Exploiting Uncertainty in Popularity Prediction of Information Diffusion Cascades Using Self-exciting Point Processes

Quyu Kong[1,2,3], **Marian-Andrei Rizoiu**[2,3] and **Lexing Xie**[1,2]
[1]Australian National University
[2]Data61, CSIRO
[3]University of Technology Sydney
Quyu.Kong@anu.edu.au, Marian-Andrei.Rizoiu@uts.edu.au, Lexing.Xie@anu.edu.au

Abstract

Hawkes processes have been successfully applied to understand online information diffusion and popularity of online items. Most prior work concentrate on individually modeling successful diffusion cascades, while discarding smaller cascades which, however, account for a majority proportion of the available data. In this work, we propose a set of tools to leverage information in the small cascades: a joint fitting procedure that accounts for cascade size bias in the sample, a Borel mixture model and a clustering algorithm to uncover latent groups within these cascades, and the posterior final size distribution of Hawkes processes. On a dataset of Twitter cascades, we show that, compared to the state-of-art models, the proposed method improves the generalization performance on unseen data, delivers better prediction for final popularity and provides means to characterize online content from the way Twitter users discuss about it.

1 Introduction

User-generated online information in the forms of posts, videos and images today stimulate widespread discussion within or across online social media platforms such as Twitter and Youtube. Among the broad classes of models successfully applied to understand the popularity of such online information [Jin *et al.*, 2013; Zhang *et al.*, 2013], a class of point process based models — dubbed the Hawkes processes — has seen increasing attention [Zhao *et al.*, 2015; Kong *et al.*, 2020]. Such processes learn from the temporal patterns of sharing events and from fine-grained features of online diffusions to produce explainable parameters to quantify and predict popularity. Most modeling efforts are generally aimed towards "popular" diffusions, whereas unpopular ones are usually discarded — Zhao *et al.* [2015] only study cascades with at least 50 retweets, and Mishra *et al.* [2016] threshold at 20 retweets —, the goal being to learn what makes a popular diffusion. Given that Cheng *et al.* [2014] and later Rizoiu *et al.* [2018] have shown that luck is an important factor in online popularity, and that the discarded diffusions make up for a large portion of the available data (more than 40% events as seen in Section 6), this paper aims to establish a procedure to leverage the temporal information in "unpopular" diffusions by jointly modeling these with Hawkes processes.

Specifically, in this work, we address three open questions concerning online information diffusion modeling.

Cheng *et al.* [2014] have shown that the final popularity, i.e., total number of events, of retweet diffusions is unpredictable due to uncertainties contributed by various factors. Recent work from [Rizoiu *et al.*, 2018] seconds this conclusion with distributions of final popularity where the same information has high probabilities of both remaining unknown or getting extremely popular. The popular diffusions are the "lucky" ones, and the first open question is: **can we gain knowledge about popularity of online information from short diffusions?** Here, we first jointly model multiple diffusion sequences using a single Hawkes process by summing up their log-likelihood functions, and we show that this introduces bias in parameter estimation due to the non-representativity of the sample as only small cascades are observed. Next, we show how the parameter bias can be adjusted by accounting for the distribution of final event counts of Hawkes processes [Daw and Pender, 2018]. We also show that fitting short diffusions enjoys a better time efficiency.

The aforementioned joint modeling assumes that the observed diffusions are generated by the same process, however in real-life situations multiple latent processes are simultaneously generating cascades. The second challenge emerges — **how do we both heuristically and systematically uncover diffusions coming from same models, and how to simultaneously learn the models parameters?** The heuristic method is with regard to the information content, e.g., grouping retweet diffusions about the same video in Twitter. We design a Borel mixture model and a clustering algorithm to systematically regroup similar cascades, while simultaneously learning the model parameters.

After learning the models from short diffusions, the next question is: **what are the tools we can develop to explore the uncertainty of predicting popularity**. While fitted parameters or derived quantities are commonly used for analyzing content popularity [Rizoiu *et al.*, 2017], we show that we can characterize the virality of online content based solely on how people discuss about it. Moreover, we present tools for predicting final popularity along with its uncertainty.

The main contributions of this work are:

- We refine the likelihood function of Hawkes processes

to jointly and correctly model size-biased diffusions.

- We design procedures to uncover latent clusters through heuristics and algorithms. Specifically, we apply a mixture model and the k-means on fitting branching factors and kernel functions, respectively. We also study the quantification of popularity with parameters fitted via the procedures on real data.
- On a real-world Twitter diffusion dataset, we show that better generalization performances are achieved on hold-out proportions by learning from short diffusions compared to benchmarks trained on individual popular diffusions. We then show an improved early prediction of final popularities.
- We construct the *ActiveRT2017* Twitter cascade dataset.

Related work. Generative models are commonly employed for modeling temporal diffusions of online information. Such models are designed to predict final popularities [Zhao *et al.*, 2015; Samanta *et al.*, 2017], uncover hidden diffusion networks [Gomez-Rodriguez *et al.*, 2011] and detect rumors [Ma *et al.*, 2016]. The same tasks were also approached using feature-driven models, which train machine learning algorithms that use temporal features — statistical summaries of temporal patterns — together with user features and content features [Bakshy *et al.*, 2011; Martin *et al.*, 2016]. However, to our knowledge, most of the prior work concentrate on large (popular) cascades, and the complete temporal information of the unpopular diffusions is rarely considered.

Hawkes processes [Hawkes and Oakes, 1974] are a class of self-excitating point processes — past events excites future events happening — widely applied in analyzing social media [Kobayashi and Lambiotte, 2016; Lukasik *et al.*, 2016; Farajtabar *et al.*, 2015], earthquake aftershocks [Ogata, 1988], crime rate [Mohler and others, 2013], invasive species [Gupta *et al.*, 2018], energy consumption [Li and Zha, 2016] and finance [Bacry *et al.*, 2015]. Event-level and sequence-level clusterings of Hawkes processes have been discussed in [Du *et al.*, 2015] and [Yang and Zha, 2013; Xu and Zha, 2017], and particular attention has been given to the inference of Hawkes processes [Yan *et al.*, 2018; Guo *et al.*, 2018; Liu *et al.*, 2018]. The present work extends the prior literature in several ways. First, we propose a joint inference procedure which accounts for length-biased diffusions — i.e. observing short cascades only. Second, we propose a Borel Mixture Model and an efficient clustering procedure that regroups similar diffusions together.

2 Preliminaries

In this section, we first define our data objects: the diffusion cascades. Next, we introduce the Hawkes processes, together with essential concepts including its cluster representations, branching factor, size distribution and likelihood functions.

Diffusion cascades. In online social media platforms, such as Twitter, users read contents posted by others, and they can share/retweet, exposing the contents to broader audience. This diffusion usually continues until the content shifts away from the users' attention. The initial posting event and the following share/retweet events together constitute a diffusion

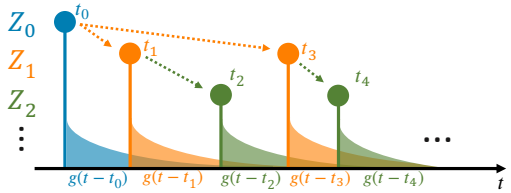


Figure 1: The cluster representation of a Hawkes process. Each individual event t_i initiates an inhomogeneous Poisson process with the intensity function $g(t - t_i)$ (identical for all events). Different generations of events are shown in distinct colors, arrows indicate the parent-offspring relation, and the event counts at each generation form a branching process, i.e., $\{Z_0, Z_1, Z_3, \dots\}$.

cascade. Mathematically, we denote a cascade i as $\mathcal{H}_i = \{t_0, t_1, t_2, \dots, t_{N_i-1}\}$ where $N_i \geq 1$ is a random event number count, $\forall t_j \in \mathcal{H}_i$ are random event times on $[0, \infty)$ relative to t_0 and $t_0 = 0$ is the initial event time. Let $\mathcal{H}_i(T), N_i(T)$ represent the event set and the event count before time T, respectively, i.e., $\mathcal{H}_i(T) = \{t_j \mid t_j \in \mathcal{H}_i, t_j < T\}$ and $N_i(T) = |\mathcal{H}_i(T)|$. The *popularity* of the online content associated with the cascade i is defined by N_i its event count.

Hawkes processes are particular classes of self-exciting processes — in which the occurrence of new events will increase the likelihood of future event happening [Hawkes, 1971]. In Hawkes processes, the event intensity is a function conditioned on the past occurred events:

$$\lambda(t \mid \mathcal{H}_i(t)) = \mu + \sum_{t_j \in \mathcal{H}_i(t)} n^* g(t - t_j) \qquad (1)$$

where μ is the background event rate, n^* is known as the *branching factor* and $g : R^+ \to R^+$ is a memory kernel encoding the time-decaying influence of past events on future events. Note that for information cascades (such as retweet cascades on Twitter) there is no background intensity, as all the retweets are considered to be spawning from the original tweet. Therefore, $\mu = 0$, and $\int_0^\infty g(\tau)d\tau = 1$. Common choices of the memory kernels include the exponential kernel function [Xu *et al.*, 2016], $g_{EXP}(\tau) = \theta e^{-\theta\tau}$, the power-law kernel [Mishra *et al.*, 2016], $g_{PL}(\tau) = \theta c^\theta (\tau + c)^{-(1+\theta)}$, among others (see [Kong *et al.*, 2020] for a review of kernels used with diffusion cascades).

Cluster representation and size distribution. An alternate representation of the Hawkes self-exciting process is that of a latent cluster of Poisson processes, introduced by Hawkes and Oakes [1974]. Fig. 1 depicts the cluster representation of an example Hawkes process, with highlighted parent-offspring relations between events. Each event generates offspring events following an inhomogeneous Poisson process with the intensity function $n^* g(t)$ and, therefore, its number of offspring follows a Poisson distribution of parameter $\int_0^T n^* g(t)dt$. When $T \to \infty$, the event counts at each generation — denoted as $\{Z_0, Z_1, Z_2, \dots\}$ — produce a Galton-Watson branching process whose offspring distribution is a Poisson distribution with the intensity n^* [Durrett, 2010]. The total size of a Hawkes process can be then computed as $N = \sum_n Z_n$. This quantity is known as *total progeny number* of the branching process, and its distribution is the Borel

distribution [Borel, 1942], denoted as $\mathbb{B}(\kappa \mid n^*)$:

$$\mathbb{B}(\kappa \mid n^*) = \mathbb{P}[N = \kappa \mid n^*] = \frac{(\kappa n^*)^{\kappa-1} e^{-\kappa n^*}}{\kappa!} \quad (2)$$

This analytical form for the Hawkes process size distribution has not been discussed until recently [Daw and Pender, 2018]. Eq. (2) holds for branching factors bounded by $n^* \leq 1$.

Parameter estimation. The parameters of a Hawkes process can be estimated by maximizing the likelihood function of a general point process [Daley and Vere-Jones, 2008]:

$$L(\Theta \mid \mathcal{H}_i(T)) = e^{-\int_0^T \lambda(\tau) d\tau} \prod_{t_j \in \mathcal{H}_i(T)} \lambda(t_j) \quad (3)$$

in which $\lambda(\cdot)$ is the intensity function defined in Eq. (1).

3 Joint Fitting of Hawkes Cascades

In this section, we propose a method to jointly learn a single set of parameters from a collection of Hawkes realizations biased in term of event count. We first discuss the estimation bias when modeling on size-biased realizations, and next we propose a modified likelihood function to eliminate such bias.

Joint likelihood function. Let $\mathcal{H}^r = \{\mathcal{H}_1, \mathcal{H}_2, \dots\}$ be a *representative* set of independent Hawkes realizations, assumed to be generated from the same model Θ and without any post-generation filtering applied. It is then straightforward to estimate Θ by maximizing the joint likelihood $\mathcal{L}^r(\Theta \mid \mathcal{H})$ defined as the sum of the individual log-likelihoods (i.e., the log of Eq. (3)):

$$\mathcal{L}^r(\Theta \mid \mathcal{H}^r) = \sum_{\mathcal{H}_i \in \mathcal{H}^r} \log L(\Theta \mid \mathcal{H}_i) \quad (4)$$

This is asymptotically equivalent to minimizing the Kullback-Leibler (KL) divergence between the underlying Hawkes process where cascades are sampled from and the theoretical distribution parameterized by Θ.

Jointly fitting a biased set. Any filtering applied on the realizations post-generation — such as selecting realizations based on their final size — renders Eq. (4) non-applicable, and introduces systematic bias in parameter estimation (as shown empirically in our experiments in Section 6.1). Let N^* be a set of positive integers defining selected realization sizes, and let $\mathcal{H}^b$ be the *biased* set of realizations of sizes in N^*, i.e. $\mathcal{H}^b = \{\mathcal{H}_i \in \mathcal{H}^r \mid N_i \in N^*\}$. Using Bayes theorem and the Borel distribution of Hawkes sizes, we compute the joint likelihood for the set $\mathcal{H}^b$:

$$\begin{aligned} \mathcal{L}^b(\Theta \mid \mathcal{H}^b) &= \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \frac{f(\mathcal{H}_i \mid \Theta)}{\mathbb{P}[N_i \in N^* \mid \Theta]} \\ &= \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \frac{L(\Theta \mid \mathcal{H}_i)}{\sum_{j \in N^*} \mathbb{B}(j \mid \Theta)} \end{aligned} \quad (5)$$

where $f(\mathcal{H}_i \mid \Theta)$ is the probability density of realization i under model parameters Θ, therefore $f(\mathcal{H}_i \mid \Theta) = L(\Theta \mid \mathcal{H}_i)$). Finally, we plug Eq. (3) into Eq. (5) and we see that the joint likelihood function can be rearranged as a sum of two functions of independent parameters:

$$\mathcal{L}^b(\Theta \mid \mathcal{H}^b) = \mathcal{L}_g(\Theta_g \mid \mathcal{H}^b) + \mathcal{L}_n(n^* \mid \mathcal{H}^b) \quad (6)$$

$\mathcal{L}_g$ is a function of Θ_g — the parameter set of $g(\cdot)$ — $\mathcal{L}_n$ is a function of n^* the branching factor, and $\Theta = \Theta_g \cup \{n^*\}$:

$$\mathcal{L}_g(\Theta_g \mid \mathcal{H}^b) = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \sum_{t_j \in \mathcal{H}_i, j \geq 1} \log \sum_{t_k < t_j} g(t_j - t_k \mid \Theta_g) \quad (7)$$

$$\mathcal{L}_n(n^* \mid \mathcal{H}^b) = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \frac{(n^*)^{N_i - 1} e^{-N_i n^*}}{\sum_{j \in N^*} \mathbb{B}(j \mid n^*)} \quad (8)$$

The above results indicate that Θ_g and n^* can be learned independently in two separate phases, by maximizing $\mathcal{L}_g$ and $\mathcal{L}_n$. This amounts to fitting n^* from observed final realization sizes only, and Θ_g from inter-arrival times between events.

4 Uncovering Clusters of Hawkes Models

In practice, it is often unknown which realizations were generated from the same model parameters. In this section, we examine several strategies to construct clusters of diffusion cascades. Finally, we introduce a Borel mixture model (BMM) and a modified k-means algorithm to automatically discover clusters based on cascade sizes and time intervals.

Heuristic grouping. For diffusion cascades relating to online content, one natural grouping is based on the explicit features of the content. For instance, in the *ActiveRT* dataset [Rizoiu *et al.*, 2018], each cascade records a retweet event series relating to a Youtube video, so one could group together cascades about the same video. Another example would be grouping cascades that are initiated by same users. On the up side, the heuristic grouping builds content-related models depending on the grouping criterion — i.e., models describing the online videos or Twitter users — in addition to describing generated cascades. On the flip side, not all cascades relating to a video or a user might be generated by the same process, and they might in reality not share the same parameters.

Algorithmic grouping. We are given a set of cascades $\mathcal{H}^b$ with a known cascade size filtering condition N^*, and K latent generative models with an unknown relation to the cascades in $\mathcal{H}^b$. We seek to learn the K values of n^* and Θ_g, and the membership of each cascade to the models (denoted as clusters). As indicated in Section 3, we cluster n^* and $g(t)$ separately.

We model the n^* for each cascade using a mixture model of Borel distributions (BMM), and we present an efficient EM estimation algorithm. A BMM can be fitted on $\mathcal{H}^b$ by maximizing the likelihood

$$L_{BMM} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \sum_{k=1}^{K} \underbrace{p_k \frac{\mathbb{B}(N_i \mid n_k^*)}{\sum_{j \in N^*} \mathbb{B}(j \mid n_k^*)}}_{q(k, N_i)} \quad (9)$$

where $N_i = \mid \mathcal{H}_i \mid$, and p_k denotes the mixture probability of the kth cluster parameterized by n_k^*. As maximizing Eq. (9) directly suffers from the identifiability [Bishop, 2006], we apply the Expectation-Maximization (EM) algorithm commonly used for learning mixture models. Next we give the update formulas for the E and M steps, the detailed derivations can be found in [supplement, 2020].

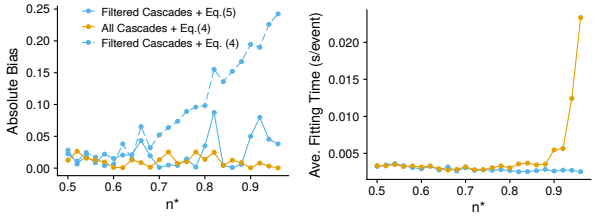


Figure 2: Validation of cascade joint learning on synthetic data. On simulated cascades at given n^*, the absolute fitting bias of n^* with Eq. (5) and Eq. (4) are compared for fitting on $N^* = \{1, \ldots, 20\}$ and all cascades. The right panel shows average times per event a method takes.

Update membership probabilities (E-step): The probability of N_i being a member of k is defined as

$$p(k \mid N_i) = \frac{q(k, N_i)}{\sum_{j=1}^{K} q(j, N_i)} \tag{10}$$

Update n_k^* and p_k (M-step): When N^* is the natural number set, namely there is no filtering, n_k^* is updated analytically, and the update formulas for n_k^* and p_k are

$$(n_k^*)^{new} = \frac{\sum_{N_i} p(k \mid N_i)(N_i - 1)}{\sum_{N_i} p(k \mid N_i)(N_i)}, (p_k)^{new} = \frac{\sum_{N_i} p(k \mid N_i)}{|\mathbb{N}|}$$

When filtering is imposed, n_k^* can be efficiently solved by numerically finding roots of the simplified first partial derivative of Eq. (9) w.r.t. n_k^*.

For finding $g_k(t)$ kernel functions, we employ a K-means like procedure in which Θ_g serve as the equivalent of centroids. We start by randomly assigning cascades to clusters, and we use maximize Eq. (7) for each cluster to recompute its centroid. Cluster reassignment is performed by selecting for each cascade the cluster whose centroid (Θ_g) maximizes its likelihood function (Eq. (3)).

As the branching factors n_k^* and the kernel functions parameters Θ_g are inferred separately, there is no exact matched pair of parameters between them — two cascades might have the same n^* but different Θ_g, or the other way around.

5 Prediction for Partial Observations

In this section, we first describe how fitted parameters are chosen for newly observed sequences, based on previously trained clusters of Hawkes models. Next, we then derive predictions for their popularities — i.e., the final cascade size.

Parameter selection. Given a partially observed event sequence $\mathcal{H}_i(T)$ where T is the observation time, we select for it a parameter set Θ_g and a branching factor n^* from the candidates constructed using the clustering procedure. We first chose the kernel function parameters that maximize its $\mathcal{L}_g$ likelihood function (Eq. (7)), denoted as $\hat{\Theta}_g$. We then use $\hat{\Theta}_g$ to select the best n_k^* that maximizes the probability

$$\mathbb{P}[K = k \mid \mathcal{H}_i(T)] = \frac{p_k L(\{n_k^*, \hat{\Theta}_g\} \mid \mathcal{H}_i(T))}{\sum_{j=1}^{K} p_j L(\{n_j^*, \hat{\Theta}_g\} \mid \mathcal{H}_i(T))} \tag{11}$$

When $T = 0$, however, the kernel function cannot be identified and we only maximize the $\mathcal{L}_n$ function (Eq. (8)).

Posterior size distribution. To be able to make prediction using the chosen parameters, it is desirable to derive the posterior size distribution given $\mathcal{H}_i(T)$. The future events after time T are of two kinds: direct offspring of observed events (their count denoted as N_i^d) and indirect offspring (children of children, total count denoted as N_i^{ind}). The process generating direct offspring is an inhomogeneous Poisson process of conditional intensity $\lambda(t \mid \mathcal{H}_i(T)), t > T$ — note that this is not a stochastic function as only the history up to time T is accounted in the intensity function. Consequently, N_i^d follows a Poisson distribution of parameter $\Lambda_i(T) = \int_T^\infty \lambda(\tau \mid \mathcal{H}_i(T)) d\tau$. Furthermore, each direct offspring initiated a Hawkes process and its total progeny number follows a Borel distribution. Given the number of direct offspring N_i^d, the total number of direct and indirect offspring follows a Borel-Tanner distribution (also known as the generalized Borel distribution):

$$\mathbb{B}(\kappa \mid n^*, N_i^d) = \frac{(\kappa n^*)^{\kappa - N_i^d} e^{-\kappa n^*}}{\kappa(\kappa - N_i^d)!} \tag{12}$$

The proof of this is straightforward and leverages the general hitting time theorem [Van der Hofstad and Keane, 2008], which can be found in [supplement, 2020].

Finally, the posterior cascade size distribution is therefore

$$\mathbb{P}[N_i = n \mid \mathcal{H}_i(T)] = N_i(T) + \sum_{z=0}^{n - N_i(T)} Poi(z \mid \Lambda_i(T)) \mathbb{B}(n - z - N_i(T) \mid n^*, z) \tag{13}$$

where $Poi(\cdot)$ is the Poisson distribution. Eq. (13) leads to a quadratic complexity in computing the final size distribution, which is intractable in most real-life scenarios. We apply a numerical trick to reduce the complexity by introducing a threshold probability ϵ_p and summing until $Poi(z \mid \Lambda_i(T)) < \epsilon_p$.

Prediction. The size of real-life diffusion cascades is mechanically limited by the available population and the span of human attention. Therefore, it is logical impose an upper bound on the cascade size n_{max}, leading to $\mathbb{P}[N_i = n \mid \mathcal{H}_i(T), n \leq n_{max}]$. In particular, for retweet cascades, n_{max} can be estimated by the cumulative number of users exposed by the online item (computed as the sum of the followers of the users that were observed retweeting). Given the distribution, one is able to compute the expected final event count, its variance and the probability of a cascade diffusing beyond certain sizes. It is worth noting that when $n_{max} = \infty$, the expected final event count has an analytical solution as applied in [Zhao *et al.*, 2015; Mishra *et al.*, 2016].

6 Experiments

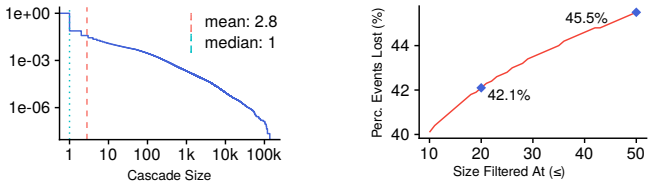
This section provides evaluation results of the proposed modeling procedures on both synthetic data and real data. On synthetic data, we assess the effect of different cascade size filtering conditions. We then present a retweet cascade dataset together with some measurements on it. Finally, we conduct experiments evaluating model generalization on unseen data and final popularity prediction performance compared to the state-of-art models [Zhao *et al.*, 2015; Mishra *et al.*, 2016].

Table 1: Chi-square tests of BMM fitted on size-biased cascades with various filtering conditions. Each cell shows the percentage of tests passing 0.05 significance level.

#cluster	$N^{max} = 20$	$N^{max} = 30$	$N^{max} = 40$	$N^{max} = 50$	All sizes
1	90.1%	92.4%	93.7%	95.3%	96.1%
2	82%	89.7%	93.7%	94.7%	98.2%
3	80.5%	88.5%	93.9%	94.7%	98.8%

Table 2: Purity tests of the power law kernel function clustering experiments with various filtering condition. Each cell shows the mean purity measures and the standard deviation.

#cluster	$N^{max} = 20$	$N^{max} = 30$	$N^{max} = 40$	$N^{max} = 50$	All sizes
2	$88.5 \pm 11.4\%$	$91.8 \pm 9.3\%$	$92.1 \pm 9.5\%$	$92.2 \pm 8.6\%$	$93.0 \pm 8.0\%$
3	$76.5 \pm 11.2\%$	$80.5 \pm 9.9\%$	$79.3 \pm 10.9\%$	$80.9 \pm 10.3\%$	$83.6 \pm 9.6\%$



(a) (b)

Figure 3: Profiling the *ActiveRT2017* dataset: **(a)** the empirical complementary cumulative density (CCDF) of cascade sizes in the dataset; **(b)** the percentages of events filtered when imposing various minim cascade sizes. The two points highlight the event loss as a minimum threashold of 20 and 50 events, introduced in [Mishra *et al.*, 2016] and [Zhao *et al.*, 2015], respectively.

6.1 Synthetic experiments

Bias from joint fitting. Given various n^* values, we simulate 100 cascades for each. On simulated cascades, we fit Hawkes processes with three different settings: fit on all cascades with Eq. (4), and fit on cascades filtered at the event size 20 with Eq. (4) and Eq. (5). The absolute bias of fitted n^* is reported along with the average times the fitting takes. Fig. 3 shows the bias in learning n^* values when fitting on size-biased cascades with Eq. (4) and the correctness of using Eq. (5) to adjust this bias. It also highlights the efficiency when learning on small cascades. As the computational complexity of Eq. (4) is quadratic to cascade sizes, Eq. (5) allows one to bound the complexity to a maximum size thus scaling to more cascades.

BMM goodness-of-fit. We train BMMs on different size-biasing conditions imposed on simulated cascades and examine goodnesses of the fitted BMM on complete cascade data by conducting Chi-square tests between the empirical and learnt size distributions. For each experiment, branching factors are randomly sampled for clusters and used to simulate to the same number of cascades for each cluster summing to 1000 cascades in total. This is repeated for 1000 times.

Table 1 shows the proportion of repeated experiments that pass the tests at a 0.05 significance level. We tabulate the tests against two dimensions: the number of clusters — up to 3 clusters — and cascade size filters — cascades with sizes less than or equal to N^{max} are kept where $N^{max} \in \{20, 30, 40, 50\}$ or $N^{max} \to \infty$. The passing rates decrease as less cascades and more clusters are provided. However, we can see that, overall, BMMs fitted only on short diffusions can still generalize well to the whole dataset.

Kernel clustering. We measure the correctness of clustering on kernels and power law kernels are used. During each experiment, the sampled kernel parameters are enforced to have absolute difference 1 between cluster parameters to ensure the clusters are distinguishable. Other setup follows the BMM experiments and the clustering purity values are reported in Table 2. The same observation of performance increase during the growth of N^{max} and cluster numbers is presented.

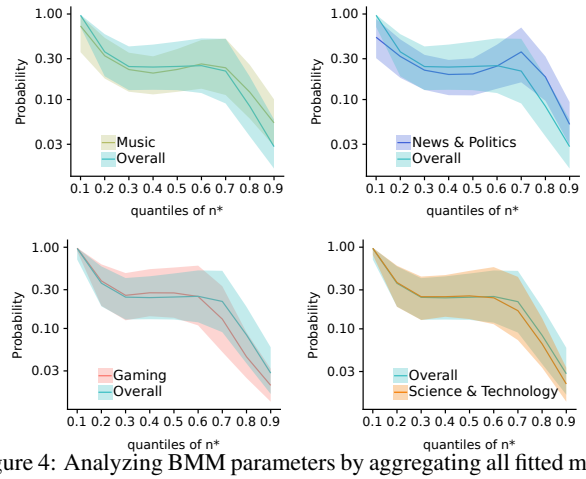
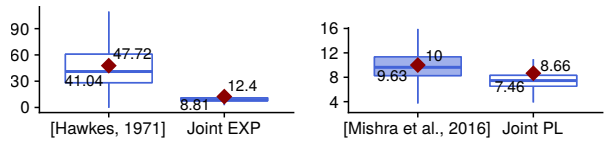


Figure 4: Analyzing BMM parameters by aggregating all fitted models at video categories. At each quantile value of the fitted branching factors (x axis), mixture probabilities are summarized with median (solid line) $\pm 25\%$ quantiles (the colored area). The three panels compare *Music*, *News & Politics*, *Gaming* and *Science & Technology* categories to the average, respectively.

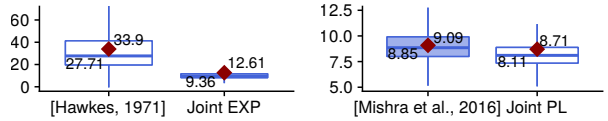
6.2 Jointly modeling diffusions on Twitter

Dataset. Retweet cascade datasets provided in prior works typically have filters on cascade sizes. In order to obtain a complete set of cascades, we produce the *ActiveRT2017* dataset crawled via Twitter public APIs through the entire 2017. In this dataset, all tweets are related to Yotube videos whose video ids are found in tweet contents. Selected videos are published by the *active* Youtube channels where each video has the maximum cascade size larger than 50 and associated to at least 500 cascades. The definition of *active* videos can be found in [Rizoiu *et al.*, 2017] and video meta data is collected using [Wu *et al.*, 2018]. In total, there are about 110k videos and 45 million cascades. Fig. 3a presents a statistical summary of cascade sizes, in which the CCDF shows a long tail distribution of cascade sizes. To quantify the data loss due to cascade size filters, Fig. 3b presents the proportions of events (tweets) being lost when filtering out cascades smaller than certain sizes. We note that, in our experiments, we adopt the filter applied in [Zhao *et al.*, 2015] at cascade size 50 to distinguish short diffusions, accounting for about 46% of the total events.

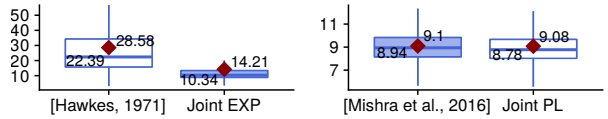
Pre-learn from short diffusions. In experiments, we conduct cascade joint fitting on the diffusions happened at the early part of 2017, and we then compare it to the state-of-art models on the cascades initiated later in 2017. While methods in Section 3 requires terminated cascades, we assume all cascades in our dataset have stopped diffusing given the data collection time. Specifically, within *ActiveRT2017* dataset, cascades have sizes less than 50 and stopped earlier than 1st of May in 2017 are selected, resulting $12,690,817$ cascades



(a) 1 minute observation time



(b) 10 minutes observation time



(c) 1 hour observation time

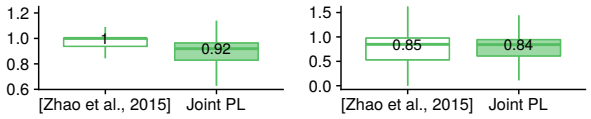
Figure 5: **Holdout negative log-likelihood per event** of four models on *ActiveRT2017* given three different observation times. Red diamonds show the mean values for each boxplot — lower is better.

in total. As Youtube videos are associated to cascades, we first apply it as a simple grouping heuristic. Then the clustering procedures described in Section 4 are used to discover latent groups on this sample. During the clustering phase, we fix cluster numbers for fitting both BMM and kernel functions at 8 and 3, respectively. This design choice is applied in consideration of efficiency. The fitted models will be used in all following experiments.

Profiling Youtube videos. By jointly modeling cascades related to same videos, each fitted model extracts general diffusion patterns from individual cascades. Here, we investigate the measurement of fitted parameters at an aggregation level. Fig. 4 depicts fitted BMM parameters of each video aggregated at a video category level where the categories of Youtube videos are thematically similar. The branching factors determine the diffusion speed and thus can be seen as indications of content virality. We find that overall music and news video diffusions are more viral than average, whereas gaming and scientific videos show the opposite behavior.

Generalization on unseen data. We evaluate the improvement of model generalization performances by leveraging pre-learnt parameters from short diffusions. Given the trained models for each video, the generalization performance is evaluated on all cascades related to this video with more than 50 events (33,023 cascades). With a cascade $\mathcal{H}_i$ and an observation time T, we first obtain the cluster parameters through its video id and parameter selections are conducted on the observed part $\mathcal{H}_i(T)$ following Section 5. We apply this setting to Hawkes processes with an exponential kernel function (Joint EXP) and a power-law kernel function (Joint PL), comparing to their counterparts fitted only on $\mathcal{H}_i(T)$ ([Hawkes, 1971] and [Mishra *et al.*, 2016]). We report the negative log-likelihood values normalized by event counts on the remaining part, i.e., $L(\theta \mid \mathcal{H}_i(T)) - L(\theta \mid \mathcal{H}_i)$.

Fig. 5 gives the generalization performances as boxplots with mean values. At individual observation times, the direct comparison between models with pre-learnt parameters on short diffusions consistently outperform models trained on given observations. This highlights the improvements from the proposed fitting approach especially for the exponential kernel function where a large enhancement is shown. We also note that models with power law kernels are better than those with exponential kernels which reinforces the known conclusion from prior works [Mishra *et al.*, 2016]. When observation times are concerned, we can see that the advantage of applying the pre-learnt parameters diminishes as time lengths increase. This provides a hint for our proposed pre-learning procedure to handle the early-start modeling.



(a) 1 minute observation time (b) 10 minute observation time

Figure 6: Cascade final popularity prediction of Joint PL and Seismic, evaluated with Absolute Relative Error — lower is better.

Final popularity prediction. We compare the Joint PL against *Seismic* [Zhao *et al.*, 2015] on *ActiveRT2017*. Instead of using the same cascades, we only use these cascades initiated after 1st of June for measuring the effect of historical diffusion data. This provides a sample of 2,731 cascades. The parameters are selected on the observed part for each cascade and the corresponding final popularity is predicted as detailed in Section 5. We apply the Absolute Relative Error (ARE) metric as *Seismic* for performance evaluation. ARE is defined as $\frac{|\hat{N}_i - N_i|}{N_i}$ where $\hat{N}_i$ and N_i are the predicted popularity and the actual popularity of the cascade i respectively.

From Fig. 6, we can see that the relative prediction performance of Joint PL compared to *Seismic* is better on a shorter observation time. This supports the pattern shown in generalization performances amplifying our conclusion of the benefits from pre-training to early predictions.

7 Conclusion

Overall, this work is concerned with the way how short cascades are handled for modeling information diffusions with Hawkes processes. Instead of filtering, we propose to jointly pre-train Hawkes processes on cascades from same groups. We first adjust the Hawkes likelihood function to correctly fit on size-biased cascades by leveraging an analytical diffusion size distribution. To group cascades in real data, apart from applying simple heuristics, we future propose the procedures to automatically identify groups from a collection of cascades which we validate by conducting experiments on synthetic data. On a retweet cascade dataset, we analyze fitted models as indications of content virality. We also measure the improvement on models augmented with our pre-trained parameters and compare to the state-of-art generative model on predicting final popularities.

Limitations and future work. Due to the restriction of the size distribution of Hawkes processes, the current joint fitting on size-biased cascades is restricted to complete and unmarked processes. We plan to relax these constraints to allow for joint modeling with more flexibility.

References

Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 2015.

Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *WSDM*, 2011.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Émile Borel. Sur l'emploi du théoreme de bernoulli pour faciliter le calcul d'une infinité de coefficients. application au probleme de l'attentea un guichet. 1942.

Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *WWW*, 2014.

Daryl J Daley and David Vere-Jones. Conditional intensities and likelihoods. In *An introduction to the theory of point processes*, volume I, chapter 7.2. Springer, 2008.

Andrew Daw and Jamol Pender. The queue-hawkes process: Ephemeral self-excitement. *arXiv*, 2018.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *KDD*. ACM, 2015.

Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network coevolution. In *NIPS*, 2015.

Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. 2011.

Ruocheng Guo, Jundong Li, and Huan Liu. INITIATOR: Noise-contrastive Estimation for Marked Temporal Point Process. In *IJCAI*, 2018.

Amrita Gupta, Mehrdad Farajtabar, Bistra Dilkina, and Hongyuan Zha. Discrete Interventions in Hawkes Processes with Applications in Invasive Species Management. In *IJCAI*, 2018.

Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 1974.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 1971.

Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *SNA-KDD*. ACM, 2013.

Ryota Kobayashi and Renaud Lambiotte. Tideh: Time-dependent hawkes process for predicting retweet dynamics. In *ICWSM*, 2016.

Quyu Kong, Marian-Andrei Rizoiu, and Lexing Xie. Modeling information cascades with self-exciting processes via generalized epidemic models. 2020.

Liangda Li and Hongyuan Zha. Household Structure Analysis via Hawkes Processes for Enhancing Energy Disaggregation. In *IJCAI*, IJCAI, 2016.

Yanchi Liu, Tan Yan, and Haifeng Chen. Exploiting Graph Regularized Multi-dimensional Hawkes Processes for Modeling Events with Spatio-temporal Characteristics. In *IJCAI*, 2018.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *ACL*, 2016.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, 2016.

Travis Martin, Jake M Hofman, Amit Sharma, Ashton Anderson, and Duncan J Watts. Exploring limits to prediction in complex social systems. In *WWW*, 2016.

Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *CIKM*, 2016.

George Mohler et al. Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics*, 2013.

Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, 1988.

Marian-Andrei Rizoiu, Lexing Xie, Scott Sanner, Manuel Cebrian, Honglin Yu, and Pascal Van Hentenryck. Expecting to be hip: Hawkes intensity processes for social media popularity. In *WWW*, 2017.

Marian-Andrei Rizoiu, Swapnil Mishra, Quyu Kong, Mark Carman, and Lexing Xie. Sir-hawkes: on the relationship between epidemic models and hawkes point processes. In *WWW*, 2018.

Bidisha Samanta, Abir De, Abhijnan Chakraborty, and Niloy Ganguly. LMPP: A Large Margin Point Process Combining Reinforcement and Competition for Modeling Hashtag Popularity. In *IJCAI*, 2017.

supplement. Appendix: Exploiting Uncertainty in Popularity Prediction of Information Diffusion Cascades Using Self-exciting Point Processes, 2020.

Remco Van der Hofstad and Michael Keane. An elementary proof of the hitting time theorem. *The American Mathematical Monthly*, 2008.

Siqi Wu, Marian-Andrei Rizoiu, and Lexing Xie. Beyond views: Measuring and predicting engagement in online videos. In *ICWSM*, 2018.

Hongteng Xu and Hongyuan Zha. A dirichlet mixture model of hawkes processes for event sequence clustering. In *NIPS*, 2017.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *ICML*, 2016.

Junchi Yan, Xin Liu, Liangliang Shi, Changsheng Li, and Hongyuan Zha. Improving Maximum Likelihood Estimation of Temporal Point Process via Discriminative and Adversarial Learning. In *IJCAI*, 2018.

Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *ICML*, 2013.

Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, 2013.

Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD*, 2015.

Accompanying the submission *Exploiting Uncertainty in Popularity Prediction of Information Diffusion Cascades Using Self-exciting Point Processes*.

A Borel mixture model for branching factors

As the final cascade size distribution of Hawkes processes is only determined by the branching factor (Section 2), i.e. the Borel distribution, we are able to model sizes of a group of cascades as a Borel mixture distribution. Specifically, given a filtered cascade set $\mathcal{H}^b$ and a cluster number K, we aim to find for each Borel distribution, a mixture probability $p_k = \mathbb{P}[K = k]$ and a branching factor n_k^*. We denote this parameter set as $\Theta_{BMM} = \{p_1, \ldots, p_k, n_1^*, \ldots, n_k^*\}$. The parameters are estimated via the EM algorithm following follows [Tomasi, 2004]. The log likelihood function is

$$L_{BMM} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \sum_{k=1}^{K} p_k \frac{\mathbb{B}(N_i \mid n_k^*)}{\sum_{N_j \in N^*} \mathbb{B}(N_j \mid n_k^*)} \tag{14}$$

For simplicity, let $q(k, N_i) = p_k \frac{\mathbb{B}(N_i|n_k^*)}{\sum_{N_j \in N^*} \mathbb{B}(N_j|n_k^*)}$. We first introduce the probability of $\mathcal{H}_i$ being a member of k:

$$p(k \mid N_i) = \frac{q(k, N_i)}{\sum_{j=1}^{K} q(j, N_i)} \tag{15}$$

By employing Jensen's inequality, we get

$$L_{BMM} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \sum_{k=1}^{K} q(k, N_i) \tag{16}$$

$$= \sum_{\mathcal{H}_i \in \mathcal{H}^b} \log \sum_{k=1}^{K} p(k \mid N_i) \frac{q(k, N_i)}{p(k \mid N_i)} \tag{17}$$

$$\geq \sum_{\mathcal{H}_i \in \mathcal{H}^b} \sum_{k=1}^{K} p(k \mid N_i) \log \frac{q(k, N_i)}{p(k \mid N_i)} \tag{18}$$

Optimizing Eq. (18) is equivalent to optimizing the following Q_{BMM} function

$$Q_{BMM} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \sum_{k=1}^{K} p(k \mid N_i) \log q(k, N_i) \tag{19}$$

At the *Maximization* step, the parameters are updated by maximizing Q_{BMM}.

- For updating n_k^*, we take the derivative of Q_{BMM} w.r.t. n_k^*

$$\frac{\partial Q_{BMM}}{\partial n_k^*} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} \frac{\partial \sum_{k=1}^{K} p(k \mid N_i) \log q(k, N_i)}{\partial n_k^*} \tag{20}$$

$$= \sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) \frac{\partial \log q(k, N_i)}{\partial n_k^*} \tag{21}$$

$$= \sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) \frac{\partial}{\partial n_k^*} \left[\log p_k \frac{\mathbb{B}(N_i \mid n_k^*)}{\sum_{N_j \in N^*} \mathbb{B}(N_j \mid n_k^*)} \right] \tag{22}$$

$$= \sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) \left[\frac{\partial}{\partial n_k^*} \log \mathbb{B}(N_i \mid n_k^*) - \frac{\partial}{\partial n_k^*} \log \sum_{N_j \in N^*} \mathbb{B}(N_j \mid n_k^*) \right] \tag{23}$$

$$= \sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) \left[\frac{\frac{\partial}{\partial n_k^*} \mathbb{B}(N_i \mid n_k^*)}{\mathbb{B}(N_i \mid n_k^*)} - \frac{\sum_{N_j \in N^*} \frac{\partial}{\partial n_k^*} \mathbb{B}(N_j \mid n_k^*)}{\sum_{N_j \in N^*} \mathbb{B}(N_j \mid n_k^*)} \right] \tag{24}$$

we note that $\frac{\partial \mathbb{B}(N_i|n_k^*)}{\partial n_k^*}$ has a special solution

$$\frac{\partial \mathbb{B}(N_i \mid n_k^*)}{\partial n_k^*} = \frac{\partial}{\partial n_k^*}\left[\frac{(N_i n_k^*)^{N_i-1} e^{-N_i n_k^*}}{N_i!}\right] \tag{25}$$

$$= \frac{N_i(N_i-1)(N_i n_k^*)^{N_i-2} e^{-N_i n_k^*} - N_i(N_i n_k^*)^{N_i-1} e^{-N_i n_k^*}}{N_i!} \tag{26}$$

$$= \frac{\frac{N_i-1}{n_k^*}(N_i n_k^*)^{N_i-1} e^{-N_i n_k^*} - N_i(N_i n_k^*)^{N_i-1} e^{-N_i n_k^*}}{N_i!} \tag{27}$$

$$= \frac{N_i - N_i n_k^* - 1}{n_k^*}\mathbb{B}(N_i \mid n_k^*) \tag{28}$$

Plugging this result back to Eq. (24)

$$\frac{\partial Q_{BMM}}{\partial n_k^*} = \sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i)\left[\frac{N_i - N_i n_k^* - 1}{n_k^*} - \frac{\sum_{N_j \in N^*} \frac{N_j - N_j n_k^* - 1}{n_k^*}\mathbb{B}(N_j \mid n_k^*)}{\sum_{N_j \in N^*}\mathbb{B}(N_j \mid n_k^*)}\right] \tag{29}$$

Let the derivative be 0 will lead to the equation

$$\sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) \sum_{N_j \in N^*} (N_i - N_j)(1 - n_k^*)\mathbb{B}(N_j \mid n_k^*) = 0 \tag{30}$$

Although there is no clean analytical solution available to this equation, numerical roots can still be efficiently found within $n_k^* \in (0, 1]$ which give the optimal n_k^*, i.e., $(n_k^*)^{new}$. Specifically, we note that when there is no filtering imposed on $\mathcal{H}$, an analytical solution exists,

$$(n_k^*)^{new} = \frac{\sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i)(N_i - 1)}{\sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i) N_i} \tag{31}$$

- Updating p_k shares same derivation steps from [Tomasi, 2004]

$$p_k^{news} = \frac{\sum_{\mathcal{H}_i \in \mathcal{H}^b} p(k \mid N_i)}{\mid \mathcal{H}^b \mid} \tag{32}$$

Because final sizes of Hawkes processes are highly skewed towards small sizes, the estimation complexity can be reduced by counting the number of presences of various cascade sizes in $\mathcal{H}^b$, i.e., obtaining a set $C' = \{(c_i, N_i)\}$ where there are c_i cascades with size N_i. The summation over $\mathcal{H}^b$ can be then replaced by this set for efficiency.

Appendix References

Carlo Tomasi. Estimating gaussian mixture densities with em–a tutorial. *Duke University*, 2004.