Mixture of Mutually Exciting Processes for Viral Diffusion

Shuang-Hong Yang SYANG@TWITTER.COM
Twitter Inc., 1355 Market St., San Francisco, CA 94103

Hongyuan Zha ZHA@CC.GATECH.EDU
College of Computing, Georgia Tech, Atlanta, GA 30332

Abstract

Diffusion network inference and *meme tracking* have been two key challenges in viral diffusion. This paper shows that these two tasks can be addressed simultaneously with a probabilistic model involving a mixture of mutually exciting point processes. A fast learning algorithms is developed based on mean-field variational inference with budgeted diffusion bandwidth. The model is demonstrated with applications to the diffusion of viral texts in (1) online social networks (e.g., Twitter) and (2) the blogosphere on the Web.

1. Introduction

The question of how viral signals (e.g., behaviors, ideas, diseases) evolve over time and spread through social networks has become a topic of active interest in many fields (Wortman, 2008). Research, however, has been hampered by two fundamental challenges:

- *Diffusion network inference.* The diffusion process usually occurs over a hidden network whose structure cannot be observed or identified directly. For example, in epidemiology, when a person was infected with a contagious virus, it's usually infeasible to trace this back to identify how and by whom he got infected. Similarly, in viral marketing, when a user adopted an idea or behavior, it's generally impossible to identify where he adopted it from. The inference problem for uncovering such hidden diffusion network is daunting, partly due to the long-standing challenge of recovering causality from noise-corrupted historical data.
- *Tracking trending memes.* In many scenarios, there are a number of different but related viruses[1] simultaneously diffusing and entangling with one another, yet detection and identification is nontrivial. For example, several diseases may propagate among a community of people at the same time, but only the conditions and symptoms, rather than the type of the disease, can be directly observed when one got infected. To complicate matters, viruses are often evolving over time at different time scales (e.g., the genetic structures of viruses undergo significant mutations in the course of diffusion), and different viruses can hybridize with one another.

These two challenges have been largely open until very recently a few models started to emerge. To our surprise, it seems these two tasks have to date been investigated separately, yet they rely on the solution to each other as a premise. For example, recent work on cascade diffusion models (Gomez-Rodriguez et al., 2011; Gomez-Rodriguez & Schölkopf, 2012; Snowsill et al., 2011; Ypma et al., 2012) showed that a hidden diffusion network can be recovered, to certain extent, from the contagion history. These approaches, however, require the single successive contagion history to be a priori segmented into a set of independent cascades, each of which corresponds to the contagion history of a single meme. Likewise, existing research on meme tracking devised models to trace the evolution of textual contents while assuming that the structure of the network over which memes spread is given and fixed.

In this paper, we attempt to address both tasks simultaneously. The key rationale is that the inferences and algorithmic procedures underlying the two tasks are correlated and can be tackled jointly. As a matter of fact, both tasks are essentially special cases of a more general problem: tracking the flow of memes, either spatially over a network (i.e., network diffusion), or temporally over time (i.e., meme evolution). By pro-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

[1]We use "meme" or "virus" interchangeably to refer to a distinct cluster of viral events that evolve and propagate similarly to one another but relatively independent of other events.

viding a unified solution, we will not only eliminate the dependencies on human annotations or external algorithms but also enable joint inference/optimization.

We introduce a probabilistic mixture model over a set of multivariate Hawkes process (MHP) (Hawkes, 1971). The mutually exciting nature of Hawkes process makes it a perfect choice for modeling the evolution and propagation of a single meme or independent memes. By casting meme identities as latent variables, the proposed mixture model further enables us to simultaneously track the diffusion of multiple memes temporally over time and spatially across the network. The model accounts for both exogenous (i.e., infections by outsiders) and endogenous (i.e., infections within a social network) effects in viral diffusion, and it also models the evolution of viral contents. We derive a fast inference algorithm based on mean-field variational methods with budgeted diffusion bandwidth. Experiments on controlled synthetic data and two real-world data sets (i.e., Twitter and blogosphere) demonstrate the effectiveness of the proposed model in both meme tracking and diffusion network inference.

2. Models

Let us consider a typical scenario in viral diffusion, where a set of memes $\{m|m = 1, \ldots, M\}$ simultaneously evolve over time t and propagate among a set of nodes $I = \{i|i = 1, \ldots, I\}$ through a *hidden* social network $\mathcal{G}$. The typical observations are a sequence of N events $\{E_n|n = 1, \ldots, N\}$, where an event is denoted as $E_n = (t_n, i_n, W_n)$, i.e., a node i got infected at time t and was observed with behavior (e.g., the symptoms of an infected patient) W. Our goal is to automatically (1) identify memes and track their evolutions and (2) uncover the structure of the graph $\mathcal{G}$ (i.e., how a virus propagates from one node to another). We show these two tasks can be tackled jointly with a probabilistic mixture model over Hawkes processes.

2.1. Hawkes process

Evolutionary point processes are widely used to describe the occurrence of discrete events. Typically, a point process is a list of times $\{t_1, \ldots, t_N\}$ at which an N sequence of events $\{E_1, \ldots, E_N\}$ occur. Denote $N(t)$ the number of points (i.e., occurrences of events) in $(-\infty, t]$ and $\mathcal{H}_t = \{E|t_E < t\}$ the *history* of events up to but not including t, the *conditional intensity function* (a.k.a. *hazard function*)

$$\lambda(t|\mathcal{H}_t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}[N(t + \Delta t)|\mathcal{H}_t]}{\Delta t}$$

is the most convenient way to characterize a point process, which represents the expected instantaneous rate of future events at t. Let f and F be the conditional density and the corresponding cumulative distribution for t, the intensity can be also defined by: $\lambda(t|\mathcal{H}_t) = f(t|\mathcal{H}_t)/S(t|\mathcal{H}_t)$, where $S(t|\mathcal{H}_t) = 1 - F(t|\mathcal{H}_t)$ is known as the *survival function* (the probability that an event does not happen up to t). Because of the dependence on $\mathcal{H}_t$, most point processes are *not* Markovian except for a few simple cases (e.g., Poisson processes). For clarity, hereafter we use * to imply the dependence on $\mathcal{H}_t$, e.g., $\lambda(t|\mathcal{H}_t)$ will be denoted $\lambda^*(t)$.

The Hawkes process is a class of self or mutually exciting point process models (Hawkes, 1971). A univariate Hawkes process $\{N_t\}$ is defined by

$$\lambda^*(t) = \mu(t) + \int_{-\infty}^{t} \kappa(t - s) dN(s), \qquad (1)$$

where $\mu : \mathbb{R} \to \mathbb{R}_+$ is a deterministic base intensity, $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ is a kernel function expressing the time-decay effect. The process is well known for its *self-exciting* property (i.e., the occurrence of an event increases the probabilities of future events) and has been used in modeling phenomena as widespread as earthquake aftershocks (Ogata, 1981), crimes (Mohler et al., 2011) and financial contagions (Bacrya et al., 2012). The multivariate Hawkes process $\{N_d(t)|d = 1, \ldots, D\}$ is a multi-dimensional extension to the univariate case, describing the occurrences of D coupling point series (Hawkes, 1971; Liniger, 2009). The intensity function $\lambda^* = [\lambda_1^*, \ldots, \lambda_D^*]^\top$ is defined by

$$\lambda_d^*(t) = \mu_d(t) + \sum_{d'=1}^{D} \int_{-\infty}^{t} \kappa_{d'd}(t - s) dN_{d'}(s), \qquad (2)$$

where there is a time-decaying triggering kernel $\kappa_{d'd}$ between a pair of dimensions d' and d. MHP is also known as linear mutually exciting process as the occurrence of an event in one dimension increases the likelihood of future events in all dimensions. It was recently applied to modeling viral marketing (Crane & Sornette, 2008) and finance trades (Bacrya et al., 2012).

2.2. Mixture of Hawkes processes

We are interested in modeling the diffusion process of M memes over a hidden network $\mathcal{G}$ involving I nodes. We show the observations $\{E_n = (t_n, i_n, W_n)\}$ can be modeled effectively and conveniently with a mixture of *networked* MHPs. The key idea for inferring the diffusion network is to integrate the network structure into the model and to leverage the mutually-exciting mechanism. Let us for now consider the diffusion process of a single meme, the events can be modeled with an I-dimensional MHP $\{N_i(t)|i = 1, \ldots, I\}$ where each

node corresponds to one dimension with intensity

$$\lambda_i^*(t) = \mu_i(t) + \sum_{j=1}^{I} \int_{-\infty}^{t} \kappa_{ji}(t-s) dN_j(s) \quad (3)$$

The baseline intensity $u_i(t)$ captures how often node i gets infected spontaneously (i.e., independent of other nodes in the network). We assume this meme-birth process is a homogeneous Poisson process with $\mu_i(t) = \mu$. The second term captures the mutual excitation, which is the key for modeling the network diffusion. We propose to decompose the pairwise triggering kernel into two parts, i.e.,

$$\kappa_{ji}(\Delta t) = \alpha_{ji} \times \kappa(\Delta t) \quad (4)$$

where the *asymmetric* infectivity matrix $\alpha \in \mathbb{R}_+^{I \times I}$ characterizes the structure of the diffusion network $\mathcal{G}$; each element α_{ij} expresses the degree of social influence from a node i to another node j, i.e., how likely i infects j. The kernel[2] κ is now node-independent and used only to capture the time-decay effect.

Note that the survival analysis models of (Gomez-Rodriguez et al., 2011) are equivalent to degraded special cases of the above MHP model with implicit assumptions that (1) events are not recurrent, i.e., one node can be infected only once; and (2) the network being inferred is closed: nodes can only spread memes already existing in the network; neither can they be influenced by someone outside the network nor can they create a new meme. These assumptions are rather unrealistic and have naturally been eliminated in the MHP model.

Now consider the case where we have a set of M memes diffusing at the same time, we assume a set of latent variable $\{Z_n\}$, one for each event E_n, to represents the meme-identity of each event. We use the binary one-of-M coding for Z, i.e., $Z_n = [Z_{n1}, \ldots, Z_{nM}]$ is a binary vector with $Z_{nm} = 1$ if and only if E_n belongs to the m-th meme. With Z, we essentially segment the single successive event cascade $\{E_n\}$ into M cascades: $\{\{E_n | Z_n = m\}, m = 1, \ldots, M\}$. Since the diffusion process of each meme can be modeled as an MHP, we now have a matrix-variate Hawkes processes, $\{N_{m,i}(t)\}$, whose intensity is defined by

$$\lambda_{i,m}^*(t) = \mu_{i,m} + \sum_{j=1}^{I} \int_{-\infty}^{t} \alpha_{ji,m} \kappa(t-s) dN_{m,j}(s) \quad (5)$$

[2]In this paper, we primarily use the exponential kernel, i.e., $\kappa(\Delta t) = \omega e^{-\omega \Delta t}$ if $\Delta t \geq 0$ or 0 otherwise. However, the model development is independent of kernel choice and extensions to other kernels such as power-law, Rayleigh, non-parametric kernels are straightforward.

where $\mu_{i,m}$ specifies how likely node i got infected spontaneously on meme m (e.g., i was a meme-creator or infected by an outsider). We assume $\mu_{i,m} = \mu_i \lambda_m$, where λ_m specifies the base rate of meme m's propagation. Moreover, as these M memes are spreading over *the same social network*, the topology of the network doesn't change from meme to meme. We assume the social influence rate is meme-independent and decided purely by the number of existing infections in the network, thus $\alpha_{ji,m} = \alpha_{ji}$. With these specifications and by using the piecewise-constant property of $N(t)$, we rewrite the intensity Eq(5) as follows

$$\lambda_{i,m}^*(t) = \mu_i \gamma_m + \sum_{t_l < t} Z_{lm} \alpha_{i_l i} \kappa(t - t_l). \quad (6)$$

Note that all the events have been chronologically ordered such that, for any n, $t_n \leq t_{n+1}$. The time causality induces a directed acyclic graph (DAG) over the events $\{E_n\}$; the summation in Eq(26) is over one path of the DAG, i.e., over all the events preceding t.

2.3. Modeling viral texts and mutations

The mixture of MHPs model described above only models the time t and the player (i.e., node id) i of each event. Here we extend the model to capture the content W (i.e., the message being spread). The textual content is crucial for meme tracking since messages belonging to the same meme are usually semantically related (e.g., the genetic structures of the same virus are similar; tweets of the same event are often on similar topics). The content information is also important for network recovery, e.g., the "who is spreading from whom" relationship, if identified from messages, can be of great help for eliminatation of false-positive edges in the time-causality DAGs (Snowsill et al., 2011).

Content can be incorporated into the Hawkes process model in a number of different ways. For example, we can assume messages in each meme are generated from the same language model, e.g.,:

$$p(W_n | Z_{nm} = 1) = \text{Multinomial}(\beta_m) \quad (7)$$

Here, the texts are represented by a simple bag-of-word model. Alternatively, we can extract more expressive features or use other more powerful text representation models, and relate these features to the meme identity with a classification model. For example, suppose x_n is the feature vector we extract from W_n, we can use a *Multinomial logit model*:

$$p(Z_{nm} = 1 | x_n, \theta) = \frac{\exp(\theta_m^\top x_n)}{\sum_{m'=1}^{M} \exp(\theta_{m'}^\top x_n)} \quad (8)$$

where $\{\theta_m | m = 1, \ldots, M\}$ are linear weight vectors.

In contrast to modeling the content directly, another way is to detect the mutation-from relationship (i.e., who was mutated from whom) by string matching algorithms (Leskovec et al., 2009; Snowsill et al., 2011), and model such pairwise constraints instead, for example, via regularization in MLE:

$$\max \quad \log L + \nu \sum_{l<n} C_{ln} \Omega(Z_n, Z_l) \tag{9}$$

where $C_{ln} = 1$ if events n was mutated from l or 0 otherwise, $\Omega(\cdot, \cdot)$ is a similarity function[3].

2.4. Smoothing

Note that the number of latent variables Z's grows linearly with the number of event observations N, which makes the inference vulnerable to overfitting. We regularize the inference by assuming a Bayesian prior for Z. Particularly, each Z_n is assumed to be sampled *i.i.d.* from a multinomial distribution parameterized by π, where $\pi = [\pi_1, \ldots, \pi_m]$ and $\sum_{m=1}^{M} \pi_m = 1$.

3. Inference

Statistical inference of non-Markovian point process has become a topic of active interest lately. (Guttorp & Thorarinsdottir, 2012) reviews some of the recent advances. In this section, we derive a mean-filed variational Bayesian inference algorithm for the mixture of MHPs model. This algorithm is, however, quadratic $O(N^2)$. To facilitate large-scale applications, we present a fast algorithm by controlling the bandwidth of longitudinal diffusion.

3.1. Variational inference

Take the mixture of MHPs with content model Eq(7) as an example, suppose we have observations $\{E_n = (t_n, i_n, W_n)\}$ on $[0, T]$ where $T > 0$.The likelihood for the complete data (i.e., assuming Z's are observed) is given by (*c.f.* Appendix 1)

$$L(Z, t, W) = \prod_{n=1}^{N} \prod_{m=1}^{M} \left(\pi_m \lambda_{i_n,m}(t_n) p_m(W_n)\right)^{Z_{mn}} \times \exp\left(-\sum_{i=1}^{I} \sum_{m=1}^{M} \int_0^T \lambda_{i,m}(s) ds\right). \tag{10}$$

Note that the latent variables Z's are inter-dependent, i.e., the meme identity at current step Z_n depends on all the past meme ids $\{Z_1, \ldots, Z_{n-1}\}$ as well as all the future ones $\{Z_{n+1}, \ldots\}$. Marginalizing over such interconnected series is intractable. To this end, we use mean-field variational inference by assuming a fully-factorizable variational distribution q for Z's, which is parametrized by free variables ϕ's (one ϕ per Z),

$$q(\{Z_n\}) = \prod_{n=1}^{N} \text{Multinomial}(Z_n | \phi_n). \tag{11}$$

We then lower-bound the log-likelihood with help of q:

$$\begin{aligned} \mathcal{L}(t, W) =& \log\left(\int_{\{Z\}} L(Z, t, W) d\{Z\}\right) \\ \geq& \mathbb{E}_q[\mathcal{L}(Z, t, W)] + \mathcal{E}[q], \end{aligned} \tag{12}$$

where $\mathcal{E}[q]$ denotes the Shannon entropy of Z's under q. The right-hand side Eq(12), denoted $\mathfrak{L}$, is known as the *evidence lower-bound* (ELBO), which will be used as the surrogate to the true log-likelihood in inference and learning. We have

$$\begin{aligned} \mathfrak{L} =& \sum_{n=1}^{N} \sum_{m=1}^{M} \phi_{nm} \left(\log \pi_m + \mathbb{E}_q[\log \lambda_{i_n,m}(t_n)] + \log p_m(W_n)\right) \\ & - \sum_{i=1}^{I} \sum_{m=1}^{M} \int_0^T \mathbb{E}_q[\lambda_{i,m}(s)] ds + \mathcal{E}[q]. \end{aligned} \tag{13}$$

The last term reduces to (*c.f.* Appendix 1):

$$\begin{aligned} & \sum_{i=1}^{I} \sum_{m=1}^{M} \int_0^T \mathbb{E}_q[\lambda_{i,m}(s)] ds \\ =& T \sum_{m=1}^{M} \sum_{i=1}^{I} \gamma_m \mu_i - \sum_{n=1}^{N} K(T - t_n) \sum_{i=1}^{I} \alpha_{i_n i}, \end{aligned} \tag{14}$$

where $K(t) = \int_0^t \kappa(s) ds$. The second term involves the expectation of the log-intensity $\mathbb{E}_q[\log(\gamma_m \mu_i + \sum_{t_l < t} Z_{lm} \alpha_{i_l i} k(t - t_l))]$. To break down the log-sum, we again apply Jensen's inequality

$$\begin{aligned} & \mathbb{E}_q[\log(\lambda_{i_n,m}(t_n))] \\ \geq& \eta_{nn}^m \log(\gamma_m \mu_{i_n}) + \sum_{\ell=1}^{n-1} \phi_{lm} \eta_{ln}^m \log(\alpha_{i_l i_n} k(t_n - t_l)) \\ & - \eta_{nn}^m \log(\eta_{nn}^m) - \sum_{l=1}^{n-1} \phi_{lm} \eta_{ln}^m \log(\eta_{ln}^m) \end{aligned} \tag{15}$$

where we have introduced a set of branching variables $\{\eta^m, m = 1, \ldots, M\}$. Note that each η^m is a $N \times N$ lower-triangular matrix with n-th row $\eta_n^m = [\eta_{1,n}^m, \ldots, \eta_{n,n}^m]^T$. The branching η also satisfies the following conditions:

$$\eta_{ln}^m \geq 0, \quad l = 1, \ldots, n \tag{16}$$

$$\eta_{nn}^m + \sum_{l=1}^{n-1} \phi_{lm} \eta_{ln}^m = 1 \tag{17}$$

[3] In our implement, we use $\Omega(Z, Z') = Z^\top Z'$.

Optimizing the Lagaragian of the ELBO, we have the following inference algorithm, with Z's inferred via

$$\begin{aligned}\phi_{nm} \propto\ & \pi_m \text{ : prior} \\ \times & \left(\prod_{v=1}^{V} \beta_{mv}^{w_{nv}}\right) \text{ : content} \\ \times & \left(\gamma_m \mu_{i_n}\right)^{\eta_{nn}^m} \text{ : self triggering} \\ \times & \prod_{l=1}^{n-1} \left(\alpha_{i_l i_n} \kappa(t_n - t_l)\right)^{\phi_{lm}\eta_{ln}^m} \text{ : influences from past} \\ \times & \prod_{l=n+1}^{N} \left(\alpha_{i_n i_l} \kappa(t_l - t_n)\right)^{\phi_{lm}\eta_{nl}^m} \text{ : influences to future,}\end{aligned} \tag{18}$$

where the branchings, η, are updated via

$$\eta_{nn}^m = \frac{\gamma_m \mu_{i_n}}{\gamma_m \mu_{i_n} + \sum_{l=1}^{n-1} \phi_{lm} \alpha_{i_l i_n} k(t_n - t_l)}, \tag{19}$$

$$\eta_{ln}^m = \frac{\alpha_{i_l i_n} k(t_n - t_l)}{\gamma_m \mu_{i_n} + \sum_{l=1}^{n-1} \phi_{lm} \alpha_{i_l i_n} k(t_n - t_l)}. \tag{20}$$

3.2. Learning

We derive maximum likelihood estimation for the mixture MHPs model with Eq(7) content model. The model involves five parameters, i.e., the self-instantaneous rate $\mu \in \mathbb{R}_+^I$, the per-meme infection rate $\gamma \in \mathbb{R}_+^M$, the infectivity (i.e., diffusion rate) matrix $\alpha \in \mathbb{R}_+^{I \times I}$, the regularization prior $\pi \in \mathbb{S}_M$, and the language model $\beta \in \mathbb{S}_V^M$, where $\mathbb{R}_+$ and $\mathbb{S}_K$ denote the nonnegative real domain and the K-simplex space $\{x | x \in \mathbb{R}_+^K \text{and} \sum_{k=1}^K x_k = 1\}$ respectively. The log-likelihood of the model, $\mathcal{L}(t, W | \mu, \gamma, \alpha, \beta, \pi)$, is lower-bounded by the ELBO $\mathfrak{L}$, which is tight when the variational parameters are optimal. The MLE for the model parameters can therefore be obtained by optimizing the ELBO upon the convergence of the variational inference. The update formulas are given as follows (*c.f.* Appendix 2):

$$\pi_m \propto \sum_{n=1}^{N} \phi_{nm} \tag{21}$$

$$\beta_{mv} \propto \sum_{n=1}^{N} w_{nv} \phi_{nm} \tag{22}$$

$$\mu_i = \frac{\sum_{m=1}^{M} \phi_{nm} \sum_{n=1}^{N} \delta_{i_n, i} \eta_{nn}^m}{\sum_{m=1}^{M} \gamma_m T} \tag{23}$$

$$\alpha_{ij} = \frac{\sum_{m=1}^{M} \phi_{nm} \phi_{lm} \sum_{n=1}^{N} \sum_{l=1}^{n-1} \delta_{i_l, i} \delta_{i_n, j} \eta_{ln}^m}{\sum_{n=1}^{N} \delta_{i_n, i} K(T - t_n)} \tag{24}$$

$$\gamma_m = \frac{\sum_{n=1}^{N} \phi_{nm} \eta_{nn}^m}{\sum_{i=1}^{I} \mu_i T} \tag{25}$$

where δ is the Kronecker's delta function.

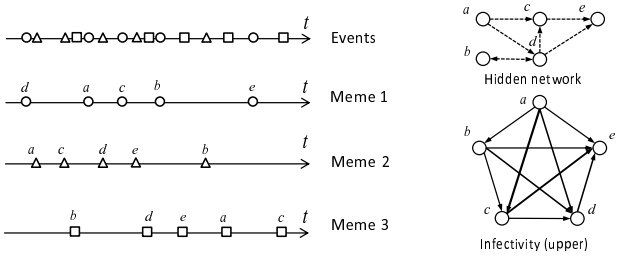


Figure 1. The successive event history (top-left) is segmented into multiple meme cascades (bottom-left), each of which induces a DAG; the structure of the hidden diffusion network (top-right) is inferred by estimating the infectivity matrix (bottom-right: only the upper triangular half is shown) from these DAGs.

3.3. Interpretation

The learning algorithm has two loops. The inner loop Eq(18) addresses meme-tracking, i.e., it *iteratively* infers the posterior of meme identities Z given the current network configurations. By inferring Z, it essentially segments the single successive event history into multiple cascades, one per meme. Each of these cascades induces a DAG, a small sub-graph of the DAG induced by the whole event sequence. The outer loop, i.e., the *closed-form* learning formulas Eq(21–25), further find the optimal network configuration α and other parameters based on the cascades inferred. This two-layer algorithm differs significantly from existing network inference techniques (Gomez-Rodriguez et al., 2011; Gomez-Rodriguez & Schölkopf, 2012; Snowsill et al., 2011) in that we segment cascades (i.e., infer memes) automatically whereas theirs rely on human annotations and/or external algorithms to do so.

The two algorithms are intuitively interpretable. The meme-tracking algorithm Eq(18) clusters events into memes based not only on the semantics of the viral contents but also on the evolution and propagation patterns. As marked in the formula, the meme identity of an event E_n is inferred by integrating five types of evidences: (1) Prior popularity of each meme; (2) Semantic clustering of the viral content (e.g., viruses with similar genetic structures will be grouped as a meme); (3) Spontaneous infection, i.e., how likely the meme was created by node i_n spontaneously; (4) Past diffusion: how the meme had been propagated before infecting i_n; (5) Future diffusion: how the meme would be propagated after infecting i_n. The algorithm resembles the mean-filed Boltzmann machine: starting from an initial meme configuration, it runs by cyclically updating each variational meme-assignment ϕ_n while keeping other ϕ's fixed until the *equilibrium configuration* is reached (i.e., all meme assignments are consistent with one another).

Table 1. Comparison of diffusion network models.

Model	Probabilistic	Continuous-time	Multiple memes	Exogenous factor	Viral contents	Sparsity	Meme-tracking	Convex
Pan	✓							
Mayers	✓	✓				✓		✓
Gomez	✓	✓				✓		✓
Ypma	✓	✓			✓			
Snowsill					✓	✓		
Ours	✓	✓	✓	✓	✓	✓	✓	

The network inference algorithm Eq(24) recovers the hidden diffusion network by estimating the infectivity matrix from the DAGs of the identified memes. The essential idea is to approximate a *causality* relationship from a set of *time causalities* (i.e., the meme DAGs), e.g., if Bob becomes infected *almost always* right after Alice's infections and *no one else* has as frequent time-causality with Bob as Alice, it is highly likely that Alice is an influencer to Bob. Figure 1 shows an illustrative example involving a five-node network.

3.4. Fast inference

The meme-tracking algorithm Eq(18) is quadratic in sequence length, i.e., $O(N^2)$ per iteration, which could be problematic since the event history could be extremely long (e.g., users post hundreds of millions of tweets per day on Twitter). The algorithm can be made linear by controlling the depth of the diffusion along the time axis. Particularly, because the decaying kernel $\kappa(\Delta t)$ approaches zero very fast as Δt goes large, it's not necessary to propagate the diffusion too far away. As such, we preset a maximum bandwidth L for the temporal diffusion and modify the last two terms in Eq(18) as follows:

$$\prod_{l=n-1}^{\max(0,n-L)} (\alpha_{i_l i_n} \kappa(t_n - t_l))^{\phi_{lm}\eta^m_{ln}} \quad : \text{influences from past}$$

$$\prod_{l=n+1}^{\min(N,n+L)} (\alpha_{i_n i_l} \kappa(t_l - t_n))^{\phi_{lm}\eta^m_{nl}} \quad : \text{influences to future}$$

Note that the branching variable η^m becomes a lower-triangular matrix with left bandwidth L.

By noticing that the infectivity matrix α is extremely sparse in practice (i.e., a node is usually connected to and can only infect a tiny fraction of nodes in a large network), the algorithm can be further speedup by hashing events of neighboring nodes and updating meme assignments based on these hashed influences.

3.5. Topological sparsity

Sparsity is an important property of real-world networks. Here we seek ways to improve the sparsity of the inferred diffusion network $\mathcal{G}$. Unfortunately, adding traditional regularization such as Lasso or ElasticNet penalty to the ELBO Eq(12) doesn't lead to sparse α (*c.f.* Appendix 4 for details). To this end, we modify the model by adding a network-independent diffusion rate ρ for each event:

$$\lambda^*_{i,m}(t) = \mu_i \gamma_m + Z_{lm} \sum_{t_l < t} (\alpha_{i_l i} + \rho) \kappa(t - t_l) \qquad (26)$$

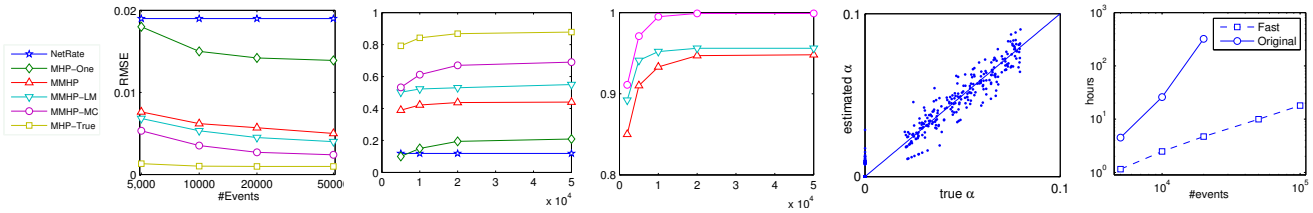
Essentially, we are assuming that each event has a minimum decaying diffusion rate allowing it to infect some nodes not directly through the network (e.g., nodes are infected occasionally by some relay nodes that are not included in the network). This modification yields

$$\alpha_{ij} = \left(\frac{\sum_{m,n} \sum_{l=1}^{n-1} \delta_{i_l,i} \delta_{i_n,j} \eta^m_{ln}}{\sum_n \delta_{i_n,i} K(T - t_n)} - \rho \right)^+ . \qquad (27)$$

Insignificant influence rates will be set automatically to exact zero. Note that Eq(18–20) also need to be slightly modified (i.e., replace α with $\alpha + \rho$).

4. Related work

Diffusion network inference and meme-tracking have been two important challenges in understanding viral diffusion in epidemiology, social science and many other disciplines (Wortman, 2008). Here we briefly introduce and compare the specs of network inference models. To date, there are only a few machine learning models for diffusion network inference in literature. (Pan et al., 2011) developed a coupled hidden Markov model for estimating symmetric social tie strengths from behavioral correlations. (Meyers & Leskovec, 2010) presented an approach for network inference using convex programming. (Gomez-Rodriguez et al., 2011; Gomez-Rodriguez & Schölkopf, 2012) presented probabilistic models for diffusion network based on survival and event history analysis and showed that the problem can be solved via submodular optimization. (Ypma et al., 2012) devised a classification model for estimating the transmission tree of diseases using both genetic and epidemiological data. (Snowsill et al., 2011) presented a suffix-tree based pattern matching algorithm for tracking text reuses and casted network inference as a set covering problem. The contributions of our work are three folds: (1) We introduce MHP as a principled framework for viral network inference; (2) We established probabilistic mixture of MHPs, a latent variable mixture model that addresses meme-tracking and network inference simultaneously; Compared with existing ones, our model



(a) (b) (c) (d) (e)

Figure 2. Experiments on synthetic data. (a–b): Comparison of network inference results in terms of RMSE and Rank Correlation. (c): An network estimated using `MMHP-MC` (88.2% of learned αs are zeros). (d): Comparison of meme-tracking accuracy. (e): Comparison of running time between the original variational algorithm and the fast algorithm.

doesn't require prior meme segmentation, or rely on human annotations/external algorithms to do so; (3) We model such important factors as the evolution of viral contents, exogenous effect due to external sources and topological sparsity of the network. Table 1 provides a brief comparison of network inference models (named after first authors) in terms of whether they: are probabilistic, model continuous time dynamics, work for multiple memes, model exogenous effect such as meme birth and external sources, model evolution of viral contents, encourage topological sparsity, do meme-tracking simultaneously, can be solved by convex optimization.

5. Experiments

5.1. On controlled synthetic data

Data. For given model dimensions (M, I, T), we drew random model parameters (μ, γ, α) and simulated events for each meme by running Ogata's modified thinning algorithm (Ogata, 1981). We mixed the memes into one sequence by ordering the events chronologically. Viral contents were sampled from randomly generated language models (one per meme). With probability 0.8, the pairwise mutation-from constraints were also simulated for events of the same meme, using the algorithm in (Snowsill et al., 2011). Experiments were done on 2,000–500,000 events involving $M = 50$ memes and $I = 50$ nodes.

Network recovery. We examine to what extent the model can recover the structure of the underlying network by measuring the difference between the learned infectivity matrix $\hat{\alpha}$ and the true α. We use *root mean square error* (RMSE) and *Kendall's rank correlation coefficient* (RCORR) as metrics. All experiments are run at least five times and the average results are reported. We evaluated the following three realistic models: `MMHP` denotes the plain mixture of MHPs model (i.e., without viral content model), `MMHP-LM` the one with language model and `MMHP-MC` with mutation constraints. For comparison, we also evaluated three baselines: (1) `MHP-True` is the model with meme identities $\{Z\}$ being known, which represents the upper-bound of the performance we can achieve; (2) `MHP-One` is the model assuming only one meme ($M = 1$), i.e., all meme identities $\{Z\}$ being equal, which represents the lower-bound performance; (3) the `NetRate` algorithm proposed in the recent work (Gomez-Rodriguez et al., 2011). Note that `NetRate` cannot model recurrent events so only the first occurrence is used.

In Figure 2(a–b), we plot the average metrics vs. the sequence length (i.e., number of events used in training) N. Because memes were not segmented a priori, the performance of `NetRate` is pretty poor, even worse than the lower-bound model. The three MMHP models substantially outperform `NetRate`; for both metrics, their performance lies between the lower bound and the upper bound. When N is moderately large, they all recover the hidden network satisfactorily, e.g., Figure 2(d) shows the network estimated from a $N = 20K$ sequence (the recovered network is very sparse, for 88.2% of the edges, the estimated infectivity rate $\hat{\alpha} = 0$). Note that on both metrics, the MMHPs with content models significantly outperforms the plain MMHPs, while `MMHP-MC` performs the best, (e.g., it approaches the upper bound as N increases).

Meme tracking. Next, we examine if the meme-tracking algorithm can segment memes from the mixture. We do so by comparing the inferred meme identities with the true labels. Note that the meme-tracking algorithm is unsupervised, i.e., the true labels were never seen in training. We assign meme labels based on the posterior score ϕ using a *winner-takes-all* scheme. Meme linkages (i.e., matching between the ground truth label set to the inferred label set) were done by linear programming using the simplex algorithm. The results were reported in Figure 2(c). Despite its unsupervised nature, the meme-tracking algorithm is surprisingly effective, e.g., all the three models can achieve accuracy of 90% and up; when data set is moderately large, the `MMHP-MC` model successfully segment memes with 100% accuracy.

Scalability. Figure 2(e) plots the running time curves of the standard variational algorithm and the fast ver-

1	search business deal microsoft billion yahoo pay buy google market
2	nba game lakers top season teams kobe sox howard win
3	honduras mark harriet global journey culture gilbert arts strand coles
4	oil hurricane european storm dollar china open tropical off bill
5	afghan killed pakistan taliban bomb kills iraq troops attack kabul
6	china iran obama russia minister president leader deal myanmar korea
7	fire ny killed nj ave dead plane crash injured hudson
8	sales profit uk loss rise prices london economy quarter june
9	obama medical health care house politics bill government plan reform
10	man police flu woman death swine murder charged court arrested

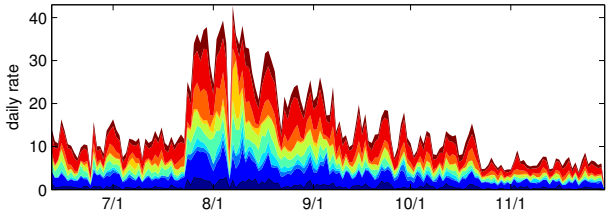


Figure 3. The ten top memes and their trends from mid Jun. to late Nov. 2009 identified by `MMHP-LM` on Twitter.

sion, both for the `MMHP` model. The fast algorithm is orders of magnitude faster and it scales linearly with data size. The speedup doesn't compromise much of the performance, e.g., the convergence and inference results are comparable to the standard one.

5.2. On Twitter & Blogosphere

We further apply the proposed model to viral text diffusion on Twitter and the blogosphere. The goal is to model real-world meme diffusion, particularly (1) identify memes (e.g., topics, ideas, behaviors) and track their trends; and (2) uncover the hidden graph over which memes transmit among in-world nodes (e.g., people, organizations). We use two fairly large public data sets[4]. Because the mutation-from relationships are unavailable for both data sets, we apply primarily the `MMHP-LM` model.

Meme tracking. Figure 3 shows the ten top memes (shown are the most representative terms of each meme) we identified from the Twitter data and their trends over time. Further analysis shows that these memes satisfactorily reflect the top trending events that occurred between mid June and late November in 2009 (e.g., the Microsoft-Yahoo! search deal, the Obama healthcare reform and the outbreak of the swine flue). Similar results were obtained on the Blogosphere data. Due to the unavailability of ground-truth, we were not able to conduct quantitative analysis of the meme tracking accuracy.

Network inference. The true transmission graphs for both data set are unknown. By tracking the information flow following the hyper-links between different blog-sites and constraining the edges with time-

[4]http://snap.stanford.edu/data

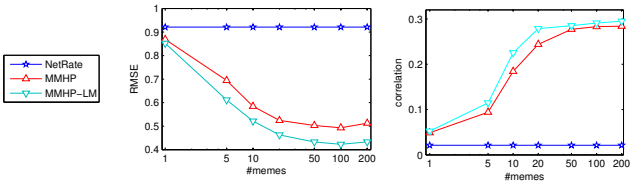


Figure 4. Network inference results on Blogosphere.

causality, we can roughly approximate the transmission graph for the blogosphere data. Figure 4 shows how the network inferred by different models (with varying number of latent memes, M) compare to this noisy ground-truth in terms of RMSE and RCorr. Again, we observe the `MMHP` models dramatically outperform the `NetRate` baseline.

6. Conclusions

Summary. We presented a viral diffusion model involving a mixture of MHPs, which addresses diffusion network inference and meme-tracking simultaneously and requires no human annotation or prior meme segmentation. Fast inference algorithms have been developed based on mean-filed method and experiments were conducted on both synthetic and real data.

Limitations of study and future work. The current work has some limitations which we plan to address in future work. (1) The current mixture model assumes the number of memes are fixed and known. The model can be enhanced by replacing the smoothing prior with a nonparametric grouping prior (e.g., Dirichlet process) that entities us identify the suitable number of clusters to automatically from data; (2) The meme tracking algorithm Eq(18) uses a fully factorized distribution to approximate the meme posteriors. The obtained meme trends are usually not fluent as transitions of memes among neighboring events are not captured. This can be improved by using a Markov chain model, the essential idea underlying Kalman filtering and hidden Markov model; (3) The current model cannot handle meme hybridizations since each event is assumed to belong to a single meme (although this assumption is relaxed in variational inference). To allow hybridization, we need mixture model for meme assignments. One suitable candidate is the Dirichlet-Multinomial model (Kvam & Day, 2001).

Software. Research code can be downloaded at: `http://www.cc.gatech.edu/~syang46/mhp.tar.gz`

Acknowledgements
This work was done while the first author was at Georgia Tech. We are grateful to Le Song and Ke Zhou for helpful discussions, and to NSF grant #IIS-1116886 for the support.

References

Bacrya, E., Delattreb, S., Hoffmannc, M. and Muzyd, J.F. Modeling microstructure noise with mutually exciting point processes. *Quantitative Finance*, 2012.

Blei, D.M. and Lafferty, J.D. Dynamic topic models. In *ICML 2006*, pp. 113–120, 2006.

Crane, R. and Sornette, D. Robust dynamic classes revealed by measuring the response function of a social system. *Proc. of the National Academy of Sciences*, 105:15649–15653, 2008.

Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. Uncovering the temporal dynamics of diffusion networks. In *ICML 2011*, pp. 561–568, 2011.

Gomez-Rodriguez, M. and Schölkopf, B. Submodular inference of diffusion networks from multiple trees. In *ICML 2012*, 2012.

Guttorp, P. and Thorarinsdottir, T.L. Bayesian inference for non-markovian point processes. In *Advances and Challenges in Space-time Modelling of Natural Events*, pp. 79–102, 2012.

Hawkes, A.G. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90, 1971.

Kvam, P. and Day, D. The multivariate Polya distribution in combat modeling. *Naval Research Logistics*, 48:1–17, 2001.

Lehmann, J., Gonçalves, B., Ramasco, J., and Cattuto, C. Dynamical classes of collective attention in twitter. In *World Wide Web 2012*, pp. 251–260, 2012.

Leskovec, J., Backstrom, L., and Kleinberg, J. Meme-tracking and the dynamics of the news cycle. In *ACM SIGKDD 2009*, pp. 497–506, 2009.

Liniger, T. Multivariate hawkes processes. *ETH Doctoral Dissertation No. 18403*, 2009.

Meyers, S. and Leskovec, J. On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems 26*, 2010.

Mohler, G.O., Short, M.B., Brantingham, P.J., Schoenberg, F.P., and Tita, G. E. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

Ogata, Y. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, Jan 1981.

Pan, W., Dong, W., Cebrian, M., Kim, T., and Pentland, A. Modeling dynamical influence in human interaction. *MIT Media Lab technical report* March 2011.

Ryu, H., Lease, M., and Woodward, N. Finding and exploring memes in social media. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, HT '12, pp. 295–304, 2012.

Snowsill, T. M., Fyson, N., De Bie, T., and Cristianini, N. Refining causality: who copied from whom? In *ACM SIGKDD 2011*, pp. 466–474, 2011.

Wortman, J. Viral marketing and the diffusion of trends on social networks. *UPenn Technical Report MS-CIS-08-19*, May 2008.

Ypma, RJ, Bataille, AM, Stegeman, A, Koch, G, Wallinga, J, and van Ballegooijen, WM. Unravelling transmission trees of infectious diseases by combining genetic and epidemiological data. *Proc. of the Royal Society B*, 279:444–450, Feb 2012.