A Hands-on Primer on Neural Spatio-Temporal Processes

Gerrit Großmann
German Research Centre for Artificial Intelligence (DFKI)
Materials: github.com/gerritgr/spatio-temporal-lecture

Abstract

This tutorial explains the foundations of (spatio-) temporal point processes and how to extend them into the neural domain. It is part of the *Collaborative Intelligence 2024* lecture series at RPTU.

1 Introduction

1.1 Data in Space and Time

Spatio-temporal data encompasses all data that has a relationship with both space and time. From a data science perspective, this type of data is distinct from other data modalities because data points are often highly correlated in both space and time.

Consider infectious events during a pandemic. These events tend to occur in close proximity because of human movement patterns, and they also unfold as a temporal cascade, with new infections following in the wake of previous ones.

There are numerous types of events occurring in both space and time, such as crimes, traffic accidents, natural disasters (e.g., earthquakes, wildfires), terrorist attacks, and events in soccer games. Essentially, anything that happens in nature, society, or science can generate spatio-temporal data.

As computer scientists, we need a framework to understand this data. Our goals are to model the underlying processes, simulate and analyze them, predict outcomes, identify anomalies, and detect causal relations to inform targeted intervention.

At a high level, there are two perspectives to approach spatio-temporal data:

- Spatio-temporal data as **Events (related to point processes)**: This perspective is useful when events are registered or filtered. It is well-suited for analyzing occurrences that happen at specific points in space and time.
- Spatio-temporal data as **Continuous Processes (related to PDEs, GPs)**: This perspective makes sense when using high-resolution sensor data, where continuous observation and measurement are available.

We focus on the former perspective: events in space and time.

It is important to note that this distinction is not always clear. In many cases, both perspectives or a combination of them can be used to study certain phenomena. For instance, neural activity can be analyzed as instantaneous spikes (events) or as continuous signals, such as local field potentials, over time.

1.2 What Makes Spatio-Temporal Event Data Special

Unlike classical ML datasets such as MNIST, where data points are typically considered i.i.d., spatio-temporal event data exhibit unique characteristics.

No Independence Assumption. Spatio-temporal event data often exhibits strong correlations in both spatial and temporal dimensions, particularly for nearby events. Events are typically interdependent. For instance, in the study of earthquakes (seismology), aftershocks are influenced by the initial earthquake, creating a series of events that are not independent of each other.

Spatial Clustering and Temporal Self-Excitation. Events tend to cluster spatially and exhibit temporal self-excitation. In crime analysis, incidents often cluster in specific areas. In phenomena like earthquakes and social media activity, we see that one event increases the likelihood of subsequent ones.

Self-Correction. Self-correction is the opposite of self-excitation. Some processes exhibit self-correction, where an event decreases the likelihood of immediate subsequent events. This can be seen in financial markets, where a large trade may temporarily reduce trading activity. Another example can be found in the study of wildfires: once a fire burns through an area, it temporarily reduces the likelihood of another fire occurring in the same location due to the lack of fuel.

Arrow of Time. The arrow of time (or *temporal priority principle*) states that all causes must precede their effects.

- **Caution 1**: In quantum physics, you can have some notion of *retrocausality*. However, these interpretations remain speculative and are not relevant for us.
- **Caution 2**: Depending on your choice of abstraction, it can seem that the temporal priority principle is broken. For instance, consider that "the rooster crows before sunrise". If we abstract the scenario to just these two events, the temporal sequence appears to suggest that Event A (rooster crows) precedes and possibly causes Event B (sun rises). This simplistic view seems to break the temporal priority principle because it incorrectly implies that a later event (sunrise) is influenced by an earlier one (rooster crowing). To fix this, we need to include the morning light as an additional event.

Seasonality and Periodicity. Seasonality and periodicity refer to patterns that repeat at regular intervals over time. These characteristics are common in spatio-temporal event data. For instance, in environmental studies, temperature and weather patterns exhibit clear seasonal cycles. In economics, consumer spending shows periodic trends related to holidays and fiscal quarters.

1.3 Examples of (Spatio-)Temporal Point Processes

1. **The flashing of a firefly**
   Temporal process. Firefly flashes can exhibit self-exciting behavior as one flash can trigger others nearby.
2. **Suicide**
   Temporal process. Suicides can exhibit both self-exciting (e.g., contagion effect) and self-correcting behaviors depending on various social factors.
3. **Gun violence**
   Spatio-temporal process. Gun violence incidents often show self-exciting behavior, where one event can lead to retaliatory acts in nearby areas.
4. **Earthquake**
   Spatio-temporal process. Earthquakes typically exhibit self-exciting behavior, where an initial quake can trigger aftershocks in the same region.
5. **A goal in a soccer match**
   Temporal process. Goals can exhibit self-exciting behavior, influencing the tempo and strategy of the game, potentially leading to more goals.
6. **Emails sent in a company**
   Temporal process. Email sending often shows self-exciting behavior, where receiving an email can prompt a reply or further communication.
7. **Terrorist attack**
   Spatio-temporal process. Terrorist attacks can exhibit self-exciting behavior, with one attack increasing the likelihood of subsequent attacks in the same or different areas.

8. **Large jump in the stock price of a company**
   Temporal process. Large stock price jumps can show self-correcting behavior, where the market adjusts and stabilizes after significant movements.
9. **Neuron firing**
   Temporal process. For a single neuron, firing typically exhibits refractory periods, a form of self-correction, where the neuron is less likely to fire again immediately after an action potential.
10. **A person coughs**
    Temporal process. The initial cough can irritate the respiratory tract further, making subsequent coughs more likely.
11. **Rainfall**
    Temporal process. Rainfall can exhibit self-correcting behavior, where heavy rain reduces the likelihood of immediate subsequent heavy rain due to the depletion of atmospheric moisture.
12. **Traffic congestion**
    Spatio-temporal process. Traffic congestion can exhibit self-correcting behavior, where severe congestion leads to temporary reductions in traffic flow as drivers avoid the congested area or time period.
13. **Market sales**
    Temporal process. Large sales events, like Black Friday, can show self-correcting behavior, where the surge in purchases is followed by a period of reduced sales activity. .

1.4 Applications

Point processes have a wide range of applications and associated tasks in various fields. Key areas include:

Prediction. Forecast the occurrence of the next event in a sequence. Examples include predicting the next earthquake, anticipating the next social media post, forecasting traffic accidents, or predicting the next transaction in financial markets.

Classification and Regression. Classify or perform regression on point process data. Examples include diagnosing heart problems based on heartbeat patterns, classifying whether a soccer team is good based on game events, or identifying fraudulent activities from transaction data.

Understanding:

- **Clustering and Hotspot Detection.** Identify clusters of events or detect hotspots. Examples include detecting areas with high crime rates, identifying regions prone to traffic accidents, or finding hotspots in disease outbreak data.
- **De-clustering.** Differentiate between primary events and subsequent related events. Examples include identifying the main shock and aftershocks in seismic data, or distinguishing initial disease outbreaks from secondary cases.
- **Parameter Estimation and Model Fitting.** Estimate parameters and fit models to understand underlying processes. For instance, analyzing how the background rate (μ) influences event occurrences and its temporal variations in contexts such as understanding the baseline rate of heartbeats in medical data or the baseline rate of incidents in safety-critical systems.

Causal Discovery. Uncover causal relationships within the data. For instance, identifying causal links between policy changes and crime rates or between the weather and traffic accidents.

Anomaly Detection. Identify unusual or rare events that deviate from the expected pattern. For instance, fraudulent transactions in financial data or spotting unusual patterns in patient vital signs in healthcare.

2 Mathematical Modeling Temporal Point Processes

We begin with temporal point processes (TPPs)—non-spatial ones—which are essentially a series of points or *timestamps* or *event times* or *arrival times*. This is the simplest form of point processes.

Our goal is to develop a model capable of generating and analyzing point process data, assuming that the process is inherently random. We can approach this in two primary ways:

Probabilistic Approach: Construct a probability measure over the space of point process data.

Algorithmic Approach: Develop a stochastic simulation algorithm that generates point-process data.

We start with the former.

2.1 Probabilistic Approach

Let's try to understand TPPs by examining how they induce a likelihood function over a set of event sequences.

A given event sequence is a finite set of non-negative, real-valued timestamps, which is often represented as an increasing sequence $H = [t_0, t_1, \ldots, t_n]$, where $t_i < t_{i+1}$ and $t_i \in \mathbb{R}_{\geq 0}$.

A TPP model identifies a set of event sequences with a probability density (or likelihood). Typically, one considers the infinite set of event sequences where t_n (the maximal time point) is smaller than a given time horizon T. This is denoted as

$$\mathcal{H}_T = \{\text{all event sequences with } t_n \leq T\}.$$

We specify $f : \mathcal{H}_T \to \mathbb{R}_{\geq 0}$ such that:

$$\int_{\mathcal{H}_T} f(H)\, dH = 1 \ .$$

2.1.1 One Technical Thing

Note that it is not obvious what we mean by an integration over a non-Euclidean set like $\mathcal{H}_T$. Integrating over the space of event sequences $\mathcal{H}_T$ means considering all possible ways events can occur up to time T. For a sequence $H = [t_0, t_1, \ldots, t_n]$, the integration involves summing (or integrating) over all possible lengths of sequences and all possible values of the timestamps within the time horizon T. Formally, we can express this as:

$$\int_{\mathcal{H}_T} f(H)\, dH = \sum_{n=0}^{\infty} \int_{0<t_0<t_1<\ldots<t_n\leq T} f(t_0, t_1, \ldots, t_n)\, dt_0\, dt_1\, \ldots\, dt_n \ .$$

2.1.2 Factorization

In temporal point processes, the future can never influence the past, thus, the likeliood can be decomposed into a factorization like in an auto-regressive model.

The factorized likelihood function for the sequence H going until time horizon T is:

$$f(H) = \left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i}) \right) \left(1 - F^{\text{pred}}(T \mid H_{t_n})\right) \ .$$

We define H_t to be he history (H) until (including) timepoint t.

The term $f^{\text{pred}}(\cdot \mid \cdot)$ is known as the *predictive distribution*. It represents the probability density of the next event occurring at time t_{i+1} given the history of all previous events up to time t_i. Formally, f^{pred} is defined on $\mathbb{R}_{\geq t_i}$ (events will always lie in the future) and integrates to one to be a proper PDF.

2.1.3 The Predictive Distribution

The first factor $\left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i})\right)$ represents the product of the predictive distributions for each event in the sequence. Each term $f^{\text{pred}}(t_{i+1} \mid H_{t_i})$ is the probability density of the next event occurring at time t_{i+1}, given the history H_{t_i} up to the previous event time t_i. This captures the sequential nature of the events, where each event is conditioned on the past events.

The second factor $(1 - F^{\text{pred}}(T \mid H_{t_n}))$ represents the probability that no events occur from the last observed event time t_n up to the time horizon T. Here, $F^{\text{pred}}(T \mid H_{t_n})$ is the CDF of the predictive distribution evaluated at the time horizon T. This term ensures that the likelihood properly accounts for the time period from the last event to the time horizon, capturing the probability of no further events occurring in that interval.

In practice, we typically define the likelihood in terms of an *intensity function* rather than a predictive distribution; more on that later.

3 Simulation of Temporal Point Processes

Let's try to understand TPPs by examining how they are simulated.

3.1 Homogeneous Point Processes

A *homogeneous* point process is a type of point process where events occur independently and uniformly over time, with a constant rate λ (lambda). That is, the inter-arrival times between consecutive events follow an exponential distribution with parameter λ. The term *homogeneous* refers to the fact that the rate λ does not change over time. A higher λ results in more frequent events.

We can generate a TPP with rate λ up to a time horizon T by iteratively generating exponentially distributed inter-arrival times until the horizon is reached (see code examples).

The homogeneous point process model, which assumes uniformly occurring events, is useful for specific cases like customers arriving at a service desk or calls at a call center (*uniformly* does not mean uniform distribution in this case). However, its assumptions of a constant event rate and independence between events limit its applicability in real-world situations where events are correlated. To address these limitations, we will next generalize this model.

First, we need some background in the theory of *exponential distributions*, *intensity functions*, *memorylessness*, and *rejection sampling*.

Exponential Distribution and Memorylessness. The *exponential distribution* has a PDF given by $f(x \mid \lambda) = \lambda e^{-\lambda x}$. The expected value is $\frac{1}{\lambda}$. Understanding the exponential distribution is the one most important parts of this lecture. A notable characteristic of the exponential distribution is that it can be interpreted as the time until the next event occurs, assuming the probability of an event in each *infinitesimal* time interval remains constant.

Imagine you flip a highly biased coin each millisecond, where the probability of tails is 0.999. How long do you have to wait until it lands on heads? The exponential distribution provides this information, assuming the limit where the time intervals converge to zero and the probability of tails converges to one.

This leads to an interesting property called *memorylessness*. Formally, memorylessness means that the probability of an event occurring in the future is independent of how much time has already elapsed: $P(T > s + t \mid T > s) = P(T > t)$. Imagine you live in a parallel dimension waiting at a bus stop and buses arrive following an exponential distribution. Assume, on average, a bus arrives every 10 minutes. You have already waited 20 minutes for a bus. What is the probability that the bus arrives within the next minute? The answer is, it does not change. You essentially flip a coin every minute and if you flipped tails 20 times, it does not increase the probability of heads.

The exponential distribution is the only continuous-time distribution with this property.

The discrete counterpart of the exponential distribution is the *geometric distribution*. The geometric distribution describes the number of Bernoulli trials needed to get the first success. Its PMF (probability mass function) is given by $P(X = k) = (1 - p)^{k-1}p$, where p is the probability of success

on each trial. Similar to the exponential distribution, the geometric distribution also exhibits the *memorylessness* property.

3.2 Intensity Functions

We have already discussed that the *exponential distribution* gives us the time until the next event happens under the assumption that the rate remains constant. We can formalize this using an *intensity function*.

The *intensity function* (also known as the *rate function* or *hazard function*) in the context of a point process is a fundamental concept that describes the *instantaneous* rate at which events occur, given that no event has occurred up to that time.

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t))}{\Delta t}$$

The intensity function gives us the rate over time. There is another intuitive interpretation in the context of TPPs: The integral of the intensity function over a given time interval provides the expected number of events within that interval. We can convert between the PDF of the predictive distribution and the corresponding intensity function:

$$f^{pred}(t) = \lambda(t) e^{-\int_0^t \lambda(s)\, ds}$$

$$\lambda(t) = \frac{f^{pred}(t)}{1 - F^{pred}(t)}$$

$$F(t) = \int_0^t f^{pred}(s)\, ds$$

Note that:

- $e^{-\int_0^t \lambda(s)\, ds}$ is called the *exponential decay*. It is the probability that no event has occurred up to time t. It decreases as the cumulative intensity increases, reflecting the decreasing likelihood of 'no event' over time.
- $1 - F(t)$ is called the *survival function*. This term represents the probability that the event has not occurred by time t. It's the complement of the CDF, indicating the likelihood of 'survival' beyond time t. Dividing the PDF by the survival function gives the instantaneous arrival rate. The need to divide the PDF by the survival function arises because the intensity function inherently assumes that the process has "survived" up to the current time without an event occurring.

Caution:

- A valid PDF is non-negative and integrates to 1.
- A valid intensity function is non-negative and integrates to ∞.

3.3 Simulating nonhomogeneous Point Processes

Now, consider the simulation algorithm for *homogeneous point processes*. We want to make two generalizations to enhance its flexibility:

- Allow arbitrary inter-event time distributions.
- Enable the event time distribution to depend on the history, allowing us to model effects such as periodicity, self-excitation, and self-correction.

Fortunately, we can achieve this with a simple modification. Instead of providing a constant rate as input, we supply an time-varying *intensity function* that is conditioned on the history to the algorithm.

This intensity function takes the history H_t and the current time as inputs, and outputs the rate at that time point.

We compute the future rate starting from some time $t \geq 0$, using

$$\lambda : \mathbb{R}_{\geq t} \times \mathcal{H} \to \mathbb{R}_{\geq 0}$$

$$\lambda(t, H_t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t + \Delta t) \mid H_t)}{\Delta t}$$

We can see the *arrow of time* again, given by the fact that only the history until time t can influence the intensity function of future events (after t).

To simulate, we now need to turn a specified $\lambda(\cdot, \cdot)$ into a sample for the next event time. How do we do it?

3.3.1 Sampling from the Intensity Function

Method I: Biased Coin for Every Step. We know that the *exponential distribution* implies flipping a highly biased coin at each infinitesimal time interval. In the context of nonhomogeneous point processes, this coin's probability changes over time based on the event history and the current time. This is what *time inhomogeneity* means: the probability of the coin changing according to the history of events and the current time.

The corresponding algorithm is:

1. **Discretize Time:** Choose a small time step Δt.
2. **Biased Coin:** At each time step t, use a biased coin to decide whether an event occurs. The probability p_t of the event occurring in each time step is given by $p_t = \lambda(t)\Delta t$.
3. **Simulate Until Event:** Continue this process until an event occurs.

While conceptually simple, this method is very inefficient.

Method II: Integrating the Intensity Function. We learned how to sample from the *exponential distribution*. This is equivalent to sampling from a constant *intensity function*. When we set $\lambda = 1$, sampling from an exponential distribution with rate one gives us the area under the curve of the intensity function, i.e., the integral. Thus, we can sample a point x from the exponential distribution and then integrate the intensity function until the area under the curve equals x.

The corresponding algorithm is:

1. **Discretize Time:** Choose Δt (the step-size of the numerical integrator).
2. **Draw a Random Variate:** Draw an exponentially distributed random variate E with $\lambda = 1$.
3. **Numerical Integration:** Integrate $\lambda(t \mid H_t)$ over time until the area under the curve of the intensity equals E. That is, find T such that:

$$\int_{t_i}^{T} \lambda(t \mid H_{t_i})\, dt = E.$$

4. **Event Occurrence:** The time T at which this condition is first met is the time at which the event occurs.

Note that this is a general approach to converting samples from the exponential distribution into samples of any (non-negative) PDF. While this method is better, it remains inefficient when numerical integration is required, as closed-form solutions are typically not available.

Method III: Thinning. We can sample from an exponential distribution very quickly. Translating this to a sample from a non-constant $\lambda(t \mid H_t)$ is not straightforward, but we can use *rejection sampling*.

First, consider an upper bound

$$\lambda^* = \max_{t \in \mathbb{R}_{\geq 0}} \lambda(t \mid H_t),$$

and sample from the exponential distribution with rate λ^*. No compute this upper bound, we can fix the history or find a value that upper bounds all possible histories.

This overestimates the rate, meaning the inter-event times are underestimated. However, we can correct this using rejection sampling.

Specifically, we accept the next event time $T_{\text{candidate}}$ with probability

$$P(\text{accept } T_{\text{candidate}}) = \frac{\lambda(T_{\text{candidate}})}{\lambda^*},$$

and reject the sample otherwise (while still increasing the time).

The advantage of this method is that we can ignore what happened before the time $T_{\text{candidate}}$. The rejection process fully corrects for the overestimation of the rate. This is the beauty of rejection sampling.

In other words, the exponential distribution's *memoryless property* ensures that each candidate time is independent of previous times. This is crucial because it means we do not need to track the history of past rejections beyond the current candidate time.

Note that this simple rejection mechanism is only possible because we translated the process to *intensity functions*.

The corresponding algorithm is:

1. **Determine Maximum Intensity:** Determine an upper bound of the intensity function $\lambda_{\max}$.
2. **Initialize Time:** Set $t = 0$.
3. **Sample Waiting Time:** Draw an exponentially distributed random variate E with rate parameter $\lambda_{\max}$.
4. **Update Time:** Update the time: $t = t + E$.
5. **Acceptance Test:** Generate another uniform random number V in the interval $[0, 1]$. If $V \leq \frac{\lambda(t|H_t)}{\lambda_{\max}}$, accept the event at time t. Otherwise, reject the event and go back to step 3.

3.3.2 Explosive Behavior

Note that, in point processes with arbitrary intensity functions, you can generate *explosive behavior* (sometimes called *zenoid behavior* - Zeno's Paradoxes). This means that the number of points you generate becomes infinite, and you never reach the time horizon. A simple method to achieve this (assuming $h = 1$) is to define $\lambda(t, H_t)$ as follows:

- If H_t is empty: Converge to infinity within $[0, 0.5]$.
- If H_t is non-empty:
  - Define t_{max} to be the maximum of H_t (the most recent event).
  - Let the intensity function convert to infinity within $(H_t - t_{\text{max}})/2$.

Note that the integral of the intensity needs to be infinite, but the intensity should typically not convert to infinity itself. It is quite easy to inadvertently cause explosive behavior when using self-exciting processes.

3.3.3 Hawkes Processes

Hawkes processes are special parameterized forms of point processes given as

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

where μ is the background intensity and ϕ is a kernel. Typical kernels include:

- *Exponential Kernel:*
$$\phi(u) = \alpha e^{-\beta u}$$
- *Power-Law Kernel:*
$$\phi(u) = \frac{\alpha}{(u+c)^p}$$
where α, c, and p are parameters.
- *Gaussian Kernel:*
$$\phi(u) = \alpha \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right)$$
where α, μ, and σ are parameters.
- *Sum of Exponentials:*
$$\phi(u) = \sum_{j=1}^{J} \alpha_j e^{-\beta_j u}$$
where α_j and β_j are parameters for each term.

You can only get an upper bound for this if you fix the number of past events.

Self-Correcting Process. A *self-correcting process* is given by:

$$\lambda(t, H_t) = \exp\left(\mu t - \sum_{i:t_i<t} \alpha\right)$$

where $\mu > 0$ is the background rate that increases over constantly (due to the multiplication with t) and each new event decreases the intensity modulated by parameter $\alpha > 0$.

Stationarity. In the context of point processes, *stationarity* refers to the property that the statistical properties of the process do not change over time. For a homogeneous point process, this means that the base rate μ remains constant, and the process behaves the same way at all points in time.

3.3.4 Rethinking the Likelihood

Earlier, we defined the likelihood of a given sequence H going until time horizon T as:

$$f(H) = \left(\prod_{i=1}^{n} f^{\text{pred}}(t_{i+1} \mid H_{t_i})\right) \left(1 - F^{\text{pred}}(T \mid H_{t_n})\right) .$$

Now, we can define the factorization based on the intensity function:

$$\begin{aligned}
& f(H) \\
&= \left(\prod_{i=1}^{n} \lambda(t_i \mid H_{t_i}) \exp\left(-\int_{t_{i-1}}^{t_i} \lambda(s \mid H_s)\, ds\right)\right) \exp\left(-\int_{t_n}^{T} \lambda(s \mid H_s)\, ds\right) \\
&= \left(\prod_{i=1}^{n} \lambda(t_i \mid H_{t_i})\right) \exp\left(-\int_{0}^{T} \lambda(t \mid H_t)\, dt\right)
\end{aligned}$$

In this factorization:

1. The fist part $\lambda(t_i \mid H_{t_i})$ accounts for the likelihood of events occurring at each t_i.
2. The exponential term $\exp\left(-\int_{t_{i-1}}^{t_i} \lambda(t \mid H_t)\, dt\right)$ represents the likelihood of not observing any events in the intervals between the observed events.
3. The exponential term $\exp\left(-\int_{t_n}^{T} \lambda(t \mid H_t)\, dt\right)$ represents the likelihood of not observing any events between the last event and the time horizon.

Maximum Likelihood Estimation. In the case of a *homogeneous* TPP with constant rate λ, the likelihood of an event sequence H with n events in $[0, T]$ reduces to:

$$L(\lambda \mid H) = \lambda^n \exp(-\lambda T) \ .$$

We can infer the maximum likelihood estimator (MLE) for λ:

$$\hat{\lambda} = \frac{n}{T} \ ,$$

which should be intuitive because we have already established that the expected number of events is given by the integral over the intensity. Hence, the integral of λ over $[0, T]$ should be n.

In the case of a *nonhomogeneous* TPP, where the rate $\lambda(t)$ varies over time, the likelihood function becomes more complex and maximizing this likelihood function generally requires numerical methods (note that you typically want to maximize some kernel parameter and not $\lambda(t)$ itself). Techniques such as Expectation-Maximization or gradient ascent can be employed.

3.4 Neural TPPs

We can transition from "normal" (i.e., parametric) TPPs to *neural* TPPs by replacing the parameterized form of the intensity function:

$$\lambda(t, H_t) = \mu + \sum_{i:t_i<t} \phi(t - t_i)$$

with a *neural network*.

The main challenge is encoding the history, which is a sequence of arbitrary length. You cannot directly feed this sequence into an MLP. Instead, you can use RNNs, transformers, or *DeepSets*. For the latter two, you need some form of *positional encoding*.

Note that the great thing about intensity functions is that we do not need to worry about normalization; we just need to ensure the value is > 0. If we want to use *thinning*, we also need to ensure that we have an upper bound for our output.

An additional challenge is training the neural TPPs. We need to ensure:

1. The intensity at the point where the next event happens is large.
2. The intensity at all points before that is comparably small.

Alternatively, you can model the *predictive distribution* with a neural network. In this case, you need to normalize it and ensure that the representation of the predictive PDF allows for efficient sampling and computation of the likelihood.

3.5 Marked TPPs

A *marked* temporal point process (MTPP) extends the concept of TPPs by associating each event time with a *mark* from a set $\mathcal{M}$. The event sequence is represented as $H = [(t_0, m_0), (t_1, m_1), \ldots, (t_n, m_n)]$, where t_i are the event times and $m_i \in \mathcal{M}$ are the corresponding marks. Each mark can represent additional information. The family of intensity functions is defined as $\{\lambda_m(t, H_t)\}_{m \in \mathcal{M}}$, where $\lambda_m(t, H_t)$ denotes the intensity function for events with mark m, conditioned on the history H_t. We assume that all intensity functions run in parallel to generate a sequence.

4 Spatio-Temporal Point Processes

Data Representation A *spatio*-temporal point process (STPP) models events that occur in both time and space. The data is represented as a sequence of tuples, $H = [(t_1, s_1), (t_2, s_2), (t_3, s_3), \ldots]$, where t_i denotes the event time and $s_i \in \mathcal{S} \subset \mathbb{R}^2$ represents the spatial location in a two-dimensional space (we only consider $2D$ data in this tutorial).

4.1 Intensity Function

To define the intensity function, we consider an infinitesimal grid over space and time:

$$\lambda(s,t \mid H_t) := \lim_{\Delta s \to 0, \Delta t \to 0} \frac{P(\text{Event in } B(s,\Delta s) \times [t,t+\Delta t) | H_t)}{|B(s,\Delta s)| \Delta t} \,.$$

where $|B(s,\Delta s)|$ is the Lebesgue measure (i.e., area) of the ball $B(s,\Delta s)$, a 2d-ball (i.e., disk), centered at s with a radius Δs).

The intensity function $\lambda(s,t \mid H)$ describes the rate at which events occur at time t and location s, given the history H. It can be defined as:

$$\lambda(s,t \mid H_t) = \mu g_0(s) + \sum_{(t_i,s_i) \in H_t} g_1(t,t_i) g_2(s,s_i)$$

where μ is the background intensity, $g_0(s)$ is a spatial distribution, $g_1(t,t_i)$ is a temporal kernel, and $g_2(s,s_i)$ is a spatial kernel.

Hawkes Process. A typical Hawkes process kernel for spatio-temporal data is:

$$g_1(t,t_i) = \alpha \exp(-\beta(t-t_i))$$

where α and β are parameters. The spatial kernel $g_2(s,s_i)$ can be a truncated normal distribution centered at s_i.

Self-Correcting Process. A typical self-correcting process for spatio-temporal data is defined as:

$$\lambda(s,t,H) = \mu \exp(g_0(s) + \beta t - \sum \alpha g_2(s,s_i))$$

where $g_0(s)$ is a spatial distribution over S and $g_2(s,s_i)$ is a spatial distribution centered at s_i.

4.2 Simulation Algorithm

The simulation algorithm is quite similar to the temporal case.

Homogeneous Process:

1. Simulate event times with rate λA, where λ is the rate and A is the area of the space.
2. For each event time, place the event uniformly over the space.

Nonhomogeneous Process:

1. Simulate candidate event times and locations as in the homogeneous case.
2. Reject each candidate point with a probability based on the intensity function.

4.3 Likelihood and Predictive Disribution

The computation of the likelihood translates from the temporal case, we simply add another integration step over the spatial domain:

$$f(H) = \left(\prod_{i=1}^{n} \lambda(t_i \mid H_{t_i}) \right) \exp\left(- \int_S \int_0^T \lambda(s,t \mid H_t) \, dt \right)$$

Likewise, we have a predictive distribution (for $t > t_n$) given by

$$f^{\text{pred}}(s,t \mid H_t) = \lambda(s,t \mid H_t) \exp\left(- \int_S \int_{t_n}^{t} \lambda(u,\tau \mid H_t) \, d\tau \, du \right)$$

5 Neural STPPs

The challenges with *neural* models for spatio-temporal point processes (STPPs) are similar to those in the temporal case:

1) We need a neural network that can effectively process the history.
2) We need a method to predict a *predictive distribution* (or *intensity function*) that can be evaluated and trained efficiently.

In this section, we will examine a simplified version of a specific paper that addresses these challenges: Zhou et al. [2022].

In the paper, they solve the first challenge by using a transformer architecture that processes the sequence of events. They solve the second challenge by predicting weight for each event (that encode the events influence) and expressing the intensity function as parametric PDF that re-weights the history as specified by the neural network.

More formally, for an input event sequence $H = [(t_1, s_1), (t_2, s_2), \ldots, (t_n, s_n)]$, they use a NN that maps each event i to three weights $[w_i, \gamma_i, \beta_i]$ and then parameterizes the intensity function as:

$$\lambda(s, t|H_t) = \sum_{i=1}^{n} w_i k_s(s, s_i; \gamma_i) k_t(t, t_i; \beta_i)$$

The **spatial kernel** k_s is defined as:

$$k_s(s, s_i) = \alpha^{-1} \exp(-\gamma_i \|s - s_i\|)$$

where $\alpha = \int_S \exp(-\gamma_i \|s - s_i\|) ds$ is the normalization constant. Note that, strictly speaking, we don't need a normalized kernel.

The **temporal kernel** k_t is defined as:

$$k_t(t, t_i) = \exp(-\beta_i \|t - t_i\|)$$

The parameters w_i, γ_i, and β_i for each event have specific roles in the intensity function:

- w_i: Intensity magnitude. It scales the contribution of each event to the overall intensity.
- γ_i: Spatial influence. It controls how quickly the event's influence decays over space. Larger values mean more localized influence.
- β_i: Temporal decay rate. It determines how quickly the event's influence diminishes over time. Larger values mean quicker decay.

To predict $[w_i, \gamma_i, \beta_i]$, the authors employ a Transformer architecture. They use an encoder in a sequence-to-sequence framework to map the event sequence to a sequence of embeddings.

Specifically, for each event, they predict a latent embedding $[\mu_i, \sigma_i]$. They then sample a random variate z_i based on $[\mu_i, \sigma_i]$ using the reparameterization trick. A decoder subsequently maps z_i to $[w_i, \gamma_i, \beta_i]$.

6 Resources

1. **A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications**
   `https://arxiv.org/pdf/1708.02647`
2. **Neural Temporal Point Processes: A Review**
   `https://arxiv.org/abs/2104.03528`
3. **Lecture Notes: Temporal Point Processes and the Conditional Intensity Function**
   `https://arxiv.org/pdf/1806.00221`
4. **Modeling Events and Interactions through Temporal Processes – A Survey**
   `https://arxiv.org/abs/2303.06067`
5. **ICML 2018 Tutorial - Learning with Temporal Point Processes**
   `https://learning.mpi-sws.org/tpp-icml18/`

References

Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. Neural point process for learning spatiotemporal event dynamics. In *Learning for Dynamics and Control Conference*, pages 777–789. PMLR, 2022.