Counterfactual Temporal Point Processes

Kimia Noorbakhsh*
Sharif University of Technology
kimianoorbakhsh@gmail.com

Manuel Gomez Rodriguez
Max Planck Institute for Software Systems
manuelgr@mpi-sws.org

Abstract

Machine learning models based on temporal point processes are the state of the art in a wide variety of applications involving discrete events in continuous time. However, these models lack the ability to answer counterfactual questions, which are increasingly relevant as these models are being used to inform targeted interventions. In this work, our goal is to fill this gap. To this end, we first develop a causal model of thinning for temporal point processes that builds upon the Gumbel-Max structural causal model. This model satisfies a desirable counterfactual monotonicity condition, which is sufficient to identify counterfactual dynamics in the process of thinning. Then, given an observed realization of a temporal point process with a given intensity function, we develop a sampling algorithm that uses the above causal model of thinning and the superposition theorem to simulate counterfactual realizations of the temporal point process under a given alternative intensity function. Simulation experiments using synthetic and real epidemiological data show that the counterfactual realizations provided by our algorithm may give valuable insights to enhance targeted interventions.

1 Introduction

In recent years, machine learning models based on temporal point processes have become increasingly popular for modeling discrete event data in continuous time [1, 2]. This type of data is ubiquitous in a wide range of application domains, from social and information networks to finance or epidemiology. For example, in social and information networks, events may represent users’ posts, clicks or likes [3, 4]; in finance, they may represent buying and selling orders [5]; or, in epidemiology, they may represent when an individual gets infected or recovers [6]. In many of these domains, these models have become state of the art at predicting future events given a sequence of past events [7].

Building upon the above models, a recent line of work [8–11] has developed machine learning methods to automate online, adaptive targeted interventions using reinforcement learning and stochastic optimal control. While this line of work has shown early promise, particularly in personalized teaching and viral marketing, there are many high-stakes application in which targeted interventions are unlikely to be automated. For example, in epidemiology, fine-grained interventions that are targeted at particular sites or individuals (e.g., hygienic measures at work places, closures of schools, or contact tracing) are likely to be decided by governments, policy makers and health authorities, at least in the foreseeable future. In this work, our goal is to develop machine learning methods that, given the outcome of an intervention implemented in the past, are able to assist decision makers at implementing better interventions in these high-stakes applications.

More specifically, we focus on facilitating counterfactual thinking, a type of thinking that has been argued to help humans correct and improve behavior that has been unsuccessful in the past [12, 13]. In counterfactual thinking, given a history of past events that have already occurred, one asks what past events would have instead occurred if certain intervention had been in place. For example, in epidemiology, assume that, during a pandemic, a government decides to implement business

*This work was done during Kimia Noorbakhsh’s internship at MPI-SWS.

restrictions every time the weekly incidence—the (relative) number of new cases—is larger than certain threshold but unfortunately the incidence nevertheless spirals out of control. In this case, counterfactual thinking would help the government understand retrospectively to what extent the incidence would have grown had a lower threshold been implemented.

Our contributions. Our starting point is Lewis’ thinning algorithm [20], one of the most popular techniques for sampling in temporal point processes. Lewis’ thinning algorithm first samples a sequence of potential events from a temporal point process with a constant intensity that upper bounds the intensity of the temporal point process of interest. Then, it accepts each of these events with probability proportional to the ratio between the intensity of the temporal point process of interest at the time of the sampled event and the constant intensity. In our work, the key idea is to augment the above thinning process using a particular class of structural equation models (SCMs)—the Gumbel-Max SCM [21]. This causal model satisfies a desirable monotonicity condition and, given a sequence of accepted and rejected events, it allows us to reliably estimate what events would have been accepted and rejected under an alternative intensity. Put differently, it can be used to answer counterfactual questions about a set of previously accepted and rejected events by the thinning algorithm.

Unfortunately, the above causal model on its own is not sufficient to answer counterfactual questions about observed sequence of real events. This is because, in general, real event data is not generated by a thinning process and, as a consequence, it does not include rejected events. However, we are able to overcome this limitation by using the superposition theorem [22] to sample plausible sequences of events that the thinning process would have rejected if it had accepted the observed sequence of real events. Then, these generated sequences of rejected events, together with the observed events, can be fed into the above causal model of thinning to sample counterfactual events given an alternative intensity. Importantly, while our causal model of thinning only allows for inhomogeneous Poisson processes (i.e., it requires the intensities of interest to be deterministic), we can still use it to sample counterfactual events from linear Hawkes processes [23], a popular type of temporal point processes with stochastic self-exciting intensities, by exploiting their branching process interpretation [24]. Finally, we evaluate our sampling algorithm using both synthetic and real epidemiological data and show that the counterfactual events provided by our algorithm can give valuable insights to enhance targeted intervention.

Further related work. The literature on temporal point processes related to causal inference has mostly focused on measuring causal influence by means of, e.g., Granger causality [25–28], integrated cumulants [29, 30] or Wold processes [31, 32], and on predicting quantities related to an interventional distribution of interest [33–35] (e.g., conditional average treatment effect (CATE)). However, there exist a few notable exceptions, which have focused on counterfactual reasoning [36–38]. The work by Schulam and Saria [36] focuses on reasoning about the counterfactual distributions of the marks in a marked temporal point process, rather than reasoning about the counterfactual intensities of the events as we do. The works by Ryalen et al. [37] and Roysland [38] focus on survival analysis, i.e., temporal point processes that terminate after one event, and are very different to ours at a technical level. In contrast, we focus on temporal point processes with multiple events.

More broadly, the literature on causal inference has a long and rich history [40]. However, most of this literature has used counterfactual reasoning to predict quantities related to the interventional distribution of interest such as, e.g., the conditional average treatment effect (CATE). A few recent notable exceptions are by Oberst and Sontag [21] and Tsirtsis et al. [41], which have used the Gumbel-Max SCM to reason about counterfactual distributions in Markov decision processes (MDPs), and by Buesing et al. [42] and Pitis et al. [43], which have used counterfactuals to improve the training of reinforcement learning agents. However, to the best of our knowledge, the Gumbel-Max SCM has not been used previously to reason about counterfactual events in temporal point processes.

2 Preliminaries

In this section, we first briefly revisit the frameworks of temporal point processes [44] and structural causal models [45].

---

2 Note that existing epidemiological models [14], including those developed in the context of COVID-19 [15–19], are unable to answer such counterfactual questions—they can only predict what the (average) future would look like under certain interventions given the past.

3 To facilitate research in this area, we release an open-source implementation of our algorithms and data at https://github.com/Networks-Learning/counterfactual-tpp

4 The work by Zhang et al. [39] is contemporary to ours and presents a method to estimate individual treatment effects (ITE) on the intensity of a temporal point process.
Temporal point processes. A temporal point process is a stochastic process whose realization consists of a sequence of discrete events localized in continuous time, \( \mathcal{H} = \{ t_i \in \mathbb{R}^+ | i \in \mathbb{N}^+, t_i < t_{i+1} \} \). A temporal point process can be equivalently represented as a counting process, \( N(t) \), which records the number of events before time \( t \). Moreover, in an infinitesimally small time window \( dt \) around time \( t \), it is assumed that only one event can happen, i.e., \( dN(t) \in \{0, 1\} \).

A temporal point process is typically characterized via its intensity function \( \lambda(t) \geq 0 \), which determines the probability of observing an event in \([t, t + dt]\), i.e.,

\[
\lambda(t) dt = P\{dN(t) = 1\} = \mathbb{E}[dN(t)].
\]  

In general, the intensity \( \lambda(t) \) may depend on the history \( \mathcal{H}(t) = \{ t_i \in \mathcal{H} | t_i < t \} \) up to time \( t \) and its functional form is often chosen to capture the phenomena of interest. Throughout the paper, we will consider inhomogeneous Poisson processes, i.e., \( \lambda(t) = g(t) \), where \( g(t) \geq 0 \) is a time-varying function, and linear Hawkes processes, i.e.,

\[
\lambda(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} g(t - t_i),
\]

where \( \mu \geq 0 \) and the second term, with \( \alpha \geq 0 \), \( g(t) \geq 0 \) and \( g(t) = 0 \) for all \( t < 0 \), denotes the influence of previous events on the current intensity\(^5\).

Structural causal models. Given a set of random variables \( X = \{X_1, \ldots, X_n\} \), a structural causal model (SCM) \( \mathcal{C} \) defines a complete data-generating process via a collection of assignments

\[
X_i := f_i(PA_i, U_i),
\]

where \( PA_i \subseteq X \setminus X_i \) are the direct causes of \( X_i \), \( U = \{U_1, \ldots, U_n\} \) are noise variables, and \( P(U) \) denotes the (prior) distribution of the noise variables. Here, note that, given an observational distribution \( P(X_1, \ldots, X_n) \), there always exists a distribution for the noise variables and functions \( f_i \), so that \( P = P^C \), where \( P^C \) is the distribution entailed by \( \mathcal{C} \). In general, the noise variables \( U \) may not be jointly independent if there are unmeasured confounders\(^6\). However, in our work, we will assume there are no unmeasured confounders and \( U \) are jointly independent.

Given a SCM \( \mathcal{C} \), we can express (atomic) interventions \( \mathcal{I} \) using the do-operator, e.g., \( \mathcal{I} = \text{do}(X_i = x) \) corresponds to replacing the causal mechanism \( f_i(\text{PA}_i, U_i) \) with \( x \). The intervened SCM is typically denoted as \( \mathcal{C}^\mathcal{I} \) and the interventional distribution entailed by the intervened SCM as \( p^C:\mathcal{I} \). Moreover, given a SCM \( \mathcal{C} \) and an observed realization of assignments \( X = x \), we can define a counterfactual SCM \( \mathcal{C}_X=x \) where the noise \( U \) variables are distributed according to the posterior distribution \( P(U | X = x) \) and not necessarily jointly independent anymore. Counterfactual statements can now be seen as interventions in a counterfactual SCM \( \mathcal{C}_X=x \) and, given an intervention \( \mathcal{I} \), we denote the interventional counterfactual distribution entailed by \( \mathcal{C}_X=x;\mathcal{I} \) as \( p^C^{\mathcal{I}}|X=x;\mathcal{I} \). However, the posterior distribution of the noise variables may be non-identifiable without further assumptions. This is because there may be several noise distributions and functions \( g_i \) consistent with the observational distribution but result in different counterfactual distributions.

In the context of binary random variables, monotonicity is an assumption that avoids the above mentioned non-identifiability issues—it restricts the class of possible SCMs to those which all yield equivalent counterfactual distributions over a binary variable of interest\(^[46][21]\). More specifically, a SCM \( \mathcal{C} \) of a binary variable \( Y \) is monotonic with respect to a binary variable \( T \) if and only if the condition

\[
p^C;\text{do}(T=t)(Y = y) \geq p^C;\text{do}(T=t')(Y = y)
\]

implies that \( p^C|Y=y, T=t';\text{do}(T=t)(Y = y') = 0 \), where \( y' \neq y \).

3 A Causal Model of Thinning

In this section, we first revisit Lewis’ thinning algorithm\(^[20]\), one of the most popular techniques to simulate event data from inhomogeneous Poisson processes. Then, we augment this classical algorithm using a particular class of SCMs satisfying the monotonicity assumption, the Gumbel-Max SCMs\(^[21]\). Finally, we demonstrate that the resulting algorithm can be used to answer counterfactual questions about a set of previously simulated events.

\(^5\)The function \( g(t) \) is often called triggering kernel.

\(^6\)An unmeasured confounder is an unobserved variable that is a direct cause of two (observed) variables \( X_i \) and \( X_j \).
Algorithm 1 It samples a counterfactual sequence of accepted events given a sequence of accepted and rejected events provided by Lewis’ thinning algorithm

1: Input: $\lambda_m(t), \lambda_m^0(t), \mathcal{H}_m, \mathcal{H}_{\text{max}}, \lambda_{\text{max}}$
2: Initialize: $\mathcal{H}_{m^t} = \emptyset$
3: function ACC($\lambda_m(t), \lambda_m^0(t), \mathcal{H}_m, \mathcal{H}_{\text{max}}, \lambda_{\text{max}}$)
4: $\mathcal{H}_{m^t} \leftarrow \emptyset$
5: for $t_i \in \mathcal{H}_{\text{max}}$
6: $x_i \leftarrow \{t_i \in \mathcal{H}_m\}$
7: $x_i' \sim P_c\{x_i \in x_i, \lambda_i = \lambda_m(t_i), \text{do}(\lambda_i = \lambda_m(t_i))(x)\}$
8: if $x_i' = 1$
9: $\mathcal{H}_{m^t} \leftarrow \mathcal{H}_{m^t} \cup \{t_i\}$
10: end if
11: end for
12: return $\mathcal{H}_{m^t}$
13: end function

Let $\mathcal{M}$ be a set of inhomogeneous Poisson processes of interest and, for each $m \in \mathcal{M}$, assume its corresponding intensity function $\lambda_m(t) \leq \lambda_{\text{max}}$ for all $t \in \mathbb{R}^+$. To sample a sequence of events $\mathcal{H}_m$ from any process $m \in \mathcal{M}$, Lewis’ thinning algorithm first samples a sequence of potential events $\mathcal{H}_{\text{max}}$ from a homogeneous Poisson process with intensity $\lambda_{\text{max}}$. Then, for each event $t_i \in \mathcal{H}_{\text{max}}$, it additionally samples a Bernoulli random variable $X_i$ with parameter $p = p(\lambda_m(t_i)) = \lambda_m(t_i)/\lambda_{\text{max}}$. Finally, it accepts all the events $t_i$ such that $X_i = 1$. i.e., $\mathcal{H}_m = \{t_i \in \mathcal{H}_{\text{max}} | X_i = 1\}$. Here, note that the specific choice of $\lambda_{\text{max}}$ does not affect the distribution of accepted events as long as $\lambda_{\text{max}} \geq \lambda_m(t)$ for all $m \in \mathcal{M}$ and $t \in \mathbb{R}^+$. Algorithm 4 in Appendix B.1 summarizes the overall procedure, where the parameter $p$ is often called the thinning probability.

Given a process of interest with intensity $\lambda_m(t)$, Lewis’ thinning algorithm is helpful to make predictions about future events. For example, it can be used to compute Monte Carlo estimates of the average number of events $\mathbb{E}[N(t)]$ at a time $t$ in the future. However, it is not sufficient to make counterfactual predictions, e.g., given the sequences of events $\mathcal{H}_{\text{max}}$ and $\mathcal{H}_m$, we cannot know what would have happened if, at time $t_i$, the intensity had been $\lambda_m^0(t_i)$, with $m \neq m'$, instead of $\lambda_m(t_i)$. To overcome this limitation, we will now augment the above thinning algorithm using a Gumbel-Max SCM. More specifically, let $\mathcal{C}$ be a SCM defined by the assignments:

$$X_i = \arg \max_{x \in \{0,1\}} g(x, \Lambda_i, U_i) \quad \text{and} \quad \Lambda_i = \lambda(t_i),$$

where

$$g(x, \Lambda_i, U_i) = \log p(X_i = x | \Lambda_i) + U_i,x,$$

with $p(X_i = x | \Lambda_i) = p(\Lambda_i = (1-x)(1-p(\Lambda_i)))$, $p(\Lambda_i) = \Lambda_i/\lambda_{\text{max}}$, $U_i,x \sim \text{Gumbel}(0,1)$, and $t_i \sim \lambda_{\text{max}}$. Then, the thinning probabilities in Lewis’ thinning algorithm are given by the following interventional distributions over $\mathcal{C}$

$$P_c^{\text{do}(\lambda_i = \lambda(t_i))}(X_i = 1) = p(\lambda(t_i)) = \frac{\lambda(t_i)}{\lambda_{\text{max}}}.\tag{5}$$

Under this view, given a sequence of accepted events $\mathcal{H}_m$ and rejected events $\mathcal{H}_{\text{max}} \setminus \mathcal{H}_m$ under an intensity $\lambda_m(t)$, as determined by the binary samples $\{x_i\}$, we can estimate the posterior distribution $P_c^{\text{do}(\lambda_i = \lambda(t_i))}(X_i = x)$ using an efficient procedure [38] (refer to Appendix B.3). Then, we can use these noise posterior to compute an unbiased finite sample Monte-Carlo estimate of the counterfactual thinning probability, i.e.,

$$P_c^{\text{do}(\lambda_i = \lambda(t_i))}(X_i = x) = \mathbb{E}U_i | X_i = x, \Lambda_i = \lambda_m(t_i)] \{ \max_{x' \in \{0,1\}} g(x', \lambda_m^0(t_i), U_i) \},$$

where we drop $\text{do}(\cdot)$ because $U_i$ and $\lambda_m(t_i)$ are independent in the counterfactual SCM. Importantly, the above counterfactual thinning probability allows us to make counterfactual predictions, e.g., given

\[\text{Source: Lewis and Sheldor, 2020.}\]
Algorithm 2: It samples a counterfactual sequence of events given a sequence of observed events from an inhomogeneous Poisson process.

1: Input: $\lambda_m(t), \lambda_m'(t), H_m, \lambda_{\text{max}}, T$.
2: Initialize: $H_{m'} = \emptyset$.
3: function $Cf(\lambda_m(t), \lambda_m'(t), H_m, \lambda_{\text{max}}, T)$
4: $H_{\text{max}} \leftarrow \text{LEWIS}(\lambda_{\text{max}} - \lambda_m(t), \lambda_{\text{max}}, T)$
5: $\lambda_{\text{max}} \leftarrow \lambda_{\text{max}} | H_m$
6: $H_{m'} \leftarrow \text{ACC}(\lambda_m(t), \lambda_m'(t), H_m, \lambda_{\text{max}}, \lambda_{\text{max}})$
7: return $H_{m'}$.
8: end function

a sequence of accepted events $H_m$ and rejected events $H \setminus H_m$ under intensity $\lambda_m(t)$, we can use the counterfactual thinning probability to predict which events, among those in $H$, would have been accepted if the intensity had been $\lambda_{m'}(t)$ instead of $\lambda_m(t)$. Algorithm 2 summarizes the algorithm. Here, it is important to note that the specific choice of $\lambda_{\text{max}}$ does not affect the distribution of counterfactual events as long as $\lambda_{\text{max}} \geq \lambda_m(t)$ for all $m \in M$ and $t \in \mathbb{R}^+$, similarly as in standard thinning (refer to Appendix A.1 for a proof).

Finally, it is important to note that, by using Gumbel-Max SCMs, our causal model of thinning satisfies the monotonicity assumption [46, 21], discussed in Section 2, and thus the counterfactual thinning probability does not suffer from non-identifiability issues. More formally, we have the following proposition (proven in Appendix A.2):

**Proposition 1** Let $M = \{\lambda_m(t), \lambda_{m'}(t)\}$ and $C$ be the corresponding causal model of thinning. Then, if $\lambda_m(t_i) \geq \lambda_{m'}(t_i)$, it holds that $P_C | X_i = 0; A_i = \lambda_m(t_i); \delta\theta(A_i = \lambda_{m'}(t_i))(X_i = 1) = 0$. Conversely, if $\lambda_m(t_i) \leq \lambda_{m'}(t_i)$, it holds that $P_C | X_i = 1; A_i = \lambda_m(t_i); \delta\theta(A_i = \lambda_{m'}(t_i))(X_i = 0) = 0$.

The above result directly implies that, if a potential event $t_i \in H_{\text{max}}$ was rejected under $\lambda_m(t)$, i.e., $t_i \notin H_m$, then in a counterfactual scenario, if $\lambda_{m'}(t) \leq \lambda_m(t)$, it is impossible that the event is accepted under $\lambda_{m'}(t)$, i.e., $t_i \in H_{m'}$. Conversely, if a potential event $t_i \in H_{\text{max}}$ was accepted under $\lambda_m(t)$, i.e., $t_i \in H_m$, then in a counterfactual scenario, if $\lambda_{m'}(t) \geq \lambda_m(t)$, it is impossible that the event is rejected under $\lambda_{m'}(t)$, i.e., $t_i \notin H_{m'}$.

4 Sampling Counterfactual Events

In this section, we develop a sampling algorithm that, given an observed realization from a temporal point process with a given intensity function, it uses Algorithm 1 and the superposition theorem [22] to generate counterfactual realizations of the temporal point process under a given alternative intensity function. To ease the exposition, we first focus on inhomogeneous Poisson processes and then generalize our algorithm to linear Hawkes processes.

**Inhomogeneous Poisson processes.** Assume we have observed a sequence of real events $H_m$ and this sequence can be accurately characterized, observationally, using an inhomogeneous Poisson process with intensity $\lambda_m(t) = g(t)$, where $g(t) \geq 0$ is a time-varying function. In reality, this sequence of events has not been generated by Lewis’ thinning algorithm but by a natural phenomena of interest. As a result, we cannot directly apply Algorithm 1 since it requires both a sequence of accepted and rejected events. However, we can find plausible sequences of events that Lewis’ thinning algorithm would have rejected if it had accepted the observed sequence of events.

By construction, the intensity of accepted and rejected events $\lambda_{\text{accepted}}(t)$ and $\lambda_{\text{rejected}}(t)$ provided by Lewis’ thinning algorithm should satisfy that $\lambda_{\text{max}} = \lambda_{\text{accepted}}(t) + \lambda_{\text{rejected}}(t)$. Then, if $\lambda_{\text{accepted}}(t) = \lambda_m(t)$, $\lambda_{\text{max}} \geq \max \lambda_m(t)$ and $H_{\text{accepted}} = H_m$, by the superposition theorem, we can find plausible sequences of events that Lewis’ algorithm would have rejected just by sampling from the intensity $\lambda_{\text{rejected}}(t) = \lambda_{\text{max}} - \lambda_m(t)$. Then, these generated sequences of rejected events, together with the sequence of observed events, can be fed into Algorithm 1 to sample sequences of counterfactual events given an alternative intensity $\lambda_{m'}(t)$. Algorithm 2 summarizes the algorithm, where LEWIS(·) samples a sequence of events using Algorithm 4 in Appendix B.1 and ACC(·) samples a counterfactual sequence of accepted events using Algorithm 1.

**Linear Hawkes processes.** Assume we have observed a sequence of real events $H_m$ and it can be accurately characterized, observationally, using a linear Hawkes process with an intensity $\lambda_m(t)$.
Within the algorithm, it is also worth noting that lines 6-11 need to go through the observed events Algorithm 4 in Appendix B.1. Further, for each of these sampled events \( t \) with intensity \( \gamma \), the corresponding process with intensity \( \lambda \) using Algorithm 2, and \( \gamma \) the resulting Algorithm 3 summarizes the resulting algorithm, where ASSIGN(·) returns the observed events \( \mathcal{H}_{m,j} \) assigned to each process \( j \). CR(·) samples a counterfactual sequence of events using Algorithm 2 and LEWIS(·) samples a sequence of events using Algorithm 2 in Appendix B.1. As a result, given two intensities of interest \( \lambda_m(t) \) and \( \lambda_m'(t) \), we can sample sequences of counterfactual events by running Algorithm 3 independently for each of the above processes with

\[
\lambda_{\max} \geq \max \{ \mu, \mu', \max_t \alpha_m g_m(t), \max_t \alpha_m' g_m'(t) \}.
\]

However, to do so, we also need to assign each observed event \( t \) \( t_j \in \mathcal{H}_{m,j} \) to one of the above processes with probability \( \gamma_{m,j}(t_j) / \sum_k \gamma_{m,k}(t_j) \), which is the probability that the process has caused the event. Algorithm 3 summarizes the resulting algorithm, where ASSIGN(·) returns the observed events \( \mathcal{H}_{m,j} \) assigned to each process \( j \). CR(·) samples a counterfactual sequence of events using Algorithm 2 and LEWIS(·) samples a sequence of events using Algorithm 2 in Appendix B.1. Within the algorithm, it is also worth noting that lines 6-11 need to go through the observed events \( t_j \) in chronological order and, for each \( t_j \), the algorithm only accepts counterfactual events from the corresponding process with intensity \( \gamma_{m,j}'(t) \) if the event \( t_j \) has been previously accepted in the counterfactual realization. Moreover, lines 12-18 sample a sequence of events for each of the processes triggered by the counterfactual events that did not exist in the observed sequence of events.

---

**Algorithm 3** It samples a counterfactual sequence of events given a sequence of observed events from a Hawkes process.

1. **Input:** \( \mu_m, \alpha_m, g_m(\cdot), \mu_{m'}, \alpha_{m'}, g_{m'}(\cdot), \mathcal{H}_m, \lambda_{\max}, T \).
2. **Initialize:** \( \mathcal{H}_{m'} = \emptyset \).
3. \( \{ \mathcal{H}_{m,j} \} \leftarrow \text{ASSIGN}(\mathcal{H}_m, \lambda_m(t)) \)
4. \( \mathcal{H}_{m',0} \leftarrow \text{CR}(\gamma_{m,0}(t), \gamma_{m',0}(t), \mathcal{H}_m, 0, \lambda_{\max}, T) \)
5. \( \mathcal{H}_{m'} \leftarrow \mathcal{H}_{m'} \cup \mathcal{H}_{m',0} \)
6. **for** \( t_j \in \mathcal{H}_m \) **do**
7. **if** \( t_j \in \mathcal{H}_{m'} \) **then**
8. \( \mathcal{H}_{m',j} \leftarrow \text{CR}(\gamma_{m,j}(t), \gamma_{m',j}(t), \mathcal{H}_{m,j}, \lambda_{\max}, T) \)
9. \( \mathcal{H}_{m'} \leftarrow \mathcal{H}_{m'} \cup \mathcal{H}_{m',j} \)
10. **end if**
11. **end for**
12. \( \mathcal{H} \leftarrow \mathcal{H}_{m'} \setminus \mathcal{H}_m \)
13. **while** \( |\mathcal{H}| > 0 \) **do**
14. \( t_k \leftarrow \min_{t \in \mathcal{H}} t \)
15. \( \mathcal{H}_{m,k} \leftarrow \text{LEWIS}(\gamma_{m',k}(t), \lambda_{\max}, T) \)
16. \( \mathcal{H} \leftarrow \mathcal{H}_{m,k} \cup \mathcal{H} \setminus \{t_k\} \)
17. \( \mathcal{H}_{m'} \leftarrow \mathcal{H}_{m'} \cup \mathcal{H}_{m,k} \)
18. **end while**
19. **return** \( \mathcal{H}_{m'} \).
(c) Medium # of events

(d) High # of events

Figure 1: Effect of interventions in two inhomogeneous Poisson processes. Panel (a) shows the intensities of the original and the intervened processes (in blue and green, respectively) and a window of interest (dashed vertical lines). Panels (b-d) show the difference in average number of events over time between the counterfactual and the original realizations, where each row corresponds to one process and we group the original realizations in three quantiles according to their overall number of events in the time window of interest. Positive (negative) differences are shown in green (red).

In each experiment, we sample 1,000 realizations from the original process and, for each of these realizations, we sample 100 counterfactual realizations from the intervened process.

5 Experiments on Synthetic Data

In this section, we feed Algorithms[2] and [3] with realizations of synthetic inhomogeneous Poisson processes and linear Hawkes processes and investigate to what extent the counterfactual realizations under alternative intensity functions differ from the original realizations fed to them[4].

Experimental setup. We consider the family of inhomogeneous Poisson processes \( \mathcal{M}(\phi, \alpha, \tau) \) parameterized by weighted combinations of RBF kernels, i.e., \( \lambda(t) = \sum \phi_j \exp(-\alpha_j(t - \tau_j)) \), \( t \geq 0 \), where \( \phi_j, \alpha_j, \tau_j \geq 0 \), and the family of linear Hawkes processes \( \mathcal{M}(\mu, \alpha, \omega) \) defined in Eq. 2.

In each experiment, we first sample 1,000 realizations from a process with one set of parameters using Algorithm 2 (or Algorithm 3). Then, we carry out the above mentioned intervention and, for each of the sampled realizations, we use Algorithm 2 (or Algorithm 3) to sample 100 counterfactual realizations under the resulting alternative set of parameters. Finally, we partition the original realizations in three quantiles according to their overall number of events in a time window of interest and, for each quantile, we look at the number of events in the corresponding counterfactual realizations in the same window of interest. In all experiments, Algorithms 2–3 use 100 samples from the posterior distribution \( P^c | X_t = x; \Lambda_i = \lambda(t_i) \) of each Gumbel noise variable \( U_{i,x} \) to estimate the counterfactual thinning probabilities \( P^c | X_t = x; \Lambda_i = \lambda(t_i) \) of each Gumbel noise variable \( U_{i,x} \).

Results. Figure 1 summarizes the results for two specific inhomogeneous Poisson processes undergoing one of the above mentioned interventions—we found qualitatively similar results for other inhomogeneous Poisson processes. In the top row, the results show that, under our model, an intervention that increases the original intensity of the process, by increasing the amplitude of one of the RBF kernels by approximately an additional half does not have the same effect across realizations. In realizations with a low (high) number of events, the average number of counterfactual events in the window of interest increases up to \( \sim 60\% \) (\( \sim 40\% \)) over time with respect to the average number of events in the original realizations. In the bottom row, we find that this is also true for an intervention that decreases the original intensity of the process, by approximately halving the amplitude of another of the RBF kernels. However, the difference is smaller in relative terms.

Figures 2 summarizes the results for a specific Hawkes process undergoing two of the above mentioned interventions—we found qualitative similar results for other Hawkes processes. We find that,
similarly as in inhomogeneous Poisson processes, the interventions do not have the same effect across realizations. However, in this case, the difference among them is more stark—while the average number of counterfactual events (a) increases by 115% and (b) decreases by 11% for realizations with a low number of events, it (a) increases by 216% and (b) decreases by 21% for realizations with a high number of events. Moreover, we also find that, there is a high variability across counterfactual realizations, especially when $\alpha_m' > \alpha_m$. For example, while the original realizations with a low number of events never contained more than 9 events, there exists counterfactual realizations with more than 200 events. This is due to the self-exciting property of Hawkes processes by which counterfactual events may trigger the emergence of additional counterfactual events, shown as yellow dots in Figure 2 in Appendix C.

6 Experiments on Epidemiological Data

In this section, we run a (simple) variation of Algorithm 3 (refer to Appendix B.4) to quantify the effect of interventions on a networked Susceptible-Infectious-Recovered (SIR) epidemiological model [51] fitted using real event data from an Ebola outbreak in West Africa in 2013-2016 [52].

Experimental setup. We build upon the networked Susceptible-Infectious-Recovered (SIR) epidemiological model introduced by Lorch et al. [51], which is based on temporal point processes. Given a contact network $G = (V, E)$, we represent the times when each node gets infected and recovered using a collection of binary counting processes $Y(t)$ and $W(t)$ and we track the current state of each node using a collection of state variables $X(t) = Y(t) - W(t)$, where $X_i(t) = 1$ indicates node $i \in V$ is infected at time $t$ and $X_i(t) = 0$ indicates it is susceptible or recovered. For each node $i \in V$, we characterize the above counting processes using the following intensities

$$\mathbb{E}[dY_i(t) | \mathcal{H}(t)] = (1 - X_i(t)) \sum_{j \in \mathcal{N}(i)} \beta_{X_j(t)}dt \quad \text{and} \quad \mathbb{E}[dW_i(t) | \mathcal{H}(t)] = \delta X_i(t)dt,$$  \hspace{1cm} (6)

where note that the counting process $Y(t)$ can be viewed as a (networked) multidimensional Hawkes process with stochastic triggering kernels defined by step functions. Refer to Appendix D for more details on how we set the parameters $\beta$ and $\delta$ and how we generate the contact network $G$.

In our experiments, we simulate a realistic outbreak by sampling a realization from the above fitted model. To this end, for each real recorded case up to January 1, 2014 [52], we sample a seed node at random from the same district as the observed case. Figure 9 in Appendix D shows the geographical distribution of seed infections and overall cumulative number of infections in the outbreak. Then, given this sampled realization, we quantify the effect of two types of interventions by sampling counterfactual realizations using a variation of Algorithm 3 (refer to Appendix B.4):

The first real case was recorded on Dec 26, 2013. By January 1, 2014, there were six recorded cases in four different districts.
Figure 3: Effect of interventions where individuals reduce their contacts after the overall number of active infections reaches a varying threshold. In all figures, the black line ("Observation") corresponds to an outbreak sampled from the fitted SIR model, the remaining lines correspond to counterfactual realizations for this outbreak under different interventions, and the shaded regions correspond to 95% confidence intervals. For each threshold and reduction level, we repeat the experiment five times.

Figure 4: Geographical distribution of the overall cumulative number of infections per district in counterfactual outbreaks where all individuals reduce their individual contacts by 25% after the overall number of active infections reaches a certain threshold.

---

**Reduction of number of contacts:** individuals reduce their individual contacts after the overall number of active infections reaches a certain threshold. In one scenario, only individuals from the district with the highest incidence reduce their contacts within the district and get isolated from all other districts and, in another scenario, everyone reduces their individual contacts.

**Vaccination:** a percentage of the overall population receives a vaccine with a certain level of efficacy. Within our model, we measure vaccine efficacy in terms of reduction of the value of the parameter $\beta$, which controls the infection rate between individuals.

In each experiment, to estimate the average and confidence intervals of the outcome of interest (e.g., number of cases), we sample 20 counterfactual realizations.

**Results.** Figures 3-4 summarize the results for the interventions where individuals reduce their individual contacts after the overall number of active infections reaches a certain threshold. Our results suggest that, at all threshold levels, reducing the individual contacts across all districts, even by just 5%, would have been more effective than isolating and reducing the contacts by 50% in the district with higher incidence. Moreover, we also find that, for lower threshold values, the counterfactual outbreaks would have spread to fewer districts and the overall number of infections would have been significantly lower. Figures 5-6 summarize the results for the interventions where a percentage of the overall population receives a vaccine with a certain level of efficacy. Our results suggest that a high level of vaccine effectivity would have not been sufficient to reduce the number of infections if
the percentage of the population who had received the vaccine was low. For example, if less than 20% of the population had received the vaccine, even a vaccine with a 90% effectivity would have been unable to reduce the infections by >70%. If >80% of the population had received the vaccine, a vaccine with just a 60% effectivity would have reduced the infections by >95%. Finally, we find that, similarly as in the scenario where individuals reduce their individual contacts, the reductions in the overall number of infections would have also led to lower geographical dispersion.

7 Conclusions, Limitations and Future Work

Since counterfactual reasoning lies within level three in the “ladder of causation” [45], we have been unable to validate our counterfactual predictions using observational or interventional experiments. That being said, we have made an intuitive assumption—monotonicity—about the causal mechanism of the world—our causal model of thinning—which specifies how changes on the intensity function of a temporal point process may have lead to particular outcomes while holding “every-thing else” fixed, similarly as previous work [21, 41]. In this context, it would be worth to understand the sensitivity of counterfactual realizations to the specific choice of SCM. In our work, we have focused on temporal point processes, however, some of our ideas can be readily extended to spatial point processes. Moreover, it would be interesting to extend our methodology to support nonlinear Hawkes processes and neural Hawkes processes [53, 54]. To this end, the main challenge is that there is no branching process interpretation for nonlinear or neural Hawkes and thus we would have to find alternative ways to bound the value of the relevant intensity functions. Finally, it would be important to carry out a user study in which the counterfactual realizations provided by our algorithm are shared with domain experts (e.g., epidemiologists) and evaluate our sampling algorithm using other real datasets from other applications such as climate change, recommendation systems and voting data.

Acknowledgments and Disclosure of Funding

We would like to thank William Trouleau for sharing with us a pre-processed version of the Ebola dataset as well as the fitted stochastic block model we used in our simulation experiments and thank Stratis Tsirtsis and Abir De for fruitful discussions. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 945719).
References


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Refer to “Remarks” in Section[6] and Section[7]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We cannot think of any potential negative societal impact for our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix[A.2]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please, refer to the README file in the Supplementary Material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Refer to Section[6] and Appendix[5]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Refer to footnote[9]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] The dataset we used was released by the World Health Organization (WHO) for general distribution. It can be reviewed, reproduced or translated for research or private study. The WHO does not explicitly mention any license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]