

Contents lists available at ScienceDirect

Physica D



journal homepage: www.elsevier.com/locate/physd

Influence network reconstruction from discrete time-series of count data modelled by multidimensional Hawkes processes

Naratip Santitissadeekorn^{a,*}, Martin Short^b, David J.B. Lloyd^{a,c}

^a Department of Mathematics, University of Surrey, Guildford, UK

^b Department of Mathematics, Georgia Institute of Technology, USA

^c Centre for Criminology, University of Surrey, Guildford, UK

ARTICLE INFO

Communicated by Katarzyna Sznajd-Weron

Dataset link: https://github.com/naratips/EM-MM-ExPKF

MSC: 62F15 62F30 62M20

Keywords: Influence network Causal network Nonlinear filtering Hawkes process Count data Expectation maximization

1. Introduction

This work is motivated by conventional applications of continuoustime Hawkes processes utilized to model the temporal clustering as well as mutual excitation network driven by the timestamp data, *i.e.* the times of events. The continuous-time point-process Hawkes model was first introduced by Hawkes [1] to capture a self-excitation process, used particularly for seismic events [2]. Since then, it has been extended to multivariate Hawkes process to model the mutual excitation structure or influence network structure. This development has led to emerging applications of point-process Hawkes model to seismic analysis [2,3], urban crime analysis [4–7] social network analysis [8–15], financial time-series analysis [16–19], contagious disease network [20–22] and deep learning network [23–27].

To reconstruct the influence network from a time series of timestamp data, most of the above work typically used the Expectation– Maximization (EM) or Minimization–Majorization (MM) framework to construct a surrogate function (*i.e.*, tight upper bound function) for the negative log-likelihood function. The main advantage of this approach is that it may help decouple the parameter space when optimizing the

ABSTRACT

Identifying key influencers from time series data without a known prior network structure is a challenging problem in various applications, from crime analysis to social media. While much work has focused on eventbased time series (timestamp) data, fewer methods address count data, where event counts are recorded in fixed intervals. We develop network inference methods for both batched and sequential count data. Here the strong network connection represents the key influences among the nodes. We introduce an ensemblebased algorithm, rooted in the expectation-maximization (EM) framework, and demonstrate its utility to identify node dynamics and connections through a discrete-time Cox or Hawkes process. For the linear multidimensional Hawkes model, we employ a minimization-majorization (MM) approach, allowing for parallelized inference of networks. For sequential inference, we use a second-order approximation of the Bayesian inference problem. Under certain assumptions, a rank-1 update for the covariance matrix reduces computational costs. We validate our methods on synthetic data and real-world datasets, including email communications within European academic communities. Our approach effectively reconstructs underlying networks, accounting for both excitation and diffusion influences. This work advances network reconstruction from count data in real-world scenarios.

surrogate function, accelerating the computational task. For a simple excitation kernel, such as the exponential decay kernel, a closed-form method for the parameter update can be derived. A nonparametric excitation kernel can also be used within the EM and MM approach, where the Euler–Lagrange equation can be derived for the optimization of the surrogate function [28] and the regularization to promote sparsity [29]. Other techniques have also been developed to estimate non-parametric kernels; see for instance [16,30,31]. A fully-Bayesian, parallel inference algorithm was also developed in [32] to model the excitation structure by random graph models which allows conjugate prior for efficient inference via Markov chain Monte Carlo.

The influence network within multi-dimensional Hawkes models can also be linked to Granger-causality in temporal point processes [33]. In the context of this framework, an event generated by x_j is considered to "Granger-cause" the event associated with x_i if the likelihood function of events in $x_i(t)$, given the history of all events up to time *t*, decreases when the history of events generated by x_j is omitted. The application of the multivariate Hawkes model in the

* Corresponding author. *E-mail address:* n.santitissadeekorn@surrey.ac.uk (N. Santitissadeekorn).

https://doi.org/10.1016/j.physd.2025.134705

Received 24 August 2024; Received in revised form 13 February 2025; Accepted 18 April 2025 Available online 5 May 2025 0167-2789/© 2025 Published by Elsevier B.V. context of Granger causality provides interpretability of the results. It was demonstrated in [31] that x_j does not Granger-cause x_i when the (pairwise) excitation kernel used by x_j to 'excite' x_i is zero. Applications of Hawkes model to discover Granger-causality were investigated with real-world data in [34–36]. However, this work is limited to certain conditional independence of the excitation process, while some data may exhibit inhibition or interaction. Additionally, to the best of our knowledge, the connection between the excitation (or influence) network and Granger causality has not been extended to the case of count data modelled by the discrete-time Hawkes process. Therefore, the influence network derived from count data may or may not correspond to the causal network.

Similar to the timestamp data, a time series of count data may exhibit self-excitation, wherein a high count is often followed by several higher counts, e.g., in a time series of epileptic seizure counts [37]. This data is often easier to collect and more common in applications, for instance in epidemiology, where one can only sensibly collate count data. Moreover, multiple time series may demonstrate an "influencing" characteristic, where a high count from one series is followed by high counts in others. For instance, a cluster of earthquakes in a particular region could trigger seismic activities in adjacent regions, while incidents occurring in one area of a city could lead to similar occurrences in other areas, e.g., urban crimes [6,38]. We can conceptualize the sources of these multiple time series as nodes in a network, and by uncovering the influence structure of such a network, we can gain insights into the evolving dynamics of the network over time, such as the emergence of synchronization of node dynamics. However, there are no methods (to the best of the author's knowledge) for carrying out the network reconstruction problem from time series count data.

The primary aim of this work is to identify an influence network from a time series consisting of count data that opens up more real-world applications of network reconstruction via Hawkes-type processes. The count data inference problem is significantly more challenging since data is aggregated and therefore there is a loss of information relative to the time-stamp data which needs to be accounted for in the inference problem. Recent research [39] has developed a novel Hawkes-type non-homogeneous Poisson process for count data, with a deterministic intensity function equal to the expectation of a Hawkes process intensity and parameters are fitted based on the intervalcensored likelihood function derived therein. Here we adopt different models that utilize a discrete-time, multidimensional Cox or Hawkes process to model the excitation effects among nodes driven by count data. Within this framework, the magnitude of the influence can be quantified by the excitation rate parameters incorporated into the model. Therefore, the task of identifying the influence network reduces to estimating the parameters of the Cox or Hawkes model. This work presents three distinct methodologies for parameter estimation for count data problems: (1) Ensemble-based EM, (2) Minimization-Majorization (MM) technique and (3) a sequential algorithm based on approximate second-order filtering.

For the ensemble-based EM, the Hawkes process can take a general "state-space" form (*e.g.*, doubly stochastic Poisson point process) that consists of state dynamical system and observation equation. Within the context of EM, the "missing" data is the unobserved sample path of the state. Therefore, the E-step requires a Monte Carlo sampling of the sample paths. This approach was previously used in the context of the model identification for the Kalman filter [40,41]. We demonstrate how this idea can be applied to network reconstruction of a small network.

The MM algorithm is developed for batch data inference. For this algorithm, we limit the node dynamic model to a discrete-time dynamical system analogous to the exponential-decay kernel of the multivariate Hawkes process. We show how one can derive an iterative method that minimizes a surrogate function such that the surrogate function is a "tight" upper bound of the negative log-likelihood function for count data. We show how the iterative method can be parallelized. The sequential algorithm based on approximate second-order filtering, called the extended Poisson–Kalman filter (ExPKF), is derived by approximating the mean and covariance matrix of the posterior density for the same dynamical systems Hawkes model used for the MM algorithm. We show how this leads to an efficient method under an assumption, where one can use a rank-1 update for the update of the covariance matrix with parallelization.

Our main contributions are the development of the foundations for a systematic approach to dealing with network influence reconstruction from count data. We present methods that deal with either complex state–space models for small networks or linear Hawkes processes for large networks. The ensemble-based EM method also captures uncertainty quantification of the intensity estimate, while ExPKF provides a second-order moment for the network estimate. Uncertainty quantification is very important in network reconstruction applications so that one can gain an understanding of the uncertainty of a link between two nodes occurring. This work opens up new avenues of research involving count data collected on networks, and the development of new methods for more general or other types of stochastic processes.

The paper is outlined as follows. In Section 2, we develop the ensemble-based EM algorithm for small networks. We then focus on a moderate-size network for a count-data model, inspired by the discretization of the exponential decay kernel of the continuous-time Hawkes process. The MM algorithm is developed for batch data inference in Section 3 and the extended Poisson–Kalman filter (ExPKF) is derived in Section 4. In Section 5, we demonstrate the validity of the proposed methods to reconstruct the influence network with various numerical experiments with known ground truths. We also demonstrate the utility of the method on large real-world email network data in Section 6 and conclude in Section 7.

2. Ensemble-based EM

We are interested in an inhomogeneous Poisson point process on a network with *m* nodes, where the conditional intensity λ_k^i at the *i*th node is assumed to be a constant in the *k*-th time interval (t_k, t_{k+1}) . In other words, if ΔN_k^i is the number of events observed for the *i*-th node at the *k*-th time interval, we assume that $Pr(\Delta N_k^i \mid \lambda_k^i)$ is a Poisson probability with mean $\lambda_k^i \delta t_k$ where $\delta t_k = t_{k+1} - t_k$. The intensity λ_k^i depends on a *n*-dimensional parameter vector, $\theta^i := [\theta^{i,1}, \dots, \theta^{i,n}]^T$. We concatenate all vectors θ^i to form a parameter vector θ , *i.e.*, $\theta := [\theta^{1}, \dots, \theta^{m}]$

For simplicity, all the time intervals are assumed to have the same length δt . At any given time step k, we assume conditional independence so that the conditional joint density is given by

$$p(\underbrace{\Delta N_k^1, \dots, \Delta N_k^m}_{\equiv \Delta N_k} | \lambda_k^1, \dots, \lambda_k^m) \propto \prod_{i=1}^m (\lambda_k^i)^{\Delta N_k^i} \exp(-\lambda_k^i \delta t).$$
(2.1)

Let $\Delta N_{1:K} := [\Delta N_1, \dots, \Delta N_K]$ denote time-series of count data up to the time step *K* for all nodes. The log-likelihood function is then given by

$$\mathbf{L}(\theta) := \log p(\Delta N_{1:K} \mid \theta) = \sum_{i=1}^{m} \sum_{k=1}^{K} \log(\lambda_k^i(\theta^i)) \Delta N_k^i - \delta t \sum_{i=1}^{m} \sum_{k=1}^{K} \lambda_k^i(\theta^i) + C,$$
(2.2)

where *C* is independent of θ . The maximum likelihood method estimates the model parameter vector θ (in a parameter space Θ) by maximizing the log-likelihood function

$$\widehat{\theta} := \arg \max_{\theta \in \Theta} \mathbf{L}(\theta). \tag{2.3}$$

We assume that the discrete-time dynamic of λ_k^i is governed by a stochastic process of an unobserved "state" vector denoted by $\mathbf{x}_k := [\mathbf{x}_k^1, \dots, \mathbf{x}_k^m]$ with $\mathbf{x}_k^i \in \mathbf{R}^d$:

$$\mathbf{x}_k = \Psi(\mathbf{x}_{k-1}; \mathbf{p}) + \eta_k, \tag{2.4}$$

where a function Ψ can be nonlinear, **p** is a fixed parameter vector and $\eta_k \sim N(\mathbf{0}, \mathbf{Q})$. The conditional intensity, λ_k^i , is assumed to be a function of the state vector, *i.e.*,

$$\lambda_k^i = h(\mathbf{x}_k^i; \mathbf{q}), \tag{2.5}$$

where the link function of observation, *h*, is usually nonlinear and **q** is a fixed parameter vector. In this setting, the parameter vector is given by the augmented vector $\theta = [\mathbf{p} \ \mathbf{q}]$. The so-called complete data likelihood function is the joint probability density $p(\mathbf{x}_{0:K}, \Delta N_{1:K} | \theta)$, where $\mathbf{x}_{0:k}$ denotes the sequence of \mathbf{x}_0 up to \mathbf{x}_k . The maximization problem for (2.2) can also be expressed by

$$\widehat{\theta} := \arg \max_{\theta \in \Theta} \log \int p\left(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta\right) d\mathbf{x}_{0:K}.$$
(2.6)

We adopt the EM framework to construct an iterative algorithm for the state–space model to avoid a direct integration of the above joint density. The construction of our algorithm follows a similar approach for model identification for the Kalman filter presented in [40,41]. To this end, we denote the parameter estimate after κ iterations by $\theta^{(\kappa)}$. In the EM approach, we have to design a tight lower-bound function (*i.e.* minorization) that would be more tractable for maximization than the original marginal likelihood function. For the current case, a tight lower-bound (or surrogate) function for maximization is given by

$$\mathcal{Q}\left(\theta;\theta^{(\kappa)}\right) = \int p\left(\mathbf{x}_{0:K} \mid \Delta N_{1:K}, \theta^{(\kappa)}\right) \log p\left(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta\right) d\mathbf{x}_{0:K}$$

= $\mathbb{E}\left[\log p\left(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta\right)\right].$ (2.7)

which represents the E-step of the EM algorithm. The M-step then solves the maximization problem

$$\theta^{(\kappa+1)} := \arg \max_{\theta \in \Theta} Q\left(\theta; \theta^{(\kappa)}\right).$$
(2.8)

Under the (first-order) Markovian assumption, we can decompose the surrogate function $Q(\theta, \theta^{(\kappa)})$ by

$$Q\left(\theta,\theta^{(\kappa)}\right) = Q_{0}\left(\theta,\theta^{(\kappa)}\right) + Q_{x}\left(\theta,\theta^{(\kappa)}\right) + Q_{\Delta N}\left(\theta,\theta^{(\kappa)}\right),$$

$$Q_{0}\left(\theta,\theta^{(\kappa)}\right) = \int p\left(\mathbf{x}_{0} \mid \Delta N_{1:K},\theta^{(\kappa)}\right) \log p\left(\mathbf{x}_{0} \mid \theta\right) d\mathbf{x}_{0},$$

$$= \mathbb{E}\left[\log p\left(\mathbf{x}_{0} \mid \theta\right)\right],$$

$$Q_{x}\left(\theta,\theta^{(\kappa)}\right) = \sum_{k=1}^{K} \int p\left(\mathbf{x}_{k},\mathbf{x}_{k-1} \mid \Delta N_{1:K},\theta^{(\kappa)}\right) \log p\left(\mathbf{x}_{k} \mid \mathbf{x}_{k-1},\Delta N_{1:K},\theta\right)$$

$$\times d\mathbf{x}_{k}d\mathbf{x}_{k-1},$$

$$= \sum_{k=1}^{K} \mathbb{E}\left[\log p\left(\mathbf{x}_{k} \mid \mathbf{x}_{k-1},\Delta N_{1:K},\theta\right)\right],$$

$$Q_{\Delta N}\left(\theta,\theta^{(\kappa)}\right) = \sum_{k=1}^{K} \int p\left(\mathbf{x}_{k} \mid \Delta N_{1:K},\theta^{(\kappa)}\right) \log p\left(\Delta N_{k} \mid \mathbf{x}_{k},\theta\right) d\mathbf{x}_{k},$$

$$= \sum_{k=1}^{K} \mathbb{E}\left[\log p\left(\Delta N_{k} \mid \mathbf{x}_{k},\theta\right)\right].$$
(2.9)

To maximize Q, we must assume the availability of $p(\mathbf{x}_0 | \theta)$, $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \Delta N_{1:K}, \theta)$, and $p(\Delta N_k | \mathbf{x}_k, \theta)$. The initial density $p(\mathbf{x}_0 | \theta)$ may depend on the model parameter in general, depending on how we would like to generate the initial density for the state. If not, Q_0 can be excluded from the maximization. The transition density $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \Delta N_{1:K}, \theta)$ will depend on the model (2.4). Assuming a normal distribution for η_k in (2.4), $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \Delta N_{1:K}, \theta)$ is also normal. The likelihood function $p(\Delta N_k | \mathbf{x}_k, \theta)$ then follows the assumption in (2.1).

The expression in (2.7) suggests that if we can sample from $p(\mathbf{x}_{0:K} | \Delta N_{1:K}, \theta^{(\kappa)})$, we can then estimate all the expectations in (2.9) using the sample paths, which we denote by $\mathbf{x}_{0:K}^s$. The superscript *s* stands for "smoothing" which will be explained below. The efficiency of the EM algorithm in this setting will depend strongly on the design

of the path sampling technique. We will use the forward filteringbackward sampling procedure to obtain samples approximately from the joint smoothing distribution [41], which is a combination of particle filtering (PF) and backward simulation smoother (BSS) to generate $\mathbf{x}_{0:K}^{s}$.

Particle filtering is a sequential Monte Carlo (SMC) technique for non-linear filtering. In the current application, it can be used to sample $p(\mathbf{x}_k \mid \Delta N_{1:k}, \theta^{(\kappa)})$, *i.e.*, only the observation up to the time step k is used to estimate \mathbf{x}_k . It enjoys great flexibility but suffers from filtering degeneracy, where most of the sample weights become zero as time increases. A resampling is required to mitigate this issue. We will still, however, employ it in our work for a low-dimensional problem. A brief discussion of PF algorithm is provided in Appendix A. An extensive review of SMC and PF can be found in many review literature, to name a few here, [42–45].

Suppose that we have obtained the weighted, filtered particle $\left(\mathbf{x}_{k}^{f(i)}, \mathbf{w}_{k}^{f(i)}\right)$ for $i = 1, \dots, N_{f}$, where N_{f} is the number of particles used in PF. The particles approximate $p(\mathbf{x}_k \mid \Delta N_{1:k}, \theta^{(\kappa)})$ but the EM algorithm requires $p(\mathbf{x}_k \mid \Delta N_{1:K}, \theta^{(\kappa)})$ for any k = 1, ..., K. The BSS uses the filtered particles $\left(\mathbf{x}_k^{f(i)}, \mathbf{w}_k^{f(i)}\right)$ to generate the smoothing particles, $\left(\mathbf{x}_{\iota}^{s(\ell)}, \mathbf{w}_{\iota}^{s(\ell)}\right)$ for $\ell = 1, \dots, N_s$, where N_s and N_f can be different. The algorithm for BSS is also provided in Appendix A. We can then approximate the expectations in (2.9) based on $(\mathbf{x}_k^{s(\ell)}, \mathbf{w}_k^{s(\ell)})$. Maximization of Q in (2.9) to find $\theta^{(\kappa+1)}$ is then carried out numerically. We use the function fmincon in MATLAB to optimize Q. A useful byproduct of this approach is that the ensemble of sample paths of the conditional intensity $\lambda_{1\cdot K}^{i}$ can be directly computed from the particles $\left(\mathbf{x}_{k}^{s(\ell)}, \mathbf{w}_{k}^{s(\ell)}\right)$ at the last iteration of the EM algorithm. The ensemblebased EM method allows for uncertainty quantification of the intensity paths. In the subsequent subsections, we will demonstrate how the ensemble-based EM may be used for some discrete-time Hawkes model that can be represented in a state-space form.

2.1. Log-Gaussian Cox process (LGCP)

We consider a univariate LGCP (i.e. m = 1) given by

$$x_{k} = \underbrace{(1 - \omega_{1}\delta t)x_{k-1} + \omega_{1}\mu\delta t}_{:=\Psi_{x}(x_{k-1})} + \epsilon\sqrt{\delta t}\eta_{k},$$

$$g_{k} = \underbrace{(1 - \omega_{2}\delta t)g_{k-1} + \alpha\Delta N_{k-1}}_{:=\Psi_{g}(g_{k-1})},$$

$$\lambda_{k} = \exp(x_{k}) + g_{k}.$$
(2.10)

We assume η_k has the standard normal distribution N(0, 1). The parameter vector is $\theta = [\mu, \omega_1, \epsilon, \alpha, \omega_2]$ and the state variable is $x_k \in \mathbb{R}$. All parameters are assumed to be positive. Hence, "inhibition" (*i.e.* $\alpha < 0$) is not considered in this work.

We first consider a synthetic experiment with a ground truth $\theta^* = [1.5, 0.5, 2.5, 0.5, 1.5]$ and simulate $\Delta N_{1:K}$ and $\lambda_{1:K}$ for K = 4000 with $\delta t = 0.1$ and initial condition $x_0 = 1.5$ and $g_0 = 0$. We initialize the EM algorithm with parameter vector $\theta^{(0)} = [3, 0.25, 1.25, 0.25, 0.75]$. At the κ -th iteration, the E-step requires a prior sample of the state vector \mathbf{x}_0 . We use a prior assumption $x_0 \sim N(\mu^{(\kappa)}, \epsilon^{(\kappa)} \delta t)$ and set $g_0 = 0$. The number of particles is $N_f = 400$ and we set $N_s = 0.25N_f$ in our experiment. After obtaining the smoothing particle $\mathbf{x}^{s(\kappa)}$ from E-step (using a combination of PF and BSS as explained in Appendix A), we can evaluate the Q-function in (2.9). The Q-function (after omitting



Fig. 1. (Top) Comparison between filtered particles $\lambda_{0:k}^{(\ell)}$, smoothed particles $\lambda_{0:k}^{s(\ell)}$ and ground truth. The simulated data is also shown in the bar plot. (Bottom, left) The progress of the relative error of the parameter vector θ over 16 EM iterations. (Bottom, right) The propagation of the relative error for the conditional intensity. Note that we have fixed the number of EM iterations to 16; no stopping criteria have been implemented in this example.

the terms irrelevant to maximization) has the following form:

$$\begin{aligned} \mathcal{Q}_{0}(\theta, \theta^{(\kappa)}) &= -\frac{1}{2N_{s}} \sum_{\ell=1}^{N_{s}} \frac{\left(x_{0}^{s(\ell)} - \mu^{(\kappa)}\right)^{2}}{\delta t} \\ \mathcal{Q}_{\mathbf{x}}(\theta, \theta^{(\kappa)}) &= -\frac{1}{2N_{s}} \sum_{\ell=1}^{N_{s}} \left[\sum_{k=1}^{K} \frac{\left(x_{k}^{s(\ell)} - \Psi_{x}\left(x_{k-1}^{s(\ell)}\right)\right)^{2}}{\epsilon^{2} \delta t} + K \log \epsilon \right] \end{aligned}$$
(2.11)
$$\mathcal{Q}_{\Delta N}(\theta, \theta^{(\kappa)}) &= \frac{1}{2N_{s}} \sum_{\ell=1}^{N_{s}} \left[\sum_{k=1}^{K} \Delta N_{k} \log \lambda_{k} - \exp(\lambda_{k}) \delta t \right] \end{aligned}$$

When K is large, the term $Q_0(\theta, \theta^{(\kappa)})$ in (2.9) can be neglected. Furthermore, Q_x depends on only μ, ω_1, ϵ and $Q_{\Delta N}$ depends only on α, ω_2 . We can then solve the two maximization problems in parallel to obtain $\theta^{(\kappa+1)}$. Maximizing Q_x has a closed-form solution, see Appendix B, and can be readily computed. However, since Q_x and $Q_{\Delta N}$ are maximized in parallel, numerically maximizing $Q_{\Delta N}$ becomes a bottleneck to the speed of the algorithm. For this experiment, we numerically maximize both Q_x and $Q_{\Delta N}$ using the interior point algorithm implemented in fmincon in MATLAB [46]. The constraint optimization is required to ensure the positive values of parameters.

The results are shown in Fig. 1.

We also compare the filtered intensity $\lambda_{0:k}^{f(\ell)}$ with the smoothed intensity $\lambda_{0:k}^{s(\ell)}$ in Fig. 1, which are computed from $x_{0:k}^{f(\ell)}$, $x_{0:k}^{s(\ell)}$ using the observation equation in (2.10), respectively. Fig. 1 clearly shows that the smoothed particles have smaller variation (or uncertainty) than that of the filtered particles. The relative error for the parameter vector decays quickly after a few steps and then becomes stable. We compute the relative error for each particle $\lambda_{0:k}^{s(\ell)}$ and report the average relative error in Fig. 1.

2.2. Logistic LGCP

We modify the conditional intensity function in (2.10) to

$$\lambda_k = h\left(\exp(x_k) + g_k\right), \qquad h(z) = \frac{A}{1 + B\exp(-z)}.$$
(2.12)

This modification incorporates the upper bound $\lambda_k \leq A$ to the conditional intensity. The parameter vector is $\theta = [\mu, \omega_1, \epsilon, \alpha, \omega_2, A, B]$ and the state vector is $\mathbf{x}_k \in \mathbb{R}$. The required *Q*-function is the same as (2.11).

We select a ground truth $\theta^* = [0.5, 0.5, 0.25, 9, 0.5, 12, 4]$ and simulate $\Delta_{1:K}$ and $\lambda_{1:K}$ for K = 2000 with $\delta t = 0.1$ and initial condition $x_0 = 1$ and $g_0 = 0$. We initialize the EM algorithm with parameter vector $\theta^{(0)} = [1, 0.25, 0.5, 4.5, 1, 24, 8]$. As shown in Fig. 2, a fast reduction of both relative errors is obtained at the beginning and then slows down as the algorithm converges. This observation is common for MM algorithms in general. In the early stages of the iterations, the parameter estimates are often far from the nearest stationary point (local maximum), resulting in relatively large "gradient components" (in an informal sense). Since the EM algorithm can be roughly interpreted as performing coordinate ascent in the latent or complete-data space [47–49], these large gradients lead to significant updates in the parameters and, consequently, substantial increases in the likelihood.

As the algorithm approaches a local maximum (or saddle point), the likelihood surface becomes flatter due to reduced curvature. This causes the parameter updates to shrink, as the re-estimated values in each iteration undergo smaller changes once they are closer to the optimal fit. As a result, the likelihood's improvements diminish with each subsequent iteration.

2.3. Log-Gaussian Cox process (LGCP) on a small network

We consider a multivariate extension of LGCP on a network where the links of the network describe an "influence" structure through a



Fig. 2. (Top) Comparison between smoothed particles $\lambda_{0:k}^{(\ell)}$ and ground truth for the Logistic LGCP. The simulated data is also shown in the bar plot. (Bottom, left) The progress of the relative error of the parameter vector θ over 35 EM iterations where the change in the relative error of θ is lower than 10^{-5} . (Bottom, right) The progress of the relative error for the conditional intensity.

(pairwise) excitation process. We demonstrate the utility of the ensemble EM approach for the multivariate LGCP on the *m* nodes, denoted by x^j for j = 1, ..., m, that has the following form:

$$x_{k+1}^{i} = \underbrace{\left[(1-\eta^{i})x_{k}^{i} + \eta^{i}\sum_{j\neq i} x_{k}^{j} \right] (1-\omega_{1}^{i}\delta t) + \omega_{1}^{i}\mu^{i}\delta t}_{m} + \epsilon^{i}\sqrt{\delta t}\zeta_{k},$$

$$\underbrace{:=\Psi_{x}^{i}(x_{k}^{i})}_{m}$$
(2.13)

$$g_{k+1}^{i} = (1 - \omega_{2}^{i} \delta t) g_{k}^{i} + \sum_{j=1}^{m} \alpha^{ij} \Delta N_{k}^{j}, \quad \lambda_{k+1}^{i} = \exp(x_{k+1}^{i}) + g_{k+1}^{i},$$

where η^j is the diffusion coefficient strength at each node, ω_1^j and ω_2^j are the decay rates at each node, α^{ij} are the excitation coupling parameters, $\epsilon^j > 0$ and $\zeta_k \sim \mathcal{N}(0, 1)$ describes the noise at each node. The Q-function is defined similarly to (2.11). A (homogeneous) diffusion effect is included in the dynamics of x_k^j and the event-driven excitation process between nodes is incorporated in the dynamic of g_k^j . The mutual excitation of g_k^j is driven not only by the count data from the node j itself, but also by all other nodes. Both diffusion and excitation contribute to the increment of the conditional intensity of other nodes in the network in the next time step. The state variable is the vector $\mathbf{x}_k = [x_k^1, \ldots, x_k^m] \in \mathbb{R}^m$. Note that if there is no diffusion term, we could compute the smoothed path of each x_k^j in parallel for the E-step.

We consider an experiment with the following parameters: $\delta t = 0.1$, $\omega_1^j = 0.5/\delta t, \omega_2^j = 0.9/\delta t, \eta^j = 0.1, \mu^j = 0.5, \epsilon^j = 0.125$ and m = 3. The ground truth of the mutual excitation structure α^{ij} is given in Fig. 5. We test the experiment with different simulated data lengths K = 500, 1000, 2000. The initial ensemble for x_0^j is drawn independently from $N(0, 5\epsilon^{j})$ using N = 600 particles. The initial structure of the network is set to $\alpha^{ij} = 0.9$, *i.e.*, a fully connected network with a uniform excitation rate of 0.9. The experimental results are shown in Figs. 3 to 5. The errors of the parameter estimation for various values of K are shown in Fig. 3, all of which exhibit fast error reduction in the first step and then slowly decrease afterwards, similar to the univariate case shown in Section 2.1. The smoothed path at the final EM step is shown in Fig. 4 for K = 2000, which demonstrates a good estimate of the true intensity. The results for the other data lengths are similar. Most importantly, the network structure, which is the main interest of this work, can be accurately captured as shown in Fig. 5 if the data length is sufficiently long enough. Note that we have tested several cases and found similar results when the initial guess of the parameters is "close

enough" to the true parameters in a sense that the stability of the model is sustained; if the initial parameter is not "close enough" to the true values, the method may fail to converge.

3. Majorization-minimization (MM)

In this section, we focus only on the discrete-time model analogous to the exponential-decay kernel of the multivariate Hawkes process. In particular, the conditional intensity λ_{k}^{i} is given by

$$\lambda_{k+1}^{i} = \mu^{i} + (\lambda_{k}^{i} - \mu^{i})\gamma^{i} + \sum_{j=1}^{m} \alpha^{ij} \Delta N_{k}^{j}, \quad i = 1, \dots, m,$$
(3.1)

where $\mu^i > 0$, $\alpha^{ij} \ge 0$, $0 < \gamma^i < 1$ and $\Delta N_0^j = 0$. The parameter μ^i represents the baseline rate where the number of events are endogenously generated based on a Poisson distribution with the mean μ^i . We assume that the initial condition is the same as the baseline, *i.e.*, $\lambda_0^i = \mu^i$. The parameter α^{ij} for $i \neq j$ models increases in the likelihood to generate more counts for the *i*th node immediately after observing counts for the *j*th node. The parameter γ^i is the decay rate of λ_k^i toward the baseline rate. The recursive model (3.1) can also be rewritten in a closed form by

$$\lambda_{k}^{i} = \mu^{i} + \sum_{\ell=1}^{k} B_{\ell}^{i} \left(\gamma^{i} \right)^{k-\ell} \quad \& \quad B_{\ell}^{i} := \sum_{j=1}^{m} \alpha^{ij} \Delta N_{\ell-1}^{j}.$$
(3.2)

This section provides an iterative procedure to minimize the negative log-likelihood function of the discrete-time Hawkes process. In the continuous-time setting, where the timestamp data is available, the branching process can be (artificially) assumed to define the missing data (*i.e.* immigrant, ancestor or descendent) and the complete likelihood function. However, it is difficult to replicate this idea in the count data setting. Instead, we employ the MM technique to derive an EM-like algorithm for the multivariate Hawkes model driven by count data.

We first present a derivation of the MM algorithm for the case that the decay rate $\gamma^i = \gamma$ is fixed and known. Due to the conditional independence of λ_k^i for all i = 1, ..., m and k = 1, ..., K given the parameters, we can separately minimize the negative log-likelihood function of each node to estimate μ^i and α^{ij} . For this reason, when estimating parameters for the *i*th node, we will avoid the notation clutter by omitting the superscript *i* from the subsequent discussion



Fig. 3. (Left) Relative error of the parameter vector at each EM-step.(Right) Relative error of the conditional intensity at each EM-step. We stop at 5 iterations when the change in the relative error of θ is lower than 10^{-5} .



Fig. 4. From the top to bottom, the plot shows the smoothed path at the final step of the EM for node 1 to 3, respectively. For a clear visualization, only part of the trajectory is shown at the time step k = 100-500. The bar plot beneath the intensity shows the simulated count data for each node.



Fig. 5. Estimated values of α^{ij} for various data length and the ground truth. The *i*-th row and *j*- column in the plot indicates α^{ij} .

in this section, *e.g.*, α^{ij} will be written by α^j without confusion. The negative log-likelihood for a given node is

$$\mathcal{L}(\theta) := -\sum_{k=1}^{K} \log(\lambda_k) \Delta N_k + \delta t \sum_{k=1}^{K} \lambda_k + C,$$
(3.3)

where $\theta = (\mu, \alpha^1 \dots, \alpha^m)$ and *C* is a constant. The MM algorithm is an iterative technique that updates the estimation of $\theta^{(n+1)}$ at the n + 1 iteration by minimizing a surrogate function $Q(\theta \mid \theta^{(n)})$. For the minimization problem, the surrogate function is chosen to be a "tight" upper bound function so that $Q(\theta \mid \theta^{(n)}) \ge \mathcal{L}(\theta)$ for any θ and $Q(\theta^{(n)} | \theta^{(n)}) = \mathcal{L}(\theta^{(n)})$. Without loss of generality, we may assume $\delta t = 1$ here and ignore the constant *C* as well.

By applying Jensen's inequality, a tight upper bound function of $-\log(\lambda_k(\theta))$ in (3.3), denoted by $Q_k(\theta \mid \theta^{(n)})$, can be constructed by

$$-\log(\lambda_{k}) \leq Q_{k}(\theta \mid \theta^{(n)}) := -\frac{\mu^{(n)}}{\lambda_{k}^{(n)}} \log\left(\frac{\lambda_{k}^{(n)}}{\mu^{(n)}}\mu\right) - \sum_{l=0}^{k-1} \sum_{j=1}^{m} \frac{\phi_{klj}^{(n)}}{\lambda_{k}^{(n)}} \log\left(\frac{\lambda_{k}^{(n)}}{\phi_{klj}^{(n)}}\phi_{klj}\right),$$
(3.4)

where $\phi_{klj} := \alpha^j (\gamma^j)^{k-l-1} \Delta N_k^j$. Clearly, we have $Q_k(\theta^{(n)} | \theta^{(n)}) = -\log(\lambda_k(\theta^{(n)}))$. Here $\lambda_k^{(n)}$ is a short hand for $\lambda_k^{(\theta(n))}$ and likewise for $\mu^{(n)}$

and $\phi_{kli}^{(n)}$. We now define a tight upper bound function of $\mathcal{L}(\theta)$ by

$$Q(\theta \mid \theta^{(n)}) = -\sum_{k=1}^{K} Q_k(\theta \mid \theta^{(n)}) \Delta N_k + \sum_{k=1}^{K} \lambda_k.$$
(3.5)

We obtain the update equations by setting the derivative of Q to zero to yield

$$\mu^{(n+1)} = \frac{1}{K} \sum_{k=1}^{K} \frac{\mu^{(n)}}{\lambda_k^{(n)}} \Delta N_k, \quad \left(\alpha^j\right)^{(n+1)} = \frac{\sum_{k=1}^{K} \sum_{l=0}^{k-1} \frac{\phi_{klj}^{(n)}}{\lambda_k^{(n)}} \Delta N_k}{(1+\gamma)\mathcal{N}^j + \Delta N_{K-1}^j},$$
(3.6)

where $\mathcal{N}^{j} = \Delta N_{1}^{j} + \dots + \Delta N_{K-2}^{j}$. Note that we make a second-order approximation of small γ^{n} so that $\gamma^{n} \approx 0$ for $n \geq 2$ in order to obtain the update equation of $(\alpha^{j})^{(n+1)}$. Notice that the update equations for each parameter are decoupled, so this step can be computed in parallel.

In general, we can also derive MM algorithm that allows the decay rate to be dependent for every node pair. In other words, the parameter vector for the *i*th node is given by $\theta = (\mu^i, \alpha^{i1}, \dots, \alpha^{im}, \gamma^{i1}, \dots, \gamma^{im})$. Again, we will omit the superscript *i* in the algorithm below because of independence of parameters between nodes. Hence, we will write $\theta = (\mu, \alpha^1, \dots, \alpha^m, \gamma^1, \dots, \gamma^m)$. To obtain independent update equations for each parameter (similarly to (3.6)), further work is required to deal with the second term in (3.6). Through the Arithmetic–Geometric Mean inequality, the upper bound function can be obtained by

$$Q(\theta \mid \theta^{(n)}) = -\sum_{k=0}^{K} Q_k(\theta \mid \theta^{(n)}) \Delta N_k + K\mu + \sum_{j=1}^{m} H^j \mathcal{N}^j,$$
(3.7)

where

$$H^{j} = \frac{(\alpha^{j})^{(n)}}{2\left(1 + (\gamma^{j})^{(n)}\right)} (1 + \gamma^{j})^{2} + \frac{2\left(1 + (\gamma^{j})^{(n)}\right)}{(\alpha^{j})^{(n)}} (\alpha^{j})^{2}.$$
 (3.8)

Note that we use a second-order approximation of small γ^{j} to obtain the upper bound function (3.7). By setting the derivative of *Q* in (3.7) to zero, we obtain the following update equations

$$\mu^{(n+1)} = \frac{1}{K} \sum_{k=1}^{K} \frac{\mu^{(n)}}{\lambda_k^{(n)}} \Delta N_k,$$

$$(\alpha^j)^{(n+1)} = -B^j + \frac{\sqrt{(B^j)^2 + 4A^j C^j}}{2A^j},$$

$$(\gamma^j)^{(n+1)} = -D^j + \frac{\sqrt{(D^j)^2 + 4D^j E^j}}{2D^j},$$
(3.9)

where

$$A^{j} = \mathcal{N}^{j} \frac{1 + (\gamma^{j})^{(n)}}{(\alpha^{j})^{(n)}}, \qquad B^{j} = \Delta N_{K-1}^{j}, \qquad C^{j} = \sum_{k=1}^{N} \sum_{l=0}^{k-1} \frac{\Delta N_{k} \phi_{klj}^{(n)}}{\lambda_{k}^{(n)}},$$
$$D^{j} = \mathcal{N}^{j} \frac{(\alpha^{j})^{(n)}}{1 + (\gamma^{j})^{(n)}}, \qquad E^{j} = \sum_{k=1}^{K} \sum_{l=0}^{k-1} (k-l-1) \frac{\Delta N_{k} \phi_{klj}^{(n)}}{\lambda_{k}^{(n)}}.$$
(3.10)

In practice, a regularization scheme may be required to obtain useful results. For the system with known decay rate, we apply a regularization only to the baseline rate update in (3.6) and keep the update equation of the excitation network unchanged. One of the simplest ways to do this is to change (3.6) to

$$\mu^{(n+1)} = \frac{1}{K+b} \left(\sum_{k=1}^{K} \frac{\mu^{(n)}}{\lambda_k^{(n)}} \Delta N_k + a - 1 \right),$$
(3.11)

for some hyperparameters a, b > 0. This is equivalent to solving the Maximum a posterior (MAP) problem with a gamma prior distribution of μ with shape parameter a and inverse scale parameter b. We will discuss how we choose hyperparameters a and b in the synthetic experiment in the subsequent section.

Similarly, we may regularize the update equations for both μ and γ^{j} in (3.9) by applying a gamma prior distribution for μ with the

hyperparameters *a* and *b* and a beta prior distribution for each γ^j with hyperparameters *c* and *d*. Note that for simplicity, we assume the prior distribution of all parameters to be independent and use the same value of the hyperparameters for all γ^j . The beta distribution is selected to constrain γ^j within the desired interval (0, 1). The update equation for $\mu^{(n+1)}$ under this regularization will be the same as (3.11). However, to update γ^j , we must solve a quartic polynomial of the following form

$$-D^{j}x^{4} + (D^{j} + E^{j})x^{2} + (c - d - E^{j})x - a = 0,$$
(3.12)

where *x* denotes $(\gamma^j)^{(n+1)}$ and E^j and D^j are defined in (3.10). We can either try to solve (3.12) analytically or numerically. In our work, we solve this numerically at every iteration.

4. ExPKF for the discrete-time Hawkes model

The MM algorithm in the previous section is designed to estimate the model parameters in batch. Alternatively, we can also develop a sequential procedure to estimate the parameters. This section presents a sequential (second-order) approximation of the posterior density $p(\theta_k \mid \Delta N_{1:k})$. In particular, we are interested in approximating only the mean and covariance matrix associated with $p(\theta_k \mid \Delta N_{1:k})$. Suppose that we have obtained the approximation of the mean and covariance matrix at the time step k - 1 denoted $\bar{\theta}_{k-1}$ and \mathbf{P}_{k-1} , respectively. Based on this approximation, we assume a prediction model to generate a prior mean, denoted by $\bar{\theta}_{k|k-1}$, and prior covariance, denoted by $\mathbf{P}_{k|k-1}$. Following the derivation in [50], a second-order approximation of $p(\theta_k \mid \Delta N_{1:k})$ (called the Extended Poisson–Kalman Filter (ExPKF)) has the following mean $\bar{\theta}_k$ and covariance update \mathbf{P}_k given by

$$\mathbf{P}_{k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^{m} \left[\left(\frac{\partial \log \lambda_{k}^{i}}{\partial \theta_{k}} \right) \left(\frac{\partial \log \lambda_{k}^{i}}{\partial \theta_{k}} \right)^{\mathsf{T}} \lambda_{k}^{i} \delta t - (\Delta N_{k}^{i} - \lambda_{k}^{i} \delta t) \frac{\partial^{2} \log \lambda_{k}^{i}}{\partial^{2} \theta_{k}} \right],$$
$$\bar{\theta}_{k} = \bar{\theta}_{k|k-1} + \mathbf{P}_{k} \sum_{i=1}^{m} \left[\left(\frac{\partial \log \lambda_{k}^{i}}{\partial \theta_{k}} \right) (\Delta N_{k}^{i} - \lambda_{k}^{i} \delta t) \right],$$
(4.1)

where the gradient vector $\frac{\partial \log \lambda_k^i}{\partial \theta_k}$ and Hessian matrix $\frac{\partial^2 \log \lambda_k^i}{\partial^2 \theta_k}$ are both evaluated at $\bar{\theta}_{k|k-1}$.

The filtering equation (4.1) can be used to sequentially approximate the parameters of the model (3.1). We will assume that γ^i is fixed for a reason that will be explained later; hence there are m + 1 unknown parameters for each λ_{k}^i . To ensure the positivity of the parameters, we will estimate the log-transformed parameter instead,

$$\theta_k^i := [\log \mu_k^i, \log \alpha_k^{i1} \dots, \log \alpha_k^{im}]^{\mathsf{T}}.$$
(4.2)

Therefore, we have $\theta_k \in \mathbf{R}^{m(m+1)}$. If θ_k is meant to be a static parameter vector, it is reasonable to assume the following random-walk model, $\theta_k = \theta_{k-1} + \eta_k$, where $\eta_k \sim N(0, \mathbf{Q}_k)$. Thus, we have $\bar{\theta}_{k|k-1} = \bar{\theta}_{k-1}$ and $\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1} + \mathbf{Q}_k$. Let $S_k^i = \Delta N_k^i + \gamma^i \Delta N_{k-1}^i + \dots + (\gamma^i)^{k-1} \Delta N_0^i$, which can be recursively computed by $S_{k+1}^i = \gamma^i S_k^i + \Delta N_{k+1}^i$. It can be checked that the gradient vector required by (4.1) has the following form

$$\frac{\partial \log \lambda_k^i}{\partial \theta_k} = \left[\frac{\partial \log \lambda_k^i}{\partial \theta_k^1}, \frac{\partial \log \lambda_k^i}{\partial \theta_k^2}, \dots, \frac{\partial \log \lambda_k^i}{\partial \theta_k^m} \right]^{\dagger},$$
(4.3)

where

$$\frac{\partial \log \lambda_k^i}{\partial \theta_k^i} = \begin{cases} \frac{1}{\lambda_k^i} \left[e^{\mu_k^i}, S_k^1 e^{\alpha_k^{j1}}, \dots, S_k^m e^{\alpha_k^{jm}} \right], & \text{if } i = j, \\ \mathbf{0}, & \text{if } i \neq j. \end{cases}$$
(4.4)

Thus, only the *i*-th "block" of $\frac{\partial \log \lambda_k^i}{\partial \theta_k}$ is non-zero. Recall, that γ^i is fixed and known. It follows that the Hessian has a simple form, given by

$$\left(\frac{\partial \log \lambda_k^i}{\partial \theta_k}\right) \left(\frac{\partial \log \lambda_k^i}{\partial \theta_k}\right)^{\mathsf{T}} = -\frac{\partial^2 \log \lambda_k^i}{\partial^2 \theta_k} + \Lambda^{(i)},\tag{4.5}$$

where $\Lambda^{(i)}$ is a diagonal matrix with the diagonal vector $\frac{\partial \log \lambda_k^i}{\partial \theta_k}$. Substituting the above results into (4.1) yields

$$\mathbf{P}_{k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^{m} \Delta N_{k}^{i} \left(\frac{\partial \log \lambda_{k}^{i}}{\partial \theta_{k}}\right) \left(\frac{\partial \log \lambda_{k}^{i}}{\partial \theta_{k}}\right)^{\mathsf{T}} + \left(\lambda_{k}^{i} \delta t - \Delta N_{k}^{i}\right) \Lambda^{(j)}.$$
(4.6)

With the form in (4.6), a rank-1 update can be efficiently used to compute \mathbf{P}_k^{-1} . Also, if $\mathbf{P}_{k|k-1}$ has a block-diagonal form where each block corresponds to the parameters of each node, \mathbf{P}_k^{-1} will also have the same block-diagonal structure where the *i*-th block corresponds to the parameters of the *i*th node. Therefore, by ensuring that $\mathbf{P}_{k|k-1}$ has the same block-diagonal structure, \mathbf{P}_k will inherit the same block-diagonal structure, \mathbf{P}_k will inherit the same block-diagonal structure, \mathbf{P}_k will inherit the same block-diagonal structure. Consequently, the update system (4.1) can be implemented for each node in parallel. To this end, we will always enforce the block-diagonal structure to \mathbf{P}_0 and \mathbf{Q}_k in all of our numerical experiments.

5. Synthetic data tests

5.1. Test experiment

We set up this experiment to generate synthetic test data for three different scenarios based on (3.1). The true network has m = 9 nodes with the following baseline rates: $\mu^1 = 5, \mu^2 = 4.6, \mu^3 = 4.2, \mu^4 = 0.5, \mu^5 = 0.46, \mu^6 = 0.42, \mu^7 = 0.38, \mu^8 = 0.34$, and $\mu^9 = 0.3$.

We assume $\gamma^i := \gamma = 0.175$ for all nodes. Note that the model (3.1) is stable if the magnitude of the largest eigenvalue of $\delta t(1-\gamma)^{-1}$ A is less than 1, where A is a matrix with α^{ij} entries on the *i*-th row and *j*-th column. The value of $\gamma = 0.175$ is selected so that the model (3.1) is stable for all three different ground truths of the excitation matrix, A, examined in this experiment. The structures of three different ground truths are shown in Fig. 6 representing the different scenarios: only self-excitation (top row), localized excitation (middle row), and random excitation structure (bottom row). We generate the test data with $\delta t = 1$ for various data lengths, K = 2000, 4000, 8000, 20000. A test data is simulated by running the model (3.1) and sampling ΔN_k^i from a Poisson distribution with the mean rate $\lambda_k^i \delta t$ with the initial condition $\lambda_0^i = \mu^i$.

The results for MM method with and without regularization are shown in Figs. 6 and 7, respectively. For any *i*-th node, the hyperparameter for the regularized MM algorithm is set to $a = 0.5 \overline{N^i} K$, where $\overline{N^i}$ is the average count of the *i*-th node over K time steps and b = K. This is equivalent to choosing the gamma prior distribution with mean $0.5N^i$ (half of the total count for the given node) and variance $0.5N^i/K$. Although the selection of these prior parameter values may be arbitrary in general, we based our choice of the prior mean on the reasoning that the baseline rate should be lower than the data average due to the creation of certain events by excitation. In addition, the variance is sufficiently small to ensure that the prior information, or regularization, is not dominated by the sample size. We also set c = 2.5K and d =10.25K for the beta prior distribution, which gives the mean 1/6 and variance in the order of o(1/K). We make this selection to prolong the impact of excitation by avoiding a decay rate that is too close to 1. This selection is again arbitrary. For the remainder of our work, we will utilize this approach to select prior information for the MM algorithm. Our results, as demonstrated in Figs. 6 and 7, clearly illustrate that the use of regularization enables the algorithm to produce significantly more accurate outcomes that closely approximate the ground truth.

We use the same simulated data to test ExPKF. Recall that we must assume a fixed decay rate for ExPKF to achieve efficient algorithms via the rank-1 update. We will discuss this issue later in this section. We set $\mathbf{P}_0 = 10^{-4}\mathbf{I}$ and $\mathbf{Q}_k = 10^{-5}\mathbf{I}$ for all k = 1, ..., K. The initial guess of the μ^j is set to be half of the data average of the *j*-th node. Fig. 8 illustrates that the ExPKF algorithm can achieve accuracy comparable to that of the MM algorithm with regularization in capturing network structures.

However, ExPKF requires a known decay rate value for all nodes, which was used in this experiment. In practice, decay rates may vary and be unknown for different nodes. To overcome this issue, we explore a method to identify the optimal decay rate γ^i , assuming a uniform decay rate for all nodes. Specifically, we perform a one-dimensional maximization based on the average log predictive probability (2.1). To calculate the average log predictive probability, we use the parameter estimate obtained from the previous time step ($\bar{\theta}_{k-1}$) and evaluate (2.1) at time step k, averaging over all time steps. Fig. 9 demonstrates that maximizing the predictive probability yields the optimal decay rate.

Moreover, the network structure appears to be highly robust to parameter misspecification, as shown in Fig. 10 for the ground truth 3 scenario. Despite the presence of spurious links caused by incorrect decay rates, the primary network structure closely resembles the true structure. Similar results are observed for the other ground truths, although they are not presented in this study.

5.2. Estimation under model misspecification

In this experiment, we will generate test data from an agent-based model (ABM) on a set of nodes featuring an excitation network structure, which will then be estimated within the Hawkes model; hence, a model misspecification problem. We adopt a model inspired by the ABM in [6] to simulate the random movement of an "agent" between nodes through the network edges. During a time interval $(t, t + \delta t)$, an agent located at a node *s* creates a number of "events" independently, following a Poisson probability with mean $A_s(t)\delta t$. The total number of events generated at *s* during $(t, t + \delta t)$ is denoted by $E_s(t)$. The discrete-time dynamic of $A_s(t)$ is given by $A_s(t) = A_s^0 + B_s(t)$, where A_s^0 is a static, node-dependent baseline rate, and $B_s(t)$ is a dynamic component that follows the rule:

$$B_{s}(t+\delta t) = \left[(1-\eta_{s})B_{s}(t) + \eta_{s} \sum_{s' \sim s} B_{s'}(t) \right] (1-\omega_{s}\delta t) + \sum_{s' \sim s} w(s,s')E_{s'}(t).$$
(5.1)

The interpretation of these parameters is listed below:

- $0 < \omega_s < 1$ is the node-dependent decay rate;
- $0 < \eta_s < 1$ is the node-dependent diffusion rate;
- $w(s, s') \ge 0$ defines the strength of the event-driven excitation rate that the node s' has on the node s and we write $s' \sim s$ if w(s, s') > 0.

If an agent generates one or more events, the agent will be removed from the simulation. Otherwise, the agent will move from a node *s* to s'' such that w(s, s'') > 0 based on a discrete probability distribution

$$q(s, s''; t) := \frac{A_{s''}(t)}{\sum_{s' \sim s} A_{s'}(t)}.$$
(5.2)

Prior to a simulation of the next time step, new agents will be independently created for each node according to a Poisson distribution with mean $\Gamma_s \delta t$ where Γ_s is a fixed parameter. The key difference between the Hawkes and ABM models is that the Hawkes model's network structure provides only data-driven excitation, whereas the ABM network structure determines the agents' excitation, diffusion, and probabilistic movement of agents.

The focus of this experiment is on the structure of w(s, s'). To this end, we define an "influence" matrix W such that its *s*-th row and *s'*-th column entry is w(s, s'), and reconstruct the pattern of non-zero elements of W using the ExPKF and MM methods. It is important to note that no ground truth is available for this experiment. Although the dynamics of the ABM exhibit similarities to the Hawkes process in terms of influence effects, diffusion effects are also present in the ABM but absent from the Hawkes process. Nevertheless, we anticipate that the influence structure of the Hawkes model will be akin to that of the ABM when fitting the Hawkes model with ABM-simulated count data.



Fig. 6. The estimated network structure obtained from the MM algorithm without regularization for three different ground truths. The length of the data is varied to demonstrate the convergence to the ground truth.



Fig. 7. The estimated network structure obtained from the MM algorithm with regularization for three different ground truths. The length of the data is varied to demonstrate the convergence to the ground truth.



Fig. 8. The estimated network structure obtained from the ExPKF algorithm using a true decay rate. The length of the data is varied to demonstrate the convergence to the ground truth.



Fig. 9. Comparing the predictive log likelihood for various values of decay rate, γ . The vertical line indicates the true value of the decay rate. The data with length of 20000 is used to produce this result.



Fig. 10. Comparing the network structure for the ground truth 3 when the value of decay rate, γ , is incorrectly specified.

We generated data of various lengths based on the same influence matrix pattern, represented by W with 64 nodes, as illustrated in Fig. 11. The influence structure is sparse and irreducible, and we considered

two cases. In Case 1, we set all non-zero influences to w(s, s') = 3 for $s, s' \in 1, ..., 64$, with $B_s(0) = 0$ for all s and $\delta t = 0.05$. Other static parameters were spatially uniform: $\omega_s = 5$, $\eta_s = 0.25$, and $\Gamma_s = 3$, for all



Fig. 11. Data simulated from ABM model. The left most plot illustrates the influence matrix *W*. The second plot from the left shows the cumulative number of events for all nodes. The third plot is the frequency distribution of the count. The right most plot is the sample covariance matrix.

s. Additional information on the simulated data for Case 1 is provided in Fig. 11. The cumulative counts of the data can be grouped into three distinct categories based on the unique values of the row sums of W. Most time intervals exhibit either no events or a single event, and the sample covariance of the count time-series indicates little correlation among the nodes.

We use the simulated data to test ExPKF and MM. For ExPKF, we set $\mathbf{Q}_k = 10^{-5}\mathbf{I}$ for all k and $\mathbf{P}_0 = 10^{-4}\mathbf{I}$ for all nodes. Again, we set the initial value $\bar{\theta}_0$ in the same manner as done with the previous experiment in Section 5.1. Fig. 12 shows the estimated \boldsymbol{W} based on ExPKF and MM. Interestingly, both methods can correctly reconstruct the pattern of the influence matrix \boldsymbol{W} despite the model misspecification. The results improve with longer data series. Notably, ExPKF produces a result with less "noise" in the part that is supposed to have zero influence.

We also examine Case 2 where we change the non-zero influences to w(s, s') = 0.5, $\eta_s = 1$, $\Gamma_s = 0.5$, keeping all the other parameter values the same. While the network pattern in Case 1 is manifested mostly through the excitation process (*i.e.* larger values of w(s, s') and smaller values of η_s and Γ_s), the Case 2 has a weak excitation and generation rate of the new agents but increased diffusion. This change would make it more difficult to detect the influence pattern. Nonetheless, the network structure can still be detected as displayed in Fig. 13. However, the data length required to achieve a good result has to be longer than that of the Case 1; note that the average number of counts per time step is 0.15 for Case 1 but only 0.025 for Case 2.

The errors based on the Frobenius norm and the Hellinger distance for different data lengths are analysed in Fig. 14. When computing both error measures, we normalize W so that the sum of all elements is 1. This is necessary since we have no numerical ground truth to compare against and should evaluate the error based only on the network structure. Both error measures suggest a small improvement of the ExPKF results, which is consistent with the visualization of Fig. 12.

6. Email network data

6.1. Small email network

In this section, we analyse the Ikenet dataset consisting of log files from email transactions between 22 anonymized officers at West Point Military Academy over a one-year period, which is available to download via https://github.com/naratips/Ikenet.git. The dataset contains the time-stamps of outgoing emails and their corresponding receivers. Table 1 displays the top 9 sender-receiver pairs in the dataset, ranked by the number of out-going emails. The data clearly highlights the overwhelmingly large amount of mutual email correspondence for the pairs (9, 18) and (11, 22). Previous studies on this dataset have utilized information about both the sender and recipients of emails [15,51]. The Hawkes model with the exponential decay rate was used in [51] Table 1

Top 9	sender-receiver	for	the	Ikenet	data	ranked	by	the	total	number	of	outgoing
emails.												

sender	18	9	22	11	15	8	18	13	18
receiver	9	18	11	22	13	18	8	17	22
% of total	6.95	5.97	3.97	3.01	1.96	1.89	1.87	1.78	1.75

where the rate of sending out an email for a given node is driven by the events of emails received by the given node. For our experiment, we will focus solely on information about the outgoing emails. Therefore, the "influence" in our analysis can be interpreted as the effect of the number of emails sent out by other nodes on the rate of sending out emails (without any knowledge of the recipients). To demonstrate the methods in terms of count data, we aggregated the timestamp data of outgoing emails into a time-series of count data with a uniform temporal interval of dt = 0.1 days. Fig. 15 shows the total number of counts for each node and the proportion of non-zero counts per time step. Note that despite having the number of outgoing emails as large as node 18, node 13 is not among the top pairs (9, 18) and (11, 22).

As shown in Fig. 16, networks constructed by MM and ExPKF are very similar. By comparing the dominant connections in the network with Table 1, we can see that the influence network highlights the top sender–receiver pairs (9, 18) and (11, 22) even though no knowledge of the email recipient network is used in the experiment.

6.2. Large email network

In this section, we carry out an experiment on a real-world (anonymized) email timestamp data similar to the previous section but at a much larger size. The original data can be found from the following link: https://snap.stanford.edu/data/email-Eu-core-temporal.html. However, we focus only on the outgoing emails and we "cleaned up" the data by removing a continuous period of extremely low count due to missing data, weekends and holidays. The cleaned-up data has 545 "nodes" and 61821 intervals (each interval is one hour long) in total with approximately 97% of zero counts, 2% of one count per interval and the rest of the data has more than one count. Fig. 17 shows the top 50 nodes with the highest number of emails sent and the cumulative count for all nodes.

We test only ExPKF for this experiment since it requires less computer memory and runs faster than MM on our computational resources. We set the initial values $\alpha^{ij} = 0.1$ and initialize μ^i by the average count on the *i*-th node. We set the decay rate $\beta = 0.15$ for all nodes; we tested a few other values, and the results are qualitatively the same. We present the estimated influence network in Fig. 18. It is clear that the network is extremely sparse. We can identify only 5 edges that would suggest a strong influence. Although the number of emails sent by node 308 is close to the median value, we can identify its relatively



Fig. 12. MM and ExPKF estimation of the influence matrix using different data length for Case 1, which has a strong excitation effects.



Fig. 13. MM and ExPKF estimation of the influence matrix using different data length for Case 2, which has a weak excitation effect but strong diffusion.



Fig. 14. MM and ExPKF Error for different data lengths. (Left) Frobenius norm. (Right) Hellinger distance. Two cases are presented: Case 1 (plotted with the circle markers) corresponds to Fig. 12 and Case 2 (plotted with the square markers) corresponds to Fig. 13.



Fig. 15. The total count of emails sent by each node (Left) and proportion of non-zero counts (Right).



Fig. 16. Influence networks associated with the 22-node Ikenet email data constructed by MM algorithm (Left) and ExPKF algorithm (Right).



Fig. 17. (Left) Histogram of the top 50 users by number of emails sent. The black bar is associated with node 308, which is identified by ExPKF as the most influential node. (Right) The cumulative counts of all nodes. The node 308 has the cumulative counts shown in a black solid curve. The cumulative counts of the nodes influenced by node 308 are plotted in the black dash curves.

higher influence on a few other nodes, all of which have a low number of sent-out emails.

7. Conclusion

This work presents a significant development in foundations and methods for reconstructing influence networks from a time-series of count data through parameter estimation of discrete-time, multivariate Hawkes or Cox processes. Developing methods for inference for count data is important as it is very common in applications when timestamp data is not available or does not make sense to collect, *e.g.* in epidemiology applications, but this area is significantly less developed than for timestamp data where, to the best of the authors' knowledge, there were previously no methods for dealing with count data. Despite count data having less information than the time-stamp data, we find that network reconstruction is still possible. We demonstrate an application of the ensemble-based EM algorithm for certain doubly-stochastic processes (such as Log-Gaussian Cox process) that can be presented in state–space form. Our implementation is based on the forward filteringbackward smoothing procedure using the bootstrap particle filter for the forward filtering, which is followed by backward smoothing simulation. We demonstrated the that the Ensemble-EM method is able to carry out the network reconstruction through synthetic experiments with known ground truths for small networks.

We observe that the parameter settings in our synthetic experiments are carefully adjusted to ensure bounded intensity. When the intensity grows unbounded, network recovery becomes infeasible within our experimental framework. This challenge translates to practical scenarios where intensity values become excessively large. A similar issue is discussed in [22,39], where the process eigenvalue approaches 1. It is evident that further research advancements are necessary to address this limitation effectively.

This paper lays the foundations for other smoothing methods that could be used instead of the forward-filtering-backward smoothing approach for the ensemble-based EM depending on the structure of the state model and the observational likelihood. For example, it was



Fig. 18. (Left) Histogram of the edge weights, *i.e.*, α_{ii} for the large email network. (Right) The subnetwork with the edges weight above 0.15.

demonstrated in [52] that it is possible to bypass entirely the backward smoothing to compute expectations in the setting of an online EM method. This would significantly reduce the memory storage requirements for the backward smoothing simulation and allow for larger networks to be handled. Future work will look at the development of the ensemble-based approximate filtering using similar concepts from the ensemble-based Kalman smoother (EnKS) developed in geophysical applications [53]. The EnKS uses the ensemble to approximate the density of the one-step push-forward state. This ensemble is then updated to fit the observation problem under the approximately linear model and Gaussian observational noise. In the current context, however, the observation equation can be nonlinear in the parameters and may not be close to Gaussian for a time-series with small counts. Further study in this direction to improve the E-step of the ensemble-based EM will open up applications to the influence network reconstruction problem with more complicated state-space models.

We then presented the MM-based algorithm and the ExPKF algorithm to handle real-world applications when the linear Hawkes model is a reasonable assumption. The MM algorithm is designed to handle large batch data. We select a tight upper bound so that each parameter can be updated separately in a parallel manner. The ExPKF algorithm is a sequential approach that assumes a known decay rate for the Hawkes model. This key assumption enables the rank-1 update in the algorithm to avoid the costly inversion of a large matrix and by estimating each node independently, our algorithm can be efficiently parallelized. Investigation of the ExPKF on synthetic data again showed excellent results in determining the hidden network structure.

We demonstrated the performance of the methods using numerical experiments with known ground truths for both perfect and imperfect model scenarios, and both ExPKF and MM algorithms can recover the influence network structure when compared with the ground truth with good estimates of the strengths of the connections in the network. Several exciting areas for future research include looking at when the ExPKF algorithm becomes expensive for general Hawkes models where the inversion of the Hessian term in ExPKF cannot be performed via a rank-1 update; hence it can become a numerical issue for large-scale problems. For MM algorithms, a tight upper bound must be specifically designed for a given model. Therefore, for more general models, finding a tight upper bound allowing for parallel update of parameters is an interesting area to investigate. One thing this work opens up is the possibility of network reconstruction for applications in social networks and neural networks that hitherto remained out of reach with current methods.

CRediT authorship contribution statement

Naratip Santitissadeekorn: Writing – original draft, Visualization, Validation, Methodology, Investigation, Funding acquisition, Formal

analysis, Conceptualization. **Martin Short:** Writing – review & editing, Validation, Investigation. **David J.B. Lloyd:** Writing – review & editing, Validation, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Naratip SANTITISSADEEKORN reports financial support was provided by Engineering and Physical Sciences Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Naratip Santitissadeekorn is supported by the EPSRC grant EP/W02084X/1.

Appendix A. Forward filtering-backward smoothing

We provide a brief review of the particle filtering (PF) and backward smoothing simulation (BSS) used to generate smoothed particles required to evaluate the surrogate function in (2.9) for the ensemble-based EM algorithm.

Particle Filter (PF): Let $\mathbf{x}_k^{f(\ell)}$ and $w_k^{f(\ell)}$ denote, respectively, the ℓ -th (filtered) particle and its corresponding normalized weight at time step k = 0, 1, ..., K, where K is the length of the time-series of count data.

- 1. Initialization: Randomly generate N_f particles from an initial distribution $\mathbf{x}_0^{f(\ell)} \sim p(\mathbf{x}_0)$ and set initial weights: $w_0^{f(\ell)} = 1/N_f$ for $i = 1, \dots, N_f$.
- 2. Repeat this step for k = 1, ..., K,
 - (a) Draw random samples from the conditional predictive distribution, denoted by x_k^(ℓ) based on (2.4), *i.e.*, x_k^(ℓ) ~ N(x_k^(ℓ); Ψ(x_{k-1}^{f(ℓ)}), Q), and then generate the predictive conditional intensity λ_k^{j,(ℓ)} for all nodes j = 1,..., *m* based on (2.5)
 - (b) Update (unnormalized) weights based on the likelihood model

$$\widetilde{w}_k^{f(\ell)} \propto w_{k-1}^{f(\ell)} \prod_{j=1}^m (\lambda_k^{j,(\ell)})^{\Delta N_k^i} \exp(-\lambda_k^{j,(\ell)} \delta t).$$

and then normalize the weight by $w_k^{f(\ell)} := w_k^{f(\ell)} \left(\sum_{\ell=1}^N \tilde{w}_k^{f(\ell)}\right)^{-1}$.

(c) Perform resampling to add additional Monte Carlo variation when the effective sample size is low. We use the criteria below:

$$N_{eff} := \left(\sum_{\ell=1}^{N_f} \left(w_k^{(\ell)} \right)^2 \right)^{-1} < 0.5 N_f.$$

There are a number of methods for resampling. For simplicity, we use the systematic sampling algorithm described in [45], which costs $O(N_f)$. At the end of this step, we obtain $\mathbf{x}_k^{f(\ell)}$ and $w_k^{f(\ell)}$.

Backward Smoothing Simulation (BSS):

Let $\mathbf{x}_{k}^{s(\ell)}$ denote the particle of the ℓ -th smoothing path at time step k = 0, 1, ..., K.

- 1. Initialization: Suppose $\mathbf{x}_{0:K}^{f(\ell)}$ and $w_{0:K}^{f(\ell)}$, for $i = 1, ..., N_f$, have been computed from (and stored during) the filtering process. Select $\mathbf{x}_K^{s(\ell)} = \mathbf{x}_K^{f(\ell)}$ with probability $w_K^{f(\ell)}$.
- 2. Repeat this step (backward in time) for k = K 1, ..., 0,
 - (a) For $\ell = 1, ..., N_f$, calculate new weights according to (2.4)

$$w_k^{s(\ell)} \propto w_k^{f(\ell)} p\left(\mathbf{x}_{k+1}^{s(\ell)} \mid \mathbf{x}_k^{f(\ell)}\right) = w_k^{f(\ell)} N\left(\mathbf{x}_k^{f(\ell)}; \mathbf{x}_{k+1}^{s(\ell)}, \mathbf{Q}\right)$$

(b) Randomly select $\mathbf{x}_{k}^{s(\ell)} = \mathbf{x}_{k}^{f(\ell)}$ with probability $w_{k}^{s(\ell)}$. Repeat Step 1 and Step 2 N_{S} times, where N_{s} is the desired number of smoothing trajectories.

The smoothing trajectories, $\mathbf{x}_{0:K}^{s(\ell)}$ for $\ell = 1, ..., N_s$ will then be used to estimate the parameters in the M-step of the EM algorithm, see again (2.9).

Appendix B. Maximization of Q_r in (2.11)

By neglecting Q_0 in (2.11), we can find $\mu_1^{(\kappa+1)}, \omega_1^{(\kappa+1)}$ and $\epsilon^{(\kappa+1)}$ by maximizing Q_x only. Let $\beta = (1 - \omega_1 \delta t)$ and $\gamma = \omega_1 \mu$. We can rewrite Q_x by

$$\begin{aligned} \mathcal{Q}_{x} &= -\frac{1}{2N_{s}\epsilon^{2}\delta t} \|\mathbf{y} - \mathbf{A}\mathbf{z}\|^{2} - \frac{1}{2}K\log\epsilon, \\ \text{where } \mathbf{z} &= [\beta, \gamma]^{\mathsf{T}}, \, \mathbf{y} = \begin{bmatrix} x_{1}^{s(1)}, \dots, x_{K}^{s(N_{s})}, \dots, x_{K}^{s(1)}, \dots, x_{K}^{s(N_{s})} \end{bmatrix}^{\mathsf{T}} \text{ and} \\ \mathbf{A}^{\mathsf{T}} &= \begin{pmatrix} x_{0}^{s(1)} & \cdots & x_{K-1}^{s(1)} & \cdots & x_{0}^{s(N_{s})} & \cdots & x_{K-1}^{s(N_{s})} \\ \delta t & \cdots & \delta t & \cdots & \delta t & \cdots & \delta t \end{pmatrix}. \end{aligned}$$

Thus, maximizing Q_x is equivalent to finding z to "solves" the problem min $||\mathbf{y} - \mathbf{A}\mathbf{z}||^2$, which is nothing but the normal equation if $\mathbf{A}^{\top}\mathbf{A}$ is full-rank or other techniques may be required to regularize the solution. However, since z has to be positive, a quadratic programming should be used if unconstrained minimization fails to produce the desired positive solution.

After obtaining $\mathbf{z}^{(\kappa+1)} = [\beta^{(\kappa+1)}, \gamma^{(\kappa+1)}]^{\mathsf{T}}$, we can recover $\omega_1^{(\kappa+1)}, \mu^{(\kappa+1)}$ from $\beta^{(\kappa+1)}, \gamma^{(\kappa+1)}$. We also find the maximizing solution of ϵ by

$$\epsilon^{(\kappa+1)} = \frac{1}{N_S K} \sum_{\ell=1}^{N_s} \sum_{k=1}^K \left(x_k^{s(\ell)} - \Psi_x(x_{k-1}^{s(\ell)}) \right)^2,$$

using $\omega_1^{(\kappa+1)}, \mu^{(\kappa+1)}$ in Ψ_x above.

Data availability

Code will be made available at https://github.com/naratips/EM-MM-ExPKF when accepted.

References

- A.G. Hawkes, Spectra of some self-exciting and mutually exciting point processes, Biometrika 58 (1) (1971) 83–90.
- [2] Y. Ogata, Seismicity analysis through point-process modeling: A review, Pure Appl. Geophys. 155 (1999) 471–507.
- [3] A. Veen, F.P. Schoenberg, Estimation of space-time branching process models in seismology using an EM-type algorithm, J. Amer. Statist. Assoc. 103 (482) (2008) 614–624.
- [4] G.O. Mohler, M.B. Short, Geographic profiling from kinetic models of criminal behavior, SIAM J. App. Math. 72 (1) (2012) 163–180.
- [5] G.O. Mohler, Modeling and estimation of multi-source clustering in crime and security data, Ann. Appl. Stat. 7 (3) (2013) 1525–1539.
- [6] M.B. Short, M.R. D'Orsogna, V.B. Pasour, G.E. Tita, P.J. Brantingham, A.L. Bertozzi, L.B. Chayes, A statistical model of criminal behavior, Math. Model. Meth. App. Sci. 18 (2008) 1249–1267.
- [7] B. Yuan, H. Li, A.L. Bertozzi, P.J. Brantingham, M.A. Porter, Multivariate spatiotemporal Hawkes processes and network reconstruction, SIAM J. Math. Data Sci. 1 (2) (2019) 356–382.
- [8] R. Lima, Hawkes processes modeling, inference, and control: An overview, SIAM Rev. 65 (2) (2023) 331–374.
- [9] K. Zhou, K. Zha, Le. Song, Learning social infectivity in sparse low-rank networks using multidimensional Hawkes processes, in: AISTATS, vol. 31, 2013, pp. 641–649.
- [10] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. Leskovec, SEISMIC: A selfexciting point process model for predicting tweet popularity, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1513–1522.
- [11] H.Y. Chen, C.T. Li, PSEISMIC: A personalized self-exciting point process model for predicting tweet popularity, in: 2017 IEEE International Conference on Big Data, 2017, pp. 2710–2713.
- [12] R. Kobayashi, R. Lambiotte, TiDeH: Time-dependent Hawkes process for predicting retweet dynamics, in: Proceedings of the Tenth International AAAI Conference OnWeb and Social Media, ICWSM 2016, 2016.
- [13] F. Chen, W.H. Tan, Marked self-exciting point process modelling of information diffusion on Twitter, Ann. Appl. Stat. 12 (2018) 2175–2196.
- [14] E.C. Hall, R.M. Willett, Tracking dynamic point processes on networks, IEEE Trans. Inform. Theory 62 (7) (2016) 4327–4346.
- [15] J.R. Zipkin, F.P. Schoenberg, K. Coronges, A.L. Bertozzi, Point-process models of social network interactions: Parameter estimation and missing data recovery, European J. Appl. Math. 27 (3) (2016) 502–529.
- [16] E. Bacry, K. Dayri, J.F. Muzy, Non-parametric kernel estimation for symmetric Hawkes processes. application to high frequency financial data, Eur. Phys. J. B 85 (2011).
- [17] E. Bacry, I. Mastromatteo, J.F. Muzy, Hawkes processes in finance, Mark. Microstruct. Liq. 01 (2015).
- [18] H. Eyjolfsson, D. Tjøstheim, Multivariate self-exciting jump processes with applications to financial data, Bernoulli 29 (3) (2023) 2167–2191.
- [19] A.G. Hawkes, Hawkes processes and their applications to finance: A review, Quant. Finance 18 (2018) 193–198.
- [20] H.J.T. Unwin, I. Routledge, S. Flaxman, M.-A. Rizoiu, S. Lai, J. Cohen, D.J. Weiss, S. Mishra, S. Bhatt, Using Hawkes processes to model imported and local malaria cases in near-elimination settings, PLoS Comput. Biol. 17 (4) (2021) e1008830.
- [21] E. Choi, N. Du, R. Chen, L. Song, J. Sun, Constructing disease network and temporal progression model via context-sensitive Hawkes process, in: 2015 IEEE International Conference on Data Mining, 2015, pp. 721–726.
- [22] Y. Diouane, F. Schoenberg, G. Mohler, Accurate estimation of cross-excitation in multivariate Hawkes process models of infectious diseases, in: IEEE 11th International Conference on Data Science and Advanced Analytics, 2024.
- [23] J. Shang, M. Sun, Geometric Hawkes processes with graph convolutional recurrent neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4878–4885.
- [24] U. Upadhyay, A. De, M. Gomez-Rodriguez, Deep reinforcement learning of marked temporal point processes, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 3172–3182.
- [25] Y. Wang, B. Xie, N. Du, L. Song, Isotonic Hawkes processes, in: Proceedings of the 33rd International Conference on Machine Learning, vol. 48, PMLR, 2016, pp. 2226–2234.
- [26] S. Xiao, J. Yan, X. Yang, H. Zha, S.M. Chu, Modeling the intensity function of point process via recurrent neural networks, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 1597–1603.
- [27] S. Zuo, H. Jiang, Z. Li, T. Zhao, H. Zha, Transformer Hawkes process, in: Hal Daumé III, Aarti Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 11692–11702.
- [28] E. Lewis, G.O. Mohler, A nonparametric EM algorithm for multiscale Hawkes processes, J. Nonpara. Stat. (2011) 1–16.
- [29] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. Leskovec, SEISMIC: A self-exciting point process model for predicting tweet popularity, KDD '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1513–1522.

- [30] R. Lemonnier, N. Vayatis, Nonparametric Markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate Hawkes processes, 2014.
- [31] M. Eichler, R. Dahlhaus, J. Dueck, Graphical modeling for multivariate Hawkes processes with nonparametric link functions, J. Time Series Anal. 38 (2) (2016) 225–242.
- [32] S.W. Linderman, R.P. Adams, Discovering latent network structure in point process data, in: ICML' 14, vol. 32, 2014, pp. 1413–1421.
- [33] C.W.J. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica 37 (3) (1969-08-01).
- [34] H. Xu, M. Farajtabar, H. Zha, Learning Granger causality for Hawkes processes, in: Proceedings of the 33rd International Conference on Machine Learning, vol. 48, 2016, pp. 1717–1726.
- [35] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, J.-F. Muzy, Uncovering causality from multivariate Hawkes integrated cumulants, in: Doina Precup, Yee Whye Teh (Eds.), in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1–10.
- [36] S. Kim, D. Putrino, S. Ghosh, Emery N. Brown, A granger causality mesure of point process models of ensemble neural spiking activity, PLoS Comp. Bio. 7 (3) (2011) 1–13.
- [37] P. Albert, A two-state markov mixture model for a time series of epileptic seizure counts, Biometrics (1991) 1371–1381.
- [38] M.B. Short, A.L. Bertozzi, Nonlinear patterns in urban crime: Hotspots, bifurcations, and suppression, SIAM J. Appl. Dyn. Syst. 9 (2) (2010) 462–483.
- [39] M.A. Rizoiu, A. Soen, S. Li, P. Calderon, L.J. Dong, A.K. Menon, L. Xie, Interval-censored Hawkes processes, J. Mach. Learn. Res. 23 (338) (2022) 1–84.
- [40] R.H. Shumway, D.S. Stoffer, An approach to time series smoothing and forecasting using the EM algorithm, J. Time Series Anal. 3 (4) (1982) 253–264.
- [41] S.J. Godsill, A. Doucet, M.A. West, Monte Carlo smoothing for nonlinear time series, J. Amer. Statist. Assoc. 99 (2004) 156–168.

- [42] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in: Rad. and Sig. Pro., IEE Proc. F, vol. 140, (2) 1993, pp. 107–113.
- [43] A. Doucet, Sequential Monte Carlo methods, in: Handbook of Graphical Models, 2006.
- [44] A. Doucet, A.M. Johansen, The Oxford handbook of nonlinear filtering, in: Chapter A Tutorial on Particle Filtering and Smoothing: Fifteen years later, Oxford University Press, New York, 2008, pp. 656–704.
- [45] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comput. Graph. Statist. 5 (1996) 1–25.
- [46] R.H. Byrd, M.E. Hribar, J. Nocedal, An interior point algorithm for large-scale nonlinear programming, SIAM J. Optim. 9 (4) (1999) 877–900.
- [47] K. Lange, MM Optimization Algorithms, SIAM, 2016.
- [48] A. Mkhadri, M. Ouhourane, K. Oualkacha, A coordinate descent algorithm for computing penalized smooth quantile regression, Stat. Comput. 28 (2017) 865–883.
- [49] D. Hunter, K. Lange, A tutorial on MM algorithms, Amer. Statist. 58 (1) (2004) 30–37.
- [50] N. Santitissadeekorn, M.B. Short, D.J.B. Lloyd, Sequential data assimilation for 1D-self exciting process with application to urban crime data, Comp. Stat. Dat. Anal. 128 (2018) 163–183.
- [51] E.W. Fox, M.B. Short, K.D. Schoenberg, A.L. Bertozzi, Modeling E-mail networks and inferring leadership using self-exciting point processes, J. Amer. Statist. Assoc. 111 (514) (2016) 564–584.
- [52] P.D. Moral, A. Doucet, S.S. Singh, Forward Smoothing Using Sequential Monte Carlo, Technical report, Cambridge, 2010.
- [53] G. Evensen, Analysis of iterative ensemble smoothers for solving inverse problems, Comput. Geosci. 22 (2018) 885–908.