The Indian Buffet Hawkes Process to Model Evolving Latent Influences

Xi Tan[1]**, Vinayak Rao**[2]**, and Jennifer Neville**[1,2]
[1]Department of Computer Science and [2]Department of Statistics
Purdue University
West Lafayette, IN 47907

Abstract

Temporal events in the real world often exhibit reinforcing dynamics, where earlier events trigger follow-up activity in the near future. A canonical example of modeling such dynamics is the Hawkes process (HP). However, previous HP models do not capture the rich dynamics of real-world activity—which can be driven by multiple latent triggering factors shared by past and future events, with the latent features themselves exhibiting temporal dependency structures. For instance, rather than view a new document just as a response to other documents in the recent past, it is important to account for the factor-structure underlying all previous documents. This structure itself is not fixed, with the influence of earlier documents decaying with time. To this end, we propose a novel Bayesian nonparametric stochastic point process model, the Indian Buffet Hawkes Processes (IBHP), to learn multiple latent triggering factors underlying streaming document/message data. The IBP facilitates the inclusion of multiple triggering factors in the HP, and the HP allows for modeling latent factor evolution in the IBP. We develop a learning algorithm for the IBHP based on Sequential Monte Carlo and demonstrate the effectiveness of the model. In both synthetic and real data experiments, our model achieves equivalent or higher likelihood and provides interpretable topics and shows their dynamics.

1 INTRODUCTION

Temporal activity in real applications exhibit rich dynamics, with past events influencing the future through multiple structured latent factors. For example, the ideas in a research paper may be derived from multiple existing works in the literature, each of which contributes one or more factors, with only their combination serving to trigger the event. Similarly, a conversation among individuals may heat up or cool down due to the topics being discussed (e.g., politics vs. weather). Communications via email or on social media platforms like Facebook or Twitter may exhibit analogous dynamics. In addition, individual checkin data on platforms like Foursquare or Yelp may depend on combinations of characteristics and activities from previous visited locations. Finally in biological data, pathways are often only activated when a set of genes is expressed together.

Latent feature models (both parametric and nonparametric) have found wide application in settings where exchangeability holds. A canonical model from Bayesian nonparametrics is the Indian Buffet process (IBP). While there has been some work towards relaxing exchangeability assumptions to allow for temporal dynamics, modeling the full richness of interactions remains an open challenge. Our main contribution in this work is a framework that facilitates the modeling of temporal dynamics through a combination of ideas from the IBP with those of Hawkes processes (HP).

In recent years, Hawkes processes [11, 10, 12, 15, 22] have become a popular modeling choice to capture such temporal dynamics [4, 18, 14, 19, 13, 7, 17, 8, 16, 21]. As we outline later, the standard Hawkes process has a number of limitations centering around the fact that each event is triggered by a single observation instead of possibly multiple events and/or factors. To this end, we propose a novel Bayesian nonparametric stochastic point process model, the Indian Buffet Hawkes Processes (IBHP), that synergizes ideas between the IBP and the HP. The contributions of our work include:

1. The use of the IBP to add multiple triggering factors to the HP, which helps to better model dynamics and improves interpretation.

2. Embedding the temporal information from the HP into the IBP to drive the latent factor estimation,

which expands its capability to model factor evolution over time.

3. Developing an efficient and scalable learning algorithm for the IBHP model, based on Sequential Monte Carlo (SMC).
4. Demonstrating the effectiveness of the IBHP on both synthetic and four real-world datasets, where we also show how our framework enables the construction of more flexible (e.g., multi-event) triggering rules.

2 PRELIMINARIES

Formally, we are given a sequence of N observations $\mathbf{y}_n = \{t_n, \mathcal{T}_n\}$, $n = 1, \ldots, N$. For the nth event, t_n is the time of occurrence, and $\mathcal{T}_n$ represents observed attributes attached to it. Since our focus is mostly on settings where events are messages, we will refer to attributes as text. We will model the event times t_n and event text $\mathcal{T}_n$ as realizations of a process whose states depend on a hidden state variable $\mathbf{z}_n$, summarizing the past observations $\mathbf{y}_{1:(n-1)}$. Before outlining our model, we first review existing work related to our problem.

2.1 HAWKES PROCESSES (HP)

Hawkes processes (HP) [11, 10, 12, 15, 22] are self-exciting point processes [5] where earlier events have a time-decaying influence on future events. Parametrized by a base rate γ and a non-negative triggering kernel $\kappa(\cdot)$ (the latter models the contribution of each past observation), the rate function at time t can be written as:

$$\lambda(t) = \gamma + \int_0^t \kappa(t-s)\,\mathrm{d}\,N(s) \tag{1}$$

where $N(s)$ is the number of observations within $[0, s)$. Given the rate function $\lambda(t)$ and observation history $\mathcal{H}_{(0,T]} = (t_1, \cdots, t_n)$, the likelihood function of a Hawkes process is:

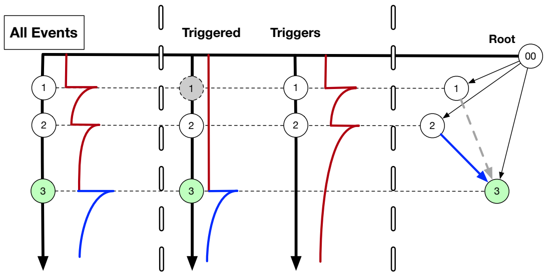
$$\mathcal{L}(\mathcal{H}) = \exp\{-\Lambda(0,T)\} \prod_{i=1}^{n} \lambda(t_i) \tag{2}$$

where $\Lambda(0,T) = \int_0^T \lambda(t)\,\mathrm{d}\,t$ is the cumulative rate. The events in a standard HP are triggered by a single event at a time (see Figure 1).

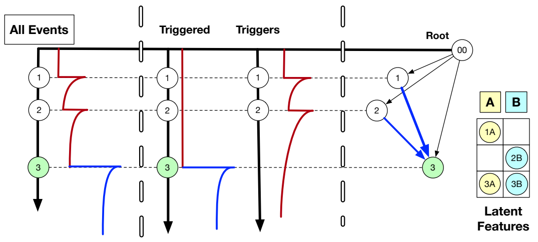
2.2 INDIAN BUFFET PROCESSES (IBP)

The Indian Buffet Process (IBP) [9] is a Bayesian non-parametric prior over an infinite dimensional binary matrix whose columns represent exchangeable factors underlying observations. Suppose there are N customers (observations) arriving sequentially in a restaurant with infinite number of dishes (factors). Each customer is assigned dishes as follows:

- The first customer comes in and helps herself to Poisson(α) dishes.



(a) A Hawkes process with single triggers.



(b) A Hawkes process with multiple triggers.

Figure 1: HP with single and multiple triggers. In (a), #3 is triggered by a single event #1, while in (b) it is triggered by #1 and #2. The triggering kernels can be quite different depending on how the triggering has happened. HP with single triggers would fail to model influences from both #1 and #2 at the same time, as shown in (b).

- When the n^{th} customer arrives, they independently choose each existing dish with probability m_k/n, where m_k is the number of customers that have already sampled dish k (the popularity of the dish).
- In addition, they sample Poisson(α/n) new dishes.

The IBP has several distinctive features: 1) Each observation can have multiple factors; 2) The number of factors grows non-parametrically depending on the size of the dataset; 3) The probability of adding new factors decreases – since the number of new factors follows Poisson(α/n) which decreases as n increases; 4) The rows are exchangeable, with the row sums distributed Poisson(α).

We write the binary feature matrix as $\mathbf{C}$. The conditional probability that element $c_{ik} = 1$ is given by

$$P(c_{ik} = 1 | \mathbf{c}_{-i,k}) = \frac{m_{-ik}}{N} \tag{3}$$

where $\mathbf{c}_{-ik}$ is the k^{th} column without considering the i^{th} observation, and m_{-ik} is the sum of $\mathbf{c}_{-ik}$. We need only condition on $\mathbf{c}_{-ik}$ rather than including other columns because the columns of the matrix are generated independently under this prior. In a Bayesian framework with observations $\mathbf{X}$, the posterior can be written as:

$$P(c_{ik} = 1 | \mathbf{C}_{-ik}, \mathbf{X}) \propto P(\mathbf{X}|\mathbf{C}) P(c_{ik} = 1 | \mathbf{c}_{-ik}) \tag{4}$$

where $P(\mathbf{X}|\mathbf{C})$ is the data likelihood.

3 MODEL

Our proposed model, the IBHP, can be viewed as a non-parametric latent state space model, where past events $\mathbf{y}_n = \{t_n, \mathcal{T}_n\}$ influence future observations through latent state variables $\mathbf{z}_n = \{\mathbf{K}_n, \mathbf{V}_n\}$ (described below). The $\mathbf{z}_n$'s summarize information about the past, and themselves evolve following dynamics based on the IBP. Algorithmically, the generative process (see Algorithm 1 for the pseudocode and Figure 2 for an illustrative example) can be described in the following three steps.

3.1 INITIALIZATION ($\mathcal{M}, \mathbf{\Pi}, \mathbf{\Theta}$)

To setup the model, we first specify a triplet $\mathcal{M} = \{\mathcal{S}, D, L\}$, with $\mathcal{S}$ representing the vocabulary of all possible words in the observations, D representing the length of each document (for simplicity we assume all are equal), and L representing the number of basis kernels. We also require a pair of hyper parameters $\mathbf{\Pi} = \{\mathbf{w}_0, \mathbf{v}_0\}$ for the priors of the kernel and word distribution weights.

Each latent factor influences the content of future events through a set of *dictionary weights*, which are used to generate text. We write $\mathbf{v}_k$ for the vector of weights over the words in the vocabulary for the k^{th} factor – a length $|S|$ vector which sums to one. The weights $\mathbf{v}_k$ are sampled from a Dirichlet prior (with hyper parameter $\mathbf{v}_0$) whenever a new factor is created (see later). Each latent factor also influences the timing of future events through a *triggering kernel*, and we assume each kernel is a linear combination of a set of L bases. Throughout, we assume L *exponential* basis kernels:

$$\gamma_l(\delta) = \beta_l e^{-\frac{\delta}{\tau_l}}, \quad l = 1, \dots, L. \tag{5}$$

This requires a set of parameters $\{(\beta_l, \tau_l)\}$, each of which captures a distinct type of excitation pattern. A binary matrix $\mathbf{C}$ indicates which factors are associated with each observation. The k^{th} *factor kernel* for the i^{th} observation κ_{ik} is a weighted sum of the L basis kernels:

$$\kappa_{ik}(\delta|\mathbf{w}_k, c_{ik}) = \begin{cases} \sum_{l=1}^{L} w_{kl} \cdot \gamma_l(\delta), & \text{if } c_{ik} = 1 \\ 0, & \text{if } c_{ik} = 0 \end{cases} \tag{6}$$

The weights w_{kl}, loadings of the basis kernels for the factors, are sampled from a Dirichlet prior (with hyper parameter $\mathbf{w}_0$) whenever a new factor is created (see later). Thus, immediately after an event (when $\delta = 0$), there is a jump in the event rate with amplitude equal to $\kappa_{ik} = \mathbf{w}_k^\intercal \boldsymbol{\beta}$. Observations with the same factor share the factor kernel. We write the model parameters as $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$, where λ_0 is a base-rate at which events happen spontaneously.

3.2 THE FIRST EVENT ($\mathcal{M}, \mathbf{\Theta} \to \mathbf{z}_1 \to \mathbf{y}_1$)

To generate the observation $\mathbf{y}_1 = \{t_1, \mathcal{T}_1\}$, we first sample the auxiliary variables $\mathbf{c}_1$ and $\mathbf{w}_{1:K}$. The *factor label variable* $\mathbf{c}_1$ is a binary vector of length K, where $K \sim \text{Poisson}(\lambda_0)$ is the number of existing factors. $c_{nk} = 1$ implies that the n^{th} observation has a label of factor k. Set $c_{1k} = 1$ for $k = 1, \dots, K$. The *kernel weights* $\mathbf{w}_k$ is a vector of weights for the k^{th} factor to load the basis kernels. Each $\mathbf{w}_k$ is of length L (the number of basis kernels), and sums to one. Sample $\mathbf{w}_k \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ for $k = 1, \dots, K$.

Given the values of $\mathbf{c}_1$ and $\mathbf{w}_{1:K}$, we can sample the associated latent variables $\mathbf{z}_1 = \{\mathbf{K}_1, \mathbf{V}_1\}$. Define the $1 \times K$ *IBHP matrix* $\mathbf{K}_1$, whose rows are $\boldsymbol{\kappa}_1$, with values $\mathbf{w}_k^\intercal \boldsymbol{\beta}$ (see Equation 6) – since $\delta = 0$. For $n = 1$, sample $\mathbf{v}_k \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$ for $k = 1, \dots, K$, and define the $|\mathcal{S}| \times K$ matrix $\mathbf{V}_1$, whose columns are $\mathbf{v}_k$. Conditioned on these state variables $\mathbf{z}_1$, we sample the first observation $\mathbf{y}_1 = \{t_1, \mathcal{T}_1\}$: The *time stamp* t_1 is sampled from a Poisson process with rate λ_0; and the *text* $\mathcal{T}_1$ is sampled from $\text{Multi}(D, \sum_{k=1}^{K} \mathbf{v}_k / K)$, where the weight parameter is the averaged factor weight of the first observation.

3.3 FOLLOW-UP EVENTS ($\mathbf{z}_{n-1} \to \mathbf{z}_n \to \mathbf{y}_n$)

For a new event, we first decide its associated factors, and sample its time stamp afterwards. Conditioning on $\mathbf{z}_{n-1}$, suppose there are K existing factors, each of which can be represented by an independent Hawkes process. At time t_{n-1}, the *factor rate* for the k^{th} factor is:

$$\lambda_k(t_{n-1}) = \sum_{i=1}^{n-1} \frac{\kappa_{ik}(t_{n-1} - t_i)}{\|\boldsymbol{\kappa}_i\|_0} \tag{7}$$

As with the generation of the initial event, follow-up events ($n > 1$) are also generated by two steps. First, we sample the auxiliary variables $\mathbf{c}_n$ and set $\mathbf{w}$ and $\mathbf{v}$ for any newly generated factors. The first K components of the *factor label variable* $\mathbf{c}_n$ is sampled independently from a Bernoulli distribution with probability parameter

$$p_k = \frac{\lambda_k(t_{n-1})}{\lambda_0/K + \lambda_k(t_{n-1})} \tag{8}$$

Meanwhile, K^+ new factors are created by setting $c_{nk'} = 1$, for $k' = K + \{1, \dots, K^+\}$, where

$$K^+ \sim \text{Poisson}\left(\frac{\lambda_0}{\lambda_0 + \sum_{k=1}^{K} \lambda_k(t_{n-1})} \right) \tag{9}$$

If κ are binary, which is the case in IBP, and $\lambda_0 = 1$, then the mean of K^+ becomes $1/n$ and $p_k = (n-1)/n$, which reduces to the case of IBP with parameter 1:

$$\sum_{k=1}^{K} \sum_{i=1}^{n-1} \frac{\kappa_{ik}(t_{n-1} - t_i)}{\|\boldsymbol{\kappa}_i\|_0} = \sum_{i=1}^{n-1} \frac{\|\boldsymbol{\kappa}_i\|_0}{\|\boldsymbol{\kappa}_i\|_0} = n - 1 \tag{10}$$

For each new factor k', we draw from the corresponding priors for $\mathbf{w}_{k'} \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ and $\mathbf{v}_{k'} \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$.

Next, we decide the hidden state variables $\mathbf{z}_n = \{\mathbf{K}_n, \mathbf{V}_n\}$. $\mathbf{V}_n$ is constructed by simply adding columns for the $\mathbf{v}_{k'}$ for newly sampled factors to $\mathbf{V}_{n-1}$. $\mathbf{K}_n$ is constructed by first updating $\mathbf{K}_{n-1}$ with respect to the new lag time $\delta = t_n - t_i$. This step is done *symbolically*, since we do not know t_n yet. Then we add the rows $\kappa_{ik'}$ for the newly sampled event based on Equation 6 with $\delta = 0$. We emphasize that $K_n(t_n) : \mathbf{R}^+ \to \mathbf{R}^{n\times(K+K')}$ at this moment is a symbolic function of t_n.

Conditioned on these state variables $\mathbf{z}_n$, we sample the n^{th} observation $\mathbf{y}_n = \{t_n, \mathcal{T}_n\}$: The *time stamp* t_n, depending on its related factors, is sampled from a Poisson process with rate

$$\lambda(t_n) = \sum_{\kappa_{nk}\neq 0} \lambda_k(t_n) = \sum_{\kappa_{nk}\neq 0} \sum_{i=1}^{n} \frac{\kappa_{ik}(t_n - t_i)}{\|\boldsymbol{\kappa}_i\|_0} \quad (11)$$

The overall rate of IBHP, however, includes the base rate and other factors too:

$$\bar{\lambda}(t_n) = \lambda_0 + \lambda(t_n) + \sum_{\kappa_{nk}=0} \lambda_k(t_n) \quad (12)$$

Now, at this point, since t_n is known, we can proceed to compute the actual values of $\mathbf{K}_n$. Finally, we sample the *dictionary text* $\mathcal{T}_n$ from Multi$(D, \sum_{\kappa_{nk}\neq 0} \mathbf{v}_k/\|\boldsymbol{\kappa}_n\|_0)$, where the weight parameter is the averaged of all $\|\boldsymbol{\kappa}_n\|_0$ factor weights associated with the n^{th} observation.

4 POSTERIOR INFERENCE

4.1 SMC FOR IBHP

Sequential Monte Carlo [6] (SMC) methods are powerful and flexible tools for posterior inference in time-series models such as ours. Here, we adapt particle filtering methods to our set up, allowing us to scale our model to large-data regimes. The idea here is to represent the state of the system at any time (from 1 to N) with a set of F particles. We build on ideas from [20], extending them to our more structured setting.

The idea at a high level is to propagate each particle forward by one time step according to the prior, and then reweight each particle by how "compatible" it is with the observation at that time. If the effective number of particles is small (according to their weights), then the algorithm resamples the particles with replacement. Our algorithm to learn the IBHP factors and model parameters can be described as follows (see Algorithm 2 for the pseudocode):

A. Initialize Particle Weights. The particle weights are initialized uniformly: $u_1^f = \frac{1}{F}$, for $f = 1, \ldots, F$. Then for each time step $i = [1 \ldots N]$, we do the following:

```
1. Initialization:
- Model specifications: $\mathcal{M} = \{L, D, \mathcal{S}\}$;
- Model hyper parameters: $\mathbf{\Pi} = \{\mathbf{w}_0, \mathbf{v}_0\}$;
- Model parameters: $\mathbf{\Theta} = \{\lambda_0, \{\beta_l, \tau_l\}\}$;
2. Generate the First Event:
- Set $c_{1,1:K} = 1$, where $K \sim Poisson(\alpha_0)$;
- Sample $\mathbf{w}_k \sim$ Dir$(\mathbf{w}|\mathbf{w}_0)$ and set $\boldsymbol{\kappa}_1$;
- Sample $\mathbf{v}_k \sim$ Dir$(\mathbf{v}|\mathbf{v}_0)$;
- Sample $t_1 \sim \mathcal{PP}(\lambda_0)$;
- Sample $\mathcal{T}_1 \sim$ Multi $\left(D, \sum_{\kappa_{1k}\neq 0} \mathbf{v}_k/\|\boldsymbol{\kappa}_1\|_0\right)$
3. Generate Follow-up Events:
for $n = 2, \ldots, N$ do
    - Sample $\mathbf{c}_n$ according to Equations 8 and 9.
    - Sample $\mathbf{w}_{k'} \sim$ Dir$(\mathbf{w}|\mathbf{w}_0)$ and set $\boldsymbol{\kappa}_n$;
    - Sample $\mathbf{v}_{k'} \sim$ Dir$(\mathbf{v}|\mathbf{v}_0)$;
    - Sample $t_n \sim \mathcal{PP}(\lambda(t_n))$ by Equation 11.
    - Sample
      $\mathcal{T}_n \sim$ Multi $\left(D, \sum_{\kappa_{nk}\neq 0} \mathbf{v}_k/\|\boldsymbol{\kappa}_n\|_0\right)$.
end
```

Algorithm 1: Generative process of IBHP.

B. Sample Particles. According to [20], our particles $\tilde{\mathbf{z}}_i^f = \{\tilde{\mathbf{K}}_i^f, \tilde{\mathbf{V}}_i^f\}$ are sampled based on the conditional distributions $p(\mathbf{z}_i|\mathbf{z}_{i-1})$ described in Section 3.3.

C. Sample Model Parameters. Since the posterior of the model parameter $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$ is proportional to the product of its prior and the data likelihood described in Equation 2 and Section 3, we can first sample from its prior, and then use the product of the prior and the HP data likelihood as weights of the samples to approximate the posterior [8]. We update the triggering kernels using the new parameters.

D. Update Particle Weights. The importance weight is the ratio between the true posterior and the proposal distribution. Since we use the prior as the proposal, we update the particle weights by $u_i^f = u_{i-1}^f p(\mathbf{y}_i|\tilde{\mathbf{z}}_i^f, \mathbf{\Theta})$ and then normalize them to $u_j^f = u_j^f/(\sum_{f=1}^F u_j^f)$.

E. Resample Particles. If the effective number of particles is too small, we resample with replacement F particles from the existing ones with the normalized weights.

4.2 COMPLEXITY AND SCALABILITY

The SMC algorithm for the IBHP is easy to implement and scalable. Due to the sequential updating strategy, the time complexity of this algorithm is $\mathcal{O}(NF)$, where N is the number of observations and F is the number of particles. We will demonstrate and discuss the effectiveness of the algorithm in the experiment section in more detail.

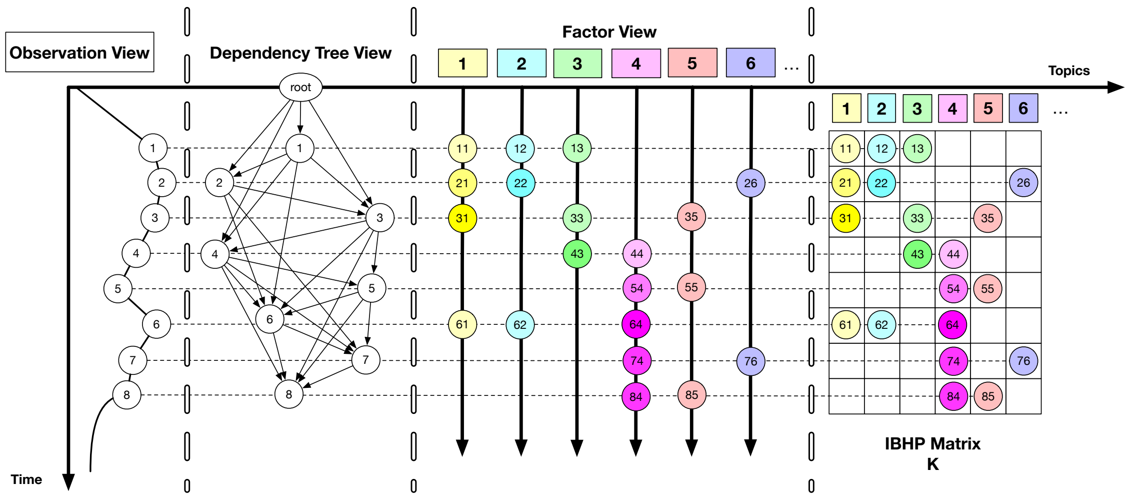


Figure 2: An example of IBHP. In this IBHP realization, the first 8 observations created 6 factors. Each factor has a distinctive color, and color intensities represent instantaneous factor popularities. An observation may be labeled with multiple factors, and are colored in its decomposed factor view accordingly. The dependency tree describes the related events for each observation, where the directed arrows indicate dependency relations. The rate for any *observation* is the aggregation of all its *related factor* rates (see Equation 11), whereas the overall rate at any *time* is the sum of *all factor* rates – so the overall rate can be excited by one observation multiple times through different factors. The overall rate is represented by its height relative to the reference time line. See Section 6.1 for more details.

```
Initialize the $F$ uniform particle weights.
for each observation $\mathbf{y}_i = \{t_i, \mathcal{T}_i\}$, $i = 1, \dots, n$ do
    for each particle $\mathbf{z}_i^f = \{\mathbf{K}_i, \mathbf{V}_i\}$ of observation $\mathbf{y}_i$, $f \in \{1, \dots, F\}$ do
        - Sample the auxiliary variables $\mathbf{w}_i, \mathbf{c}_i$ and latent factor particles $\mathbf{z}_i^f = \{\mathbf{K}_i, \mathbf{V}_i\}$.
        - Sample the model parameters $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$.
        - Update the triggering kernels.
        - Update the particle weights $\mathbf{u}_i^f$.
    end
    Normalize the particle weights.
    if $\|\mathbf{u}_i\|_2^{-2} <$ threshold, i.e., the effective number of particles is too low then
        Resample particles with replacement based on the particle weights.
    end
end
```

Algorithm 2: SMC inference algorithm for IBHP.

5 RELATED WORK

The idea of considering nonparametric Bayesian models with temporal point processes in a unified framework has been popular in recent years. For example, [4] proposed a Bayesian nonparametric model that utilizes the Chinese Restaurant Processes (CRP) as a prior for the clusters among individuals, whose rates of communications are modeled by HP. [18] used a similar idea but further extended the model by modeling the jump sizes of HP using Gaussian Processes (GP). HP models with various generalizations of a CRP, such as the distance dependent CRP (ddCRP) [14], the nested CRP (nCRP) [19], and the Chinese Restaurant Franchise Processes (CRFP) [13], have also been explored.

Other attempts have been made by borrowing the ideas from Deep Learning. For example, [7] proposed a model to view the intensity function of a temporal point process as a nonlinear function of the history, and use recurrent neural networks to automatically learn a representation of the influences from the event history. [17] modeled streams of events by constructing a neurally self-modulating multivariate point process where the intensities of multiple event types evolve based on a continuous-time LSTM. Lastly, [21] considered the use of latent factors in HP models to represent dependencies among instances that influence reciprocity over time. But the work focused on modeling static factors of homophily and reciprocity in social networks and not the evolution of factors over time.

Perhaps the closest works to our model are [8] and [16]. In [8], the authors proposed a Dirichlet Hawkes Processes (DHP) model that combines the CRP with HP in a unified framework, where the cluster assignment in CRP is driven by the intensities of HP. [16] further developed this in their Hierarchical Dirichlet Hawkes Processes (HDHP) model by replacing the CRP with a CRFP that is capable of modeling steaming data for multiple users. However, there are several major distinctions compared to our IBHP: 1) In both the DHP and HDHP models, events are triggered by single factors, while in our IBHP, multiple latent triggering factors are introduced; 2) the form of the triggering kernels do not depend on history events, and in contrast, our IBHP model is more flexible to be able to adopt non-additive triggering rules to learn different perspectives of the observed data. We will compare our model to [8] and [16] next.

6 EXPERIMENTS

We compare IBHP with three methods from the previous section: the vanilla Hawkes process (HP), the Dirichlet Hawkes (DHP; [8]), and the Hierarchical Dirichlet Hawkes (HDHP; [16]). We evaluate the models on both synthetic and real-world data.

6.1 SYNTHETIC DATASETS

The purpose of our synthetic-data experiments is twofold: 1) to understand the identifiability of our model and the accuracy of our SMC algorithm when the true data generation process satisfies the model assumptions, and 2) to understand the effects of misspecification.

Our setup is as follows. The Hawkes process base rate is $\lambda_0 = 2$. For the basis kernels, we use: $\gamma_1(\delta) = e^{-\delta/0.3}, \gamma_2(\delta) = 2e^{-\delta/0.2}, \gamma_3(\delta) = 3e^{-\delta/0.1}$. γ_1 has the smallest jump but also the largest time-scale; at the other extreme, γ_3 has the largest jump with a fast decay-parameter. γ_2 might be used to model 'regular' events, while γ_1 and γ_3 are for non-urgent and urgent ones respectively. We construct the dictionary $\mathcal{S}$ from the top 1000 words from the NIPS dataset [2], and the document lengths are set to $D = 20$. The hyperparameters, which are not to be estimated, are set as $\mathbf{w}_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), \mathbf{v}_0 = (\frac{1}{1000}, \ldots, \frac{1}{1000})$. We generate $N = 1000$ observations with this setup, and use the first 80% of the dataset for training, and the last 20% for testing. For each SMC iteration, we use 10 particles, and report averages and error bars based on 10 runs with different random seeds.

A. Parameter learning and prediction. Experiments A1 and A2 shown in Table 1 are the parameter estimates and the log-likelihoods over training and test datasets. Our model outperforms other models in terms of predictive log-likelihood. This demonstrates two points. Firstly, our SMC algorithm is able to accurately recover the underlying model parameters. Furthermore, estimating parameters for the misspecified models on this dataset is fair, since they have the same interpretation. Thus for instance, our results tell us that fitting a Hawkes model that does not include multiple triggering factors results in a significant overestimation of the base rate λ_0: a result that one might have expected.

A1. Parameter Estimation			
Parameter Values	λ_0 2	$\{\beta_l\}$ 1,2,3	$\{\tau_l\}$ 0.3,0.2,0.1
IBHP	**1.8**	**0.92, 1.63, 2.71**	**0.33, 0.18, 0.09**
HDHP	3.3	0.77, 4.56 6.11	3.75, 3.20, 2.94
DHP	2.9	0.83, 5.72, 5.83	1.21, 1.58, 1.28
HP	5.4	2.25, 4.38, 3.01	0.73, 2.54, 3.56

A2. Log-likelihoods		
	Training	Test
IBHP	**318.52**	**47.68**
HDHP	192.74	12.23
DHP	201.96	11.78
HP	81.68	6.18

B. Learn Latent State Variables ($K = 5, 10, 20$)		
	Jaccard($\mathbf{K}$)	1 - Hellinger($\mathbf{V}$)
IBHP	**0.83, 0.81, 0.77**	**0.79, 0.73, 0.68**
HDHP	0.56, 0.40, 0.35	0.51, 0.44, 0.29
DHP	0.61, 0.42, 0.38	0.64, 0.41, 0.36

Table 1: Model comparison over the synthetic datasets.

B. Learn latent state variables. Table 1 part B focuses on learning the latent state variables. Now, rather that generating data from our nonparametric model, we fix $K = 5, 10, 20$ in the data-generating process, and then compare these with our nonparametric esimates using two metrics: the Jaccard Index to compare the *binary* matrices $\mathbf{C}$ and the Hellinger distance for $\mathbf{V}$. A first complication is that these matrices need not have the same number of columns, and so for each comparison, we pad the smaller matrix with zero-columns to facilitate comparison. A bigger challenge is a 'label-switching' issue that arises since column permutations do not effect the quality of the estimates. To overcome this, after matching dimensions, we greedily match columns, and then compute scores. We point out that padding with zeros favors the alternative methods, since their solutions have many zeros; nevertheless, our model still gives the best Jaccard scores as well as Hellinger distances (we actually report *complementary* Hellinger distances (viz. one minus the actual distance), so that large numbers imply better performance for both statistics. As before, our results demonstrate the insufficiency of the alternate models and justifies the need for multiple factors.

C. The effects of base rate and basis kernels. The base rate λ_0, together with the evolving kernels, control the dynamics of latent factors. In Table 2 part C, we see that increasing λ_0 increases the average number of factors per observation increases—more strongly violating the *single* factor assumption of competing methods. This observation is also accompanied by a widening performance gap between our model and the alternatives.

C. Effects of Base Rate					
	λ_0	Topics	Jaccard	Hellinger	Test
IBHP	4	9.01	0.79	0.75	50.21
	8	12.28	0.72	0.69	68.37
	16	28.33	0.64	0.61	72.07
HDHP	4		0.32	0.40	43.78
	8	1	0.28	0.38	51.06
	16		0.31	0.26	50.79
DHP	4		0.29	0.37	41.67
	8	1	0.33	0.31	49.18
	16		0.27	0.28	52.33

Table 2: Effects of model specifications.

D. The effects of triggering rule. In Equation 11, the event rate depends on the rates of the underlying factors in an additive manner. We can allow more flexible triggering rules by allowing richer interactions among factor dynamics. For example, we define a "double-sharing" triggering rule as follows: *trigger a jump in the rate function only when two or more factors are shared with a previous observation.* Thus Equation 11 becomes:

$$\lambda(t_n) = \sum_{\kappa_{nk}\neq 0} \left[\sum_{i=1}^{n-1} \frac{\kappa_{ik}(t_n, t_i)}{\|\boldsymbol{\kappa}_i\|_0} + \phi \left(\frac{\kappa_{ik}(t_n, t_n)}{\|\boldsymbol{\kappa}_n\|_0} \right) \right] \quad (13)$$

where $\phi = 0$ if the rule is not triggered—there is no "jump", otherwise $\phi = \mathbf{w}_k^\intercal \boldsymbol{\beta} / \|\boldsymbol{\kappa}_n\|_0$—there is a "jump". We sketch this out in Figure 3.

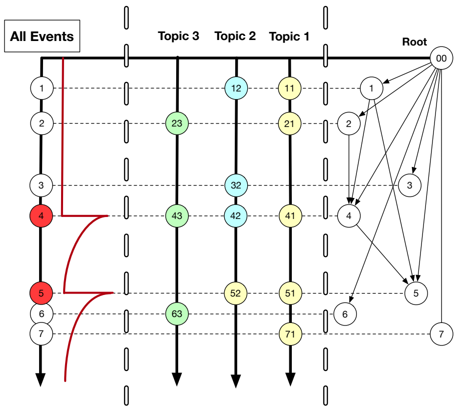


Figure 3: IBHP with "double-sharing" rule. Obs. 2 does not trigger a "jump" because no previous observations share more than two factors with it. However, obs. 4 triggers *two* jumps because it shares two factors with obs. 1 (factor 1 and 2), and two with obs. 2 (factor 1 and 3).

Incorporating such nonlinearities result in dynamics that are significantly different from the additive setup: this is evidenced in Table 3, where the simpler additive version the IBHP now has a degraded score. There are numerous variations to our simple "double-sharing" rule that are relevant across a variety of situations.

D. Predictive Log-likelihoods on Double-sharing Data		
	Additive Model	Double-sharing Model
IBHP	15.38 ± 3.82	20.82 ± 3.23
HDHP	8.97 ± 4.07	12.36 ± 3.18
DHP	8.26 ± 3.19	10.17 ± 3.20
HP	4.98 ± 3.61	5.04 ± 3.22

Table 3: Model comparison with "double-sharing" data.

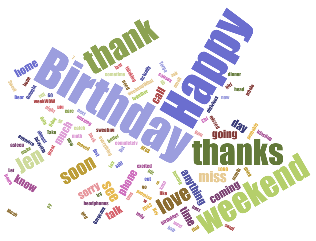
6.2 REAL DATASETS

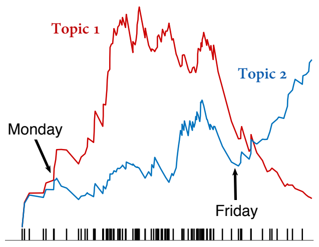
The purpose of our real data experiments is threefold: 1) to verify that the multiple triggering factors in IBHP are indeed relevant to real applications, 2) to demonstrate that our SMC inference algorithm is scalable for real-world datasets, and 3) to use our IBHP model to present meaningful findings, both quantitative and qualitative. We consider four different datasets: **Facebook Dataset.** This data contains Facebook message communications among 20,603 individuals. We pick the top 10 most connected individuals (based on the number of friends), and add in their one-hop and two-hop friends. This results in a total of 376 individuals. **NIPS Dataset [2].** The Kaggle NIPS dataset contains the title, authors, abstracts, and extracted text for all 7241 NIPS papers from the first 1987 conference to the current 2017 conference. This dataset is different in that it contains rich message information; however the number of time-points is just 30. **Santa Barbara Corpus Dataset [3].** This is a standard dataset used for applications involving Hawkes processes. We use conversation #33, a lively family discussion which centers around a disagreement that an individual, Jennifer, is having with her mother, Lisbeth. **Enron Email Dataset[1].** The Enron dataset contains about half a million email messages communicated among about 150 senior managers of the Enron corporation. We pick the longest thread of emails.

For each experiment, we use the first 80% of the dataset as training set, the next 10% as validation set, and the last 10% as test set. We train our model on training sets with different hyperparameters, then pick the best one based on their performances on the validation set, and use this model to report performances on the test set. The reported values are based on ten runs with different random number seeds. The dictionary $\mathcal{S}$ is all the unique words in the dataset; the document length D_n is counted from each observed text $\mathcal{T}_n$; and we use the three ($L = 3$) exponential basis kernels defined in Equation 5.

A. Predictive log-likelihood. The log-likelihoods in Table 4 show that for three of four datasets, our model outperforms the alternatives. The performance gaps exhibit a range of values. On the NIPS dataset, our model shows a massive improvement over the competition, while there is no significant improvement for the Enron dataset. The numbers in parentheses, giving the average number of

(a) Word cloud of factor 1.


(b) Word cloud of factor 2.



(c) Rate functions of factors 1/2.

Figure 4: **FB dataset.** Topic dynamics.

topics associated with each message, provides a partial explanation. For the Enron dataset, this number is just two, suggesting that there is limited benefit from modeling multiple factors, and that the simpler HDHP model may be more appropriate. For the NIPS dataset, this number is about 10, explaining the gap in performance.

FB Dataset (average # factors = 4.19)			
	Training	Validation	Test
IBHP	**1822** ± 96	**219** ± 10	**277** ± 11
HDHP	1083 ± 88	123 ± 10	133 ± 10
DHP	1058 ± 90	144 ± 9	200 ± 14
HP	782 ± 75	62 ± 7	69 ± 7
NIPS Dataset (average # factors = 10.21)			
	Training	Validation	Test
IBHP	**8378** ± 172	**913** ± 23	**1012** ± 28
HDHP	3229 ± 169	216 ± 12	191 ± 11
DHP	2018 ± 164	203 ± 10	202 ± 10
HP	390 ± 48	49 ± 8	40 ± 7
SB Dataset (average # factors = 6.52)			
IBHP	**520** ± 62	**187** ± 12	**137** ± 9
HDHP	132 ± 9	32 ± 6	34 ± 6
DHP	169 ± 10	51 ± 7	78 ± 9
HP	96 ± 10	15 ± 4	23 ± 4
Enron Dataset (average # factors = 2.17)			
IBHP	2602 ± 101	**313** ± 12	381 ± 12
HDHP	2322 ± 117	203 ± 10	**392** ± 11
DHP	**2639** ± 118	268 ± 11	339 ± 12
HP	729 ± 92	28 ± 5	19 ± 5

Table 4: Model comparisons over the real datasets.

B. Latent structure vs. dynamics. The rich structure of the NIPS dataset is balanced by its simple temporal structure just with 30 time points. This raises the question: how much of our models performance is due to the latent structure incorporated into our modeling framework, and how much is due to temporal dynamics of this structure. To study this more carefully, we shuffle the publication years (documents published in the same year remain together, however), thus destroying temporal information. Table 5 shows that this incurs a relatively small loss now, suggesting that most of the performance gains observed in Table 4 are due to the latent factors. However, removing temporal information still incurs enough of a hit in performance to justify our methodology.

Test Log-likelihoods on the NIPS Dataset			
	Original	Shuffled	Relative Loss
IBHP	**1012.08**	**914.76**	**-9.62%**
HDHP	191.29	88.19	-53.90%
DHP	201.73	79.05	-60.81%
HP	40.17	18.22	-54.64%

Table 5: Model comparison on the shuffled NIPS dataset.

C. Discovering popular topics and words. One of the immediate benefits of our IBHP is that it returns the factor rate matrix $\mathbf{K}$ and the word-distribution matrix $\mathbf{V}$, providing a rich summary of popular topics and words. Figure 5 shows, in the NIPS dataset, the most popular three topics at the end of the training dataset time span. The lists of words suggest that the first topic is related to kernel methods, the second to deep learning, and the third to Bayesian methods. The intensity of the colors indicates popularities. Our model suggests that topic 2, which hypothetically is related to deep learning, has been increasingly more popular in the NIPS community.

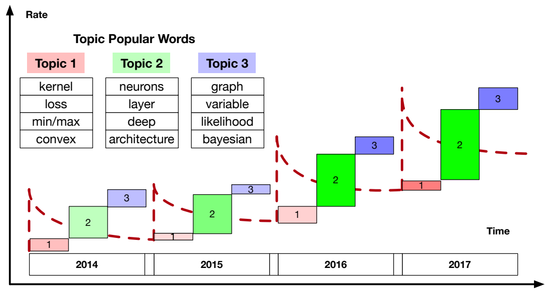


Figure 5: **NIPS dataset.** Popular topics and words.

D. Learning factor dynamics. Unlike the IBP, the IBHP matrix not only carries binary "present/missing" information, but also real-valued kernel weights κ_{ik}, which reveal the temporal dynamics of the factors. Figure 4 shows two most popular factors from the FB dataset. The first relates to school life, and the second to off-class activities. To confirm this, we plot the average of the estimated rate functions across four similar one week periods in Figure 4. The patterns of the two factor rates are quite different: The first factor is active after Monday, and peaks in the middle of the week, before cooling down near the weekend. The second factor, however, climbs steadily and becomes more excited during the weekend.

E. Infering dependencies and causalities. According to Equation 11, the rate after an event depends on earlier events that share factors with it. Figure 6 provides a detailed view of IBHP on the SB dataset under the usual additive rule. We also apply the "double-sharing" rule to the dataset. In Figure 7, we see several consequences: 1) the rate functions are not triggered until the 6^{th} observation under the double-sharing rule, 2) the IBHP matrices are different, and 3) the inferred factors are different. Further investigation shows the first red circle corresponds to the observation with text *"I am mean to you all the time!"* and the last red circle to *"What time is it?"*—one to heat up the process and one to cool it down. This suggests that adopting different triggering rules may allow us to capture different aspects of the data, which in our SB double-sharing case, bookends an active family discussion.

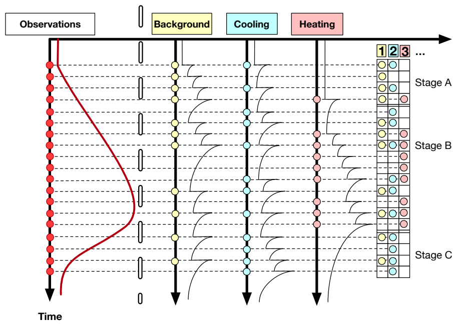


Figure 6: **SB Dataset.** Additive rule. Every observation creates a jump of the rate function. Topics can be interpreted as background, cooling, and heating activities.

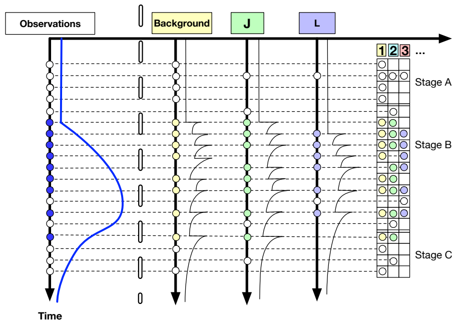


Figure 7: **SB Dataset with double-sharing.** White circles represent observations that do not trigger the rule. Topics can be interpreted as background activities, and those of Jennifer and Lisbeth.

F. Predict future event times. In Table 4, we report the log-likelihoods on the test datasets for each a model. To evaluate the predictive ability of our model in more depth, we use it for a different predictive task: predict the time of the next event in windows of increasing sizes, and for each case, report the absolute different from the observed data. Table 6 shows that, as the size of predictions increases, the mean absolute error increases, as well as the standard error: as the predictions becomes harder, the predictions becomes inaccurate and unreliable. Nonetheless, our model outperforms competing models according to this metric as well.

	Prediction Window Size		
	$pws = 1$	$pws = 5$	$pws = 10$
IBHP	0.61 $\pm$0.11	0.97 $\pm$0.18	1.37 $\pm$0.28
HDHP	0.82 $\pm$0.13	1.24 $\pm$0.20	2.18 $\pm$0.33
DHP	0.87 $\pm$0.10	1.19 $\pm$0.16	2.21 $\pm$0.29
HP	0.92 $\pm$0.17	2.06 $\pm$0.23	3.56 $\pm$0.31

Table 6: **FB Dataset.** Predicting future event times.

G. Predicting future topics and words. Our last experiment is concerned with the prediction of the latent state variables. The dotted line in Figure 8 represents the end of the training phase, where we have obtained the latent factor rate matrix $\mathbf{K}$ and the latent factor word distribution matrix $\mathbf{V}$. To the right of the dotted line, we show the projected rate function, along with the first three predictions and their predicted top words. Our model suggests that, for the NIPS dataset, topic 2 is taking over topic 3 and may become dominant in the next few events.

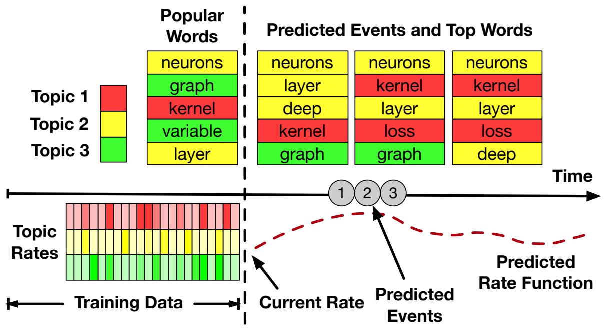


Figure 8: **NIPS dataset.** Predicted events.

7 CONCLUSION

In this paper, we proposed the Indian Buffet Hawkes Process (IBHP)—a novel Bayesian nonparametric stochastic point process model for learning multiple latent triggering factors of streaming document/message data. Our approach establishes the synergy between Indian Buffet Processes (IBP) and Hawkes processes (HP): on the one hand, we use the IBP to add multiple triggering factors to the HP, which helps to better model dynamics and improves interpretation, and on the other hand, the temporal information from the HP is embedded into the IBP to drive the latent factor estimation, which expands its capability to model evolution of factors.

Acknowledgment

This research is supported by NSF under contract numbers IIS-1546488 and IIS-1618690.

References

[1] Enron Email Data Set. `https://www.cs.cmu.edu/~./enron/`.

[2] NIPS Dataset. `http://www.kaggle.com/`.

[3] SB Dataset. `www.linguistics.ucsb.edu/research/santa-barbara-corpus`.

[4] Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2600–2608, 2012.

[5] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: Elementary Theory and Methods*. Springer Science & Business Media, 2003.

[6] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[7] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 1555–1564. ACM, 2016.

[8] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 219–228. ACM, 2015.

[9] Thomas L Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research (JMLR)*, 12(Apr):1185–1224, 2011.

[10] Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443, 1971.

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

[12] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503, 1974.

[13] Peng Lin, Ting Guo, Yang Wang, Fang Chen, et al. Infinite hidden semi-markov modulated interaction point process. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3900–3908, 2016.

[14] Peng Lin, Bang Zhang, Ting Guo, Yang Wang, and Fang Chen. Interaction point processes via infinite branching model. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1853–1859, 2016.

[15] Thomas Josef Liniger. *Multivariate hawkes processes*. PhD thesis, ETH Zurich, 2009.

[16] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez-Rodriguez. Modeling the dynamics of learning activity on the web. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1421–1430, 2017.

[17] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6757–6767, 2017.

[18] Xi Tan, Syed AZ Naqvi, Yuan (Alan) Qi, Katherine A Heller, and Vinayak Rao. Content-based modeling of reciprocal relationships using hawkes and gaussian processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[19] Xi Tan, Vinayak Rao, and Jennifer Neville. Nested crp with hawkes-gaussian processes. In *Artificial Intelligence and Statistics (AISTATS)*, 2018.

[20] Frank Wood and Thomas L Griffiths. Particle filtering for nonparametric bayesian matrix factorization. In *Advances in neural information processing systems*, pages 1513–1520, 2007.

[21] Jiasen Yang, Vinayak Rao, and Jennifer Neville. Decoupling homophily and reciprocity with latent space network models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[22] Lingjiong Zhu. *Nonlinear Hawkes Processes*. PhD thesis, New York University, 2013.