Dirichlet-Hawkes Processes with Applications to Clustering Continuous-Time Document Streams

Nan Du
Georgia Institute of Technology
Atlanta, GA, USA
dunan@gatech.edu

Mehrdad Farajtabar
Georgia Institute of Technology
Atlanta, GA, USA
mehrdad@gatech.edu

Amr Ahmed
Google Strategic Technologies
Mountain View, CA, USA
amra@google.com

Alexander J. Smola
Carnegie Mellon University
Pittsburgh, PA, USA
alex@smola.org

Le Song
Georgia Institute of Technology
Atlanta, GA, USA
lsong@cc.gatech.edu

ABSTRACT

Clusters in document streams, such as online news articles, can be induced by their textual contents, as well as by the temporal dynamics of their arriving patterns. Can we leverage both sources of information to obtain a better clustering of the documents, and distill information that is not possible to extract using contents only? In this paper, we propose a novel random process, referred to as the Dirichlet-Hawkes process, to take into account both information in a unified framework. A distinctive feature of the proposed model is that the preferential attachment of items to clusters according to cluster sizes, present in Dirichlet processes, is now driven according to the intensities of cluster-wise self-exciting temporal point processes, the Hawkes processes. This new model establishes a previously unexplored connection between Bayesian Nonparametrics and temporal Point Processes, which makes the number of clusters grow to accommodate the increasing complexity of online streaming contents, while at the same time adapts to the ever changing dynamics of the respective continuous arrival time. We conducted large-scale experiments on both synthetic and real world news articles, and show that Dirichlet-Hawkes processes can recover both meaningful topics and temporal dynamics, which leads to better predictive performance in terms of content perplexity and arrival time of future documents.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

DOI: http://dx.doi.org/10.1145/2783258.2783411.

Keywords

Dirichlet Process, Hawkes Process, Document Modeling

1. INTRODUCTION

Online news articles, blogs and tweets tend to form clusters around real life events and stories on certain topics [2, 3, 9, 22, 7, 24]. Such data are generated by myriads of online media sites in real-time and in large volumes. It is a critically important task to effectively organize these articles according to their contents such that online users can quickly sift and digest them.

Besides textual information, temporal information also provides very good clues on the clustering of online document streams. For instance, weather reports, forecasts and warnings of the blizzard in New York city this year appeared online even before the snowstorm actually started[1]. As the blizzard conditions gradually intensified, more subsequent blogs, posts and tweets were triggered around this event in various online social media. Such self-excitation phenomenon often leads to many closely related articles within a short period of time. Later, after the influence of the event past its peak, *e.g.*, the blizzard eventually stopped, public attention gradually turned to other events, and the following articles on the blizzard faded out eventually.

Furthermore, depending on the nature of real life events, relevant news articles can exhibit very different temporal dynamics. For instance, articles on emergency or incidents may rise and fall quickly, while some other stories, gossips and rumors may have a far reaching influence, *e.g.*, related posts about a Hollywood blockbuster can continue to appear as more details and trailers are revealed. As a consequence, the clustering of document streams can be improved by taking into account the underlying heterogeneous temporal dynamics. Distinctive temporal dynamics will also help us to disambiguate different clusters of similar topics emerging closely in time, to track their popularity and to predict the future trends.

Such problem of modeling time-dependent topic-clusters has been attempted by [2, 3], where the Recurrent Chinese Restaurant Process(RCRP) [4] has been proposed to model

[1] http://www.usatoday.com/story/weather/2015/01/25/northeast-possibly-historic-blizzard/22310869/

each topic-cluster of a news stream. However, one of the main deficiencies of the RCRP and related models [9] is that they require an explicit division of the event stream into unit episodes. Although this was ameliorated in the DD-CRP model [6] simply by defining a continuous weighting function, it does not address the issue that the actual *counts* of events are nonuniform over time. Artificially discretizing the time line into bins introduces additional tuning parameters, which are not easy to choose optimally. Therefore, in this work, we propose a novel random process, referred to as the Dirichlet-Hawkes process (DHP), to take into account both sources of information to cluster *continuous-time* document streams. More precisely, we make the following contributions :

- We establish a previously unexplored connection between Bayesian Nonparametrics and Temporal Point Processes, which allows the number of clusters to grow in order to accommodate the increasing complexity of online streaming contents, while at the same time learns the ever changing latent dynamics governing the respective continuous arrival patterns inherently.
- We point out that our combination of Dirichlet processes and Hawkes processes has implications beyond clustering document streams. We will show that our construction can be generalized to other Nonparametric Bayesian models, such as the Pitman-Yor processes [23] and the Indian Buffet processes [16].
- We propose an efficient online inference algorithm which can scale up to millions of news articles with near constant processing time per document and moderate memory consumptions.
- We conduct large-scale experiments on both synthetic and real-world datasets to show that Dirichlet-Hawkes processes can recover meaningful topics and temporal dynamics, leading to better predictive performance in terms of both content perplexity and document arriving time.

2. PRELIMINARIES

We first provide a brief introduction to the two major building blocks for the Dirichlet-Hawkes processes: Bayesian nonparametrics and temporal point processes. Bayesian nonparametrics, especially Chinese Restaurant Processes, are a rich family of models which allow the model complexity (*e.g.*, number of latent clusters, number of latent factors) to grow as more data are observed [18]. Temporal point processes, especially Hawkes Processes [17], are the mathematical tools for modeling recurrent patterns and continuous-time nature of real world events.

2.1 Bayesian Nonparametrics

The Dirichlet process (DP) [5] is one of the most basic Bayesian nonparametric processes, parameterized by a concentration parameter $\alpha > 0$ and a base distribution $G_0(\theta)$ over a given space $\theta \in \Theta$. A sample $G \sim DP(\alpha, G_0)$ drawn from a DP is a discrete distribution by itself, even the base distribution is continuous. Furthermore, the expected value of G is the base distribution, and the concentration parameter controls the level of discretization in G: in the limit of $\alpha \to 0$, a sampled G is concentrated on a single value, while in the limit of $\alpha \to \infty$, a sampled G becomes continuous. In between are the discrete distributions with less concentration as α increases.

Since G itself is a distribution, we can draw samples $\theta_{1:n}$ from it, and use these samples as the parameters for models of clusters. Equivalently, let $\theta_{1:n}$ denote the collection of $\{\theta_1, \ldots, \theta_{n-1}\}$, and $\{\theta_k\}$ be the set of distinct values in $\theta_{1:n}$. Instead of first drawing G and then sampling $\theta_{1:n}$, this two-stage process can be simulated as follows :

1. Draw θ_1 from G_0.
2. For $n > 1$:
   (a) With probability $\frac{\alpha}{\alpha+n-1}$ draw θ_n from G_0.
   (b) With probability $\frac{m_k}{\alpha+n-1}$ reuse θ_k for θ_n, where m_k is the number of previous samples with value θ_k.

This simulation process is also called Chinese Restaurant Process(CRP), which captures the "rich get richer" or preferential attachment phenomenon. Essentially, in this CRP metaphor, a Chinese restaurant has an infinite number of tables (each corresponding to a cluster). The nth customer θ_n can either choose a table with m_k existing customers with probability $\frac{m_k}{n-1+\alpha}$, or start a new table with probability $\frac{\alpha}{n-1+\alpha}$. Formally, the conditional distribution of the θ_n can be written as a mixture:

$$\theta_n \,|\, \theta_{1:n-1} \sim \sum_k \frac{m_k}{n-1+\alpha}\delta(\theta_k) + \frac{\alpha}{n-1+\alpha}G_0(\theta). \quad (1)$$

In other words, it is more likely to sample from larger clusters, and the probability is proportional to the size of that cluster. Since the model allows new clusters to be created with a small probability, the model has the potential to generate infinite number of clusters adapted to the increasing complexity of the data. Thus the Dirichlet process is often used as a prior for the parameters of clustering models.

The recurrent Chinese restaurant process (RCRP) is an extension of the DP which takes into account the temporal coherence of clusters for documents divided into episodes [4].

One can think of RCRP as a discrete-time sequence of DPs, one for the data in each episode. The clusters in these DPs are shared, and the DPs appearing later in time can have a small probability to create new clusters. More specifically, the conditional distribution of the nth value, $\theta_{t,n}$, sampled in episode t can be written as a mixture

$$\theta_{t,n} \,|\, \theta_{1:t-1,:}, \theta_{t,1:n-1} \sim \sum_k \frac{m_{k,t} + m'_{k,t}}{\sum_j (m_{j,t} + m'_{j,t}) + \alpha}\delta(\theta_k) + \frac{\alpha}{\sum_j (m_{j,t} + m'_{j,t}) + \alpha}G_0, \quad (2)$$

where $\theta_{1:t-1,:}$ is the set of all samples drawn in previous episodes from 1 to $t-1$, and $\theta_{t,1:n-1}$ is the set of samples drawn in the current episode t before $\theta_{t,n}$. The statistic $m_{k,t}$ is the number of previous samples in episode t with value θ_k, and $m'_{k,t}$ captures related information in $\theta_{1:t-1,:}$ about the value θ_k. The latter quantity, $m'_{k,t}$, can be modeled in many ways. For instance, the original model in [4] applies a Markov Chain to model $m'_{k,t}$, and later follow-ups [2, 3] use a weighted combination of counts from recent Δ episodes

$$m'_{k,t} = \sum_{j=1}^{\Delta} e^{-\frac{j}{\beta}} m_{k,t-j}, \quad (3)$$

with an exponential kernel parametrized by the decaying factor β. Essentially, it models the decaying influence of counts from previous episodes across time. RCRP can also be used as a prior for the parameters of clustering models. For instance, Figure 2(a) shows a combination of RCRP and a bag-of-words model for each cluster.

However, RCRP requires artificially discretizing the time line into episodes, which is unnatural for continuous-time online document streams. Second, different type of clusters is likely to occupy very different time scales, and it is not clear how to choose the time window for each episode a-priori. Third, the temporal dependence of clusters across episodes is hard-coded in Equation (3), and it is the same for different clusters. Such design cannot capture the distinctive temporal dynamics of different type of clusters, such as related articles about disasters vs. Hollywood blockbusters, and fail to learn such dynamics from real data. We will use the Hawkes process introduced next to address these drawbacks of RCRP when handling temporal dynamics.

2.2 Hawkes Process

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. Many different types of data produced in online social networks can be represented as temporal point processes, such as the event time of retweets and link creations. A temporal point process can be equivalently represented as a counting process, $N(t)$, which records the number of events before time t. Let the history $\mathcal{T}$ be the list of event time $\{t_1, t_2, \ldots, t_n\}$ up to but not including time t. Then in a small time window dt between $[0, t)$, the number of observed event is

$$dN(t) = \sum_{t_i \in \mathcal{T}} \delta(t - t_i)\, dt, \tag{4}$$

and hence $N(t) = \int_0^t dN(s)$, where $\delta(t)$ is a Dirac delta function. It is often assumed that only one event can happen in a small window of size dt, and hence $dN(t) \in \{0, 1\}$.

An important way to characterize temporal point processes is via the conditional intensity function — the stochastic model for the next event time given all previous events. Within a small window $[t, t + dt)$, $\lambda(t)dt$ is the probability for the occurrence of a new event given the history $\mathcal{T}$:

$$\lambda(t)dt = \mathbb{P}\left\{\text{event in } [t, t + dt) | \mathcal{T}\right\}. \tag{5}$$

The functional form of the intensity $\lambda(t)$ is often designed to capture the phenomena of interests [1]. For instance, in a **homogeneous Poisson process**, the intensity is assumed to be independent of the history $\mathcal{T}$ and constant over time, *i.e.*, $\lambda(t) = \lambda_0 \geqslant 0$. In an **inhomogeneous Poisson process**, the intensity is also assumed to be independent of the history $\mathcal{T}$ but it can be a function varying over time, *i.e.*, $\lambda(t) = g(t) \geqslant 0$. In both case, we will use notation Poisson$(\lambda(t))$ to denote a Poisson process.

A Hawkes process captures the mutual excitation phenomena between events, and its intensity is defined as

$$\lambda(t) = \gamma_0 + \alpha \sum_{t_i \in \mathcal{T}} \gamma(t, t_i), \tag{6}$$

where $\gamma(t, t_i) \geqslant 0$ is the triggering kernel capturing temporal dependencies , $\gamma_0 \geqslant 0$ is a baseline intensity independent of the history and the summation of kernel terms is history dependent and a stochastic process by itself. The kernel function can be chosen in advance, *e.g.*, $\gamma(t, t_i) = \exp(-|t - t_i|)$ or $\gamma(t, t_i) = \mathbb{I}[t > t_i]$, or directly learned from data.

A distinctive feature of a Hawkes process is that the occurrence of each historical event increases the intensity by a certain amount. Since the intensity function depends on the history $\mathcal{T}$ up to time t, the Hawkes process is essentially a conditional Poisson process (or doubly stochastic Poisson process [19]) in the sense that conditioned on the history $\mathcal{T}$, the Hawkes process is a Poisson process formed by the superposition of a background homogeneous Poisson process with the intensity γ_0 and a set of inhomogeneous Poisson processes with the intensity $\gamma(t, t_i)$. However, because the events in a past interval can affect the occurrence of the events in later intervals, the Hawkes process in general is more expressive than a Poisson process. Hawkes process is particularly good for modeling repeated activities, such as social interactions [14], search behaviors [21], or infectious diseases that do not convey immunity.

Given a time $t' \geqslant t$, we can also characterize the conditional probability that no event happens during $[t, t')$ and the conditional density that an event occurs at time t', using the intensity $\lambda(t)$ [1], as

$$S(t'|\mathcal{T}) = \exp\left(-\int_t^{t'} \lambda(\tau)\, d\tau\right), \; f(t'|\mathcal{T}) = \lambda(t')\, S(t'|\mathcal{T}). \tag{7}$$

With these two quantities, we can express the likelihood of a list of event times $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ in an observation window $[0, T)$ with $T \geqslant t_n$ as

$$\mathcal{L} = \prod_{t_i \in \mathcal{T}} f(t_i|\mathcal{T}) = \prod_{t_i \in \mathcal{T}} \lambda(t_i) \cdot \exp\left(-\int_0^T \lambda(\tau)\, d\tau\right) \tag{8}$$

which will be useful for learning the parameters of our model from observed data. With the above backgrounds, now we proceed to describe our Dirichlet-Hawkes process.

3. DIRICHLET-HAWKES PROCESS

The key idea of Dirichlet-Hawkes process (DHP) is to have the Hawkes Process model the rate *intensity* of events (*e.g.*, the arrivals of documents), while the Dirichlet Process captures the *diversity* of event types (*e.g.*, clusters of documents). More specifically, DHP is parametrized by an intensity parameter $\lambda_0 > 0$, a base distribution $G_0(\theta)$ over a given space $\theta \in \Theta$ and a collection of triggering kernel functions $\{\gamma_\theta(t, t')\}$ associated with each event type of parameter θ. Then, we can generate a sequence of samples $\{(t_i, \theta_i)\}$ as follows :

1. Draw t_1 from Poisson(λ_0) and θ_1 from $G_0(\theta)$.
2. For $n > 1$:
   (a) Draw $t_n > t_{n-1}$ from Poisson$\big(\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t, t_i)\big)$.
   (b) Draw θ_n from $G_0(\theta)$ with probability :
   $$\frac{\lambda_0}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)}.$$
   (c) Reuse θ_k for θ_n with probability :
   $$\frac{\lambda_{\theta_k}(t_n)}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)},$$
   where $\lambda_{\theta_k}(t_n) := \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)\mathbb{I}[\theta_i = \theta_k]$ is the intensity of a Hawkes process for previous events with value θ_k.

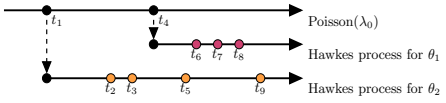


Figure 1: A sample from Dirichlet-Hawkes process. A background Poisson process with intensity λ_0 sampled the starting time points t_1 and t_4 for two different event types with the respective parameter θ_1 and θ_2. These two initial events then generate a Hawkes process of their own, with events at time $\{t_2, t_3, t_5, t_9\}$ and $\{t_6, t_7, t_8\}$, respectively.

Figure 1 gives an intuitive illustration of the Dirichlet-Hawkes Process. Compared to the Dirichlet process, the intensity parameter λ_0 here serves the similar role to the concentration parameter α in the Dirichlet process. Instead of counting the number, m_k, of samples within a cluster, the Dirichlet-Hawkes process uses the intensity function $\lambda_{\theta_k}(t)$ of a Hawkes process which can be considered as a temporally weighted count. For instance, if $\gamma_\theta(t, t_i) = \mathbb{I}[t > t_i]$, then $\lambda_{\theta_k}(t) = \sum_{i=1}^{n-1} \mathbb{I}[t > t_i]\mathbb{I}[\theta_i = \theta_k]$ is equal to m_k. If $\gamma_\theta(t, t_i) = \exp(-|t - t_i|)$, then $\lambda_{\theta_k}(t) = \sum_{i=1}^{n-1} \exp(-|t - t_i|)\mathbb{I}[\theta_i = \theta_k]$, and each previous event in the same cluster contributes a temporally decaying increment. Other triggering kernels associated with θ_i can also be used or learned from data. Thus the Dirichlet-Hawkes process is more general than the Dirichlet process and can generate both preferential-attachment type of clustering and rich temporal dynamics.

From the view of a temporal point process, the generation of the event timing in Dirichlet-Hawkes process can also be viewed as the superposition of a Poisson process λ_0 and several Hawkes processes (conditional Poisson processes), one for each distinctive value of θ_d and with intensity $\lambda_{\theta_d}(t)$. Thus the overall event intensity is the sum of the intensities from individual processes [20]

$$\bar{\lambda}(t) = \lambda_0 + \sum_{d=1}^{D} \lambda_{\theta_d}(t),$$

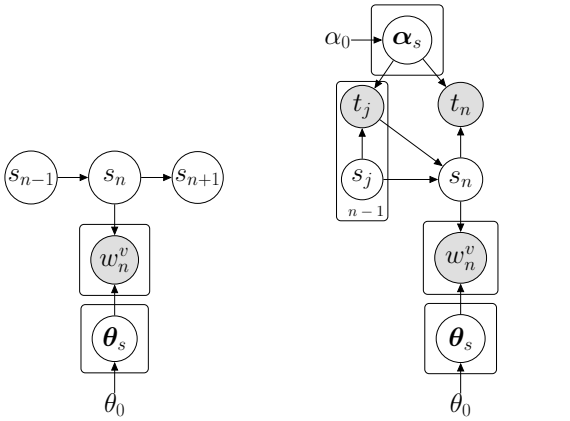
where D is the total number of distinctive values $\{\theta_i\}$ in the DHP up to time t.

Therefore, the Dirichlet-Hawkes process can capture the following four desirable properties :

1. Preferential attachment: Draw θ_n according to $\lambda_{\theta_k}(t_n)$. The larger the intensity for a Hawkes process, the more likely the next event is from that cluster.
2. Adaptive number of clusters: Draw θ_n according to λ_0. There is always some probability of generating new cluster with λ_0.
3. Self-excitation: This is captured by the intensity of the Hawkes process $\lambda_{\theta_k}(t) = \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)\mathbb{I}[\theta_i = \theta_k]$.
4. Temporal decays: This is captured by the triggering kernel function $\gamma_\theta(t, t_i)$ which is typically decaying over time.

Finally, given the sequence of events (or samples) $\mathcal{T} = \{(t_i, \theta_i)\}_{i=1}^n$ from a Dirichlet-Hawkes process, the likelihood of the event arrival times can be evaluated based on (8) as

$$\mathcal{L}(\mathcal{T}) = \exp\left(-\int_0^T \bar{\lambda}(\tau)d\tau\right) \prod_{(t_i,\theta_i)\in\mathcal{T}} \lambda_{\theta_i}(t_i), \qquad (9)$$



(a) RCRP (b) Dirichlet-Hawkes Process

Figure 2: Generative models for different processes.

Compared to the recurrent Chinese restaurant process (RCRP) appeared in [4], a distinctive feature of the Dirichlet-Hawkes process is that there is no need to discretize the time and divide events into episodes. Furthermore, the temporal dynamics is controlled by more general triggering kernel functions, and can be statistically learned from data.

4. GENERATING TEXT WITH DHP

As we have defined the Dirichlet-Hawkes process (DHP), we will use it as a prior for modeling continuous-time document streams. The goal is to discover clusters from the document stream based on both contents and temporal dynamics. Essentially, the set of $\{\theta_i\}$ sampled from the DHP will be used as the parameters for document content model, and each cluster will have a distinctive value of $\theta_d \in \{\theta_i\}$. Furthermore, we will allow different clusters to have different temporal dynamics, with the corresponding triggering kernel drawn from a mixture of K base kernels. We will first present the overall generative process of the model before going into details of these components in Figure 2(b).

1. Draw t_1 from Poisson(λ_0), θ_1 from Dir($\theta|\theta_0$), and α_{θ_1} from Dir($\alpha|\alpha_0$).
2. For each word v in document 1: $w_1^v \sim$ Multi(θ_1)
3. For $n > 1$:
   (a) Draw $t_n > t_{n-1}$ from Poisson$\big(\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)\big)$,
   $$\text{where } \gamma_{\theta_i}(t_n, t_i) = \sum_{l=1}^{K} \alpha_{\theta_i}^l \cdot \kappa(\tau_l, t_n - t_i)$$
   (b) Draw θ_n from Dir($\theta|\theta_0$) with probability
   $$\frac{\lambda_0}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)}, \qquad (10)$$
   and draw α_{θ_n} from Dir($\alpha|\alpha_0$)
   (c) Reuse previous θ_k for θ_n with probability
   $$\frac{\lambda_{\theta_k}(t_n)}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)}, \qquad (11)$$
   where $\lambda_{\theta_k}(t_n) = \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)\mathbb{I}[\theta_i = \theta_k]$.
   (d) For each word v in document n:
   $$w_n^v \sim \text{Multi}(\theta_n)$$

Content model. Many document content models can be used here. For simplicity of exposition, we have used a simple bag-of-word language model for each cluster in the above generative process. In this case, the base distribution $G(\theta)$ in the DHP is now chosen as a Dirichlet distribution, $\text{Dir}(\theta|\theta_0)$, with parameter θ_0. Then, w_n^v, the vth word in the nth document is sampled according to a multinomial distribution

$$w_n^v \sim \text{Multi}(\theta_{s_n}). \tag{12}$$

where s_n is the cluster indicator variable for the nth document, and the parameter θ_{s_n} is a sample drawn from the DHP process.

Triggering kernel. We allow different clusters to have different temporal dynamics, by representing the triggering kernel function of the Hawkes Process as a non-negative combination of K base kernel functions,*i.e.*,

$$\gamma_\theta(t_i, t_j) = \sum_{l=1}^{K} \alpha_\theta^l \cdot \kappa(\tau_l, t_i - t_j), \tag{13}$$

where $t_j < t_i, \sum_l \alpha_\theta^l = 1, \alpha_\theta^l > 0$, and τ_i is the typical reference time points, *e.g.*, 0.5, 1, 8, 12, 24 hours *etc.*

To simplify notations, define $\Delta_{ij} = t_i - t_j$, $\boldsymbol{\alpha}_\theta = (\alpha_\theta^1, \dots, \alpha_\theta^K)^\top$ and $\boldsymbol{k}(\Delta_{ij}) = (\kappa(\tau_1, \Delta_{ij}), \dots, \kappa(\tau_K, \Delta_{ij}))^\top$, so $\gamma_\theta(t_i, t_j) = \boldsymbol{\alpha}_\theta^\top \boldsymbol{k}(\Delta_{ij})$. Since each cluster has its own set of kernel parameters $\boldsymbol{\alpha}_\theta$, we are able to track their different evolving processes. Given $\mathcal{T} = \{(t_n, \theta_n)\}_{n=1}^N$, the intensity function of the cluster with parameter θ is represented as

$$\lambda_\theta(t) = \sum_{t_i < t} \boldsymbol{\alpha}_\theta^\top \boldsymbol{k}(t - t_i) \mathbb{I}[\theta_i = \theta], \tag{14}$$

and the likelihood $\mathcal{L}(\mathcal{T})$ of observing the sequence $\mathcal{T}$ before time T based on Equation (9) is

$$\exp\left(-\sum_{\theta_i=\theta} \boldsymbol{\alpha}_\theta^\top \boldsymbol{g}_\theta - \Lambda_0\right) \prod_{\theta_i=\theta} \sum_{t_j<t_i, \theta_j=\theta} \boldsymbol{\alpha}_\theta^\top \boldsymbol{k}(\Delta_{ij}), \tag{15}$$

where $g_\theta^l = \sum_{t_i<T, \theta_i=\theta} \int_{t_i}^T \kappa(\tau_l, t - t_i) dt$ and $\Lambda_0 = \int_0^T \lambda_0 dt$. This can be done efficiently for many kernels, such as the Gaussian RBF kernel [12, 13], Rayleigh kernel [1], *etc.* Here, we choose the Gaussian RBF kernel $\kappa(\tau_l, \Delta) = \exp(-(\Delta - \tau_l)^2)/2\sigma_l^2)/\sqrt{2\pi\sigma_l^2}$, so the integral g_θ^l has the analytic form:

$$\sum_{t_i<T, \theta_i=\theta} \frac{1}{2}\left(\text{erfc}\left(-\frac{\tau_l}{\sqrt{2\sigma_l^2}}\right) - \text{erfc}\left(\frac{T - t_i - \tau_l}{\sqrt{2\sigma_l^2}}\right)\right) \tag{16}$$

Inexact event timing. In practice, news articles are usually automatically collected and indexed by web crawlers. Sometimes, due to unexpected errors or the available minimum timing resolution, we can observe a few m documents at the same time t_n. In this case, we assume that each of the m documents actually arrived between t_{n-1} and t_n. To model this rare situation, we can randomly pick t_n and replace the exact timestamps within the interval $[t_n, t_{n+m-1}]$ by t_{n+m-1} to take that into account.

5. INFERENCE

Given a stream of documents $\{(d_i, t_i)\}_{i=1}^n$, at a high level, the inference algorithm alternates between two subroutines. The first subroutine samples the latent cluster membership (and perhaps the missing time) for the current document d_n by Sequential Monte Carlo [10, 11]; and then, the second subroutine updates the learned triggering kernels of the respective cluster on the fly.

Sampling the cluster label. Let $s_{1:n}$ and $t_{1:n}$ be the latent cluster indicator variables and document time for all the documents $d_{1:n}$. For each s_n, we have $s_n \in \{0, 1, \dots, D\}$, where D is the total number of distinctive values $\{\theta_i\}$, and $s_n = 0$ refers to the background Poisson process Poisson(λ_0). In the streaming context, it is shown by [2, 3] that it would be more suitable to efficiently draw a sample for the latent cluster labels $s_{1:n}$ shown in Figure 2(b) from $P(s_{1:n}|d_{1:n}, t_{1:n})$ by reusing the past samples from $P(s_{1:n-1}|d_{1:n-1}, t_{1:n})$, which motivates us to apply the Sequential Monte Carlo method [10, 11, 2, 3]. Briefly, a particle keeps track of an approximation of the posterior $P(s_{1:n-1}|d_{1:n-1}, t_{1:n-1})$, where $d_{1:n-1}$, $t_{1:n-1}$, $s_{1:n-1}$ represent all past documents, timestamps and cluster labels, and updates it to get an approximation for $P(s_{1:n}|d_{1:n}, t_{1:n})$. We maintain a set of particles at the same time, each of which represents a hypothesis about the latent random variables and has a weight to indicate how well its hypothesis can explain the data. The weight w_n^f of each particle $f \in \{1, \dots, F\}$ is defined as the ratio between the true posterior and a proposal distribution $w_n^f = \frac{P(s_{1:n}|d_{1:n}, t_{1:n})}{\pi(s_{1:n}|d_{1:n}, t_{1:n})}$. To minimize the variance of the resulting particle weight, we take $\pi(s_n|s_{1:n-1}, d_{1:n}, t_{1:n})$ to be the posterior distribution $P(s_n|s_{1:n-1}, d_{1:n}, t_{1:n})$ [11, 2]. Then, the unnormalized weight w_n^f can be updated by

$$w_n^f \propto w_{n-1}^f \cdot P(d_n|s_n^f, d_{1:n-1}). \tag{17}$$

Because the posterior is decomposed as $P(s_n|d_n, t_n, \text{rest}) \sim P(d_n|s_n, \text{rest}) \cdot P(s_n|t_n, \text{rest})$, by the Dirichlet-Multinomial conjugate relation, the likelihood $P(d_n|s_n, \text{rest})$ is given by

$$\frac{\Gamma\left(C^{s_n \backslash d_n} + V\theta_0\right) \prod_v^V \Gamma\left(C_v^{s_n \backslash d_n} + C_v^{d_n} + \theta_0\right)}{\Gamma\left(C^{s_n \backslash d_n} + C^{d_n} + V\theta_0\right) \prod_k^K \Gamma\left(C_v^{s_n \backslash d_n} + \theta_0\right)}, \tag{18}$$

where $C^{s_n \backslash d_n}$ is the word count of cluster s_n excluding the document d_n, C^{d_n} is the word count of document d_n, $C_v^{s_n \backslash d_n}$ and $C_v^{d_n}$ refer to the count of the vth word, and V is the vocabulary size. Finally, $P(s_n|t_n, \text{rest})$ is the prior given by the Dirichlet-Hawkes process (10) and (11) as

$$P(s_n = k|t_n, \text{rest}) = \begin{cases} \frac{\lambda_{\theta_k}(t_n)}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)} & \text{if } k \text{ occupied} \\ \frac{\lambda_0}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i}(t_n, t_i)} & \text{otherwise} \end{cases} \tag{19}$$

Updating the triggering kernel. Given s_n, we denote the respective triggering kernel $\boldsymbol{\alpha}_{\theta_{s_n}}$ by $\boldsymbol{\alpha}_{s_n}$ for brevity. By the Bayesian rule, the posterior is given by $P(\boldsymbol{\alpha}_{s_n}|\mathcal{T}_{s_n}) \sim P(\mathcal{T}_{s_n}|\boldsymbol{\alpha}_{s_n}) P(\boldsymbol{\alpha}_{s_n}|\alpha_0)$, where $\mathcal{T}_{s_n} = \{(t_i, s_i)|s_i = s_n\}$ is the set of events in cluster s_n. We can either update the estimation of $\boldsymbol{\alpha}_{s_n}$ by MAP for that the log-likelihood of (15) is concave in $\boldsymbol{\alpha}_{s_n}$. Alternatively, we can draw a set of samples $\{\boldsymbol{\alpha}_{s_n}^i\}_{i=1}^N$ from the prior $P(\boldsymbol{\alpha}_{s_n}|\alpha_0)$ and calculate the weighted average:

$$\hat{\boldsymbol{\alpha}}_{s_n} = \sum_{i=1}^{N} w_i \cdot \boldsymbol{\alpha}_{s_n}^i, \tag{20}$$

where $w_i = P(\mathcal{T}_{s_n}|\boldsymbol{\alpha}_{s_n}^i) P(\boldsymbol{\alpha}_{s_n}^i|\alpha_0) / \sum_i P(\mathcal{T}_{s_n}|\boldsymbol{\alpha}_{s_n}^i) P(\boldsymbol{\alpha}_{s_n}^i|\alpha_0)$. For simplicity, we choose the latter method in our implementation.

Sampling the missing time. In the rare case when m documents arrive with the same timestamp t_n, the precise document time is missing during the interval $[t_{n-1}, t_n]$. As a result, we need joint samples for $\{(s_n^i, t_n^i)\}_{i=1}^m$ where s_n^i and t_n^i are the cluster membership and the precise arriving time for the ith document d_n^i. However, since m is expected to be small in practice, we can use Gibbs sampling to draw samples from the distribution $P(t_n^{1:m}, s_n^{1:m} | t_{1:n-1}, s_{1:n-1}, \text{rest})$ where $s_n^{1:m}$ and $t_n^{1:m}$ are the cluster labels and document time for the m documents in the current nth interval. The initial values for $t_n^{1:m}$ can be assigned uniformly from the interval $[t_{n-1}, t_n]$. After fixing the sampled $s_n^{1:m}$ and the other $\{t_n^k\}_{k \neq i}$, we are going to draw a new sample $t_n^{i\prime}$ from $P(t_n^i | s_n^i, \text{rest})$. Let $\mathcal{T}_{s_n^i} \backslash t_n^i$ be the set of all the document time excluding t_n^i in cluster s_n^i. The posterior of t_n^i is proportional to the joint likelihood $P(t_n^i | s_n^i, \mathcal{T}_{s_n^i} \backslash t_n^i, \text{rest}) \propto P(t_n^i, \mathcal{T}_{s_n^i} \backslash t_n^i | s_n^i, \text{rest})$. Therefore, we can apply Metropolis algorithm in one dimension to draw the next sample $t_n^{i\prime}$. Specifically, let's first uniformly draw $t_n^{i\prime}$ from $[t_{n-1}, t_n]$ and calculate the following ratio between the two joint likelihoods $r = \frac{P(t_n^{i\prime}, \mathcal{T}_{s_n^i} \backslash t_n^i | s_n^i, \text{rest})}{P(t_n^i, \mathcal{T}_{s_n^i} \backslash t_n^i | s_n^i, \text{rest})}$. We then accept $t_n^{i\prime}$ if $r > 1$; otherwise, we accept it with the probability r. With the new sample $t_n^{i\prime}$, we can update the kernel parameter by (20). Finally, we need to update the particle weight by considering the likelihood of generating such m documents as

$$w_n^f \propto w_{n-1}^f \times \prod_{i=1}^m P(d_n^i | s_n^{f,i}, d_{1:n-1}, d_n^{1:m\backslash i}, \text{rest}) \times \prod_{s \in \{s_n^i\}_{i=1}^m} P(m_s | \mathcal{T}_s, \text{rest}), \quad (21)$$

where $d_n^{1:m\backslash i}$ is the set of m documents excluding d_n^i, m_s is the number of documents with cluster membership s among the m documents, and $\mathcal{T}_s$ is the set of document time in cluster s. Conditioned on the history up to t_n, the Hawkes process is an inhomogeneous Poisson process, and thus we know that $P(m_s | \mathcal{T}_s, \text{rest})$ is simply a Poisson distribution with mean $\Lambda_s = \int_{t_{n-1}}^{t_n} \lambda_s(t) dt$. For Gaussian kernels, we can use (16) to obtain the analytic form of Λ_s. The overall pseudocode for Sequential Monte Carlo is formally presented in Algorithm 1, and the Gibbs sampling framework is given by Algorithm 2.

Efficient Implementation. In order to scale with large datasets, the online inference algorithm should be able to process each individual document in an expected constant time. Particularly, the expected time cost of sampling the cluster label and updating the triggering kernel should not grow with the amount of documents we have seen so far. The most fundamental operation in Algorithm 1 and 2 is to evaluate the joint likelihood (15) of all the past document time to update the triggering kernel in every cluster. A straightforward implementation requires repeated computation of a sum of Gaussian kernels over the whole history, which tends to be quadratic to the number of past documents.

Based on the fast decaying property that the Gaussian kernel decreases exponentially as the distance deviating from its center increases quadratically, we can alleviate the problem by ignoring those past time far away from the kernel center. Specifically, given an error tolerance ϵ, we only need

Algorithm 1: The SMC Framework

```
Initialize w_1^f to 1/F for all f ∈ {1...F};
for each event time t_n, n = 1, 2, ... do
    for f ∈ {1, ..., F} do
        if one document d_n at the time t_n then
            sample s_n from (19) and add t_n to s_n;
            update the triggering kernel by (20);
            update the particle weight by (17);
        else if m > 1 documents d_n^{1:m} with the same t_n
        then
            sample {s_n^{1:m}}, {t_n^{1:m}} by Algorithm 2;
            update the particle weight by (21);
        end
    end
    Normalize particle weight;
    if ||w_n||_2^{-2} < threshold then
        resample particles;
end
```

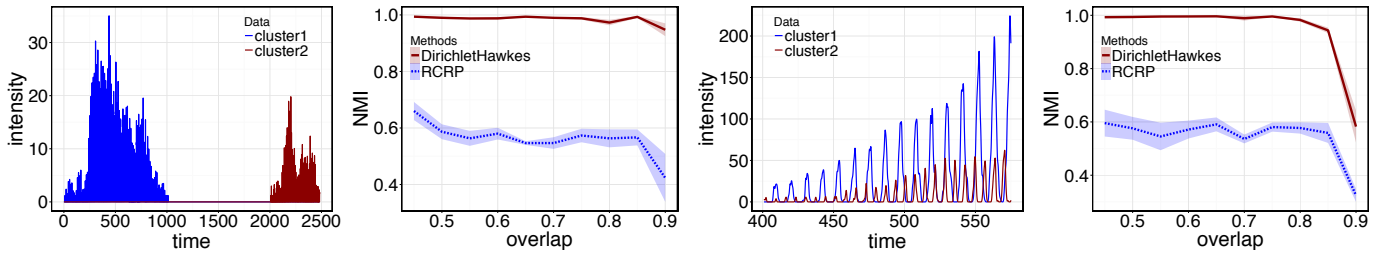
Algorithm 2: Gibbs sampling for cluster label and time

```
for iter = 1 to MaxIterG do
    if update cluster label s_n^i then
        remove t_n^i from cluster s_n^i and update the
        triggering kernel by (20);
        draw a new sample s_n^{i'} from (19);
        add t_n^i into cluster s_n^{i'} and update the triggering
        kernel by (20);
    else if update document time t_n^i then
        for iter = 1 to MaxIterM do
            draw a new sample t_n^{i'} ~ Unif(t_{n-1}, t_n);
            if r = P(t_n^{i'}, T_{s_n^i} \ t_n^i | s_n^i, rest) / P(t_n^i, T_{s_n^i} \ t_n^i | s_n^i, rest) > 1 then
            t_n^i ← t_n^{i'};
            else t_n^i ← t_n^{i'} with probability r;
        end
        update the triggering kernel of s_n^i by (20);
    end
end
```

to look back until we reach the time

$$t_u = t_n - \left(\tau_m + \sqrt{-2\sigma_m \log\left(0.5\epsilon\sqrt{(2\pi\sigma_m^2)}\right)} \right), \quad (22)$$

where $\tau_m = \max_l \tau_l$, $\sigma_m = \max_l \sigma_l$ and t_n is the current document time to guarantee that the error of the Gaussian summation with respect to each reference point τ_l is at most ϵ. Because the number of documents within $[t_u, t_n]$, referred to as the *active interval*, is expected to be constant as we run the algorithm for a while when the Hawkes Process becomes stationary, the average running time will keep stable in the long run. In addition, from the log of (15), for the newly added time t_n, we only need to add the new intensity value $\lambda_{s_n}(t_n)$, set the observation window $T = t_n$, and update the integral of the intensity function (16). Therefore, we can precompute and store the likelihood value for each sample $\boldsymbol{\alpha}_{s_n}^k$ and incrementally update it in each cluster. Similarly, in the Metropolis loop of Algorithm 2, we need to



(a) Temporally well-separated clusters. (b) Temporally interleaved clusters.

Figure 3: Effectiveness of Temporal Dynamics. Panel (a) and (b) show different cases where the clusters are temporally well-separated and interleaved, respectively. In each case, the left plot shows the intensity function of each cluster, and the right plot compares the performance by Normalized Mutual Information.

update the triggering kernel whenever a document time t_j is updated or deleted from a cluster. In this case, since the observation window T is fixed, we only need to recompute the affected intensity value $\lambda_{s_n}(t_j)$ and the affected individual summation terms in (16) for those time $t_i < t_n, t_i \in \mathcal{T}_{s_n}$.

In a nutshell, because we depend on the past documents only within the *active interval*, the above partial updating only performs the necessary calculations, and the overall memory usage and the expected time cost per document tend to be constant with respect to the number of incoming documents and the existing number of clusters, which is empirically verified in Figure 6 of the following experiments.

6. EXPERIMENTS

On massive synthetic and real-world datasets, in this section, we demonstrate that DHP not only can provide clusters of relevant news articles but also is able to uncover meaningful latent temporal dynamics inherent inside those clusters.

6.1 Synthetic Data

On synthetic data, we investigate the effectiveness of the temporal dynamics for improving clustering, the learning performance of the inference algorithm and the efficacy of the sampling method for missing time.

6.1.1 *Do temporal dynamics help?*

Because DHP exploits both temporal dynamics and textual contents, we expect that documents with similar topics and temporal patterns should be related to each other. On the other hand, for those documents with similar topics but different temporal behaviors, our model should still be able to disambiguate them to certain extent.

Experimental Setup. We simulate two clusters on a vocabulary set of 10,000 words. The word distribution of one cluster mainly concentrates on the first 8,000 words, and we shift the word distribution of the other one to have a varying vocabulary overlap from 45 to 90 percent. The triggering kernels of the clusters have two basic RBF kernels at 7 and 11 on the time line with bandwidth 0.5. We set $\alpha_0 = 1$ of the language model for both methods, and set $\lambda_0 = 0.01$ for DHP. We use the Normalized Mutual Information (NMI) to compare the uncovered clusters with the ground-truth clusters. The range of NMI is from 0 to 1, so larger values indicate better performance. All experiments are repeated for 10 times.

Results. In Figure 3(a), we first consider an easy case where the clusters are well-separated in time, which corresponds to the usual case that each cluster corresponds to a single short lifetime event. Because the clusters come and go sequentially in time, it helps to differentiate the clusters as their topics become more similar. This effect is verified in the right panel of Figure 3(a) where the NMI value is still close to one even when the topic vocabularies have 90% overlap. Besides, in Figure 3(b), we consider a more challenging situation where the clusters evolve side-by-side in time. Because the clusters have different triggering kernels, we can still expect the Dirichlet-Hawkes model to perform well. In the right panel of Figure 3(b), the performance of DHP only starts to decrease when the overlapping grows to 85-percent. Overall, because RCRP does not explicitly learn the temporal dynamics of each cluster, it cannot tell the temporal difference. In contrast, as DHP clusters the incoming documents, it also automatically updates its inference about the temporal dynamics of each cluster, and Figure 3 demonstrates that this temporal information could be useful to have better clustering performance.

6.1.2 *Can we learn temporal dynamics effectively?*

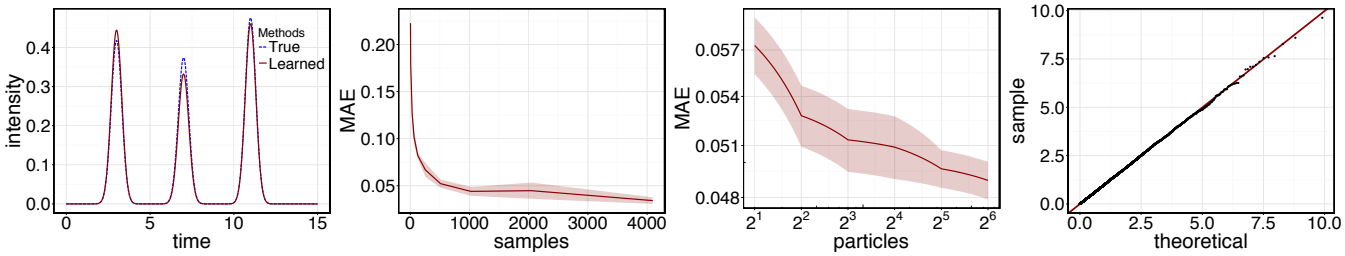
Experimental Setup. Without loss of generality, each cluster has RBF kernels located at 3, 7, and 11 with bandwidth 0.5. We let true coefficients of the triggering kernels for each cluster be uniformly generated from the simplex and simulated 1,000,000 documents. We randomly produce the missing time to allow at most three documents to arrive at the same time. The maximum Gibbs and Metropolis iteration is set to 100 and 50 with 8 particles in total.

Results. Figure 4(a) shows the learned triggering kernel for one randomly chosen cluster against the ground-truth. Because the size of the simulated clusters is often skew, we only compare the estimated triggering kernels with the ground-truth for the top-100 largest clusters. Moreover, Figure 4(b) presents the estimation error with respect to the number of samples drawn from the Dirichlet prior. As more samples are used, the estimation performance improves, and Figure 4(c) shows that only a few particles are enough to have good estimations.

6.1.3 *How well can we sample the missing time?*

Finally, we check whether the sampled missing time from Algorithm 2 are valid samples from a Hawkes Process.

Experimental Setup. Fixing an observation window $T = 100$, we first simulate a sequence of events $\mathcal{H}_T$ with the true triggering kernels described in 6.1.2. Given $\mathcal{H}_T$, the form of the intensity function $\lambda(t|\mathcal{H}_T)$ is fixed. Next, we



(a) Triggering Kernel (b) MAE vs. # Samples (c) MAE vs. # Particles (d) QQ-Plot of Samples

Figure 4: (a) Learned triggering kernels of one cluster from 1,000,000 synthetic documents; (b) Mean absolute error decreases as more samples are used for learning the triggering kernels; (c) A few particles are sufficient to have good estimation; (d) Quantile plot of the intensity integrals from the sampled document time.

equally divide the interval $[0, T]$ into five partitions $\{\mathcal{T}_i\}_{i=1}^5$, in each of which we incrementally draw $\Lambda_{\mathcal{T}_i} = \int_{\mathcal{T}_i} \lambda(t|\mathcal{H}_T)dt$ samples by Algorithm 2 using the true kernels. Then, we collect the samples from all partitions to see whether this new sequence is a valid sample from the Hawkes Process with the intensity $\lambda(t|\mathcal{H}_T)$.

Results. By the Time Changing Theorem [8], the intensity integrals $\int_{t_{i-1}}^{t_i} \lambda(\tau)d\tau$ from the sampled sequence should conform to the unit-rate exponential distribution. Figure 4(d) presents the quantiles of the intensity integrals against the quantiles of the unit-rate exponential distribution. It clearly shows that the points approximately lie on the line indicating that the two distributions are very similar to each other and thus verifies that Algorithm 2 can effectively generate samples for the missing time from the Hawkes Process of each cluster.

6.2 Real World Data

We further examine our model on a set of 1,000,000 mainstream news articles extracted from the Spinn3r[2] dataset from 01/01 to 02/15 in 2011.

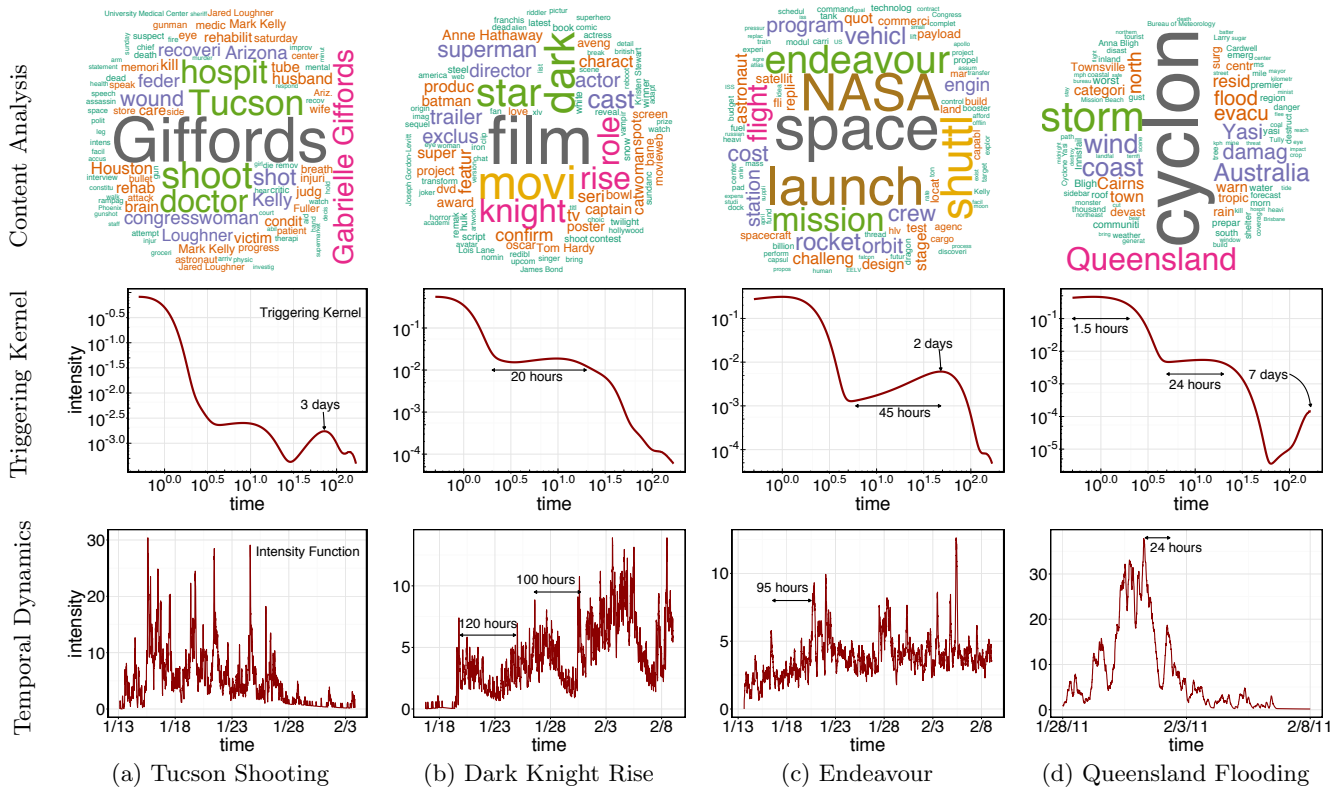
Experimental Setup. We apply the Named Entity Recognizer from Stanford NER system [15] and remove common stop-words and tokens which are neither verbs, nouns, nor adjectives. The vocabulary of both words and named entities is pruned to a total of 100,000 terms. We formulate the triggering kernel of each cluster by placing a RBF kernel at each typical time point : 0.5, 1, 8, 12, 24, 48, 72, 96, 120, 144 and 168 hours with the respective bandwidth being set to 1, 1, 8, 12, 12, 24, 24, 24, 24, 24, and 24 hours, in order to capture both the short-term and long-term excitation patterns. To enforce the sparse structure over the triggering kernels, we draw 4,096 samples from the Dirichlet prior with the concentration parameter $\alpha_0 = 0.1$ for each cluster. The intensity rate for the background Poisson process is set to $\lambda_0 = 0.1$, and the Dirichlet prior of the language model is set to $\phi_0 = 0.01$. In fact, the results are robust across a wide range of settings from 0.01 to 0.1 for both θ_0 and λ_0, which can be further tuned by following the techniques in [2]. We report the results by using 8 particles.

Content Analysis. Figure 5 shows four discovered example stories, including the 'Tucson shooting' event[3], the movie 'Dark Knight Rises[4]', Space Shuttle Endeavor's last mission[5], and 'Queensland Flooding' disaster[6]. The top row lists the top-100 frequent words in each story, showing that DHP can deduce the clusters with meaningful topics.

Triggering Kernels. The middle row of Figure 5 gives the learned triggering kernel of each story, which quantifies the influence over future events from the occurrence of the current event. For the 'Tucson Shooting' story, its triggering kernel reaches the peak within half an hour since its birth, decays quickly until the 30th hour, and then has a weak tailing influence around the 72nd hour, showing that it has a strong short-term effect, that is, most related articles and posts arrive closely in time. In contrast, the triggering kernel of the story 'Dark Knight Rises' keeps stable for around 20 hours before it decays below 10^{-4} by the end of a week. The continuous activities of this period indicate that the current event tends to have influence over the events 20 hours later.

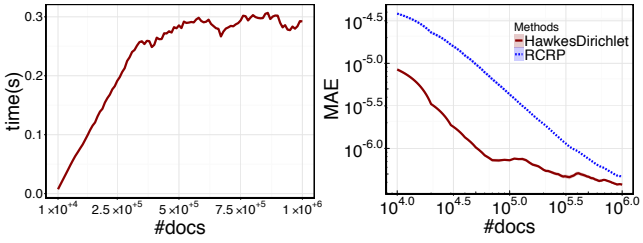
Temporal Dynamics. The bottom row of Figure 5 plots the respective intensity functions which indicate the popularity of the stories along time. We can observe that most reports of 'Tucson Shooting' concentrate within the following two weeks starting from 01/13/2011 and fade out quickly by the end of the month. In contrast, we can verify the longer temporal effect of the 'Dark Knight Rises' movie in Figure 5(b) where the temporal gaps between two large spikes are about several multiples of the 20-hour period. Because this story is more about entertainment, including the articles about Anne Hathaway's playing of the Cat-woman in the film as well as other related movie stars, it maintains a certain degree of hotness by attracting people's attention as more production details of the movie are revealed. For the NASA event we can see in the intensity function of Figure 5(c) the elapsed time between two observed large spikes is around a multiple of 45-hour, which is also consistent with its corresponding triggering kernel. Finally, for the event of 'Queensland Flooding', the 'Cyclone Yasi' intensified to a Category 3 cyclone on 01/31/2011, to a Category 4 on 02/01/2011, and to a Category 5 on 02/02/2011[7]. These critical events again coincide with the observed spikes in the intensity function of the story in Figure 5(d). Because the intensity functions depend on both the triggering ker-

[2] http://www.icwsm.org/data/
[3] http://en.wikipedia.org/wiki/2011_Tucson_shooting
[4] http://www.theguardian.com/film/filmblog/2011/jan/13/batman-dark-knight-rises
[5] http://en.wikipedia.org/wiki/Space_Shuttle_Endeavour
[6] http://en.wikipedia.org/wiki/Cyclone_Yasi
[7] http://en.wikipedia.org/wiki/Cyclone_Yasi



(a) Tucson Shooting (b) Dark Knight Rise (c) Endeavour (d) Queensland Flooding

Figure 5: Four example stories extracted by our model, including the 'Tucson Shooting' event, the movie of 'Dark Knight Rises', Space Shuttle's final mission and Queensland flooding disaster. For each story, we list the top 100 most frequent words on the top row. The middle row shows the learned triggering kernel in the log-log scale, and the last row presents the respective intensity functions along time.



(a) Scalability (b) Time prediction

Figure 6: Scalability and time prediction in real world news stream.

nels and the arriving rate of news articles, news reports of emergent incidents and disasters tend to be concentrated in time to form strong short-term clusters with higher magnitude of intensity values. In Figure 5, the intensity functions of both 'Tucson Shooting' and 'Queensland Flooding' have value greater than 20. In contrast, other types of stories in entertainment and scientific explorations might have continuous longer-term activities as more and more related details get revealed. Overall, the ability of uncovering topic-specific clusters with learned latent temporal dynamics of our model provides a better and intuitive way to track the trend of each evolving story in time.

Scalability. Figure 6(a) shows the scalability of our learning algorithm. Since the number of clusters grows logarithmically as the number of data points increases for CRP, we expect the average time cost of processing each document is keeping roughly constant after running for a long time period. This is verified in Figure 6(a) where after the build-up period, the average processing time per 10,000-document keeps stable.

Prediction. Finally, we evaluate how well the learned temporal model of each cluster can be used for predicting the arrival of the next event. Starting from the 5,000th document, we predict the possible arriving time of the next document for the clusters with size larger than 100. Since RCRP does not learn the temporal dynamics, we use the average inter-event gap between two successive documents as the predicted time interval between the most recent document and the next one in the future. For DHP, we simulate the next event time based on the learned triggering kernels and the timestamps of the documents observed so far. We treat the average of five simulated time as our final prediction and report the cumulative mean absolute prediction error in Figure 6(b) in the log-log scale. As more documents are observed, the prediction errors of both methods decrease. However, the prediction performance of DHP is even better from the very beginning when the number of documents is still relatively small, showing that the Hawkes model indeed can help to capture the underlying temporal dynamics of the evolution of each cluster.

7. DISCUSSIONS

In addition to RCRP, several other well-known processes can also be incorporated into the framework of DHP. For instance, we may generalize the Pitman-Yor Process [23] to incorporate the temporal dynamics. This simply brings back the constant rate for each Hawkes Process. A small technical issue arises from the fact that if we were to decay the counts $m_{k,t}$ as in the RCRP, we would obtain negative counts from $m_{k,t}-a$, where a is the parameter of the Pitman-Yor Process to increase the skewness of the cluster size distribution. However, this can be addressed, e.g., by clipping the terms by 0 via $\max(0, m_{k,t})$. In this form we obtain a model that further encourages the generation of new topics relative to the RCRP. Moreover, The Distance-Dependent Chinese Restaurant Process (DD-CRP) of [6] attempts to address spatial interactions between events. This generalizes the CRP and, with a suitable choice of distance function, can be shown to contain the RCRP as a special case. The same notion can be used to infer spatial / logical interactions between Hawkes Processes to obtain spatiotemporal effects. That is, we simply use spatial excitation profiles to model the rate of each event.

To conclude, we present the Dirichlet-Hawkes Process which is a scalable probabilistic generative model inheriting the advantages from both the Bayesian nonparametrics and the Hawkes Process to deal with asynchronous streaming data in an online manner. Experiments on both synthetic and real world news data demonstrate that by explicitly modeling the textual content and the latent temporal dynamics of each cluster, it provides an elegant way to uncover topically related documents and track their evolutions in time simultaneously.

Acknowledge

The research was supported in part by NSF IIS-1116886, NSF/NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983.

8. REFERENCES

[1] O. Aalen, O. Borgan, and H. Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

[2] A. Ahmed, J. Eisenstein, Q. Ho, E. P. Xing, A. J. Smola, and C. H. Teo. The topic-cluster model. In *Artificial Intelligence and Statistics AISTATS*, 2011.

[3] A. Ahmed, Q. Ho, J. Eisenstein, E. Xing, A. Smola, and C. Teo. Unified analysis of streaming news. In *Proceedings of WWW*, Hyderabad, India, 2011. IW3C2, Sheridan Printing.

[4] A. Ahmed and E. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *SDM*, pages 219–230. SIAM, 2008.

[5] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

[6] D. Blei and P. Frazier. Distance dependent chinese restaurant processes. In *ICML*, pages 87–94, 2010.

[7] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[8] D. Daley and D. Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*, volume 2. Springer, 2007.

[9] Q. Diao and J. Jiang. Recurrent chinese restaurant process with a duration-based discount for event identification from twitter. In *SDM*, 2014.

[10] A. Doucet, J. F. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In C. Boutilier and M. Goldszmidt, editors, *UAI*, pages 176–183, SF, CA, 2000.

[11] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[12] N. Du, L. Song, A. Smola, and M. Yuan. Learning networks of heterogeneous influence. In *NIPS*, pages 2789–2797, 2012.

[13] N. Du, L. Song, H. Woo, and H. Zha. Uncover Topic-Sensitive Information Diffusion Networks. In *Artificial Intelligence and Statistics (AISTATS)*, 2013.

[14] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping Social Activity by Incentivizing Users. In *NIPS*, 2014.

[15] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[16] T. Griffiths and Z. Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

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

[18] N. L. Hjort, C. Holmes, P. Muller, and S. G. Walker. *Bayesian Nonparametrics*. Cambridge University Press, 2010.

[19] J. Kingman. On doubly stochastic poisson processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, pages 923–930, 1964.

[20] J. F. C. Kingman. *Poisson processes*, volume 3. Oxford university press, 1992.

[21] L. Li, H. Deng, A. Dong, Y. Chang, and H. Zha. Identifying and labeling search tasks via query-based hawkes processes. In *KDD*, pages 731–740, 2014.

[22] C. Suen, S. Huang, C. Eksombatchai, R. Sosic, and J. Leskovec. Nifty: A system for large scale information flow tracking and clustering. In *WWW*, 2013.

[23] Y. W. Teh. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992, 2006.

[24] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *KDD*, 2006.