NetCodec: Community Detection from Individual Activities

Long Tran * Mehrdad Farajtabar † Le Song ‡ Hongyuan Zha §

Abstract

The real social network and associated communities are often hidden under the declared friend or group lists in social networks. We usually observe the manifestation of these hidden networks and communities in the form of recurrent and time-stamped individuals' activities in the social network. Inferring the underlying network and finding coherent communities are therefore two key challenges in social networks analysis.

In this paper, we address the following question: *Could we simultaneously detect community structure and network infectivity among individuals from their activities?* Based on the fact that the two characteristics intertwine and that knowing one will help better revealing the other, we propose a *multidimensional Hawkes process* that can address them simultaneously. To this end, we parametrize the network infectivity in terms of individuals' participation in communities and the popularity of each individual. We show that this modeling approach has many benefits, both conceptually and experimentally. We utilize Bayesian variational inference to design NetCodec, an efficient inference algorithm which is verified with both synthetic and real world data sets. The experiments show that NetCodec can discover the underlying network infectivity and community structure more accurately than baseline method.

1 Introduction

The exponential growth of recorded social activities has inspired many interesting research directions. From individual activities, a curious analyzer would like to infer more about the social networks as a whole. For example, how contagious individuals' activities are on each other? Are people forming coherent groups or communities in their activities? What is a person's role in his/her perceived community? Is it possible to process the massively available data to answer these crucial questions? These are naturally very interesting and important research questions. The answers to these questions are already having significant impact in practice. For example, in viral marketing, one would like to maximize influence of product advertisement with the least cost. To that end, it is highly beneficial to correctly detect social communities and pinpoint popular individuals whose popularity assures maximized product adoption [8].

Both network infectivity inference and community detection from activities have been addressed extensively. While they are usually studied separately [22, 12, 2], event cascades and clusters are natural duals: clusters block the spread of influence, i.e., whenever a cascade of events comes to a boundry, there is a cluster that can be used to explain why [6]. On the other hand, if a cluster can justify a cascade comes to a stop, then past chain of events can find out something about the clusters.

Based on this fact, we propose a modeling approach that takes into account both network infectivity and community structure in modeling individual activities. Our modeling approach leverages a key observation that these characteristics of a social network intertwine and knowing one would help better understanding and revealing the other. As a result, it is possible to simultaneously infer network infectivity and to detect community structure from individual activities. The proposed method also benefits from having fewer model parameters than existing approaches in literature. This is highly useful as one usually only has limited event data and having fewer model parameters often implies less variance and less algorithmic complexity.

In particular, we propose NetCodec (NETwork COmmunity DEteCtion), a scalable variational inference algorithm for simultaneous network infectivity inference and community detection from individual activities or events. The key idea of the algorithm is to factorize network infectivity into community participation and individual popularity and to leverage the *mean field variation inference framework* to estimate the community participations. Our algorithm can estimate the network infectivity and community structure of a network with I nodes, G groups with $O(kNG+IG)$ computations per iteration, where N is the number of recorded events in a certain time frame, and k is the average number of relevant historical events ($k \ll N$). We validate NetCodec in various simulated and real-world situations.

*VNU-Hanoi, tqlong@vnu.edu.vn.
†Georgia Tech, mehrdad@gatech.edu.
‡Georgia Tech, lsong@cc.gatech.edu.
§Georgia Tech, zha@cc.gatech.edu.

I	number of individuals/nodes
G	number of groups
N	number of events
$\lambda_i(t)$	intensity at time t of user i
μ_i	spontaneous rate of user i
β_i	the celebrity index of user i
$\mathbf{Z}_i \in \mathbb{R}_+^G$	group participation vector of user i
α_{ij}	infectivity rate from user j to user i
$\kappa(t)$	triggering kernel
$K(t)$	the integral $\int_0^t \kappa(\tau)d\tau$
a_g, b_g	Gamma distribution parameters
ℓ, n	event indices, $\ell < n$ if both present

Table 1: Notations

1.1 Problem settings We assume that there are I identities (e.g. individuals, users, sources) that could be grouped into G groups and that their activities are contagious following some network infectivity rate. The community structure and network infectivity are *unknown* to us. Instead, we only know the time and the identity of events (e.g. posts, comments, purchases, earthquakes) occurred in a time frame. The natural question is that "Could we recover both community structure and network infectivity simultaneously from their activities?".

Specifically, let the time and identity of events form a set of C cascades $\{(t_n^c, i_n^c)_{n=1...N_c}\}_{c=1...C}$, where t's are the time of events and i's are the identities. The observation time frame for the c-th cascade is $[0, T_c]$. We would like to find a *participation matrix* $\mathbf{Z} = [z_{ig}] \in \mathbb{R}_+^{I\times G}$ where z_{ig} represents how strong the i-th node associates to the g-th group. We also want to find an *infectivity matrix* $\mathbf{F} = [\alpha_{ij}] \in \mathbb{R}_+^{I\times I}$ where α_{ij} represents how the j-th node influences the i-th node. In the following, the terms "identity", "user", "node" have the same meaning.

In Section 2, we discuss our approach and the modelling technique in more details. In Section 3 we derive NetCodec, a variational inference algorithm that efficiently infers network infectivity and detects coherent communities. In Section 4, we report the experiment results where we apply the model on various simulated and real world situations. In Section 5, we conclude the paper with some remarks on the proposed method and future directions. Before proceeding, let us discuss the related literature on the proposed problem.

1.2 Related Works. Recently, there has been a growing interest in network inference from event data. Authors in [9] were one of the first who tackle the problem of inferring network from the event data. Given the times when nodes adopt pieces of information or become infected, they approximate the optimal network that best explains the observed infection times. Perry et al. [19] introduced a model for treating directed interactions as a multivariate Cox intensity model with covariates that depend on the history of the process and learned the parameters using partial likelihood. Authors in [15] proposed a probabilistic model that combines mutually- exciting point processes with random graph models to infer latent networks. These models, while not being closely related, try to answer how nodes in the network are generally connected or how they influence each other. In contrast our model, directly involves community structure in the modeling.

More closely, authors in [1] proposed a generative model, Community-Cascade Network, based on mixture membership that can fit, at the same time, the social graph and the observed set of cascades. This model, nicely elaborates on the community detection and network inference, however, the nature of events data observed is too far from real applications. They require the data has been observed along with the chain of influence, i.e., which event causes this event. Furthermore, [14] aims at a similar problem, however, as the previous work the definition of event is far from the real data in hand. The event, contains some nodes participating in an event (eg. a party) along with the edges (friendships) between them. In their promising work, Zhou et al. [23], considered the community structure of the network in point process data via adding a regularization term based on nuclear norm. The community structure is only captured indirectly via regularization to enhance parameters estimation and thus cannot find the underlying modules in the network.

After Hawkes [10] originally proposed this mutually-exciting process it has been proved to be useful in various areas such as finance [7], seismology [18, 16], crime [20], and recently causal militant conflict events [13]. For social and influence networks, there are also recent uses of variants of Hawkes processes for analyzing Youtube movies [3], news websites [11, 23], and book sales [5].

2 Modeling Network Activities

In this section, we will discuss our approach to the problem set out in Section 1. We will first review the multidimensional Hawkes process as the basis for modeling event data. We then discuss our modeling technique where one could leverage community structure to help better revealing network infectivity. The readers could refer to Table 1 for the notations used in this paper.

2.1 Multidimensional Hawkes processes. The Hawkes process is an important model for time-stamped events. In its simplest form, the *one-dimensional Hawkes process* is a point process $N(t)$ with its conditional intensity being [10]

$$\lambda(t) = \mu + \alpha \int_{-\infty}^{t} \kappa(t-s) dN(s) = \mu + \alpha \sum_{t_\ell < t} \kappa(t - t_\ell),$$

where $\mu > 0$ is the spontaneous (base) intensity, $\mathcal{H}_t = \{t_\ell < t\}$ are the timestamps of historical events before time t and $\kappa(t)$ is the decay triggering kernel. We focus on the *exponential kernel* $\kappa(t) = he^{-ht}\mathbb{I}(t \geq 0)$ where $\mathbb{I}(\cdot)$ is the *indicator function* and h is the mean parameter. The intensity $\lambda(t)$ is the rate at which new event happens in a infinitesimal interval after t.

The *multidimensional Hawkes process* is a multi-dimensional point process that models time-stamped events from multiple individuals/entities. It allows explicit representation of network infectivity among individuals. The intensity function for the i-th dimension depends on past events as followings

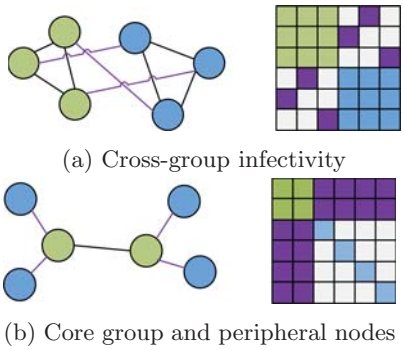
$$\lambda_i(t) = \mu_i + \sum\nolimits_{t_\ell < t} \alpha_{ii_\ell} \kappa(t - t_\ell), \tag{2.1}$$

where $\mu_i > 0$ is the spontaneous intensity for the i-th dimension and i_ℓ is the dimension identity of the ℓ-th event. The *nonnegative coefficient* α_{ij} captures the mutually-exciting property between the i-th and the j-th dimensions. It shows how much influence the events in j-th dimension has on future events in i-th dimension. Larger values of α_{ij} indicates that events in j-th dimension are more likely to trigger an event in the i-th dimension in the future.

In the next section, we will discuss our modeling technique that takes into account the community structure of the networks. We propose that the community structure helps not only better revealing the network infectivity but also reducing the number of parameters of the models.

2.2 Modeling network activities. From the modeling perspective, we would like to incorporate as many key characteristics of network infectivity as possible. Regarding *within-community infectivity*, naturally, individuals affiliated with same communities would have more influence on each other than individuals affiliated with different communities. This natural and key observation inspires us to make an assumption that network infectivity among users' activities depends on how strongly each individual *participates* in his/her community activities. The network infectivity matrix is also *asymmetric* in that a node could have strong influence on another node but not vice versa. These popular nodes' activities tend to trigger a wider wave of events.

Regarding *cross-community infectivity*, individuals in a community often share some common understandings about individuals in other communities. For example, people in a country X have some stereotype about people in country Y. Therefore, a post by a person in country Y or about country Y will trigger certain common responses from people in country X. This situation happens regularly in chat rooms, blogs, and comment sections in the World Wide Web. The marginalization effect of the latent group identity therefore implies a *low-rank structure* of network infectivity. We also would like to incorporate this crucial observation in our modeling approach.



(a) Cross-group infectivity

(b) Core group and peripheral nodes

Figure 1: Different network scenarios and the corresponding infectivity matrices.

To proceed, let $\mathbf{Z}_i = (z_{i1}, \ldots, z_{iG}) > 0$ be user i's degree of participation to the G groups. Furthermore, let $\beta_i > 0$ represents how popular user i is on the network, a *celebrity index*. We propose the following factorization of the infectivity of user j to another user i's activities

$$\alpha_{ij} = \beta_j \langle \mathbf{Z}_i, \mathbf{Z}_j \rangle = \beta_j \sum\nolimits_{g=1}^{G} z_{ig} z_{jg}, i \neq j.$$

As one could see, the more user i and user j participate in the same communities, the stronger the infectivity is. Besides, the popularity of user j also boosts his/her influence on user i's activities. The decomposition also shows the asymmetry as well as the low-rank implication of network infectivity. Note that, we only enforce the low-rank structure on the *off-diagonal elements* of network infectivity. This is a crucial difference in comparison to methods in matrix factorization literature.

Regarding the self-exciting property, we propose that one should not decompose the self-exciting rate α_{ii} and that one should consider it as a model parameter to infer from observed data. The reason is that self-exciting characteristic is an intrinsic property of each individual that is unrelated to his/her relation with other individuals. To keep the notation clear, we denote $\alpha_i = \alpha_{ii}, i = 1 \ldots I$.

To summarize, the previous reasoning leads to the following decomposition of the intensity function

$$\lambda_i(t) = \mu_i + \sum_{t_\ell < t}^{i_\ell \neq i} \sum_{g=1}^{G} \beta_{i_\ell} z_{ig} z_{i_\ell g} \kappa(t - t_\ell) + \alpha_i \sum_{t_\ell < t}^{i_\ell = i} \kappa(t - t_\ell). \tag{2.2}$$

Before we proceed, let us discuss some properties

and advantages of this modeling approach. *First*, the most obvious advantage is that the number of parameters to infer from observed data is $O(I \times G)$ instead of $O(I^2)$ in the case of the original Hawkes process. This reduction is very beneficial given the fact that one often does not have infinite data. The reduction in number of parameters tends to make inference less variant. Besides, fewer number of parameters implies less complexity per iteration of the inference algorithm. *Second*, the decomposition of network infectivity α_{ij} still has more space for extensions. For example, in social networks, one could defines another decomposition that takes into account other activity's feature such as the post content and/or ratings. The interested reader could find some extensions to our model in the *supplemental material*. Another interesting observation is that one could factorize $\mathbf{F}$ into

$$\mathbf{F} - \text{diag}(\mathbf{F}) = \mathbf{Z}\mathbf{Z}^T\text{diag}(\boldsymbol{\beta}) - \text{diag}(\mathbf{Z}\mathbf{Z}^T\text{diag}(\boldsymbol{\beta})).$$

This is a *non-negative matrix factorization* (NMF) of the off-diagonal elements of $\mathbf{F}$ into the off-diagonal elements of $\mathbf{Z}\mathbf{Z}^T\text{diag}(\boldsymbol{\beta})$. Thus, one could view our modeling approach as an implicit factorization of the infectivity matrix where the infectivity matrix is unknown but we know the timestamps of users' activities. One could easily see that depending on the structure of the community participation $\mathbf{Z}$, this point of view allows many interesting scenarios on network infectivity $\mathbf{F}$. For example, cross-group infectivity (Figure 1b); dominant rows/columns for a core group and that peripheral individuals only connect via this core group (Figure 1c). Note that, in these scenarios, network infectivity has a low-rank structure if we only consider the off-diagonal elements. This factorization perspective opens more research directions to investigate in the future.

The above reasoning inspires us to propose that one should conceptually views network infectivity and community structure being *two sides of the same problem*. We postulate that these characteristics intertwine and that knowing one characteristic of the network should help better revealing the other. In the subsequent sections, we will focus on the technical aspects of the proposed model. We will start with *joint likelihood* definition.

2.3 Joint likelihood. In this section, we will define the joint likelihood of the event data. First, we choose a *conjugate prior* for the community participation matrix $\mathbf{Z}$. As it turns out later, we can choose a *Gamma distribution*, $\text{Gamma}(a_g^0, b_g^0)$, as conjugate prior for each of $z_{ig}, i = 1 \dots I, g = 1 \dots G$.

Let us assume that we observed set of C cascades $\{(t_n^c, i_n^c)\}$, $n = 1 \dots N_c$, $c = 1 \dots C$, where t's are the time of events and i's are the identity of users. Given

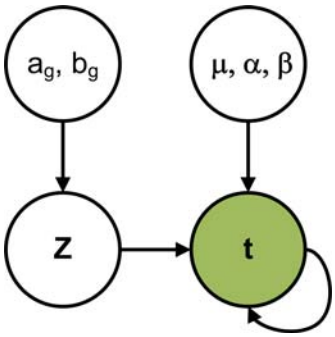


Figure 2: The simplified graphical model of the proposed Hawkes process: solid circle indicates observed time-stamped data.

$\mathbf{Z}$, the likelihood of this set of cascades is [4]

$$L(\mathbf{t}|\mathbf{Z}) = \prod_{c=1}^{C}\left[\prod_{n=1}^{N_c}\lambda_{i_n^c}^c(t_n^c) \times \exp\left(-\sum_{i=1}^{I}\int_0^{T_c}\lambda_i^c(t)dt\right)\right],$$

where $\lambda_i^c(t)$ is defined in Eq. (2.2) using history of events up to time t in the c-th cascade. The *joint likelihood*, the basis of all derivations that follow, is[1]

$$L(\mathbf{Z}, \mathbf{t}) \propto L(\mathbf{t}|\mathbf{Z}) \times \prod_{i=1}^{I}\mathbf{P}(\mathbf{Z}_i).$$

In Figure 2, we present the simplified graphical model corresponding to the proposed Hawkes process. In later derivations of the proposed method, we will mainly work with the *log-likelihood* (detailed expression in supplemental material). We will first develop a method for inferring community participation $\mathbf{Z}$ from the observed cascades, i.e. finding the posterior distribution $\mathbf{P}(\mathbf{Z}|\mathbf{t})$.

3 Variational Inference

As the posterior distribution $\mathbf{P}(\mathbf{Z}|\mathbf{t})$ does not have a nice factorized form, in order to proceed, one could apply the *mean field variational inference* framework [21]. Specifically, we use an approximation distribution q to the posterior distribution on $\mathbf{Z}$ such that $\mathbf{Z}_i$'s are independent,

$$q(\mathbf{Z}) = \prod_{i=1}^{I} q_i(\mathbf{Z}_i).$$

Remarkably, this is the only assumption that one needs on the approximation distribution q. The goal here is to find a distribution q as close as possible to the true posterior distribution $\mathbf{P}(\mathbf{Z}|\mathbf{t})$. To that end, one could utilize the following famous decomposition of the likelihood of observed data

$$\ln \mathbf{P}(\mathbf{t}) = E_q\left[\mathcal{L}(\mathbf{Z}, \mathbf{t})\right] + \mathcal{E}[q] + \text{KL}\left(q \| \mathbf{P}(\mathbf{Z}|\mathbf{t})\right),$$

[1] It is possible to put prior distributions on $\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ and to work in full Bayesian fashion. However, in this work, we only consider these parameters fixed for clarity.

where $\mathcal{E}[q]$ is the entropy of q and $\mathrm{KL}\left(q\|p\right) = E_q\left[\ln(q/p)\right]$ is the Kullback-Leibler divergence between two distribution q and p. As one could see from this decomposition, the better $E_q\left[\mathcal{L}(\mathbf{Z},\mathbf{t})\right]+\mathcal{E}[q]$ approximates the evidence of observed data, the closer q is to $\mathbf{P}(\mathbf{Z}|\mathbf{t})$.

3.1 Evidence lower bound. In the followings, we will bound the the expectation of the joint log-likelihood $E_q\left[\mathcal{L}(\mathbf{Z},\mathbf{t})\right]$ from below so that the inference of $\mathbf{Z}$ is tractable.

THEOREM 3.1. (ELBO) *The expectation of joint log-likelihood $E_q\left[\mathcal{L}(\mathbf{Z},\mathbf{t})\right]$ is lower-bounded by*

$$
\begin{aligned}
&\sum\nolimits_{i=1}^{I}\sum\nolimits_{g=1}^{G}(a_g^0-1)E_q\left[\ln z_{ig}\right]-b_g^0E_q\left[z_{ig}\right]\\
&+\sum_{c=1}^{C}\Bigg\{\sum_{n=1}^{N_c}\Bigg[\eta_n^c\ln\frac{\mu_{i_n^c}}{\eta_n^c}+\gamma_n^c\ln\left(\frac{\alpha_{i_n^c}\sum_{\ell<n}^{i_\ell^c=i_n}\kappa(t_n^c-t_\ell^c)}{\gamma_n^c}\right)\\
&+\sum_{\ell<n}^{i_\ell^c\neq i_n^c}\sum_{g=1}^{G}\eta_{\ell n}^{gc}\left\{E_q\left[\ln(\beta_{i_\ell^c}z_{i_n^cg}z_{i_\ell^cg}\kappa(t_n^c-t_\ell^c))\right]-\ln\eta_{\ell n}^{gc}\right\}\Bigg]\\
&-T_c\sum_{i=1}^{I}\mu_i-\sum_{i=1}^{I}\sum\nolimits_{\substack{n=1\\i_n^c\neq i}}^{N_c}\sum_{g=1}^{G}\beta_{i_n^c}E_q\left[z_{ig}z_{i_n^cg}\right]K(T_c-t_n^c)\\
&-\sum\nolimits_{i=1}^{I}\sum\nolimits_{\substack{n=1\\i_n^c=i}}^{N_c}\alpha_iK(T_c-t_n^c)\Bigg\},
\end{aligned}
$$

in which for the n-th event in the c-th cascade, we have non-negative auxilliary variables $\eta_n^c,\eta_{\ell n}^{gc},\gamma_n$ such that $\eta_n^c+\sum_{\ell<n}^{i_\ell^c\neq i_n}\sum_{g=1}^{G}\eta_{\ell n}^{gc}+\gamma_c^c=1$.

The proof could be found in supplemental material. Next, we will optimize the distribution $q(\mathbf{Z})$ and other model parameters (i.e. $\boldsymbol{\mu},\boldsymbol{\alpha},\boldsymbol{\beta}$). As we are going to see, the optimal approximation to the posterior distribution turns out to have a nice factorization form.

3.2 Inferring community participation. Following the procedure in [21] for mean field variation inference, given the lower bound in the previous section, the optimal distribution $q_i^\star(\mathbf{Z}_i)$ satisfies

$$\ln q_i^\star(\mathbf{Z}_i)=\mathrm{E}_{q_{-\mathbf{Z}_i}}\left[\mathcal{L}(\mathbf{Z},\mathbf{t})\right]+\text{const},$$

where the expectation is over all $\mathbf{Z}_j, j\neq i$.

From the expression of $\ln q_i^\star(\mathbf{Z}_i)$ (details in supplemental material), one could easily verify that the optimal distribution for $\mathbf{Z}_i$ has a nice factorization into G Gamma distributions. This is remarkable because we do not make any assumption on the parametric form of the distributions $q_i(\mathbf{Z}_i)$'s other than their independence. For each $z_{ig}, g=1,\ldots,G$, one could update its Gamma distribution parameters as followings

$$
\begin{aligned}
a_{ig}&=a_g^0+\sum\nolimits_{c=1}^{C}\sum\nolimits_{n=1}^{N_c}\sum\nolimits_{\ell<n}\eta_{\ell n}^{gc}\delta_{\ell n}^{ic},\\
b_{ig}&=b_g^0+\sum\nolimits_{c=1}^{C}\Bigg[\sum\nolimits_{\substack{n=1\\i_n^c\neq i}}^{N_c}\beta_{i_n^c}E_q\left[z_{i_n^cg}\right]K(T_c-t_n^c)\\
&\qquad+\sum\nolimits_{j\neq i}\sum\nolimits_{\substack{n=1\\i_n^c=i}}^{N_c}\beta_iE_q\left[z_{jg}\right]K(T_c-t_n^c)\Bigg],
\end{aligned}
\tag{3.3}
$$

where $\delta_{\ell n}^{ic}=\begin{cases}1, & i_n^c=i, i_\ell^c\neq i \text{ or } i_n^c\neq i, i_\ell^c=i,\\ 0, & \text{otherwise}\end{cases}$.

The definition of $\delta_{\ell n}^{ic}$ represents the influence of both *past and future events* on the posterior distribution. The other terms involving $K(\cdot)$ come from the normalization term (also known as the survival term in the field of survival analysis) of the likelihood.

3.3 Updating auxilliary variables. After each update of q, one could further tighten the bound by the following update formulas[2]

$$
\begin{aligned}
&\eta_n^c\propto\mu_{i_n^c},\quad \gamma_n^c\propto\alpha_{i_n^c}\sum\nolimits_{\ell<n}^{i_\ell^c=i_n^c}\kappa(t_n^c-t_\ell^c),\\
&\eta_{\ell n}^{gc}\propto\beta_{i_\ell^c}\kappa(t_n^c-t_\ell^c)e^{E_q\left[\ln z_{i_n^cg}\right]+E_q\left[\ln z_{i_\ell^cg}\right]}.
\end{aligned}
\tag{3.4}
$$

Note that, one needs to normalize these auxiliary variables so that their sum is equal to 1. From Eq. (3.4), one could interpret these auxiliary variables as the responsibilities of spontaneous rate $\mu_{i_n^c}$, the previous events from other users (i.e. the infectivity $\alpha_{i_n^ci_\ell^c}$), and the self-exciting rate $\alpha_{i_n^c}$. In other words, these auxiliary variables are the probabilities that the n-th event is triggered by these characteristics of the network.

3.4 Inferring individual parameters. For each individual, we need to estimate the spontaneous rate, self-exciting rate, and the celebrity index. As it turns out, these parameters also have the following nice closed-form updates[3]

$$
\begin{aligned}
\mu_i&=\frac{\sum_{c=1}^{C}\sum_{\substack{n=1\\i_n^c=i}}^{N_c}\eta_n^c}{\sum_{c=1}^{C}T_c},\quad \alpha_i=\frac{\sum_{c=1}^{C}\sum_{\substack{n=1\\i_n^c=i}}^{N_c}\gamma_n^c}{\sum_{c=1}^{C}\sum_{\substack{n=1\\i_n^c=i}}^{N_c}K(T_c-t_n^c)},\\
\beta_i&=\frac{\sum_{c=1}^{C}\sum_{\substack{n=1\\i_n^c\neq i}}^{N_c}\sum_{\substack{\ell<n\\i_\ell^c=i}}\sum_{g=1}^{G}\eta_{\ell n}^{gc}}{\sum_{c=1}^{C}\sum_{j\neq i}\sum_{\substack{n=1\\i_n^c=i}}^{N_c}\sum_{g=1}^{G}E_q\left[z_{jg}z_{ig}\right]K(T_c-t_n^c)}.
\end{aligned}
\tag{3.5}
$$

Fortunately, one could compute the expectations in the updates (3.3), (3.4), and (3.5) efficiently as z's are Gamma random variables[4]. To summarize, Algorithm

[2] Given $\sum_i x_i=1$ and $x_i\geq 0,\forall i$, the function $\sum_i a_ix_i-\sum_i x_i\ln x_i$ attains maximum at $x_i^\star=e^{a_i}/\sum_j e^{a_j},\forall i$.

[3] We use the general result $\frac{a}{b}=\arg\max_{x\geq 0}a\ln x-bx,\forall a,b>0$.

[4] Specifically, if $z\sim\mathrm{Gamma}(a,b)$, $E[z]=\frac{a}{b}, E[\ln z]=\psi(a)-\ln b$, where $\psi(\cdot)$ is the *digamma* function.

3.1 outlines the steps of our proposed community detection algorithm, NetCodec. In the output step, we output the mean of Gamma distributions $\mathbf{Z} = \mathbf{A} \oslash \mathbf{B}$ where $\oslash$ is the *element-wise division* operator.

ALGORITHM 3.1. (NETCODEC)

1. Input: Set of cascades $\{(t_n^c, i_n^c)_{n=1\ldots N_c}\}_{c=1\ldots C}$.
2. Initialization: $\mathbf{A}, \mathbf{B} \in \mathbb{R}_+^{I \times G}$, $\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_+^I$.
3. While not converged
   (a) For all user i
      i. Update i-th row of $\mathbf{A}$ and $\mathbf{B}$ using (3.3).
      ii. Update auxiliary variables using (3.4).
   (b) Update $\boldsymbol{\mu}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ using (3.5).
4. Output: $\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{Z} = \mathbf{A} \oslash \mathbf{B}$.

3.5 Implementation issues.

Stopping criteria. The convergence detection involves computing the evidence lower bound, ELBO, to $E_q[\mathcal{L}(\mathbf{Z}, \mathbf{t})] + \mathcal{E}[q]$, where $\mathcal{E}[q]$ is the entropy of the current approximation distribution q. In our implementation, we stop the iterations when the relative change of ELBO is below a threshold (e.g. 10^{-4}). In our experience, the algorithm often stops after less than 40 iterations.

Number of data sweeps. From Algorithm 3.1, we could see that, for every update of $\mathbf{Z}_i$ (i.e. the update of the i-th row of $\mathbf{A}$ and $\mathbf{B}$), one needs to update the auxiliary variables. This results in one sweep over the data for every update of $\mathbf{Z}_i$. However, to scale to large number of individuals and lengthy cascades, one could leverage a key observation on the evidence lower bound. That is, the lower-bound is valid for *any set* of auxiliary variables. Using careful book-keeping technique, one could reduce the number of data sweeps to *one* in order to update all Gamma distributions of all users.

Number of relevant historical events. The computation of the auxiliary variables and the accumulation of the denominators and numerators of model parameters (i.e. $\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta}$) involves a nested loop over indices ℓ of events that happened before the n-th event leading to undesirable $O(N^2)$ complexity. This results in the complexity of each iteration being proportional to N^2, where N is the number of events in a cascade. Luckily, one could skip irrelevant historical events where the kernel value $\kappa(t_n - t_\ell)$ is small because the corresponding auxilliary variables would also be very small. This greatly reduces the complexity of the computation to $O(kNG + IG)$ per iteration where k is the average number of relevant historical events.

Speed up with parallelization. The computation of auxiliary variables for each event is completely independent of each other. The accumulation of Gamma distribution parameters as well as individual parameters are also independent. These observations are great sources for a parallelized implementation.

4 Experiment results

4.1 Performance Evaluation. We evaluate the performace of the proposed method using the following criteria

- *Normalized Mutual Information* (NMI): We compare the estimated clusterings Ω with the ground truth clusterings Γ using the NMI score

$$\mathrm{NMI}(\Omega, \Gamma) = \frac{\sum_k \sum_j \mathbf{P}(\Omega_k \cap \Gamma_j) \log \frac{\mathbf{P}(\Omega_k \cap \Gamma_j)}{\mathbf{P}(\Omega_k)\mathbf{P}(\Gamma_j)}}{(\mathcal{E}[\Omega] + \mathcal{E}[\Gamma])/2},$$

  where Ω_k, Γ_j is the k-th and j-th clusters in Ω and Γ, respectively, and $\mathcal{E}[\Omega]$, $\mathcal{E}[\Gamma]$ are their entropies. The NMI score is a value between 0 and 1, with 1 representing perfect cluster matching. To assign users to clusters, we use the maximum elements in each row of $\mathbf{Z}$.

- *Kendall Rank Correlation* (RankCorr): We compare the estimated celebrity index $\boldsymbol{\beta}$ with the ground truth using the following score

$$\mathrm{RankCorr}(\mathbf{x}, \mathbf{y}) = \frac{N_{\text{concordant}} - N_{\text{discordant}}}{I * (I-1)/2},$$

  where $N_{\text{concordant}}$ is the number of pairs of indices (i, j) that $x_i > x_j$ and $y_i > y_j$, or $x_i < x_j$ and $y_i < y_j$. The RankCorr score is a value between -1 and 1, with 1 representing perfect rank matching.

- *Relative error* (RelErr): We compare the infectivity matrices $\mathbf{F}$ using the average relative errror of their elements. Specifically, we have

$$\mathrm{RelErr}(\mathbf{F}_1, \mathbf{F}_2) = \frac{1}{I^2} \sum\nolimits_{i,j=1}^{I} |\alpha_{ij}^2 - \alpha_{ij}^1| / |\alpha_{ij}^1|$$

- *Predictive log-likelihood* (PredLik): We also compute the log-likelihood of a hold-out test data set in order to show the predictive power of the compared models.

Note that, because of the factorization, at best, one could only recover $\mathbf{Z}$ and $\boldsymbol{\beta}$ up to a constant factor. Therefore, the NMI and RankCorr scores are more suitable criteria than the absolute error or squared error when comparing the participation matrices and the vectors of celebrity indices.

4.2 Synthetic data. We start with experiments with simulated data where we know the ground truth

network infectivity. We generate the ground truth parameters $\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ randomly to satisfy certain stability conditions[5]. The parameters form a network of 500 nodes. We then generate event cascades with different time frame length settings and also generate a hold-out set of the same size to use as test set. The time frame lengths are $(10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6)$. In total, there are about 3×10^5 events when $T = 10^6$. We run each experiment 10 times and take the average of the scores over all the 10 runs. We then verify the convergence of the proposed method by varying the time frame of the simulations.

We generate data according to two scenarios:

- The nodes form 10 clusters and there are some cross-group infectivity.
- There is a core group and the remaining nodes only connect via this core group.

In Figure 3 and 4, we report the performance of the proposed method in comparison with the Hawkes MLE solver (denote HAWKES in the figures) in [23] in the two aforementioned scenarios. The figures show that both NetCodec and HAWKES are able to increase their performance when the time frame length increases. However, in comparison to the ground truth, NetCodec outperforms HAWKES in all performance measures given the same time frame length. This could be attributed to the fact that NetCodec models the low rank assumption directly and as a result, it needs to estimate fewer parameters, hence the better performance in both area. Especially in the case that there is a core group (Figure 4), there are a lot of near zero elements in the infectivity matrix making accurate recovery of these elements very difficult. This explains the high RelErr that both algorithms encounters. However, when there are enough data, NetCodec is able to recover the infectivity matrix much better than HAWKES.

In Figure 3d, we show that NMI score of NetCodec and HAWKES with respect to the ground truth clusterings. As HAWKES provides no clustering, its clusterings are computed via a spectral clustering [17] of the infectivity matrix. One could see that while both algorithms are able to recover the clusterings with enough data, NetCodec outperforms HAWKES when data are insufficient.

4.3 Real-world event data.

MemeTracker. We extract events of the most active sites from the MemeTracker dataset[6]. This dataset contains the times that articles are published in various websites/blogs from August 2008 to April 2009. We select most active 500 sites with about 8 million events from these sites.

We use the MemeTracker data provided links between articles to establish an estimated ground truth of the clusters. To this end, we count the number of links between all pairs of sites to build a similarity matrix. We then run a spectral clustering algorithm [17] on this similarity matrix with different settings on the number of clusters. While one could choose the number of clusters based on model scores (i.e. data log-likelihood plus model complexity) such as Bayesian or Akaike information criterion, here, for demonstration purpose, we set the number of clusters to be 10 and 20. We then run NetCodec and HAWKES on the timestamped data only (i.e. without the link information) to find out if these algorithms could recover the estimated ground truth clusterings. As mentioned in the experiments on synthetic datasets, the clusterings for HAWKES are computed via spectral clustering on the estimated infectivity matrix.

In Figure 5b and Figure 5b we shows the NMI scores of these algorithms with respect to the ground truth estimated from the similarity (count) matrix when the number of clusters set to 10 and 20. One could see that in both settings NetCodec is able to recover part of the clusterings while HAWKES fails on this dataset.

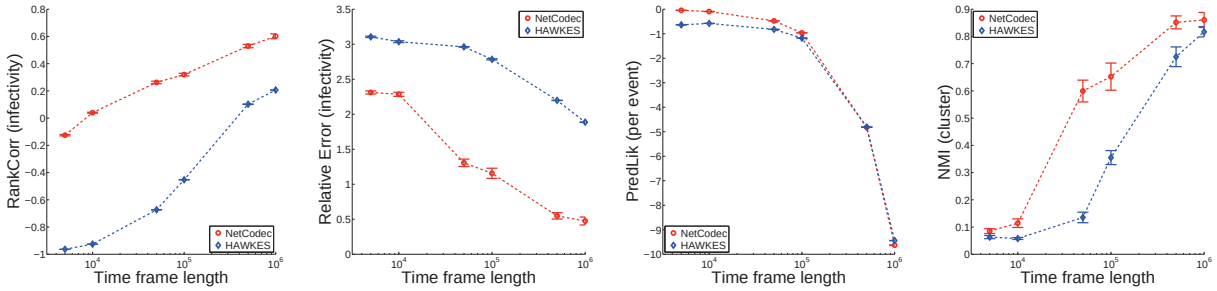
In Figure 5c, we visualize the clustering result (i.e. the participation matrix $\mathbf{Z}$). Detailed examination of the clusters produced by NetCodec shows some consistent clusters spanning common categories. Examples of clusters found by NetCodec and their respective popular websites having with high celebrity index are news (reuters.com, npr.org), business (businessweek.com, forbes.com, cbsnews.com), and technology (hardwarezone.com, digitalcamerareview.com). There are consistent clusters with nationality identity such as Brazilian sites, Japanese sites, Italian sites. One should note that the clusters are formed using purely timestamps of activities/events happened on this sites. The results show that the activities on these sites allow us to group them into meaningful clusters.

Earthquake. The next dataset that we investigate is the Earthquake dataset[7]. We download 16000 earthquakes that have minimum magnitude 4 in the 12 months from Oct. 2013 to Oct. 2014. The earthquake information contains location (i.e. longtitude, lattitude) and timestamps in seconds (see Figure 6, red dots are big cities, colored bigger dots are earthquake locations). In this experiments, we only use the timestamps of the earthquakes (divided by 3600 to convert to hours) as input to the inference algorithms to investigate if timestamped information results in a coherent

[5] The spectral norm of $\mathbf{F}$ is less than 1.

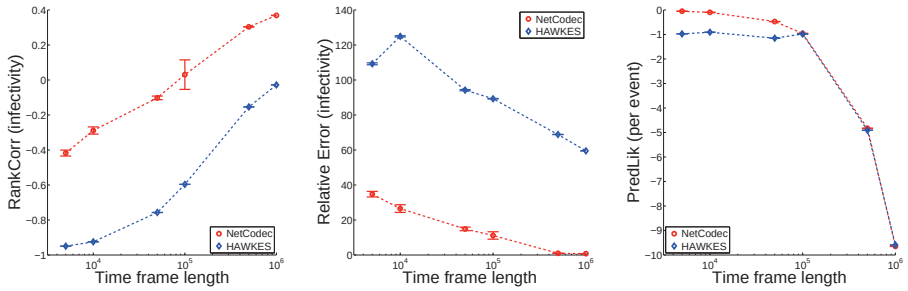
[6] http://www.memetracker.org/data.html

[7] http://earthquake.usgs.gov/earthquakes/



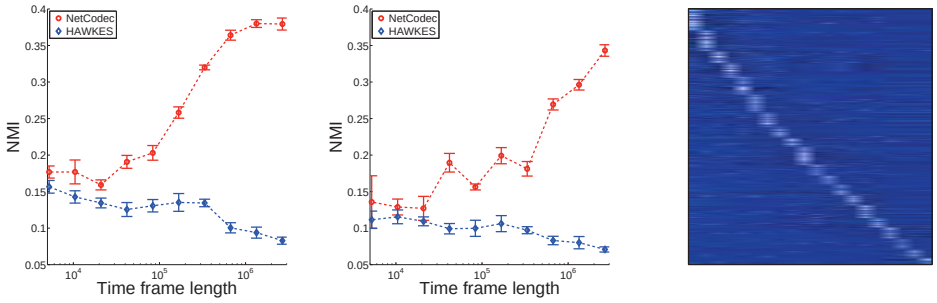
(a) RankCorr (b) RelErr (c) PredLik (d) NMI

Figure 3: Cross-group infectivity scenario: comparison to ground truth (left) average RankCorr of columns of network infectivity matrix; (middle) average RelErr of elements of the infectivity matrix; (right) predictive log-likelihood on test data.



(a) RankCorr (b) RelErr (c) PredLik

Figure 4: Core group scenario.



(a) NMI (10 clusters) (b) NMI (20 clusters) (c) 20 clusters (NetCodec)

Figure 5: Clustering results on MemeTracker dataset.

clustering. To establishes the identities of events (i.e. the i's variables), we draw a longtitude/lattitude grid on the global map and all earthquakes that occur in a grid square (of size 2×2) will have same identity. In total we have 1021 identities and our goal is to classify these identities into clusters. We run NetCodec with exponential kernel ($\lambda = 0.04$) and report the clustering result in Figure 6. One could see that there are geological regions where earthquakes form clusters. This is remarkable as we use only timestamped information. The location information are used only to form identies and then discarded. More detailed discussion could be found in the Appendix. A future experiment direction would be incorporating location information, possibly by using a kernel that takes location of events into account.

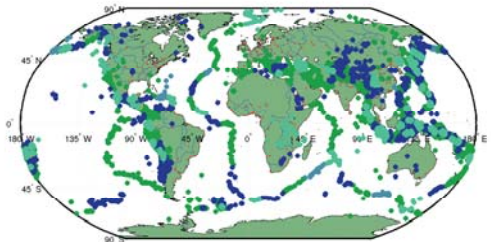


Figure 6: Clustering results on Earthquake dataset.

5 Conclusion

In this work, we propose that one could infer the network of social influence along with its community structure from the observed recurrent events in the social

networks. To that end, we utilize the key observation that regular activities often raise influence among users in the same group. The proposed model based on the Hawkes model is designed to take into account this observation and other assumptions such as the low-rank structure. The inference algorithm following the mean-field variational principle nicely consists of closed form updates that could be sped up by various implementation techniques including parallelism. The experiments on simulated dataset show that the proposed model could estimate both network infectivity and and community structure and produce better predictive model with less training samples than the baseline methods. Experiments on real dataset show that the proposed method are able to produce meaningful clusters using only activities from websites.

There are interesting paths to extend this study: First, we plan to investigate the extensions that cover other features of an event, for example, document content and ratings. The content and ratings effects on community structure could be expressed in the factorization of the influence between events. Moreover, it is also interesting to incorporate the memes/trends and community structure in one model.

Acknowledgement: Our work is supported in part by NSF/NIH BIGDATA 1R01GM108341-01.

References

[1] N. Barbieri, F. Bonchi, and G. Manco. Cascade-based community detection. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 33–42. ACM, 2013.

[2] C. Blundell, K. A. Heller, and J. M. Beck. Modelling reciprocating relationships with hawkes processes. In *NIPS*, pages 2609–2617, 2012.

[3] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[4] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes*, volume 2. Springer, 1988.

[5] F. Deschatres and D. Sornette. Dynamics of book sales: Endogenous versus exogenous shocks in complex networks. *Physical Review E*, 72(1):016112, 2005.

[6] D. Easley and J. Kleinberg. Networks, crowds, and markets. *Cambridge Univ Press*, 6(1):6–1, 2010.

[7] P. Embrechts, T. Liniger, and L. Lin. Multivariate hawkes processes: an application to financial data. *Journal of Applied Probability*, 48:367–378, 2011.

[8] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. In *NIPS '14: Advances in Neural Information Processing Systems*, 2014.

[9] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.

[10] A. G. Hawkes and D. Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

[11] A. Johansen and D. Sornette. Download relaxation dynamics on the www following newspaper publication of url. *Physica A: Statistical Mechanics and its Applications*, 276(1):338–345, 2000.

[12] M. Kim and J. Leskovec. Nonparametric multi-group membership model for dynamic networks. *Advances in Neural Information Processing Systems*, pages 1385–1393, 2013.

[13] L. Li and H. Zha. Dyadic event attribution in social networks with mixtures of hawkes processes. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1667–1672. ACM, 2013.

[14] X.-L. Li, A. Tan, S. Y. Philip, and S.-K. Ng. Ecode: event-based community detection from social networks. In *Database Systems for Advanced Applications*, pages 22–37. Springer, 2011.

[15] S. W. Linderman and R. P. Adams. Discovering latent network structure in point process data. *arXiv preprint arXiv:1402.0914*, 2014.

[16] D. Marsan and O. Lengline. Extending earthquakes' reach through cascading. *Science*, 319(5866):1076–1079, 2008.

[17] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering1 analysis and an algorithm. *Proceedings of Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press*, 14:849–856, 2001.

[18] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

[19] P. O. Perry and P. J. Wolfe. Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(5):821–849, 2013.

[20] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11):115013, 2011.

[21] J. Winn, C. M. Bishop, and T. Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4), 2005.

[22] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. *arXiv preprint arXiv:1401.7267*, 2014.

[23] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 641–649, 2013.