The Block Point Process Model for Continuous-time Event-based Dynamic Networks

Ruthwik R. Junuthula
University of Toledo
Toledo, OH, USA
rjunuth@utoledo.edu

Maysam Haghdan*
ETH Zürich
Zürich, Switzerland
maysam.haghdan@inf.ethz.ch

Kevin S. Xu
University of Toledo
Toledo, OH, USA
kevin.xu@utoledo.edu

Vijay K. Devabhaktuni
University of Toledo
Toledo, OH, USA
Vijay.Devabhaktuni@utoledo.edu

ABSTRACT

We consider the problem of analyzing timestamped relational events between a set of entities, such as messages between users of an on-line social network. Such data are often analyzed using static or discrete-time network models, which discard a significant amount of information by aggregating events over time to form network snapshots. In this paper, we introduce a block point process model (BPPM) for continuous-time event-based dynamic networks. The BPPM is inspired by the well-known stochastic block model (SBM) for static networks. We show that networks generated by the BPPM follow an SBM in the limit of a growing number of nodes. We use this property to develop principled and efficient local search and variational inference procedures initialized by regularized spectral clustering. We fit BPPMs with exponential Hawkes processes to analyze several real network data sets, including a Facebook wall post network with over 3,500 nodes and 130,000 events.

CCS CONCEPTS

• **Mathematics of computing** → **Random graphs**; *Probabilistic representations*; Probabilistic reasoning algorithms; Stochastic processes; • **Computing methodologies** → *Learning latent representations*; Spectral methods.

KEYWORDS

block Hawkes model; event-based network; continuous-time network; timestamped network; relational events; stochastic block model; Hawkes process; asymptotic independence

ACM Reference Format:
Ruthwik R. Junuthula, Maysam Haghdan, Kevin S. Xu, and Vijay K. Devabhaktuni. 2019. The Block Point Process Model for Continuous-time Event-based Dynamic Networks. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313633

*Research was conducted while author was at the University of Toledo.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313633

Sender	Receiver	Time
1	2	0.1
2	3	0.4
3	2	0.6
1	2	1.2
1	3	1.3
2	1	1.6

(a) Event table

$$A^{[0,1)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A^{[1,2)} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(b) Adjacency matrix

Figure 1: Two representations of a continuous-time event-based dynamic network. The adjacency matrices discard the exact times and ordering of events.

1 INTRODUCTION

Many application settings involve analysis of timestamped relational event data in the form of triplets (sender, receiver, timestamp), as shown in Figure 1a. Examples include analysis of messages between users of an on-line social network, emails between employees of a company, and transactions between buyers and sellers on e-commerce websites. These types of data can be represented as dynamic networks evolving in continuous time due to the fine granularity on the timestamps of events and the irregular time intervals at which events occur.

Statistically modeling these types of relations and their dynamics over time has been of great interest, especially given the ubiquity of such data in recent years. Most prior work has involved modeling these relations using network representations, with nodes representing senders and receivers, and edges representing events. Such representations often either discard the timestamps altogether, which transforms the dynamic network into a static network, or aggregate events over time windows to form network snapshots evolving in discrete time as in Figure 1b. There have been numerous statistical models proposed for static networks dating back to the 1960s [12, 18], and more recently, for discrete-time networks [5, 15, 17, 26, 32, 39–42], but comparatively less attention has been devoted to continuous-time networks of timestamped relations.

The development of such continuous-time or point process-based network models [2, 7, 8, 11, 27, 38] appears to have progressed separately from recent advances in static and discrete-time network models. There have been many recent developments on estimation

for static network models such as the stochastic block model (SBM), including the development of consistent estimators such as regularized spectral clustering [31, 34]. Although some continuous-time models have drawn their inspiration from static network models, to the best of our knowledge, there has not been prior work connecting the two types of models, and in particular, examining whether provably accurate estimators for static network models can be used to estimate continuous-time models.

In this paper we introduce the block point process model (BPPM) for continuous-time event-based dynamic networks, inspired by the SBM for static networks. Our main contributions are as follows:

- We demonstrate an asymptotic equivalence between our proposed BPPM and the SBM in the limit of growing number of nodes, which allows us to use provably accurate and efficient estimators for the SBM, such as regularized spectral clustering, as a starting point to fit BPPMs.
- We develop efficient local search and variational inference procedures for the BPPM initialized by regularized spectral clustering on an aggregated adjacency matrix.
- We fit the BPPM to several real network data sets, including a Facebook network with over 3,500 nodes and 130,000 events and demonstrate that it is more accurate at predicting future interactions compared to discrete-time SBMs.

2 BACKGROUND

We consider dynamic networks evolving in continuous time through the observation of events between pairs of nodes at recorded timestamps, as shown in Figure 1a. We assume that events are directed, so we refer to the two nodes involved in an event as the sender and receiver (although the model we propose can be trivially modified to handle undirected events by reducing the number of parameters). Such event data can be represented in the form of a matrix E where each row is a triplet $\mathbf{e} = (u, v, t)$ denoting an event from node u to node v at timestamp t. Let N denote the total number of nodes in the network, and let T denote the time of the last interaction, so that the interaction times are all in $[0, T]$.

From an event matrix E, one can obtain an adjacency matrix $A^{[t_1, t_2)}$ over any given time interval $[t_1, t_2)$ such that $0 \leq t_1 < t_2 \leq T$. To simplify notation, we drop the time interval from the adjacency matrix, i.e. $A = A^{[t_1, t_2)}$. In this adjacency matrix, $a_{ij} = 1$ if there is at least one event from node i to node j in $[t_1, t_2)$, and $a_{ij} = 0$ otherwise. For example, Figure 1b shows two adjacency matrices constructed by aggregating events from the event table shown in Figure 1a over $[0, 1)$ and $[1, 2)$.

2.1 The Stochastic Block Model

Most statistical models for networks consider an adjacency matrix rather than event-based representation; many commonly used models of this type are discussed in the survey by Goldenberg et al. [12]. One model that has received significant attention is the *stochastic block model* (SBM), which is defined as follows (adapted from Definition 3 in Holland et al. [18]):

Definition 2.1 (Stochastic block model). Let A denote a random adjacency matrix for a static network, and let $\mathbf{c}$ denote a class membership vector. A is generated according to a stochastic block model with respect to the membership vector $\mathbf{c}$ if and only if,

(1) For any nodes $i \neq j$, the random variables a_{ij} are statistically independent.
(2) For any nodes $i \neq j$ and $i' \neq j'$, if i and i' are in the same class, i.e. $c_i = c_{i'}$, and j and j' are in the same class, i.e. $c_j = c_{j'}$, then a_{ij} and $a_{i'j'}$ are identically distributed.

The classes in the SBM are also commonly referred to in the literature as blocks. The class membership vector $\mathbf{c}$ has N entries where each entry $c_i \in \{1, \ldots, K\}$ denotes the class membership of node i, and K denotes the total number of classes. Recent work has focused on estimating the class memberships from the adjacency matrix A. In this setting, spectral clustering (and regularized variants) has emerged as an efficient estimator that has theoretical accuracy guarantees [22, 31, 33–35], scales to large networks with thousands of nodes, and is generally not sensitive to initialization.

2.2 Related Work

Most existing work on modeling dynamic networks has considered a discrete-time representation, where the observations consist of a sequence of adjacency matrices. This observation model is ideally suited for network data collected at regular time intervals, e.g. weekly surveys. In practice, however, dynamic network data is often collected at much finer levels of temporal resolution (e.g. at the level of a second or millisecond), in which case it likely makes more sense to treat time as continuous rather than discrete. In order to apply discrete-time dynamic network models to such data, it must first be pre-processed by aggregating events over time windows to form network snapshots, and this technique is used in many real data experiments [15, 27, 40–42]. For example, an aggregated representation of the network in Figure 1a with time window of 1 is shown in Figure 1b.

Aggregating continuous-time network data into discrete-time snapshots presents several challenges. One would ideally choose the time window to be as short as possible for the maximum temporal resolution. However, this increases the number of snapshots, and accordingly, the computation time (typically linear in the number of snapshots). More importantly, models fit using shorter time windows can lead to worse predictors than models fit using longer time windows because the models often assume short-term memory, such as the Markovian dynamics in many discrete-time SBMs [17, 26, 39–42]. We demonstrate some of these practical challenges in an experiment in Section 6.2.2.

Another line of research that has evolved independently of discrete-time network models involves the use of point processes to estimate the structure of a latent network from observations at the nodes [10, 13, 16, 24, 36]. These models are often used to estimate networks of diffusion from information cascades. Such work differs from the setting we consider in this paper, where we directly observe events *between pairs of nodes* and seek to model the dynamics of such event sequences.

There have been several other models proposed using point processes to model continuous-time event-based networks [2, 7, 8, 11, 23, 27, 38], which is the setting we consider in this paper. These models are typically fit using Markov chain Monte Carlo (MCMC) methods, which do not scale to large networks with thousands of nodes. The BPPM that we propose in this paper is a simpler version of the Hawkes IRM [2]. The relational event model (REM) [7] is

Algorithm 1 Generative process for BPPM

1: **for** node $i = 1$ **to** N **do**
2: Sample class c_i from categorical distribution with parameter vector $\boldsymbol{\pi}$
3: **for** block pair $b = 1$ **to** p **do**
4: **loop**
5: Sample next event time t_b from bth point process
6: **if** $t_b > T$ **then**
7: **break**
8: Randomly select nodes $i \in b_1, j \in b_2$ to form an edge from node i to node j at time t_b

related to the BPPM in that it is also inspired by the SBM and shares parameters across nodes in the network in a similar manner. We discuss the Hawkes IRM and REM in greater detail and compare them to our proposed model in Section 3.4.

3 THE BLOCK POINT PROCESS MODEL

3.1 Model Specification

We propose to model continuous-time dynamic networks using a generative point process network model. Motivated by the SBM for static networks, we propose to divide nodes into K classes or blocks and to associate a univariate point process with each pair of node blocks $b = (b_1, b_2) \in \{1, \ldots, K\}^2$, which we refer to as a *block pair*. Let $p = K^2$ denote the total number of block pairs. Let $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ denote the class membership probability vector, where π_q denotes the probability that a node belongs to class q. We call our model the *block point process model* (BPPM). The generative process for the BPPM for a network of duration T time units is shown in Algorithm 1.

The BPPM is a very general model—notice that we have not specified what type of point process to use in the model (we discuss this in Section 3.3). The proposed BPPM is less flexible than existing point process network models such as the Hawkes IRM and the REM (we compare the BPPM to these models in Section 3.4), but its simplicity enables theoretical analysis of the model. We then use the findings of our analysis to develop principled and efficient inference procedures that scale to large networks with thousands of nodes and hundreds of thousands of events. The proposed inference procedures, which we discuss in Section 4, take advantage of the close relationship between the BPPM and the SBM, which we discuss next.

3.2 Asymptotic Equivalence with the Stochastic Block Model

The BPPM is motivated by the SBM, where the probability of forming an edge between two nodes depends only the classes of the two nodes. Given the relation between the point process and adjacency matrix representations discussed in Section 2, a natural question is whether there is any equivalence between the BPPM and the SBM. Specifically, does an adjacency matrix $A = A^{[t_1, t_2)}$ constructed from an event matrix E generated by the BPPM follow an SBM? As far as we know, this connection between point process and static network models has not been previously explored in the literature.

We first note that A meets criterion 2 (identical distribution within a block pair) in Definition 2.1 due to the random selection of node pair for each event in step 8 of Algorithm 1. To check criterion 1 (independence of all entries of A), we first note that entries a_{ij} and $a_{i'j'}$ in different block pairs, i.e. $(c_i, c_j) \neq (c_{i'}, c'_j)$, depend on different independent point processes (unlike in the Hawkes IRM), so a_{ij} and $a_{i'j'}$ are independent.

Next, consider entries a_{ij} and $a_{i'j'}$ in the same block pair $b = (c_i, c_j) = (c_{i'}, c_{j'})$. In general, these entries are dependent so that criterion 1 is not satisfied.[1] For example, if a Hawkes process [21] is used in step 5 of Algorithm 1, then $a_{i'j'} = 1$ indicates that at least one event was generated in block pair b, i.e. there was at least one jump in the intensity of the process. This indicates that the probability of another event is now higher, so the conditional probability $\Pr(a_{ij} = 1 | a_{i'j'} = 1)$ should be higher than the marginal probability $\Pr(a_{ij} = 1)$. Thus a_{ij} and $a_{i'j'}$ are *dependent*, so A does *not follow* an SBM!

We denote the *deviation from independence* using the terms δ_0 and δ_1 defined by

$$\delta_0 = \Pr(a_{ij} = 0 | a_{i'j'} = 0) - \Pr(a_{ij} = 0) \tag{1}$$

$$\delta_1 = \Pr(a_{ij} = 0 | a_{i'j'} = 1) - \Pr(a_{ij} = 0). \tag{2}$$

If $\delta_0 = \delta_1 = 0$, then the two adjacency matrix entries are independent. If $\delta_0 \neq 0$ or $\delta_1 \neq 0$, then the two entries are dependent, with smaller values of $|\delta_0|, |\delta_1|$ indicating less dependence. The following theorem bounds these values.

Theorem (Asymptotic Independence Theorem)**.** *Consider an adjacency matrix A constructed from the BPPM over some time interval $[t_1, t_2)$. Then, for any two entries a_{ij} and $a_{i'j'}$ both in block pair b, the deviation from independence given by δ_0, δ_1 defined in* (1), (2) *is bounded in the following manner:*

$$|\delta_0|, |\delta_1| \leq \min\{1, \mu_b / n_b\} \tag{3}$$

where μ_b denotes the expected number of events in block pair b in $[t_1, t_2)$, and n_b denotes the size of block pair b. In the limit as $n_b \to \infty$, $\delta_0, \delta_1 \to 0$ provided μ_b grows at a slower rate than n_b. Thus a_{ij} and $a_{i'j'}$ are asymptotically independent for growing n_b.

The proof of the Asymptotic Independence Theorem is provided in Appendix A. We evaluate the tightness of the bound in (3) via simulation in Section 5.1. Since it depends only on the expected number of events μ_b and not the distribution, it is likely to be loose in general but applies to any choice of point process.

The Asymptotic Independence Theorem states that the deviation given by δ_0, δ_1 is non-zero in general for fixed n_b, so the entries a_{ij} and $a_{i'j'}$ are dependent, but the dependence decreases as the size of a block (and thus, a block pair) grows. This can be achieved by letting the number of nodes N in the network grow while holding the number of classes K fixed. In this case, the sizes of block pairs would be growing at rate $O(N^2)$, so the asymptotic behavior should be visible for networks with thousands of nodes. Thus, an adjacency matrix constructed from the BPPM approaches an SBM in the limit of a growing network! To the best of our knowledge, this is the first such result linking networks constructed from point process models and static network models. It is also practically useful in

[1] An exception is the case of a homogeneous Poisson process, for which the entries are independent by the splitting property.

that it allows us to leverage recent work on provably accurate and efficient inference on the SBM for the BPPM.

3.3 Choice of Point Process Model

Any temporal point process can be used to generate the event times in the BPPM. We turn our attention to a specific point process: the Hawkes process [21], which is a self-exciting process where the occurrence of events increases the probability of additional events in the future. The self-exciting property tends to create clusters of events in time, which are empirically observed in many settings. Prior work has suggested that Hawkes processes with exponential kernels provide a good fit to many real social network data sets, including email and conversation sequences [14, 25] and re-shares of posts on Twitter [43]. Hence, we also adopt the exponential kernel, which has intensity function

$$\lambda(t) = \lambda^{\infty} + \sum_{t_i < t} \alpha e^{-\beta(t - t_i)},$$

where λ^{∞} denotes the background rate that the intensity reverts to over time, α denotes the jump size for the intensity function, β denotes the exponential decay rate, and the t_i's denote times of events that occurred prior to time t. We refer to this model as the *block Hawkes model* (BHM).

3.4 Relation to Other Models

The block Hawkes model we consider is a simpler version of the Hawkes IRM, which couples a non-parametric Bayesian version of the SBM called the infinite relational model (IRM) [20] with mutually-exciting Hawkes processes. By utilizing mutually-exciting Hawkes processes, the Hawkes IRM allows for reciprocity of events between block pairs. Similar to the BHM, node pairs in a block pair are selected at random to form an edge. The authors use MCMC-based inference that scales only to very small networks.

The BHM simplifies the Hawkes IRM by using a fixed number of classes K and univariate rather than multivariate Hawkes processes. The use of univariate Hawkes processes is crucial because it allows for independence between block pairs, which we used in the analysis in Section 3.2 to demonstrate an asymptotic equivalence with an SBM. We use this asymptotic equivalence in Section 4 to devise an efficient inference procedure that scales to networks with thousands of nodes.

The BHM also has similarities with the relational event model (REM) [7], which associates a non-homogeneous Poisson process with each pair of nodes, where the intensity function is piecewise constant with knots (change points) at the event times. Different node pairs belonging to the same block pair are governed by the same set of parameters. The REM also incorporates other edge formation mechanisms within block pairs such as reciprocity and transitivity, similar to an exponential random graph model. The authors also use MCMC for inference.

4 INFERENCE PROCEDURE

The observed data is in the form of triplets $\mathbf{e}_s = (u_s, v_s, t_s)$ for each event s denoting the nodes u_s, v_s involved and the timestamp t_s. Consider an event matrix E where each row corresponds to an event in the form of a triplet $\mathbf{e}_s$. Fitting the BPPM involves estimating both the unknown classes or blocks for each node and the point process parameters θ_b for each block pair b from E. In the case of an exponential Hawkes process, the parameters are given by

$$\theta_b = \left(\alpha_b, \beta_b, \lambda_b^{\infty}\right)$$

for each block pair b. Let $\theta = \{\theta_b\}_{b=1}^{p}$ denote the set of point process parameters over all p block pairs.

Exact inference is impractical for all but the smallest networks due to the discrete class memberships $\mathbf{c}$. Thus, we consider two approximate inference methods: a greedy local search (Section 4.1) and variational inference (Section 4.2). Both approaches are iterative and converge to a local maximum and are thus sensitive to the choice of initialization, which we discuss in Section 4.3.

4.1 Local Search

Consider a conditional likelihood function for the point process parameters θ given the values of the class memberships $\mathbf{c}$ that determine the block pairs. Let $E^{(b)}$ denote rows of E corresponding to events involving block pair $b = (b_1, b_2)$; that is, rows $\mathbf{e}_s$ where $u_s \in b_1$ and $v_s \in b_2$. The p row blocks $E^{(b)} = [\mathbf{u}^{(b)}, \mathbf{v}^{(b)}, \mathbf{t}^{(b)}]$ form a partition of the rows of matrix E. Let m_b denote the number of events observed in block pair b. Let n_b denote the size of block pair b, i.e. the number of node pairs or possible edges in block pair b, which is given by $|b_1|(|b_1| - 1)$ if $b_1 = b_2$ and $|b_1||b_2|$ otherwise. The conditional log-likelihood function is given by

$$\begin{aligned}
\log \Pr(E|\theta, \mathbf{c}) &= \log \prod_{b=1}^{p} \Pr\left(\mathbf{u}^{(b)}, \mathbf{v}^{(b)}, \mathbf{t}^{(b)} \middle| \theta_b, \mathbf{c}\right) \\
&= \log \prod_{b=1}^{p} \Pr\left(\mathbf{t}^{(b)} \middle| \theta_b\right) \prod_{s \in b} \Pr(u_s, v_s | \mathbf{c}) \\
&= \log \prod_{b=1}^{p} \Pr\left(\mathbf{t}^{(b)} \middle| \theta_b\right) \left(\frac{1}{n_b}\right)^{m_b} \\
&= \sum_{b=1}^{p} \left[\log \Pr\left(\mathbf{t}^{(b)} \middle| \theta_b\right) - m_b \log n_b\right] \qquad (4)
\end{aligned}$$

where the expression for $\Pr(u_s, v_s|\mathbf{c})$ follows from the random selection of nodes in step 8 of the BPPM generative process. The term $\log \Pr\left(\mathbf{t}^{(b)}|\theta_b\right)$ is simply the log-likelihood of the point process model parameters given the timestamps of events in block pair b. For the block Hawkes model, this term can be expressed in the following form [21, 30]:

$$\begin{aligned}
\log \Pr\left(t_{(1)}, \ldots, t_{(m)} | \alpha_b, \beta_b, \lambda_b^{\infty}\right) = \sum_{s=1}^{m} \Bigg\{ \frac{\alpha_b}{\beta_b} \left[e^{-\beta_b\left(t_{(m)} - t_{(s)}\right)} - 1\right] \\
+ \log \left[\lambda_b^{\infty} + \alpha_b \sum_{r=1}^{s-1} e^{-\beta_b\left(t_{(s)} - t_{(r)}\right)}\right] \Bigg\} - \lambda_b^{\infty} t_{(m)}, \qquad (5)
\end{aligned}$$

where $t_{(s)}$ denotes the sth event corresponding to block pair b. (5) can be written in a recursive form as shown in [29].

The conditional log-likelihood (4) requires knowledge of class memberships $\mathbf{c}$, which are used to partition the event matrix into row blocks $E^{(b)}$ and thus affect both terms in (4) through $\mathbf{t}^{(b)}$, m_b, and n_b. However, in practice, class memberships are unknown, so we must maximize (4) over all possible class assignments.

We use a local search (hill climbing) procedure, which is also often referred to as label switching or node swapping in the network science literature [19, 44] to iteratively update the class assignments to reach a local maximum in a greedy fashion. Recent work has found that such greedy algorithms are competitive with more computationally demanding estimation algorithms in both the static SBM [4] and discrete-time dynamic SBM [5, 41] while scaling to much larger networks. At each iteration, we swap a single node to a different class by choosing the swap that increases the log-likelihood the most. For each possible swap, we evaluate the log-likelihood by partitioning events using to the new class assignments, obtaining the maximum-likelihood estimates of the point process model parameters, and substituting these estimates along with the new class assignments into (4). For the block Hawkes model, we maximize (5) with respect to $(\alpha_b, \beta_b, \lambda_b^\infty)$ for each block using a standard interior point optimization routine [3].

Each iteration of the local search considers $N(K-1)$ possible swaps. Computing the log-likelihood for each swap involves iterating over the timestamps of all M events. Thus, each iteration of the local search has time complexity $O(KMN)$, which is linear in both the number of nodes and events, allowing it to scale to large networks. We verify this time complexity experimentally in Section 5.3. The local search is easily parallelized by evaluating each possible swap on a separate CPU core. We terminate the local search procedure when no swap is able to increase the log-likelihood, indicating that we have reached a local maximum.

4.2 Variational Inference

Variational inference is commonly used as an optimization-based alternative to MCMC and scales to much larger data sets. We implement a mean-field variational inference approach for the BPPM to approximate the intractable posterior distribution by a fully factorizable variational distribution. To reduce the Kullback-Leibler (KL) divergence between the true posterior and the mean-field approximation, we derive the evidence lower bound (ELBO) using an approach similar to the derivation in [6] for a static SBM.

Let Z denote an $N \times K$ class membership matrix, where the notation $z_{iq} = 1$ is equivalent to $c_i = q$, both denoting that node i is in class q. We use a K-dimensional multinomial distribution for each row $\mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{iK}]$ of Z resulting in the following variational distribution:

$$R_E(Z) = \prod_{i=1}^{N} \text{Multinomial}(\mathbf{z}_i | \boldsymbol{\tau}_i), \tag{6}$$

where $\boldsymbol{\tau}_i$ denotes the variational parameter for node i. Unlike for a static SBM, we don't have a closed-form update equation for the block Hawkes model, so we optimize the ELBO using coordinate ascent. The derivation of the ELBO and the variational expectation-maximization algorithm are provided in Appendix B.

4.3 Spectral Clustering Initialization

In order to ensure that the local search or variational inference procedures do not get stuck in poor local maxima, it is important to provide a good initialization. Methods used to initialize class estimates in static and discrete-time SBMs include variants of k-means clustering [4, 26, 27] and spectral clustering [40, 41]. Variational inference is often executed with multiple random initializations, although some structured approaches have been used successfully in practice for certain models. Given the close relationship between the proposed BPPM and the SBM discussed in Section 3.2, we use a spectral clustering initialization, which is much faster and more principled than the typical approach of multiple random initializations. Spectral clustering is an attractive choice because it scales to large networks containing thousands of nodes and has theoretical performance guarantees applicable to the BPPM, as we discuss next.

Algorithm 2 Regularized spectral clustering algorithm used to initialize the local search in the BPPM inference procedure

Require: Adjacency matrix A, number of classes K, regularization parameter $\tau \geq 0$ (Default: τ = average node degree)

1: Compute diagonal matrices O^τ with entries $o_{ii}^\tau = \sum_j a_{ij} + \tau$ and P^τ with entries $p_{jj}^\tau = \sum_i a_{ij} + \tau$
2: Compute regularized graph Laplacian $L = (O^\tau)^{-1/2} A (P^\tau)^{-1/2}$
3: Compute singular value decomposition of L
4: $\tilde{\Sigma} \leftarrow$ diagonal matrix of K largest singular values of L
5: $(\tilde{U}, \tilde{V}) \leftarrow$ left and right singular vectors for $\tilde{\Sigma}$
6: $\tilde{Z} \leftarrow [\tilde{U}, \tilde{V}]$ {concatenate left and right singular vectors}
7: Normalize each row of $\tilde{Z}$ to have magnitude of 1
8: $\hat{\mathbf{c}} \leftarrow$ k-means clustering on rows of $\tilde{Z}$
9: **return** $\hat{\mathbf{c}}$

Recent work has demonstrated that applying spectral clustering (or a regularized variant) to a network generated from an SBM results in consistent estimates of class assignments as the number of nodes $N \to \infty$ [22, 31, 33–35]. These theoretical guarantees typically require the expected degrees of nodes to grow polylogarithmically with the number of nodes so that the network is not too sparse. Networks that satisfy this requirement belong to the polylog degree regime. On the other hand, the Asymptotic Independence Theorem shows an asymptotic equivalence between the BPPM and SBM provided that there are not too many events, i.e. the network is not too dense.

In the polylog degree regime, the ratio

$$\frac{\mu_b}{n_b} = O\left(\frac{N \,\text{poly}(\log N)}{N^2}\right) \to 0 \text{ as } N \to \infty,$$

so the network is not too dense, and the Asymptotic Independence Theorem holds. Thus, spectral clustering should provide an accurate estimate of the class assignments in the polylog degree regime, which is commonly observed in real networks such as social networks. Since we consider directed relations, we use a regularized spectral clustering algorithm for directed networks (pseudocode provided in Algorithm 2) to initialize the local search. It is a variant of the DI-SIM co-clustering algorithm [34] modified to produce a single set of clusters for directed graphs by concatenating scaled left and right singular vectors in a manner similar to Sussman et al. [35]. For variational inference, we require an initialization on the variational parameters $\boldsymbol{\tau}_i$ rather than a hard clustering solution, so we set $\boldsymbol{\tau}_i$ to the ith row of $\tilde{Z}$ to represent a soft clustering initialization.

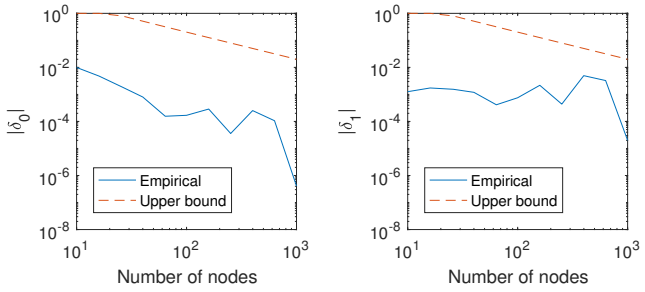


Figure 2: Comparison of empirical deviation from independence with theoretical upper bound. Empirical deviations are well within the theoretical bound.

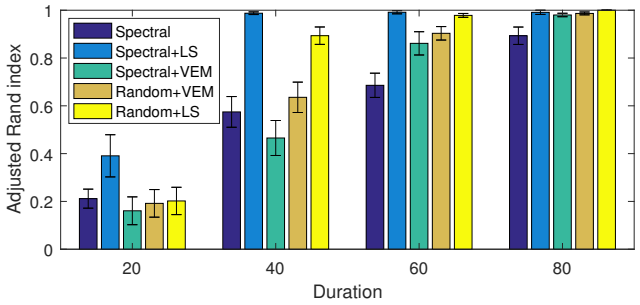


Figure 3: Mean adjusted Rand indices (± standard error) for class estimation simulation experiment with varying durations. Spectral clustering with local search produces the most accurate class estimates, while the spectral initialization for variational EM does not appear to work well.

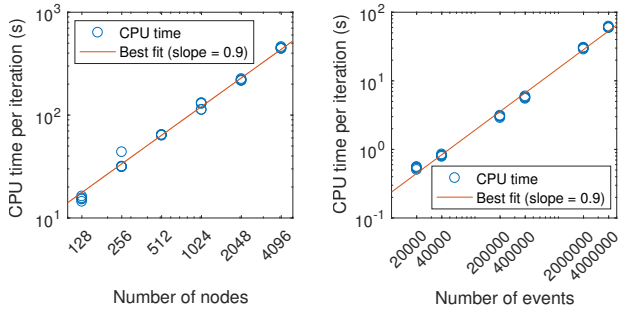


Figure 4: CPU time per iteration of local search inference in seconds with varying number of nodes and events.

5 SIMULATED DATA EXPERIMENTS

5.1 Deviation from Independence

The Asymptotic Independence Theorem demonstrates that pairs of adjacency matrix entries in the same block pair are dependent, but that the dependence is upper bounded by (3), and that the dependence goes to 0 for growing blocks. To evaluate the slackness of the bounds, we simulate networks from the block Hawkes model (BHM). Since δ_0 and δ_1 depend only on the size of the blocks, we simulate networks with a single block and let the number of nodes N grow from 10 to 1,000. For each number of nodes, we simulate 100,000 networks from the block Hawkes model for a duration of $T = 20$ time units. We choose the Hawkes process parameters to be $\alpha = 5N$, $\beta = 10N$, and $\lambda^{\infty} = 0.5N$. The expected number of events $\mu = NT = 20N$, which grows with N and is slower than the growth of the size $n = N(N-1)$ of the block pair, so the Asymptotic Independence Theorem applies.

We evaluate the absolute difference between the empirical marginal probability $\widehat{\Pr}(a_{ij} = 0)$ and the empirical conditional probabilities $\widehat{\Pr}(a_{ij} = 0 | a_{i'j'} = 0)$ and $\widehat{\Pr}(a_{ij} = 0 | a_{i'j'} = 1)$. The empirical deviation from independence is shown to be well below the upper bound in Figure 2. The bound (3) in the Asymptotic Independence Theorem depends only on the mean number of events, so it is somewhat loose when applied to the block Hawkes model.

5.2 Class Estimation

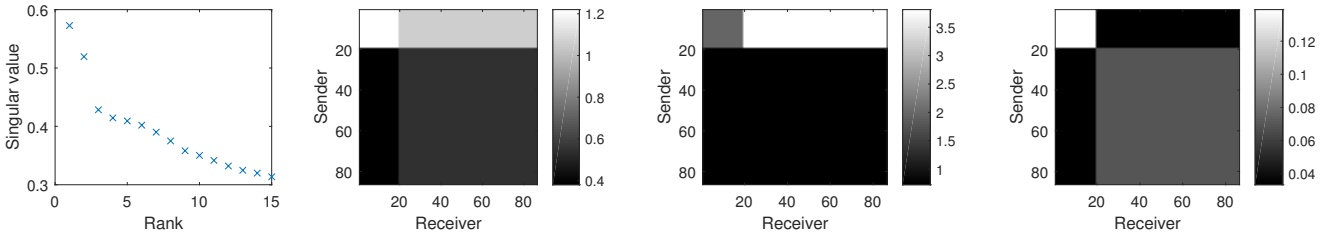
This simulation experiment is based on the synthetic network generator from Newman and Girvan [28], where all the diagonal block pairs have the same parameters, and the off-diagonal block pairs have the same parameters, but different from the diagonal block pairs. We generate networks with 128 nodes and 4 classes from the block Hawkes model using Algorithm 1 with varying durations from 20 to 80 time units. We generate 10 networks for each duration, with Hawkes process parameters α and β being 0.6 and 0.8, respectively, for all block pairs. The baseline rates λ^{∞} are 1.8 for diagonal block pairs and 0.6 for off-diagonal block pairs. Classes are estimated using 5 methods: spectral clustering, local search initialized with spectral clustering, variational EM initialized with spectral clustering, variational EM with 10 random initializations, and local search with 10 random initializations.

The results shown in Figure 3 demonstrate that local search initialized with spectral clustering (Spectral+LS) is the most accurate. Conversely, the spectral clustering initialization for variational EM (Spectral+VEM) results in significantly worse results. Variational EM and local search using 10 random initializations (Random+VEM and Random+LS) take significantly longer than Spectral+LS; however, they are also less accurate, so we use Spectral+LS in the remainder of this paper.

5.3 Scalability of Local Search

We evaluate the scalability of the proposed local search inference procedure by generating networks with varying number of nodes and events from the block Hawkes model. When varying the number of nodes, we choose a time duration of 1,200 time units and set the Hawkes process parameters $(\alpha, \beta, \lambda^{\infty})$ to $(1.6, 2, 1.2)$ for diagonal block pairs and $(0.6, 0.8, 0.6)$ for off-diagonal block pairs. When varying the number of events, we choose a network with 128 nodes and set the Hawkes process parameters $(\alpha, \beta, \lambda^{\infty})$ to $(0.6, 0.8, 1.8)$ for diagonal block pairs and $(0.6, 0.8, 0.6)$ for off-diagonal block pairs. We keep the number of classes fixed to be 4 in both settings and simulate 5 networks for each configuration.

The CPU times per iteration for both varying number of nodes and events are shown in Figure 4 along with best-fit lines for a



(a) Singular values (b) α: Jump sizes (c) β: Jump decay rates (d) λ^{∞}: Background intensity

Figure 5: Block Hawkes model fit on Reality Mining data with 2 blocks. Nodes 1-15 belong to block 1 and the rest to block 2. (a) First 15 singular values of the adjacency matrix. (b)-(d) Hawkes process parameter estimates.

power law relationship (beginning with 512 nodes for the varying nodes experiment). The best-fit line has slope 0.9 in both cases, confirming the linear time complexity both in terms of the number of nodes N and the number of events M, which is as expected according to the computed time complexity in Section 4.1. All CPU times are recorded on a Linux workstation using 18 Intel Xeon processor cores operating at 2.8 GHz.

6 REAL DATA EXPERIMENTS

6.1 MIT Reality Mining

We analyze the MIT Reality Mining data [9], using start times of phone calls as events from the caller to the recipient. Nodes in this data set correspond to students and staff at MIT. The data were collected over approximately 10 months beginning in August 2004. We remove all nodes who do not either make or receive a call at any point in the data trace, resulting in a network with 86 nodes.

We observe a significant gap after the second largest singular value of the regularized graph Laplacian, as shown in Figure 5a so we choose $K = 2$ blocks. From examining the block Hawkes model parameter estimates shown in Figure 5b-5d, block pair $(1, 1)$ has both higher background intensity λ^{∞} and longer bursts due to high α and low β. Thus, not only does it appear to be a community, but phone calls within the community tend to happen in prolonged bursts. On the other hand, block pair $(2, 2)$ has higher background intensity than the off-diagonal block pairs but without the large jump sizes, indicating a lack of bursty behavior compared to block pair $(1, 1)$. Block pair $(1, 2)$ has high values for both α and β, indicating large bursts of short duration. However, since block pair $(1, 2)$ has a lower background intensity than $(2, 2)$, the overall density of the block pair in the aggregated adjacency matrix is not necessarily higher. Indeed, block pair $(2, 2)$ has a density of 0.041, while block pair $(2, 1)$ has a density of 0.034. Thus, the continuous-time model of the network enables greater analysis of the dynamics of interactions over both short and long periods of time, and the findings may be quite different from those of static and discrete-time network representations.

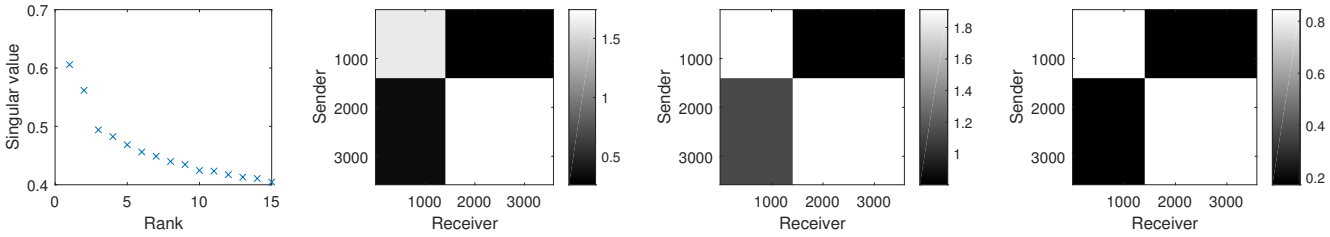
6.2 Facebook Wall Posts

6.2.1 Model-Based Exploratory Analysis. We analyze the Facebook wall post data collected by Viswanath et al. [37], which contains over 60,000 nodes. We consider events between January 1, 2007 and January 1, 2008. We remove nodes with degree less than 10, resulting in a network with 137,170 events among 3,582 nodes.

We select a model with $K = 2$ blocks, as suggested by the singular values of the regularized graph Laplacian shown in Figure 6a. The parameters inferred from the BHM fit on the Facebook data are shown in Figure 6b-6d. The diagonal block pairs have larger values of background intensity λ^{∞}, indicating that the blocks form communities. This finding could also have been yielded by static and discrete-time SBMs. The diagonal block pairs also have higher values of jump sizes α, indicating that wall posts between members of a community are more bursty. A portion of the Hawkes process intensity function for block pair $(1, 1)$ is shown in Figure 7. In addition to diurnal patterns, one can observe bursty periods of wall posts throughout the day. This finding could not have been obtained from static and discrete-time SBMs. From the values of α on the diagonal block pairs, we see that wall posts within block 2 are more bursty, with higher jump sizes and roughly equal jump decay rates compared to posts within block 1. By observing the values of α on off-diagonal block pairs, we notice that there isn't much asymmetry, but the decay rate β exhibits asymmetry. Specifically, events from block 1 to 2 have longer sustained bursts than events from block 2 to 1 due to the lower value of β_{12} compared to β_{21}.

6.2.2 Comparison with Discrete-Time SBM. To compare our proposed continuous-time BHM with a discrete-time SBM [41], we consider the task of predicting the time to the next event in each block pair. We believe that this is a fair comparison because both the BHM and discrete-time SBM require that pairs of nodes in the same block pair have identical edge probabilities. We split the data into two sets, with the first 8 months of data for training and the last 4 months for testing. During each week of the test set, we attempt to predict the time to the next event in each block pair. Afterwards, we update the model with all events during that week. This results in 16 predictions in the test set for each block pair.

The BHM directly models event times, so we use the expected next event time for each block as the prediction. The discrete-time SBM does not directly model event times, so we multiply the expected number of time snapshots that will elapse before the next edge formation, which is geometrically distributed, by the snapshot length and then subtract half the snapshot length (to center the prediction within the snapshot) to get a prediction for the next event time. Since the prediction for the discrete-time SBM is dependent on the snapshot length, we test several different snapshot lengths.



(a) Singular values (b) α: Jump sizes (c) β: Jump decay rates (d) λ^{∞}: Background intensity

Figure 6: Block Hawkes model parameter estimates on Facebook wall post data with 2 blocks. Nodes 1-1421 belong to block 1 and the rest to block 2. (a) First 15 singular values of the adjacency matrix. (b)-(d) Hawkes process parameter estimates.

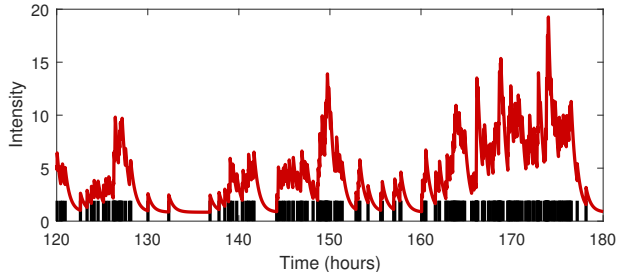


Figure 7: Hawkes process intensity function for events in block pair (1,1) of the Facebook data. Event times are denoted by black ticks along the horizontal axis.

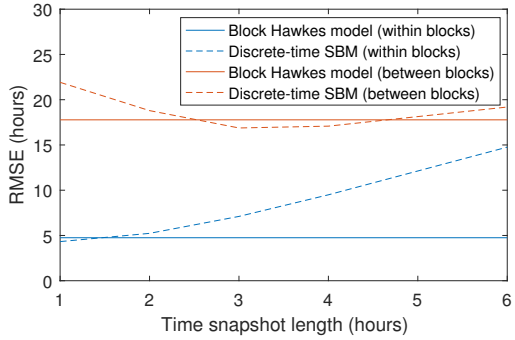


Figure 8: Prediction RMSE in hours for block Hawkes model and discrete-time SBMs on Facebook data.

We estimate class assignments for both models using regularized spectral clustering with 2 classes (with no local search iterations) so that differences in class estimates between the two models do not play a role in the accuracy. We believe this is a valid comparison because spectral clustering is used as the initialization to local search in the inference procedure for both models, as discussed in Section 4.1 for the BHM and in [41] for the discrete-time SBM.

We evaluate the accuracy of the predictions by computing the root mean-squared error (RMSE) between predicted event times and actual event times for the first event in each block pair during a week. Since the blocks form communities, we expect events to arrive much more frequently within blocks. Thus we separate the evaluation into within-block and between-block prediction RMSE. As shown in Figure 8, the accuracy of the discrete-time SBM is highly dependent on the snapshot length. For snapshots of 6 hours and longer, the loss in temporal resolution is the main contributor to the high RMSE. The shorter snapshots such as 1 hour and 2 hours have excellent temporal resolution for within-block prediction but are less accurate for between-block prediction. Choosing 3-hour long snapshots results in the most accurate between-block predictions. Due to the different rates of events within and between blocks, the discrete-time representation must trade off between within-block and between-block prediction accuracy when choosing the snapshot length. Using our continuous-time BHM, we avoid this complex problem of choosing the snapshot length and produce more accurate predictions (in total RMSE) than the discrete-time model for any snapshot length, as shown in Figure 8.

7 CONCLUSION

In this paper, we introduced the block point process model (BPPM) for dynamic networks evolving in continuous time in the form of timestamped events between nodes. Our model was inspired by the well-known stochastic block model (SBM) for static networks and is a simpler version of the Hawkes IRM. We demonstrated that adjacency matrices constructed from the BPPM follow an SBM in the limit of a growing number of nodes. To the best of our knowledge, this is the first result of this type connecting point process network models with adjacency matrix network models. Additionally we proposed a principled and efficient algorithm to fit the BPPM using spectral clustering and local search that scales to large networks and apply it to analyze several real networks.

A PROOF OF ASYMPTOTIC INDEPENDENCE THEOREM

We begin with a well-known lemma on the difference of powers that will be used both to upper and lower bound the deviation from independence.

Lemma A.1 (Difference of powers). *For a real number $x > 1$ and integer $m \geq 1$, we have the following identity:*

$$m(x-1)^{m-1} \leq x^m - (x-1)^m \leq mx^{m-1}. \tag{7}$$

Proof. The proof follows straightforwardly from factorizing a difference of powers. Specifically, for real numbers $x > y > 0$ and

integer $m \geq 1$,

$$x^m - y^m = (x-y)\sum_{i=0}^{m-1} x^{m-1-i}y^i. \tag{8}$$

If $x > 1$ and $y = x - 1$, then (8) becomes

$$x^m - (x-1)^m = \sum_{i=0}^{m-1} x^{m-1-i}(x-1)^i.$$

There are m terms in the summation. The largest term is x^{m-1}, and the smallest term is $(x-1)^{m-1}$. Thus, we can upper and lower bound the sum by mx^{m-1} and $m(x-1)^{m-1}$, respectively, to arrive at (7). □

The next lemma will be used in the upper bound.

Lemma A.2. *For a real number $x > 1$ and integer $m \geq 1$,*

$$\left(\frac{n-1}{n}\right)^m \geq 1 - \frac{m}{n}.$$

Proof.

$$\begin{aligned}\left(\frac{n-1}{n}\right)^m - 1 &= \frac{(n-1)^m - n^m}{n^m} \\ &= -\frac{n^m - (n-1)^m}{n^m} \\ &\geq -\frac{mn^{m-1}}{n^m} \qquad (9) \\ &= -\frac{m}{n}, \qquad (10)\end{aligned}$$

where (9) follows from applying Lemma A.1. Adding 1 to both sides of (10), we arrive at the desired result. □

We now prove the Asymptotic Independence Theorem.

Proof of Asymptotic Independence Theorem. First compute the marginal probability $\Pr(a_{ij} = 0)$. $a_{ij} = 0$ implies that no events between nodes i and j occurred. To compute this probability, we first compute the conditional probability given that the number of events in block pair b is m_b. To simplify notation, we drop the subscript b from m_b and n_b in the remainder of the proof, so the conditional probability can be written as

$$\Pr(a_{ij} = 0|m) = \left(\frac{n-1}{n}\right)^m,\ m \geq 0,$$

where the equality follows by noting that, conditioned on m total events in block pair b, the number of events between nodes i and j follows a binomial distribution with m trials and success probability $1/n$. The $1/n$ success probability is due to step 8 of the generative process of the BPPM, which involves selecting node pairs randomly to receive an event. By the Law of Total Probability, the marginal probability is

$$\begin{aligned}\Pr(a_{ij} = 0) &= \sum_{m=0}^{\infty} p(m)\Pr(a_{ij} = 0|m) \\ &= \sum_{m=0}^{\infty} p(m)\left(\frac{n-1}{n}\right)^m, \qquad (11)\end{aligned}$$

where the probability mass function $p(m)$ denotes the probability that m events in block pair b occurred.

Next consider the joint probability $\Pr(a_{ij} = 0, a_{i'j'} = 0)$. As before, condition on the number of events m. The conditional joint probability is

$$\Pr(a_{ij} = 0, a_{i'j'} = 0|m) = \left(\frac{n-2}{n}\right)^m,\ m \geq 0,$$

because the number of events for each node pair in block pair b follow a multinomial distribution with m trials and all event probabilities equal to $1/n$. By the Law of Total Probability,

$$\Pr(a_{ij} = 0, a_{i'j'} = 0) = \sum_{m=0}^{\infty} p(m)\left(\frac{n-2}{n}\right)^m. \tag{12}$$

We first lower bound δ_0 by noting that

$$\begin{aligned}\delta_0 &= \Pr(a_{ij} = 0|a_{i'j'} = 0) - \Pr(a_{ij} = 0) \\ &\geq \Pr(a_{ij} = 0, a_{i'j'} = 0) - \Pr(a_{ij} = 0) \\ &= \sum_{m=0}^{\infty} p(m)\left[\left(\frac{n-2}{n}\right)^m - \left(\frac{n-1}{n}\right)^m\right] \qquad (13) \\ &= -\sum_{m=0}^{\infty} p(m)\frac{(n-1)^m - (n-2)^m}{n^m} \\ &\geq -\sum_{m=0}^{\infty} p(m)\frac{m(n-1)^{m-1}}{n^m} \qquad (14) \\ &= -\frac{1}{n}\sum_{m=0}^{\infty} mp(m)\left(\frac{n-1}{n}\right)^{m-1} \\ &\geq -\frac{1}{n}\sum_{m=0}^{\infty} mp(m) \qquad (15) \\ &= -\frac{\mu}{n}, \qquad (16)\end{aligned}$$

where (13) follows from (11) and (12), (14) follows from Lemma A.1, and (15) follows by observing that $\left(\frac{n-1}{n}\right)^{m-1} \leq 1$.

We now upper bound δ_0 by noting that

$$\begin{aligned}\delta_0 &= \Pr(a_{ij} = 0|a_{i'j'} = 0) - \Pr(a_{ij} = 0) \\ &\leq 1 - \Pr(a_{ij} = 0) \\ &= 1 - \sum_{m=0}^{\infty} p(m)\left(\frac{n-1}{n}\right)^m \\ &\leq 1 - \sum_{m=0}^{\infty} p(m)\left(1 - \frac{m}{n}\right) \qquad (17) \\ &= 1 - \sum_{m=0}^{\infty} p(m) + \frac{1}{n}\sum_{m=0}^{\infty} mp(m) \\ &= 1 - 1 + \frac{\mu}{n} \\ &= \frac{\mu}{n}, \qquad (18)\end{aligned}$$

where (17) follows from Lemma A.2.

Next we lower bound δ_1 by noting that

$$\begin{aligned}
\delta_1 &= \Pr(a_{ij} = 0 | a_{i'j'} = 1) - \Pr(a_{ij} = 0) \\
&= \frac{\Pr(a_{ij} = 0, a_{i'j'} = 1) - \Pr(a_{ij} = 0)\Pr(a_{i'j'} = 1)}{\Pr(a_{i'j'} = 1)} \\
&\geq \Pr(a_{ij} = 0, a_{i'j'} = 1) - \Pr(a_{ij} = 0)\Pr(a_{i'j'} = 1) \\
&= \Pr(a_{ij} = 0) - \Pr(a_{ij} = 0, a_{i'j'} = 0) \\
&\quad - \Pr(a_{ij} = 0)[1 - \Pr(a_{i'j'} = 0)] \\
&= -[\Pr(a_{ij} = 0, a_{i'j'} = 0) - \Pr(a_{ij} = 0)\Pr(a_{i'j'} = 0)] \\
&= -\Pr(a_{i'j'} = 0)[\Pr(a_{ij} = 0 | a_{i'j'} = 0) - \Pr(a_{ij} = 0)] \\
&= -\Pr(a_{i'j'} = 0)\delta_0 \\
&\geq -\delta_0 \\
&\geq -\frac{\mu}{n}, \qquad (19)
\end{aligned}$$

where (19) follows from (18).

Finally we upper bound δ_1 using the same approach as for δ_0:

$$\begin{aligned}
\delta_1 &= \Pr(a_{ij} = 0 | a_{i'j'} = 1) - \Pr(a_{ij} = 0) \\
&\leq 1 - \Pr(a_{ij} = 0) \\
&\leq \frac{\mu}{n}, \qquad (20)
\end{aligned}$$

where the final inequality is obtained from (18).

Combining (16), (18), (19), and (20), and noting that the maximum deviation between two probabilities is 1, we arrive at the desired result $|\delta_0|, |\delta_1| \leq \min\{1, \mu/n\}$, which completes the proof. □

B VARIATIONAL INFERENCE DETAILS

We begin by deriving the complete-data log-likelihood for the block point process model.

$$\begin{aligned}
&\log \Pr(E, Z | \theta, \boldsymbol{\pi}) \\
&\quad = \log \Pr(Z | \boldsymbol{\pi}) + \log \Pr(\mathbf{u}, \mathbf{v}, \mathbf{t} | Z, \theta) \\
&\quad = \log \Pr(Z | \boldsymbol{\pi}) + \log \Pr(\mathbf{u}, \mathbf{v} | Z) + \log \Pr(\mathbf{t} | \mathbf{u}, \mathbf{v}, Z, \theta) \qquad (21)
\end{aligned}$$

The first term in (21) represents the prior class probability and is given by

$$\log \Pr(Z | \boldsymbol{\pi}) = \log \left(\prod_{i=1}^{N} \prod_{q=1}^{K} \pi_{iq}^{z_{iq}} \right) = \sum_{i=1}^{N} \sum_{q=1}^{K} z_{iq} \log(\pi_{iq}).$$

The second term in (21) represents the uniform distribution of events to nodes inside a block pair and is given by

$$\begin{aligned}
\log \Pr(\mathbf{u}, \mathbf{v} | Z) &= \log \left[\prod_{q=1}^{K} \prod_{l=1}^{K} \left(\frac{1}{n_{ql}} \right)^{m_{ql}} \right] \\
&= -\sum_{q=1}^{K} \sum_{l=1}^{K} m_{ql} \log(n_{ql}) \\
&= -\sum_{q=1}^{K} \sum_{l=1}^{K} \mathbf{z}_{:q}^T W \mathbf{z}_{:l} \log(\mathbf{z}_{:q}^T \mathbf{z}_{:l}),
\end{aligned}$$

where m_{ql} and n_{ql} are defined in the same manner as m_b and n_b in Section 4.1. The last equality represents these two terms as a function of Z, where $\mathbf{z}_{:q}$ denotes the qth column of $\mathbf{z}$, and a weighted adjacency matrix W where the weight w_{ij} corresponds to the total number of events from node i to node j. The third term in (21) depends on the choice of point process; for the block Hawkes model, it is given by

$$\begin{aligned}
\log \Pr(\mathbf{t} | \mathbf{u}, \mathbf{v}, Z, \theta) = \sum_{q=1}^{K} \sum_{l=1}^{K} \Bigg(\sum_{s=1}^{M} \Bigg\{ & z_{u_s q} z_{v_s l} \frac{\alpha_{ql}}{\beta_{ql}} \left[e^{-\beta_{ql}(t_M - t_s)} - 1 \right] \\
& + \log \left[\lambda_{ql}^{\infty} + \sum_{r=1}^{s-1} z_{u_r q} z_{v_r l} \alpha_{ql} e^{-\beta_{ql}(t_s - t_r)} \right] \Bigg\} - t_M \lambda_{ql}^{\infty} \Bigg),
\end{aligned}$$

where u_s and v_s denote the sender and receiver nodes for event s. The complete-data log-likelihood (21) is intractable since changing the class membership of one node might affect the class memberships of other nodes, so all possibilities of Z need to be considered.

We derive a mean-field variational inference procedure in which we approximate the posterior with the fully factorizable variational distribution given by (6). We now try to find the best distribution from the family of multinomial distributions that can get us closest in KL divergence to the posterior. But calculating KL divergence requires calculating the intractable posterior, so we instead maximize the evidence lower bound (ELBO), which is equivalent to minimizing KL divergence [1]. The ELBO is given by

$$\text{ELBO}(R_E) = \mathbb{E}\left[\log \Pr(E, Z | \theta, \boldsymbol{\pi})\right] - \mathbb{E}\left[\log R_E(Z)\right].$$

Now expand and calculate expectation of individual terms in the complete-data log-likelihood (21). The expectation of the log of the summations in the second and third terms in (21) can be bounded using Jensen's inequality:

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}(x)).$$

After bounding the log sums using Jensen's inequality and calculating the expectations with respect to R_E, we arrive at the simplified expression for the ELBO:

$$\begin{aligned}
\mathcal{E}(R_E) = & \sum_{i=1}^{N} \sum_{q=1}^{K} \tau_{iq} \log(\pi_{iq}) - \sum_{q=1}^{K} \sum_{l=1}^{K} \boldsymbol{\tau}_{:q}^T W \boldsymbol{\tau}_{:l} \log(\boldsymbol{\tau}_{:q}^T \boldsymbol{\tau}_{:l}) \\
& + \sum_{q=1}^{K} \sum_{l=1}^{K} \Bigg(\sum_{s=1}^{M} \Bigg\{ \tau_{u_s q} \tau_{v_s l} \frac{\alpha_{ql}}{\beta_{ql}} \left[e^{-\beta_{ql}(t_M - t_s)} - 1 \right] + \log \Bigg[\lambda_{ql}^{\infty} \\
& + \sum_{r=1}^{s-1} \tau_{u_r q} \tau_{v_r l} \alpha_{ql} e^{-\beta_{ql}(t_s - t_r)} \Bigg] \Bigg\} - t_M \lambda_{ql}^{\infty} \Bigg) + \sum_{i=1}^{N} \sum_{q=1}^{K} \tau_{iq} \log(\tau_{iq}) \qquad (22)
\end{aligned}$$

where $\boldsymbol{\tau}_{:q}$ denotes the vector of variational parameters for class q over all N nodes. The variational expectation-maximization (VEM) algorithm alternates between a variational E step, in which we use coordinate ascent to optimize the ELBO (22) over the variational parameters $\boldsymbol{\tau}_i$ for each node i, and an M step, in which we optimize the ELBO over the Hawkes process parameters $\left(\alpha_{ql}, \beta_{ql}, \lambda_{ql}^{\infty}\right)$ for all block pairs $(q, l) \in \{1, \ldots, K\}^2$.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation grants IIS-1755824 and DMS-1830412.

REFERENCES

[1] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[2] Charles Blundell, Jeff Beck, and Katherine A. Heller. 2012. Modelling Reciprocating Relationships with Hawkes Processes. In *Advances in Neural Information Processing Systems 25*. 2600–2608.

[3] Richard H. Byrd, Jean Charles Gilbert, and Jorge Nocedal. 2000. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming* 89, 1 (2000), 149–185.

[4] Etienne Côme and Pierre Latouche. 2015. Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modelling* 15, 6 (2015), 564–589.

[5] Marco Corneli, Pierre Latouche, and Fabrice Rossi. 2016. Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. *Neurocomputing* 192 (2016), 81–91.

[6] Jean-Jacques Daudin, Franck Picard, and Stéphane Robin. 2008. A mixture model for random graphs. *Statistics and Computing* 18, 2 (2008), 173–183.

[7] Christopher DuBois, Carter T. Butts, and Padhraic Smyth. 2013. Stochastic blockmodeling of relational event dynamics. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*. 238–246.

[8] Christopher DuBois and Padhraic Smyth. 2010. Modeling relational events via latent classes. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 803–812.

[9] Nathan Eagle, Alex Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences* 106 (2009), 15274–15278.

[10] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. 2015. COEVOLVE: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems 28*. 1945–1953.

[11] Eric W. Fox, Martin B. Short, Frederic P. Schoenberg, Kathryn D. Coronges, and Andrea L. Bertozzi. 2016. Modeling e-mail networks and inferring leadership using self-exciting point processes. *J. Amer. Statist. Assoc.* 111, 514 (2016), 564–584.

[12] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. 2010. A survey of statistical network models. *Foundations and Trends® in Machine Learning* 2, 2 (2010), 129–233.

[13] Eric C. Hall and Rebecca M. Willett. 2016. Tracking dynamic point processes on networks. *IEEE Transactions on Information Theory* 62, 7 (2016), 4327–4346.

[14] Peter F. Halpin and Paul De Boeck. 2013. Modelling dyadic interaction with Hawkes processes. *Psychometrika* 78, 4 (2013), 793–814.

[15] Qiuyi Han, Kevin S. Xu, and Edoardo M. Airoldi. 2015. Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd International Conference on Machine Learning*. 1511–1520.

[16] Xinran He, Theodoros Rekatsinas, James Foulds, Lise Getoor, and Yan Liu. 2015. HawkesTopic: A joint model for network inference and topic modeling from text-based cascades. In *Proceedings of the 32nd International Conference on Machine Learning*. 871–880.

[17] Qirong Ho, Le Song, and Eric P. Xing. 2011. Evolving cluster mixed-membership blockmodel for time-varying networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. 342–350.

[18] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5, 2 (1983), 109–137.

[19] Brian Karrer and M. E. J. Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E* 83, 1 (2011), 016107.

[20] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*. 381–388.

[21] Patrick J. Laub, Thomas Taimre, and Philip K. Pollett. 2015. Hawkes processes. *arXiv preprint arXiv:1507.02822* (2015). http://arxiv.org/abs/1507.02822

[22] Jing Lei and Alessandro Rinaldo. 2015. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* 43, 1 (2015), 215–237.

[23] Scott W. Linderman and Ryan P. Adams. 2014. Discovering latent network structure in point process data. In *Proceedings of the 31st International Conference on Machine Learning*. 1413–1421.

[24] Scott W. Linderman and Ryan P. Adams. 2015. Scalable Bayesian inference for excitatory point process networks. *arXiv preprint arXiv:1507.03228* (2015). https://arxiv.org/abs/1507.03228

[25] Naoki Masuda, Taro Takaguchi, Nobuo Sato, and Kazuo Yano. 2013. Self-exciting point process modeling of conversation event sequences. In *Temporal Networks*. Springer, 245–264.

[26] Catherine Matias and Vincent Miele. 2017. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79, 4 (2017), 1119–1141.

[27] Catherine Matias, Tabea Rebafka, and Fanny Villers. 2018. A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika* 105, 3 (2018), 665–680.

[28] M. E. J. Newman and M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113.

[29] Yosihiko Ogata. 1981. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory* 27, 1 (1981), 23–31.

[30] T. Ozaki. 1979. Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics* 31 (1979), 145–155.

[31] Tai Qin and Karl Rohe. 2013. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems 26*. 3120–3128.

[32] Riccardo Rastelli. 2017. Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks. *arXiv preprint arXiv:1710.03551* (2017). https://arxiv.org/abs/1710.03551

[33] Karl Rohe, Sourav Chatterjee, and Bin Yu. 2011. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics* 39, 4 (2011), 1878–1915.

[34] Karl Rohe, Tai Qin, and Bin Yu. 2016. Co-clustering directed graphs to discover asymmetries and directional communities. *Proceedings of the National Academy of Sciences* 113, 45 (2016), 12679–12684.

[35] Daniel L. Sussman, Minh Tang, Donniell E. Fishkind, and Carey E. Priebe. 2012. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *J. Amer. Statist. Assoc.* 107, 499 (2012), 1119–1128.

[36] Long Tran, Mehrdad Farajtabar, Le Song, and Hongyuan Zha. 2015. NetCodec: Community detection from individual activities. In *Proceedings of the SIAM International Conference on Data Mining*. 91–99.

[37] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the evolution of user interaction in Facebook. In *Proceedings of the 2nd ACM Workshop on Online Social Networks*. ACM, 37–42.

[38] Lu Xin, Mu Zhu, and Hugh Chipman. 2017. A continuous-time stochastic block model for basketball networks. *The Annals of Applied Statistics* 11, 2 (2017), 553–597.

[39] Eric P. Xing, Wenjie Fu, and Le Song. 2010. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics* 4 (2010), 535–566.

[40] Kevin S. Xu. 2015. Stochastic block transition models for dynamic networks. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*. 1079–1087.

[41] Kevin S. Xu and Alfred O. Hero III. 2014. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing* 8, 4 (2014), 552–562.

[42] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. 2011. Detecting communities and their evolutions in dynamic social networks—a Bayesian approach. *Machine Learning* 82, 2 (2011), 157–189.

[43] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. 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*. 1513–1522.

[44] Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. 2012. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics* 40, 4 (2012), 2266–2292.