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

Naratip Santitissadeekorn[1], Martin Short[2], and David J. B. Lloyd[1,3]

[1]Department of Mathematics, University of Surrey, Guildford, UK
[2]Department of Mathematics, Georgia Institute of Technology, USA
[3]Centre for Criminology, University of Surrey, Guildford, UK

February 12, 2024

Abstract

Discovering connections and identifying key influencers from time series data when no prior network structure is known is an important and challenging problem in many applications, from crime to social media. Much attention has been paid to event-based time series (timestamp) data, in which the sequence of times of events is reported, but few methods consider count data in which discrete counts of events are given in a fixed time interval, which frequently occurs for real-world applications. Here, we lay the foundation to systematically develop methods covering a range of network inference problems for both sequential and batched count data involving different scales and complexity. For small scale networks and batch-data, we develop an algorithm using an ensemble-based expectation maximization framework where the node dynamics and influence of connections are modelled by a general discrete-time Cox or Hawkes process. The second method is a batch-data algorithm designed for linear multidimensional Hawkes model of the node dynamics based on a minimization majorization approach, leading to an iterative method that solves the maximum likelihood problem that can be parallelised at each iteration to enable the inference of large-scale network structure. The third method is a sequential data assimilation method that is based on a second-order approximation of the Bayesian inference problem that, under certain assumptions, a rank-1 update for the covariance matrix can be employed to reduce the computational cost; the method is also parallelizable, allowing applicability to large-scale problems. The methods are tested on synthetic data constructed from a multidimensional Cox, a Hawkes-Poisson process, an agent-based model of urban crime and on real-world email communications between European academic communities. We demonstrate the robustness of the methods to construct the underlying network where influencing is driven not only by excitation but also diffusion. This work opens the field to develop new methods for network reconstruction from count data for real-world problems.

1 Introduction

This work is motivated by conventional applications of continuous-time 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 [18] to capture a self-excitation process, used particularly for seismic events [30]. 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 [38, 30], urban crime analysis [28, 27, 34, 42] social network analysis [25, 45, 43, 6, 22, 5, 17, 46], financial time-series analysis [3, 4, 12, 19], contagious disease network [36, 7] and deep learning network [32, 37, 39, 40, 47].

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 to decouple the parameter space when optimizing the surrogate function, speeding up 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 non-parametric 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 [24] and the regularization to promote sparsity [44]. Other techniques have also been developed to estimate non-parametric kernels; see for instance [3, 23, 10]. A fully-Bayesian, parallel inference algorithm was also developed in [26] 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 [16]. 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 context of Granger causality provides interpretability of the results. It was demonstrated in [10] 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 [41, 1, 20]. 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 modeled 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 [2]. 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 [34, 33]. 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 synchronisation 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. We 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 [35, 14]. We demonstrate how this idea can be applied to network reconstruction of a small network.

The MM algorithm is developed for batch data inference aimed at large-scale network problems. 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 minimises 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 parallelised for large-scale problems.

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 and with parallelisation the method is then applicable for large-scale network problems.

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 large-scale 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, $\theta^i := [\theta^{i,1}, \dots, \theta^{i,n}]^\intercal$. We concatenate all vectors θ^i to form a parameter vector θ, i.e., $\theta := [\theta^1; \cdots; \theta^m]$.

Without loss of generality, 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

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

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

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

where $\mathcal{C}$ 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 $\mathbf{x}_k := \left[\mathbf{x}_k^1, \ldots, \mathbf{x}_k^m\right]$ 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, $\mathbf{p}$ 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 $\mathbf{q}$ 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 $p(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta)$, where $\mathbf{x}_{0:k}$ denotes the sequence of $\mathbf{x}_0$ up to $\mathbf{x}_k$. The (marginal) likelihood function in (2.2) can be expressed by

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

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 [35, 14]. 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

$$\begin{aligned} \mathcal{Q}\left(\theta; \theta^{(\kappa)}\right) &= \int p\left(\mathbf{x}_{0:K} \mid \Delta N_{1:K}, \theta^{(\kappa)}\right) \log p\left(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta\right) d\mathbf{x}_{0:K} \\ &= \mathbb{E}\left[\log p\left(\mathbf{x}_{0:K}, \Delta N_{1:K} \mid \theta\right)\right]. \end{aligned} \tag{2.7}$$

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

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

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

$$\begin{aligned} &\mathcal{Q}\left(\theta, \theta^{(\kappa)}\right) = Q_0\left(\theta, \theta^{(\kappa)}\right) + Q_x\left(\theta, \theta^{(\kappa)}\right) + Q_{\Delta N}\left(\theta, \theta^{(\kappa)}\right), \\ &Q_0\left(\theta, \theta^{(\kappa)}\right) = \int p\left(\mathbf{x}_0 \mid \Delta N_{1:K}, \theta^{(\kappa)}\right) \log p\left(\mathbf{x}_0 \mid \theta\right) d\mathbf{x}_0, \\ &\ = \mathbb{E}\left[\log p\left(\mathbf{x}_0 \mid \theta\right)\right], \\ &Q_{\mathbf{x}}\left(\theta, \theta^{(\kappa)}\right) = \sum_{k=1}^{K} \int p\left(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \Delta N_{1:K}, \theta^{(\kappa)}\right) \log p\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \Delta N_{1:K}, \theta\right) d\mathbf{x}_k d\mathbf{x}_{k-1}, \\ &\ = \sum_{k=1}^{K} \mathbb{E}\left[\log p\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \Delta N_{1:K}, \theta\right)\right], \\ &Q_{\Delta N}\left(\theta, \theta^{(\kappa)}\right) = \sum_{k=1}^{K} \int p\left(\mathbf{x}_k \mid \Delta N_{1:K}, \theta^{(\kappa)}\right) \log p\left(\Delta N_k \mid \mathbf{x}_k, \theta\right) d\mathbf{x}_k, \\ &\ = \sum_{k=1}^{K} \mathbb{E}\left[\log p\left(\Delta N_k \mid \mathbf{x}_k, \theta\right)\right]. \end{aligned} \tag{2.9}$$

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

The expression in (2.7) suggests that if we can sample from $p\left(\mathbf{x}_{0:K} \mid \Delta N_{1:K}, \theta^{(\kappa)}\right)$, we can then estimate all the expectations in (2.9) using the sample paths, which we denote by $\mathbf{x}^s_{0:K}$. 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 filtering-backward sampling procedure to obtain samples approximately from the joint smoothing distribution [14], which is a combination of particle filtering (PF) and backward simulation smoother (BSS) to generate $\mathbf{x}^s_{0:K}$.

Particle filtering is a sequential Monte Carlo (SMC) technique for non-linear filtering. In the current application, it can be used to sample $p\left(\mathbf{x}_k \mid \Delta N_{1:k}, \theta^{(\kappa)}\right)$, 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, [15, 8, 9, 21].

Suppose that we have obtained the weighted, filtered particle $\left(\mathbf{x}_k^{f(\ell)}, \mathbf{w}_k^{f(\ell)}\right)$ for $i = 1, \ldots, N_f$, where N_f is the number of particles used in PF. The particles approximate $p\left(\mathbf{x}_k \mid \Delta N_{1:k}, \theta^{(\kappa)}\right)$ but the EM algorithm requires $p\left(\mathbf{x}_k \mid \Delta N_{1:K}, \theta^{(\kappa)}\right)$ for any $k = 1, \ldots, K$. The BSS uses the filtered particles $\left(\mathbf{x}_k^{f(\ell)}, \mathbf{w}_k^{f(\ell)}\right)$ to generate the smoothing particles, $\left(\mathbf{x}_k^{s(\ell)}, \mathbf{w}_k^{s(\ell)}\right)$ for $j = 1, \ldots, N_s$, 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 $\left(\mathbf{x}_k^{s(\ell)}, \mathbf{w}_k^{s(\ell)}\right)$. Maximization of $\mathcal{Q}$ in (2.9) to find $\theta^{(\kappa+1)}$ is then carried out numerically. We use the function `fmincon` in `MATLAB` to optimize $\mathcal{Q}$. A useful by-product of this approach is that the ensemble of sample paths of the conditional intensity $\lambda^i_{1:K}$ 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 ensemble-based 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

$$\begin{aligned} x_k &= \underbrace{(1-\omega_1\delta t)x_{k-1} + \omega_1\mu\delta t}_{:=\Psi_x(x_{k-1})} + \epsilon\sqrt{\delta t}\eta_k, \\ g_k &= \underbrace{(1-\omega_2\delta t)g_{k-1} + \alpha\Delta N_{k-1},}_{:=\Psi_g(g_{k-1})} \\ \lambda_k &= \exp(x_k) + g_k. \end{aligned} \tag{2.10}$$

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 $x_k \in \mathbb{R}$.

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 $\kappa-$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 $\mathcal{Q}-$function in (2.9). The $\mathcal{Q}-$function (after omitting the terms irrelevant to maximization) has the following form:

$$
\begin{aligned}
\mathcal{Q}_0(\theta,\theta^{(\kappa)}) &= -\frac{1}{2N_s}\sum_{\ell=1}^{N_s}\frac{\left(x_0^{s(\ell)}-\mu^{(\kappa)}\right)^2}{\delta t} \\
\mathcal{Q}_{\mathbf{x}}(\theta,\theta^{(\kappa)}) &= -\frac{1}{2N_s}\sum_{\ell=1}^{N_s}\left[\sum_{k=1}^{K}\frac{\left(x_k^{s(\ell)}-\Psi_x\left(x_{k-1}^{s(\ell)}\right)\right)^2}{\epsilon^2\delta t}+K\log\epsilon\right] \qquad (2.11)\\
\mathcal{Q}_{\Delta N}(\theta,\theta^{(\kappa)}) &= \frac{1}{2N_s}\sum_{\ell=1}^{N_s}\left[\sum_{k=1}^{K}\Delta N_k\log\lambda_k-\exp(\lambda_k)\delta t\right]
\end{aligned}
$$

When K is large, the term $\mathcal{Q}_0(\theta,\theta^{(\kappa)})$ in (2.9) can be neglected. Furthermore, $\mathcal{Q}_x$ depends on only μ,ω_1,ϵ and $\mathcal{Q}_{\Delta N}$ depends only on α,ω_2. We can then solve the two maximization problems in parallel to obtain $\theta^{(\kappa+1)}$. Maximizing $\mathcal{Q}_x$ has a closed-form solution, see Appendix B, and can be readily computed. However, since $\mathcal{Q}_x$ and $\mathcal{Q}_{\Delta N}$ are maximized in parallel, numerically maximizing $\mathcal{Q}_{\Delta N}$ becomes a bottleneck to the speed of the algorithm. For this experiment, we numerically maximize both $\mathcal{Q}_x$ and $\mathcal{Q}_{\Delta N}$ using `fmincon` in `MATLAB`. The constraint optimization is required to ensure the positive values of parameters.

The results are shown in Figure 1. We also compare the filtered intensity $\lambda_{0:k}^{f(\ell)}$ with the smoothed intensity $\lambda_{0:k}^{s(\ell)}$ in Figure 1, which are computed from $x_{0:k}^{f(\ell)}$, $x_{0:k}^{s(\ell)}$ using the observation equation in (2.10), respectively. Figure 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 Figure 1.

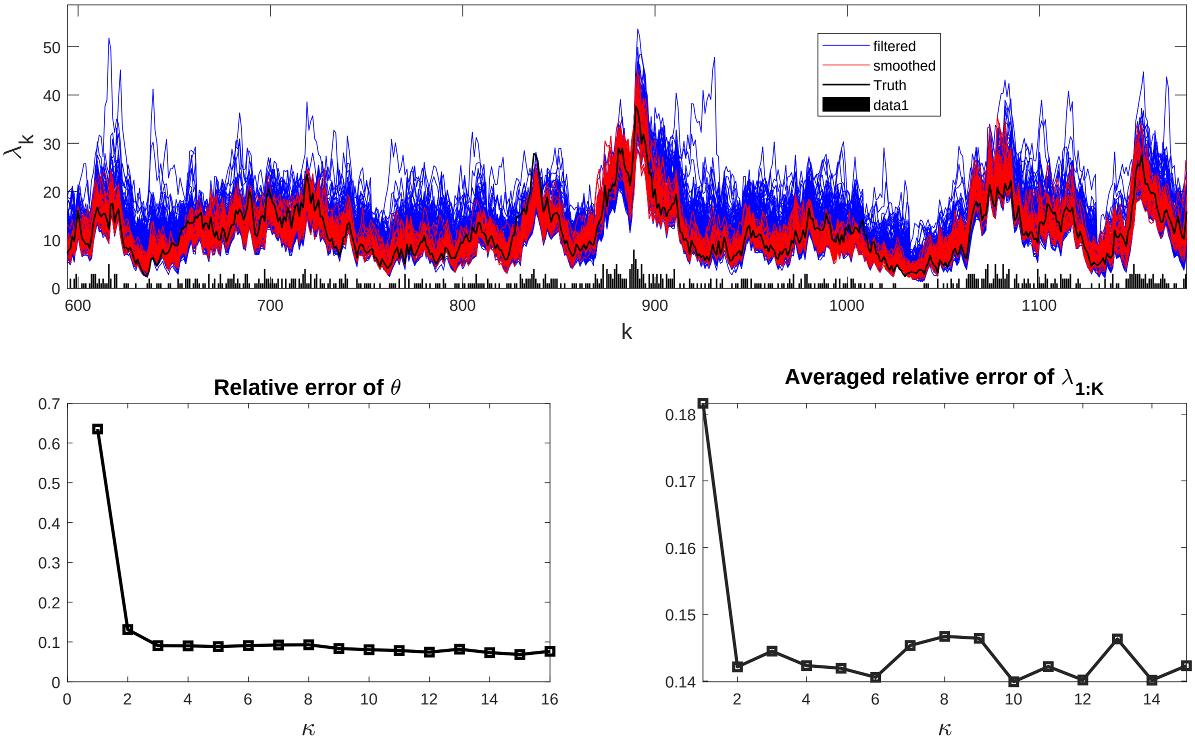


Figure 1: *(Top) Comparison between filtered particles $x_{0:k}^{f(\ell)}$, smoothed particles $x_{0:k}^{s(\ell)}$ and ground truth. The simulated data is also shown in the bar plot. (Bottom, left) The progress of the relative error of the parameter vector θ over 15 EM iterations. (Bottom, right) The propagation of the relative error for the conditional intensity.*

2.2 Logistic LGCP

We modify the conditional intensity function in (2.10) to

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

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 $\mathcal{Q}-$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 Figure 2, a fast reduction of both relative errors is obtained at the beginning and then becomes stable.

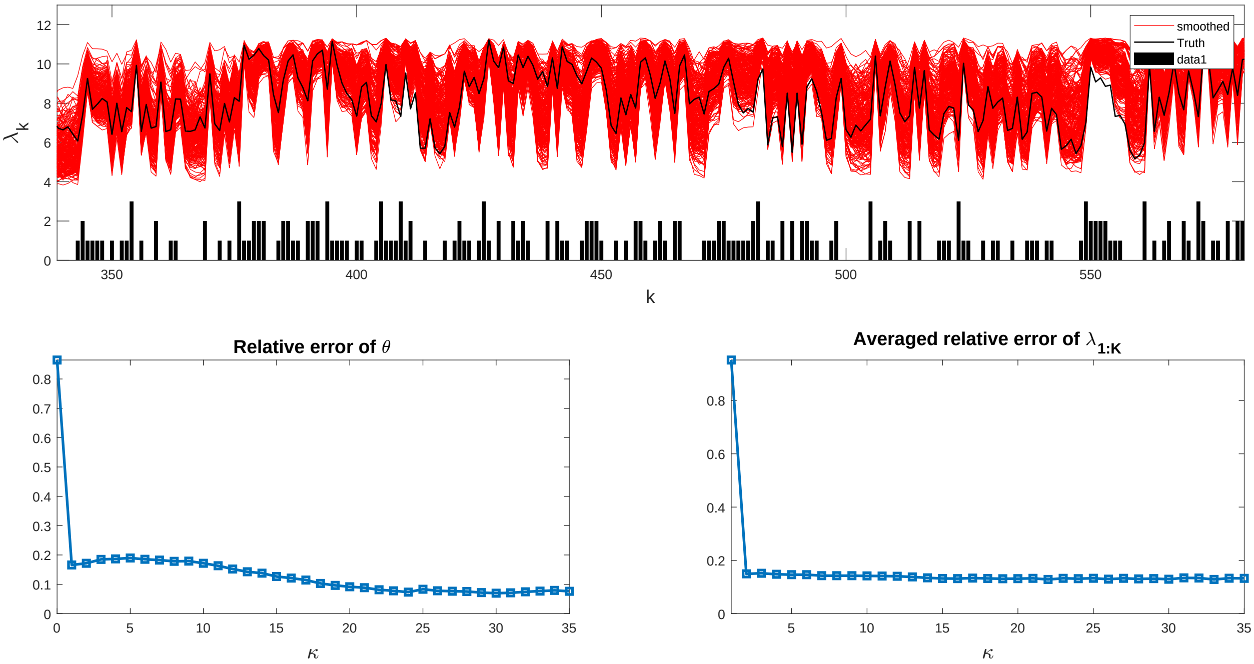


Figure 2: *(Top) Comparison between filtered particles $x_{0:k}^{f(\ell)}$, smoothed particles $x_{0:k}^{s(\ell)}$ and ground truth for the Logistic LGCP. The simulated data is also shown in the bar plot. (Bottom, left) The progress of the relative error of the parameter vector θ over 35 EM iterations. (Bottom, right) The progress of the relative error for the conditional intensity.*

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 (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 $j = 1, \ldots, m$, that has the following form:

$$\begin{aligned} x_{k+1}^i &= \underbrace{\left[(1-\eta^i)x_k^i + \eta^i \sum_{j \neq i} x^j\right](1 - \omega_1^i \delta t) + \omega_1^i \mu^j \delta t}_{:=\Psi_x^i(x_k^i)} + \epsilon^i \sqrt{\delta t}\zeta_k, \\ g_{k+1}^i &= (1 - \omega_2^i \delta t) g_k^i + \sum_{j=1}^m \alpha^{ij} \Delta N_k^j, \\ \lambda_{k+1}^i &= \exp(x_{k+1}^i) + g_{k+1}^i, \end{aligned} \tag{2.13}$$

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_i^j are the excitation coupling parameters, $\epsilon^j > 0$ and $\zeta_k \sim \mathcal{N}(0,1)$ describes the noise at each node. The $\mathcal{Q}-$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 $\mathbf{x}_k = \left[x_k^1, \ldots, x_k^m\right] \in \mathbb{R}^m$. 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 Figure 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 Figures 3 to 5. The errors of the parameter estimation for various values of K are shown in Figure 3, all of which exhibit fast error reduction in the first step and then slowly decrease afterwards, similar to the univariate case shown in §2.1. The smoothed path at the final EM step is shown in Figure 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 Figure 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.

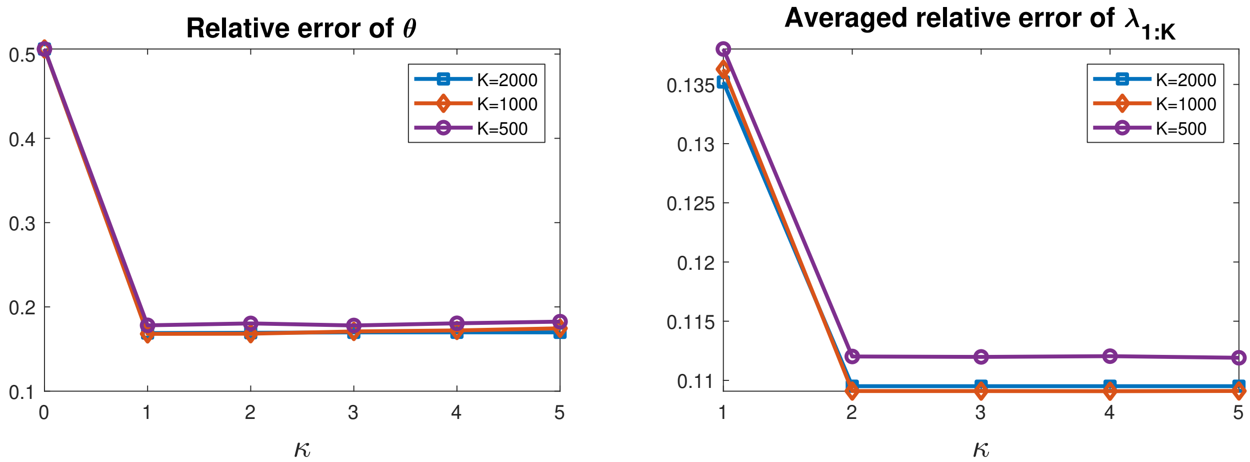


Figure 3: *(Left) Relative error of the parameter vector at each EM-step.(Right) Relative error of the conditional intensity at each EM-step*

3 Majorization-Minimization (MM)

In this section, we consider a large-scale network problem and 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^j is given by

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

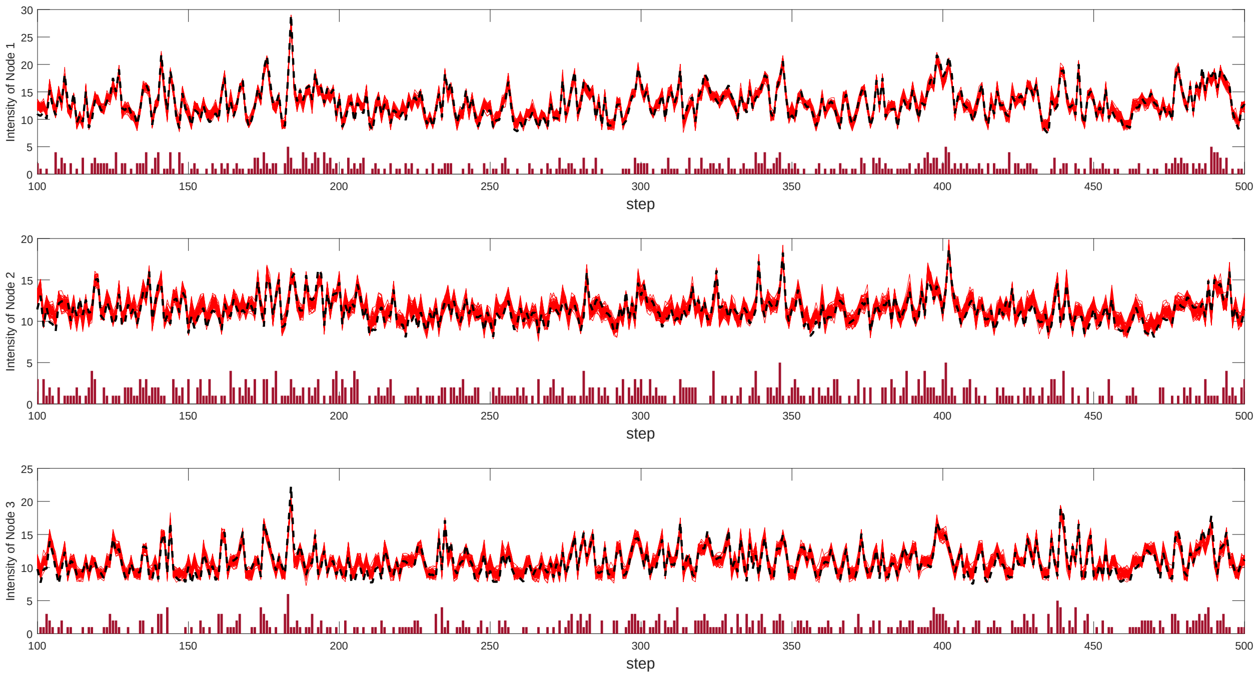
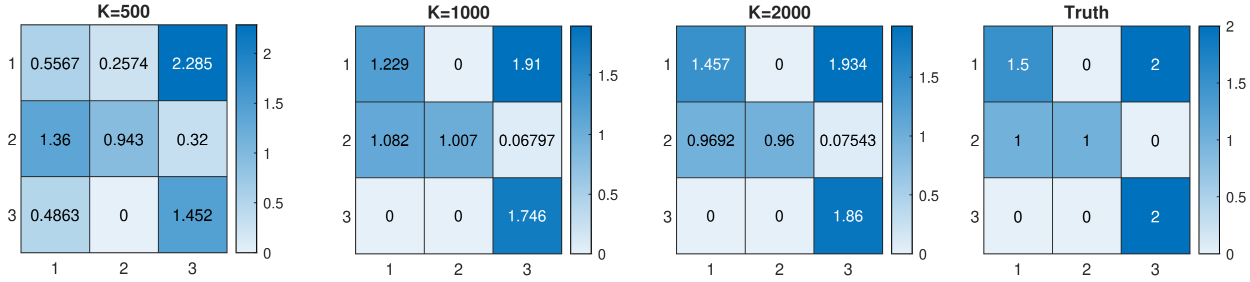


Figure 4: *From the top to bottom, the plot shows the smoothed path at the final step of the EM for node* 1 *to* 3*, respectively. For a clear visualisation, only part of the trajectory is shown at the time step* $k = 100 - 500$*. The bar plot beneath the intensity shows the simulated count data for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



K=500

	1	2	3
1	0.5567	0.2574	2.285
2	1.36	0.943	0.32
3	0.4863	0	1.452

K=1000

	1	2	3
1	1.229	0	1.91
2	1.082	1.007	0.06797
3	0	0	1.746

K=2000

	1	2	3
1	1.457	0	1.934
2	0.9692	0.96	0.07543
3	0	0	1.86

Truth

	1	2	3
1	1.5	0	2
2	1	1	0
3	0	0	2

Figure 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}*.*

where $\mu^i > 0, \alpha^{ij} \geq 0$ and $0 < \gamma^i < 1$. 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., $\lambda^i_0 = \mu^i$. 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^i_k toward the baseline rate. The recursive model (3.1) can also be rewritten in a closed form by

$$\lambda^i_k = \mu^i + \sum_{\ell=1}^{k} B^i_\ell \left(\gamma^i\right)^{k-\ell} \quad \& \quad B^i_\ell := \sum_{j=1}^{m} \alpha^{ij} \Delta N^i_{\ell-1}. \tag{3.2}$$

This section provides an iterative procedure to minimise 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. Due to the conditional independence of λ^i_k for all $i = 1, \ldots, m$ and $k = 1, \ldots, K$ given the parameters, we can separately minimise 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 in this section, e.g., α^{ij} will be written by α^j without confusion. The negative log-likelihood for a given node is

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

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

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

where $\phi_{klj} := \alpha^j(\gamma^j)^{k-l-1}\Delta N_k^j$. Clearly, we have $Q_k(\theta^{(n)} \mid \theta^{(n)}) = -\log(\lambda_k(\theta^{(n)}))$. We now define a tight upper bound function of $\mathcal{L}(\theta)$ by

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

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

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

where $\mathcal{N}^j = \Delta N_1^j + \ldots + \Delta N_{K-2}^j$. 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 $\left(\alpha^j\right)^{(n+1)}$. 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 $\theta = \left(\mu^i, \alpha^{i1}, \ldots, \alpha^{im}, \gamma^{i1}, \ldots, \gamma^{im}\right)$. Again, we will omit the superscript i in the algorithm below because of independence of parameters between nodes. Hence, we will write $\theta = \left(\mu, \alpha^1, \ldots, \alpha^m, \gamma^1, \ldots, \gamma^m\right)$. 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 inequality, the upper bound function can be obtained by

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

where

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

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

$$\begin{aligned}\mu^{(n+1)} &= \frac{1}{K}\sum_{k=1}^{K}\frac{\mu^{(n)}}{\lambda_k^{(n)}}\Delta N_k,\\ \left(\alpha^j\right)^{(n+1)} &= -B^j + \frac{\sqrt{(B^j)^2+4A^jC^j}}{2A^j},\\ \left(\gamma^j\right)^{(n+1)} &= -D^j + \frac{\sqrt{(D^j)^2+4D^jE^j}}{2D^j},\end{aligned} \tag{3.9}$$

where

$$\begin{aligned}A^j &= \mathcal{N}^j\frac{1+(\gamma^j)^{(n)}}{(\alpha^j)^{(n)}},\\ B^j &= \Delta N_{K-1}^j,\\ C^j &= \sum_{k=1}^{N}\sum_{l=0}^{k-1}\frac{\Delta N_k\phi_{klj}^{(n)}}{\lambda_k^{(n)}},\\ D^j &= \mathcal{N}^j\frac{(\alpha^j)^{(n)}}{1+(\gamma^j)^{(n)}},\\ E^j &= \sum_{k=1}^{K}\sum_{l=0}^{k-1}(k-l-1)\frac{\Delta N_k\phi_{klj}^{(n)}}{\lambda_k^{(n)}}.\end{aligned} \tag{3.10}$$

In practice, a regularisation scheme may be required to obtain useful results. For the system with known decay rate, we apply a regularisation 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

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

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 hyperparameters a and b. We will discuss how we choose a and b in the synthetic experiment in the subsequent section.

Similarly, we may regularise the update equations for μ and γ^j in (3.9) by applying a gamma prior distribution for μ with the standard shape parameter a and inverse scale parameter b and a beta prior distribution for each γ^i 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 regularisation will be the same as (3.11). However, to update γ^j, we must solve a quartic polynomial of the following form

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

where x denotes $\left(\gamma^j\right)^{(n+1)}$ 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 Discrete-time Hawkes model and filtering

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 [31], 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

$$\begin{aligned} \mathbf{P}_k^{-1} &= \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^{m} \left[\left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right) \left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right)^{\intercal} \lambda_k^i \delta t - (\Delta N_k^i - \lambda_k^i \delta t) \frac{\partial^2 \log \lambda_k^i}{\partial^2 \theta_k} \right], \\ \bar{\theta}_k &= \bar{\theta}_{k|k-1} + \mathbf{P}_k \sum_{i=1}^{m} \left[\left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right) (\Delta N_k^i - \lambda_k^i \delta t) \right], \end{aligned} \tag{4.1}$$

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,

$$\theta_k^i := [\log \mu_k^i, \log \alpha_k^{i1} \ldots, \log \alpha_k^{im}]^{\intercal}. \tag{4.2}$$

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,

$$\theta_k = \theta_{k-1} + \eta_k, \tag{4.3}$$

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 $S_k^i = \Delta N_k^i + \gamma^i \Delta N_{k-1}^i + \cdots + (\gamma^i)^{k-1} \Delta N_0^i$, 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

$$\frac{\partial \log \lambda_k^i}{\partial \theta_k} = \left[\frac{\partial \log \lambda_k^i}{\partial \theta_k^1}, \frac{\partial \log \lambda_k^i}{\partial \theta_k^2}, \ldots, \frac{\partial \log \lambda_k^i}{\partial \theta_k^m} \right]^{\intercal}, \tag{4.4}$$

where

$$\frac{\partial \log \lambda_k^i}{\partial \theta_k^i} = \begin{cases} \frac{1}{\lambda_k^i} \left[e^{\mu_k^i}, \;\; S_k^1 e^{\alpha_k^{j1}}, \ldots, \;\; S_k^m e^{\alpha_k^{jm}} \right], & \text{if } i = j, \\ \mathbf{0}, & \text{if } i \neq j. \end{cases} \tag{4.5}$$

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. It follows that the Hessian has a simple form, given by

$$\left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right) \left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right)^{\intercal} = -\frac{\partial^2 \log \lambda_k^i}{\partial^2 \theta_k} + \Lambda^{(i)}, \tag{4.6}$$

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

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^{m} \Delta N_k^i \left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right) \left(\frac{\partial \log \lambda_k^i}{\partial \theta_k} \right)^{\intercal} + \left(\lambda_k^i \delta t - \Delta N_k^i \right) \Lambda^{(j)}. \tag{4.7}$$

With the form in (4.7), 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. Consequently, the update system (4.1) can be implemented for each node in parallel, which enhances the feasibility of the proposed algorithm for a large-scale problem. 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 1

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 $\delta t(1-\gamma)^{-1}\mathbf{A}$ is less than 1, where $\mathbf{A}$ 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, $\mathbf{A}$, examined in this experiment. The structures of three different ground truths are shown in Figure 7 representing the different scenarios: only self-excitation (top row), localised 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 regularisation are shown in Figures 6 and 7, respectively. For any $i-$th node, the hyperparameter for the regularised 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 $0.5\overline{N^i}$ (half of the total count for the given node) and variance $0.5\overline{N^i}/K$. 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 regularisation, 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 Figures 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 $k = 1, \ldots, K$. The initial guess of the μ^j is set to be half of the data average of the $j-$th node. Figure 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. Figure 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 Figure 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 [34] to simulate

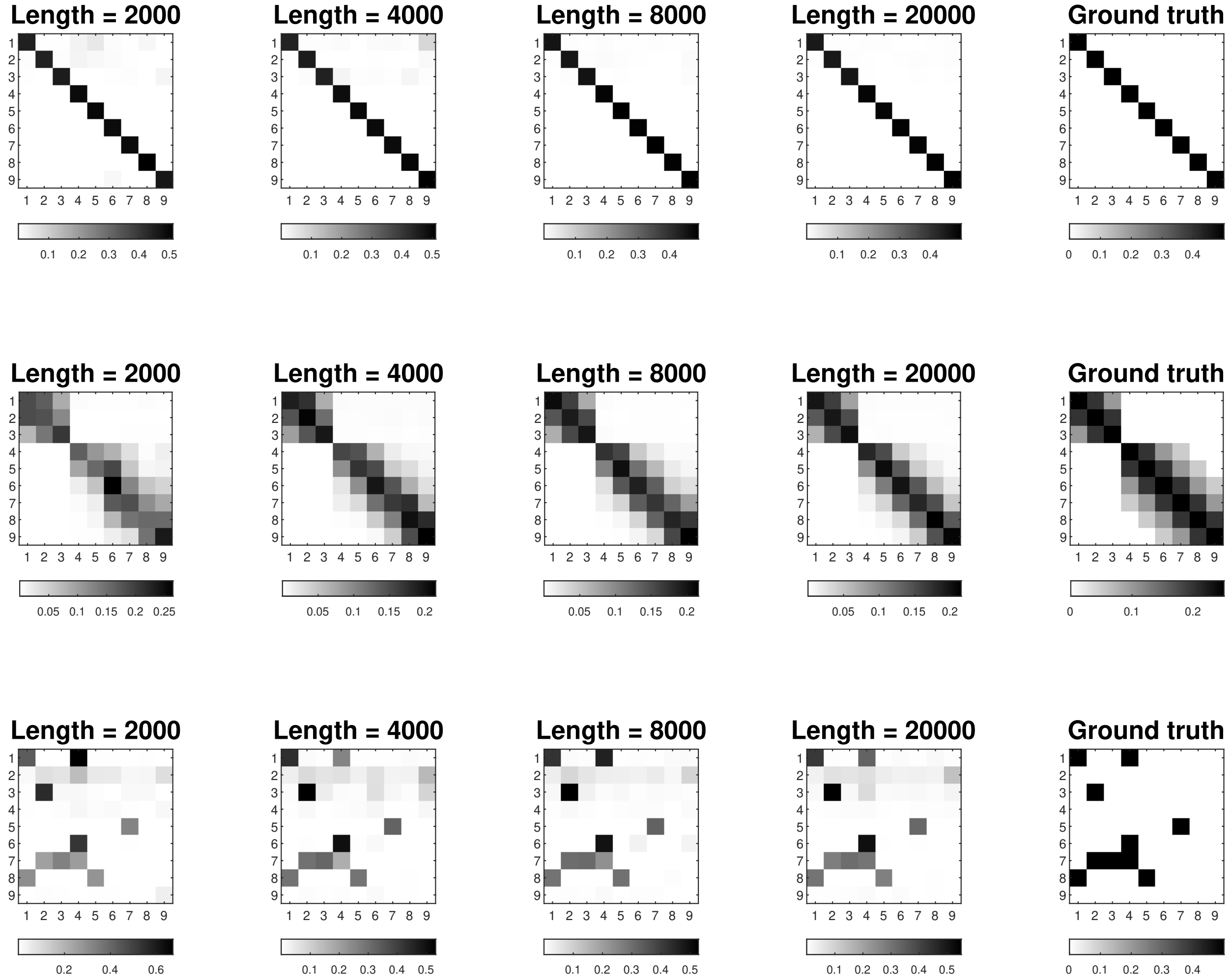


Figure 6: *The estimated network structure obtained from the MM algorithm* ***with*** *regularisation for three different ground truths. The length of the data is varied to demonstrate the convergence to the ground truth.*

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:

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

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;
- $w(s,s') \geq 0$ 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

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

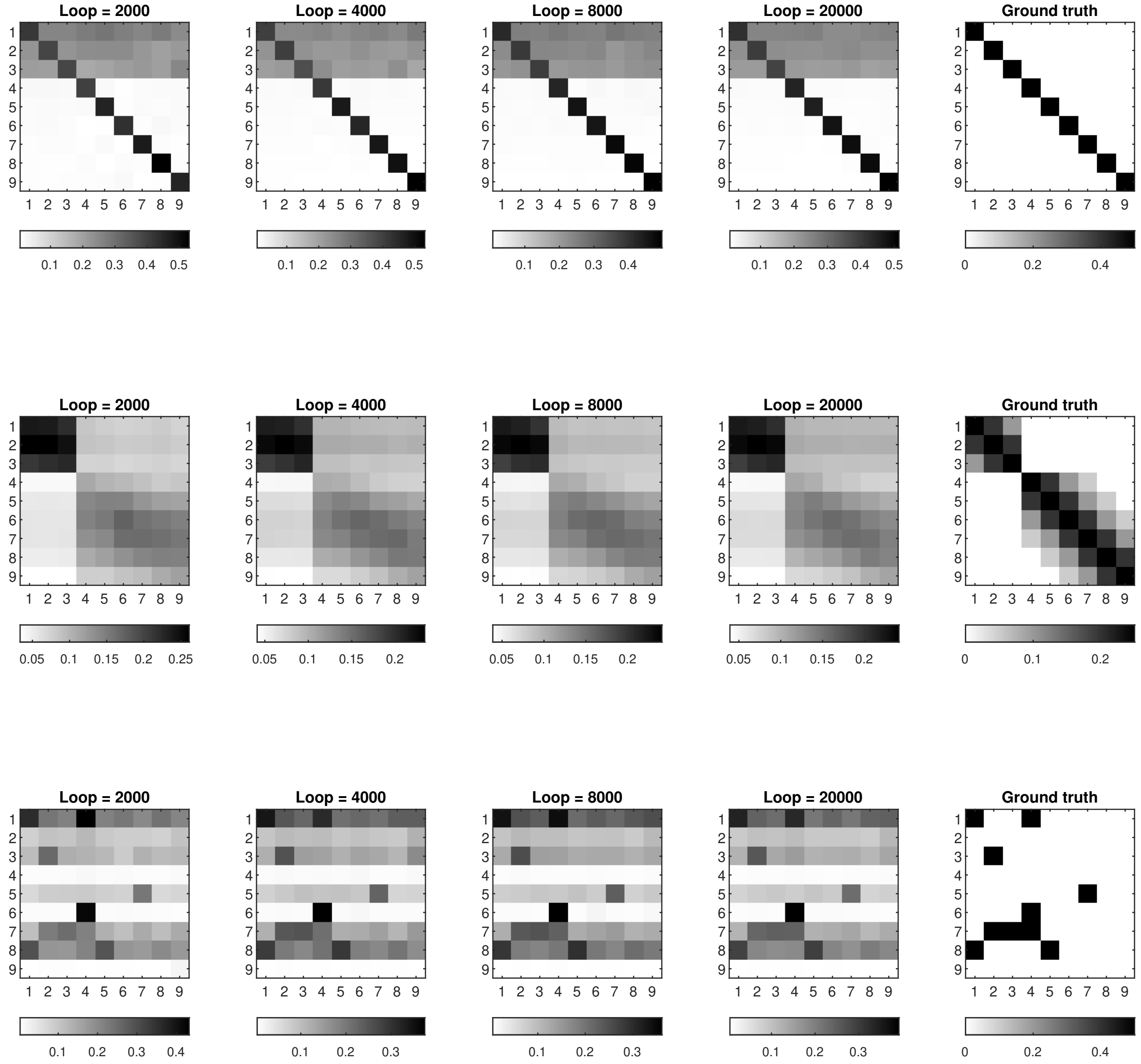


Figure 7: *The estimated network structure obtained from the MM algorithm* ***without*** *regularisation for three different ground truths. The length of the data is varied to demonstrate the convergence to the ground truth.*

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 $\boldsymbol{W}$ such that its $s-$th row and $s'-$th column entry is $w(s, s')$, and reconstruct the pattern of non-zero elements of $\boldsymbol{W}$ 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.

We generated data of various lengths based on the same influence matrix pattern, represented by $\boldsymbol{W}$ with 64 nodes, as illustrated in Figure 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 $s, s' \in 1, \ldots, 64$,

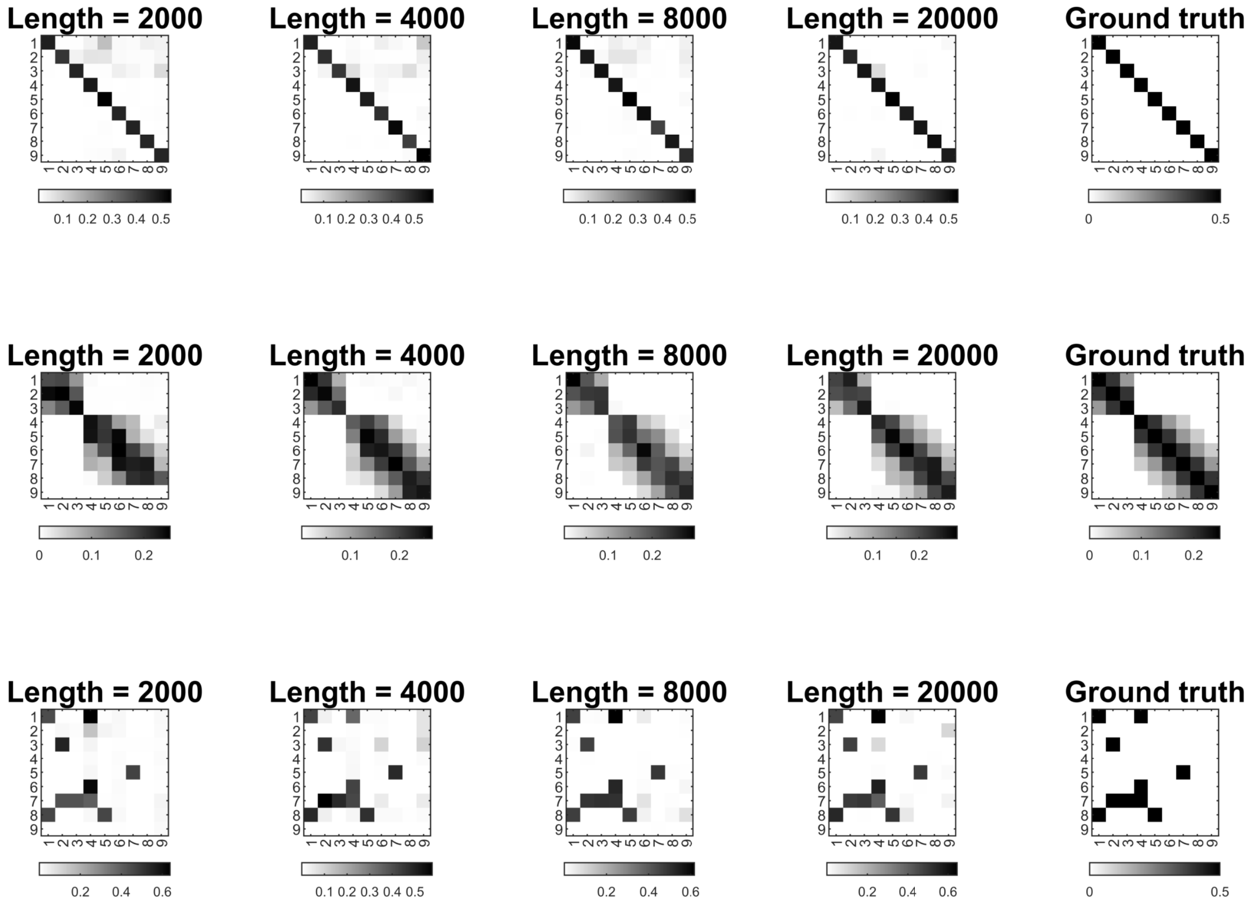


Figure 8: *The estimated network structure obtained from the ExPKF algorithm using a true decay rate. The length of the data is varied to demonstrate the convergence to the ground truth.*

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 s. Additional information on the simulated data for Case 1 is provided in Figure 11. The cumulative counts of the data can be grouped into three distinct categories based on the unique values of the row sums of $\boldsymbol{W}$. 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. Figure 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 Figure 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 Figure 14. When computing both error measures, we normalise $\boldsymbol{W}$ 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

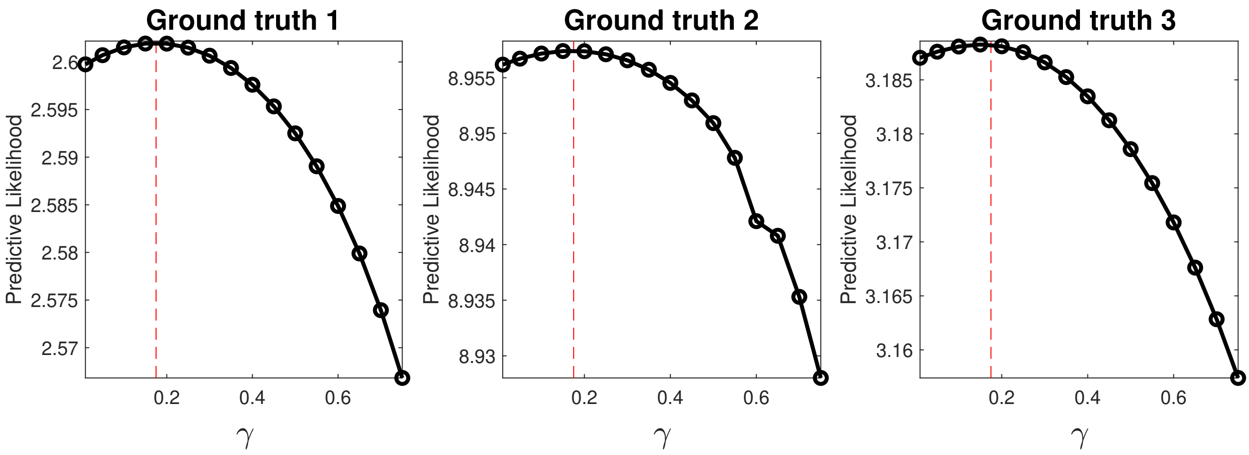


Figure 9: *Comparing the predictive log likelihood for various values of decay rate, γ. The vertical line indicates the true value of the decay rate. The data with length of 20000 is used to produce this result.*

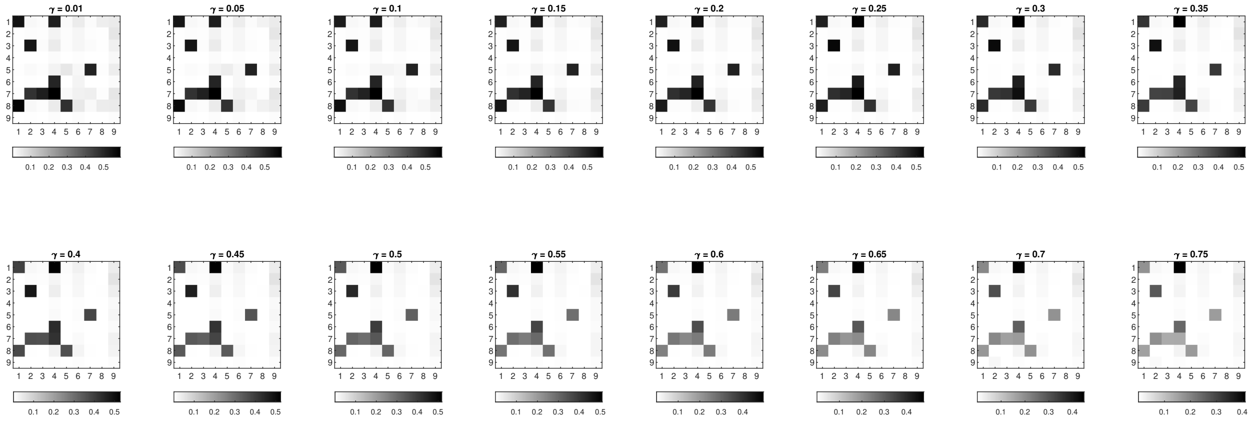


Figure 10: *Comparing the network structure for the ground truth 3 when the value of decay rate, γ, is incorrectly specified.*

small improvement of the ExPKF results, which is consistent with the visualisation of Figure 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 [13, 46]. The Hawkes model with the exponential decay rate was used in [13] 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. Figure 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

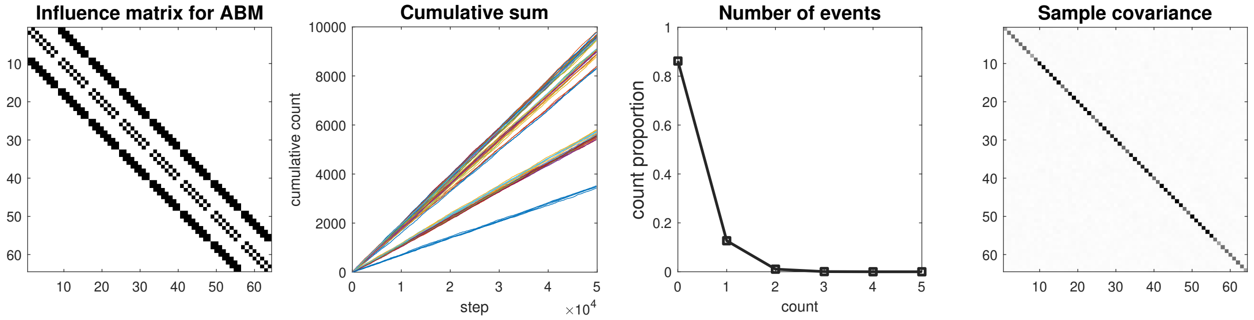


Figure 11: *Data simulated from ABM model. The left most plot illustrates the influence matrix $\boldsymbol{W}$. The second plot from the left shows the cumulative number of events for all nodes. The third plot is the frequency distribution of the count. The right most plot is the sample covariance matrix.*

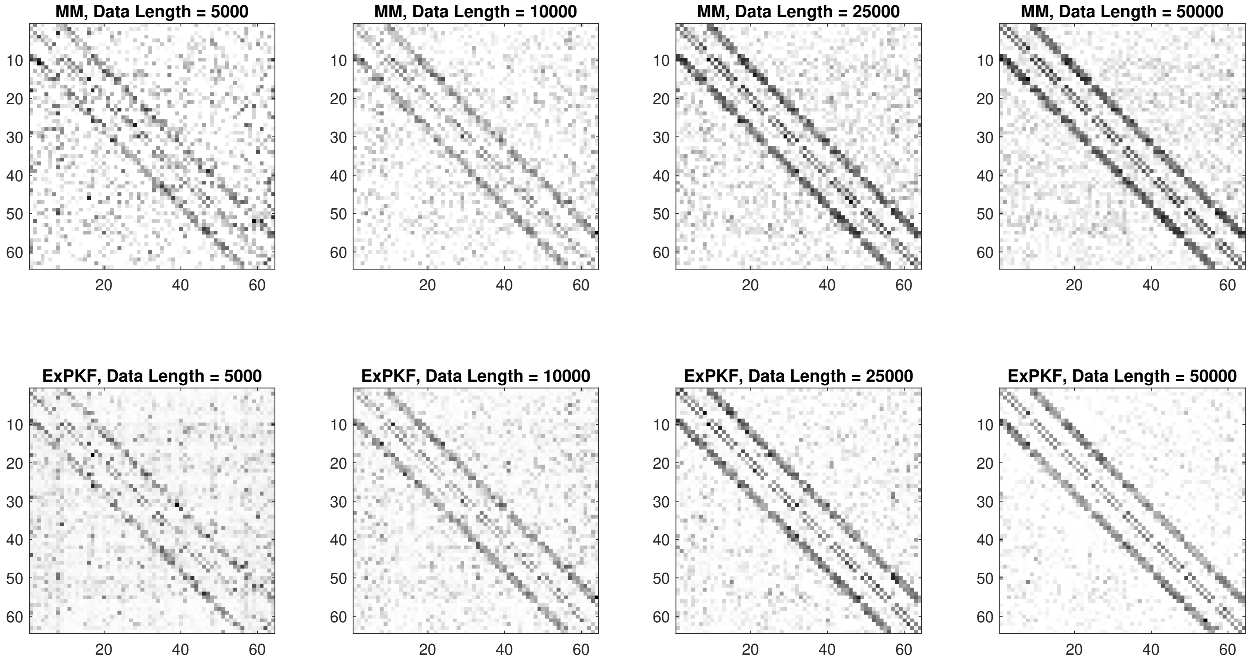


Figure 12: *MM and ExPKF estimation of the influence matrix using different data length for Case 1, which has a strong excitation effects.*

top pairs $(9, 18)$ and $(11, 22)$.

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

Table 1: *Top 9 sender-receiver for the Ikenet data ranked by the total number of outgoing emails.*

As shown in Figure 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 (anonymised) 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

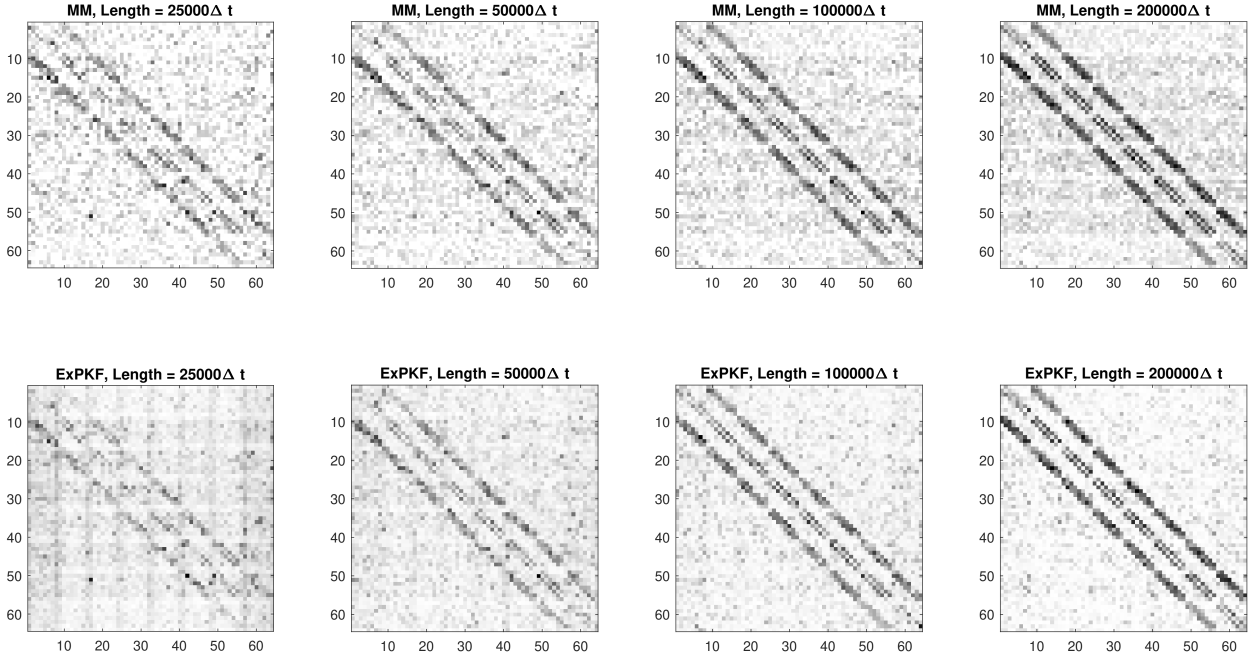


Figure 13: *MM and ExPKF estimation of the influence matrix using different data length for Case 2, which has a weak excitation effect but strong diffusion.*

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

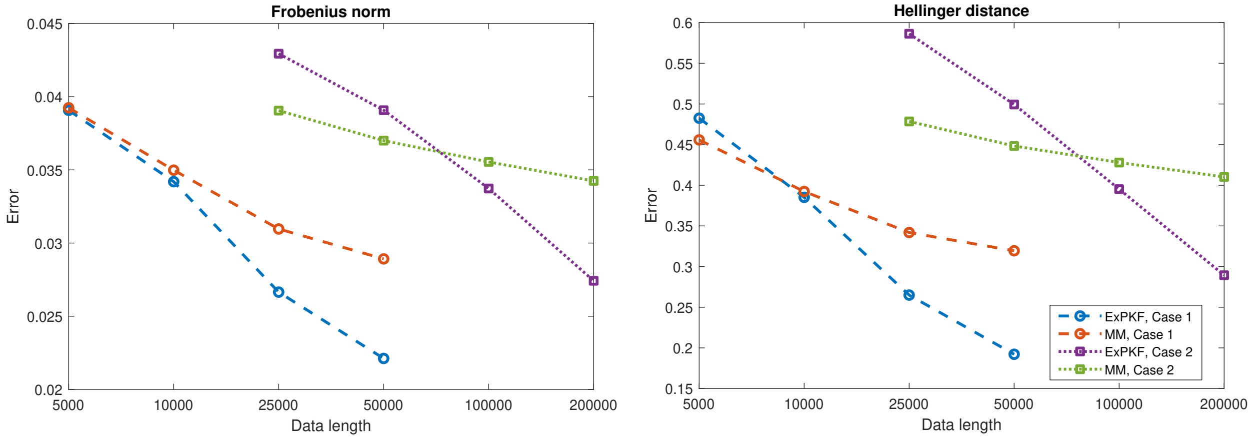


Figure 14: *MM and ExPKF Error for different data lengths. (Left) Frobenius norm. (Right) Hellinger distance. Two cases are presented: Case 1 (plotted with the circle markers) corresponds to Figure 12 and Case 2 (plotted with the square markers) corresponds to Figure 13.*

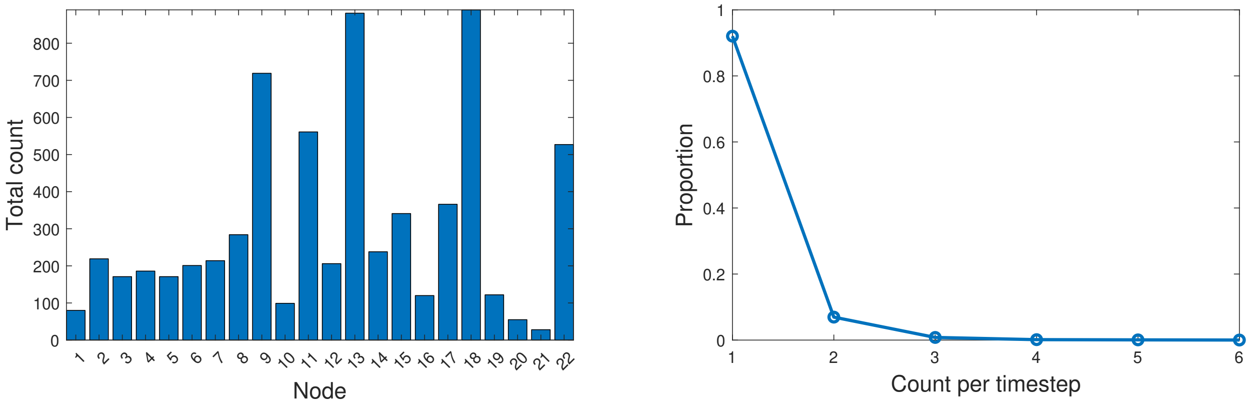


Figure 15: *The total count of emails sent by each node (Left) and proportion of non-zero counts (Right).*

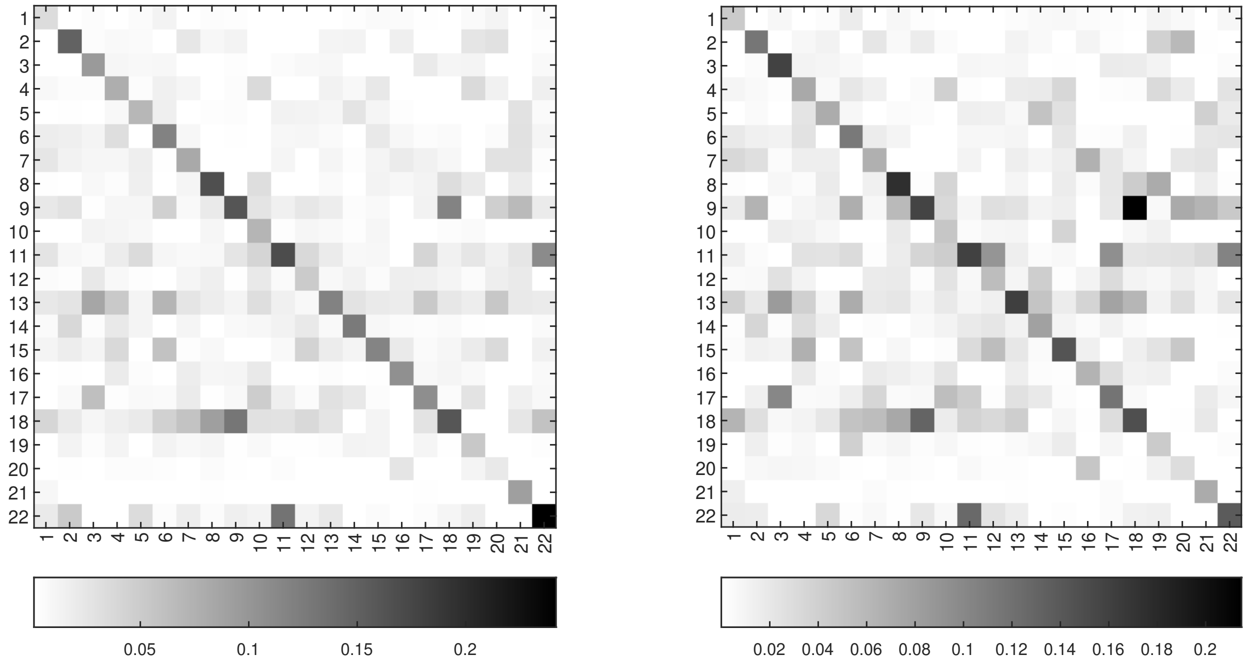


Figure 16: *Influence networks associated with the 22-node Ikenet email data constructed by MM algorithm (Left) and ExPKF algorithm (Right).*

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 filtering-backward 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.

This paper lays the foundations for other smoothing methods that could be used instead of

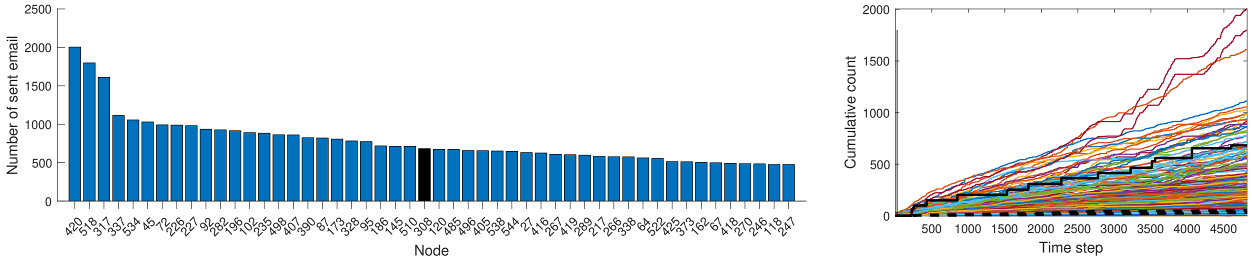


Figure 17: *(Left) Histogram of the top 50 users by number of emails sent. The black bar is associated with node 308, which is identified by ExPKF as the most influential node. (Right) The cumulative counts of all nodes. The node 308 has the cumulative counts shown in a black solid curve. The cumulative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

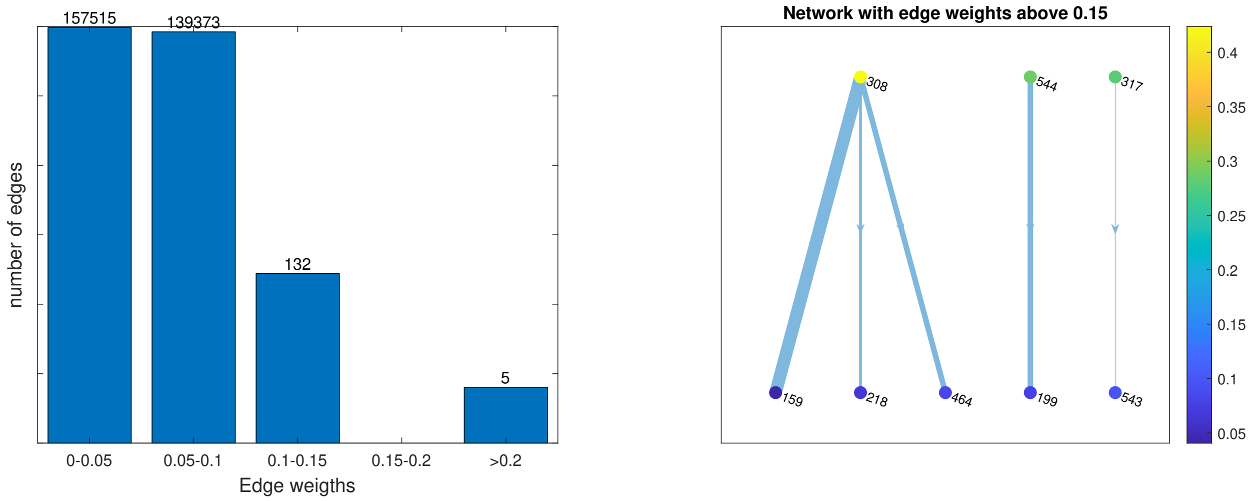


Figure 18: *(Left) Histogram of the edge weights, i.e., α_{ij} for the large email network. (Right) The subnetwork with the edges weight above 0.15.*

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 demonstrated in [29] 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 [11]. 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 large-scale network problems, making them ideal for 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 applied to large-scale problems potentially for networks involving $\mathcal{O}(10^6)$ nodes. 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 large-scale network reconstruction for applications in social networks and neural networks that hitherto remained out of reach with current methods.

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 particle and its corresponding normalized weight respectively at time step $k = 0, 1, \ldots, K$, where K is the length of the time-series of count data.

1. Initialization: Randomly generate N_f particles from an initial distribution

$$\mathbf{x}_0^{(\ell)} \sim p\left(\mathbf{x}_0\right)$$

and set initial weights: $w_k^{(\ell)} = 1/N_f$ for $i = 1, \ldots, N_f$.

2. Repeat this step for $k = 1, \ldots, K$,

    (a) Draw random samples from the conditional predictive distribution, denoted by $\mathbf{x}_k^{p(\ell)}$ based on (2.4).

    $$\mathbf{x}_k^{(\ell)} \sim N(\mathbf{x}_k^{(\ell)}; \Psi\left(\mathbf{x}_{k-1}^{(\ell)}\right), \mathbf{Q})$$

    and then generate the predictive conditional intensity $\lambda_\kappa^{j,(\ell)} = \exp\left(\mathbf{x}_k^{j,(\ell)}\right) + g_k^j$ for all nodes $j = 1, \ldots, m$ based on (2.5)

    (b) Update (unnormalized) weights based on the likelihood model

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

    and then normalize the weight by

    $$w_k^{(\ell)} := \frac{\widetilde{w}_k^{f(\ell)}}{\sum_{\ell=1}^{N} \widetilde{w}_k^{f(\ell)}}.$$

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

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

There are a number of methods for resampling. For simplicity, we use the systematic sampling algorithm described in [21], which costs $O(N_f)$

Backward Smoothing Simulation (BSS):

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

1. Initialization: Suppose $\mathbf{x}_{0:K}^{f(\ell)}$ and $w_{0:K}^{f(\ell)}$, for $i = 1, \ldots, N_f$, 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 $k = K-1, \ldots, 0$,

   (a) For $\ell = 1, \ldots, N_f$, 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 $\ell = 1, \ldots, N_s$ will then be used to estimate the parameters in the M-step of the EM algorithm, see again (2.9).

Appendix B: Maximization of $\mathcal{Q}_x$ in (2.11)

By neglecting $\mathcal{Q}_0$ in (2.11), we can find $\mu_1^{(\kappa+1)}, \omega_1^{(\kappa+1)}$ and $\epsilon^{(\kappa+1)}$ by maximizing $\mathcal{Q}_x$ only. Let $\beta = (1 - \omega_1 \delta t)$ and $\gamma = \omega_1 \mu$. We can rewrite $\mathcal{Q}_x$ by

$$\mathcal{Q}_x = -\frac{1}{2N_s \epsilon^2 \delta t} \|\mathbf{y} - \mathbf{A}\mathbf{z}\|^2 - \frac{1}{2} K \log \epsilon,$$

where $\mathbf{z} = [\beta, \gamma]^\intercal$, $\mathbf{y} = \left[x_1^{s(1)}, \ldots, x_K^{s(N_s)}, \ldots, x_K^{s(1)}, \ldots, x_K^{s(N_s)}\right]^\intercal$ and

$$\mathbf{A} = \begin{pmatrix} x_0^{s(1)} & \delta t \\ \vdots & \vdots \\ x_{K-1}^{s(1)} & \delta t \\ \vdots & \vdots \\ x_0^{(sN_s)} & \delta t \\ \vdots & \vdots \\ x_{K-1}^{s(N_s)} & \delta t \end{pmatrix}.$$

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

After obtaining $\mathbf{z}^{(\kappa+1)} = \left[\beta^{(\kappa+1)}, \gamma^{(\kappa+1)}\right]^\mathsf{T}$, 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.

References

[1] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, and J.-F. Muzy. Uncovering causality from multivariate Hawkes integrated cumulants. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR, 06–11 Aug 2017.

[2] P. Albert. A two-state markov mixture model for a time series of epileptic seizure counts. *Biometrics*, pages 1371–1381, 1991.

[3] E. Bacry, K. Dayri, and J.F. Muzy. Non-parametric kernel estimation for symmetric hawkes processes. application to high frequency financial data. *The European Physical Journal B*, 85, 2011.

[4] E. Bacry, I. Mastromatteo, and J.F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 01, 2015.

[5] F. Chen and W. H. Tan. Marked self-exciting point process modelling of information diffusion on twitter. *Annals of Applied Statistics*, 12:2175–2196, 12 2018.

[6] H. Y. Chen and C. T. Li. PSEISMIC: A personalized self-exciting point process model for predicting tweet popularity. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2710–2713, 2017.

[7] E. Choi, N. Du, R. Chen, L. Song, and J. Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. *2015 IEEE International Conference on Data Mining*, pages 721–726, 2015.

[8] A. Doucet. Sequential monte carlo methods. *Handbook of Graphical Models*, 2006.

[9] A. Doucet and A. M. Johansen. *The Oxford Handbook of Nonlinear Filtering*, chapter A Tutorial on Particle Filtering and Smoothing: Fifteen years later, pages 656–704. Oxford University Press, New York, 2008.

[10] M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate Hawkes processes with nonparametric link functions. *J. Time Ser. Anal.*, 38(2):225–242, 2016.

[11] G. Evensen. Analysis of iterative ensemble smoothers for solving inverse problems. *Computational Geosciences*, 22:885–908, 2018.

[12] H. Eyjolfsson and D. Tjøstheim. Multivariate self-exciting jump processes with applications to financial data. *Bernoulli*, 29(3):2167 – 2191, 2023.

[13] E. W. Fox, M. B. Short, K. D. Schoenberg, F. P.and Coronges, and A. L. Bertozzi. Modeling E-mail Networks and Inferring Leadership Using Self-Exciting Point Processes. *Journal of the American Statistical Association*, 111(514):564–584, 2016.

[14] S. J. Godsill, A. Doucet, and M. A. West. Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, 99:156 – 168, 2004.

[15] N. J. Gordon, D.J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Rad. and Sig. Pro., IEE Proc. F*, 140(2):107–113, 1993.

[16] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3), 1969-08-01.

[17] E. C. Hall and R. M. Willett. Tracking Dynamic Point Processes on Networks. *IEEE Transactions on Information Theory*, 62(7):4327–4346, 2016.

[18] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[19] A. G. Hawkes. Hawkes processes and their applications to finance: A review. *Quant. Finance*, 18:193–198, 2018.

[20] S. Kim, D. Putrino, S. Ghosh, and Emery N. Brown. A Granger causality mesure of point process models of ensemble neural spiking activity. *PLOS comp. Bio.*, 7(3):1–13, 2011.

[21] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

[22] R. Kobayashi and 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.

[23] R. Lemonnier and N. Vayatis. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes. In *Machine Learning and Knowledge Discovery in Databases*, 2014.

[24] E. Lewis and G. O. Mohler. A nonparametric EM algorithm for multiscale Hawkes processes. *J. Nonpara. Stati.*, pages 1–16, 2011.

[25] R. Lima. Hawkes processes modeling, inference, and control: An overview. *SIAM Review*, 65(2):331–374, 2023.

[26] S. W. Linderman and R. P. Adams. Discovering latent network structure in point process data. In *ICML' 14*, volume 32, pages 1413–1421, 2014.

[27] G. O. Mohler. Modeling and estimation of multi-source clustering in crime and security data. *Ann. Appl. Stat.*, 7(3):1525–1539, 2013.

[28] G. O. Mohler and M. B. Short. Geographic profiling from kinetic models of criminal behavior. *SIAM J. on App. Math.*, 72(1):163–180, 2012.

[29] P. D. Moral, A. Doucet, and S. S. Singh. Forward smoothing using sequential monte carlo. Technical report, Cambridge, 2010.

[30] Y. Ogata. Seismicity analysis through point-process modeling: A review. *pure and applied geophysics*, 155:471–507, 1999.

[31] N. Santitissadeekorn, M. B. Short, and D. J. B. Lloyd. Sequential data assimilation for 1d-self exciting process with application to urban crime data. *Comp. Stat. Dat. Anal.*, 128:163–183, 2018.

[32] J. Shang and M. Sun. Geometric Hawkes Processes with Graph Convolutional Recurrent Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4878–4885, 2019.

[33] M. B. Short and A. L. Bertozzi. Nonlinear patterns in urban crime: Hotspots, bifurcations, and suppression. *SIAM J. Appl. Dyn. Syst.*, Vol. 9, No. 2:pp. 462–483, 2010.

[34] M. B. Short, M. R. D'Orsogna, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes. A statistical model of criminal behavior. *Math. Model. and Meth. in App. Sci.*, 18:1249–1267, 2008.

[35] R. H. Shumway and D. S. Stoffer. An Approach To Time Series Smoothing And Forecasting Using The EM Algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[36] H. J. T. Unwin, I. Routledge, S. Flaxman, M.-A. Rizoiu, S. Lai, J. Cohen, D. J. Weiss, S. Mishra, and S. Bhatt. Using hawkes processes to model imported and local malaria cases in near-elimination settings. *PLoS Comput Biol*, 17(4):e1008830, 2021.

[37] U. Upadhyay, A. De, and M. Gomez-Rodriguez. Deep reinforcement learning of marked temporal point processes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3172–3182, 2018.

[38] A. Veen and F. P. Schoenberg. Estimation of Space–Time Branching Process Models in Seismology Using an EM–Type Algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

[39] Y. Wang, B. Xie, N. Du, and L. Song. Isotonic hawkes processes. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 2226–2234. PMLR, 2016.

[40] S. Xiao, J. Yan, X. Yang, H. Zha, and 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*, pages 1597–1603. AAAI Press, 2017.

[41] H. Xu, M. Farajtabar, and H. Zha. Learning Granger causality for Hawkes processes. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1717–1726, 2016.

[42] B. Yuan, H. Li, A. L. Bertozzi, P. J. Brantingham, and M. A. Porter. Multivariate spatiotemporal hawkes processes and network reconstruction. *SIAM Journal on Mathematics of Data Science*, 1(2):356–382, 2019.

[43] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1513–1522. ACM, 2015.

[44] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1513–1522, New York, NY, USA, 2015. Association for Computing Machinery.

[45] K. Zhou, K. Zha, and Le. Song. Learning social infectivity in sparse low-rank networks using multidimensional Hawkes processes. In *AISTATS*, volume 31, pages 641–649, 2013.

[46] J. R. Zipkin, F. P. Schoenberg, K. Coronges, and A. L. Bertozzi. Point-process models of social network interactions: Parameter estimation and missing data recovery. *European Journal of Applied Mathematics*, 27(3):502–529, 2016.

[47] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha. Transformer Hawkes process. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11692–11702. PMLR, 2020.