Multi-Task Multi-Dimensional Hawkes Processes for Modeling Event Sequences

Dixin Luo[1]*, Hongteng Xu[2]*, Yi Zhen[2], Xia Ning[3], Hongyuan Zha[2,4], Xiaokang Yang[1], Wenjun Zhang[1]
[1]SEIEE, Shanghai Jiao Tong University, Shanghai, China
[2]College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
[3]Department of Computer and Information Science, IUPUI, Indianapolis, IN, USA
[4]Software Engineering Institute, East China Normal University, Shanghai, China

Abstract

We propose a Multi-task Multi-dimensional Hawkes Process (MMHP) for modeling event sequences where there exist multiple triggering patterns within sequences and structures across sequences. MMHP is able to model the dynamics of multiple sequences jointly by imposing structural constraints and thus systematically uncover clustering structure among sequences. We propose an effective and robust optimization algorithm to learn MMHP models, which takes advantage of alternating direction method of multipliers (ADMM), majorization minimization and Euler-Lagrange equations. Our experimental results demonstrate that MMHP performs well on both synthetic and real data.

1 Introduction

In practical information systems (e.g., social networks, financial systems, IPTV systems), a temporal event sequence of a certain user can be modeled as a point process [Rodriguez *et al.*, 2011; Yang and Zha, 2013; Linderman and Adams, 2014; Li and Zha, 2014]. Typically, an event in the sequence may trigger a series of later events, forming a triggering pattern of the event sequence. On the other hand, multiple event sequences may exhibit similar triggering patterns and therefore can be characterized as a cluster. For example, in an IPTV system, users buy and watch various TV programs, and their watching behaviors (when and what they watch) form a large number of temporal event sequences. For each user, watching an early episode of a drama may trigger the events of watching its following episodes (i.e., self-triggering) and other related news (i.e., mutual-triggering). The triggering pattern among watching behaviors reflects the watching preferences of the user. Furthermore, the sequences of the users having similar preferences can be clustered according to the similarity of their triggering patterns. Simultaneously learning the triggering pattern of each individual event sequence and the clustering structure across all the sequences has great practical significance, because it enables both local and global depictions of the entire dynamic system. However, such a learning task is very challenging as it demands concurrent modeling point processes individually and globally.

Mathematically, suppose that each user's behaviors can be represented as a event sequence $\{(t_1, E_1), ..., (t_N, E_N)\}$ $(0 < t_i \leq T)$. The event E_i occurs at time t_i, which is an element of a event set $\mathcal{E} = \{1, ..., C\}$. Given a set of event sequences from different users, we aim to 1) find the triggering pattern among events for each user, including the influence of the event c on the event c', $c, c' \in \mathcal{E}$, denoted as $\boldsymbol{A}^u = [a_{cc'}] \in \mathbb{R}_+^{C\times C}$ and the temporal dynamic of the influence; 2) explore the clustering structure of $\boldsymbol{A}^u$'s.

In this paper, we propose a novel point process model, namely Multi-task Multi-dimensional Hawkes Process (MMHP), to learn triggering patterns and clustering structures from a number of event sequences. Specifically, given a set of temporal event sequences, MMHP models the corresponding dynamic system using an intrinsic intensity matrix, a structured infectivity tensor and a triggering kernel. The intrinsic intensity matrix captures the basic instantaneous happening rate of various events; the tensor represents the infectivity among events, which reveals the triggering pattern among events; the triggering kernel measures the time decay effect of the infectivity. As the learning of event sequences from the same cluster should share implicit relatedness, we learn our model under the multi-task learning framework [Evgeniou and Pontil, 2007; Liu *et al.*, 2009]. We impose sparse and low rank constraints on the infectivity tensor so as to induce the clustering structure among sequences, and propose an effective algorithm that takes advantage of alternating direction method of multipliers (ADMM), majorization minimization and Euler-Lagrange equations to solve the optimization problem.

Different from traditional multi-dimensional Hawkes process models, which only consider the triggering pattern within sequences, MMHP considers both the triggering pattern within sequences and the clustering structure across sequences. In MMHP, the triggering patterns of sequences and their similarities are captured by the structured infectivity tensor. An advantage of MMHP is that they can avoid overfitting for the sequences where only a few events are observed. For those cases, the structure of the infectivity ten-

*These two authors contribute equally.

sor facilitates the knowledge transfer from sequences with many observed events to those short of observations. We evaluate MMHP and compare it with other state-of-the-art methods. Our experimental results on both synthetic and real datasets demonstrate the superior performance and robustness of MMHP.

2 Related Work

Point processes. Point processes [Daley and Vere-Jones, 2007] are popular models for sequential and temporal data, which have been successfully applied to model the occurrences of earthquakes [Ogata, 1988; 1999], the transactions in the stock market [Chavez-Demoulin and McGill, 2012; Bacry *et al.*, 2013], asset management [Yan *et al.*, 2013], meme tracking [Yang and Zha, 2013] and user interactions [Zhou *et al.*, 2013a] on social networks. A point process is typically represented as an event sequence $\{(t_1, E_1), ..., (t_N, E_N)\}$ $(0 < t_i \leq T)$, where the event E_i occurs at time t_i. Denote $N(t)$ as the number of points (i.e., events) happening in the time interval $(-\infty, t]$ and $\mathcal{H}_t = \{E_i | t_i < t\}$ as the set of events happening before t, a point process is characterized by its conditional intensity function

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}(N(t + \Delta t) - N(t) | \mathcal{H}_t)}{\Delta t} = \frac{\mathbb{E}(dN(t) | \mathcal{H}_t)}{dt},$$

where $\mathbb{E}(dN(t)|\mathcal{H}_t)$ is the expectation of the number of events happening in the interval $(t, t + dt]$ given historical observations $\mathcal{H}_t$. The conditional intensity function represents the expected instantaneous rate of events at time t.

Hawkes processes. A Hawkes process [Hawkes, 1971; Hawkes and Oakes, 1974] is a point process having a self-triggering property, that is, the occurrences of previous events trigger the occurrences of future events. Its conditional intensity function is defined as follows,

$$\lambda(t) = \mu + \sum\nolimits_{j: t_j < t} g(t - t_j), \tag{1}$$

where $\mu \in \mathbb{R}_+$ is an *intrinsic intensity* for the occurrences of events, $g : \mathbb{R}_+ \to \mathbb{R}_+$ is a *triggering kernel* function quantifying the triggering effects from previous events.

Multi-dimensional Hawkes processes. In most situations, there can be multiple types of events happening sequentially, e.g., multiple users post information and interact with each other on a social network, multiple items are sold on an online shop. In those cases, each event not only triggers the events of its type, but also triggers the events of other different types. Multi-dimensional Hawkes processes are used to model such processes. Specifically, given C types of events, the conditional intensity $\boldsymbol{\lambda}(t) = [\lambda_1(t), ..., \lambda_C(t)]^\top$ is a size-C vector, where $\lambda_c(t)$ is the conditional intensity for type-c events defined as follows,

$$\lambda_c(t) = \mu_c + \sum\nolimits_{j: t_j < t} a_{cc_j} g(t - t_j), \tag{2}$$

where μ_c is the intrinsic intensity of the event of type c. Compared with (1), an *infectivity* matrix $\boldsymbol{A} \in \mathbb{R}_+^{C \times C}$ is introduced to measure the influence across events of different types, that is, $a_{cc'}$ represents the infectivity of type-c' events to type-c events. $\sum_{j: t_j < t} a_{cc_j} g(t - t_j)$ represents the influence of historical events on the instantaneous rate of event at time t.

Multi-dimensional Hawkes processes have been proposed and applied to analyze the topic diffusion [Rodriguez *et al.*, 2011; Du *et al.*, 2013; Yang and Zha, 2013] and the user interactions [Zhou *et al.*, 2013a; Blundell *et al.*, 2012] on social networks, the transactions in the stock market [Chavez-Demoulin and McGill, 2012; Bacry *et al.*, 2013], etc. In these works, the event types can correspond to topics, users, transaction types and any other objects. Among these works, Blundell *et al.* proposed a multi-dimensional Hawkes processes with the Infinite Relational Model (IRM) [Blundell *et al.*, 2012] to simulate and predict the social interactions among users on social networks. Zhou *et al.* proposed a multi-dimensional Hawkes process model, which learns an infectivity matrix of users explicitly with sparse and low-rank constraints [Zhou *et al.*, 2013a]. Furthermore, the triggering kernel of the model is learned by nonparametric estimation in [Zhou *et al.*, 2013b]. Yang *et al.* proposed to use multi-dimensional Hawkes process to perform diffusion network inference and meme tracking jointly [Yang and Zha, 2013]. Recently, an online learning algorithm for multi-dimensional Hawkes processes is proposed in [Hall and Willett, 2014], which approximates continuous Hawkes processes in a discrete manner. Multi-dimensional Hawkes processes have achieved promising results in many challenging tasks. However, most of the existing works focus on learning triggering patterns of sequences while few of them consider the clustering structure across sequences. The recent works in [Du *et al.*, 2013; Li and Zha, 2014; Linderman and Adams, 2014] start to explore the relationship among sequences by learning parametric models. Different from these works, we take advantage of the multi-task learning strategy [Evgeniou and Pontil, 2007; Liu *et al.*, 2009; Jacob *et al.*, 2009] and add structural regularization directly to the proposed model.

3 Multi-task Multi-dimensional Hawkes Processes

We propose the following **Multi-task Multi-dimensional Hawkes Process (MMHP)** models to learn triggering patterns of each event sequence and structures across various sequences jointly, where each event sequence includes multiple types of events. Given U sequences with C event types, we represent the uth event sequence as $\mathcal{S}_u = \{(t_i^u, c_i^u)\}_{i=1}^{n_u}$, where $u = 1, 2, \cdots, U$. $c_i^u \in \{1, ..., C\}$ and $t_i^u \in (0, T_u]$ are the event type and time stamp of the i-th event in $\mathcal{S}_u$, respectively. T_u is the time span of $\mathcal{S}_u$, and n_u is the number of events in $\mathcal{S}_u$. Each sequence $\mathcal{S}_u$ can be modeled by a multi-dimensional Hawkes process. Based on (2), the conditional intensity function for $\mathcal{S}_u$ on type-c event at time t is as follows,

$$\lambda_c^u(t) = \mu_c^u + \sum\nolimits_{j: t_j^u < t} a^u_{cc_j^u} g(t - t_j^u). \tag{3}$$

Different from the conventional multi-dimensional Hawkes processes as in (2), the model parameters in (3) fall in two categories: 1) a *global* triggering kernel $g(t)$, which reflects the attenuating influential effects from historical events and is shared by all the sequences; 2) a *local* infectivity matrix $\boldsymbol{A}^u = [a^u_{cc'}]$ and a *local* natural intensity vector $\boldsymbol{\mu}^u = [\mu^u_1, ..., \mu^u_C]^\top$, which are specific to sequence $\mathcal{S}_u$. We represent such $\boldsymbol{A}^u$'s and $\boldsymbol{\mu}^u$'s into an infectivity tensor $\mathcal{A} = [a^u_{cc'}] \in \mathbb{R}^{C\times C\times U}_+$ and an intrinsic intensity matrix $\boldsymbol{\mu} = [\mu^u_c] \in \mathbb{R}^{C\times U}_+$, respectively.

3.1 Structural Constraints on MMHPs

We impose structural constraints on the model parameters in (3) so as to enable pattern learning across sequences. Specifically, we impose sparse and low-rank constraints on the flattened matrix $\boldsymbol{A} = [\mathrm{vec}(\boldsymbol{A}^1), ..., \mathrm{vec}(\boldsymbol{A}^U)] \in \mathbb{R}^{C^2\times U}$ from tensor $\mathcal{A}$. The sparsity constraint is based on the observations that typically within each sequence, only a subset of event types happens and triggers others. The intuition behind the low-rank constraint is to uncover clustering structures when the overall triggering patterns of multiple sequences in terms of their infectivity effects are similar and thus introduce similar $\boldsymbol{A}^u$'s. Here we assume that the similarity is described by the self-representation property of infectivity matrices, which implies the low-rank structure of $\boldsymbol{A}$. These two structural constraints are reasonable for practical systems. For example, in an IPTV system, each user typically has a preference over a small number of program categories, and such a preference can be common among a large number of users.

3.2 Learning Algorithms for MMHPs

We learn our MMHP model via Maximum Likelihood Estimation (MLE). Specifically, we learn the parameters $\boldsymbol{\mu}$, $\boldsymbol{A}$ and $g(t)$ by solving the following optimization problem.

$$\begin{aligned} \min_{\boldsymbol{\mu},\boldsymbol{A},g} \quad & \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{A}, g) + \alpha\mathcal{R}(g) + \lambda_1\|\boldsymbol{A}\|_* + \lambda_2\|\boldsymbol{A}\|_1 \\ s.t. \quad & \boldsymbol{A} \geq \boldsymbol{0},\ \boldsymbol{\mu} \geq \boldsymbol{0},\ g(t) \geq 0, \end{aligned} \tag{4}$$

where $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{A}, g)$ is the negative log-likelihood of MMHP that can be written as:

$$\begin{aligned} \mathcal{L} =& -\sum_{u=1}^{U}\left(\sum_{i=1}^{n_u}\log\lambda^u_{c^u_i}(t^u_i) - \sum_{c=1}^{C}\int_0^{T_u}\lambda^u_c(t)dt\right) \\ =& -\sum_{u=1}^{U}\Bigg[\sum_{i=1}^{n_u}\log\Bigg(\mu^u_{c^u_i} + \sum_{j:t^u_j<t^u_i} a^u_{c^u_i c^u_j} g(t^u_i - t^u_j)\Bigg) \\ & - T_u\sum_{c=1}^{C}\mu^u_c - \sum_{c=1}^{C}\sum_{i=1}^{n_u} a^u_{cc^u_i}\int_0^{T-t^u_i} g(t)dt\Bigg], \end{aligned} \tag{5}$$

The second term $\mathcal{R}(g)$ in (4), defined as

$$\mathcal{R}(g) = \int_0^\infty [g'(t)]^2 dt,$$

regularizes the triggering kernel [Zhou *et al.*, 2013b] to ensure the triggering kernel is smooth and differential energy limited (i.e., $\int_0^\infty [g'(t)]^2 dt < \infty$). The nuclear norm and ℓ_1 norm on $\boldsymbol{A}$ in (4) impose low rank and sparsity on $\boldsymbol{A}$.

To solve the optimization problem (4), we apply the scheme of ADMM [Boyd *et al.*, 2011; Ouyang *et al.*, 2013; Zhou *et al.*, 2013a] and introduce two auxiliary variables Z_1 and Z_2, two dual variables U_1 and U_2, and thus convert the problem in (4) as follows.

$$\begin{aligned} \min \quad & \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{A}, g) + \alpha\mathcal{R}(g) + \lambda_1\|\boldsymbol{Z}_1\|_* + \lambda_2\|\boldsymbol{Z}_2\|_1 \\ & + \rho(tr(\boldsymbol{U}_1^\top(\boldsymbol{A} - \boldsymbol{Z}_1))) + \rho(tr(\boldsymbol{U}_2^\top(\boldsymbol{A} - \boldsymbol{Z}_2))) \\ & + \frac{\rho}{2}(\|\boldsymbol{A} - \boldsymbol{Z}_1\|_F^2 + \|\boldsymbol{A} - \boldsymbol{Z}_2\|_F^2) \\ s.t. \quad & \boldsymbol{A} \geq \boldsymbol{0},\ \boldsymbol{\mu} \geq \boldsymbol{0},\ g(t) \geq 0. \end{aligned} \tag{6}$$

We solve the problem in (6) by an iterative algorithm, which updates $\boldsymbol{A}$ and $\boldsymbol{\mu}$, $g(t)$, Z_1 and Z_2, U_1 and U_2 iteratively.

Step 1: Update $\boldsymbol{A}$ and $\boldsymbol{\mu}$. We first update $\boldsymbol{A}$ and $\boldsymbol{\mu}$ by a majorization-minimization algorithm. Given the parameters of k-th iteration as $\Theta^{(k)}$, we find a surrogate function of objective function by Jensen's inequality as follows.

$$\begin{aligned} &\mathcal{Q}(\Theta|\Theta^{(k)}) = \\ &-\sum_{u=1}^{U}\Bigg[\sum_{i=1}^{n_u}\Bigg(p^u_{ii}\log\frac{\mu^u_{c^u_i}}{p^u_{ii}} + \sum_{j=1}^{i-1}p^u_{ij}\log\frac{a^u_{c^u_i c^u_j}g(t^u_i - t^u_j)}{p^u_{ij}}\Bigg) \\ &- T_u\sum_{c=1}^{C}\mu^u_c - \sum_{c=1}^{C}\sum_{i=1}^{n_u}\int_0^{T-t^u_i}\Bigg((a^u_{cc^u_i})^2\frac{g^{(k)}(t)}{2a^{u(k)}_{cc^u_i}} \\ &+ (g(t))^2\frac{a^{u(k)}_{cc^u_i}}{2g^{(k)}(t)}\Bigg)dt\Bigg] + \frac{\rho}{2}(\|\boldsymbol{A} - \boldsymbol{Z}_1^{(k)} + \boldsymbol{U}_1^{(k)}\|_F^2 \\ &+ \|\boldsymbol{A} - \boldsymbol{Z}_2^{(k)} + \boldsymbol{U}_2^{(k)}\|_F^2) + \alpha\mathcal{R}(g). \end{aligned} \tag{7}$$

where

$$p^u_{ii} = \frac{\mu^{u(k)}_{c^u_i}}{\mu^{u(k)}_{c^u_i} + \sum_{j=1}^{i-1} a^{u(k)}_{c^u_i c^u_j} g(t^u_i - t^u_j)},$$

$$p^u_{ij} = \frac{a^{u(k)}_{c^u_i c^u_j} g(t^u_i - t^u_j)}{\mu^{u(k)}_{c^u_i} + \sum_{j=1}^{i-1} a^{u(k)}_{c^u_i c^u_j} g(t^u_i - t^u_j)}.$$

Considering the terms related to $\{\boldsymbol{\mu}, \boldsymbol{A}\}$ and setting $\frac{\partial Q}{\partial \mu^u_c} = 0$ and $\frac{\partial Q}{\partial a^u_{cc'}} = 0$, we obtain close-form solutions for $\boldsymbol{\mu} = [\mu^u_c]$ and $\boldsymbol{A} = [a^u_{cc'}]$:

$$\mu^{u(k+1)}_c = \frac{\sum_{i:c^u_i=c} p^u_{ii}}{T_u}, \tag{8}$$

$$a^{u(k+1)}_{cc'} = \frac{-B + \sqrt{B^2 - 4AC}}{2A}, \tag{9}$$

where

$$\begin{aligned} A =& 2\rho a^{u(k)}_{cc'} + \sum\nolimits_{i:c^u_i=c'}\int_0^{T-t^u_i} g(t)dt, \\ B =& \rho a^{u(k)}_{cc'}(u^{u(k)}_{1,cc'} - z^{u(k)}_{1,cc'} + u^{u(k)}_{2,cc'} - z^{u(k)}_{2,cc'}), \\ C =& -a^{u(k)}_{cc'}\sum\nolimits_{i:c^u_i=c}\sum\nolimits_{j:j<i,c^u_j=c'} p^u_{ij}, \end{aligned}$$

$z^{u(k)}_{cc'}$ and $u^{u(k)}_{cc'}$ are the elements in $\boldsymbol{Z}^{u(k)}$ and $\boldsymbol{U}^{u(k)}$ *w.r.t.* $a^u_{cc'}$, respectively.

Step 2: Update $g(t)$. The triggering kernel g can be updated in an infinite dimensional space [Zhou *et al.*, 2013b]. Considering the terms of (7) related to $g(t)$, the solution of $g(t)$ satisfies the Euler-Lagrange equation:

$$-\frac{D(t)}{g(t)} + F(t)g(t) - 2\alpha g''(t) = 0, \tag{10}$$

where $g''(t)$ is the second-order derivative of $g(t)$ and

$$F(t) = \sum\nolimits_{u=1}^{U} \sum\nolimits_{c=1}^{C} \sum\nolimits_{i=1}^{n_u} \frac{a_{cc_i^u}^{u(k)}}{g^{(k)}(t)} \mathbb{I}(t < T_u - t_i^u),$$
$$D(t) = \sum\nolimits_{u=1}^{U} \sum\nolimits_{i=1}^{n_u} \sum\nolimits_{j=1}^{i-1} p_{ij}^u \mathbb{I}(t = t_i^u - t_j^u).$$

$\mathbb{I}(\cdot)$ is the indicator function which returns 1 if the input predicate is true and 0 otherwise. We solve (10) numerically using the following efficient Seidel-type iterations. Specifically, setting the sampling interval as Δt, we discretize the differential equation over small intervals $m\Delta t$, for $m = 1, ..., M$, as follows:

$$-\frac{D_m}{g_m} + F_m g_m - 2\alpha \frac{g_{m+1} + g_{m-1} - 2g_m}{\Delta t^2} = 0, \tag{11}$$

where $g_m = g(m\Delta t)$, $F_m = F(m\Delta t)$ and $D_m = D(m\Delta t)$. M is the number of samples of g and $M\Delta t$ is the length of g. Therefore, we can solve for g_m by fixing all other $g_{m'}$, $m' \neq m$ and solving the above quadratic equation.

Step 3: Update $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$. Given the updated $\boldsymbol{A}^{(k+1)}$ from **Step 1**, we solve for $\boldsymbol{Z}_1$ by solving the following optimization problem.

$$\min_{\boldsymbol{Z}_1} \; \lambda_1 \|\boldsymbol{Z}_1\|_* + \rho(tr((\boldsymbol{U}_1^{(k)})^\top (\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_1)))$$
$$+ \frac{\rho}{2} \|\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_1\|_F^2.$$

The solution is obtained by shrinking the singular values of $\boldsymbol{A}^{(k+1)} + \boldsymbol{U}_1^{(k)}$ by soft-thresholding [Zhou *et al.*, 2013a] as

$$\boldsymbol{Z}_1^{(k+1)} = S_{\lambda_1/\rho}(\boldsymbol{A}^{(k+1)} + \boldsymbol{U}_1^{(k)}), \tag{12}$$

where $S_{\lambda_1/\rho}(\cdot)$ is the soft-thresholding function with the threshold λ_1/ρ. Similarly, we solve for $\boldsymbol{Z}_2$ by solving the following optimization problem,

$$\min_{\boldsymbol{Z}_2} \; \lambda_2 \|\boldsymbol{Z}_2\|_1 + \rho(tr((\boldsymbol{U}_2^{(k)})^\top (\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_2)))$$
$$+ \frac{\rho}{2} \|\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_2\|_F^2,$$

and the solution is

$$\boldsymbol{Z}_2^{(k+1)} = E_{\lambda_2/\rho}(\boldsymbol{A}^{(k+1)} + \boldsymbol{U}_2^{(k)}), \tag{13}$$

where $E_{\lambda_2/\rho}(\cdot)$ is the soft-thresholding function of matrix's elements. The threshold is λ_2/ρ.

Step 4: Update $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$. The dual variables $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ are solved as follows based on ADMM.

$$\boldsymbol{U}_1^{(k+1)} = \boldsymbol{U}_1^{(k)} + (\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_1^{(k+1)}), \tag{14}$$
$$\boldsymbol{U}_2^{(k+1)} = \boldsymbol{U}_2^{(k)} + (\boldsymbol{A}^{(k+1)} - \boldsymbol{Z}_2^{(k+1)}). \tag{15}$$

The whole learning algorithm is summarized in Algorithm 1.

Algorithm 1 MMHP Learning Algorithm

```
Input: event sequences $\{\mathcal{S}_u\}$, parameters $\{\lambda_1, \lambda_2, M, \alpha\}$
Output: $\boldsymbol{A}$, $\boldsymbol{\mu}$ and $g(t)$
  Initialize $\boldsymbol{\mu} \in \mathbb{R}_+^{C\times U}$, $\boldsymbol{A} \in \mathbb{R}_+^{C^2\times U}$, $g \in \mathbb{R}_+^M$ randomly.
  $\boldsymbol{Z}_1^{(0)} = \boldsymbol{Z}_2^{(0)} = \boldsymbol{A}^{(0)}$, $\boldsymbol{U}_1^{(0)} = \boldsymbol{U}_2^{(0)} = \boldsymbol{0}$, $k = 0$
  repeat
    $k = k + 1$
    Update $\{\boldsymbol{\mu}^{(k)}, \boldsymbol{A}^{(k)}\}$ by (8) and (9)
    repeat
      for $m = 1 : M$ do
        Update $g_m^{(k)}$ by (11)
      end for
    until convergence
    Update $\boldsymbol{Z}_1^{(k)}$, $\boldsymbol{Z}_2^{(k)}$ by (12) and (13)
    Update $\boldsymbol{U}_1^{(k)}$, $\boldsymbol{U}_2^{(k)}$ by (14) and (15)
  until convergence
  $\boldsymbol{A} = \boldsymbol{A}^{(k)}$, $\boldsymbol{\mu} = \boldsymbol{\mu}^{(k)}$, $g(t) = \{g_m^{(k)} | m = 1 : M\}$
```

4 Experimental Results

We evaluate the performance of MMHP on both synthetic and real-world data. Specifically, we compare the learning algorithm MMHP in Algorithm 1 with the following alternatives:

- Full: the infectivity tensor $\mathcal{A}$ has no structures (i.e., $\lambda_1 = \lambda_2 = 0$ in (4)). This method is similar to that in [Zhou *et al.*, 2013b], where they learn triggering patterns for each sequence independently.
- Sparse: only the sparsity constraint is imposed on $\mathcal{A}$ (i.e., $\lambda_1 = 0$ in (4)).
- LowRank: only the low-rank constraint is imposed on $\mathcal{A}$ (i.e., $\lambda_2 = 0$ in (4)).

We use the following metrics to evaluate the performance of various methods:

- ***LogLik***: the log-likelihood of testing data using the trained model.
- ***EstErr***: the averaged estimation error of instantaneous infectivity $a_{cc'}^u g(t)$, defined as

  $$\textbf{\textit{EstErr}} = \frac{1}{UC^2} \sum_{u=1}^{U} \sum_{c,c'=1}^{C} \int_0^\infty [a_{cc'}^u g(t) - \hat{a}_{cc'}^u \hat{g}(t)]^2 dt,$$

  where $\{\widehat{\boldsymbol{A}}, \hat{g}(t)\}$ represents real parameters and $\{\boldsymbol{A}, g(t)\}$ represents the corresponding estimates.
- ***RankCorr***: the averaged Kendall's rank correlation coefficient between each row of the real $\widehat{\boldsymbol{A}}$ and that of the estimated $\boldsymbol{A}$.
- ***ClusAcc***: the clustering accuracy, defined as the percentage of sequences clustered correctly based on the learned infectivity tensor. It is only used for synthetic data.
- ***ClusDiff***: a metric of clustering accuracy for real-world data, where the ground truth of parameters and the clustering indices are unavailable. In specific, we first cluster sequences by applying k-means [Zha *et al.*, 2001; Ng *et al.*, 2002] on the estimated $\mathcal{A}$. After computing centers of clusters from estimated $\mathcal{A}$, we then construct a clustered infectivity tensor $\widetilde{\mathcal{A}}$, where the u-th slide $\widetilde{\boldsymbol{A}}^u$ is the center of the

cluster the $\boldsymbol{A}^u$ belongs to. Denote the set of possible methods as $\mathcal{I}$ (Here $\mathcal{I} = \{\mathsf{Full}, \mathsf{Sparse}, \mathsf{LowRank}, \mathsf{MMHP}\}$). For a certain method, ***ClusDiff*** measures the difference of the log-likelihood calculated from the ever best performing method and the log-likelihood calculated using the constructed $\widetilde{\mathcal{A}}$ from that method, that is, ***ClusDiff*** of the i-th method is defined as

$$\textbf{\textit{ClusDiff}}(i) = \max_{j \in \mathcal{I}} \textbf{\textit{LogLik}}(\mathcal{A}_j) - \textbf{\textit{LogLik}}(\widetilde{\mathcal{A}}_i).$$

If a clustering result is good, then each cluster center should be representative for capturing the dynamics of the sequences belonging to the cluster, and the difference of log-likelihood caused by replacing the specific infectivity matrices with the cluster centers should be small.

4.1 Experimental Results on Synthetic Data

We generate a synthetic dataset in which there are $U = 40$ sequences and $C = 5$ event types. The sequences are generated so as to fall into two clusters of equal size. We generate the flatten version of the infectivity tensor as $\boldsymbol{A} = [\text{thres}(\boldsymbol{u}_1\boldsymbol{v}_1^\top), \text{thres}(\boldsymbol{u}_2\boldsymbol{v}_2^\top)]$, where $\boldsymbol{u}_1, \ \boldsymbol{u}_2 \in \mathbb{R}_+^{C^2}$ and $\boldsymbol{v}_1, \ \boldsymbol{v}_2 \in \mathbb{R}_+^{U/2}$ are four random vectors with values uniformly distributed over $[0, 1]$. The function thres$(\boldsymbol{X})$ randomly sets half of the rows in $\boldsymbol{X}$ as zero. In this way, the generated $\boldsymbol{A}$ is low-rank and sparse, and inherently represents two clusters. We generate an intensity matrix $\boldsymbol{\mu}$ from a uniform distribution over $[0, 0.001]$. We use an exponential kernel $g(t) = \exp(-t)$ with $t \in (0, 20]$ as the triggering kernel for the synthetic data. Given the above parameters, we simulate 100 training sequences and 100 testing sequences respectively. Each training sequence contains 2500 events, and each testing sequence contains 500 events.

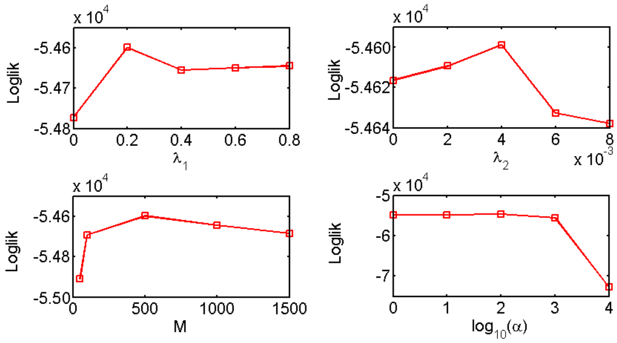


Figure 1: Parameter Study for MMHP.

Parameter Studies. We first conduct a parameter study on the four parameters $\{\lambda_1, \ \lambda_2, \ M, \ \alpha\}$ involved in our MMHP algorithm. The parameters λ_1 and λ_2 control the rank and sparsity level of the infectivity tensor, and M and α control the sampling rate and the smoothness of the triggering kernel. We first identify the optimal parameter configuration by doing a grid search in the parameter space of $\lambda_1 \in [0, 0.8]$, $\lambda_2 \in [0, 0.008]$, $M \in [50, 1500]$ and $\alpha \in [1, 10000]$ on a training set where each sequence has 500 events. With the identified optimal configuration as $\lambda_1 = 0.2$, $\lambda_2 = 0.004$, $M = 500$, $\alpha = 100$, we fix 3 parameters as their optimal values each time and alter the fourth parameter to train a different MMHP model. Fig. 1 represents the ***LogLik*** of such MMHPs as the different parameters vary. As the value of λ_1 or λ_2 grows larger, the ***LogLik*** of MMHPs first increases and then decreases, and this demonstrates the effectiveness of λ_1 and λ_2 in controlling the rank and sparsity of the infectivity tensor, respectively, and thus the model quality. Similar trends apply for M corresponding to the fact that extremely small M leads to a coarse estimation of the triggering kernel while extremely large M leads to over-fitting. The performance is relatively stable for $\alpha \in [1, 1000]$ as for larger α values, the strong smoothness regularization leads to an over-smoothed triggering kernel. Overall, the relatively small performance changes around the optimal parameter configuration demonstrate the robustness of our algorithm with respect to its parameters.

Performance Comparisons. We compare MMHP with the Full, Sparse and $\mathsf{LowRank}$ methods on ***LogLik***, ***EstErr*** and ***RankCorr***, respectively. For each method, we evaluate its performance when the number of events in each training sequence varies. We run the experiment 10 times with various parameter configurations. For each configuration, except the $\lambda_1, \ \lambda_2$ in Full, the λ_1 in Sparse and the λ_2 in $\mathsf{LowRank}$, which are fixed to be 0's, the rest parameters are sampled from a small neighborhood of the optimal configuration shown in the parameter studies. Fig. 2 presents the averaged results of the 10 runs, which shows that MMHP consistently achieves significantly better performance (i.e., higher ***LogLik***, lower ***EstErr*** and higher ***RankCorr***) than other methods over different training sets. This is particularly true when the training sequences have fewer events. The experimental results demonstrate that by learning from multiple sequences concurrently, MMHP is able to leverage information from other sequences for the sequences with fewer events, and thus effectively prevent over-fitting.

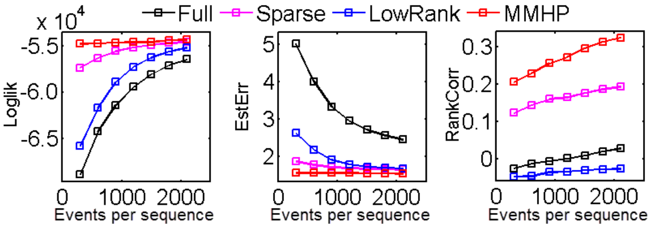


Figure 2: Experimental Results on Synthetic data.

Clustering Effects Comparisons. We cluster the training sequences by applying k-means clustering algorithm on the learned infectivity tensors and evaluate the clusters using ***ClusAcc***. The results in Table. 1 demonstrate that MMHP outperforms others in uncovering structures across sequences and the performance difference is more significant when training sequences have fewer events, indicating the effectiveness of MMHP in preventing over-fitting.

4.2 Experimental Results on Real IPTV Systems

We apply MMHP to model the data from a real IPTV system. The dataset is collected from China Telecom, in Shanghai,

Table 1: Clustering on Synthetic Data (***ClusAcc***($\times 100\%$)).

#Events per sequences	Full	Sparse	LowRank	MMHP
50	0.67	0.81	0.80	**0.88**
75	0.86	0.92	0.90	**0.94**
100	0.97	0.97	0.98	**0.99**

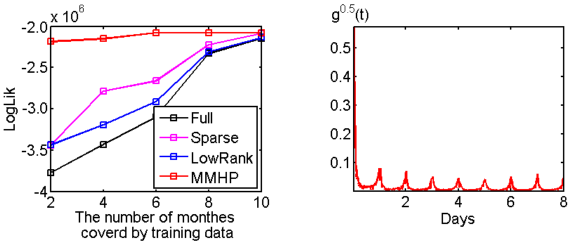
Bold numbers correspond to the best performance.

China [Luo *et al.*, 2014], which consists logs of TV program watching events from multiple users, time stamps for the beginning and endings of each watching session, and the names and the categories (labeled manually) of the TV programs. The dataset contains 2967 users (i.e., $U = 2967$) and 9,000 TV programs belonging to 25 categories (i.e., $C = 25$) that these users ever watched during 11 months in 2012.

Modeling Watching Behaviors. We model the watching behaviors of users using MMHP. Specifically, for each user u, the event sequence corresponding to her watching behaviors is $\mathcal{S}_u = \{(t^u_i, c^u_i)\}_{i=1}^{n_u}$, where the event (t^u_i, c^u_i) represents that the user u watches a program of c^u_i-th class at time t^u_i. Given the sequences of all the users during a period, we learn models using MMHP, Full, Sparse and LowRank, respectively. For all the methods, we set the length of the triggering kernel as 11520 minutes (8 days) and the sampling interval Δt as 20 minutes($M = 576$). Such a configuration ensures that 1) for daily and weekly TV programs, the triggering kernel will capture their periodic influence on its own, and 2) for most TV programs, which are 20-40 minutes in length, the triggering kernel has a good resolution to capture the influence from a previous watching event.

For all the methods, we learn the models from the first N months and test them on the data from the $(N + 1)$-th month. Running the experiments in the same way as for the synthetic data in Section. 4.1, we obtain the averaged ***LogLik*** with respect to $N = 2, ..., 10$ shown in Fig. 3(a). Similar to the results on the synthetic data, MMHP produces better ***LogLik*** than other methods. Additionally, we visualize the triggering kernel learned by MMHP in Fig. 3(b), which clearly shows that the triggering kernel captures the temporal influence decay of a program on its following programs, which corresponds to the expected nature of user watching behaviors well. There are 8 spikes in the kernel with 1 spike per day periodically, which corresponds to the self-triggering pattern of daily programs. The first spike is the highest and corresponds to the influence of previous watching events on that day. It indicates that the mutual-triggering patterns among various program categories mainly exist in the watching behaviors happening in the same day. With time elapsing, the intensity of the spikes is reduced gradually, which corresponds to the decay of influences over time. However, the spike of the 7th day is a little higher than those of its adjacent days, indicating the existence of the self-triggering pattern of weekly programs.

Learning User Clusters. We evaluate the performance of various methods on clustering users using ***ClusDiff***. We cluster the users into 5 clusters by considering the infectivity matrix $\{\boldsymbol{A}^u\}_{u=1}^U$ as the feature of a certain user and applying k-means. Table. 2 shows that compared with the other methods, MMHP not only achieves the best learning result ($\max$***LogLik***$(\mathcal{A})$) given the data from 11 months, but also obtains the smallest ***ClusDiff***. It means that the clustering results based on the proposed MMHP learning method are reasonable, where the clustering centers are representative for most of users' watching behaviors.



(a) ***LogLik***. (b) Triggering kernel

Figure 3: Experimental Results on an IPTV system.

Table 2: Clustering Performance on IPTV Data ($\times 10^7$).

Metric	Full	Sparse	LowRank	MMHP
$\max$***LogLik***$(\mathcal{A})$	—	—	—	**-1.02**
LogLik$(\widetilde{\mathcal{A}})$	-1.78	-1.18	-1.41	**-1.09**
ClusDiff	0.76	0.16	0.39	**0.07**

Bold numbers correspond to the best performance.

5 Conclusion

In this paper, we propose a multi-task multi-dimensional Hawkes process model and the corresponding learning algorithm. By considering the sparse and low-rank structure of the infectivity tensor, the proposed model captures the triggering patterns within sequences and the clustering structure across sequences jointly. The proposed learning method has superior performance compared to the other methods on both synthetic data and real-world data. In the future, we plan to propose practical applications based on the model, e.g., personalization of IPTV service.

Acknowledgement. This work is supported in part by NIH R01 GM108341,NSFC-61221001 and NSFC-61420106008.

Appendix

Feasibility of Surrogate Function. The surrogate function $\mathcal{Q}(\Theta|\Theta^{(k)})$ in Eq. (7) is induced as follows. Based on Jensen's inequality, we have

$$\log\left(\mu^u_{c^u_i} + \sum\nolimits_{j:t^u_j<t^u_i} a^u_{c^u_i c^u_j} g(t^u_i - t^u_j)\right)$$
$$\geq p^u_{ii}\log\frac{\mu^u_{c^u_i}}{p^u_{ii}} + \sum_{j=1}^{i-1} p^u_{ij}\log\frac{a^u_{c^u_i c^u_j} g(t^u_i\text{-}t^u_j)}{p^u_{ij}},$$

The equality holds if and only if $\boldsymbol{\mu} = \boldsymbol{\mu}^{(k)}$ and $\boldsymbol{A} = \boldsymbol{A}^{(k)}$. Similarly, we also have

$$a^u_{cc^u_i} g(t) \leq (a^u_{cc^u_i})^2 \frac{g^{(k)}(t)}{2a^{u(k)}_{cc^u_i}} + (g(t))^2 \frac{a^{u(k)}_{cc^u_i}}{2g^{(k)}(t)}.$$

The equality holds if and only if $g(t) = g^{(k)}(t)$ and $\boldsymbol{A} = \boldsymbol{A}^{(k)}$. Denote the objective function as $\mathcal{L}(\Theta)$. As a result, the surrogate function satisfies

$$\mathcal{Q}(\Theta|\Theta^{(k)}) \geq \mathcal{L}(\Theta),\ \ \mathcal{Q}(\Theta^{(k)}|\Theta^{(k)}) = \mathcal{L}(\Theta^{(k)}).$$

Therefore,

$$\mathcal{L}(\Theta^{(k)}) = \mathcal{Q}(\Theta^{(k)}|\Theta^{(k)}) \geq \mathcal{Q}(\Theta^{(k+1)}|\Theta^{(k)}) \geq \mathcal{L}(\Theta^{(k+1)}).$$

Euler-Lagrange Equations. Similar to that in [Zhou *et al.*, 2013b], the optimization of $\mathcal{Q}(\Theta|\Theta^{(k)})$ *w.r.t.* to $g(t)$ is equivalent to minimize $\int_0^\infty f(g, g')dt$, where

$$\begin{aligned} f(g,g') = &-\sum_{u=1}^{U}\sum_{i=1}^{n_u}\sum_{j=1}^{i-1} p_{ij}^u \log g(t)\mathbb{I}(t = t_i - t_j) \\ &+ \sum_{u=1}^{U}\sum_{c=1}^{C}\sum_{i=1}^{n_u}\frac{g^2(t)a_{cc_i^u}^{u(k)}}{2g^{(k)}(t)}\mathbb{I}(t \leq T_u - t_i) + \alpha[g'(t)]^2. \end{aligned}$$

By Euler-Lagrange equation, the solution satisfies

$$\frac{\partial f}{\partial g} - \frac{d}{dt}\frac{\partial f}{\partial g'} = 0,$$

which corresponds to Eq. (10).

References

[Bacry *et al.*, 2013] Emmanuel Bacry, Sylvain Delattre, Marc Hoffmann, and Jean-François Muzy. Modelling microstructure noise with mutually exciting point processes. *Quantitative Finance*, 13(1):65–77, 2013.

[Blundell *et al.*, 2012] Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *NIPS*, pages 2600–2608, 2012.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[Chavez-Demoulin and McGill, 2012] Valérie Chavez-Demoulin and JA McGill. High-frequency financial data modeling using hawkes processes. *Journal of Banking & Finance*, 36(12):3415–3426, 2012.

[Daley and Vere-Jones, 2007] DJ Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*, volume 2. Springer, 2007.

[Du *et al.*, 2013] Nan Du, Le Song, Hyenkyun Woo, and Hongyuan Zha. Uncover topic-sensitive information diffusion networks. In *AISTATS*, pages 229–237, 2013.

[Evgeniou and Pontil, 2007] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *NIPS*, 19:41, 2007.

[Hall and Willett, 2014] Eric C Hall and Rebecca M Willett. Tracking dynamic point processes on networks. *arXiv preprint arXiv:1409.0031*, 2014.

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

[Hawkes, 1971] Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443, 1971.

[Jacob *et al.*, 2009] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *NIPS*, pages 745–752, 2009.

[Li and Zha, 2014] Liangda Li and Hongyuan Zha. Learning parametric models for social infectivity in multi-dimensional hawkes processes. In *AAAI*, 2014.

[Linderman and Adams, 2014] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1413–1421, 2014.

[Liu *et al.*, 2009] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *UAI*, pages 339–348. AUAI Press, 2009.

[Luo *et al.*, 2014] Dixin Luo, Hongteng Xu, Hongyuan Zha, Jun Du, Rong Xie, Xiaokang Yang, and Wenjun Zhang. You are what you watch and when you watch: Inferring household structures from iptv viewing data. *Broadcasting, IEEE Transactions on*, 60(1):61–72, 2014.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2:849–856, 2002.

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

[Ogata, 1999] Yosihiko Ogata. Seismicity analysis through point-process modeling: A review. *Pure and applied geophysics*, 155(2-4):471–507, 1999.

[Ouyang *et al.*, 2013] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *ICML*, pages 80–88, 2013.

[Rodriguez *et al.*, 2011] Manuel G Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, pages 561–568, 2011.

[Yan *et al.*, 2013] Junchi Yan, Yu Wang, Ke Zhou, Jin Huang, Chunhua Tian, Hongyuan Zha, and Weishan Dong. Towards effective prioritizing water pipe replacement and rehabilitation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2931–2937. AAAI Press, 2013.

[Yang and Zha, 2013] Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *ICML*, pages 1–9, 2013.

[Zha *et al.*, 2001] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral relaxation for k-means clustering. In *NIPS*, pages 1057–1064, 2001.

[Zhou *et al.*, 2013a] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, pages 641–649, 2013.

[Zhou *et al.*, 2013b] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML*, pages 1301–1309, 2013.