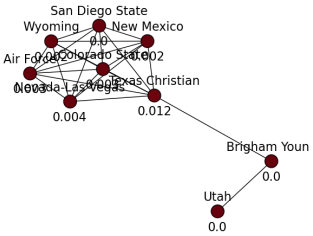
(a) Unprocessed 2011 Season Graph



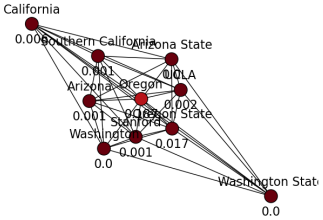
(b) Coarsened 2011 Season Graph



(c) 2011 Mountain West Conference Graph



(d) 2011 PAC-10 Conference Graph



Upon detection of an anomalous graph, users can now zoom into areas of interest, and form and resolve hypotheses about how the anomaly occurred.

8 Conclusions and Future Work

As many applications involve representing data with known entities and time-varying relationships, this work considers a sequence of graphs with node labels and changing edges. Our goals were to investigate a method for finding abnormalities in such a graph sequence that (1) use multiple, related levels of granularity to facilitate an understanding of why/how an anomaly occurred, and (2) to leverage node labels for more accurate detection. To this end, we introduced GBTER, a generalization of the BTER graph model, that allows more accurate modeling of community structure, and built two hierarchical streaming anomaly detectors. The first intuitively uses the graph's probability as given by the model, yet more thorough analysis suggests that the inability of graph models to distinguish isomorphic copies with different node labels will inhibit detection accuracy. Secondly, a statistics-based detector,

that respects the node labels in each graph is created. Our hypothesis that the statistics-based detector will give more accurate results is verified in two tests on synthetic data where ground-truth is known at the node, subgraph, and graph levels. Additionally, both detectors outperform a baseline detector that fits Gaussian distributions to observed statistics of the full graph. In order to illustrate the insight facilitated by the multi-scale detection capability, the superior multi-scale detector is applied to NCAA football data. In both the synthetic experiment and the application to NCAA data, the Multi-Scale Statistics Detector was able to accurately pinpoint anomalies at the node, subgraph, and graph level, exhibiting the advantage of drilling into anomalous graphs to see exactly what has deviated from expectation. Furthermore, we exhibit a preliminary interactive visualization, which informed by the analytics allows operators to drill into communities of interest to see anomalies at all levels. We believe applications of this method to other time-sampled social networks will enable discovery of underlying structure and anomalies within the context they occur.

While investigations of scalability are outside the scope of this work, we expect applications of this approach to necessitate larger data. Here we identify the bottlenecks in the current implementation for future efforts.

Firstly, this approach requires a partition of the nodes into communities, but is agnostic to the method used. This flexibility gives opportunities to optimize performance by the partitioning algorithm chosen. As mentioned above, using communities known from context (e.g., assuming knowledge of the NCAA conferences in each previous year) can obviate this step and provide groupings that are familiar to the operator.

Secondly, estimating the p-values of a given distribution can be computationally expensive, especially if it requires sampling large graphs and calculating their probabilities. Our experiments used Monte Carlo simulations to estimate the p-values; consequently, estimating binomial and Poisson distributed p-values was fast, while sampling the graph from the GBTER model was not. This means that node and community level detection are happening quickly (matter of seconds on graphs up to 1000 nodes) in the initial implementation. We note that Kolda et al. [13] have published a scalable implementation of the BTER model leveraging a critical observation—that the within-community degrees and excess-expected degrees of each node are known before sampling, so the two-stage edge generation process can happen in parallel. As this observations holds for the GBTER model, similar scalable implementation techniques should be possible. More generally, the problem of computationally expensive p-value calculations has other solutions. For example, importance sampling, in which one over-samples from a subset of the event space, can aid in Monte-Carlo simulations, although further research is required to optimize performance gains for our needs.

Thirdly, the choice of probability models of the parameters could be changed to admit easier p-value computation. For example, multinomials become robust with abundant observations, and often admit more efficient calculations. Furthermore, if one modeled the within and external degrees of a node sep-

arately, rather than as a joint distribution, one may obviate the p-value calculations for each (depending on the distribution). In short, given a specific application, flexibility in the modeling may yield increased performance with negligible effects on accuracy. Lastly, adapting the overall workflow to fit a specific application may admit performance gains. For example, updating parameters less often (in a batch process periodically) or discarding anomalous data (after detection) from the update step are options that have yet to be explored. In summary, while the current implementation is suitable only for small datasets, the approach gives opportunities for scalability. We believe that the overall method can be adapted to much higher volume and/or larger network settings, but such implementations are left for follow-on research.

Finally, investigations of the GBTER modeling capabilities is a promising avenue for further investigations. Fitting the model to a collection of known data sets and testing the model fit with respect to different metrics (e.g., degree distribution, characteristic path length, clustering coefficient, etc.) would provide empirical evidence for how and when the GBTER model performs well.

9 Appendix

This supplement provides the algorithm used for the streaming anomaly detection experiment and additional relevant definitions.

9.1 Markov Clustering Algorithm

A brief description of the Markov Clustering Algorithm is provided below. The original citation is [21] and for completeness see [19]. Conceptually, we considered the adjacency matrix an (unnormalized) transition matrix. In order to ensure jumping from a given node to itself is possible, the identity matrix is added to the graph's adjacency matrix. Next, to make the rows indeed probability distributions over nodes, this resultant matrix is converted to a row stochastic matrix. The algorithm iterates the following five steps until convergence,

1. The matrix is squared.
2. Each element in the matrix is squared. (This means relatively small/large probability edges get smaller/larger relative to each other.)
3. The matrix is row stochasticized.
4. Entries with values below a given parameter are removed to save memory.
5. The matrix is row stochasticized.

In this work we have chosen parameter values such that the fourth step is never completed since the graphs in this paper can fit into memory easily. Convergence is not mathematically guaranteed and oscillation has been observed. To our knowledge finding necessary and sufficient conditions for convergence is

an open problem. In practice this process almost always converges to a matrix in which two columns are either identical or have disjoint support. Defining communities as the set of nodes which have a non-zero entry in a fixed column gives a partition of the nodes into communities.

9.2 Clustering Coefficients

Given a graph, $G = (V, E)$, the *global clustering coefficient*, denoted c, is defined as the ratio of the number of wedges incident to any node by the number of triangles incident to any node. Explicitly, if

$$T_j := |\{i, k \in V : i \neq k, (i, j), (j, k), \text{ and } (k, i) \in E\}|$$

is the number of triangles incident to node j and

$$W_j := |\{i, k \in V : i \neq k, (i, j) \text{ and } (j, k) \in E\}|$$

is the number of wedges incident to j,

$$c := \frac{\sum_{j \in V} T_j}{\sum_{j \in V} W_j}. \tag{4}$$

This definition can be restricted to subgraphs by changing the summation. As used in the BTER model, for a fixed degree, d, the clustering coefficient of nodes with degree d, denoted c_d is defined as in Equation 4 with summation only over those nodes with degree d. Lastly, as used in [17] and the Gaussian Baseline method described in section 5.3, *average clustering coefficient* is defined as the average of the clustering coefficient of each node; formally,

$$\frac{1}{|V|} \sum_{j \in V} \frac{T_j}{W_j}. \tag{5}$$

9.3 Spectral Norms

Given both a graph and a generative graph model, the *spectral norm* is defined as the spectral radius of residual matrix, that is, the largest eigenvalue in absolute value. The residual matrix is $A - E$, were A denotes the adjacency matrix of the given graph, and E is the expected adjacency matrix. The expected adjacency matrix entries are given by the edge probabilities, i.e., $E(i, j) := P(i, j)$, as determined by the graph model. This definition follows the formulation in [16].

9.4 Geodesic Distances

The *geodesic distance* between two nodes, i, j of a graph $G = (V, E)$ is the length of a shortest path between i and j, given such a path exists. In the case that i, j lie in different components, their geodesic distance is ∞. Intuitively, the *average geodesic distance* of a graph G is the average of the geodesic distances over all possible pairs of nodes.

Algorithm 1: Streaming Detection Experiment

Input: N = number of training graphs
K = number of streamed graphs
k = factor of streamed graphs to come from seeded anomaly model
M_r ="regular" model
M_a = seeded-anomaly model
α = threshold
Output: $\{(G_{n_i}, p-\text{value}(G_{n_i}))\}$ = set of anomalous graphs with corresponding p-values
Initialize $Graphs \leftarrow \{\}$, $Anomalous \leftarrow \{\}, i \leftarrow 0$
while $i < N$ **do**
  Sample G_i from M_r
  Store G_i in $Graphs$
  $i+ = 1$
Learn M_N (anomaly detector model) parameters from $Graphs$
while $i < N + K$ **do**
  if $i\%k == 0$ **then**
    Sample G_i from M_a
  else
    Sample G_i from M_r
  Store G_i in $Graphs$
  Set $pv \leftarrow P(\{G : P(G) \leq P(G_i)\}|M_{i-1})$
  if $pv \leq \alpha$ **then**
    Store (G_i, pv) in $Anomalous$
  Update M_i to M_{i+1} using $Graphs$
  $i+ = 1$
return $Anomalous$

References

1. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery pp. 1–63 (2014)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
3. Bridges, R.A., Collins, J.P., Ferragut, E.M., Laska, J.A., Sullivan, B.D.: Multi-level anomaly detection on time-varying graph data. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pp. 579–583. ACM (2015)

4. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Computing Surveys (CSUR) **38**(1), 2 (2006)
5. Chung, F., Lu, L.: The average distances in random graphs with given expected degrees. Proceedings of the National Academy of Sciences **99**(25), 15,879–15,882 (2002)
6. Eberle, W., Holder, L.: Anomaly detection in data represented as graphs. Intelligent Data Analysis **11**(6), 663–689 (2007)
7. Erdős, P., Rényi, A.: On random graphs. Publicationes Mathematicae Debrecen **6**, 290–297 (1959)
8. Ferragut, E.M., Laska, J., Bridges, R.A.: A new, principled approach to anomaly detection. In: International Conference on Machine Learning and Applications, vol. 2, pp. 210–215. IEEE (2012)
9. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3), 75–174 (2010)
10. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy2008), pp. 11–15. Pasadena, CA USA (2008)
11. Harshaw, C.R., Bridges, R.A., Iannacone, M.D., Reed, J.W., Goodall, J.R.: Graphprints: Towards a graph analytic method for network anomaly detection. In: Proceedings of the 11th Annual Cyber and Information Security Research Conference, p. 15. ACM (2016)
12. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: First steps. Social Networks **5**(2), 109–137 (1983)
13. Kolda, T.G., Pinar, A., Plantenga, T., Seshadhri, C.: A scalable generative graph model with community structure. SIAM Journal on Scientific Computing **36**(5), C424–C452 (2014)
14. LLC, S.R.: Sports Reference College Football Statistics & History. `http://www.sports-reference.com/cfb/`. Used with permission
15. Miller, B.A., Bliss, N.T., Wolfe, P.J., Beard, M.S.: Detection theory for graphs. Lincoln Laboratory Journal **20**(1) (2013)
16. Miller, B.A., Stephens, L.H., Bliss, N.T.: Goodness-of-fit statistics for anomaly detection in Chung-Lu random graphs. In: International Conference on Acoustics, Speech and Signal Processing, pp. 3265–3268. IEEE (2012)
17. Moreno, S., Neville, J.: Network hypothesis testing using mixed kronecker product graph models. In: IEEE 13th International Conference on Data Mining (ICDM), pp. 1163–1168. IEEE (2013)
18. Peel, L., Clauset, A.: Detecting change points in the large-scale structure of evolving networks. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2914–2920. AAAI Press (2015)
19. Satuluri, V., Parthasarathy, S.: Scalable graph clustering using stochastic flows: applications to community discovery. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 737–746. ACM (2009)
20. Seshadhri, C., Kolda, T.G., Pinar, A.: Community structure and scale-free collections of Erdős-Rényi graphs. Physical Review E **85**(5), 056,109 (2012)
21. Van Dongen, S.M.: Graph clustering by flow simulation. Ph.D. thesis, University of Utrecht (2000)
22. Wong, P.C., Foote, H., Mackey, P., Chin, G., Sofia, H., Thomas, J.: A dynamic multiscale magnifying tool for exploring large sparse graphs. Information Visualization **7**(2), 105–117 (2008)