Detecting Changes in Dynamic Events Over Networks

Shuang Li, Yao Xie, Mehrdad Farajtabar, Apurv Verma, and Le Song

***Abstract*—Large volumes of networked streaming event data are becoming increasingly available in a wide variety of applications such as social network analysis, Internet traffic monitoring, and health care analytics. Streaming event data are discrete observations occurring in continuous time, and the precise time interval between two events carries substantial information about the dynamics of the underlying systems. How does one promptly detect changes in these dynamic systems using these streaming event data? In this paper, we propose a novel change-point detection framework for multidimensional event data over networks. We cast the problem into a sequential hypothesis test, and we derive the likelihood ratios for point processes, which are computed efficiently via an expectation-maximization (EM) like algorithm that is parameter free and can be computed in a distributed manner. We derive a highly accurate theoretical characterization of the false-alarm rate, and we show that the method can provide weak signal detection by aggregating local statistics over time and networks. Finally, we demonstrate the good performance of our algorithm on numerical examples and real-world datasets from Twitter and Memetracker.**

***Index Terms*—Change-point detection for event data, Hawkes process, online detection algorithm, false alarm control.**

I. Introduction

NETWORKS have become a convenient tool for people to efficiently disseminate, exchange and search for information. Recent attacks on very popular web sites, such as Yahoo and eBay [1], leading to a disruption of services to users, have triggered increased interest in network anomaly detection. On the positive side, the surge of hot topics and breaking news can provide business opportunities. Therefore, the *early detection* of changes, such as anomalies, epidemic outbreaks, hot topics, or new trends, among streams of data from networked entities is a very important task and has been attracting significant interest [1]–[3].

Manuscript received September 15, 2016; revised August 4, 2016, March 7, 2017, and December 23, 2016; accepted April 7, 2017. Date of publication April 24, 2017; date of current version May 17, 2017. The work of Y. Xie was supported in part by CMMI-1538746 and CCF-1442635, and the work of L. Song was supported by Intel and NVIDIA under Grants NSF/NIH BIGDATA 1R01GM108341, ONR N00014-15-1-2340, NSF IIS-1218749, NSF IIS-1639792, and NSF CAREER IIS-1350983. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Wang. (*Corresponding author: Yao Xie*.)

S. Li and Y. Xie are with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: sli370@gatech.edu; yao.xie@isye.gatech.edu).

M. Farajtabar, A. Verma, and L. Song are with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: mehrdad@gatech.edu; apurvverma@gatech.edu; lsong@cc.gatech.edu).

This paper has supplemental downloadable multimedia material available at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSIPN.2017.2696264

All types of the above-mentioned changes can be more concretely formulated as the changes in time interval distributions between events, combined with the alteration of interaction structures across components in networks. However, change-point detection based on event data occurring over a network topology is nontrivial. Apart from the possible temporal dependency of the event data as well as the complex cross-dimensional dependence among components in a network, event data from networked entities are usually not synchronized in time. Because they are dynamic in nature, much of the collected data are discrete events observed irregularly in continuous time [4], [5]. The precise time interval between two events is random and carries substantial information about the dynamics of the underlying systems. These characteristics make such event data fundamentally different from independently and identically distributed (*i.i.d.*) data and time-series data where time and space are treated as indices rather than random variables (see Fig. 1 for further illustrations of the distinctive nature of event data vs. *i.i.d.* and time-series data). Clearly, the *i.i.d.* assumption cannot capture the temporal dependency between data points, while time-series models require us to discretize the time axis and aggregate the observed events into bins (such as in the approach in [6] for neural spike train change detection). If this approach is applied, it is unclear how one can choose the size of the bin and how to best address the case where there is no event within a bin.

In addition to the distinctive temporal and spatial aspects, there are three additional challenges when using event data over networks: (*i*) how to detect weak changes, (*ii*) how to update the statistics efficiently online, and (*iii*) how to provide a theoretical characterization of the false-alarm rate for the statistics. To address the first challenge, many approaches use random or ad-hoc aggregations, which may not pool data efficiently or may lose statistical power when detecting weak signals. The occurrence of change points (e.g., epidemic outbreaks and hot topics) over networks usually evidences a certain clustering behavior over dimensions and tend to synchronize in time. Smart aggregation over dimensions and time horizons would improve the strengths of signals and allow changes to be detected quicker [7]. To address the second challenge, many change-point detection methods based on likelihood ratio statistics do not consider computational complexity nor can be computed in a distributed manner and, hence, are not scalable to large networks. Temporal events can arrive at social platforms in very high volumes and at very high velocities. For instance, every day, on average, approximately 500 million tweets are tweeted on

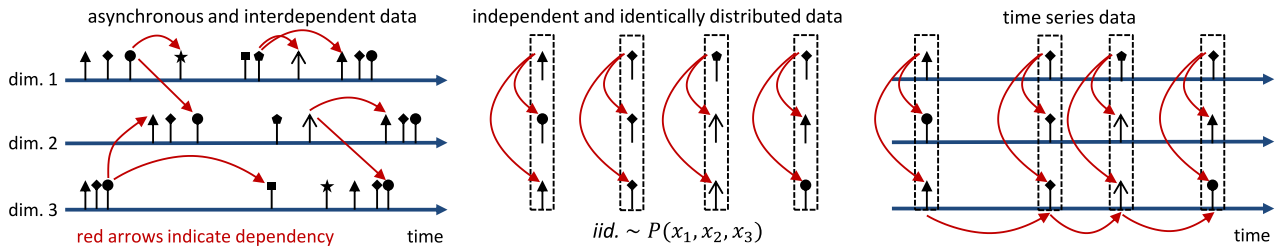

Fig. 1. Asynchronously and interdependently generated high-dimensional event data are fundamentally different from *i.i.d.* and time-series data. First, observations for each dimension can be collected at different time points. Second, there can be temporal dependences as well as cross-dimensional dependences. In contrast, the dimensions of *i.i.d.* and time-series data are sampled at the same time point, and in the figure, different marks indicate potentially different values or features of an observation.

Twitter [8]. There is a great need for the development of efficient algorithms for updating the detection statistics online. To address the third challenge, it is usually very difficult to control false alarms for change-point detection statistics over a large network. When applied to real network data, traditional detection approaches usually have a high false-alarm rate [1]. This would lead to a huge waste of resources because, every time a change point is declared, subsequent diagnoses are needed. Lacking an accurate theoretical characterization of false alarms, existing approaches usually have to perform expensive Monte Carlo simulations to determine the false alarms and are prohibitive for large networks.

Our contributions: In this paper, we present a novel online change-point detection framework tailored to multi-dimensional intertwined event data streams over networks (or conceptual networks), therein addressing the above challenges. We formulate the problem by leveraging the mathematical framework of sequential hypothesis testing and point process modeling, where before the change, the event stream follows one point process, and after the change, the event stream becomes a different point process. Our goal is to detect such changes *as quickly as possible* after the occurrences. We derive generalized likelihood ratio statistics, and we present an efficient EM-like algorithm to compute the statistics online with streaming data. The EM-like algorithm is parameter free and can be implemented in a distributed manner; hence, it is suitable for large networks.

Specifically, our contributions include the following:

i) We present a new sequential hypothesis test and likelihood ratio approach for detecting changes for event data streams *over networks*. We will use either the Poisson process as the null distribution to detect the appearance of temporal independence or the Hawkes process as the null distribution to detect the possible alteration of the dependency structure. For the (inhomogeneous) Poisson process, time intervals between events are assumed to be independent and exponentially distributed. For the Hawkes process, the occurrence intensity of events depends on the events that have occurred, which implies that the time intervals between events would be correlated. Therefore, the Hawkes process can be thought of as a special autoregressive process in time, and the multivariate Hawkes process also provides a flexible model for capturing cross-dimension dependencies in addition to temporal dependencies. Our model explicitly captures the information diffusion (and dependencies) both over networks and time, and it allows us to aggregate information for weak signal detection. Our proposed detection framework is quite general and can be easily adapted to other point processes.

In contrast, existing work on change-point detection for point processes has also been focused on a single stream rather than the multidimensional case with networks. These works include detecting changes in the intensity of a Poisson process [9]–[11] and the coefficients of continuous diffusion processes [12]; detecting changes using the self-exciting Hawkes processes, including trend detection in social networks [13]; and detecting Poisson processes using a score statistic [14].

ii) We present an efficient expectation-maximization (EM)-like algorithm for updating the likelihood-ratio detection statistic online. The algorithm can be implemented in a *distributed* manner due to its structure: only neighboring nodes need to exchange information for the E-step and M-step.

iii) We also present an accurate theoretical approximation of the false-alarm rate (formally the average-run length or ARL) of the detection algorithm via the recently developed change-of-measure approach to handle highly correlated statistics. Our theoretical approximation can be used to determine the threshold in the algorithm accurately.

iv) Finally, we demonstrate the performance gain of our algorithm over two baseline algorithms (which ignore the temporal correlation and correlation between nodes) using synthetic experiments and real-world data. These two baseline algorithms represent current approaches for processing event stream data. We also show that our algorithm is very sensitive to true changes, and the theoretical false-alarm rates are very accurate compared to the experimental results.

Related work: Recently, there has been a surge of interest in using multidimensional point processes for modeling dynamic event data over networks. However, most of these works focus on modeling and inference of the point processes over networks. Related works include modeling and learning bursty dynamics [5], shaping social activity by incentivization [15], learning information diffusion networks [4], inferring causality [16], learning mutually exciting processes for viral diffusion [17], learning triggering kernels for multi-dimensional Hawkes processes [18] in networks where each dimension is a Poisson process [19], learning latent network structures for general counting processes [20], tracking parameters of dynamic point process networks [21], and estimating point process models for

the co-evolution of network structures and information diffusion [22], just to name a few. These works provide a wealth of tools through which we can, to some extent, keep track of the network dynamics if the model parameters can be sequentially updated. However, when only given the values of the up-to-date model parameters, especially in high-dimensional networks, it remains unclear how one can perform change detection based on these models in a principled manner.

Classical statistical sequential analysis (see, e.g., [23], [24]), where one monitors *i.i.d.* univariate and low-dimensional multivariate observations from a single data stream, is a well-developed area. Outstanding contributions include Shewhart's control chart [25], the minimax approach in Page's CUSUM procedure [26], [27], the Bayesian approach in the Shiryaev-Roberts procedure [28], [29], and window-limited procedures [30]. However, there is limited research on monitoring large-scale data streams over networks or even event streams over networks. The detection of change points in point processes has so far mostly focused on simple Poisson process models without considering temporal dependency, and most detection statistics are computed in a discrete-time manner, that is, one needs to aggregate the observed events into bins and then apply traditional detection approaches to the time series of count data. Examples include [2], [31], [32].

The notations used herein are standard. The remaining sections are organized as follows. Section II presents the point process model and derives the likelihood functions. Section III presents our sequential likelihood ratio procedure. Section IV presents the EM-like algorithm. Section V presents our theoretical approximation of the false-alarm rate. Section VI contains the numerical examples. Section VI presents our results for real data. Finally, Section VIII summarizes the paper. All proofs are relegated to the Appendix.

II. Model and Formulation

Consider a sequence of events over a network with d nodes, represented as a double sequence

$$(t_1, u_1), (t_2, u_2), \ldots, (t_n, u_n), \ldots \tag{1}$$

where $t_i \in \mathbb{R}^+$ denotes the real-valued time when the ith event occurs and $i \in \mathbb{Z}^+$ and $u_i \in \{1, 2, \ldots, d\}$ indicate the node index where the event occurs. We use temporal point processes [33] to model the discrete event streams because they represent a convenient tool for directly modeling the time intervals between events, avoid the need to pick a time window to aggregate events, and allow temporal events to be modeled in a fine-grained manner.

A. Temporal Point Processes

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$, with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. We start by considering one-dimensional point processes. Let the list of times of events up to but not including time t be the history

$$\mathcal{H}_t = \{t_1, \ldots, t_n : t_n < t\}.$$

Let N_t represent the total number of events until time t. Then, the counting measure can be defined as

$$dN_t = \sum_{t_i \in \mathcal{H}_t} \delta(t - t_i) dt, \tag{2}$$

where $\delta(t)$ is the Dirac delta function.

To define the likelihood ratio for point processes, we first introduce the concept of the *conditional intensity function* [34]. The conditional intensity function is a convenient and intuitive way of specifying how the present depends on the past in a temporal point process. Let $F^*(t)$ be the conditional probability that the next event t_{n+1} occurs before t given the history of previous events

$$F^*(t) = \mathbb{P}\{t_{n+1} < t | \mathcal{H}_t\},$$

and let $f^*(t)$ be the corresponding conditional density function. The conditional intensity function (or the hazard function) [34] is defined by

$$\lambda_t = \frac{f^*(t)}{1 - F^*(t)}, \tag{3}$$

and it can be interpreted as the probability that an event occurs in an infinitesimal interval

$$\lambda_t dt = \mathbb{P}\{\text{event in } [t, t + dt) | \mathcal{H}_t\}. \tag{4}$$

This general model includes the Poisson process and the Hawkes process as special cases.

i) For (inhomogeneous) Poisson processes, each event is stochastically independent of all the other events in the process, and the time intervals between consecutive events are independent of each other and are exponentially distributed. As a result, the conditional intensity function is independent of the past, which is simply deterministic $\lambda_t = \mu_t$.

ii) For one-dimensional Hawkes processes, the intensity function is history dependent and models a mutual excitation between events

$$\lambda_t = \mu_t + \alpha \int_0^t \varphi(t - \tau) dN_\tau, \tag{5}$$

where μ_t is the base intensity (deterministic), $\alpha \in (0, 1)$ (due to the requirement of stationary condition) is the influence parameter, and $\varphi(t)$ is a normalized kernel function $\int \varphi(t) dt = 1$. Together, they characterize how the history influences the current intensity. Fixing the kernel function, a higher value of α means a stronger temporal dependency between events. A commonly used kernel function is the exponential kernel $\varphi(t) = \beta e^{-\beta t}$, which we will use throughout the paper.

iii) The multi-dimensional Hawkes process is defined similarly, with each dimension being a one-dimensional counting process. It can be used to model the sequence of events over a network such as (1). We may convert a multi-dimensional process into a double sequence, therein using the first coordinate to represent the time of the event and the second coordinate to represent the index of the corresponding node.

Define a multivariate counting process $(N_t^1, N_t^2, \ldots, N_t^d)$, $t \geqslant 0$, with each component N_t^i recording the number of events

of the i-th component (node) of the network during $[0,t]$. The intensity function is

$$\lambda_t^i = \mu_t^i + \sum_{j=1}^{d} \int_0^t \alpha_{ij}\varphi(t-\tau)dN_\tau^j, \tag{6}$$

where α_{ij}, $j,i \in \{1,\ldots,d\}$ represents the strength of influence of the j-th node on the i-th node by affecting its intensity process λ^i. If $\alpha_{ij}=0$, then N^j is not influencing N^i. Written in matrix form, the intensity can be expressed as

$$\boldsymbol{\lambda}_t = \boldsymbol{\mu}_t + \boldsymbol{A}\int_0^t \varphi(t-\tau)d\boldsymbol{N}_\tau, \tag{7}$$

where

$$\boldsymbol{\mu}_t = [\mu_t^1, \mu_t^2, \ldots, \mu_t^d]^\intercal, d\boldsymbol{N}_\tau = [dN_\tau^1, dN_\tau^2, \ldots, dN_\tau^d]^\intercal,$$

and $\boldsymbol{A} = [\alpha_{ij}]_{1\leqslant i,j\leqslant d}$ is the *influence matrix*, which is our main quantity-of-interest when detect a change. The diagonal entries characterize the self-excitation, and the off-diagonal entries capture the mutual excitation among nodes in the network. The influence matrix can be asymmetric because influence can be bidirectional.

he topology of the network has been embedded in the sparsity pattern of the influence matrix A, which are given as a priori. The dependency between different nodes in the network and the temporal dependency over events can be captured by updating (or tracking) the influence matrix A with the event stream. This can be achieved through an EM-like algorithm, which results from solving a sequential optimization problem with warm start (i.e., we always initialize the parameters using the optimal solutions of the previous step).

B. Likelihood Function

In the following, we will explicitly denote the dependence of the likelihood function on the parameters in each setting. The following three cases are useful for our subsequent derivations. Let $f(t)$ denote the probability density function. For the one-dimensional setting, given a sequence of n events (event times) $\{t_1,t_2,\ldots,t_n\}$ before time t, using the conditional probability formula, we obtain

$$\begin{aligned}\mathcal{L} &= f(t_1,\ldots,t_n) = (1-F^*(t))\prod_{i=1}^{n} f(t_i|t_1,\ldots,t_{i-1}) \\ &= (1-F^*(t))\prod_{i=1}^{n} f^*(t_i) = \left(\prod_{i=1}^{n}\lambda_{t_i}\right)\exp\left\{-\int_0^t \lambda_s ds\right\}.\end{aligned} \tag{8}$$

The last equation is based on the following argument. From the definition of the conditional density function, we have

$$\lambda_t = \frac{d}{dt}F^*(t)/(1-F^*(t)) = -\frac{d}{dt}\log(1-F^*(t)).$$

Hence, $\int_{t_n}^t \lambda_s ds = -\log(1-F^*(t))$, where $F^*(t_n)=0$ because event $n+1$ cannot occur at time t_n. Therefore,

$$F^*(t) = 1-\exp\left\{-\int_{t_n}^t \lambda_s ds\right\}, f^*(t) = \lambda_t \exp\left\{-\int_{t_n}^t \lambda_s ds\right\}.$$

The likelihood function for the multi-dimensional Hawkes process can be derived similarly by redefining $f^*(t)$ and $F^*(t)$ according to the intensity functions of the multi-dimensional processes.

Based on the above principle, we can derive the following likelihood functions.

1) Homogeneous Poisson process: For the homogeneous Poisson process, $\lambda_t = \mu$. Given a constant intensity, the log-likelihood function for a list of events $\{t_1,t_2,\ldots,t_n\}$ in the time interval $[0,t]$ can be written as

$$\log\mathcal{L}(\mu) = n\log\mu - \mu t. \tag{9}$$

2) One-Dimensional Hawkes Process: For the one-dimensional Hawkes process with constant baseline intensity $\mu_t=\mu$ and exponential kernel, we may obtain its log-likelihood function based on the above calculation. By substituting the conditional intensity function (5) into (8), the log-likelihood function for events in the time interval $[0,t]$ is given by

$$\begin{aligned}\log\mathcal{L}(\alpha,\beta,\mu) &= \sum_{i=1}^{n}\log\left(\mu+\alpha\sum_{t_j<t_i}\beta e^{-\beta(t_i-t_j)}\right) \\ &\quad -\mu t - \sum_{t_i<t}\alpha\left[1-e^{-\beta(t-t_i)}\right].\end{aligned} \tag{10}$$

To obtain the above expression, we have used the following two simple results for exponential kernels based on the property of counting measure defined in (2):

$$\lambda_t = \mu+\alpha\int_{-\infty}^t \varphi(t-\tau)dN_\tau = \mu+\alpha\sum_{t_i<t}\beta e^{-\beta(t-t_i)}, \tag{11}$$

and

$$\int_0^t \lambda_s ds = \mu t + \sum_{t_i<t}\alpha\left[1-e^{-\beta(t-t_i)}\right]. \tag{12}$$

3) Multi-Dimensional Hawkes Process: For multi-dimensional point processes, we consider event stream such as (1). Assume that the base intensities are constants with $\mu_t^i \triangleq \mu_i$. Using similar calculations as above, we obtain the log-likelihood function for events in the time interval $[0,t]$ as

$$\begin{aligned}\log\mathcal{L}(\boldsymbol{A},\beta,\boldsymbol{\mu}) &= \sum_{i=1}^{n}\log\left[\mu_{u_i}+\sum_{t_j<t_i}\alpha_{u_i,u_j}\beta e^{-\beta(t_i-t_j)}\right] \\ &\quad -\sum_{j=1}^{d}\mu_j t - \sum_{j=1}^{d}\sum_{t_i<t}\alpha_{u_i,j}\left[1-e^{-\beta(t-t_i)}\right].\end{aligned} \tag{13}$$

III. Sequential Change-Point Detection

We are interested in detecting two types of changes sequentially from event streams, which capture two general scenarios in real applications (Fig. 2 illustrates these two scenarios for the one-dimensional setting): (*i*) The sequence before the change is a Poisson process, and after the change, it is a Hawkes process. This can be useful for applications where we are interested in detecting an emergence of self- or mutual-excitation between nodes. (*ii*) The sequence before the change is a Hawkes process,

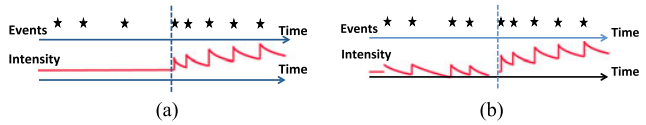

Fig. 2. Illustration of scenarios for one-dimensional examples: (a) Poisson to hawkes; (b) Hawkes to hawkes.

and after the change, the magnitude of the influence matrix increases. This can be a more realistic scenario because, often, nodes in a network will influence each other initially. This can be useful for applications where a triggering event changes the behavior or structure of the network, for instance, in detecting the emergence of a community in a network [35].

In the following, we cast the change-point detection problems as the sequential hypothesis test [36], and we derive the generalized likelihood ratio (GLR) statistic for each case. Suppose that there may exist an unknown change-point κ such that, after that time, the distribution of the point process changes.

A. Change From Poisson to Hawkes

First, we are interested in detecting the events over a network changing from d-dimensional independent Poisson processes to an intertwined multi-variate Hawkes process. This models the effect that the change affects the spatial dependency structure over the network. Below, we first consider the one-dimensional setting, and then, we generalize to the multi-dimensional case.

1) One-Dimensional Case: The data consist of a sequence of events occurring at time $\{t_1, t_2, \ldots, t_n\}$. Under the hypothesis of no change (i.e., H_0), the event time is a one-dimensional Poisson process with intensity λ. Under the alternative hypothesis (i.e., H_1), there exists a change-point κ. The sequence is a Poisson process with intensity λ initially, and it changes to a one-dimensional Hawkes process with parameter α after the change. Formally, the hypothesis test can be stated as

$$\begin{cases} \mathsf{H}_0: \ \lambda_s = \mu, & 0 < s < t; \\ \mathsf{H}_1: \ \lambda_s = \mu, & 0 < s < \kappa, \\ \quad\ \lambda_s^* = \mu + \alpha \int_\kappa^s \varphi(s-\tau) dN_\tau, & \kappa < s < t. \end{cases} \tag{14}$$

Assume that the intensity μ can be estimated from reference data and that β is given a priori. We treat the post-change influence parameter α as an unknown parameter because it represents an anomaly.

Using the likelihood functions derived in Section II-B, equations (9) and (10), for a hypothetical change-point location τ, the log-likelihood ratio as a function of α, β and μ is given by

$$\begin{aligned} \ell_{t,\tau,\alpha} &= \log \mathcal{L}(\alpha, \beta, \mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau, t)} \log \left[\mu + \alpha \sum_{t_j \in (\tau, t_i)} \beta e^{-\beta(t_i - t_j)} \right] \\ &\quad - \mu(t-\tau) - \alpha \sum_{\tau_i \in (\tau, t)} \left[1 - e^{-\beta(t-t_i)} \right]. \end{aligned} \tag{15}$$

Note that the log-likelihood ratio only depends on the events in the interval (τ, t) and α. We maximize the statistic with respect to the unknown parameters α and τ to obtain the log GLR statistic. Finally, the sequential change-point detection procedure is a stopping rule (related to the non-Bayesian minimax type of detection rules; see [37]):

$$T_{\text{one-dim}} = \inf\{t : \max_{\tau < t} \max_{\alpha} \ \ell_{t,\tau,\alpha} > x\}, \tag{16}$$

where x is a prescribed threshold, the choice of which will be discussed later. Although there does not exist a closed-form expression for the estimator of α, we can estimate α via an EM-like algorithm, which will be discussed in Section IV-B.

Remark 1 (Offline detection): We can adapt the procedure for offline change-point detection by considering the fixed-sample hypothesis test. For instance, for the one-dimensional setting, given a sequence of n events with $t_{\max} \triangleq t_n$, we may detect the existence of a change when the detection statistic, $\max_{\tau < t_{\max}} \max_\alpha \ \ell_{t_{\max},\tau,\alpha}$, exceeds a threshold. The change-point location can be estimated as the τ^* that obtains the maximum. However, the algorithm considerations for online and offline detection are very different, as discussed in Section IV.

2) Multi-Dimensional Case: For the multi-dimensional case, the event stream data can be represented as a double sequence defined in (1). We may construct a similar hypothesis test as above. Under the hypothesis of no change, the event times are a multi-dimensional Poisson process with a vector intensity function $\boldsymbol{\lambda}_s = \boldsymbol{\mu}$. Under the alternative hypothesis, there exists a change point κ. The sequence is initially a multi-dimensional Poisson process but changes to a multi-dimensional Hawkes process with influence matrix $\boldsymbol{A}$ afterward. We omit the formal statement of the hypothesis test, as it is similar to (14).

Again, using the likelihood functions derived in Section II-B, we obtain the likelihood ratio. The log-likelihood ratio for data up to time t, given a hypothetical change-point location τ and parameter $\boldsymbol{A}$, is given by

$$\begin{aligned} \ell_{t,\tau,\boldsymbol{A}} &= \log \mathcal{L}(\boldsymbol{A}, \beta, \mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau, t)} \log \left[1 + \frac{1}{\mu_{u_i}} \sum_{t_j \in (\tau, t_i)} \alpha_{u_i, u_j} \beta e^{-\beta(t_i - t_j)} \right] \\ &\quad - \sum_{j=1}^{d} \sum_{t_i \in (\tau, t)} \alpha_{j, u_i} \left[1 - e^{-\beta(t-t_i)} \right]. \end{aligned} \tag{17}$$

Here, recall from the original form of the data (1) that (t_i, u_i) represents the ith event's occurrence time and the corresponding node u_i where the event occurs. Hence, (17) means that, to evaluate the likelihood for a time window (τ, t), one should consider all events that fall within that interval and aggregate the intensities using nodes corresponding to these events. The sequential change-point detection procedure is a stopping rule:

$$T_{\text{multi-dim}} = \inf\{t : \max_{\tau < t} \max_{\mathbf{A}} \ \ell_{t,\tau,\mathbf{A}} > x\}, \tag{18}$$

where x is a pre-determined threshold. The multi-dimensional maximization can be computed efficiently via the EM algorithm described in Section IV-B .

B. Changes From Hawkes to Hawkes

Next, consider the scenario where the process prior to a change is a Hawkes process, and the change occurs in the influence parameter α or the influence matrix $\boldsymbol{A}$.

1) One-Dimensional Case: Under the hypothesis of no change, the event stream is a one-dimensional Hawkes process with parameter α. Under the alternative hypothesis, there exists a change point κ. The sequence is a Hawkes process with intensity α, and after the change, the intensity changes to α^*. Assume that the parameter α prior to the change is known.

Using the likelihood functions derived in Section II-B, we obtain the log-likelihood ratio

$$\begin{aligned}\ell_{t,\tau,\alpha^*} &= \log \mathcal{L}(\alpha^*, \beta, \mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau,t)} \log \left[\frac{\mu + \alpha^* \sum_{t_j \in (\tau,t_i)} \beta e^{-\beta(t_i - t_j)}}{\mu + \alpha \sum_{t_j \in (\tau,t_i)} \beta e^{-\beta(t_i - t_j)}} \right] \\ &\quad - (\alpha^* - \alpha) \sum_{t_i \in (\tau,t)} \left[1 - e^{-\beta(t - t_i)} \right], \end{aligned} \tag{19}$$

and the change-point detection is achieved through a procedure in the form of (16) by maximizing with respect to τ and α. Here, recall from the original form of the data (1) that t_i represents the ith event's occurrence time. Hence, (19) means that, to evaluate the likelihood for a time window (τ, t), one should consider all events that fall within that interval and use their occurrence times.

2) Multi-Dimensional Case: For the multi-dimensional setting, we assume that the change will alter the influence parameters of the multi-dimensional Hawkes process over the network. This captures the effect that, after the change, the influence between nodes becomes different. Assume that ,under the hypothesis of no change, the event stream is a multi-dimensional Hawkes process with parameter $\boldsymbol{A}$. Alternatively, there exists a change point κ. The sequence is a multi-dimensional Hawkes process with influence matrix $\boldsymbol{A}$ before the change, and after the change, the influence matrix becomes $\boldsymbol{A}^*$. Assume that the influence matrix $\boldsymbol{A}$ prior to the change is known.

Using the likelihood functions derived in Section II-B, the log-likelihood ratio at time t for a hypothetical change-point location τ and post-change parameter value $\boldsymbol{A}^*$ is given by

$$\begin{aligned}\ell_{t,\tau,\boldsymbol{A}^*} &= \log \mathcal{L}(\boldsymbol{A}^*, \beta, \mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau,t)} \log \left[\frac{\mu_{u_i} + \sum_{t_j \in (\tau,t_i)} \alpha^*_{u_i,u_j} \beta e^{-\beta(t_i - t_j)}}{\mu_{u_i} + \sum_{t_j \in (\tau,t_i)} \alpha_{u_i,u_j} \beta e^{-\beta(t_i - t_j)}} \right] \\ &\quad - \sum_{j=1}^{d} \sum_{t_i \in (\tau,t)} \left(\alpha^*_{j,u_i} - \alpha_{j,u_i} \right) \left[1 - e^{-\beta(t - t_i)} \right], \end{aligned} \tag{20}$$

and the change-point detection is applied through a procedure in the form of (18) by maximizing with respect to τ and $\boldsymbol{A}^*$. Here, recall from the original form of the data (1) that (t_i, u_i) represents the ith event's occurrence time and the node where the event occurs. Hence, (17) means that, to evaluate the likelihood for a time window (τ, t), one should consider all events that fall within that interval and aggregate the intensities using the edges $\alpha^*_{u_i,u_j}$ (null influence parameter) or α_{u_i,u_j} (alternative influence parameter) across nodes that correspond to these events.

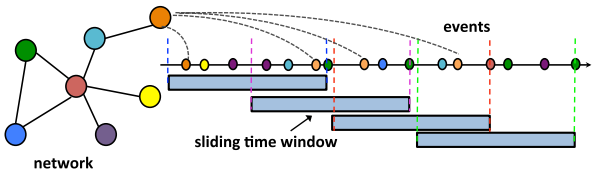

Fig. 3. Illustration of the sliding window approach for online detection.

IV. Algorithm for Computing Likelihood Online

In the online setting, we obtain new data continuously. Hence, to perform online detection, we need to update the likelihood efficiently to incorporate the new data. To reduce the computational cost, update of the likelihood function can be computed recursively, and the update algorithm should have a low cost. To reduce memory requirements, the algorithm should only store the minimum amount of data necessary for detection rather than the complete history. These requirements make online detection drastically different from offline detection because, in the offline setting, we can afford greater computational complexity.

A. Sliding Window Procedure

The basic idea of the online detection procedure is illustrated in Fig. 3. We adopt a *sliding window* approach to reduce the computational complexity as well the memory requirements. We update the detection statistic *asynchronously* every γ events, i.e., when $\text{mod}(i, \gamma) = 0$, where i is the event index (in all our examples in Sections VI and VII, we set $\gamma = 1$, i.e., update the detection statistic upon every new event). When evaluating the likelihood function, instead of maximizing over ever possible change-point location $\tau < t$, we pick several possible change-point locations within a window size L and maximize the statistics over several values of τ, e.g., $\tau \in \Omega_t \triangleq \{t - \Delta_1, t - \Delta_2, \ldots, t - \Delta_k\}$, where Δ_i are the chosen offsets of possible change-point locations from the current time. In this way, we reduce the computational complexity because we eliminate the maximization over all possible change-point locations before time t. This also reduces the memory requirement, as we only need to store events that fall into the sliding window. The drawback is that, by doing this, some statistical detection power is lost because we do not use the most likely change-point location, and this may increase the detection delay.

When implementing the algorithm, we choose the Ω_t to achieve a good balance in these two aspect. We have to choose a window length that is sufficiently large so that there are enough events stored for us to make a consistent inference. In practice, a proper length of window relies on the nature of the data. If the data are noisy, a longer time window is usually needed to achieve a better estimation of the parameter and reduce the false-alarm rate.

B. Parameter-Free EM-Like Algorithm

We consider a one-dimensional point process to illustrate the derivation of the EM-like algorithm. It can be shown that the

Algorithm 1: Online Detection Algorithm.

Require: Data $\{(t_i, u_i)\}$.
Grid points of change-point locations: $\Omega_t \triangleq \{t-\Delta_1, t-\Delta_2, \ldots, t-\Delta_k\}$.
Update frequency γ (events).
Initialization for parameters α (one-dimension) or $\boldsymbol{A}$ (multi-dimension).
Pre-defined threshold: x.
Estimation accuracy: ϵ.

repeat
  if $\mod(i,\gamma)=0$ **then**
    Initialize $\alpha^{(0)}=\hat{\alpha}$ or $\boldsymbol{A}^{(0)}=\hat{\boldsymbol{A}}$ {warm start}
    repeat
      Perform {E-step} and {M-step} from Section IV-B
    until $\|\alpha^{(k+1)}-\alpha^{(k)}\|<\epsilon$ or $\|\boldsymbol{A}^{(k+1)}-\boldsymbol{A}^{(k)}\|<\epsilon$
    Let $\hat{\alpha}=\alpha^{(k+1)}$ and $\hat{\boldsymbol{A}}=\boldsymbol{A}^{(k+1)}$.
    Use $\hat{\alpha}$ or $\hat{\boldsymbol{A}}$ to compute log likelihood using (15), (17), (19) or (20).
  end if
until $\max_{\tau\in\Omega_t}\ell_{t,\tau,\hat{\alpha}}>x$ or $\max_{\tau\in\Omega_t}\ell_{t,\tau,\hat{\boldsymbol{A}}}>x$ and announce a change.

likelihood function (15) is a concave function with respect to the parameter α. One can use gradient descent to optimize this objective function (maximizing the likelihood function), where the algorithm will typically involve some additional tuning parameters such as the learning rate. In this problem, however, we determine that we may use an EM algorithm that is free of any tuning parameters. Although there does not exist a closed-form estimator for the influence parameter α or the influence matrix $\boldsymbol{A}$, we develop an efficient EM algorithm to update the likelihood, therein exploiting the structure of the likelihood function [38]. The iterations in the EM method are performed before any new observation. The overall algorithm is summarized in Algorithm 1.

First, we obtain a concave lower bound of the likelihood function using Jensen's inequality. Consider that all events fall into a sliding window $t_i\in(\tau,t)$ at time t. Introduce the auxiliary variables p_{ij} for all pairs of events (i,j) within the window such that $t_j<t_i$. The variables are subject to the constraint

$$\forall i,\ \sum_{t_j<t_i} p_{ij}=1,\quad p_{ij}\geqslant 0. \tag{21}$$

These p_{ij} can be interpreted as the probability that the j-th event influences the i-th event in the sequence. It can be shown that the likelihood function defined in (10) can be lower bounded

$$\ell_{t,\tau,\alpha}\geqslant \sum_{t_i\in(\tau,t)}\left(p_{ii}\log(\mu)+\sum_{t_j\in(\tau,t_i)}p_{ij}\log\left[\alpha\beta e^{-\beta(t_i-t_j)}\right]\right.$$
$$\left.-\sum_{t_j\in(\tau,t)}p_{ij}\log p_{ij}\right)-\mu(t-\tau)-\alpha\sum_{t_i\in(\tau,t)}\left[1-e^{-\beta(t-t_i)}\right],$$

Note that the lower bound is valid for every choice of $\{p_{ij}\}$ that satisfies (21).

To make the lower bound tight and ensure improvement in each iteration, we will maximize it with respect to p_{ij} and obtain (22) (assuming that we have $\alpha^{(k)}$ from a previous iteration or initialization). Once we have a tight lower bound, we will take the gradient of this lower bound with respect to α. When updating from the k-th iteration to the $(k+1)$-th iteration, we obtain (23)

$$p_{ij}^{(k)}=\frac{\alpha^{(k)}\beta e^{-\beta(t_j-t_i)}}{\mu+\alpha^{(k)}\beta\sum_{t_m\in(\tau,t_j)}e^{-\beta(t_j-t_m)}}\quad\{\text{E-step}\} \tag{22}$$

$$\alpha^{(k+1)}=\frac{\sum_{i<j}p_{ij}^{(k)}}{\sum_{t_i\in(\tau,t)}[1-e^{-\beta(t-t_i)}]}\quad\{\text{M-step}\} \tag{23}$$

where the superscript denotes the number of iterations. The algorithm iterates these two steps until the algorithm converges and obtains the estimated α. In practice, we find that we only need 3 or 4 iterations to converge if using warm start.

Similarly, an online estimate for the influence matrix for the multi-dimensional case can be estimated by iterating the following two steps:

$$p_{ij}^{(k)}=\frac{\alpha_{u_i,u_j}^{(k)}\beta e^{-\beta(t_i-t_j)}}{\mu_{u_i}+\beta\sum_{t_m\in(\tau,t_i)}\alpha_{u_i,u_m}^{(k)}e^{-\beta(t_i-t_m)}},\quad\{\text{E-step}\}$$

$$\alpha_{u,v}^{(k+1)}=\frac{\sum_{i:\,u_i=u}\sum_{j<i:\,u_j=v}p_{ij}^{(k)}}{\sum_{j:\,t_j\in(\tau,t),u_j=v}\left[1-e^{-\beta(t-t_j)}\right]}.\quad\{\text{M-step}\}$$

The overall detection procedure is summarized in Fig. 3 and Algorithm 1.

Remark 2 (Computational complexity): The key computation is to compute pairwise inter-event times for pairs of events t_i-t_j, $i<j$. This is related to the window size (because we have adopted a sliding window approach), the size of the network, and the number of EM steps. However, note that, in the EM algorithm, we only need to compute the inter-event times for nodes that are connected by an edge because the summation is weighted by α_{ij}, and the term only counts if α_{ij} is non-zero. Hence, the updates only involve neighboring nodes, and the complexity is proportional to the number of edges in the network. Because most social networks are sparse, the complexity will be lowered significantly. We may reduce the number of EM iterations for each update by leveraging a warm start for initializing the parameter values because, for two adjacent sliding windows, the corresponding optimal parameter values typically should be very close to the previous values.

Remark 3 (Distributed implementation): Our EM-like algorithm in a network setting can be implemented in a distributed manner. This has been embedded in the form of the algorithm already. Hence, the algorithm can be used for processes in large networks. In the E-step, when updating the p_{ij}, we need to evaluate a sum in the denominator, and this is the only place where different nodes need to exchange information, i.e., the event times occurred at that node. Because we only need to sum over all events such that the corresponding α_{u_i,u_j} is non-zero, each node only needs to consider the events that occurred at the neighboring nodes. Similarly, in the M-step, only neighboring

TABLE I
EXPRESSIONS FOR I, I_0, σ^2 AND σ_0^2 UNDER DIFFERENT SETTINGS

Setting	I	I_0	σ^2	σ_0^2
Poi. $\to$ Haw. (one dim.)	$\frac{\mu}{1-\alpha}\log\left(\frac{1}{1-\alpha}\right)-\frac{\alpha}{1-\alpha}\mu$	$\mu\log\left(\frac{1}{1-\alpha}\right)-\frac{\alpha}{1-\alpha}\mu$	$\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2\cdot\left[\frac{\mu}{1-\alpha}+\frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$	$\mu\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2$
Poi. $\to$ Haw. (high dim.)	$\bar{\boldsymbol{\lambda}}^{*\intercal}\left(\log(\bar{\boldsymbol{\lambda}}^*)-\log(\boldsymbol{\mu})\right)-\boldsymbol{e}^\intercal(\bar{\boldsymbol{\lambda}}^*-\boldsymbol{\mu})$	$\boldsymbol{\mu}^\intercal\left(\log(\bar{\boldsymbol{\lambda}}^*)-\log(\boldsymbol{\mu})\right)-\boldsymbol{e}^\intercal(\bar{\boldsymbol{\lambda}}^*-\boldsymbol{\mu})$	$\boldsymbol{e}^\intercal(\boldsymbol{H}\circ\boldsymbol{C})\,\boldsymbol{e}$	$\boldsymbol{\mu}^\intercal\left(\log(\bar{\boldsymbol{\lambda}}^*)-\log(\boldsymbol{\mu})\right)^{(2)}$
Haw. $\to$ Haw. (one dim.)	$\frac{\mu}{1-\alpha^*}\log\left(\frac{1-\alpha}{1-\alpha^*}\right)-\frac{\mu}{1-\alpha^*}+\frac{\mu}{1-\alpha}$	$\frac{\mu}{1-\alpha}\log\left(\frac{1-\alpha}{1-\alpha^*}\right)-\frac{\mu}{1-\alpha^*}+\frac{\mu}{1-\alpha}$	$\left[\log\left(\frac{1-\alpha}{1-\alpha^*}\right)\right]^2\cdot\left[\frac{\mu}{1-\alpha^*}+\frac{\alpha^*(2-\alpha^*)\mu}{(1-\alpha^*)^3}\right]+\left(1-\frac{1-\alpha}{1-\alpha^*}\right)^2\cdot\left[\frac{\mu}{1-\alpha}+\frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$	$\left[1-\frac{1-\alpha^*}{1-\alpha}\right]^2\cdot\left[\frac{\mu}{1-\alpha^*}+\frac{\alpha^*(2-\alpha^*)\mu}{(1-\alpha^*)^3}\right]+\left[\log\left(\frac{1-\alpha}{1-\alpha^*}\right)\right]^2\cdot\left[\frac{\mu}{1-\alpha}+\frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$
Haw. $\to$ Haw. (multi dim.)	$\bar{\boldsymbol{\lambda}}^{*\intercal}\left[\log\bar{\boldsymbol{\lambda}}^*-\log\bar{\boldsymbol{\lambda}}\right]-\boldsymbol{e}^\intercal[\bar{\boldsymbol{\lambda}}^*-\bar{\boldsymbol{\lambda}}]$	$\bar{\boldsymbol{\lambda}}^{\intercal}\left[\log\bar{\boldsymbol{\lambda}}^*-\log\bar{\boldsymbol{\lambda}}\right]-\boldsymbol{e}^\intercal[\bar{\boldsymbol{\lambda}}^*-\bar{\boldsymbol{\lambda}}]$	$\boldsymbol{e}^\intercal\left(\boldsymbol{G}\circ\boldsymbol{C}^*+\boldsymbol{F}\circ\boldsymbol{C}\right)\boldsymbol{e}$	$\boldsymbol{e}^\intercal\left(\boldsymbol{R}\circ\boldsymbol{C}^*+\boldsymbol{G}\circ\boldsymbol{C}\right)\boldsymbol{e}$

In the table above, $\boldsymbol{M}^{(2)}=\boldsymbol{M}\circ\boldsymbol{M}$ denotes the Hadamard product, and the related quantities are defined as

$$\bar{\boldsymbol{\lambda}}^*=(\boldsymbol{I}-\boldsymbol{A}^*)^{-1}\boldsymbol{\mu},\quad \bar{\boldsymbol{\lambda}}=(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu},\quad \boldsymbol{H}=\left[\log\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right)-\log\left(\boldsymbol{\mu}\right)\right]\cdot\left[\log\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right)-\log\left(\boldsymbol{\mu}\right)\right]^\intercal,$$
$$\boldsymbol{C}=(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\left(2\boldsymbol{I}+(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right)\operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right)+\operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right),$$
$$\boldsymbol{C}^*=(\boldsymbol{I}-\boldsymbol{A}^*)^{-1}\boldsymbol{A}^*\left(2\boldsymbol{I}+(\boldsymbol{I}-\boldsymbol{A}^*)^{-1}\boldsymbol{A}^*\right)\cdot\operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A}^*)^{-1}\boldsymbol{\mu}\right)+\operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A}^*)^{-1}\boldsymbol{\mu}\right),$$
$$\boldsymbol{G}_{ij}=[\log\left(\bar{\lambda}_i^*/\bar{\lambda}_i\right)]\cdot[\log\left(\bar{\lambda}_j^*/\bar{\lambda}_j\right)],\quad \boldsymbol{F}_{ij}=\left(1-\bar{\lambda}_i^*/\bar{\lambda}_i\right)\left(1-\bar{\lambda}_j^*/\bar{\lambda}_j\right),\quad \boldsymbol{R}_{ij}=\left(\bar{\lambda}_i/\bar{\lambda}_i^*-1\right)\left(\bar{\lambda}_j/\bar{\lambda}_j^*-1\right),\quad 1\leqslant i\leqslant j\leqslant d.$$

nodes need to exchange their values of p_{ij} and event times to update the influence parameter values.

V. THEORETICAL THRESHOLD

A key step in implementing the detection algorithm is to set the threshold. The choice of threshold involves a trade-off between two standard performance metrics for sequential change-point detection: the false-alarm rate and how fast we can detect the change. Formally, these two performance metrics are (i) the expected stopping time when there are no change points, called the average run length (ARL), and (ii) the expected detection delay when a change point exists.

Typically, a higher threshold x results in a larger ARL (and hence a smaller false-alarm rate) but a larger detection delay. A typical practice is to set the false-alarm rate (or ARL) to a pre-determined value and find the corresponding threshold x. The pre-determined ARL depends on how frequent we can tolerate false detection (once a month or once a year). Usually, the threshold is estimated via direct Monte Carlo by relating the threshold to the ARL assuming that the data follow a null distribution. However, Monte Carlo is not only computationally expensive, but in some practical problems, repeated experiments would be prohibitive. Therefore, it is important to find a cheaper method to accurately estimate the threshold.

We develop an analytical function that relates the threshold to ARL for the special case in which we set $\tau=t-L$ or equivalently $\Omega_t=\{(t-L)\}$, where L is the window length. This means that we consider all events within the time interval $(t-L,t)$. Given a prescribed ARL, we can solve for the corresponding threshold x analytically. We first characterize the property of the likelihood ratio statistic in the following lemma, which states that the mean and variance of the log-likelihood ratios both scale roughly linearly with the post-change time duration. This property of the likelihood ratio statistics is key to developing our main result.

Lemma 1 (Mean and variance of log-likelihood ratios): When the number of post-change samples $(t-\tau)$ is large, the mean and variance of the log-likelihood ratio for the single-dimensional and multi-dimensional cases, denoted as $\ell_{t,\tau,\cdot}$, for our cases converge to a simple linear form. Under the null hypothesis, $\mathbb{E}[\ell_{t,\tau,\cdot}]\approx(t-\tau)I_0$ and $\mathbb{E}[\ell_{t,\tau,\cdot}]\approx(t-\tau)\sigma_0^2$. Under the alternative hypothesis, $\mathbb{E}[\ell_{t,\tau,\cdot}]\approx(t-\tau)I$ and $\mathbb{E}[\ell_{t,\tau,\cdot}]\approx(t-\tau)\sigma^2$. Above, I, I_0, σ^2, and σ_0^2 are defined in Table I for various settings that we considered.

Our main theoretical result is the following general theorem that can be applied for all hypothesis tests that we consider. Denote the probability and expectation under the hypothesis of no change by $\mathbb{P}^\infty$ and $\mathbb{E}^\infty$, respectively.

Theorem 1 (ARL under the null distribution): When $x\to\infty$ and $x/\sqrt{L}\to c'$ for some constant c', the average run length (ARL) of the stopping time T defined in (16) for the one-dimensional case is given by

$$\mathbb{E}^\infty[T_{\text{one-dim}}]=e^x\left[\int_{\alpha\in\Theta}\nu\left(\frac{2\xi}{\eta^2}\right)\frac{\phi\left(\frac{LI-x}{\sqrt{L\sigma^2}}\right)}{\sqrt{L\sigma^2}}d\alpha\right]^{-1}\cdot(1+o(1)).\tag{24}$$

For the multi-dimensional case, the same expression holds for $\mathbb{E}^\infty[T_{\text{multi-dim}}]$, except that $\int_\alpha$ is replaced by $\int_{\boldsymbol{A}}$, which means taking the integral with respect to all *nonzero entries* of the matrix $\int_{\boldsymbol{A}}=\int\cdots\int_{\{\alpha_{ij},\alpha_{ij}\neq0\}}$. Above, the special function

$$\nu(\mu)\approx\frac{(2/\mu)\left(\Phi(\mu/2)-0.5\right)}{(\mu/2)\Phi(\mu/2)+\phi(\mu/2)}.$$

The specific expressions for I, I_0, σ^2, and σ_0^2 for various settings are summarized in Table I, and

$$\xi = -(I_0 - I), \quad \eta^2 = \sigma_0^2 + \sigma^2. \tag{25}$$

Above, $\Phi(x)$ and $\phi(x)$ are the cumulative distribution function (CDF) and the probability density function (PDF) of the standard normal, respectively.

Remark 4 (Evaluating integral): The multi-dimensional integral can be evaluated using the Monte Carlo method [39]. We use this approach for our numerical examples as well.

Remark 5 (Interpretation): The parameters I_0, I, σ_0^2 and σ^2 have the following interpretation:

$$\begin{aligned} &I_0 = \mathbb{E}[\ell_{t,\tau,\alpha}]/L, \quad \sigma_0^2 = \mathrm{Var}[\ell_{t,\tau,\alpha}]/L, \\ &I = \mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]/L, \quad \sigma^2 = \mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]/L, \end{aligned} \tag{26}$$

which are the mean and variance of the log-likelihood ratio under the null and alternative distributions, *per unit time*, respectively. Moreover, I can be interpreted roughly as the Kullback-Leibler information per time for each of the hypothesis tests that we consider.

The proof of Theorem 1 combines the recently developed change-of-measure techniques for sequential analysis with properties of the likelihood ratios for point processes, the mean field approximation for point processes, and the Delta method [40].

VI. Numerical Examples

In this section, we present some numerical experiments using synthetic data. We focus on comparing the EDD of our algorithm with two baseline methods, and we demonstrate the accuracy of the analytic threshold.

A. Comparison of EDD

1) Two Baseline Algorithms: We compare our method to two baseline algorithms. Baseline approaches 1-2 are implemented using a fixed window as the proposed method to achieve a fair comparison.

i) Baseline 1: is related to the commonly used "data binning" approach for processing discrete event data such as in [6]. This approach, however, ignores temporal correlations and correlations between nodes. Here, we convert the event data into counts by discretize time into a uniform grid, and we count the number of events occurring in each interval. Such counting data can be modeled via a Poisson distribution. We may derive a likelihood ratio statistic to detect a change. Suppose that $n_1, n_2, \ldots, n_c$ are the sequence of counting numbers following the Poisson distribution with intensity λ_i, $i = 1, 2, \ldots, c$ is the index of the discrete time step. Assume that, under the null hypothesis, the intensity function is $\lambda_i = \mu$. Alternatively, there may exist a change point κ such that, before the change, $\lambda_i = \mu$, and after the change, $\lambda_i = \mu^*$. It can be shown that the log-likelihood ratio statistic as

$$\ell_{c,k,\mu^*} = -(c-k)(\mu^* - \mu) + \sum_{i=k+1}^{c} n_i \log \frac{\mu^*}{\mu}.$$

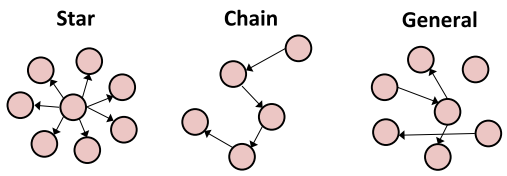

Fig. 4. Illustration of network topology.

We detect a change whenever $\max_{k<c} \max_{\mu^*} \ell_{k,c,\mu^*} > x$ for a pre-determined threshold x. Assume that every dimension follows an independent Poisson process; then, the log-likelihood ratio for the multi-dimensional case is simply a summation of the log-likelihood ratio for each dimension. Suppose that the total dimension is d; then,

$$\ell_{k,c,\boldsymbol{\mu}^*} = \sum_{j=1}^{d} \left[-(c-k)(\mu_j^* - \mu_j) + \sum_{i=k+1}^{c} n_i^j \log \frac{\mu_j^*}{\mu_j} \right].$$

We detect a change whenever $\max_{k<c} \max_{\boldsymbol{\mu}^*} \ell_{k,c,\boldsymbol{\mu}^*} > x$.

ii) The **Baseline 2** method calculates the one-dimensional change-point detection statistic at each node separately as (15) and (19), and then, it combines the statistics through summation into a *global statistic* to perform detection. This approach, however, ignores the correlation between nodes and can also be viewed as a *centralized* approach for change-point detection. In addition, it is related to multi-chart change-point detection [37].

2) Set-Up of Synthetic Experiments: We consider the following scenarios and compare the EDD of our method to two baseline methods. EDD is defined as the average time (delay) it takes before we can detect the change and can be understood as the power of the test statistic in the sequential setting. The thresholds of all three methods are calibrated such that the ARL under the null model is 10^4 unit time, and the corresponding thresholds are obtained via direct Monte Carlo to provide a fair comparison. The sliding window is set to be $L = 10$ unit time. The exponential kernel $\varphi(t) = \beta e^{-\beta t}$ is used, and $\beta = 1$. The scenarios that we considered are described below. The illustrations of the *Case 1* and *Case 2* scenarios are presented in Fig. 2. The network topology for *Case 3* to *Case 7* is demonstrated in Fig. 4.

Case 1: Consider a situation in which the events first follow a one-dimensional Poisson process with intensity $\mu = 10$ but then shift to a Hawkes process with influence parameter $\alpha = 0.5$. This scenario describes the emergence of temporal dependency in the event data.

Case 2: The process shifts from a one-dimensional Hawkes process with parameters $\mu = 10$ and $\alpha = 0.3$ to another Hawkes process with a larger influence parameter $\alpha = 0.5$. The scenario represents the change of the temporal dependency in the event data.

Case 3: Consider a star network scenario with one parent and nine children, which is commonly used in modeling information broadcasting over the network. Before the change point, each note has a base intensity $\mu = 1$ and self-excitation $\alpha_{i,i} = 0.3$, $1 \le i \le 10$. The mutual excitation from the parent to each child is set to be $\alpha_{1,j} = 0.3$, $2 \le j \le 10$ (if we use the first node to represent the parent). After the change point, all the self- and mutual- excitations increase to 0.5.

TABLE II
EDD COMPARISON. THRESHOLDS FOR ALL METHODS ARE CALIBRATED SUCH THAT $ARL = 10^4$

	Baseline 1	Baseline 2	Our Method
Case 1	22.1	–	**4.8**
Case 2	19.6	–	**18.8**
Case 3	8.2	6.9	**4.3**
Case 4	×	×	**19.8**
Case 5	6.1	5.7	**4.7**
Case 6	×	**10.5**	10.8
Case 7	×	**32.5**	**32.5**

Note: '×' indicates that the corresponding method fails to detect the changes; '–' indicates that, in the one-dimensional case, Baseline 2 is identical to ours.

Case 4: The network topology is the same as in Case 3. However, we consider a more challenging scenario. Before the change, the parameters are set to be the same as in Case 3. After the change, the self-excitation $\alpha_{i,i}$, $1 \leq i \leq 10$, deteriorates to 0.01, and the influence from the parent to the children increases to $\alpha_{1,j} = 0.6$, $j = 2 \leq j \leq 10$. In this case, for each note, the occurring frequency of events would be almost the same before and after the change points. However, the influence structure embedded in the network has actually changed.

Case 5: Consider a network with a chain of ten nodes, which is commonly used to model information propagation over the network. Before the change, each note has a base intensity $\mu = 1$, self-excitation $\alpha_{i,i} = 0.3$, $1 \leq i \leq 10$, and mutual-excitation $\alpha_{i,j} = 0.3$, where $j - i = 1, 1 \leq i \leq 9$. After the change point, all the self- and mutual-excitation parameters increase to 0.5.

Case 6: Consider a *sparse* network with an arbitrary topology and one-hundred nodes. Each note has a base intensity $\mu = 0.1$ and self-excitation $\alpha_{i,i} = 0.3, 1 \leq i \leq 100$. We randomly select twenty directed edges over the network and set the mutual excitation to be $\alpha_{i,j} = 0.3$, where $i \neq j$, i, j are randomly selected. After the change point, all the self- and mutual-excitations increase to 0.5.

Case 7: The *sparse* network topology and the pre-change parameters are the same as in Case 6. The only difference is that, after the change point, only half of the self- and mutual-excitation parameters increase to 0.5.

3) EDD Results and Discussion: For the above scenarios, we compare the EDD of our method and two baseline algorithms. The results are shown in Table II. We see that our method compares favorably to the two baseline algorithms. In the first five cases, our method presents a significant performance gain. Especially for Case 4, which is a challenging setting, only our method succeeds in detecting the spatial structure changes. For Case 6 and Case 7, our method achieves similar performance as Baseline 2. One possible reason for this is that, in these cases, the network topology is a sparse graph; thus, the nodes are "loosely" correlated. Hence, the advantage of combining over graphs is not significant in these cases.

Moreover, we observe that the Baseline 1 algorithm is not stable. In certain cases (Case 6 and Case 7), it completely fails to detect the change. An explanation for this is that there is a chance that the number of events falling into a given time bin is extremely small or close to zero, and this causes numerical issues when calculating the likelihood function (because there is a log function of the number of events). On the other hand, our proposed log-likelihood ratio is event triggered and hence will avoid such numerical issues.

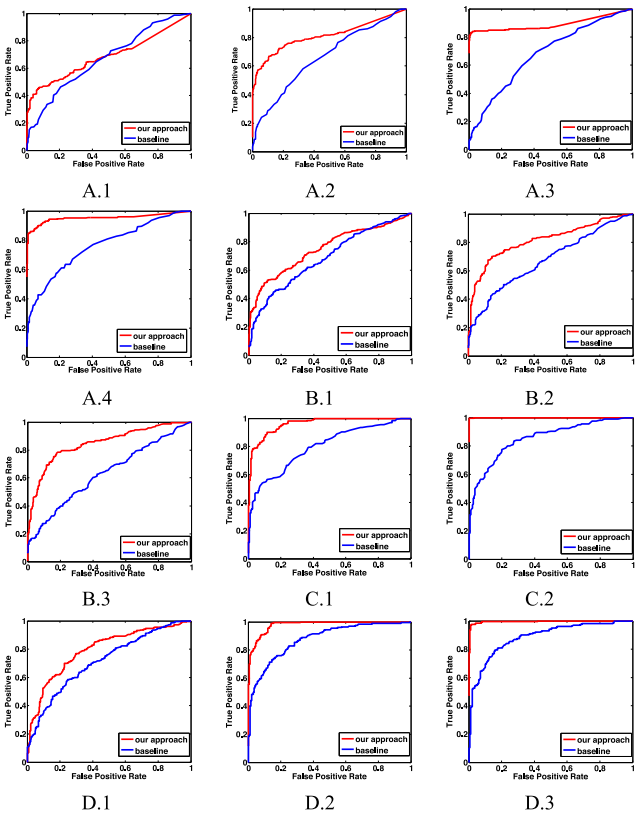

Fig. 5. AUC curves: Comparison of our method with Baseline 1. The window size L used is 1000.

B. Sensitivity Analysis

We also perform the sensitivity analysis by comparing our method to Baseline 1 algorithm via numerical simulation. The comparison is conducted under various kernel decay parameter β, and the strength of the post-change signals, which can be controlled by the magnitudes of the changes in α (or $\boldsymbol{A}$). For each dataset, we created 500 samples of sequences, with half of them containing one true change point and the other half containing no change points. We then plot the *area under the curve* (AUC) (defined as the true positive rate versus the false positive rate under various thresholds) for comparison, as shown in Fig. 5.

1) Set-Up of Synthetic Experiments: Overall, we consider various decay parameters β and magnitudes of the changes in α to compare the approaches.

One-dimensional setting. First, consider that, before the change, the data are a Poisson process with base intensity $\mu = 1$. For A.1-A.4, the post-change data become a one-dimensional Hawkes process; for A.1–A.3, $\alpha = 0.2$, and $\beta = 1, 10, 100$; and

for A.4, $\alpha = 0.3$, and $\beta = 10$. By comparing the AUC curves, we see that our method has a remarkably better performance in distinguishing the true positive changes from the false positive changes compared to the baseline method. The superiority would become more evident under larger β and larger magnitudes of shifts in α. For weak changes, the baseline approach is only slightly better than random guessing, whereas our approach consistently performs well. Similar results can be found if the pre-change data follow the Hawkes process. For example, in B.1-B.3, the pre-change data follow a Hawkes process with $\mu = 1$, $\alpha = 0.3$, and $\beta = 1$, and the post-change parameters shift to a Hawkes process with $\alpha = 0.5$ and $\beta = 1, 10, 100$. We can see a similar trend as before by varying β and α.

Network setting: We first consider the two-dimensional examples in the following and obtain the same results. For C.1-C.2, the pre-change data follow two-dimensional Poisson processes with $\boldsymbol{\mu} = [0.2, 0.2]^\intercal$, and the post-change data follow two-dimensional Hawkes processes with influence parameter $\boldsymbol{A} = [0.1, 0.1; 0.1, 0.1]$, with $\beta = 1, 10$. For D.1–D.3, consider the star network with one parent and nine children. Before the change point, for each node, the base intensity is $\mu = 0.1, \beta = 1$, and the influence from the parent to each child is $\alpha = 0.3$. After the change, α changes to 0.4 for D.1, and α changes to 0.5, $\beta = 1, 10$, for D.2 and D.3.

C. Accuracy of Theoretical Threshold

We evaluate the accuracy of our approximation in Theorem 1 by comparing the threshold obtained via Theorem 1 with the true threshold obtained by direct Monte Carlo. We consider various scenarios and parameter settings. We demonstrate the results in Fig. 6 and list the parameters below.

For Fig. 6(a)–(c), the null distribution is a one-dimensional Poisson process with intensity $\mu = 1$. We choose $\beta = 1$ as a priori, and we vary the length of the sliding time window. We set $L = 10, 50, 100$. For Fig. 6(d), we select $L = 50$, and we let $\beta = 10$. By comparing these four examples, we find that our approximated threshold is very accurate regardless of L and β.

For Fig. 6(e) and (f), the null hypothesis is a one-dimensional Hawkes process with base intensity $\mu = 1$ and influence parameter $\alpha = 0.3$, $\beta = 10$. We vary the sliding window length as $L = 100, 150$. We can see the accurate approximations as before. For Fig. 6(g) and (h), we consider a multi-dimensional case. The null distribution is a two-dimensional Poisson process with base intensity $\boldsymbol{\mu} = [0.5, 0.5]^\intercal$. We set $\beta = 1$, and we vary the window length as $L = 300$ and 400 respectively. The results demonstrate that our analytical threshold is also highly accurate in the multi-dimensional situation.

VII. Real-Data

We evaluate our online detection algorithm on real Twitter and news website data. By evaluating our log-likelihood ratio statistic on the real twittering events, we see that the statistics would increase when there is an explanatory major event in an actual scenario. By comparing the detected change points to the true major event time, we verify the accuracy and effectiveness of our proposed algorithm. In all our real experiments, we set the sliding window size as $L = 500$ minutes, and we set the kernel bandwidth β to be 1. The number of total events for the tested sequences ranges from 3000 to 15000 for every dataset.

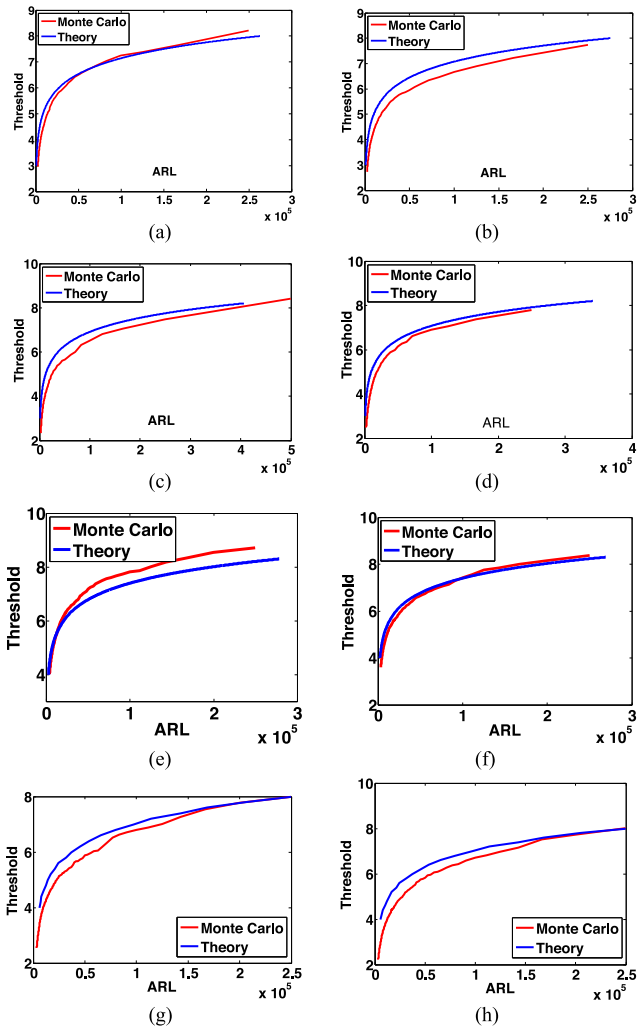

Fig. 6. Comparison of theoretical threshold obtained via Theorem 1 with simulated threshold.

A. Twitter Dataset

For the Twitter dataset, we focus on the star network topology. We create a dataset for famous users and randomly select 30 of their followers among the tens of thousands of followers. We assume that there is a star-shaped network from the celebrity to the followers, and we collect all their re/tweets in late January and early February 2016. Fig. 9(a) demonstrates the statistics computed for the account associated with a TV series named Mr. Robot. We identify that the statistics increase at approximately late January 10-th and early January 11-th. This, surprisingly corresponds to the winning of the 2016 Golden Globe Award[1]. Fig. 9(b) shows the statistics computed based on the events related to the First Lady of the USA and 30 of her randomly

[1]http://www.tvguide.com/news/golden-globe-awards-winners-2016/

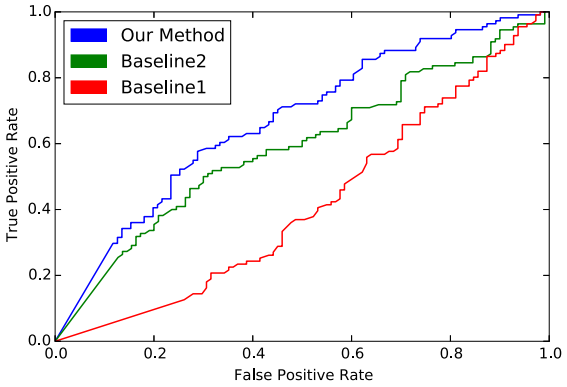

Fig. 7. AUC for twitter dataset on 116 important real-world events.

selected followers. The statistics reveal a sudden increase on the 13th of January. We find a related event - Michelle Obama stole the show during the president's final State of the Union address by wearing a marigold dress, which sold out even before the president finished the speech[2]. Fig. 9(c) is related to Suresh Raina, an Indian professional cricketer. We selected a small social circle around him as the center of a star-shaped network. We notice that he led his team to victory in an important game on January 20-th[3], which corresponds to a sharp increase in the statistics. More results for this dataset can be found in Appendix E.

We further perform sensitivity analysis using the Twitter data. We identify 116 important real-life events. Some typical examples of such events are the release of a movie/album, winning an award, and the Pulse Nightclub shooting. Next, we identify the twitter handles associated with entities representing these events. We randomly sample 50 followers from each of these accounts and obtain a star topology graph centered around each handle. We collect tweets of all users in all these networks for a window of time before and after the real-life event. For each network we compute the statistics. The AUC curves in Fig. 7 are obtained by varying the threshold. A threshold value is said to correctly identify the true change point if the statistic value to the right of the change point is greater than the threshold. This demonstrates the good performance of our algorithm against two baseline algorithms.

B. Memetracker Dataset

As a further illustration of our method, we also experiment with the Memetracker[4] dataset to detect changes in new blogs. The dataset contains the information flows captured by hyperlinks between different sites with timestamps during nine months. The dataset tracks short units of texts and short phrases, called memes, that act as signatures of topic and event propagation and diffuse over the web in mainstream media and blogs [41]. The dataset was previously used in Hawkes process models of social activity [18], [42].

TABLE III
SUMMARY INFORMATION FOR THE EXTRACTED INSTANCE FOR CHANGE-POINT DETECTION FROM THE MEMETRACKER DATASET

real-world news	n	κ	$t_{\min}$	$t_{\max}$
Obama elected president	80	11/04/08	11/02/08	11/05/08
Ceasefire in Israel	60	01/17/09	01/13/09	01/17/09
Olympics in Beijing	100	08/05/08	08/02/08	08/05/08

The keywords are highlighted in red.

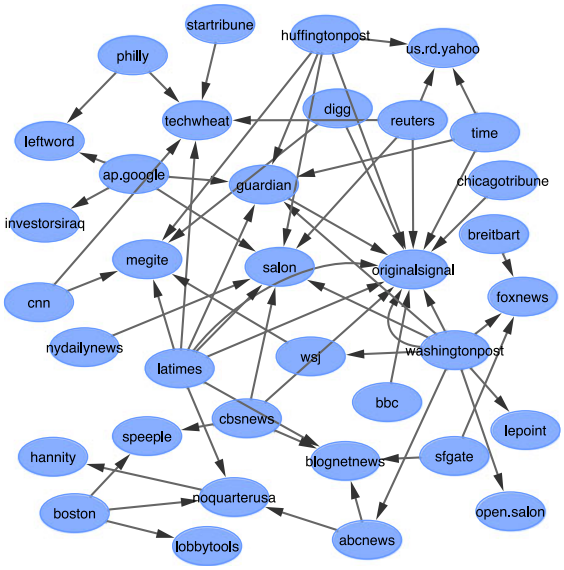

Fig. 8. Illustration of the network topology for tracking Obama's first presidential announcement.

We create three instances of change-point detection scenarios from the Memetracker dataset using the following common procedure. First, we identify a key word associated with a piece of news occurring at κ. Second, we identify the top n websites that have the most mentions of the selected key word in a time window $[t_{\min}, t_{\max}]$ around the news break time κ (i.e., $\kappa \in [t_{\min}, t_{\max}]$). Third, we extract all articles with time stamps within $[t_{\min}, t_{\max}]$ containing the keyword, and each article is treated as an event in the point process. Fourth, we construct the directed edges between the websites based on the reported linking structure. These instances correspond to real-world news whose occurrences are unexpected or uncertain and hence can cause abrupt behavior changes in the blogs. The details of these instances are shown in Table III.

The first piece of news corresponds to "Barack Obama was elected as the 44th president of the United States[5]". In this example, we also plot the largest connected component of the network, as shown in Fig. 8. It is notable that this subset includes credible news agencies such as BBC, CNN, WSJ, Huffington Post, and Guardian. As we show in Fig. 10(a), our algorithm can

[2]http://www.cnn.com/2016/01/13/living/michelle-obama-dress-marigold-narciso-rodriguez-feat/

[3]http://www.espncricinfo.com/syed-mushtaq-ali-trophy-2015-16/content/story/963891.html

[4]http://www.memetracker.org/

[5]https://en.wikipedia.org/wiki/United_States_presidential_election,_2008

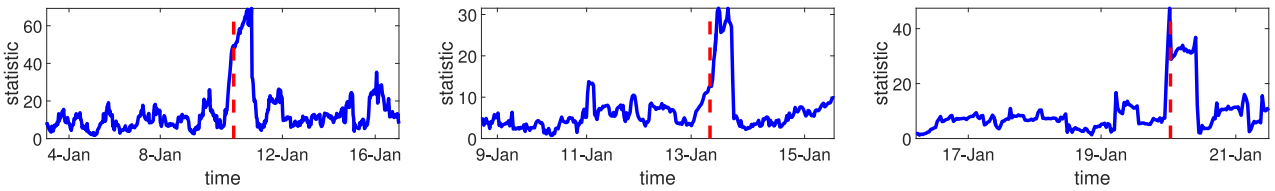

Fig. 9. Exploratory results on twitter for the detected change points: (left) Mr. Robot wins the golden globe; (middle) first lady's dress receiving attention; (right) suresh raina leads his team to victory.

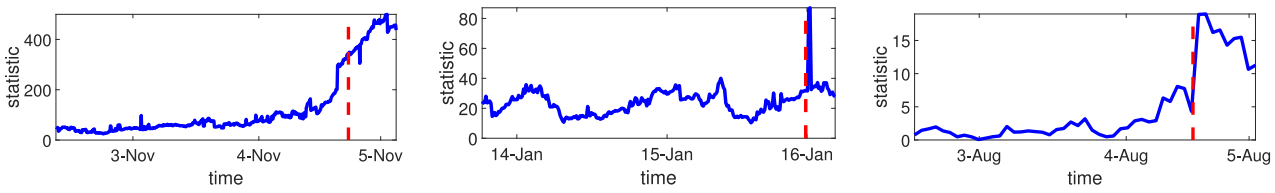

Fig. 10. Exploratory results on memetracker for the detected change points: (left) Obama wins the presidential election; (middle) israel announces a ceasefire; (right) beijing Olympics start.

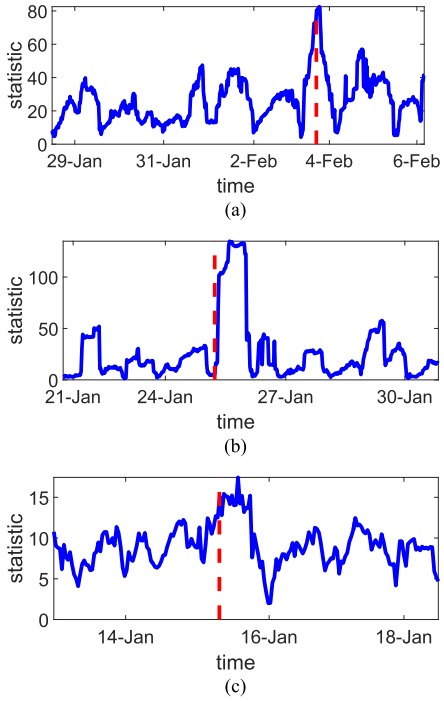

Fig. 11. Exploratory results on Twitter for the detected change points: (a) Court hearing on martin shkreli; (b) Rihanna listens to ANTI; (c) Daughter releases his new album.

successfully pinpoint a change right at the time that Obama was elected. The second piece of news corresponds to "the ceasefire in Israel-Palestine conflict back in 2009". Our algorithm detects a sharp change in the data, which is aligned closely with the time right before the peak of the war and one day before Israel announced a unilateral ceasefire during the Gaza War back in 2009[6]. The third piece of news corresponds to "the summer Olympics game in Beijing". Fig. 10(c) shows the evolution of our statistics. The change point detected is 2-3 days before the opening ceremony, when all the news websites started to talk about the event.[7]

VIII. Summary and Discussion

In this paper, we studied a set of likelihood ratio statistics for detecting changes in a sequence of event data over networks. To the best of our knowledge, our work is the first to study change-point detection for network Hawkes processes. We adopted the network Hawkes process for the event streams to model self- and mutual- excitations between nodes in the network. We cast the problem in a sequential change-point detection framework, and we derived the likelihood ratios under several models. We also presented an EM-like algorithm that can efficiently compute the likelihood ratio statistics online. The distributed nature of the algorithm enables it to be implemented on larger networks. Highly accurate theoretical approximations for the false-alarm rate, i.e., the average run length (ARL), for our algorithms are derived. We demonstrated the performance gain of our algorithms relative to two baselines, which represent the current main approaches to this problem. Finally, we also tested the performance of the proposed method on synthetic and real data.

In future work, we will extend the preliminary results for detecting the emergence of a community structure, i.e., a subset of nodes with a larger influence on each other. Such a community structure can be captured by requiring the change in the influence matrix to be low-rank [43]. To incorporate such a structure in the detection statistic, we will use a nuclear norm regularization term $\ell_{t,k,A^*} - \lambda\|A^* - A\|_*$. Here, $\lambda > 0$ is the regularization parameter, and $\|Z\|_*$ is the nuclear norm of a matrix Z, which is the sum of the absolute singular values of the matrix. Efficient online computation of the detection statistic may be achieved by extending the ADMM approach [43]. We

[6]http://news.bbc.co.uk/2/hi/middle_east/7835794.stm

[7]https://en.wikipedia.org/wiki/2008_Summer_Olympics

expect the nuclear norm regularization to lead to an additional singular value thresholding step on the difference $(A^* - A)$.

REFERENCES

[1] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1939–1947.

[2] C. Leduc and F. Roueff, "Detection and localization of change-points in high-dimensional network traffic data," *The Ann. Appl. Statist.*, vol. 3, pp. 637–662, 2009.

[3] N. Christakis and J. Fowler, "Social network sensors for early detection of contagious outbreaks," *PloS One*, vol. 5, no. 9, 2010, Art. no. e12948.

[4] M. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 561–568.

[5] S. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 913–924.

[6] R. Ratnam, J. Goense, and M. E. Nelson, "Change-point detection in neuronal spike train activity," *Neurocomputing*, vol. 52, pp. 849–855, 2003.

[7] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Distinguishing infections on different graph topologies," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3100–3120, Jun. 2015.

[8] Twitter statistics, Aug. 4, 2016. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/.

[9] J. Shen and N. Zhang, "Change-point model on nonhomogeneous poisson processes with application in copy number profiling by next-generation dna sequencing," *The Ann. Appl. Statist.*, vol. 6, no. 2, pp. 476–496, 2012.

[10] N. Zhang, B. Yakir, C. Xia, and D. Siegmund, "Scanning a poisson random field for local signals," *Ann. Appl. Stat.*, vol. 10, no. 2, pp. 726–755, 2016.

[11] T. Herberts and U. Jensen, "Optimal detection of a change point in a poisson process for different observation schemes," *Scand. J. Statist.*, vol. 31, no. 3, pp. 347–366, 2004.

[12] F. Stimberg, A. Ruttor, M. Opper, and G. Sanguinetti, "Inference in continuous-time change-point models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2717–2725.

[13] J. Pinto, T. Chahed, and E. Altman, "Trend detection in social networks using Hawkes processes," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining*, 2015, pp. 1441–1448.

[14] V. Solo and A. Pasha, "A test for independence between a point process and an analogue signal," *J. Time Series Anal.*, vol. 33, no. 5, pp. 824–840, 2012.

[15] M. Farajtabar, N. Du, M. Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2474–2482.

[16] H. Xu, M. Farajtabar, and H. Zha, "Learning granger causality for Hawkes processes," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1660–1669.

[17] S. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1–9.

[18] K. Zhou, H. Zha, and L. Song, "Learning triggering kernels for multi-dimensional Hawkes processes," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1301–1309.

[19] S. Rajaram, T. Graepel, and R. Herbrich, "Poisson-networks: A model for structured point processes," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 277–284.

[20] S. Linderman and R. Adams, "Discovering latent network structure in point process data," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.*, vol. 32, 2014, pp. 1413–1421.

[21] E. Hall and R. Willett, "Tracking dynamic point processes on networks," *IEEE Trans. Info. Theory*, vol. 62, no. 7, pp. 4327–4346, Jul. 2016.

[22] M. Farajtabar, Y. Wang, M. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1954–1962.

[23] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[24] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. London, U.K.: Chapman and Hall, 2014.

[25] A. W. Shewhart, *Economic Control of Quality of Manufactured Product*. Milwaukee, WI, USA: ASQC Quality Press, 1931.

[26] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, Jun. 1954.

[27] E. S. Page, "A test for a change in a parameter occurring at an unknown point," *Biometrika*, vol. 42, no. 3/4, pp. 523–527, 1955.

[28] W. A. Shiryaev, "On optimal methods in quickest detection problems," *Theory Prob. Appl.*, vol. 8, pp. 22–46, Jan. 1963.

[29] S. W. Roberts, "A comparison of some control chart procedures," *Technometrics*, no. 8, pp. 411–430, 1966.

[30] T. L. Lai, "Sequential changepoint detection in quality control and dynamical systems," *J. Roy. Statist. Soc. Series B (Methodological)*, vol. 57, pp. 613–658, 1995.

[31] A. Ihler, J. Hutchins, and P. Smyth, "Adaptive event detection with time-varying Poisson processes," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 207–216.

[32] Y. Mei, S. Han, and K. Tsui, "Early detection of a change in Poisson rate after accounting for population size effects," *Statistica Sinica*, vol. 21, pp. 597–624, 2011.

[33] D. Daley and D. Jones, *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. New York, NY, USA: Springer, 2007.

[34] J. G. Ransmussen, "Temporal point processes: The conditional intensity 1026 function," Lecture Notes, Aalborg Univ., Denmark, Jan. 2011.

[35] N. Barbieri, F. Bonchi, and G. Manco, "Influence-based network-oblivious community detection," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 955–960.

[36] D. Siegmund, *Sequential Analysis: Test and Confidence Intervals*. New York, NY, USA: Springer, Aug. 1985.

[37] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. London, U.K.: Chapman and Hall, 2014.

[38] A. Simma and M. Jordan, "Modeling events with cascades of poisson processes," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2012, pp. 546–555.

[39] G. Casella and C. Robert, *Monte Carlo Statistical Methods*. New York, NY, USA: Springer, 2004.

[40] G. Casella and R. L. Berger, *Statistical Inference*. Boston, MA, USA: Cengage Learning, 2001.

[41] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 497–506.

[42] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[43] K. Zhou, L. Song, and H. Zha, "Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes," in *Proc. 16th Int. Conf. Artif. Intell. Statist.*, 2013, pp. 641–649.

[44] B. Yakir, *Extremes in Random Fields: A Theory and its Applications*. Hoboken, NJ, USA: Wiley, 2013.

[45] D. Siegmund and E. S. Venkatraman, "Using the generalized likelihood ratio statistic for sequential detection of a change-point," *Ann. Statist.*, vol. 23, no. 1, pp. 255–271, 1995.

[46] D. O. Siegmund and B. Yakir, "Detecting the emergence of a signal in a noisy image," *Statist. Inference*, vol. 1, pp. 3–12, 2008.

[47] A. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[48] E. Bacry and J. Muzy, "First- and second-order statistics characterization of Hawkes processes and non-parametric estimation," *IEEE Trans. Info. Theory*, vol. 62, no. 4, pp. 2184–2202, 2016.

[49] D. Daley and D. Jones, "Scoring probability forecasts for point processes: The entropy score and information gain," *J. Appl. Probab.*, vol. 41, pp. 297–312, 2004.

[50] D. Jones, "Probabilities and information gain for earthquake forecasting," *Selected Papers From Volume 30 of Vychislitel'naya Seysmologiya*. Washington, DC, USA: American Geophysical Union, 1998, pp. 104–114.

Authors' photographs and biographies not available at the time of publication.