Analyzing Topic Transitions in Text-based Social Cascades using Dual-Network Hawkes Process

Srikanta Bedathur[1], Indrajit Bhattacharya[2], Jayesh Choudhari[3], and Anirban Dasgupta[4]

[1] Indian Institute of Technology Delhi, India
srikanta.bedathur@gmail.com
[2] TCS Research and Innovation Labs, Kolkata, India
indrajitb@gmail.com
[3] University of Warwick, UK
choudhari.jayesh@alumni.iitgn.ac.in
[4] Indian Institute of Technology Gandhinagar, India
anirbandg@iitgn.ac.in

Abstract. We address the problem of modeling bursty diffusion of text-based events over a social network of user nodes. The purpose is to recover, disentangle and analyze overlapping social conversations from the perspective of user-topic preferences, user-user connection strengths and, importantly, topic transitions. For this, we propose a Dual-Network Hawkes Process (DNHP), which executes over a graph whose nodes are user-topic pairs, and closeness of nodes is captured using topic-topic, user-user, and user-topic interactions. No existing Hawkes Process model captures such multiple interactions simultaneously. Additionally, unlike existing Hawkes Process based models, where event times are generated first, and event topics are conditioned on the event times, the DNHP is more faithful to the underlying social process by making the event times depend on interacting (user, topic) pairs. We develop a Gibbs sampling algorithm for estimating the three network parameters that allows evidence to flow between the parameter spaces. Using experiments over large real collection of tweets by US politicians, we show that the DNHP generalizes better than state of the art models, and also provides interesting insights about user and topic transitions.

1 Introduction

We address the problem of modeling text-based information cascades, generated over a social network. Observed data on social media is a tangle of multiple overlapping conversations, each propagating from users to their connections, with the rate depending on connection strengths between the users and the conversation topics. The individual conversations, their paths and topics are not directly observed and needs to be recovered. Additionally, individual conversations involve topic shifts, according to the preferences of the users [1]. Our goal is to analyze the user connection strengths, their topic preferences, and the topic-transition patterns from such social conversations.

There exists a number of models that uses a variety of Hawkes Processes to model such cascades [9,8,1]. None of these satisfactorily capture user-user, user-topic and topic-topic interactions simultaneously. Additionally, in these models, the content does not influence the response rate. This is a significant disconnect with the underlying social process, where the rate of response for a user depends on the user and topic of the 'parent' post, as well as the (possibly different) topic that triggers for the responding user. As a result, two related and important questions are yet unexplored– (1) *how to decompose the overall responsiveness for a pair of users and a pair of topics*, and (2) *how to incorporate the influence of topics on the event rate?*. For example, in the US context, our model should be able to capture a higher response rate for a user passionate about healthcare engaging with another passionate about politics, than for the same user engaging with another talking about gun violence.

In this paper, we address these two issues by extending the Network Hawkes Process [9] which executes over a one-dimensional network over users, to propose a *Dual-Network Hawkes Process* (DNHP) which unfolds over a two-dimensional space of user-topic pairs. Individual events now trigger for a user-topic pair. Each such event spawns a new Poisson process for every other user-topic pair in the neighborhood, whose rate is determined by the two (user, topic) pairs. For tractability and generalization, we decompose this 4-dimensional interaction into *three* interaction matrices. These represent the connection strengths between (a) the pair of users, (b) the pair of topics, and (c) the responding user-topic pair. This decomposition leads to significant parameter sharing between individual point processes. Thus in addition to being closer to the generation of real-life topical information cascades, the Multi-Network Hawkes Process promises significantly better generalization based on limited training data via parameter sharing.

Using the model, we address the task of recovering the user-user, user-topic and topic-topic connection strengths, along with recovering the latent topic and parent (or trigger) event for each event. A significant challenge for parameter estimation is that the user-user and topic-topic weights are intrinsically coupled in our model and cannot be integrated out analytically. We address the coupling issue by showing that the posterior distribution of the user-user (topic-topic) weights is conditionally Gamma distributed given the topic-topic (user-user) weights. Based on the conditional distributions, we propose a Gibbs sampling based inference algorithm for these tasks. In our inference algorithm, the update equations for the user-user and topic-topic weights become coupled, thereby allowing the flow of evidence between them.

We perform extensive experiments over a large real collection of tweets by US politicians. We show that by being more faithful to the underlying process, or model generalizes much better over held-out tweets compared to state of the art baselines. Further, we report revealing insights involving users groups, topics and their interactions, demonstrating the analytical abilities of our model.

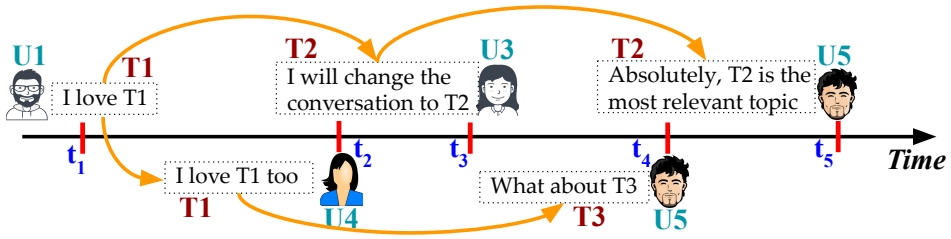

Fig. 1. Illustration of DNHP event generation process

2 Dual-Network Hawkes Process

We consider text based cascades generated by a set of users $U = \{1, 2, \ldots, n\}$, connected by edges $\mathcal{E}$. Each edge (u, v) has weight $W_{u,v}$, which indicates the extent of the influence of user u on user v. We assume that the unweighted graph over the users is known or observed but the weights $W_{u,v}$ are not. Let $E = \{e\}$ be the set of all events, which may be tweets or social media posts, created by the users U. The example in Fig.2 shows a toy collection of 5 events. Each event e is defined as a tuple $e = (t_e, c_e, d_e, \eta_e, z_e)$, where, t_e is the time at which event was created, $c_e \in U$, is the user who created this event. We assume that each event is triggered by a unique parent event. Let $z_e \in E$ indicate the parent event of e. Events which are triggered by some other event are termed as *diffusion* events, and events that happen on their own are termed as *spontaneous* events. In the example, the first event posted by $U1$ at time t_1 is a spontaneous event with no parent, while the others are diffusion events. For the other events, parents are indicated by arrows. The second event posted by user $U4$ at time t_2, and third event posted by user $U3$ at time t_3 have the first event as their parent, the fourth event posted by user $U5$ at time t_4 has the second event as parent and so on. Notice that the diffusion events leads to the formation of cascades, and the spontaneous events represent the start of the cascade. A cascade starts with a spontaneous event, which triggers diffusion events, which trigger further diffusion events, leading to a cascade. In this example, the event at time t_1 is the spontaneous event, and the others are diffusion events.

We use d_e to denote the textual content associated with event e. Let $\mathcal{V}$ denote the vocabulary of the textual content of all events, i.e. $d_e \subset \mathcal{V}$. We assume that d_e corresponds to a topic η_e. Following [1,2] and unlike [8], we model η_e as discrete variable, indexing into a component of a mixture model, which is more appropriate for short texts. Accordingly, $\eta_e \in [K]$, where K denotes the number of topics. In our example, we have three topics. The first and second events are on topic $T1$, the third and fifth on topic $T2$, and the fourth on topic $T3$. For any event e, all the events e' such that $t_{e'} < t_e$ and $c_{e'} \in \mathcal{N}(c_e)$, where $\mathcal{N}(c_e)$ denote the set of neighbors of c_e in the user-user graph, are the set of candidate parent event. Additionally, as similar to that of the HMHP model [1],

which posits the existence of the a topic-topic interaction matrix, here as well we consider the existence of the topic-topic graph over the set of nodes $[K]$. The topic-topic graph as well as the topic-topic interaction strengths $\mathcal{T}_{k,k'}$ for every pair of topics $k, k' \in [K]$ are unobserved. The topic-topic interaction strength between a pair of topics represents how quickly the conversations transit over these topics.

Hawkes processes[7,11] have been variously used to model such cascade events [1,8,9]. In all of these models, a Hawkes Process executes over a network of user nodes. Specifically, each event on a user node u triggers a Poisson Process on all neighboring nodes v, with a rate that is parameterized by the connection strength between u and v. In essence, the topics do not play a role in the Hawkes process itself. We deviate fundamentally from this by defining a super-graph $\mathcal{G}$, where each super-node corresponds to a user-topic pair $(u, k),\ u \in U,\ k \in [K]$. The Hawkes Process now executes on this super-graph. This is also illustrated in Fig. 1. Specifically, each event happens on a super-node (u, v), and spawns a Poisson Process on each 'neighboring' super-node (v, k'). In the example, according to the super-node representation, the first event happens at $(U1, T1)$, the second at $(U4, T1)$ and so on. Each event spans events on each neighboring super-node. We define two super-nodes to be neighbors is there corresponding users are neighbors in the social graph. The graph in Fig.2(a) shows the social graph for our example. As a result of this, the first event at $(U1, T1)$ will trigger Poisson Processes at super-nodes with users $U2$ (i.e. $(U2, T1), (U2, T2), (U2, T3)$), $U3$ (i.e. $(U3, T1), (U3, T2), (U3, T3)$), and $U4$ (i.e. $(U4, T1), (U4, T2), (U4, T3)$). The rate of each Poisson Process, for example that triggered $(U4, T1)$, is determined by the 'closeness' of the super-node pair. We discuss this in more detail later in this section. We call this the Dual-Network Hawkes Process (DNHP), because the process executes on a two-dimensional network, unlike those based on the Network Hawkes Process which have a one-dimensional network. Once the DNHP has generated events until some time horizon T, the textual content d_e of each event at super-node (c_e, η_e) is generated independently according to the distribution associated with its topic η_e. We first describe the generation of the super-node (c_e, η_e) and time t_e of each event, and then that of the textual content d_e.

2.1 Modeling Time and Topic

In this phase, the time t_e, user c_e, parent z_e, and topic z_e) for each event is generated using the Multivariate Hawkes Process (MHP) on graph $\mathcal{G}$. We follow the general process of existing models [12,9,8,1], but replace user nodes with user-topic super-nodes. In the following, when we refer to a pair (u, k), we will assume $u \in U$, and $k \in [K]$.

Let $\mathcal{H}_{t-}$ denote the set of all events generated prior to time t. Then, following the definition of the Hawkes Process, the intensity function $\lambda_{(v,k)}(t)$ for super-node (v, k) is given by the superposition of the base intensity $\mu_{(v,k)}(t)$ of (v, k) and the impulse responses of historical events $e \in \mathcal{H}_{t-}$ at super-nodes (c_e, η_e) at time t_e: $\lambda_{(v,k)}(t) = \mu_{(v,k)}(t) + \sum_{e \in \mathcal{H}_{t-}} h_{(c_e,\eta_e),(v,k)}\,(t - t_e)$. The base intensity for

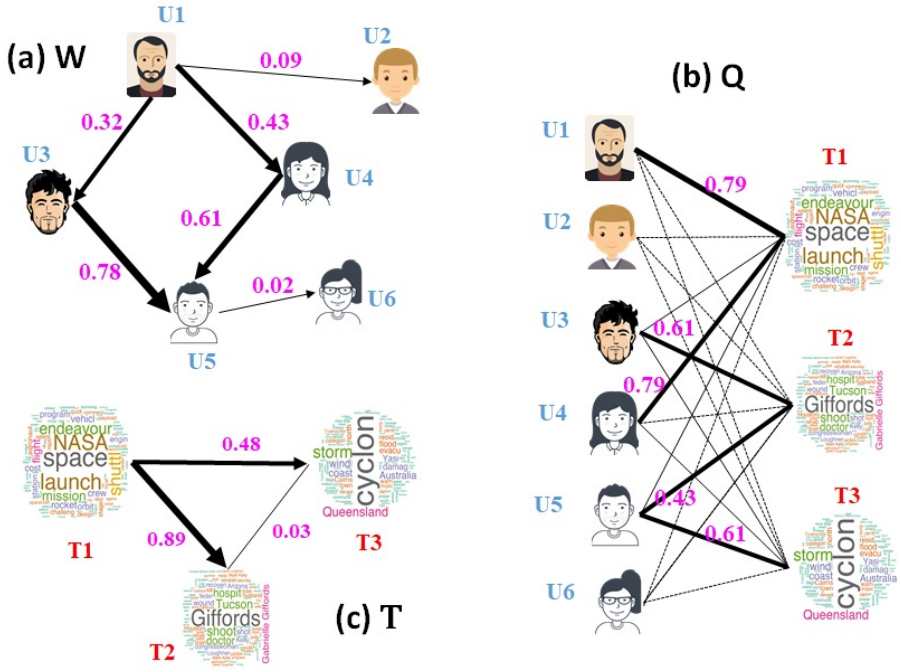

Fig. 2. Illustration of DNHP Model Parameters

node (v, k) is defined as $\mu_{(v,k)}(t) = \mu_v(t) \times \mu_k(t)$, where, $\mu_v(t)$ is base intensity associated with user v, and $\mu_k(t)$ is the base intensity for topic k.

In the context of super-nodes, the parameterization of the impulse response $h_{(u,k),(v,k')}$ becomes a challenge. The naive 4-dimensional parameterization is unlikely to have enough data for confident estimation, while complete factorization with four 1-dimensional parameters is overly biased. We propose its decomposition into three factors:

$$h_{(u,k),(v,k')}(\Delta t) = W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}f(\Delta t) \tag{1}$$

These three factors form the parameters of our model. Here, $W_{u,v}$ captures user-user preference, $\mathcal{T}_{k,k'}$ captures topic-topic interaction, and $Q_{u,k}$ user-topic preference. We believe that this captures the most important interactions in the data, while providing generalization ability.

Fig. 2 illustrates this parameterization. Fig. 2(a) shows parameter $W_{u,v}$. User pairs $(U3, U5)$ have the strongest connection, indicating the U5 responds to $U3$ with the quickest rate, followed by $(U4, U5)$, etc. Note that this parameterization is directional. Fig. 2(b) shows parameter $Q_{,}uk$. Here, $(U1, T1)$ has the strongest connection, indicating that user U1 posts on topic $T1$ with the quickest rate, followed by the others. Fig. 2(c) shows parameter $\mathcal{T}_{,}kk'$. This shows that topic

transitions happen from $T1$ to $T2$ with the quickest rate, while those from $T2$ to $T3$ happen much slower. Note that this parameter is also directional. The overall rate of the process induced at $(U2, T1)$ by the event at $(U1, T1)$ is determined by the product of the factors $W_{U1,U2}$, $Q_{U2,T1}$ and $\mathcal{T}_{T1,T1}$.

Finally, $f(\Delta t)$ is the time-kernel term. We model the time-kernel term $f(\Delta t)$ using a simple exponential function i.e. $\exp(\Delta t)$.

To generate events with the intensity function $\lambda_{(u,k)}(t)$, we follow the level-wise generation of events [12]. Let, Π_0 be the level 0 events which are generated with the base intensity of the nodes $(v, k) \in \mathcal{G}$, i.e. $\mu_{(u,k)}(t)$. In our example, this generates the first event $(U1, T1)$ with time-stamp t_1. Then, the events at level $\ell > 0$ are at each super-node (u', k') are generated as per the following non-homogeneous Poisson process:

$$\Pi_\ell \sim Poisson\left(\sum_{(t_e,(c_e,\eta_e),z_e)\in\Pi_{\ell-1}} h_{(c_e,\eta),(u',k')}\left(t - t_e\right)\right) \tag{2}$$

Influencing happens only on neighboring super-nodes (u', k') for $(c_e, \eta_e), e \in \Pi_{\ell-1}$. Recall that two super-nodes (u, k) and (u', k') are neighbors if the corresponding users u and u' are neighbors in the social network. Imagine our example set of events in Fig.1 being generated using the parameterization in Fig.2 according to the level-wise generation process. Here, $\Pi_0 = \{(U1, T1, t1)\}$, $\Pi_1 = \{(U4, T1, t2), (U3, T2, t3)\}$, and $\Pi_2 = \{(U5, T3, t4), (U5, T2, t5)\}$.

We would like to highlight the reader that in this work we model the time-kernel term $f(\Delta t)$ using a simple exponential function i.e. $\exp(\Delta t)$. We understand that there has been a number of recent works that model the time-kernel efficiently ([8], [2], [9]), but the main aim of this work is not that.

2.2 Modeling Documents

Once the events are generated on super-nodes, generation of each document d_e for event e happens conditioned only on the topic η_e of the super-node, using a distribution over words $\boldsymbol{\zeta}_{\eta_e}$ specific to topic η_e. In our example process, each of the three topics T1, T2 and T3 have their corresponding distribution over words, denoted at $\boldsymbol{\zeta}_{T1}$, $\boldsymbol{\zeta}_{T2}$ and $\boldsymbol{\zeta}_{T3}$ respectively. The words in the first and second events are generated i.i.d. from $\boldsymbol{\zeta}_{T1}$, those in the third event from $\boldsymbol{\zeta}_{T2}$, and so on.

The complete generative process for the DNHP is described using a pseudo-code in Algorithm 1.

2.3 Stability of DNHP

One of the important properties of the Hawkes processes which makes it a perfect fit for cascades of social media events is the *mutually exciting* property. Each historical event adds a non-negative impulse response to the intensity function, and thus increases the likelihood of the future events. Because of this recurrent nature of the Hawkes processes, we need to ensure that the generative process

Algorithm 1 DNHP Generative Model

for all $u \in U$ **do**
  for all $v \in \mathcal{N}_u$ **do** ▷ User-User Influence
    Sample $W_{u,v} \sim Gamma(\alpha'_1, \beta'_2)$
  for all $k \in [K]$ **do** ▷ User-Topic Preference
    Sample $Q_{u,k} \sim Gamma(\alpha'_3, \beta'_3)$
for all $k \in [K]$ **do**
  Sample $\boldsymbol{\zeta}_k \sim Dirichlet_K(\alpha)$ ▷ Topic-word Distribution
  for all $k' \in [K]$ **do** ▷ Topic-Topic Interaction
    Sample $\mathcal{T}_{k,k'} \sim Gamma(\alpha'_2, \beta'_2)$
Generate $(t_e, (c_e, \eta_e), z_e)$ for each event as described in section 2.1 (*under Modeling Time and Topic*)
for all $e \in E$ **do**
  Sample $N_{d_e} \sim Poisson(\lambda)$ ▷ #words to sample
  Sample N_{d_e} words from $\boldsymbol{\zeta}_{\eta_e}$

does not lead to a generation of infinite number of events within a finite time horizon. Here, we claim that the DNHP as defined above is stable in this sense.

Lemma 1. *To ensure that DNHP does not generate infinite number of events it is sufficient to ensure that* $\lambda_{\max}(A \odot W) < 1$, $\lambda_{\max}(B \odot \mathcal{T}) < 1$, *and* $\lambda_{\max}(Q) < 1$ *where,* λ_{max} *denotes highest eigenvalue,* A *and* B *are the adjacency matrices for the user-user and topic-topic graphs respectively, and* W, $\mathcal{T}$, *and* Q *define the user-user influence, topic-topic interaction, and user-topic preference matrices respectively.*

The proof for this is on the similar lines as that of the Network Hawkes process [9].

3 Approximate Posterior Inference

The latent variables associated with each event in case of DNHP are the parent of the event (z_e), and the topic of the event (η_e). The variable z_e for each event e can be either 0 indicating a spontaneous event or some event e' in the history of event. Along with these latent variables, the model parameters, namely, the user-user influence matrix W, the user-topic preference matrix $Q_{u,k}$, the topic-topic interaction matrix $\boldsymbol{\mathcal{T}}$, and base rates for each user and each topic need to be estimated.

As the exact inference is intractable, we perform inference using the Gibbs sampling algorithm. The topic-topic interaction strengths $\mathcal{T}_{k,k'}$ are tightly coupled with the user-user influence $W_{u,v}$ terms in the likelihood, and as a result cannot be integrated out analytically. However, we can show that the joint distribution is analytically troublesome, the conditional distributions for the parameters given the other parameters have a nicer form. The posterior distributions for each of the three parameters $W_{u,v}$, $Q_{v,k'}$, and $\mathcal{T}_{k,k'}$ are *Gamma* distributed conditioned

on the other two. Additionally, the posterior distributions for the topic (η_e) and parent (z_e) for each event are categorical conditioned on the parameters. This naturally leads to a sampling based iterative inference algorithm. Specifically, in each step, we use Gibbs sampling to sample the individual topic assignments (η_e), parent assignments (z_e), the user-user influence $W_{u,v}$, the topic-topic interaction strengths $\mathcal{T}_{k,k'}$, user-base rate μ_v for each user and topic-base rate μ_k for each topic from their conditional distributions, given current assignments to all other variables. This iterative algorithm is continued until convergence. Following are the conditional distributions for the different latent variables, which are used in each step of the Gibbs sampling algorithm.

3.1 Parent Inference

The conditional probability of event $e' = (t_{e'}, (u, k'), z_{e'})$ being a parent of an event $e = (t_e, (v, k), z_e)$ is given as:

$$P(z_e = e'|W, \mathcal{T}, Q) \propto \exp(-W_{u,v}\mathcal{T}_{k',k}Q_{v,k})W_{u,v}\mathcal{T}_{k',k}f(\Delta t_e) \tag{3}$$

Note that this depends on the association strengths between the corresponding users u and v, the corresponding topics k and k', and the time lag Δt_e between the events. On the other hand, the conditional probability of an event being spontaneous is given as:

$$P(z_e = e|\mu_v, \mu_k, T) \propto exp(-T\mu_v\mu_k)\mu_v\mu_k \tag{4}$$

Note that this depends on the base rates of the user μ_v and the topic μ_k.

3.2 Topic Assignment

The conditional probability of assigning a topic k to an event e by user v with the parent event $e' = (t_{e'}, (u, k'), z_{e'})$ is given as:

$$\begin{aligned} P(\eta_e = k|\boldsymbol{z}, W, \mathcal{T}, \boldsymbol{\alpha}) \propto\ & \frac{\prod_{w \in d_e} \prod_{i=0}^{N_{d_e}^{w}-1} \alpha_w + \mathfrak{T}_{k,w}^{\neg e} + i}{\prod_{i=0}^{N_{d_e}-1} \sum_{w \in \mathscr{V}} \alpha_w + \mathfrak{T}_{k}^{\neg e} + i} \times exp\left(-W_{u,v}\mathcal{T}_{k',k}Q_{v,k}\right)\mathcal{T}_{k',k}Q_{v,k} \\ & \times \prod_{\substack{h \in E \\ z_h = e}} \left[exp\left(-W_{v,c_h}\mathcal{T}_{k,\eta_h}Q_{c_h,\eta_h}\right)\mathcal{T}_{k,\eta_h}\right] \end{aligned} \tag{5}$$

Here, $\mathfrak{T}$ is the count matrix of dimension $K \times \mathscr{V}$ storing the count of each word for each topic. $\mathfrak{T}_{k,w}^{\neg e}$ indicates the count of word w in topic k excluding the counts from event e. Note that the first term considers the likelihood of the document d_e for the event coming from topic k, the second considers the probability of user v posting on topic k following parent event topic k', and the third term considers the various child event of this event posting on their own topics following topic k. When event e is a spontaneous event, the conditional probability has a similar form, with only the second term changing to $exp(-\mu_v\mu_k T)\mu_v\mu_k$, indicating the probability of generation of an spontaneous event with topic k.

3.3 User-User Influence ($W_{u,v}$)

The conditional probability of influence of user u on v is given as:

$$P(W_{u,v}|\mathcal{T},\mathbf{Q},\boldsymbol{z}) \propto P(W_{u,v}|\alpha_1',\beta_1') \times C' \times W_{u,v}^{\alpha_1} \exp(-\beta_1 W_{u,v}) \tag{6}$$

The first term results from a Gamma prior on $W_{u,v}$, and C' being a constant independent of $W_{u,v}$. Also, $N_{u,v}$ denotes number of times an event at u triggered event at v, and $N_{u,k}$ denotes the number of events of topic k by user u. We recognize this form as that of the Gamma distribution. Specifically, $W_{u,v}$ is $Gamma\left(\alpha^{(W)},\beta^{(W)}\right)$ distributed with parameters $\alpha^{(W)} = \alpha_1 + \alpha_1'$, and $\beta^{(W)} = \beta_1 + \beta_1'$, where,

$$\alpha_1 = \sum_k \sum_{k'} N_{(u,k),(v,k')} = N_{u,v} \qquad \beta_1 = \sum_k N_{(u,k)} \sum_{k'} (\mathcal{T}_{k,k'} Q_{v,k'})$$
$$C' = \prod_k \prod_{k'} (\mathcal{T}_{k,k'} Q_{v,k'})^{N_{(u,k)(v,k')}}$$

Note the dependence of β_1 on the current values of all $\mathcal{T}_{k,k'}$ and all $Q_{v,k'}$.

3.4 Topic-Topic Interaction ($\mathcal{T}_{k,k'}$)

The conditional probability of interaction between topics k and k' is given as:

$$P(\mathcal{T}_{k,k'}|W,Q,\mathbf{z}) \propto P(\mathcal{T}_{k,k'}|\alpha_2',\beta_2') \times C' \times \mathcal{T}_{k,k'}^{\alpha_2} \exp(-\beta_2 \mathcal{T}_{k,k'}) \tag{7}$$

Again, the first term is a Gamma prior on $\mathcal{T}$, and the term C' which is a constant independent of $\mathcal{T}$. $N_{k,k'}$ is the number of times an event with topic k triggered an event with topic k'. Therefore, $\mathcal{T}_{k,k'}$ is again $Gamma(\alpha^{(\mathcal{T})},\beta^{(\mathcal{T})})$ distributed with parameters $\alpha^{(\mathcal{T})} = \alpha_2 + \alpha_2'$, and $\beta^{(\mathcal{T})} = \beta_2 + \beta_2'$, where,

$$\alpha_2 = \sum_u \sum_v N_{(u,k),(v,k')} = N_{k,k'} \qquad \beta_2 = \sum_u N_{(u,k)} \sum_v (W_{u,v} Q_{v,k'})$$
$$C' = \prod_u \prod_v (W_{u,v} Q_{v,k'})^{N_{(u,k)(v,k')}}$$

Again, note the dependence of β_2 on the current values of all $W_{u,v}$ and all $Q_{v,k'}$.

3.5 User-Topic Preference ($Q_{v,k'}$)

The conditional probability of user v's preference towards topic k' is given as:

$$P(Q_{v,k'}|W,\mathcal{T},\mathbf{z}) \propto P(Q_{v,k'}|\alpha_3',\beta_3') \times C' Q_{v,k'}^{\alpha_3} \exp(-\beta_3 Q_{v,k'}) \tag{8}$$

Therefore, $Q_{v,k'}$ is $Gamma(\alpha^{(Q)},\beta^{(Q)})$ distributed with parameters $\alpha^{(Q)} = \alpha_3 + \alpha_3'$, and $\beta^{(Q)} = \beta_3 + \beta_3'$. Here,

$$\alpha_3 = \sum_{u \in \mathcal{N}(v)} \sum_k N_{(u,k)(v,k')} \qquad \beta_3 = \sum_{u \in \mathcal{N}(v)} \sum_k N_{(u,k)} (W_{u,v} \mathcal{T}_{k,k'})$$
$$C' = \prod_u \prod_k (W_{u,v} \mathcal{T}_{k,k'})^{N_{(u,k)(v,k')}}$$

This again depends on the current values of all ($W_{u,v}$ and all $\mathcal{T}_{k,k'}$).

3.6 Base Rate Inference

The estimates for user base rate $\mu_v \ \forall v \in U$ and topic base rates $\mu_k \ \forall k \in K$ are given as:

$$\mu_v = \frac{N_v^{(spon)}}{T \sum\limits_{k \in K} \mu_k} \qquad \mu_k = \frac{N_k^{(spon)}}{T \sum\limits_{v \in V} \mu_v} \tag{9}$$

where, $N_v^{(spon)}$ is the number of spontaneous events by user v, $N_k^{(spon)}$ is the number of spontaneous events that have topic k, and T is total time for which the events are observed.

The complete inference algorithm for DNHP is given below.

Algorithm 2 Approximate Inference - Gibbs Sampling

```
Initialize η_e, z_e for all events
for i ∈ {1 ... maxIter} do
    for all e ∈ E do
        Sample η_e using Eq. 5                          ▷ Topic
        Sample z_e using Eq. 3                          ▷ Parent
    for all (u, v) ∈ E do
        Sample W_{u,v} using Eq. 6                      ▷ User-User influence
    for all (k, k') ∈ ([K] × [K]) do
        Sample T_{k,k'} using Eq. 7                     ▷ Topic-Topic interaction
    for all (v, k') ∈ (U × K) do
        Sample Q_{v,k'} using Eq. 8                     ▷ User-Topic preference
    for all u ∈ U do
        Estimate μ_u                                    ▷ User base intensity
    for all k ∈ [K] do
        Estimate μ_k                                    ▷ Topic base intensity
```

The number of parameters to infer per iteration is $O(|E| + |E| + |\mathcal{E}| + (K^2) + (|U| \times K))$, where $|E|$ is the total number of events, and $|\mathcal{E}|$ is the number of edges in user-user graph, and K is the number of topics. This is of the same order as that of competing models [1] that consider topic transitions. The additional parameters in DNHP correspond to the matrices $\mathcal{T}$ and Q respectively. However, inference of these additional terms serves an important purpose in DNHP. Note the interdependence between the user-user influence and the topic-topic interaction in the inference equations - one directly influences the other. In fact, different user-user weights influence each other via topic-topic weights and user-topic preferences. This results in more efficient sharing of evidence across different user-user, user-topic and topic-topic weights. In contrast, user-user and topic-topic weights in HMHP and user-user weights in Network Hawkes model are conditionally independent given the event parents, and as a result cannot reinforce each other.

4 Experiments and Results

In this section we validate the strengths of DNHP empirically. We first describe the models with which we compare the performance of DNHP. We then describe the datasets and define the tasks for these models to address. All the experiments are performed on a machine with 32 cores, 2.10GHz Intel(R) Xeon(R) E5-2620 CPU, and 256 GB RAM.

4.1 Models Evaluated

We compare the performances of the following models on the datasets described in section 4.2 with respect to the tasks defined in section 4.3.

1. **HMHP**: HMHP [1] is a model for the generation of text-based cascades. HMHP incorporates user temporal dynamics, user-topic preferences, user-user influence, along with topical interactions. However, in `HMHP`, similar to the other models in the literature, for eg. [7], [12], [9] and [8], the generation of time stamp of an event is independent of the topic associated with the event. Additionally, HMHP does not capture user-topic preferences. We evaluate and compare the performance of HMHP with DNHP based on the *generalization* on real data and *reconstruction* performance on synthetic data.
2. **NHWKS & NHLDA**: Network Hawkes [9] jointly models event time stamps and the user-user network strengths. This model infers user-user influence and also the parent event for each event. As opposed to HMHP and DNHP, NHWKS does not model text content. Therefore, to we define a simple extension of NHWKS that additionally generated topic labels and content for events, following up on the NHWKS generative process. Specifically, we use an LDA mixture model, that assigns a topic to each event by sampling from a prior categorical distribution, and then draws the words of that event i.i.d by sampling from the word-distribution specific to that topic. We call this Network Hawkes LDA (NHLDA).
3. **DNHP**: This is our model with Gibbs sampling based inference algorithm. Here, as for HMHP, the topic-topic graph is considered as a complete graph and the weights over all the edges have the same prior.

4.2 Datasets

We evaluate the performance of the above mentioned models on the following two datasets:

1. **Real dataset:** The dataset that we consider here, denoted as `USPol` (US Politics Twitter Dataset), is a set of roughly $370K$ tweets extracted for 151 users who are members of the US Congress[5]. The tweets were extracted in July 2018 using the *Twitter API*. Each tweet in the dataset consists of time stamp, the user-id, and the tweet-text(content). The total vocabulary size

[5] https://bit.ly/2ufvRWR

here is roughly $33k$ after removing rare tokens. Ground truth information about parent events is not available for this dataset. Also, we do not consider retweets explicitly. Note that retweets have same topic as that of the original tweet, and retweets form only a small fraction of the parent-child relations that we intend to infer.

2. **Semi-Synthetic Dataset**: As for the `USPol` dataset the gold standard for topics and parents is not available, we also generate a semi-synthetic dataset using the `DNHP` generative model, with the same user-user graph as `USPol`. This dataset, that we call `SynthUSPol` is generated by first sampling the user-user influence and topic-topic interactions matrices from *Gamma* priors. For all $u \in U$ and $k \in [K]$, $\mu_u = \mu_k = 0.003$. Here, $K = 25$, $|\mathscr{V}| = 5000$, and the topic-word distributions are sampled from $Dirichlet(0.01)$. For each event e, the number of words in document d_e is sampled from a Poisson distribution with mean 10 to mimic tweets. Using this configuration, we generate 3 different samples of roughly $370K$ events each. For this dataset, due to space constraints, we only report parent identification performance in Table 1. Note that the user-user weight estimates depend directly and only on the identified parents.

4.3 Evaluation Tasks and Results

In this section we evaluate the models based on the (A) *Cascade Reconstruction*, and (B) *Generalization* performances.

(A) ***Cascade Reconstruction Performance***: For the parent identification task the evaluation metrics used are the accuracy and the recall. Accuracy is defined as the percentage of events for which the correct parent is identified. And, given a ranked list of the predicted parents for each event, recall is calculated by considering the top 1, 3, 5 and 7 predicted parents.

(B) ***Generalization Performance***: We compare the performance of the models using Log-Likelihood ($\mathcal{LL}$) of the held-out test set. We perform this task on the semi-synthetic dataset `SynthUSPol` and also on the real dataset `USPol`. For each event e in the `Test` set the observed variables are the time t_e, the creator-id c_e, and the words/content d_e, while the parent z_e and the topic η_e are latent.

The calculation of the loglikelihood of the test data involves a significant computational challenge. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the set of events in the `Train` and `Test` sets respectively. As per DNHP, the total log-likelihood $\mathcal{LL}(DNHP)$ of the test set $\mathcal{Y}$ is given as:

$$
\begin{aligned}
\mathcal{LL}(DNHP) &= \sum_{e \in \mathcal{Y}} \left(\log P(t_e, c_e, w_e)\right) \\
&= \sum_{e \in \mathcal{Y}} \sum_{\substack{e' \in E \\ c_e \in \mathcal{N}(c_{e'}) \\ t_{e'} < t_e}} \sum_{\eta_{e'}} \sum_{\eta_e} \left(\exp(-W_{c_{e'},c_e} \mathcal{T}_{\eta_{e'},\eta_e} Q_{c_e,\eta_e}) \times W_{c_{e'},c_e} \mathcal{T}_{\eta_{e'},\eta_e} Q_{c_e,\eta_e} \exp(\Delta t_e)\right) \times P(w_e|\eta_e) + \\
&\qquad \sum_{\eta_e} \exp(-\mu_{c_e} \mu_{\eta_e} T) \mu_{c_e} \mu_{\eta_e} \times P(w_e|\eta_e)
\end{aligned}
\tag{10}
$$

Here, the summations over $e' \in E$, over $\eta_{e'}$, and over η_e are for the marginalization over the candidate set of parents, topic of parent event, and topic of event e respectively.

Similarly, for HMHP it is given as:

$$\begin{aligned}\mathcal{LL}(HMHP) &= \sum_{e\in\mathcal{Y}} \log(P(t_e, c_e, w_e)) \\ &= \sum_{e\in\mathcal{Y}} \sum_{\substack{e'\in E \\ c_e\in\mathcal{N}(c_{e'}) \\ t_{e'}<t_e}} \sum_{\eta_{e'}} \sum_{\eta_e} \big(\exp(-W_{c_{e'},c_e})W_{c_{e'},c_e} \times \exp(\Delta t_e)\mathcal{T}_{\eta_{e'},\eta_e} \times P(w_e|\eta_e)\big) + \\ &\qquad \sum_{\eta_e} \exp(-\mu_{c_e}T)\mu_{c_e}\mathcal{U}_{c_e,\eta_e} \times P(w_e|\eta_e)\end{aligned} \tag{11}$$

As opposed to DNHP, in HMHP, $\mathcal{T}_{\eta_{e'},\eta_e}$ is probability of transition from topic $\eta_{e'}$ to topic η_e, and thus for a fixed $\eta_{e'}$, we have $\sum_k \mathcal{T}_{\eta_{e'},k} = 1$. HMHP also has users topic preference probability $\mathcal{U}_{c_e,\eta_e}$, and for a fixed user c_e, $\sum_k \mathcal{U}_{c_e,k} = 1$.

In general, when the candidate parents are not in the training set, the parent event also has latent variables. Observe the summation over candidate parent's topic $\eta_{e'}$ in Equations 10 and 11. Therefore, calculating $\mathcal{LL}$ for all the events $e \in \mathcal{Y}$ involves recursively enumerating and summing over over all possible test cascades. We avoid this summation by assuming that the parent event for each test event is in the training set, and create our test sets accordingly.

For the semi-synthetic dataset `SynthUSPol`, this is simple, since the dataset is generated according to `DNHP` and we know the actual cascades. We use this level information from the `SynthUSPol` dataset to classify the events into `Train` and `Test` sets. We take the events at a specific level as the `Test` set, and the events at all previous levels as the `Train` set. However, for the real dataset `USPol`, the true cascade structure is unknown. So we use some heuristics to ensure that the events in the `Test` set are very likely to have parents in the `Train` set. We also design controls for the `Test` set size. We process events sequentially. Each event $e \in E$ is added to the `Test` set $\mathcal{Y}$ if and only if at most p_{test} fraction of its candidate parents are already in the `Test` set $\mathcal{Y}$. This ensures that $1 - p_{test}$ fraction of its candidate parents are still in the `Train` set $\mathcal{X}$. Note that increasing (decreasing) p_{test} results in increasing (decreasing) the test set size, and decreasing (increasing) the train set size. To study the effects of increasing training data size without reducing the test size, we use an additional parameter $0 \leq p_{data} \leq 1$ to decide whether to include an event in our experiments at all. Specifically, we first randomly include each event in the dataset with probability p_{data}, and then the `Train` and `Test` split is performed.

4.4 Results

Parent Identification: Table 1 presents the results for this task. Each result presented here is an average over 5 samples of the generated dataset. For both the

Table 1. Parent Identification performance for DMHP and HMHP on Semi-Synthetic Dataset

Parent Identification				
	Accuracy	Recall@1	Recall@3	Recall@5
DNHP	**0.47**	**0.48**	**0.75**	**0.84**
HMHP	0.40	0.40	0.68	0.79

Table 2. Average $\mathcal{LL}$ for Semi-Synthetic data

Average Log-Likelihood of Time & Content		
Test On	**DNHP**	**HMHP**
2^{nd} Last Level	**-58.66**	-59.31
Last Level	**-57.6**	-58.48
Average Log-Likelihood of Time		
Test On	**DNHP**	**NHWKS**
2^{nd} Last Level	**-2.56**	-2.76
Last Level	**-2.32**	-2.48

models, recall improves significantly as we consider more candidates predicted parents. The accuracy and recall @ 1 for the `DNHP` is $\sim 20\%$ better than that of the `HMHP` model. In summary, `DNHP` outperforms the `HMHP` model with respect to the reconstruction performance for the synthetic data.

Generalization Performance: The generalization performance for the models is evaluated on the basis of their ability to estimate the heldout $\mathcal{LL}$. This task is addressed on both semi-synthetic (`SynthUSPol`) and the real (`USPol`) dataset.

1. `SynthUSPol` **Dataset:** Table 2 presents the heldout $\mathcal{LL}$ of time and content for `DNHP` and `HMHP`, and $\mathcal{LL}$ of time by for `DNHP` and `NHWKS` for the `SynthUSPol` dataset. The results are averaged over 3 independent samples each of size $\sim 370K$ The size of the `Train` set upto 3^{rd} last level and upto 2^{nd} last level is $\sim 170K$ and $\sim 340K$ respectively. In general, with more training data, for both models $\mathcal{LL}$ improves. Overall, `DNHP` outperforms `NHWKS` in explaining the time stamps by benefiting from estimating topic-topic parameters given the text, and in turn using those to better estimate the user-user parameters. In the same way, by better estimating both of these parameters using coupled updates, `DNHP` outperforms `HMHP` in explaining the time stamps and the textual content together.
2. `USPol` **Dataset:** Table 3 presents the $\mathcal{LL}$ of time and content for `DNHP` and `HMHP`, and $\mathcal{LL}$ of time for `DNHP` and `NHWKS`, on the `USPol` dataset for $K = 100$.

Table 3. Average Log-Likelihood of Time+Content and Time with $K = 100$ for the real dataset `USPol`. (The Train(Test) sizes mentioned are approximate)

p_{test} = **0.3**					
	Time + Content			**Time**	
Train(Test)	**DNHP**	**HMHP**	**NHLDA**	**DNHP**	**NHWKS**
$114K(70K)$	**-82.11**	-96.51	-96.72	**-8.03**	-24.46
$177K(100K)$	**-79.09**	-87.32	-87.63	**-7.07**	-16.32
$240K(130K)$	**-77.03**	-80.71	-81.00	**-6.34**	-10.27
p_{test} = **0.5**					
	Time + Content			**Time**	
Train(Test)	**DNHP**	**HMHP**	**NHLDA**	**DNHP**	**NHWKS**
$86K(98K)$	**-83.37**	-96.15	-105.56	**-8.09**	-23.57
$133K(144K)$	**-80.45**	-87.78	-93.18	**-7.21**	-16.02
$179K(190K)$	**-78.27**	-81.96	-85.90	**-6.53**	-10.90

The rows indicate the `Train` and `Test` when the events are selected with probability p_{data} set as 0.5, 0.7, and 1.0 (which is the complete dataset). Then the `Train-Test` split is performed with p_{test} set as 0.3 *and* 0.5, which indicate the maximum fraction of candidate parents for each event in the `Test` set.

Observe that as expected all the models, `DNHP`, `HMHP` and `NHWKS`, get better at estimating the $\mathcal{LL}$ with the increase in the size of the dataset across different values of p_{test}. However, `DNHP` performs better than the competitors by a significant margin, both for time and time+content. A significant point to note is that the gap between `DNHP` and the two baselines `HMHP` and `NHWKS` is larger when the training dataset size is smaller. This agrees with our understanding of parameter sharing leading to better generalization given limited volumes of training data. This demonstrates that `DNHP` has already learned the parameters efficiently with the smaller dataset size, using flow of evidence between the parameters in the update equations.

4.5 Analytical Insight from USPol Dataset

In order to extract analytical insights from the `USPol` dataset, we first fit the model using $K = 100$ topics. Each of these 100 topics were then manually given a topic name by looking at the set of top words in the topic. For ease of understanding, these 100 topics were further manually annotated by one of the following 8 topics– {Politics, Climate, Social, Defence, Guns, Economy, Healthcare, Technology, Guns}[6]. Henceforth we refer to these are the topics.

[6] Open sourced along with the rest of the data

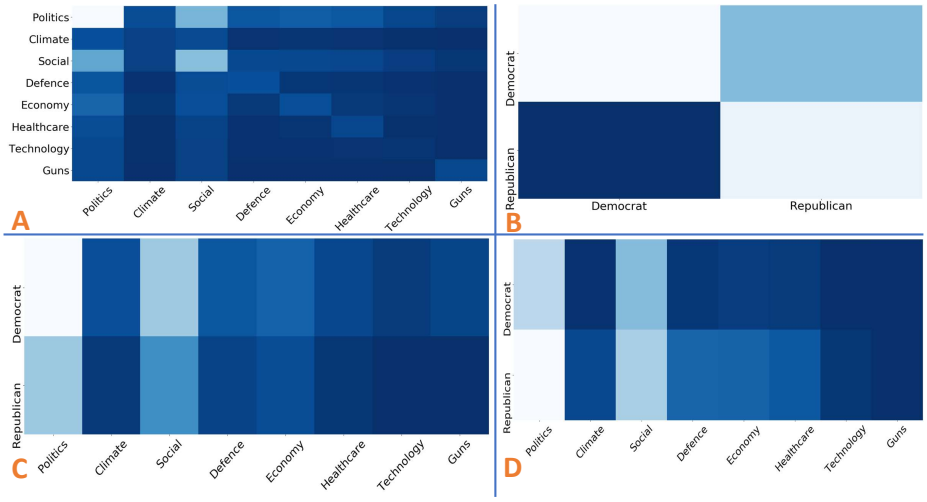

Fig. 3. (A)Topic-Topic Transition ($\sum_{u,v} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$), (B) User-User Transition ($\sum_{k,k'} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$), (C) (Source)User-(Source)Topic Emission ($\sum_{v,k'} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$), and (D) (Destination)User transiting to (Destination)Topic ($\sum_{u,k} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$)

Each user was tagged as either Democrat (D) or Republican (R) (based on their Wikipedia page).

We then extract insights by considering the set of values $\{W_{uv}\mathcal{T}_{k,k'}Q_{v,k'}\}$ for every pair of users (u, v) such that v follows u and (k, k') is a topic-pair.

Figure 3 shows the heat-maps obtained taking various marginalizations over the four tuple (u, k, v, k'). The heatmap in Figure 3A represents the matrix obtained by $\sum_{u,v} W_{uv}\mathcal{T}_{k,k'}Q_{v,k'}$, and hence estimates rate of a parent child topic pair (k, k'). It is instructive to observe that there are off-diagonal transitions (e.g. Politics $\rightarrow$ Social and Economy $\rightarrow$ Politics, Social $\rightarrow$ Politics) that have higher value than some of the diagonal entries, indicating how the conversations evolve across topics. Figure 3B indicates the aggregated user-user rates across parties obtained by aggregating across all topic-pairs and over all users in the same party. The heatmap clearly indicates that Democrat user have a higher aggregated rate, irrespective of the party affiliation of the child tweet's user. Figures 3C and 3D show two different views of the user-topic rate, where 3C includes that spontaneous posts too, but 3D includes only replies. Certain topics are equally prominent in both, but there are topics (e.g. Economy, Healthcare) that a higher rate for the reply tweets than in the source ones.

Drill-down Analysis: We then further drill down in order to identify interesting topical interactions and parent-child tweet examples.We follow two *top-down* approaches:

1. *Topics to Users Interaction*: Figure 4 explains pictorially the first approach. We start with the matrix $\sum_{u,v} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$ (which is a *topic*×*topic* matrix), and identify some asymmetric topic pairs. In Figure 4(A) the (*Economy, Healthcare*) pair is chosen for drilling down further. For this selected *topic-topic* pair we the find the aggregated user-user interaction rate. In the corresponding (*user* × *user*) matrix, (obtained by fixing the topic pairs in the set $\{W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}\}$), we identify the cells which corresponds to users with different affiliations. In Figure 4(B) the (*Democrat, Republican*) pair is chosen. We then extract some sample interactions between these users and present as anecdotes in Figure 4(C).

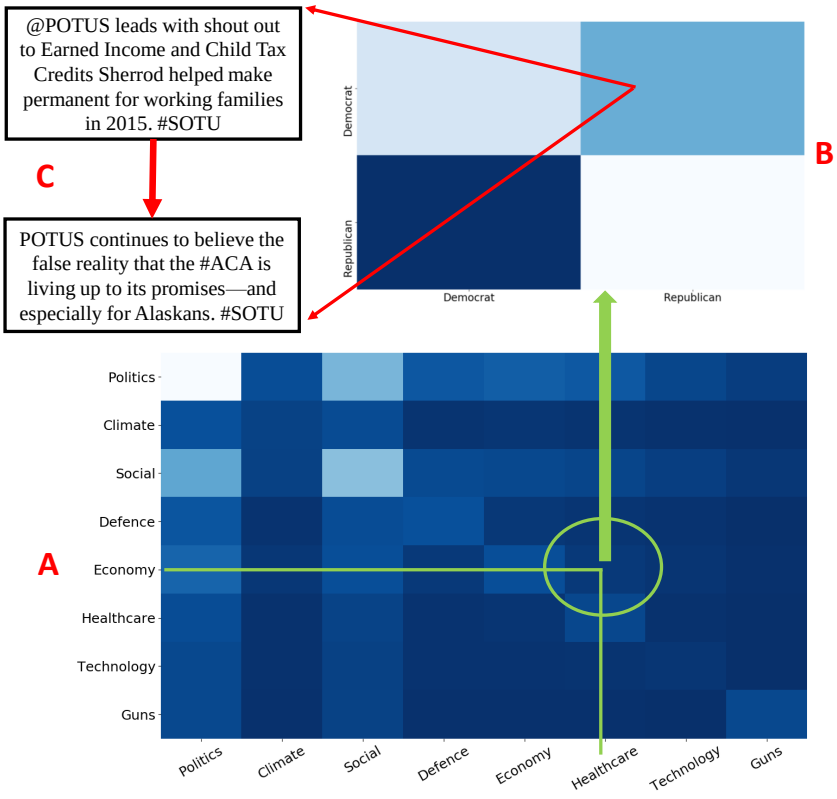

Fig. 4. User-User transition for a particular Topic-Topic transition

2. *Users to Topics Interaction*: For this case, Figure 5 explains the top-down process that we follow. Here we start with the (*user* × *user*) matrix defined by $\sum_{k,k'} W_{u,v}\mathcal{T}_{k,k'}Q_{v,k'}$ (matrix in Figure 5(A)). We then follow a similar process as in the previous case, i.e. we identify the cell which corresponds to users with different affiliations then calculate the aggregate rate of interaction

for all topic-pairs. This gives a (*topic* × *topic*) matrix restricted to the users from matrix 5(A). In this topic-topic interaction matrix we again identify dominant cells with asymmetric topics (namely, (*Social, Politics*) cell in matrix in Figure 5(B)) and then identify anecdotal parent-child tweet pairs. We note that both the (finer grained) topic assignments, as well as the relation among the tweet-pairs looks reasonable.

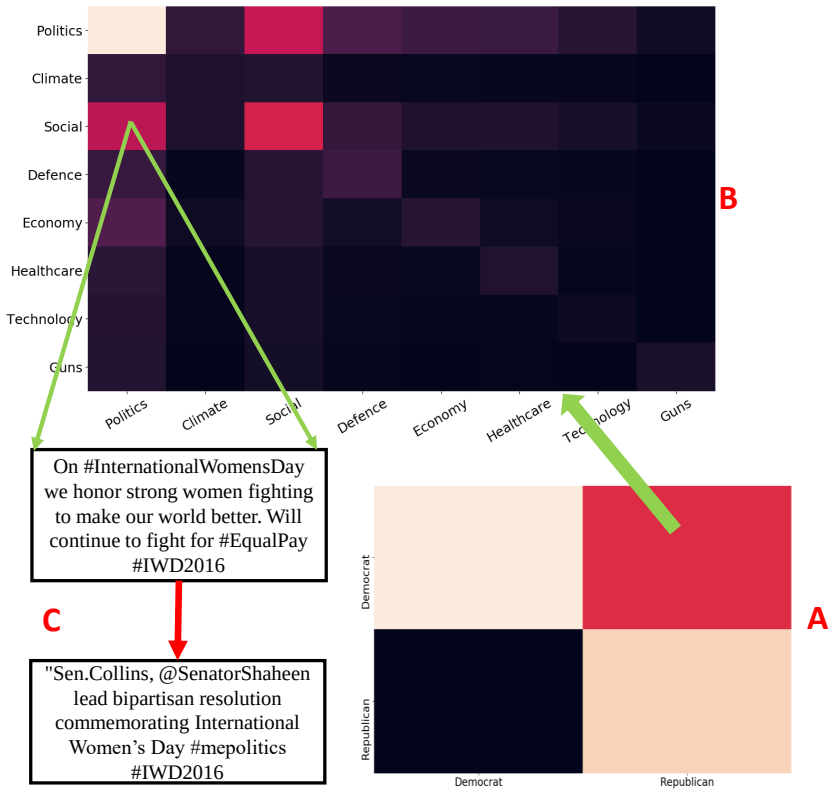

Fig. 5. Topic-Topic transition for a particular User-User transition

Finally, in Table 4, we show some additional examples of parent child tweet pairs that correspond to different topics and also users with different political affiliations. In each row, the topics of the tweets are annotated in bold. Observe that the conversation transitions naturally from one topic to another. This is difficult to capture for other state-of-the-art models.

Table 4. Example parent-child tweet pairs with different topics and different political affiliations for users

Parent Tweet	Child Tweet
(Media) "Joined Cheryl Tan & Don Roberts on WAVY News this morning, to discuss #Syria & where I stand. Watch here: http://t.co/BWqBl164GI"	**(Foreign)** #AlQueda positioned to take #Syria if US action ousts Assad. What message are we sending our troops? ""Fight'em in Iraq support'em in Syria"""
(Politics) Every American should be free to live and work according to their beliefs w/out fear of punishment by the government #Notmybossbusiness	**(Women's Rights)** "Women's private health decisions are btwn her & her doctor, not her boss. #NotMyBossBusiness http://t.co/YHRs0MybWs"
(Foreign)"Fifty years of isolating Cuba had failed to promote democracy, setting us back. Thats why we restored diplomatic relations."–@POTUS #SOTU"	**(Politics)** "Mr. President you've done enough,now it's our time to repair the damage you have done & make this country great again#FinalSOTU #SOTU"
(House Proceedings) @POTUS delivered vision for expanding opportunity. Let's build a future where anyone who works hard & plays by the rules can succeed #SOTU	**(Foreign)** Would like to hear from @POTUS how he plans to get our U.S. sailors in Iranian custody back. So far....nothing. #outoftouch

5 Related Work

Recently, there has been a spate of research work in inferring information diffusion networks. The network reconstruction task can be based on just the event times ([5], [6], [15], [4], [13], [9]), where the content of the events is not considered. Dirichlet Hawkes Process (DHP) [2] is one of the models that uses the content and time information, but the tasks performed are not related to network inference or cascade reconstruction. Similar to our model the DHP, as well is a mixture model and assigns single topic to each event, but it does not have any notion of parent event or topical interactions. The recent models such as HTM [8], and HMHP [1] show that using the content information can be profitable and can given better estimates for the network inference tasks as well the cascade inference task. HMHP model is the closest model to our model, which considers topical interactions as well. However, in both HMHP and HTM [8], the event times are not conditioned on even time stamps. Instead, the topics are generated conditioned on users and parent events.

While all of these capture interactions between users, only HMHP and HTM captures interactions between topics. None of these models capture interactions between users, between topics and between users and topics together.

Following a different line of research, recently there has been effort in using Recurrent Neural Networks (RNN) to model the intensity of point processes [3,14,10]. These look to replace pre-defined temporal decay functions with positive functions of time that are learnt from data. So far, these have not considered latent marks, such as topics, or topic-topic interactions.

6 Conclusions

In this paper, we addressed the problem of reconstructing and analyzing text-based social cascades by capturing user-topic, user-user and topic-topic interactions, by proposing Dual-Network Hawkes process. This executes on top of a super-graph with nodes as user-topic combinations, so that the event times are determined by both the posting and reacting pairs of users and topics. We have shown that this fits real social data better than state-of-the-art baselines for text-based cascades by using a large collection of US political tweets. We have also demonstrated how the model reveals interesting insights about social interactions at various levels of granularity. In future, we wish to incorporate more dimensions and study the effect of inter-dimensional flow of evidence in handling data sparsity. A concrete outcome will be to incorporate structured prior over latent topic graph or, in general, a structure over marks, and improve the existing knowledge-base (e.g. DBpedia) from this cascade evidence.

Acknowledgement

We are grateful to the anonymous reviewers for their helpful feedback. This project has received funding from the Engineering and Physical Sciences Research Council, UK (EPSRC) under Grant Ref: EP/S03353X/1. Anirban Dasgupta acknowledges the kind support of the N. Rama Rao Chair Professorship at IIT Gandhinagar, the Google India AI/ML award (2020), Google Faculty Award (2015), and CISCO University Research Grant (2016).

References

1. Choudhari, J., Dasgupta, A., Bhattacharya, I., Bedathur, S.: Discovering topical interactions in text-based cascades using hidden markov hawkes processes. In: ICDM (2018)
2. Du, N., Farajtabar, M., Ahmed, A., Smola, A., Song, L.: Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In: SIGKDD (2015)
3. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1555–1564 (2016)
4. Gomez-Rodriguez, M., Leskovec, J., Balduzzi, D., Schölkopf, B.: Uncovering the structure and temporal dynamics of information propagation. Network Science **2**(1), 26–65 (2014). https://doi.org/10.1017/nws.2014.3
5. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. ACM Transactions on Knowledge Discovery from Data (TKDD) **5**(4), 1–37 (2012)
6. Gomez-Rodriguez, M., Leskovec, J., Schölkopf, B.: Modeling information propagation with survival theory. In: International Conference on Machine Learning. pp. 666–674 (2013)

7. Hawkes, A.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58(1)** (1971)
8. He, X., Rekatsinas, T., Foulds, J., Getoor, L., Liu, Y.: Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In: ICML (2015)
9. Linderman, S., Adams, R.: Discovering latent network structure in point process data. In: ICML (2014)
10. Mei, H., Eisner, J.M.: The neural hawkes process: A neurally self-modulating multivariate point process. In: Advances in Neural Information Processing Systems. pp. 6754–6764 (2017)
11. Rizoiu, M., Lee, Y., Mishra, S., Xie, L.: A tutorial on hawkes processes for events in social media. In: arXiv (2017)
12. Simma, A., Jordan, M.I.: Modeling events with cascades of poisson processes. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. pp. 546–555 (2010)
13. Wang, S., Hu, X., Yu, P., Li, Z.: Mmrate: Inferring multi-aspect diffusion networks with multi-pattern cascades. In: SIGKDD (2014)
14. Xiao, S., Yan, J., Yang, X., Zha, H., Chu, S.M.: Modeling the intensity function of point process via recurrent neural networks. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
15. Yang, S.H., Zha, H.: Mixture of mutually exciting processes for viral diffusion. In: International Conference on Machine Learning. pp. 1–9 (2013)