Detecting change points in the large-scale structure of evolving networks

Leto Peel[1,*] and Aaron Clauset[1,2,3,†]

[1] *Department of Computer Science, University of Colorado, Boulder, CO 80309*
[2] *BioFrontiers Institute, University of Colorado, Boulder, CO 80303*
[3] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501*

Networks are an important tool for describing and quantifying data on interactions among objects or people, e.g., online social networks, offline friendship networks, and object-user interaction networks, among others. When interactions are dynamic, their evolving pattern can be represented as a sequence of networks each giving the interactions among a common set of vertices at consecutive points in time. An important task in analyzing such evolving networks, and for predicting their future evolution, is *change-point detection*, in which we identify moments in time across which the large-scale pattern of interactions changes fundamentally. Here, we formalize the network change point detection problem within a probabilistic framework and introduce a method that can reliably solve it in data. This method combines a generalized hierarchical random graph model with a generalized likelihood ratio test to quantitatively determine if, when, and precisely how a change point has occurred. Using synthetic data with known structure, we characterize the difficulty of detecting change points of different types, e.g., groups merging, splitting, forming or fragmenting, and show that this method is more accurate than several alternatives. Applied to two high-resolution evolving social networks, this method identifies a sequence of change points that align with known external shocks to these networks.

I. INTRODUCTION

Relational variables among objects or people are a common form of data, and networks provide a general framework through which to quantify and analyze their patterns. For example, online social interactions, offline friendships, and object-user interactions may all be represented as networks. In many cases, these relations are dynamic, and their evolution over time may be represented as a sequence of networks, each giving the interactions among a common set of vertices at consecutive points in time. Knowledge discovery in such networks thus depends on understanding both the large-scale structure of a network and how that structure changes over time.

A key task in this effort is *change-point detection*, in which we both identify moments in time across which the large-scale pattern of interactions changes fundamentally and quantify what kind and how large a change occurred (Fig. 1). Identifying the timing and shape of such change points divides a network's evolution into contiguous periods of relative structural stability, allowing us to subsequently analyze each period independently, while also providing hints about the underlying processes shaping the data.

For instance, in social networks, change points may be the result of normal periodic behavior, as in the weekly transition from weekdays to weekends. In other cases, change points may result from the collective anticipation of or response to external events or system "shocks". Detecting such changes in social networks could provide a better understanding of patterns of social life and an early detection of social stress caused by, e.g, natural or man-made disasters.

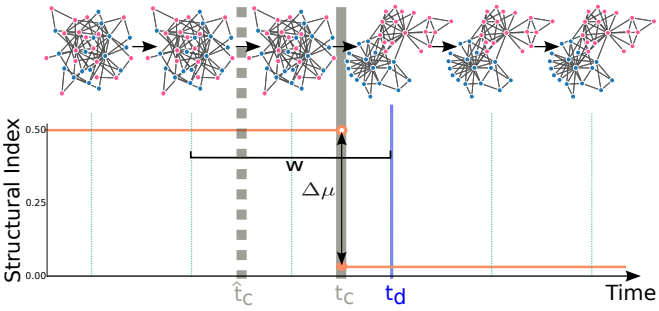

FIG. 1: Schematic of a network change point. A sequence of networks in which vertices divide into two groups at time t_c, represented by a change in an abstract structural index $\Delta\mu$. To detect this change point, we estimate the time of change $\widehat{t_c}$ within a sliding window of the last w networks, and call t_d the time of detection in which $\widehat{t_c}$ is found to be statistically significant.

The central tasks for change-point detection are to choose a model for estimating "norms" from data and a method for identifying when these norms change. When applied to evolving networks, change-point detection requires a method for defining and learning a structural norm from observed interaction data and a method for detecting when these interactions have changed significantly from that norm. To characterize what kind and how large a change occurred, we prefer interpretable models of network structure, so that changes in parameter values have direct meaning with respect to the network's large-scale structure.

Much like change-point detection methods for scalar or vector-valued time series [5], our approach for change-

* leto.peel@colorado.edu
† aaron.clauset@colorado.edu

point detection in networks has three components:

1. select a parametric family of probability distributions appropriate for the data, and a sliding window size w;

2. infer two versions of the model, one representing a change of parameters at a particular point in time within the window, and the one representing the null hypothesis of no change over the entire window; and,

3. conduct a statistical hypothesis test to choose which model, change or no-change, is the better fit.

Here we introduce a change-point detection technique based on generative models of networks, which define parametric probability distributions over graphs. Our particular choice of model is the generalized hierarchical random graph (GHRG), which compactly models nested community structure at all scales in a network. The framework, however, is entirely general, and the GHRG could, in principle, be replaced with another generative network model, e.g., the stochastic block model [16, 24], hierarchical random graph [8], or Kronecker product graph model [19]. To compare the change versus no-change models, we use a generalized likelihood ratio test, with a user-defined parameter specifying a target false-positive rate.

We then show that this approach quantitatively and accurately determines *if* the network has changed, *when* precisely the change occurred, and *how* the network has changed. Specifically, we present a taxonomy of different types and sizes of network change points and a quantitative characterization of the difficulty of detecting them, in synthetic network data with known change points. We then test the method on two real, high-resolution evolving social networks of physical and digital interactions, showing that it accurately recovers the timing of known significant external events.

II. RELATED WORK

Change-point detection in networks is a form of anomaly detection, and within this area, there are two main thrusts: anomalous subgraphs, which focuses on detecting subgraphs that are structured differently from the rest of the network, and temporal anomalies, which focuses on detecting changes in the network's structure over time. Most efforts have focused on detecting anomalous subgraphs, which differ from temporal anomalies in that the latter focuses on changes across networks rather than variations within a single network.

A. Detecting anomalous subgraphs

Early work in detecting anomalies in graphs examined networks containing vertices of different types or labels, and sought to identify unusual patterns of connectivity between the vertex types [11, 23]. Later work used principle eigenvectors of a graph's modularity matrix to detect subgraphs that may have been generated by a different process to the rest of the network [21].

Similar to traditional outlier detection, some efforts have focused on detecting individual anomalous vertices [13] by comparing vertex attributes to the distribution of attributes within a network community. Finally, rather than detecting anomalous vertices, another approach aims to detect anomalous connections by assessing the observed interactions via link prediction models [17].

B. Detecting temporal network anomalies

Detecting temporal anomalies and detecting change points are closely related tasks. On the one hand, an anomaly is an outlier relative to the distribution from which the rest of the data are drawn, while a change point is a shift from one moment to the next in the overall distribution of the data. Thus, a succession of change points away from and then back to some particular distribution can be viewed as as a temporal anomaly.

Along these lines, Ref. [28] presented a method called GraphScope, based on minimum description lengths, to detect when and how communities change in evolving networks. At each time step, this method performs a local search for alternative models that are close to the current community decomposition. As a result, non-smoothness in the underlying community structure score function [14] can prevent the detection of real changes. Other approaches use spectral techniques [4, 15], e.g., using the eigenvectors of the correlation matrix of vertex in-degrees to detect global and local anomalies in the network or by constructing "eigen-behaviors" from the eigenvalues of the correlation matrix of local network measures for each vertex in the network. Although quantitative in nature, these approaches do not provide statistically rigorous determinations of whether a change has occurred.

Statistical hypothesis tests provide a form of this rigor by estimating the probability of seeing a change as large or larger than the one we observe, relative to a model of "normal" behavior. When combined with a specified error rate, this approach provides principled estimates for detecting changes. Ref. [22] recently constructed a statistical hypothesis test for estimating the likelihood that a particular network was generated by a Kronecker product graph model fitted to other data, with good results. However, hypothesis tests based on local [26] and global [20] network measures are more common. The former approach identified local subregions of excessive activity in

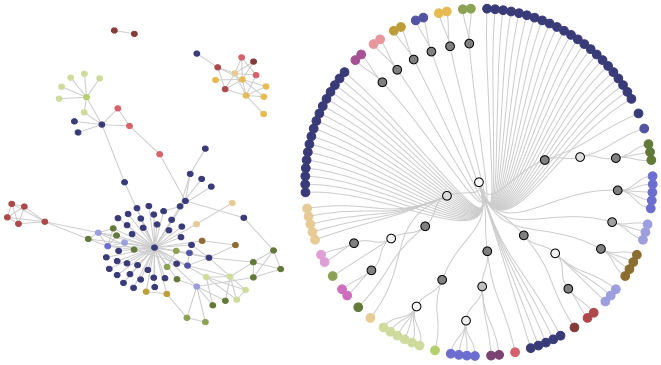

FIG. 2: A snapshot of the Enron email network from October 2001 and its corresponding GHRG dendrogram. In the dendrogram, leaves are vertices in the email network and the tree gives their nested group structure.

networks; however, in practice, this method appears to have low sensitivity, primarily identifying as anomalies events in which vertices suddenly began or ceased all activity. The latter approach employs classic statistical change-point detection methods (specifically cumulative sum [25] and exponentially weighted moving average [27]) to detect overall structural changes. By compressing an entire network into a small number of scalars, however, these methods necessarily discard information that may be crucial to detecting changes.

III. DEFINING A PROBABILITY DISTRIBUTION OVER NETWORKS

Under a probabilistic approach to change-point detection, we must choose a parametric distribution over networks. Here, we introduce the generalized hierarchical random graph (GHRG) model, which has several features that make it attractive for change-point detection and generalizes the popular hierarchical random graph (HRG) model [8]. First, this model naturally captures both assortative and disassortative community structure patterns, models community structure at all scales in the network, and provides accurate and interpretable fits to social, biological and ecological networks. Second, our generalization relaxes the requirement that the dendrogram is a full binary tree, thereby eliminating the HRG's non-identifiability and improving the model's interpretability by for quantifying how a network's structure varies across a change point. Third, we use a Bayesian model of connection probabilities that quantifies our uncertainty about the network's underlying generative model.

The GHRG models a network $G = \{V, E\}$ composed of vertices V and edges $E \subseteq \{V \times V\}$. The model decomposes the N vertices into a series of nested groups, whose relationships are represented by a dendrogram T. Vertices in G are leaves of T, and the probability that two vertices u and v connect in G is given by a parameter p_r located at their lowest common ancestor in T. In the classic HRG model [8], each tree node in T has exactly two subtrees, and p_r gives the density of connections between the vertices in the left and right subtrees. As a result, distinct combinations of dendrograms and probabilities produce identical distributions over networks, producing a non-identifiable model. In the GHRG, we eliminate this possibility by allowing tree nodes to have any number of children. Figure 2 illustrates the GHRG applied to a network of email communications.

Given tree T and set of connection probabilities $\{p_r\}$, the GHRG defines a distribution over networks and a likelihood function

$$p(G \,|\, T, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{N_r - E_r} \quad , \tag{1}$$

where E_r is the number of edges between vertices with common ancestor r and N_r is the total number of possible edges between vertices with common ancestor r:

$$N_r = \sum_{c_i < c_j \in C_r} |c_i||c_j| \quad . \tag{2}$$

By relaxing the binary-tree requirement, the GHRG produces a spectrum of hierarchical structure (Fig. 3). On one end of this spectrum, T contains only a single internal tree node—the root—and every pair of vertices connects with the same probability p_r associated with it, equivalent to the popular Erdős-Rényi random graph model. As more tree nodes and their parameters are added to T, the number of levels of hierarchy increases, allowing the model to capture more varied large-scale patterns. In the limit, T is a full binary tree, and we recover the classic HRG model [8].

IV. LEARNING THE MODEL

Fitting the GHRG model to a network requires a search over all trees on N leaves and the corresponding link probability sets $\{p_r\}$, which we accomplish using Bayesian posterior inference and techniques from phylogenetic tree reconstruction.

Tree structures are not amenable to classic convex optimization techniques, and instead must be searched explicitly. But, searching the space of all non-binary trees is costly. Phylogenetic tree reconstruction faces a similar problem, which is commonly solved by taking a majority "consensus" of a set of sampled binary trees [6]. This consensus procedure selects the set of bipartitions on the leaves that occur in a majority of the sampled binary trees, and each such set denotes a unique non-binary tree containing exactly those divisions. For instance, if every sampled tree is identical, then so too is the consensus tree, while if every sampled tree is a distinct set of bipartitions, the consensus tree has no internal structure. Thus, we estimate T in the GHRG by using a Markov

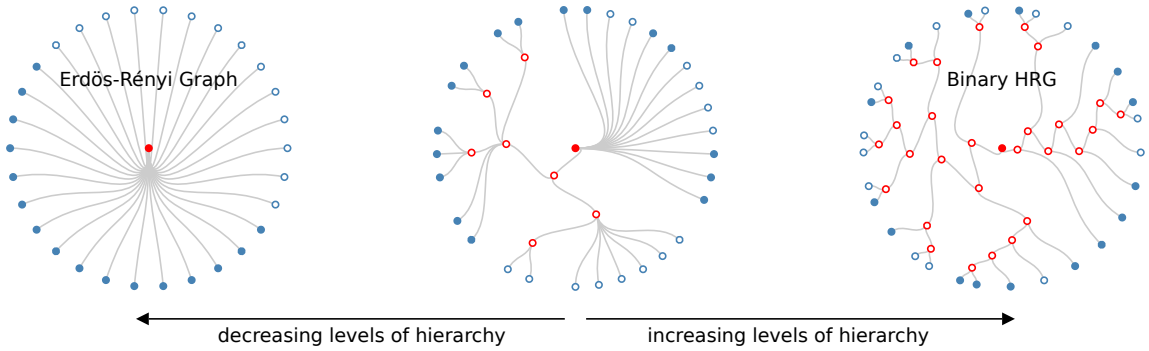

FIG. 3: A spectrum of large-scale structure, corresponding to different amounts of hierarchy in the GHRG, ranging from a simple random graph to a complete hierarchical organization.

Chain Monte Carlo (MCMC) procedure to first sample binary HRG tree structures with probability proportional to their likelihood of generating the observed network. From this set of sampled binary dendrograms, we derive their non-binary majority consensus tree (an approach previously outlined in Ref. [8], but not used to produce a probabilistic model) and assign link probabilities $\{p_r\}$ to the remaining tree nodes.

One approach would be to choose their values via maximum likelihood, setting each

$$\hat{p}_r = E_r / N_r \quad . \tag{3}$$

However, this choice provides little room for uncertainty and is likely to increase our error rate in change-point detection. For instance, consider the case where exactly zero connections $E_r = 0$, or equivalently all connections, $E_r = N_r$, are observed for a particular branch r. Under maximum likelihood, we would set $p_r = 0$ or 1. If a subsequent network has, or lacks, even a single edge whose common ancestor is r, then $E_r > 0$ or $E_r < 1$, and the likelihood given by Eq. (1) drops to 0, an unhelpful outcome.

We mitigate this behavior by assuming Bayesian priors on the p_r values [1]. Now, instead of setting p_r to a point value, we model each p_r as a distribution, which quantifies our uncertainty in its value and prevents its expected value from becoming 0 or 1. For convenience, we employ a Beta distribution with hyperparameters α and β, which act like pseudocounts for the presence or absence of connections respectively. We set these to $\alpha = \beta = 1$, which corresponds to a uniform distribution over the parameters p_r. Because the Beta distribution is conjugate with the Binomial distribution, we may integrate out each of the p_r parameters analytically

$$\begin{aligned} p(G \,|\, T, \alpha, \beta) &= \int p(G \,|\, T, \{p_r\}) p(\{p_r\} \,|\, \alpha, \beta) \, \mathrm{d}\{p_r\} \qquad (4) \\ &= \prod_r \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(E_r+\alpha)\Gamma(N_r - E_r + \beta)}{\Gamma(N_r+\alpha+\beta)} \quad . \end{aligned}$$

Given this formulation, we may update the posterior distribution over the parameter p_r given a sequence of observed networks $\{G_t\}$ by updating the hyperparameters as

$$\tilde{\alpha}_r = \alpha + \sum_{\{G_t\}} E_r^{G_t} \tag{5}$$

$$\tilde{\beta}_r = \beta + \sum_{\{G_t\}} N_r - E_r^{G_t} \quad . \tag{6}$$

Thus, we obtain the posterior hyperparameters from the sum of the prior pseudocounts of edges and the empirically observed edge counts (number of present and absent connections). This Bayesian approach produces an implicit regularization. As the number of observations N_r increases, the posterior distribution becomes increasingly peaked, reflecting a decrease in parameter uncertainty. In the GHRG model, parameters closer to the root of T represent larger-scale structures in G and govern the likelihood of more edges. These parameters are thus estimated with greater certainty, while the distribution over parameters far from the root, which represent small-scale structures, have greater variance, which prevents over-fitting to small-scale structural variations.

V. DETECTING CHANGE POINTS IN NETWORKS

The final piece of change-point detection in evolving networks requires a method to determine whether and when the parameters of our current model of "normal" connectivity have changed. Here, we develop a generalized likelihood ratio test over a sliding window of fixed length w to detect if any changes have occurred with respect to a fitted GHRG model. A change is detected if the log-likelihood ratio versus a null or "no change" model exceeds a threshold determined by a desired false positive rate.

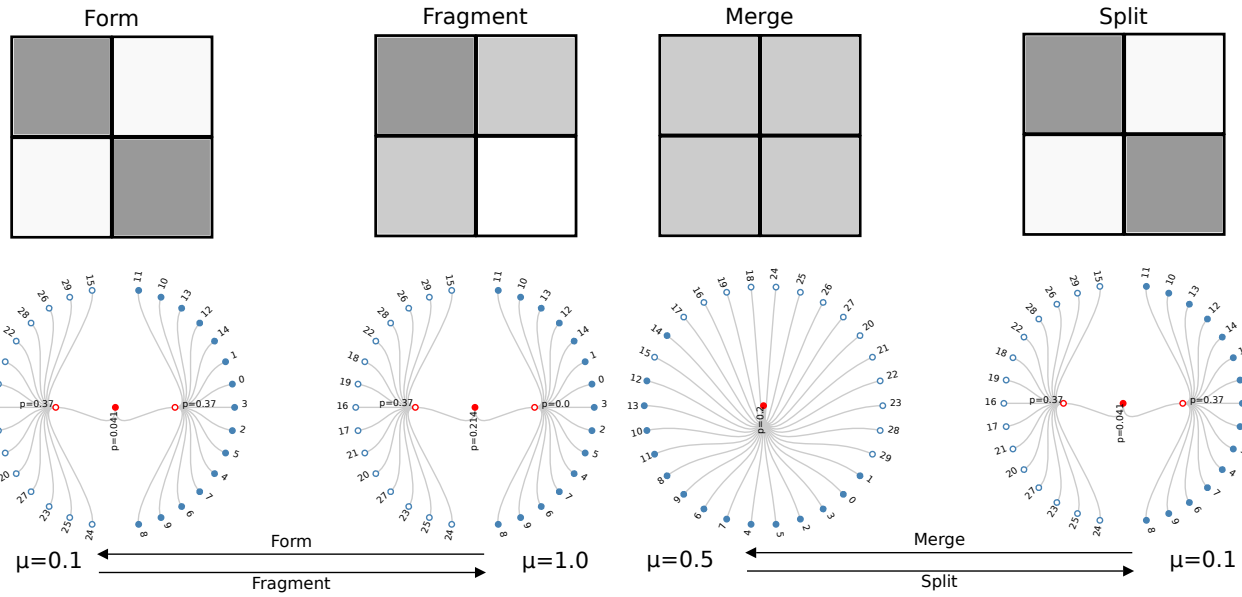

FIG. 4: Taxonomy of change points: formation versus fragmentation and merging versus splitting. In each case, we show both the block structure of the adjacency matrix for two groups and the corresponding GHRG model. For our experiments, we switch from one structure to the other by changing the structural index μ. See text for more detail.

A. Generalized likelihood ratio test

Using a likelihood ratio test, we compare the observed data's likelihood under two different models: a null hypothesis model H_0, in which no change occurs, and an alternative hypothesis model H_1, in which a change occurs at some particular time t_c. Since we use network snapshots at regular intervals, we assume t_c occurs between some pair of snapshots, which we indicate using a 0.5 offset (Fig. 1).

We restrict our consideration of change points to those within a sliding window of w networks, the last of which is at the "current" time τ. For the no-change model, we say that all networks within the window were drawn from a model with parameters $\psi^{(\emptyset)}$. For the change model, we let $\psi^{(0)}$ denote the model parameters for networks up to t_c within our window and $\psi^{(1)}$ the parameters for networks after t_c, but still within the window. Rewriting the change and no-change hypotheses in terms of such a shift in a parametric distribution over graphs at t_c, we have

$$\begin{aligned} H_0 &: \psi^{(\emptyset)} = \psi^{(\emptyset)} \qquad \text{(no change)} \\ H_1 &: \psi^{(0)} \neq \psi^{(1)} \qquad \text{(change)} \ . \end{aligned}$$

These change and no-change hypotheses are composite, meaning that the parameters ψ can take on any permissible value, so long as they meet the definitions of H_0 and H_1 given above. To test such hypotheses, we must use a generalized likelihood ratio (GLR) test.

The GLR's test statistic is the log-likelihood ratio, which compares the likelihood of the w networks under a model with a change point at some time t_c versus the no-change model, which we denote Λ_{t_c},

$$\Lambda_{t_c} = \sup_{\widehat{\psi}^{(0)},\widehat{\psi}^{(1)},\widehat{\psi}^{(\emptyset)} \in \Omega} \log \frac{L\left(\widehat{\psi}^{(0)}, \widehat{\psi}^{(1)}\right)}{L\left(\widehat{\psi}^{(\emptyset)}\right)} \ , \tag{7}$$

where Ω is the set of permissible parameter values. In our Bayesian formulation, rather than set the parameters that maximize the likelihood, we set the prior parameters to their posterior values in accordance with Eqs. (5) and (6) and integrate over the parameters [2].[1]

Finally, the time at which the change occurs t_c is itself an unknown value, and must be estimated. We make the conservative choice, choosing $\widehat{t}_c$ as the time point between a pair of consecutive networks that maximizes our test statistic Λ across the window. Letting g_τ for a given window of w networks ending at τ be that largest value

$$g_\tau = \max_{\tau - w + 1 < \widehat{t}_c < \tau} \Lambda_{\widehat{t}_c} \ , \tag{8}$$

we then say that the time of detection t_d is the first time point τ at which g_τ exceeds a threshold h (called a "stopping rule" in the change-point detection literature)

$$t_d = \min\{\tau : g_\tau > h\} \ . \tag{9}$$

[1] This ratio of marginal likelihoods is called a *posterior Bayes factor* and may be interpreted as a likelihood ratio for two simple hypotheses [3].

Using $\widetilde{\psi} = \{\tilde{\alpha}_r, \tilde{\beta}_r\}$ to denote the set of posterior hyperparameters, the GHRG's generalized likelihood ratio is

$$\Lambda_{\widehat{t}_c} = \log \frac{\prod_{t=\tau-w+1}^{\widehat{t}_c-0.5} p(G_t \,|\, T_\tau, \widetilde{\psi}^{(0)}_{\widehat{t}_c}) \prod_{t=\widehat{t}_c+0.5}^{\tau} p(G_t \,|\, T_\tau, \widetilde{\psi}^{(1)}_{\widehat{t}_c})}{\prod_{t=\tau-w+1}^{\tau} p(G_t \,|\, T_\tau, \widetilde{\psi}^{(\emptyset)})} \quad ,$$

where $\widetilde{\psi}^{(\emptyset)}$ is the set of posterior hyperparameters pertaining to the no-change hypothesis (no change point anywhere in the window of w networks), while $\widetilde{\psi}^{(0)}_{\widehat{t}_c}$ and $\widetilde{\psi}^{(1)}_{\widehat{t}_c}$ are the hyperparameters for the networks up to and then following the point $\widehat{t}_c$, respectively. These hyperparameters are estimated from the data using Eqs. (5) and (6).

B. Parametric Bootstrapping

The choice of threshold h, which g_τ must exceed for a detection to occur, sets the method's resulting false positive rate and the distribution of g_τ under the null model. Recent results on model comparison for statistical models of networks, and specifically the stochastic block model, of which the GHRG is a special case, suggest that for technical reasons the null distribution can deviate substantially from the χ^2 distribution [29]. To avoid a misspecified test, we estimate the null distribution numerically, via Monte Carlo samples from a parametric bootstrap distribution [12] defined by the GHRG for the no-change model. In this way, we estimate the null distribution exactly, rather than via a possibly misspecified approximation.

For each network we sample from the no-change GHRG model, we calculate g_τ from Eq. (8) to obtain its distribution under the hypothesis of no change (see Algorithm 1). Using the sampled distribution, the threshold h may then be chosen so that $p(g_\tau > h) = p_{\text{fp}}$ is the desired false positive rate. In practice we do this by calculating a p-value for the test case by counting the proportion of likelihood ratios in our null distribution that are higher than our test statistic, g_τ:

$$p\text{-value} = \frac{|\{g_\tau\}_{\text{null}} > g_\tau|}{|\{g_\tau\}_{\text{null}}|} \tag{10}$$

Thus, if we find a p-value below the chosen threshold, we say a change is detected, and when the no-change model is correct, we are wrong no more than p_{fp} of the time.

ALGORITHM 1: Parametric bootstrap sampling g_τ from the null model distribution

Input $\mathcal{G} = \{G\}_{\tau-w+1}^{\tau}$

$\{g_\tau\}_{\text{null}} = \emptyset$.
$\text{GHRG}(T_\tau, \widetilde{\psi}_\tau) = \text{fitGHRG}(\mathcal{G})$.
for $i = 1$ to 1000 **do**
  sample w graphs, $\{G_t\}_{t=1}^{w} \sim \text{GHRG}(T_\tau, \widetilde{\psi}_\tau)$.
  for $\tilde{t}_c = 1$ to w **do**
    calculate $\widetilde{\psi}^{\emptyset}, \widetilde{\psi}^{0}, \widetilde{\psi}^{1}$ according to Eq. (5) and Eq. (6).
    $\Lambda_{\tilde{t}_c} = \sum_{t=1}^{\tilde{t}_c-1} \log p\left(G_t | T_\tau, \widetilde{\psi}^{(0)}_{\tilde{t}_c}\right)$
    $+ \sum_{t=\tilde{t}_c}^{w} \log p\left(G_t | T_\tau, \widetilde{\psi}^{(1)}_{\tilde{t}_c}\right)$
    $- \sum_{t=1}^{w} \log p\left(G_t | T_\tau, \widetilde{\psi}^{\emptyset}\right)$
  end for
  $g_\tau^{(i)} = \max_{\tilde{t}_c} \Lambda_{\tilde{t}_c}$.
  $\{g_\tau\}_{\text{null}} = \{g_\tau\}_{\text{null}} + g_\tau^{(i)}$.
end for

VI. DETECTABILITY OF CHANGE POINTS

Before applying our method to empirical data with unknown structure and unknown change points, we first characterize the detectability of different types of network change points under controlled circumstances on synthetic data, generated using our GHRG model, with known structure and changes.

The following change-point types constitute difficult but realistic tests; specifically, we choose a modest-sized network of $N = 30$ and a sparse and constant expected number of connections (marginal link probability of 0.2). Furthermore, we define four general types of change points: *splitting*, when one large community divides in two; *merging*, when two communities combine (the time-reversal of splitting); *formation*, when one of two groups of vertices add edges to make a community; and *fragmentation*, when one of two groups loses all its edges (the time-reversal of formation). Figure 4 illustrates these types of change points.

To provide a single parameter that controls the switching between these distinct states, we define a single structural index μ as

$$\mu = \frac{p_{\text{out}}}{p_{\text{in}} + p_{\text{out}}} \quad . \tag{11}$$

For the merge/split change points, we choose the merged state to be $\mu = 0.5$, which produces a single community in which every edge occurs with the same probability $p_{\text{in}} = p_{\text{out}}$. In the split state, the network is comprised of two distinct communities. If $\mu < 0.5$ then the communities are assortative ($p_{\text{in}} > p_{\text{out}}$, and vertices prefer to connect within the community) and disassortative if $\mu > 0.5$ ($p_{\text{in}} < p_{\text{out}}$, and vertices prefer to connect across communities).

For formation/fragmentation change points, we use the same two-community model, but now fix the link probability within one community and use μ to describe relationship between the p_{in} and p_{out} of the second community.

We now summarize these change points with respect

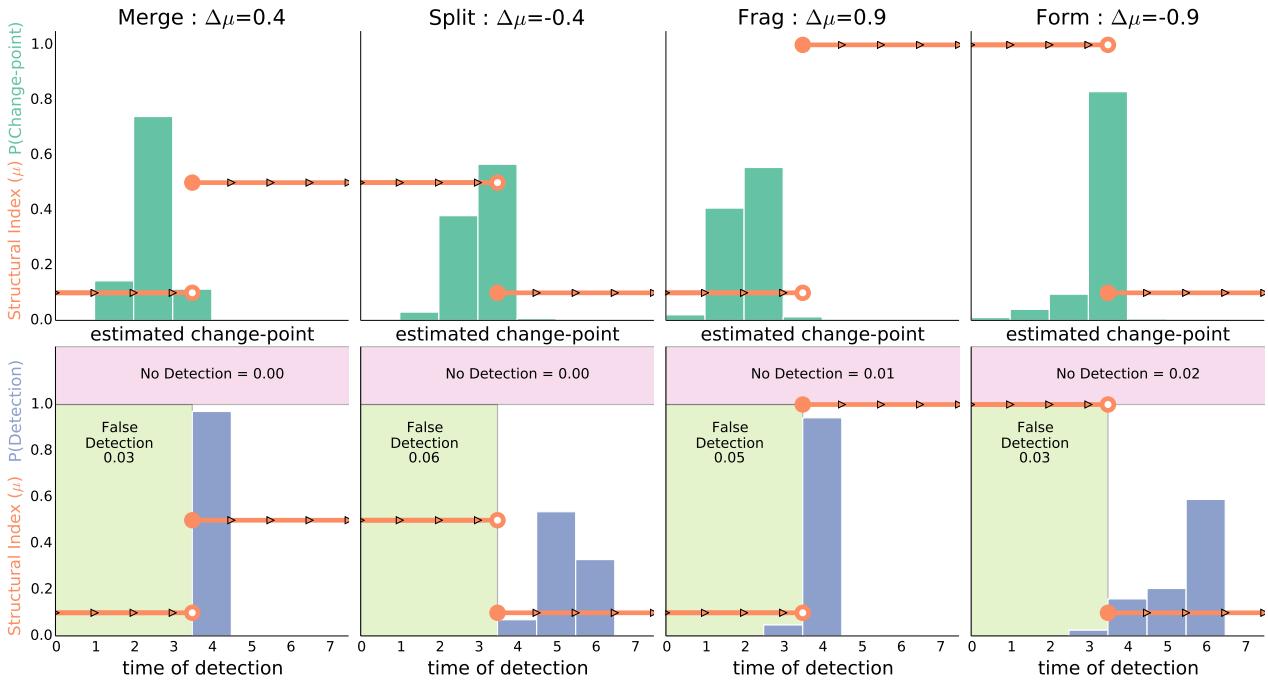

FIG. 5: Results for merge, split, fragment and form change points, where the size of the change is parameterized by the structural index μ (orange line) and a discontinuity occurs at the true point of change. The resulting distributions of (*top row*) estimated change points times $\widehat{t}_c$ (green bars), and (*bottom row*) detection times t_d (blue bars), for each change type. The 0.5 time offset indicates that the estimated change point time $\widehat{t}_c$ occurs between a pair of consecutive network snapshots, while t_d gives the time of the last network in the window when a detection event occurs. False detections (false positives) occur when $t_d < t_c$; no detection (false negatives) occur when $\tau - w + 1 > t_c$ without a detection.

to μ:

$$\begin{array}{ll} \text{merge} & \mu \neq 0.5 \rightarrow \mu = 0.5 \\ \text{split} & \mu = 0.5 \rightarrow \mu \neq 0.5 \\ \text{fragment} & \mu < 1 \quad \rightarrow \mu = 1 \\ \text{form} & \mu = 1 \quad \rightarrow \mu < 1 \ . \end{array}$$

In all of the tests a window size of $w = 4$ and a false-positive rate of 0.05 was used. For comparison we use three simple methods that use the same probabilistic framework as our method but instead parameterize the distribution over graphs using a univariate Gaussian to model a network summary statistic (mean degree, mean geodesic distance, mean local clustering coefficient).

In Figure 5 we show for each of the change types, two different distributions (over 100 runs): the estimated change point $\widehat{t}_c$ and the time of detection t_d (see Figure 1). We find that the estimated change points tend to either be correct or slightly early. The time of detection (the end of the sliding window when a change is detected) quantifies how many networks after the change we must see before we identify the change point. We find that the merge and fragment changes are quick to detect, but the change points are often estimated early. In contrast, the split and formation changes take longer to detect, but the estimation of the change points themselves is more accurate.

In Figure 6 we compare the false positive and false negative error rates among all four methods. On false positives, all methods are close to 0.05, which matches the desired false alarm rate. However, the false negative rates differ widely, with the simple methods performing terribly in nearly every case, even when the size of the change is large. In contrast, our method performs well across all four tests, except when the size of the change is very small, e.g., when $\Delta\mu \approx 0$, which represents the hardest cases, where small-sample fluctuations obscure much of the actual change.

VII. CHANGE POINTS IN REAL NETWORKS

We now apply these approaches[2] to detect changes in two high-resolution evolving networks, the MIT Reality Mining proximity network[3] [10] and the Enron email network[4] [18], for which a set of external "shocks" exists that serve as targets for change-point detection. Both data sets are evolving networks of human social interactions,

[2] Code for our change-point detection method is available at `http://gdriv.es/letopeel/code.html`

[3] `realitycommons.media.mit.edu/realitymining.html`

[4] `www.cs.cmu.edu/~enron/`

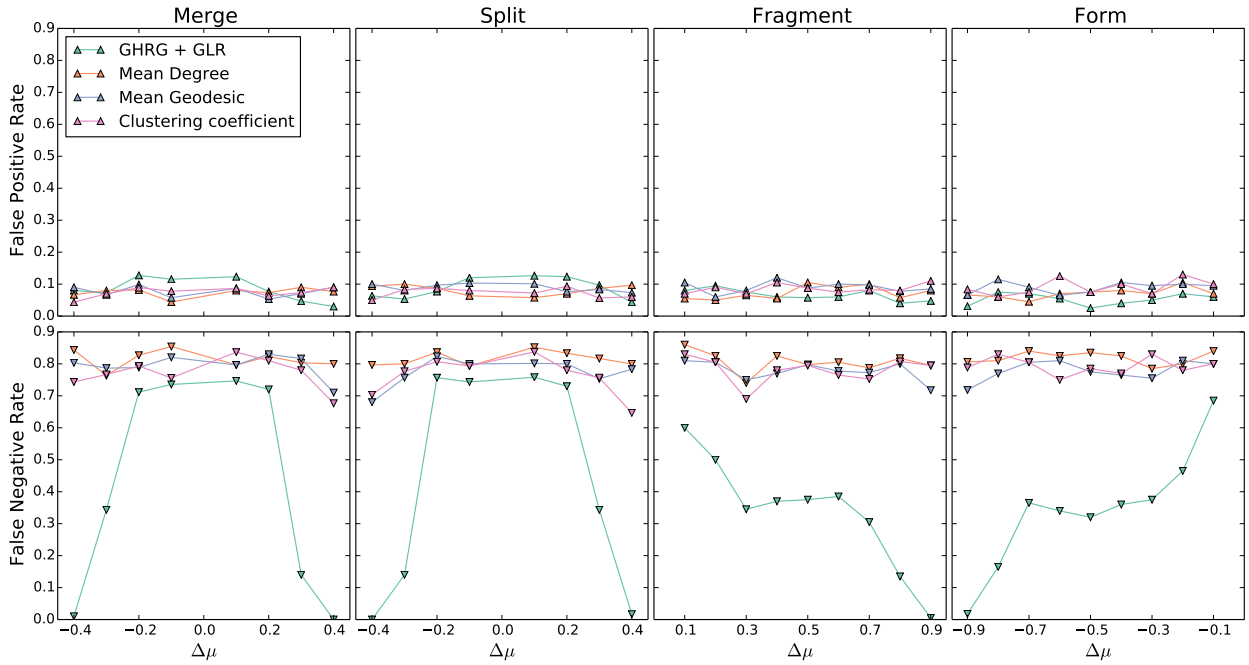

FIG. 6: False positive (*top*) and false negative (*bottom*) error rates for our method and the simple methods on the four different change types for different magnitudes of change($\Delta\mu$).

but represent different interaction types. In the MIT proximity networks, two vertices (people) are connected whenever they were in close physical proximity, while in the Enron networks, two vertices (email accounts) are connected if one emailed the other.

A. Social proximity network

The MIT network is comprised of proximity data for 97 faculty and graduate students, recorded continuously via Bluetooth scans from their mobile phone over 35 weeks [10]. From the raw scan data, we extracted a sequence of weekly networks, in which an edge denotes physical proximity to one of the 97 subjects at some point that week.

For each detection method, the GHRG and the three simple methods, we used a window size $w = 4$, the same as in the synthetic experiments. The results for each method are shown in Figure 7 (top), which includes the identified change points for each method, the time series of network measures, a selection of inferred GHRG dendrograms, and highlighted external events.

Examining the simple methods' results, we observe that they exhibit low sensitivity relative to the known external events [9]. In particular, they do pick out several real change points, including Winter break (mean geodesic distance), the beginning (mean degree) and end (clustering coefficient) of the independent activities period. However, they also miss the majority of other events, and there is little consistency across these methods (with the exception of the beginning of Spring break and the beginning of Sponsor week). Thus, these techniques seem both unreliable and inconsistent.

In contrast, the GHRG method identifies nearly all of the known external events, along with a few additional change points, e.g., one week before and one week after Sponsor week. This fact agrees well with the social dynamics of Sponsor week, an event involving 75 of the subjects and which typically shifts work schedules dramatically as they seek to meet deadlines and project goals [9].

Additionally, the GHRG method finds more change points in the Fall semester than in the Spring. Examining the dendrograms themselves, we find that the changes in the inferred structures in the Fall are much more dramatic than in the Spring; see Figure 7 (top). This agrees with the fact that 35 of the subjects were new students in the Fall semester and thus still establishing their social patterns [9]. By the Spring semester, these patterns had largely stabilized, and the large perturbation of Sponsor week was absent. Overall, the GHRG both recovers known events, highlights additional changes, and provides an interpretable basis for discovering new patterns within this evolving network.

B. Enron email network

The Enron network is comprised of emails among 151 users, mostly senior management of the Enron energy company. These data were made public by the Federal Energy Regulatory Commission during its investigation

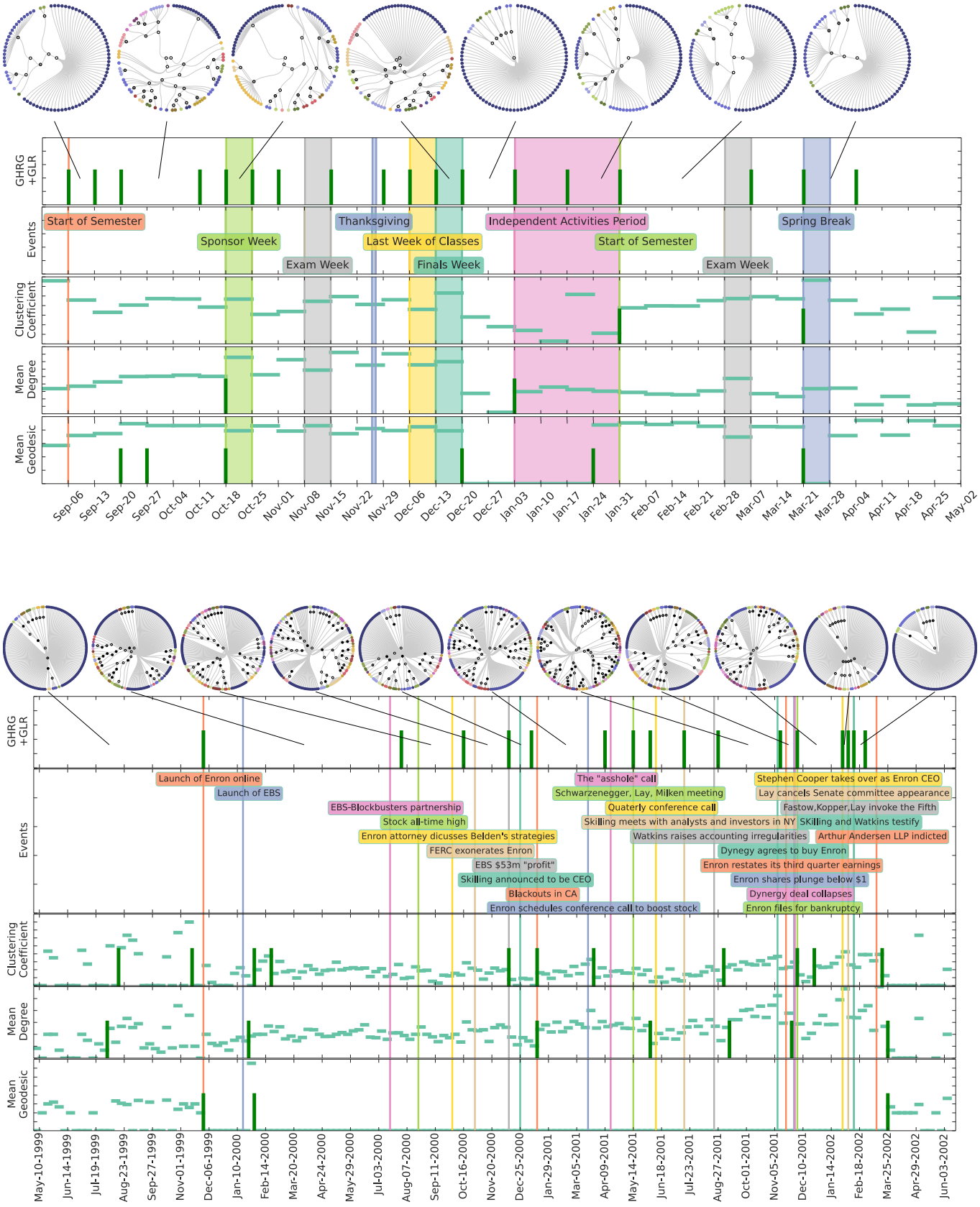

FIG. 7: Change points detected in the MIT proximity (*top*) and Enron email (*bottom*) networks. Each figure shows, from top to bottom, the structure of the inferred GHRG models, the change points detected (green bars) using our method and the associated known events. The bottom three rows show the change points detected with the simple methods along with the values of the relevant summary statistic.

after the company's collapse. Using a cleaned version of these data [18], we applied both the GHRG and simple methods to weekly snapshots from May 1999 to June 2002. The results for each method are shown in Figure 7 (bottom).

Because this network sequence is very long, we examined the impact of varying window size, choosing $w = \{4, 8, 16\}$. Results across window sizes were highly consistent, although larger values produced additional change points. This suggests that window size may operate like a temporal resolution parameter, with longer windows giving more resolution.

As with the MIT network, we again find that the simple methods perform poorly, recovering only a small fraction of known events compared with the GHRG method. Furthermore, the frequency of change points identified by the GHRG method increases as time passes, reflecting the intensification of the Enron scandal. Examining the GHRG change points and the list of external events, we find that the identified change points correlate well with key meetings and events such share price fluctuations. Examining the inferred dendrograms, we find that a particularly large structural change occurred around the launch of Enron online. Before its launch, the network's structure is very sparse, while after launch, the number of levels in the GHRG model increase dramatically reflecting the formation of many communities. We also see that the structure and density of the networks increase over the period starting immediately after the Californian blackouts through until Stephen Cooper takes over as CEO and the collapse of the company is imminent.

VIII. DISCUSSION

When analyzing a sequence of time evolving networks, a central goal is to understand how the network's structure has changed over time, and how it might change in the future. Change-point detection provides a principled approach to this problem, by decomposing the networks sequence into subsequences of distinct but relatively stable structural patterns (Fig. 1). Formalizing this problem within a probabilistic framework, we developed a statistically principled method, based on generative models and hypothesis tests, that can detect if, when and how such change points occur in the large-scale patterns of interactions. Under our framework, change points occur when the shape of an estimated probability distribution over networks changes significantly.

Not all such change points are equally easy to detect, however. Using synthetic data with known structure and known change points, we found that changes associated with two communities merging or with one of several communities losing its internal connections ("fragmentation") were more difficult to accurately detect than those associated with one community splitting in two or with many singletons connecting to form a new community ("formation"). This asymmetry in the detectability of different types of network changes begs the question of whether more sophisticated techniques can eliminate these differences, and whether adding auxiliary information like edge weights or vertex attributes makes this problem easier or harder.

That being said, change-point methods based on network measures like the mean degree, clustering coefficient, or mean geodesic path length performed poorly, yielding high false negative rates even for large structural changes (Fig. 6). This poor performance is likely the result of network measures discarding much of the specific information that generative models utilize. Applied to two high-resolution evolving social networks, our method provided very good results, recovering the timing, from network data alone, of many more known external "shock" events than the network-measure methods (Fig. 7).

A notable assumption of our method is that each observed network is an iid sample from some underlying distribution. Real evolving networks likely violate this assumption, as a result of underlying non-stationary or periodic dynamics. When each network spans more time than the natural time scale of network dynamics [7], the iid assumption may be reasonable, and our empirical results support this notion. An interesting direction for future work would relax the iid assumption perhaps using Markov models.

Although the generalized hierarchical random graph model yielded good results, in principle, any generative model could be used in its place, e.g., the stochastic block model [1, 16, 24] or the Kronecker product graph model [19]. Similarly, the recent work in graph hypothesis testing [22] could potentially be adapted to the change-point detection problem. Two key features of the generalized hierarchical random graph model (GHRG) for change-point detection, however, are its interpretability and the way it naturally adapts its dendrogram structure to fit the network, adding or removing levels in the hierarchy, as the network evolves. Combined with the strong results on synthetic and real-world data, this approach to change-point detection promises to have broad application, perhaps particularly in social networks, where interpretability provides a crucial bridge to testing hypotheses about the underlying social dynamics driving network evolution.

IX. ACKNOWLEDGEMENTS

We thank Dan Larremore for helpful conversations, and acknowledge support from Grant #FA9550-12-1-0432 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA).

[1] C. Aicher, A. Z. Jacobs, and A. Clauset. Adapting the stochastic block model to edge-weighted networks. In *ICML Workshop on Structured Learning*, 2013. arxiv:1305.5782.

[2] M. Aitkin. Posterior bayes factors (with discussion). *Journal of the Royal Statistical Society*, 53:111–142, 1991.

[3] M. Aitkin. The calibration of p-values, posterior bayes factors and the aic from the posterior distribution of the likelihood. *Statistics and Computing*, 7(4):253–261, 1997.

[4] L. Akoglu and C. Faloutsos. Event detection in time series of mobile communication graphs. In *27th Army Science Conference*, Orlando, FL, USA, 2010.

[5] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[6] D. Bryant. A classification of consensus methods for phylogenetics. In *Bioconsensus*, volume 61 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 163–183. Amer. Math. Soc., Providence, RI, 2003.

[7] A. Clauset and N. Eagle. Persistence and periodicity in a dynamic proximity network. In *DIMACS Workshop on Computational Methods for Dynamic Interaction Networks*, 2007. arxiv:1211.7343.

[8] A. Clauset, C. Moore, and M. E. J. Newman. Structural inference of hierarchies in networks. In E. Airoldi, D. M. Blei, S. E. Fienberg, A. Goldenberg, E. P. Xing, and A. X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, volume 4503 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2007.

[9] N. Eagle. *Machine perception and learning of complex social systems*. Department of media arts and sciences, Massachusetts Institute of Technology, 2005.

[10] N. Eagle and A. (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, Mar. 2006.

[11] W. Eberle and L. Holder. Discovering structural anomalies in graph-based data. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ICDMW '07, pages 393–398, Washington, DC, USA, 2007. IEEE Computer Society.

[12] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, 1993.

[13] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 813–822, New York, NY, USA, 2010. ACM.

[14] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Phys. Rev. E*, 81:046106, 2010.

[15] S. Hirose, K. Yamanishi, T. Nakata, and R. Fujimaki. Network anomaly detection based on eigen equation compression. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1185–1194, New York, NY, USA, 2009. ACM.

[16] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[17] Z. Huang and D. Zeng. A link prediction approach to anomalous email detection. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, volume 2, pages 1131–1136, Oct 2006.

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

[19] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD'05, pages 133–145, Berlin, Heidelberg, 2005. Springer-Verlag.

[20] I. McCulloh and K. M. Carley. Detecting change in longitudinal social networks. *Journal of Social Structure*, 12, 2011.

[21] B. Miller, N. Bliss, and P. Wolfe. Subgraph detection using eigenvector l1 norms. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1633–1641. 2010.

[22] S. Moreno and J. Neville. Network hypothesis testing using mixed kronecker product graph models. In *13th IEEE International Conference on Data Mining*, 2013.

[23] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 631–636, New York, NY, USA, 2003. ACM.

[24] K. Nowicki and T. A. B. Snijders. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455), 2001.

[25] E. S. Page. Cumulative sum charts. *Technometrics*, 3(1):1–9, 1961.

[26] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park. Scan statistics on enron graphs. *Comput. Math. Organ. Theory*, 11(3):229–247, Oct. 2005.

[27] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1(3):239–250, 1959.

[28] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: Parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. ACM.

[29] X. Yan, J. E. Jensen, F. Krzakala, C. Moore, C. R. Shalizi, L. Zdeborová, P. Zhang, and Y. Zhu. Model selection for degree-corrected block models. *arXiv*, 1207.3994, 2012.