
Learning Hawkes Processes from Short Doubly-Censored Event Sequences

Hongteng Xu¹ Dixin Luo² Hongyuan Zha¹

Abstract

Many real-world applications require robust algorithms to learn point processes based on a type of incomplete data — the so-called short doubly-censored (SDC) event sequences. We study this critical problem of quantitative asynchronous event sequence analysis under the framework of Hawkes processes by leveraging the idea of data synthesis. Given SDC event sequences observed in a variety of time intervals, we propose a sampling-stitching data synthesis method, sampling predecessors and successors for each SDC event sequence from potential candidates and stitching them together to synthesize long training sequences. The rationality and the feasibility of our method are discussed in terms of arguments based on likelihood. Experiments on both synthetic and real-world data demonstrate that the proposed data synthesis method improves learning results indeed for both time-invariant and time-varying Hawkes processes.

1. Introduction

Real-world interactions among multiple entities are often recorded as asynchronous event sequences, such as user behaviors in social networks, job hunting and hopping among companies, and diseases and their complications. The entities or event types in the sequences often exhibit self-triggering and mutually-triggering patterns. For example, a tweet of a twitter user may trigger further responses from her friends (Zhao et al., 2015). A disease of a patient may trigger other complications (Choi et al., 2015). Hawkes processes, an important kind of temporal point process model (Hawkes & Oakes, 1974), have capability to describe the triggering patterns quantitatively and capture the infectivity network of the entities.

¹Georgia Institute of Technology, Atlanta, Georgia, USA
²University of Toronto, Toronto, Ontario, Canada. Correspondence to: Hongteng Xu <hxu42@gatech.edu>, Dixin Luo <dixin.luo@utoronto.ca>.

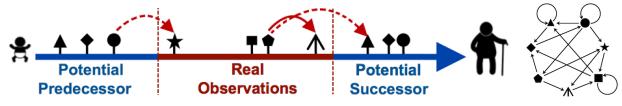


Figure 1. For each SDC sequence, i.e., incomplete disease history of a person in his lifetime, we design a mechanism to select other SDC sequences as predecessors/successors and synthesize a long sequence. Then, we can estimate the unobserved triggering patterns among diseases, i.e., the red dashed arrows, and construct a disease network.

Despite the usefulness of Hawkes processes, robust learning of Hawkes processes often needs many event sequences with events occurring over a long observation window. Unfortunately, the observation window is likely to be very short and sequence-specific in many important practical applications, i.e., within an imagined universal window, each sequence is only observed with a corresponding short sub-interval of it, and the events outside this sub-interval are not observed — we call them short doubly-censored (SDC) event sequences. Existing learning algorithms of Hawkes processes directly applied to SDCs may suffer from overfitting, and what is worse, the triggering patterns between historical events and current ones are lost, so that the triggering patterns learned from SDC event sequences are often unreliable. This problem is a thorny issue in several practical applications, especially in those having time-varying triggering patterns. For example, the disease networks of patients should evolve with the increase of age. However, it is very hard to track and record people’s diseases on a life-time scale. Instead, we can only obtain their several admissions (even only one admission) in a hospital during one or two years, which are just SDC event sequences. Therefore, it is highly desirable to propose a method to learn Hawkes processes having a longtime support from a collection of SDC event sequences.

In this paper, we propose a novel and simple data synthesis method to enhance the robustness of learning algorithms for Hawkes processes. Fig. 1 illustrates the principle of our method. Given a set of SDC event sequences, we sample predecessor for each event sequence from potential candidates and stitch them together as new training data. In the sampling step, the distribution of predecessor (and successor) is estimated according to the similarities between

current sequence and its candidates, and the similarity is defined based on the information of time stamps and (optional) features of event sequences. We analyze the rationality and the feasibility of our data synthesis method and discuss the necessary condition for using the method. Experimental results show that our data synthesis method indeed helps to improve the robustness of various learning algorithms for Hawkes processes. Especially in the case of time-varying Hawkes processes, applying our method in the learning phase achieves much better results than learning directly from SDC event sequences, which is meaningful for many practical applications, e.g., constructing dynamical disease network, and learning long-term infectivity among different IT companies.

2. Related Work

An event sequence can be represented as $s = \{(t_i, c_i)\}_{i=1}^M$, where time stamps t_i 's are in an observation window $[T_b, T_e]$ and events c_i 's are in a set of event types $\mathcal{C} = \{1, \dots, C\}$. A point process $\{N_c\}_{c \in \mathcal{C}}$ is a random process model taking event sequences as instances, where $N_c = \{N_c(t) | t \in [T_b, T_e]\}$ and $N_c(t)$ is the number of type- c events occurring at or before time t . A point process can be characterized via its conditional intensity function $\lambda_c(t) = \mathbb{E}[dN_c(t) | \mathcal{H}_t^\mathcal{C}] / dt$, where $c \in \mathcal{C}$ and $\mathcal{H}_t^\mathcal{C} = \{(t_i, c_i) | t_i < t, c_i \in \mathcal{C}\}$ is the set of history. It represents the expected instantaneous happening rate of events given historical record (Daley & Vere-Jones, 2007). The intensity is often modeled with certain parameters Θ to capture the phenomena of interests, i.e., self-triggering (Hawkes & Oakes, 1974) or self-correcting (Xu et al., 2015). Based on $\{\lambda_c(t)\}_{c \in \mathcal{C}}$, the likelihood of an event sequence s is

$$\mathcal{L}(s; \Theta) = \prod_i \lambda_{c_i}(t_i) \exp\left(-\sum_c \int_{T_b}^{T_e} \lambda_c(s) ds\right). \quad (1)$$

Hawkes Processes. Hawkes processes (Hawkes & Oakes, 1974) have a particular form of intensity:

$$\lambda_c(t) = \mu_c + \sum_{c'=1}^C \int_0^t \phi_{cc'}(t, s) dN_{c'}(s), \quad (2)$$

where μ_c is the exogenous base intensity independent of the history while $\int_0^t \phi_{cc'}(t, s) dN_{c'}(s)$ is the endogenous intensity capturing the influence of historical events on type- c ones at time t (Xu et al., 2016a). Here, $\phi_{cc'}(t, s) \geq 0$ is called *impact function*. It quantifies the influence of the type- c' event at time s to the type- c event at time t . Hawkes processes provide us with a physically-meaningful model to capture the infectivity among various events, which are used in social network analysis (Zhou et al., 2013b; Zhao et al., 2015), behavior analysis (Yang & Zha, 2013; Luo et al., 2015) and financial analysis (Bacry et al., 2013). However, the methods in these references assume

that the impact function is shift-invariant (i.e., $\phi_{cc'}(t, s) = \phi_{cc'}(t - s)$, $t \geq s$), which limits their applications on long-time scale. Recently, the time-dependent Hawkes process (TiDeH) in (Kobayashi & Lambiotte, 2016) and the neural network-based Hawkes process in (Mei & Eisner, 2016) learn very flexible Hawkes processes with complicated intensity functions. Because they highly depend on the size and the quality of data, they may fail in the case of SDC event sequences.

Learning from Imperfect Observations. In practice, we need to learn sequential models from imperfect observations (e.g., interleaved (Xu et al., 2016b), aggregated (Luo et al., 2016) and extremely-short sequences (Xu et al., 2016c)). Multiple imputation (MI) (Rubin, 2009) is a general framework to build surrogate observations from the current model. For time series, bootstrap method (Efron, 1982; Politis & Romano, 1994; Gonçalves & Kilian, 2004) and its variants (Paparoditis & Politis, 2001; Guan & Loh, 2007) have been used to improve learning results when observations are insufficient. In survival analysis, many techniques have been made to deal with truncated and censored data (Turnbull, 1974; De Gruttola & Lagakos, 1989; Klein & Moeschberger, 2005; Van den Berg & Drepper, 2016). For point processes, the global (Streit, 2010) or local (Fan, 2009) likelihood maximization estimators (MLE) are used to learn Poisson processes. Nonparametric approaches for non-homogeneous Poisson processes use the pseudo MLE (Sun & Kalbfleisch, 1995) or full MLE (Wellner & Zhang, 2000). The bootstrap methods above are also used to learn point processes (Cowling et al., 1996; Guan & Loh, 2007; Kirk & Stumpf, 2009). To learn Hawkes processes robustly, structural constraints, e.g., low-rank (Luo et al., 2015) and group-sparse regularizers (Xu et al., 2016a), are introduced. However, all of these methods do not consider the case of SDC event sequences for Hawkes processes.

3. Learning from SDC Event Sequences

Suppose that the original complete event sequences are in a long observation window. However, the observation window in practice might be segmented into several intervals $\{T_b^n, T_e^n\}_{n=1}^N$, and we can only observe K_n SDC sequences $\{s_k^n\}_{k=1}^{K_n}$ in the n -th interval, $n = 1, \dots, N$. Although we can still apply maximum likelihood estimator to learn Hawkes processes, i.e.,

$$\min_{\Theta} - \sum_{n,k} \log \mathcal{L}(s_k^n; \Theta), \quad (3)$$

the SDC event sequences would lead to over-fitting problem and the loss of triggering patterns. Can we do better in such a situation? In this work, we propose a data synthesis method based on a sampling-stitching mechanism, which extends SDC event sequences to longer ones and enhances the robustness of learning algorithms.

3.1. Data Synthesis via Sampling-Stitching

Denote the k -th SDC event sequence in the n -th interval as s_k^n . Because its *predecessor* is unavailable, if we learn the parameters of our model via (3) directly, we actually impose a strong assumption on our data that there is no event happening before s_k^n (or previous events are too far away from s_k^n to have influences on s_k^n). Obviously, this assumption is questionable — it is likely that there are influential events happening before s_k^n . A more reasonable strategy is enumerating potential predecessors and maximizing the expected log-likelihood over the whole observation window:

$$\min_{\Theta} - \sum_{n,k} \mathbb{E}_{s \sim \mathcal{H}_{T_b^n}^c} [\log \mathcal{L}([s, s_k^n]; \Theta)]. \quad (4)$$

Here $\mathbb{E}_{x \sim \mathcal{D}}[f(x)]$ represents the expectation of function $f(x)$ with random variable x yielding to a distribution \mathcal{D} . $s \sim \mathcal{H}_{T_b^n}^c$ means all possible history before T_b^n , and $\mathcal{L}([s, s_k^n]; \Theta)$ is the likelihood of stitched sequence $[s, s_k^n]$.

The stitched sequence $[s, s_k^n]$ can be generated via **sampling** SDC sequence s from previous 1st, ..., $(k-1)$ -th intervals and **stitching** s to s_k^n . The sampling process yields to the probabilistic distribution of the stitched sequences. Given s_k^n , we can compute its similarity between its potential predecessor $s_{k'}^{n'}$ in $[T_b^{n'}, T_e^{n'}]$ as

$$w(s_{k'}^{n'}, s_k^n) = \begin{cases} 0, & \text{when } T_e^{n'} \geq T_b^n \\ S(T_b^n, T_e^{n'})S(f_k^n, f_{k'}^{n'}), & \text{o.w.} \end{cases} \quad (5)$$

Here, $S(a, b) = \exp(-\|b - a\|_2^2 / \sigma_s)$ is a predefined similarity function with parameter σ_s . f_k^n is the feature of s_k^n , which is available in some applications. Note that the availability of feature is optional — even if the feature of sequence is unavailable, we can still define the similarity measurement purely based on time stamps. The normalized $\{w(s_{k'}^{n'}, s_k^n)\}$ provides us with the probability that $s_{k'}^{n'}$ appears before s_k^n , i.e., $p(s_{k'}^{n'} | s_k^n) \propto w(s_{k'}^{n'}, s_k^n)$. Then, we can sample $s_{k'}^{n'}$ according to the categorical distribution, i.e., $s_{k'}^{n'} \sim \text{Category}(w(\cdot, s_k^n))$.

We can apply such a sampling-stitching mechanism L times iteratively to the SDC sequences in both backward and forward directions and get long stitched event sequences. Specifically, we represent a stitched event sequence as $s_{stitch} = [s_1, \dots, s_{2L+1}]$, $s_l \in \{s_k^n\}$, $l = 1, \dots, 2L+1$, whose probability is

$$p(s_{stitch}) \propto \prod_{l=1}^{2L} w(s_l, s_{l+1}). \quad (6)$$

Note that our data synthesis method naturally contains two variants. When the starting (the ending) point of time window is unavailable, we use the time stamp of the first (the last) event of SDC sequence instead. Additionally, we can

relax the constraint $T_e^{n'} \leq T_b^n$ in (5) and allow a SDC sequence to have an overlap with its predecessor/successor. In this case, we preserve the overlap part randomly either from itself or its predecessor/successor before applying our sampling-stitching method. These two variants ensure that our data synthesis method is doable in practice, which are used in the following experiments on real-world data.

3.2. Justification

After applying our data synthesis method, we obtain many stitched event sequences, which can be used as instances for estimating $\mathbb{E}_{s \sim \mathcal{H}_{T_b^n}^c} [\log \mathcal{L}([s, s_k^n]; \Theta)]$. Specifically, taking advantage of stitched sequences, we can rewrite the learning problem in (4) approximately as

$$\min_{\Theta} - \sum_{s_{stitch} \in S} p(s_{stitch}) \log \mathcal{L}(s_{stitch}; \Theta), \quad (7)$$

which is actually the minimum cross-entropy estimation. $p(s_{stitch})$ represents the “true” probability that the stitched sequence happens, which is estimated via the predefined similarity measurement and the sampling mechanism. The likelihood $\mathcal{L}(s_{stitch}; \Theta)$ represents the “unnatural” probability that the stitched sequence happens, which is estimated based on the definition in (1). Our data synthesis method takes advantage of the information of time stamps and (optional) features and makes $p(s_{stitch})$ suitable for practical situations. For example, the likelihood of a sequence generally reduces with the increase of observation time window. The proposed probability $p(s_{stitch})$ yields to the same pattern — according to (6), the longer a stitched sequence is, the smaller its probability becomes.

The set of all possible stitched sequences, i.e., the S in (7), is very large, whose cardinality is $|S| = \mathcal{O}(\prod_{n=1}^N K_n)$. In practice, we cannot and do not need to enumerate all possible combinations. An empirical setting is making the number of stitched sequences comparable to that of original SDC event sequences, i.e., generating $\mathcal{O}(\sum_{n=1}^N K_n)$ stitched sequences. In the following experiments, we just apply 5 ($= U$) trials and generate 5 stitched sequences for each original SDC event sequence, which achieves a trade-off between computational complexity and performance.

3.3. Feasibility

It should be noted that our data synthesis method is only suitable for those complicated point processes whose historical events have influences on current and future ones. Specifically, we analyze the feasibility of our method for several typical point processes.

Poisson Processes. Our data synthesis method cannot improve learning results if the event sequences are generated via Poisson processes. For Poisson processes, the happening rate of future events is independent of historical

events. In other words, the intensity function of each interval can be learned independently based on the SDC event sequences. The stitched sequences do not provide us with any additional information.

Hawkes Processes. For Hawkes processes, whose intensity function is defined as (2), our data synthesis method can enhance the robustness of learning algorithm generally. In particular, consider a “long” event sequence generated via a Hawkes process in the time window $[T_b, T_e]$. If we divide the time window into 2 intervals, i.e., $[T_b, T]$ and $(T, T_e]$, the intensity function corresponding to the second interval can be written as

$$\lambda_c(t) = \mu_c + \sum_{t_i \leq T} \phi_{cc_i}(t, t_i) + \sum_{T < t_i \leq T_e} \phi_{cc_i}(t, t_i). \quad (8)$$

If the events in the first interval are unobserved, we just have a SDC event sequence, and the second term in (8) is unavailable. Learning Hawkes processes directly from the SDC event sequence ignores the information of the second term, which has a negative influence on learning results. Our data synthesis method leverages the information from other potential predecessors and generates multiple candidate long sequences. As a result, we obtain multiple intensity functions sharing the second interval and maximize the weighted sum of their log-likelihood functions (i.e., an estimated expectation of the log-likelihood of the real long sequence), as (7) does.

Compared with learning from SDC event sequences directly, applying our data synthesis method can improve learning results in general, unless the term $\sum_{t_i \leq T} \phi_{cc_i}(t, t_i)$ is ignorable. Specifically, we can model the impact functions $\{\phi_{cc'}(t, s)\}_{c, c' \in \mathcal{C}}$ of Hawkes processes based on basis representation:

$$\begin{aligned} \phi_{cc'}(t, s) &= \underbrace{\psi_{cc'}(t)}_{\text{Infectivity}} \times \underbrace{g(t - s)}_{\text{Triggering kernel}} \\ &= \sum_{m=1}^M a_{cc'm} \kappa_m(t) g(t - s). \end{aligned} \quad (9)$$

Here, we decompose impact functions into two parts: 1) *Infectivity* $\psi_{cc'}(t) = \sum_{m=1}^M a_{cc'm} \kappa_m(t)$ represents the infectivity of event type c' to c at time t .¹ 2) *Triggering kernel* $g(t) = \exp(-\beta t)$ measures the time decay of infectivity. It means that the infectivity of a historical event to current one reduces exponentially with the increase of temporal distance between them. When β is very large, $\phi_{cc'}(t, s)$ decays rapidly with the increase of $t - s$, and the events happening long ago can be ignored. In such a situation, our data synthesis method is unable to improve learning results.

¹When $M = 1$ and $\kappa_m(t) \equiv 1$, we obtain the simplest time-invariant Hawkes process. Relaxing the shift-invariant assumption, i.e., $M > 1$ and $\kappa_m(t)$ is Gaussian, we obtain a flexible time-varying Hawkes process model.

4. Implementation for Hawkes Processes

Hawkes process is a kind of physically-interpretable model for many natural and social phenomena. The proposed model in (9) reflects many common properties of real-world event sequences. First, the infectivity among various event types often changes smoothly in practice: in social networks, the interaction between two users changes smoothly, which is not established or blocked suddenly; in disease networks, the infectivity among diseases should change smoothly with the increase of patient’s age. Applying Gaussian basis representation guarantees the smoothness of infectivity function. Second, the triggering kernel measures the decay of infectivity over time. According to existing work, the decay of infectivity is exponential approximately, which has been verified in many real-world data (Zhou et al., 2013a; Kobayashi & Lambotte, 2016; Choi et al., 2015). For learning Hawkes processes from SDC event sequences, we combine our data synthesis method with an EM-based learning algorithm of Hawkes processes. Applying our data synthesis method, we obtain a set of stitched event sequences $S = \{s_n\}$ and their appearance probabilities $\{p_n\}$, where $s_n = \{(t_i^n, c_i^n)_{i=1}^{I_n} | t_i^n \in [T_b^n, T_e^n], c_i^n \in \mathcal{C}\}$ and p_n is calculated based on (5). According to (7, 9), we can learn target Hawkes process via

$$\min_{\boldsymbol{\mu} \geq \mathbf{0}, \mathbf{A} \geq \mathbf{0}} - \sum_{n=1}^{|S|} p_n \log \mathcal{L}(s_n; \Theta) + \gamma \mathcal{R}(\mathbf{A}). \quad (10)$$

Here $\Theta = \{\boldsymbol{\mu}, \mathbf{A}\}$ represents the parameters of our model. The vector $\boldsymbol{\mu} = [\mu_c]$ and the tensor $\mathbf{A} = [a_{cc'm}]$ are non-negative. Based on (1, 9), the log-likelihood function is

$$\begin{aligned} &\log \mathcal{L}(s_n; \Theta) \\ &= \sum_{i=1}^{I_n} \log \lambda_{c_i}(t_i) - \sum_{c=1}^C \int_{T_b^n}^{T_e^n} \lambda_c(s) ds \\ &= \sum_{i=1}^{I_n} \log \left[\mu_{c_i^n} + \sum_{j < i} g(\tau_{ij}^n) \sum_m a_{c_i^n c_j^n m} \kappa_m(t_i^n) \right] \\ &\quad - \sum_{c=1}^C \left(\Delta^n \mu_c - \sum_m \sum_{i=1}^{I_n} \sum_{j \leq i} a_{cc_j^n m} G_{ij} \right), \end{aligned}$$

where $\tau_{ij}^n = t_i^n - t_j^n$, $G_{ij} = \int_{t_i^n}^{t_{i+1}^n} \kappa_m(s) g(s - t_j^n) ds$, and $\Delta^n = T_e^n - T_b^n$. $\mathcal{R}(\mathbf{A})$ represents the regularizer of parameters, whose weight is γ . Following existing work in (Luo et al., 2015; Zhou et al., 2013a; Xu et al., 2016a), we assume the infectivity connections among different event types to be sparse and impose a ℓ_1 -norm regularizer on the coefficient tensor \mathbf{A} , i.e., $\mathcal{R}(\mathbf{A}) = \|\mathbf{A}\|_1 = \sum_{c, c', m} |a_{cc'm}|$.

We can solve the problem via an EM algorithm. Specifically, when sparse regularizer is applied, we take advantage of ADMM method, introducing auxiliary variable $Z = [z_{cc'm}]$ and dual variable $\mathbf{U} = [u_{cc'm}]$ for \mathbf{A} and

rewriting the objective function in (10) as

$$-\sum_n p_n \log \mathcal{L}(s_n; \Theta) + 0.5\rho \|\mathbf{A} - \mathbf{Z}\|_F^2 + \rho \text{tr}(\mathbf{U}^\top (\mathbf{A} - \mathbf{Z})) + \gamma \|\mathbf{Z}\|_1.$$

Here ρ controls the weights of regularization terms, which increases with the number of EM iterations. $\text{tr}(\cdot)$ computes the trace of matrix. Then, we can update $\{\mu, \mathbf{A}\}$, \mathbf{Z} , and \mathbf{U} alternatively.

Update μ and \mathbf{A} : Given the parameters in the k -th iteration, we apply Jensen's inequality to $-\sum_n \log \mathcal{L}(s_n; \Theta)$ and obtain a surrogate objective function for μ and \mathbf{A} :

$$\begin{aligned} & \mathcal{Q}(\mu, \mathbf{A}; \mu^k, \mathbf{A}^k, \mathbf{Z}^k, \mathbf{U}^k) \\ = & -\sum_{n=1}^N p_n \left\{ \sum_{i=1}^{I_n} \left[\sum_{j < i}^M \sum_{m=1}^M q_{ijm} \log \frac{g(\tau_{ij}^n) a_{c_i^n c_j^n m} \kappa_m(t_i^n)}{q_{ijm}} \right. \right. \\ & + q_i \log \frac{\mu_{c_i^n}}{q_i} - \sum_{c=1}^C \sum_{m=1}^M a_{cc_j^n m} G_{ij} \left. \right] - \Delta^n \sum_{c=1}^C \mu_c \left. \right\} \\ & + 0.5\rho \|\mathbf{A} - \mathbf{Z}^k + \mathbf{U}^k\|_F^2, \end{aligned}$$

where $q_i = \frac{\mu_{c_i^n}}{\lambda_{c_i^n}^k(t_i^n)}$ and $q_{ijm} = \frac{g(\tau_{ij}^n) a_{c_i^n c_j^n m} \kappa_m(t_i^n)}{\lambda_{c_i^n}^k(t_i^n)}$, and $\lambda_{c_i^n}^k(t_i^n)$ is calculated based on μ^k and \mathbf{A}^k . Then, we can update μ and \mathbf{A} via solving $\frac{\partial \mathcal{Q}}{\partial \mu} = \mathbf{0}$ and $\frac{\partial \mathcal{Q}}{\partial \mathbf{A}} = \mathbf{0}$. Both of these two equations have closed-form solution:

$$\begin{aligned} \mu_c^{k+1} &= \frac{\sum_n p_n \sum_{c_i^n=c} q_i}{\sum_n p_n \Delta_n}, \\ a_{cc'm}^{k+1} &= \frac{\sqrt{B^2 - 4\rho C} - B}{2\rho}, \end{aligned} \quad (11)$$

where

$$\begin{aligned} B &= \rho(u_{cc'm}^k - z_{cc'm}^k) + \sum_n p_n \sum_{c_i^n=c, c_j^n=c', j \leq i} G_{ij}, \\ C &= -\sum_n p_n \sum_{c_i^n=c, c_j^n=c', j \leq i} q_{ijm}. \end{aligned}$$

Update \mathbf{Z} : Given \mathbf{A}^{k+1} and \mathbf{U}^k , we can update \mathbf{Z} via solving the following optimization problem:

$$\min_{\mathbf{Z}} \gamma \|\mathbf{Z}\|_1 + 0.5\rho \|\mathbf{A}^{k+1} - \mathbf{Z} + \mathbf{U}^k\|_F^2.$$

Applying soft-thresholding method, we have

$$\mathbf{Z}^{k+1} = F_{\frac{\gamma}{\rho}}(\mathbf{A}^{k+1} + \mathbf{U}^k), \quad (12)$$

where $F_\eta(x) = \text{sign}(x) \min\{|x| - \eta, 0\}$ is the soft-thresholding function.

Update \mathbf{U} : Given \mathbf{A}^{k+1} and \mathbf{Z}^{k+1} , we can further update dual variable as

$$\mathbf{U}^{k+1} = \mathbf{U}^k + (\mathbf{A}^{k+1} - \mathbf{Z}^{k+1}). \quad (13)$$

Algorithm 1 Learning Algorithm of Hawkes Processes

- 1: **Input:** Event sequences S . The threshold V . Predefined parameters β , σ_κ , and γ .
- 2: **Output:** Parameters \mathbf{A} and μ .
- 3: Initialize \mathbf{A}^k and μ^k randomly. $\mathbf{Z}^k = \mathbf{A}^k$, $\mathbf{U}^k = \mathbf{0}$, $k = 0$, $\rho = 1$.
- 4: **repeat**
- 5: Obtain \mathbf{A}^{k+1} and μ^{k+1} via (11).
- 6: Obtain \mathbf{Z}^{k+1} via (12).
- 7: Obtain \mathbf{U}^{k+1} via (13).
- 8: $k = k + 1$, $\rho = 1.5\rho$.
- 9: **until** $\|\mathbf{A}^k - \mathbf{A}^{k-1}\|_F < V$
- 10: $\mathbf{A} = \mathbf{A}^k$, $\mu = \mu^k$.

In summary, Algorithm 1 shows the scheme of our learning method. Note that the algorithm can be applied to SDC event sequences directly via ignoring p_n 's.

5. Experiments

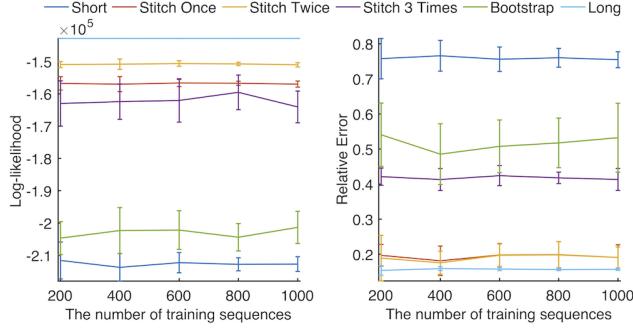
5.1. Implementation Details

To demonstrate the usefulness of our data synthesis method, we combine it with various learning algorithms of Hawkes processes and learn different models accordingly from SDC event sequences. For time-invariant Hawkes processes, we consider two learning algorithms — our EM-based learning algorithm and the least squares (LS) algorithm in (Eichler et al., 2016). For time-varying Hawkes processes, we apply our EM-based learning algorithm. In the following experiments, we use Gaussian basis functions: $\kappa_m(t) = \exp((t - t_m)^2 / \sigma_\kappa)$ with center t_m and bandwidth σ_κ . The number and the bandwidth of basis can be set according to the basis selection method proposed in (Xu et al., 2016a). Additionally, we set $V = 10^{-4}$, $\gamma = 1$, and $\sigma_s = 1$ in our algorithm. Given SDC event sequences, we learn Hawkes processes in three ways: 1) learning directly from SDC event sequences; 2) applying the stationary bootstrap method in (Politis & Romano, 1994) to generate more synthetic SDC event sequences and learning from these sequences accordingly; 3) learning from stitched sequences generated via our data synthesis method. For real-world data, whose SDC sequences do not have predefined starting and ending time stamps, we applied the variants of our method mentioned in the end of Section 3.1.

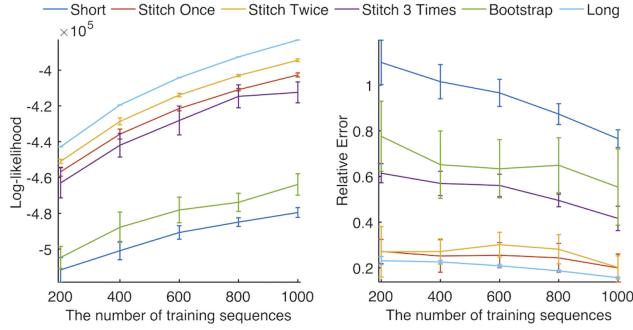
5.2. Synthetic Data

The synthetic SDC event sequences are generated via the following method: 2000 complete event sequences are simulated in the time window $[0, 50]$ based on a 2-dimensional Hawkes process. The base intensity $\{\mu_c\}_{c=1}^2$ are randomly generated in the range $[0.1, 0.2]$. The parameter of trigger-

Learning Hawkes Processes from Short Doubly-Censored Event Sequences



(a) Our learning algorithm



(b) Least squares algorithm

Figure 2. Comparisons on log-likelihood and relative error.

ing kernel, β , is set to be 0.2. For time-invariant Hawkes processes, we set the infectivity $\{\psi_{cc'}(t)\}$ to be 4 constants randomly generated in the range $[0, 0.2]$. For time-varying Hawkes processes, we set $\psi_{cc'}(t) = 0.2 \cos(2\pi \frac{\omega_{cc'}}{50} t)$, where $\{\omega_{cc'}\}$ are randomly generated in the range $[1, 4]$. Given these complete event sequences, we select 1000 sequences as testing set while the remaining 1000 sequences as training set. To generate SDC event sequences, we segment time window into 10 intervals, and just randomly preserve the data in one interval for each training sequences. We test all methods in 10 trials and compare them on the relative error between real parameters Θ and their estimation results $\hat{\Theta}$, i.e., $\frac{\|\Theta - \hat{\Theta}\|_2}{\|\Theta\|_2}$, and the log-likelihood of testing sequences.

Time-invariant Hawkes Processes. Fig. 2 shows the comparisons on log-likelihood and relative error for various methods. In Fig. 2(a) we can find that compared with the learning results based on complete event sequences, the results based on SDC event sequences degrade a lot (lower log-likelihood and higher relative error) because of the loss of information. Our data synthesis method improves the learning results consistently with the increase of training sequences, which outperforms its bootstrap-based competitor (Politis & Romano, 1994) as well. To demonstrate the universality of our method, besides our EM-based algorithm, we apply our method to the Least Squares (LS) algo-

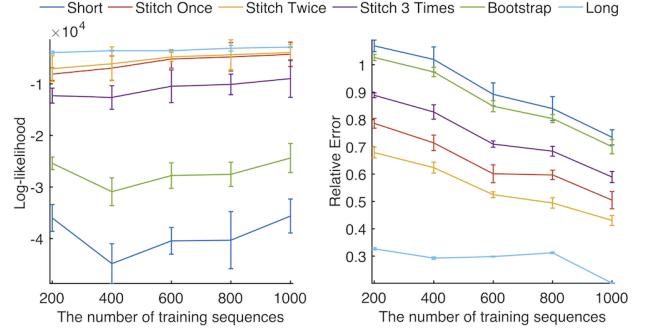


Figure 3. Comparisons on log-likelihood and relative error.

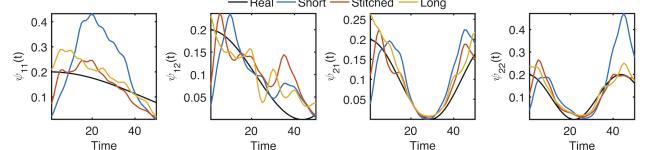


Figure 4. Comparisons on infectivity functions $\{\psi_{cc'}(t)\}$. The number of original SDC sequences is 200 and stitched via our method once.

rithm (Eichler et al., 2016). Fig. 2(b) shows that our method also improves the learning results of the LS algorithm in the case of SDC event sequences. Both the log-likelihood and the relative error obtained from the stitched sequences approach to the results learned from complete sequences.

Time-varying Hawkes Processes. Fig. 3 shows the comparisons on log-likelihood and relative error for various methods. Similarly, the learning results are improved because of applying our method — higher log-likelihood and lower relative error are obtained and their standard deviation (the error bars associated with curves) is shrunk. In this case, applying our method twice achieves better results than applying once, which verifies the usefulness of the iterative framework in our sampling-stitching algorithm. Besides objective measurements, in Fig. 4 we visualize the infectivity functions $\{\psi_{cc'}(t)\}$. It is easy to find that the infectivity functions learned from stitched sequences (red curves) are comparable to those learned from complete event sequences (yellow curves), which have small estimation errors of the ground truth (black curves).

Note that our iterative framework is useful, especially for time-varying Hawkes processes, when the number of stitches is not very large. In our experiments, we fixed the maximum number of synthetic sequences. As a result, Figs. 2 and 3 show that the likelihoods first increase (i.e., stitching once or twice) and then decrease (i.e., stitching more than three times) while the relative errors have opponent changes w.r.t. the number of stitches. These phenomena imply that too many stitches introduce too much unreliable interdependency among events. Therefore, we fix

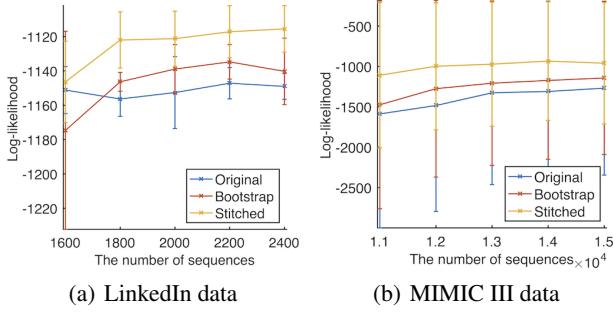


Figure 5. Comparisons on log-likelihood.

the number of stitches to 2 in the following experiments.

5.3. Real-World Data

Besides synthetic data, we also test our method on real-world data, including the LinkedIn data collected by ourselves and the MIMIC III data set (Johnson et al., 2016).

LinkedIn Data. The LinkedIn data we collected online contain job hopping records of 3,000 LinkedIn users in 82 IT companies. For each user, her/his check-in time stamps corresponding to different companies are recorded as an event sequence, and her/his profile (e.g., education background, skill list, etc.) is treated as the feature associated with the sequence. For each person, the attractiveness of a company is always time-varying. For example, a young man may be willing to join in startup companies and increase his income via jumping between different companies. With the increase of age, he would more like to stay in the same company and achieve internal promotions. In other words, the infectivity network among different companies should be dynamical w.r.t. the age of employees. Unfortunately, most of the records in LinkedIn are short and doubly-censored — only the job hopping events in recent years are recorded. How to construct the dynamical infectivity network among different companies from the SDC event sequences is still an open problem.

Applying our data synthesis method, we can stitch different users' job hopping sequences based on their ages (time stamps) and their profile (feature) and learn the dynamical network of company over time. In particular, we select 100 users with relatively complete job hopping history (i.e., the range of their working experience is over 25 years) as testing set. The remaining 2,900 users are randomly selected as training set. The log-likelihood of testing set in 10 trials is shown in Fig. 5(a). We can find that the log-likelihood obtained from stitched sequences is higher than that obtained from original SDC sequences or that obtained from the sequences generated via the bootstrap method (Politis & Romano, 1994), and its standard deviation is bounded

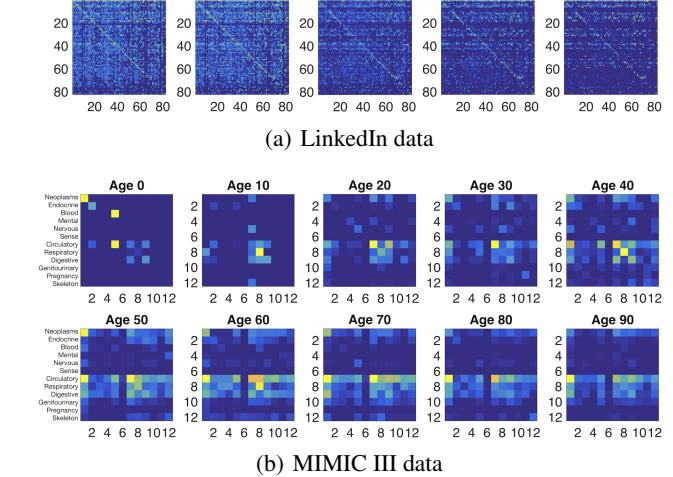


Figure 6. Comparisons on infectivity functions $\{\psi_{cc'}(t)\}$.

stably. Fig. 6(a) visualizes the adjacent matrix of infectivity network. The properties of the network verifies the rationality of our results: 1) the diagonal elements of the adjacent matrix are larger than other elements in general, which reflects the fact that most employees would like to stay in the same company and achieve a series of internal promotions; 2) with the increase of age, the infectivity network becomes sparse, which reflects the fact that users are more likely to try different companies in the early stages of their careers.

MIMIC III Data. The MIMIC III data contain admission records of over 40,000 patients in the Beth Israel Deaconess Medical Center between 2001 and 2012. For each patient, her/his admission time stamps and diseases (represented via the ICD-9 codes (Deyo et al., 1992)) are recorded as an event sequence, and her/his profile (including gender, race and chronic history) is represented as binary feature of the sequence. As aforementioned, some work (Choi et al., 2015) has been done to extract time-invariant disease network from admission records. However, the real disease network should be time-varying w.r.t. the age of patient. Similar to the LinkedIn data, here we only obtain SDC event sequences — the ranges of most admission records are just 1 or 2 years.

Applying our data synthesis method, we can leverage the information from different patients and stitch their sequences based on their ages and their profile. Focusing on 600 common diseases in 12 categories, we select 15,000 patients' admission records randomly as training set and 1,000 patients with relatively complete records as testing set. Fig. 5(b) shows that applying our data synthesis method indeed helps to improve log-likelihood of testing data. Compared with our bootstrap-based competitor, our data synthesis method gets more obvious improvements.

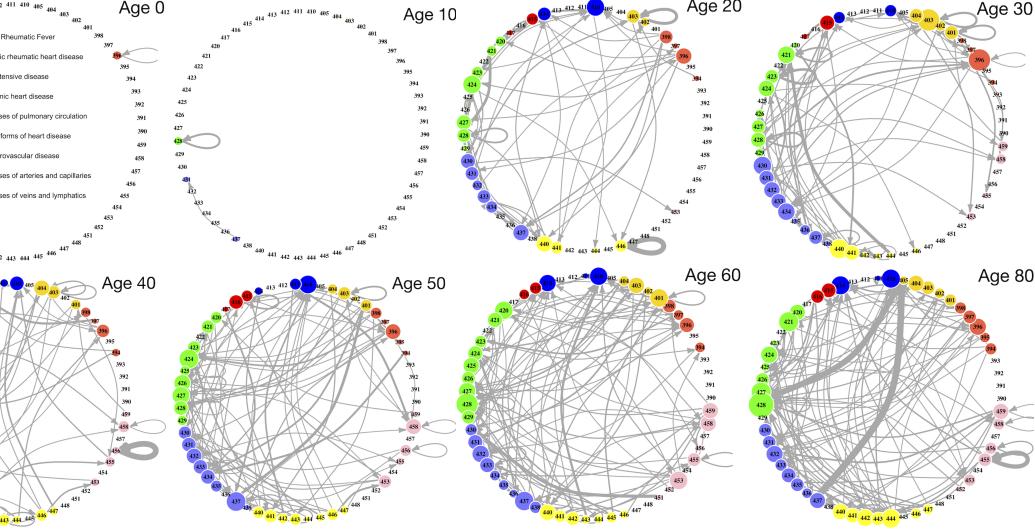


Figure 7. The diseases (nodes) are labeled with ICD-9 codes. The colors indicate their sub-categories. The size of the c -th node is $\sum_{c'} \psi_{cc'}(t)$. The width of edge is $\psi_{cc'}(t)$.

Furthermore, we visualize the adjacent matrix of dynamical network of disease categories in Fig. 6(b). We can find that: 1) with the increase of age the disease network becomes dense, which reflects the fact that the complications of diseases are more and more common when people become old; 2) the networks show that neoplasms and the diseases of circulatory, respiratory, and digestive systems have strong self-triggering patterns because the treatments of these diseases often include several phases and require patients to make admission multiple times; 3) for kids and teenagers, their disease networks (i.e., ‘‘Age 0’’ and ‘‘Age 10’’ networks) are very sparse, and their common diseases mainly include neoplasms and the diseases of circulatory, respiratory, and digestive systems; 4) for middle-aged people, the reasons for their admissions are diverse and complicated so that their disease networks are dense and include many mutually-triggering patterns; 5) for longevity people, their disease networks (i.e., ‘‘Age 80’’ and ‘‘Age 90’’ networks) are relatively sparser than those of middle-aged people, because their admissions are generally caused by elderly chronic diseases.

Additionally, we visualize the dynamical networks of the diseases of circulatory systems in Fig. 7, and find some interesting triggering patterns. For example, for kids (‘‘Age 0’’ network), the typical circulatory diseases are ‘‘diseases of mitral and aortic valves’’ (ICD-9 396) and ‘‘cardiac dysrhythmias’’ (ICD-9 427), which are common for premature babies and the kids having congenital heart disease. For the old (‘‘Age 80’’ network), the network becomes dense. We can find that 1) as a main cause of death, ‘‘heart failure’’ (ICD-9 428) is triggered via multiple other diseases, especially ‘‘secondary hypertension’’ (ICD-9 405); 2) ‘‘sec-

ondary hypertension’’ is also likely to cause ‘‘other and ill-defined cerebrovascular disease’’ (ICD-9 437); 3) ‘‘Hemorrhoids’’ (ICD-9 455), as a common disease with strong self-triggering pattern, will cause frequent admissions of patients. In summary, the analysis above verifies the rationality of our result — the dynamical disease networks we learned indeed reflect the properties of human’s health trajectory. The list of ICD-9 codes and the complete enlarged network over age are shown in the supplementary file.

6. Conclusion

In this paper, we propose a novel data synthesis method to learn Hawkes processes from SDC event sequences. With the help of temporal information and optional features, we measure the similarities among different SDC event sequences and estimate the distribution of potential long event sequences. Applying a sampling-stitching mechanism, we successfully synthesize a large amount of long event sequences and learn point processes robustly. We test our method for both time-invariant and time-varying Hawkes processes. Experiments show that our data synthesis method improves the robustness of learning algorithms for various models. In the future, we plan to provide more theoretical and quantitative analysis to our data synthesis method and apply it to more applications.

Acknowledgements

This work is supported in part via NSF IIS-1639792, DMS-1317424, NIH R01 GM108341, NSFC 61628203, U1609220 and the Key Program of Shanghai Science and Technology Commission 15JC1401700.

References

- Bacry, Emmanuel, Delattre, Sylvain, Hoffmann, Marc, and Muzy, Jean-Francois. Some limit theorems for hawkes processes and application to financial statistics. *Stochastic Processes and their Applications*, 123(7):2475–2499, 2013.
- Choi, Edward, Du, Nan, Chen, Robert, Song, Le, and Sun, Jimeng. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *ICDM*, 2015.
- Cowling, Ann, Hall, Peter, and Phillips, Michael J. Bootstrap confidence regions for the intensity of a poisson point process. *Journal of the American Statistical Association*, 91(436):1516–1524, 1996.
- Daley, Daryl J and Vere-Jones, David. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.
- De Gruttola, Victor and Lagakos, Stephen W. Analysis of doubly-censored survival data, with application to aids. *Biometrics*, pp. 1–11, 1989.
- Deyo, Richard A, Cherkin, Daniel C, and Ciol, Marcia A. Adapting a clinical comorbidity index for use with icd-9-cm administrative databases. *Journal of clinical epidemiology*, 45(6):613–619, 1992.
- Efron, Bradley. *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM, 1982.
- Eichler, Michael, Dahlhaus, Rainer, and Dueck, Johannes. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 2016.
- Fan, Chun-Po Steve. *Local Likelihood for Interval-censored and Aggregated Point Process Data*. PhD thesis, University of Toronto, 2009.
- Gonçalves, Silvia and Kilian, Lutz. Bootstrapping autoregressions with conditional heteroskedasticity of unknown form. *Journal of Econometrics*, 123(1):89–120, 2004.
- Guan, Yongtao and Loh, Ji Meng. A thinned block bootstrap variance estimation procedure for inhomogeneous spatial point patterns. *Journal of the American Statistical Association*, 102(480):1377–1386, 2007.
- Hawkes, Alan G and Oakes, David. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pp. 493–503, 1974.
- Johnson, Alistair EW, Pollard, Tom J, Shen, Lu, Lehman, Li-wei H, Feng, Mengling, Ghassemi, Mohammad, Moody, Benjamin, Szolovits, Peter, Celi, Leo Anthony, and Mark, Roger G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3, 2016.
- Kirk, Paul DW and Stumpf, Michael PH. Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics*, 25(10):1300–1306, 2009.
- Klein, John P and Moeschberger, Melvin L. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2005.
- Kobayashi, Ryota and Lambotte, Renaud. Tideh: Time-dependent hawkes process for predicting retweet dynamics. *arXiv preprint arXiv:1603.09449*, 2016.
- Luo, Dixin, Xu, Hongteng, Zhen, Yi, Ning, Xia, Zha, Hongyuan, Yang, Xiaokang, and Zhang, Wenjun. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *IJCAI*, 2015.
- Luo, Dixin, Xu, Hongteng, Zhen, Yi, Dilkina, Bistra, Zha, Hongyuan, Yang, Xiaokang, and Zhang, Wenjun. Learning mixtures of markov chains from aggregate data with structural constraints. *Transactions on Knowledge and Data Engineering*, 28(6):1518–1531, 2016.
- Mei, Hongyuan and Eisner, Jason. The neural hawkes process: A neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328*, 2016.
- Paparoditis, Efstatios and Politis, Dimitris N. Tapered block bootstrap. *Biometrika*, 88(4):1105–1119, 2001.
- Politis, Dimitris N and Romano, Joseph P. The stationary bootstrap. *Journal of the American Statistical association*, 89(428):1303–1313, 1994.
- Rubin, Donald B. *Multiple Imputation for Nonresponse in Surveys*, volume 307. John Wiley & Sons, 2009.
- Streit, Roy L. *Poisson point processes: imaging, tracking, and sensing*. Springer Science & Business Media, 2010.
- Sun, J and Kalbfleisch, JD. Estimation of the mean function of point processes based on panel count data. *Statistica Sinica*, pp. 279–289, 1995.
- Turnbull, Bruce W. Nonparametric estimation of a survivorship function with doubly censored data. *Journal of the American Statistical Association*, 69(345):169–173, 1974.
- Van den Berg, Gerard J and Drepper, Bettina. Inference for shared-frailty survival models with left-truncated data. *Econometric Reviews*, 35(6):1075–1098, 2016.

Wellner, Jon A and Zhang, Ying. Two estimators of the mean of a counting process with panel count data. *Annals of Statistics*, pp. 779–814, 2000.

Xu, Hongteng, Zhen, Yi, and Zha, Hongyuan. Trailer generation via a point process-based visual attractiveness model. In *IJCAI*, 2015.

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

Xu, Hongteng, Ning, Xia, Zhang, Hui, Rhee, Junghwan, and Jiang, Guofei. Pinfer: Learning to infer concurrent request paths from system kernel events. In *ICAC*, 2016b.

Xu, Hongteng, Wu, Weichang, Nemati, Shamim, and Zha, Hongyuan. Icu patient flow prediction via discriminative learning of mutually-correcting processes. *arXiv preprint arXiv:1602.05112*, 2016c.

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

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

Zhou, Ke, Zha, Hongyuan, and Song, Le. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, 2013a.

Zhou, Ke, Zha, Hongyuan, and Song, Le. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML*, 2013b.