点过程
Temporal Point Process

龚舒凯
中国人民大学
Renmin University of China
School of Applied Economics/ School of Statistics
shukai_gong@ruc.edu.cn

目录

1 点过程概述

1.1 点过程的概念

时空点过程

时空点过程是一类用于建模事件发生时刻或位置的**随机过程**模型。点过程的一次实现是时间或空间上一系列随机的点，用来表达事件的发生时间或位置。

- Poisson 过程、Cox 过程、Hawkes 过程都是时空点过程。

先做出如下记号约定

历史信息

记事件 i 发生的时间戳为 t_i，事件的标签/类别为 k_i(可以没有)，对于点过程的一系列观测 $\{(t_i,k_i)\}_{i=1}^N$，我们定义

$$\mathcal{H}_{t_n}=\{(t_1,k_1),\cdots,(t_n,k_n)\}$$

为包括时刻 t_n 在内的历史信息，定义

$$\mathcal{H}_{t-}=\{(t_i,k_i),t_i<t\}$$

为时刻 t 之前的所有历史信息。

时空点过程可以用两种方式进行刻画

CPDF 刻画时空点过程

通过刻画事件发生时间 $(t_1,\cdots,t_n)$ 的分布来刻画时空点过程。记历史事件为 $\mathcal{H}_{t_{n-1}}=\{t_1,\cdots,t_{n-1}\}$，$f(t_n|\mathcal{H}_{t_{n-1}})$ 为给定历史信息 $\mathcal{H}_{t_{n-1}}$ 下，事件 t_n 发生的条件概率密度函数，则所有事件的联合分布 $f(t_1,\cdots,t_n)$ 可以写成

$$f(t_1,\cdots,t_n)=\prod_{i=1}^n f(t_i|\cdots,t_{i-2},t_{i-1})=\prod_{i=1}^n f(t_i|\mathcal{H}_{t_{i-1}})$$

根据这种刻画方式，我们可以很容易将时空点过程与随机过程里学习过的**更新过程、Poisson 过程**相联系。

- 当相邻两事件之间的时间间隔 $\tau_n=t_n-t_{n-1}$ 独立同分布时，该时空点过程就是更新过程。

$$f(t_n|\mathcal{H}_{t_{n-1}})=f(t_n|t_{n-1})=f(t_n-t_{n-1})=f(\tau_n)$$

- 当相邻两事件之间的时间间隔 $\tau_n=t_n-t_{n-1}\overset{\text{i.i.d}}{\sim}\text{Exp}(\lambda)$ 时，该时空点过程就是 Poisson 过程。

$$f(t_n|\mathcal{H}_{t_{n-1}})=f(t_n|t_{n-1})=f(t_n-t_{n-1})=f(\tau_n)=\lambda e^{-\lambda\tau_n}$$

但上述两点过程都是非常特殊的点过程 (Markov 性)，有些点过程依赖于全部的历史信息 $\mathcal{H}_{t_{n-1}}$，导致使用 CPDF 刻画时空点过程是非常复杂的。因此，我们引入了另一种刻画时空点过程的方式。

条件强度函数刻画时空点过程

定义事件 t_{n+1} 发生在 t 时刻的 CPDF 为 $f(t|\mathcal{H}_{t_n}), t > t_n$，其对应的累积分布函数为 $F(t|\mathcal{H}_{t_n}) = \int_{t_n}^{t} p(\tau|\mathcal{H}_{t_n})\mathrm{d}\tau$，则条件强度函数 (Conditional Intensity Function, CIF) $\lambda^*(t) = \lambda(t|\mathcal{H}_{t-})$ 定义为

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_{t_n})}{1 - F(t|\mathcal{H}_{t_n})}, t \in (t_n, t_{n+1}]$$

[注]：我们一般用 * 表示依赖于历史信息 $\mathcal{H}_{t_n}$。

条件强度函数有一个非常好的物理意义：

条件强度函数的物理意义

条件强度函数 $\lambda^*(t)$ 表示在给定历史信息 $\mathcal{H}_{t_n}$ 下 t 时刻的平均事件点数。

证明. 考虑 t 时刻附近的一个无穷小的时间区间 $[t, t + \mathrm{d}t]$

$$\begin{aligned}
\lambda^*(t)\mathrm{d}t &= \frac{f(t|\mathcal{H}_{t_n})\mathrm{d}t}{1 - F(t|\mathcal{H}_{t_n})} = \frac{f(t_{n+1} \in [t, t + \mathrm{d}t]|\mathcal{H}_{t_n})}{1 - \int_{t_n}^{t} p(\tau|\mathcal{H}_{t_n})\mathrm{d}\tau} \\
&= \frac{p(t_{n+1} \in [t, t + \mathrm{d}t]|\mathcal{H}_{t_n})}{p(t_{n+1} \notin [t_n, t]|\mathcal{H}_{t_n})} = \frac{p(t_{n+1} \in [t, t + \mathrm{d}t], t_{n+1} \notin [t_n, t]|\mathcal{H}_{t_n})}{p(t_{n+1} \notin [t_n, t]|\mathcal{H}_{t_n})} \\
&= p(t_{n+1} \in [t, t + \mathrm{d}t]|t_{n+1} \notin [t_n, t], \mathcal{H}_{t_n}) \\
&= p(t_{n+1} \in [t, t + \mathrm{d}t]|\mathcal{H}_{t-}) \\
&= \mathbb{E}[N([t, t + \mathrm{d}t])|\mathcal{H}_{t-}]
\end{aligned}$$

由于 $\mathrm{d}t$ 是取的无穷小，所以 $\lambda^*(t)\mathrm{d}t = \mathbb{E}[N([t, t + \mathrm{d}t])|\mathcal{H}_{t-}]$ 就是在 t 时刻的平均事件点数。 □

事实上可以证明，**CPDF 和 CIF 是一一对应的**。

CPDF 和 CIF 的一一对应关系

若点过程的事件发生时刻的 CPDF 为 $f(t|\mathcal{H}_{t-})$，对应的 CIF 为 $\lambda^*(t) = \lambda(t|\mathcal{H}_{t-})$，则两者之间的关系为

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_{t-})}{1 - F(t|\mathcal{H}_{t-})}$$

$$f(t|\mathcal{H}_{t-}) = \lambda^*(t) \exp\left(-\int_{t_n}^{t} \lambda^*(s)\mathrm{d}s\right)$$

证明. 只说明第二个公式的正确性：

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_{t-})}{1 - F(t|\mathcal{H}_{t-})} = \frac{\frac{\mathrm{d}}{\mathrm{d}t}F(t|H_{t_n})}{1 - F(t|H_{t_n})} = -\frac{\mathrm{d}}{\mathrm{d}t}\log(1 - F(t|H_{t_n}))$$

因此根据 Newton-Leibniz 公式，

$$\int_{t_n}^{t} \lambda^*(s)\mathrm{d}s = -\left(\log(1-F(t|\mathcal{H}_{t_n}))\Big|_{t_n}^{t}\right) = -(\log(1-F(t|\mathcal{H}_{t_n})) - \log(1-F(t_n|\mathcal{H}_{t_n})))$$

因为 $F(t_n|\mathcal{H}_{t_n}) = 0$（第 $n+1$ 个时间戳 t_{n+1} 与第 n 个时间戳 t_n 重合，但点过程不会在同一时刻有 ≥ 2 个点），所以

$$\begin{aligned}\int_{t_n}^{t} \lambda^*(s)\mathrm{d}s = -\log(1-F(t|\mathcal{H}_{t_n})) &\Rightarrow F(t|\mathcal{H}_{t_n}) = 1-\exp\left(-\int_{t_n}^{t}\lambda^*(s)\mathrm{d}s\right)\\ &\Rightarrow f(t|\mathcal{H}_{t_n}) = \lambda^*(t)\exp\left(-\int_{t_n}^{t}\lambda^*(s)\mathrm{d}s\right)\end{aligned}$$

□

1.2 经典的点过程

齐次 Poisson 过程

齐次 Poisson 过程的条件强度函数为

$$\lambda^*(t) = \lambda$$

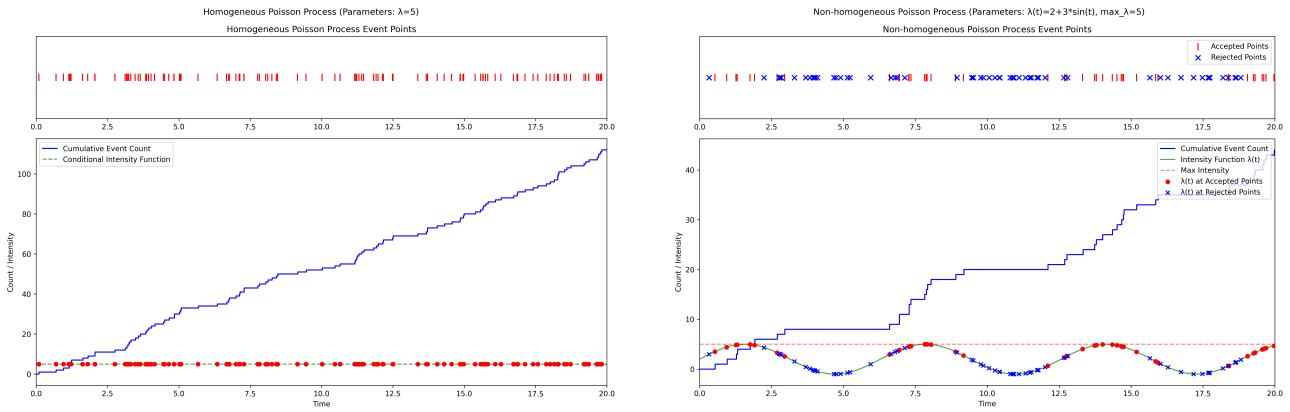
即事件发生的强度在时间上是恒定的，与历史信息和当前时刻无关。

非齐次 Poisson 过程

非齐次 Poisson 过程的条件强度函数为

$$\lambda^*(t) = \lambda(t)$$

虽然事件发生的强度在时间上是不恒定的，但是与历史信息和当前时刻无关。



(a) 齐次 Poisson 过程　　(b) 非齐次 Poisson 过程 (此处 $\lambda(t) = 2 + 3\sin(t)$)

图 1: 泊松过程对比：齐次与非齐次

Hawkes 过程

Hawkes 过程是一种自激励点过程，其条件强度函数为

$$\lambda^*(t) = \mu + \sum_{t_i<t} \phi(t-t_i), \quad \mu \geq 0, \alpha > 0, \phi(t-t_i) \in (0, \infty)$$

其中 μ 称为基础强度，$\phi(t-t_i)$ 称为激励函数，常见的选择为一个关于 t 的减函数 $\phi(t-t_i) = \alpha e^{-\beta(t-t_i)}$，用来表示事件 t_i 对事件 t 的“激励作用”。

- 每发生一个事件，CIF 会增加一个 α 的量级。
- 但事件发生过后，立刻会以指数衰减的速度往 μ 递减，直到下一个事件发生，CIF 才会再次增加。

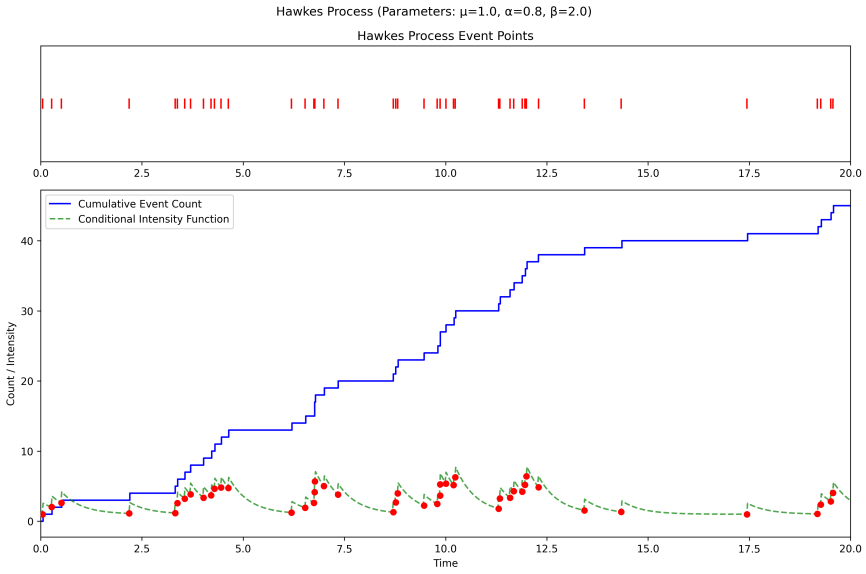


图 2: Hawkes 过程

自纠正过程 (Self-Correcting Process)

自纠正过程是一种自激励点过程，其条件强度函数为

$$\lambda^*(t) = \exp\left(\mu t - \sum_{t_i<t} \alpha\right), \quad \mu > 0, \alpha > 0$$

- 随时间的推移，$\lambda^*(t)$ 会以指数增长的速度增加。
- 但事件发生过后，$\lambda^*(t)$ 立刻衰减到原来的 $e^{-\alpha}$ 倍。这表明**随着一个事件的发生，新事件发生的强度会瞬间减小**。这是一种“自纠正”的过程。

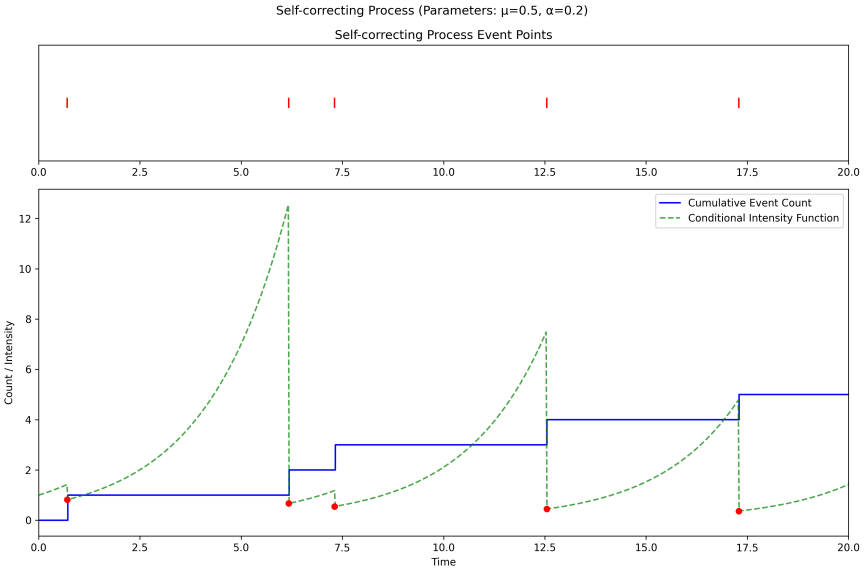

Self-correcting Process (Parameters: μ=0.5, α=0.2)
Self-correcting Process Event Points
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Cumulative Event Count
Conditional Intensity Function
Count / Intensity
0
2
4
6
8
10
12
Time

图 3: 自纠正过程

2 点过程的参数估计与采样

2.1 参数估计

对于时空点过程模型，参数估计指的是**如何从 $\{t_i\}_{i=1}^N$ 中推断出模型的条件强度函数 $\lambda^*(t)$ 的参数**。如果将以 θ 为参数的条件强度函数记为 $\lambda_\theta^*(t)$，我们关心的就是如何从数据中估计出 θ。

点过程模型的似然函数

给定点过程的观测 $\{t_i\}_{i=1}^N \in [0,T)$，其似然函数为

$$L(\theta) = \left(\prod_{i=1}^N \lambda_\theta^*(t_i)\right) \exp\left(-\int_0^T \lambda_\theta^*(s)\mathrm{d}s\right)$$

从而 $\theta^* = \arg\max_\theta L(\theta)$。

证明. 似然函数可以被如下分解

$$L(\theta) = f(t_1|\mathcal{H}_{t_0}) \cdots f(t_N|\mathcal{H}_{t_{N-1}}) \cdot (1 - F(T|\mathcal{H}_{t_N}))$$

最后一项为 $1 - F(T|\mathcal{H}_{t_N})$ 是因为 t_N 是最后一个观察点，$[t_N, T)$ 区间内是没有事件发生的。而 $F(T|\mathcal{H}_{t-}) = \int_{t_n}^T \lambda_\theta^*(s)\mathrm{d}s$ 表示在 t_n 时刻之后，事件发生在 $[t_n, T)$ 的概率。应该乘上一项 $1 - F(T|\mathcal{H}_{t_N})$。

$$\begin{aligned}
L(\theta) &= \left(\prod_{i=1}^N f(t_i|\mathcal{H}_{t_{i-1}})\right) \cdot (1 - F(T|\mathcal{H}_{t_N})) \\
&= \left(\prod_{i=1}^N \lambda_\theta^*(t_i) \exp\left(-\int_{t_{i-1}}^{t_i} \lambda_\theta^*(s)\mathrm{d}s\right)\right) \cdot \exp\left(-\int_{t_N}^T \lambda_\theta^*(s)\mathrm{d}s\right) \\
&= \left(\prod_{i=1}^N \lambda_\theta^*(t_i)\right) \exp\left(-\sum_{i=1}^N \int_{t_{i-1}}^{t_i} \lambda_\theta^*(s)\mathrm{d}s - \int_{t_N}^T \lambda_\theta^*(s)\mathrm{d}s\right) \\
&= \left(\prod_{i=1}^N \lambda_\theta^*(t_i)\right) \exp\left(-\int_0^T \lambda_\theta^*(s)\mathrm{d}s\right)
\end{aligned}$$

□

2.2 采样算法

对点过程的参数估计本质上是为了拟合出一个使得观测值似然最高的点过程模型。再介绍点过程的仿真采样算法之前，我们需要了解点过程采样的意义是什么。可以概括为以下三点：

1. **做预测**：给定观测值，我们希望仿真采样出未来的事件发生情况，以便做出预测。
2. **模型评估**：可以选定某个点过程的前一半观测值估计模型参数，用估计好参数的点过程模型仿真采样后一半数据，与真实的后一半观测值做比对（比对仿真采样出来的分布与后一半观测值的分布是否相同），从而实现模型评估。
3. **统计量计算**：有一些非常复杂的统计量，无法通过 CIF 算出闭式解，这个时候就需要对点过程进行仿真采样，通过大量的采样数据计算统计量。

齐次 Poisson 过程的采样

Algorithm 1: Simulation of a Homogeneous Poisson Process with Rate λ, on $[0,T]$.

Input: λ, T
Initialize $n=0$, $t_0=0$;
while *True* **do**
  Generate $u \sim$ `uniform(0,1)`;
  Let $w=-\ln u/\lambda$;　　// so that $w \sim$ `exponential(`λ`)`
  Set $t_{n+1}=t_n+w$;
  if $t_{n+1}>T$ **then**
    return $\{t_k\}_{k=1,2,\ldots,n}$
  else
    Set $n=n+1$;
  end
end

证明. 这是因为齐次 Poisson 过程相邻时间戳的时间间隔是独立同指数分布 Exp(λ) 的，即

$$p(\tau)=p(t_{n+1}-t_n)=\lambda e^{-\lambda\tau},\ \tau>0$$

□

那么能不能用类似的方法对**非齐次 Poisson 过程采样呢?** 注意到对于非齐次 Poisson 过程

$$P(N(t+s)-N(t)=n)=\frac{\left(\int_s^{t+s}\lambda(s)\mathrm{d}s\right)^n\exp(-\int_s^{t+s}\lambda(s)\mathrm{d}s)}{n!}$$

$$\Rightarrow P(\tau_1>t)=P(N(t)=0)=\exp\left(-\int_0^t\lambda(s)\mathrm{d}s\right)$$

$$\Rightarrow f(\tau_1)=\lambda(t)\exp\left(-\int_0^t\lambda(s)\mathrm{d}s\right)$$

考虑第二个时间间隔 τ_2 的分布:

$$P(\tau_2>t|\tau_1=s)=P(N(t+s)-N(s)=0)=\exp\left(-\int_s^{t+s}\lambda(s)\mathrm{d}s\right)$$

(注意，此时 $P(\tau_2>t|\tau_1=s)$ 与 τ_1 的取值有关！)

$$\Rightarrow P(\tau_2>t)=\int P(\tau_2>t|\tau_1=s)f(s)\mathrm{d}s=\int\exp\left(-\int_s^{t+s}\lambda(s)\mathrm{d}s\right)\lambda(s)\exp\left(-\int_0^s\lambda(s)\mathrm{d}s\right)\mathrm{d}s$$

$$\Rightarrow P(\tau_2>t)=\int P(\tau_2>t|\tau_1=s)f(s)\mathrm{d}s=\int\exp\left(-\int_0^{t+s}\lambda(s)\mathrm{d}s\right)\lambda(s)\mathrm{d}s$$

$$\Rightarrow f(\tau_2)=\int\lambda(s)\lambda(s+t)\exp\left(-\int_0^{t+s}\lambda(s)\mathrm{d}s\right)\mathrm{d}s$$

这些形式都非常复杂，需要多次积分操作，显然是不现实的，无法通过对时间间隔的分布采样来仿真模拟。

基于 Thinning Algorithm 的非齐次 Poisson 过程的采样

基于 Thinning Algorithm 的非齐次 Poisson 过程的采样指的是：假定我们感兴趣的非齐次 Poisson 过程的强度函数为 $\lambda(t)$，我们先选定一个强度为 $\bar{\lambda}=\sup_{t\in[0,T]}\lambda(t)$ 的齐次 Poisson 过程，从中采样生成

$\{t_i\}_{i=1}^N$，然后对每个 t_i，以 $\dfrac{\lambda(t_i)}{\bar{\lambda}}$ 的概率保留 t_i，以 $1-\dfrac{\lambda(t_i)}{\bar{\lambda}}$ 的概率删除 t_i，则剩下的点就形成了非齐次 Poisson 过程的一次实现。

Algorithm 2: (Lewis and Shedler, 1979, p.7, Algorithm 1) Simulation of an Inhomogeneous Poisson Process with Bounded Intensity Function $\lambda(t)$, on $[0,T]$.

Input: $\lambda(t)$, T

Initialize $n=m=0$, $t_0=s_0=0$, $\bar{\lambda}=\sup_{0\le t\le T}\lambda(t)$;
while $s_m < T$ **do**
  Generate $u \sim$ `uniform(0,1)`;
  Let $w=-\ln u/\bar{\lambda}$; `// so that` $w \sim$ `exponential(`$\bar{\lambda}$`)`
  Set $s_{m+1}=s_m+w$; `// {`s_m`} are points in the homo. Poisson process`
  Generate $D \sim$ `uniform(0,1)`;
  if $D \le \lambda(s_{m+1})/\bar{\lambda}$ **then** `// accepting with probability` $\lambda(s_{m+1})/\bar{\lambda}$
    $t_{n+1}=s_{m+1}$; `// {`t_n`} are points in the inhomo. Poisson process`
    $n=n+1$; `// updating` n `to the index of last point in {`t_n`}`
  end
  $m=m+1$; `// updating` m `to the index of last point in {`s_m`}`
end
if $t_n \le T$ **then**
  return $\{t_k\}_{k=1,2,\dots,n}$
else
  return $\{t_k\}_{k=1,2,\dots,n-1}$
end

可以证明通过“先采样、后稀疏”的方法采样出来的点过程是符合非齐次 Poisson 过程的。

证明. 根据非齐次 Poisson 过程的定义

$$P(N(b)-N(a)=0)=\exp\left(-\int_a^b \lambda(s)\mathrm{d}s\right)$$

要证明我们采出的点是符合非齐次 Poisson 过程的，只需要证明在任意的区间 $[a,b)$ 上，Thinning 算法采出来没有点发生的概率为 $\exp\left(-\int_a^b \lambda(s)\mathrm{d}s\right)$ 即可。

首先我们说明，从 $[t_0,T]$ 上的齐次 Poisson 过程中采出来的 n 个点 $(t_1,\cdots,t_n)$ 的联合分布为 $\lambda^n\exp\left(-\lambda(T-t_0)\right)$。这是因为

$$p_\lambda(t_1,\cdots,t_n,t_{n+1}>T)=\prod_{k=1}^{n}\lambda e^{-\lambda(t_k-t_{k-1})}\cdot e^{-\lambda(T-t_n)}=\lambda^n\exp\left(-\lambda(T-t_0)\right)$$

其次我们定义在区间 (a,b) 上通过齐次 Poisson 过程生成了 n 个点，且这 n 个点全部被删除的概率为 p_n，则

$$p_n=\int_a^b\int_{t_1}^b\cdots\int_{t_{n-1}}^b\left[\bar{\lambda}^n\exp\left(-\bar{\lambda}(b-a)\right)\prod_{k=1}^{n}\left(1-\frac{\lambda(t_k)}{\bar{\lambda}}\right)\right]\mathrm{d}t_1\cdots\mathrm{d}t_n$$

最中间的被积项表示“先采出来 n 个齐次 Poisson 过程的点，再全部删除的概率”。因为 $a\le t_1<\cdots<t_n\le b$ 是有序的，所以 t_1 的积分范围是 $[a,b]$，t_2 的积分范围是 $[t_1,b]$，以此类推。

再定义有序序列的 $(t_1,\cdots,t_n)$ 的全排序序列 $(s_1,\cdots,s_n)$。则这样的全排序共有 $n!$。由于排序的无序性，此时

s_i 的积分区间均为 $[a,b]$。所以

$$\begin{aligned} p_n &= \int_a^b \int_{t_1}^b \cdots \int_{t_{n-1}}^b \left[\bar{\lambda}^n \exp\left(-\bar{\lambda}(b-a)\right) \prod_{k=1}^n \left(1 - \frac{\lambda(t_k)}{\bar{\lambda}}\right) \right] \mathrm{d}t_1 \cdots \mathrm{d}t_n \\ &= \frac{\exp(-\bar{\lambda}(b-a))}{n!} \int_a^b \int_a^b \cdots \int_a^b \prod_{k=1}^n \left(\bar{\lambda} - \lambda(s_k)\right) \mathrm{d}s_1 \cdots \mathrm{d}s_n \\ &= \frac{\exp(-\bar{\lambda}(b-a))}{n!} \left(\int_a^b \bar{\lambda} - \lambda(s) \mathrm{d}s \right)^n \\ &= \frac{\exp(-\bar{\lambda}(b-a))}{n!} \left(\bar{\lambda}(b-a) - \int_a^b \lambda(s) \mathrm{d}s \right)^n \end{aligned}$$

(a,b) 上通过齐次 Poisson 过程生成了 $n = 0, 1, \cdots,$ 个点，于是在区间 (a,b) 上没有点发生的概率为

$$\begin{aligned} \sum_{i=0}^n p_i &= \sum_{n=0}^\infty \frac{\exp(-\bar{\lambda}(b-a))}{n!} \left(\bar{\lambda}(b-a) - \int_a^b \lambda(s) \mathrm{d}s \right)^n = \exp(-\bar{\lambda}(b-a)) \cdot \exp\left(\bar{\lambda}(b-a) - \int_a^b \lambda(s) \mathrm{d}s \right) \\ &= \exp\left(- \int_a^b \lambda(s) \mathrm{d}s \right) \end{aligned}$$

Thinning 算法采出来在 (a,b) 上没有点发生的概率为 $\exp\left(-\int_a^b \lambda(s)\mathrm{d}s\right)$ 和非齐次 Poisson 过程在 (a,b) 上没有点发生的概率一致，所以 Thinning 算法采出来的点过程是符合非齐次 Poisson 过程的。 □

无论是齐次还是非齐次 Poisson 过程，强度函数 $\lambda^*(t)$ 都是历史无关的。但对于 Hawkes 过程一类的点过程而言，其在 t 时刻的 CIF 依赖于历史信息 $\mathcal{H}_{t-}$，即

$$\lambda(t|\mathcal{H}_{t-}) = \mathbb{E}[N([t,t+\mathrm{d}t])|\mathcal{H}_{t-}] = \lim_{h \to 0} \frac{P\left(N(t+h) - N(t) = 1 | \mathcal{H}_{t-}\right)}{h}$$

在不同的实现中，$\mathcal{H}_{-t}$ 的取值是不同的，**这意味着** $\{\lambda(t|\mathcal{H}_{-t}), t > 0\}$ **也是一个随机过程**，我们称这样的 $\lambda(t|\mathcal{H}_{-t})$ 为**随机强度函数**。

基于 Ogata Thinning 的 Hawkes 过程采样

Algorithm 3: (Ogata, 1981, p.25, Algorithm 2) Simulation of a Univariate Hawkes Poisson with Exponential Kernel $\gamma(u) = \alpha e^{-\beta u}$, on $[0,T]$.

Input: μ, α, β, T

Initialize $\mathcal{T} = \emptyset$, $s = 0$, $n = 0$;
while $s < T$ **do**
  Set $\bar{\lambda} = \lambda(s^+) = \mu + \sum_{\tau \in \mathcal{T}} \alpha e^{-\beta(s-\tau)}$;
  Generate $u \sim$ `uniform(0,1)`;
  Let $w = -\ln u/\bar{\lambda}$; `// so that` $w \sim$ `exponential(`$\bar{\lambda}$`)`
  Set $s = s + w$; `// so that` s `is the next candidate point`
  Generate $D \sim$ `uniform(0,1)`;
  if $D\bar{\lambda} \leq \lambda(s) = \mu + \sum_{\tau \in \mathcal{T}} \alpha e^{-\beta(s-\tau)}$ **then** `// accepting with prob.` $\lambda(s)/\bar{\lambda}$
    $n = n + 1$; `// updating the number of points accepted`
    $t_n = s$; `// naming it` t_n
    $\mathcal{T} = \mathcal{T} \bigcup \{t_n\}$; `// adding` t_n `to the ordered set` $\mathcal{T}$
  end
end
if $t_n \leq T$ **then**
  return $\{t_k\}_{k=1,2,\ldots,n}$
else
  return $\{t_k\}_{k=1,2,\ldots,n-1}$
end

相比之前的 Thinnin 算法，Ogata Thinning 算法注意到，在 Hawkes 过程中，给定 $t_1, \cdots, t_k$，**在区间** (t_k, t_{k+1}) **上的强度函数** $\lambda^*(t)$ **是一个固定的值**，只有 t_{k+1} 是未知的。因此 Hawkes 过程下一个点的生成可以看作是对非齐次 Poisson 过程生成第一个点。

3 点过程面临的的问题

1. 灵活性: 相比一般的序列模型，点过程模型使用 CIF 函数的形式和 CIF 的参数给模型施加了一个较强的假设，这可能会影响了模型的表达能力。

- **解决方法:** 非参数化点过程模型 (无限参数点过程模型，例如深度点过程模型)。

2. 非时变: 点过程模型的参数 θ 被假设为不随时间变化，无法刻画时变系统。以 Hawkes 过程为例

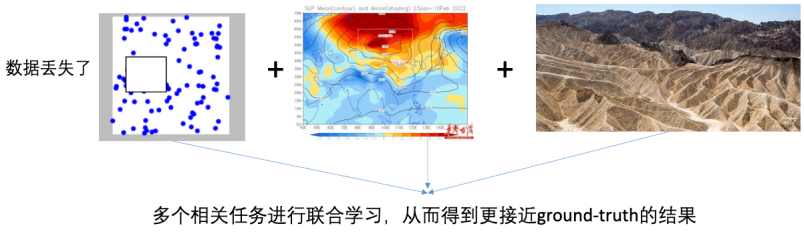
$$\lambda^*(t) = \mu + \sum_{t_i<t} \phi(t - t_i)$$

这里基础强度 μ 和激励函数 $\phi(\cdot)$ 都是非时变的，但在实际应用中，参数会随着时间会发生变化，例如**神经脉冲序列数据**。

- **解决方法:** 时变点过程模型。对于缓慢变化的情况，可以将参数随着时间的变化建模为随机过程或者确定函数；对于突变的情况，考虑变点检测 (changepoint detection)

3. 多任务性: 当单个任务上的观测数据非常充足时，可以学习到非常贴近 ground-truth 的参数。但是单个任务观测数据量不足时，模型容易出现过拟合的现象。

- **解决方法:** 多任务学习。将多个相关任务进行联合学习，从而得到更接近 ground-truth 的结果。



4. 不确定性: MLE(频率学派) 是一种点估计方法，只能给出一个确定的参数估计值，无法刻画模型的不确定性，影响了其在高风险场景中的应用。

- **解决方法:** 贝叶斯点过程模型。将点过程的参数视作随机变量，通过观测来更新随机变量的后验分布:

$$f(\theta|t_1, \cdots, t_N) = \frac{f(t_1, \cdots, t_N|\theta)f(\theta)}{f(t_1, \cdots, t_N)}$$

- 不过贝叶斯方法的计算是更加复杂的，在大数据集上，使用频率学派的方法比使用贝叶斯学派的方法更加高效。

5. 效率低: MLE(频率学派) 在计算似然的时候需要计算积分

$$\log p(t_1, \cdots, t_N|\theta) = \sum_{i=1}^{N} \log \lambda_\theta^*(t_i) - \int_0^T \lambda_\theta^*(s)\mathrm{d}s$$

积分项对于简单的 CIF 有解析解，但对大部分一般化的 CIF 无法计算，只能通过一些数值模拟的方法 (Monte Carlo, Quadrature 积分法)，效率很低。

- **解决方法:** 积分网络 (基于神经网络近似计算积分)，score，不用似然作为目标函数。

MAP(贝叶斯学派) 的问题在于：**目前没有发现与点过程似然函数共轭的先验分布**，参数后验的求解只能在非共轭的情况下进行。在非共轭的情况下，我们只能采用 MCMC、变分推断 (Variational Inference)、Laplace approximation 等方法进行，但是这些方法的计算过程也很复杂，大部分情况下是没有解析式的，导致推断效率很低。

- **解决方法:** 数据增广。

4 深度点过程

研究点过程的意义

- 事件序列 (Event sequence) 数据在生活中是非常广泛的：人们社交媒体的发帖/互动、金融交易数据、医疗记录等都是事件序列，研究这种事件序列的规律有着重要的意义。
- 与一般的序列数据相比 (比如 Time-series)，事件序列中有三个特点：
  - **异步性 (asynchronous):** 事件序列在连续时间域中发生的多个事件的**采样间隔是不相等的**。换言之，对于事件序列的建模而言，事件发生的**时间间隔** τ_i 和事件发生的顺序同等重要。
  - **多模态 (multi-modal):** 事件序列中的事件有多种类型。
  - **历史相关性 (history-dependent):** 事件的发生是依赖于历史信息的。

  最朴素的的 RNN 虽然能建模历史信息，但无法刻画事件发生的时间间隔 τ 的分布。
- 点过程是**是连续时间频域上的随机过程**，是在连续时间频域上建模离散事件的重要工具。

基本上，**深度点过程的训练 (参数估计) 可以被概括如下**：

1. 将观测到的事件序列 $\{(t_i, k_i)\}_{i=1}^N$ 转化为历史信息序列 $\{\mathbf{h}(t_i) \in \mathbb{R}^H\}_{i=1}^N$，其中 $\mathbf{h}(t_i)$ 聚合了 $\mathcal{H}_{t_i} = \{(t_j, k_j) : j \le i\}$ 的信息，事件类型 $k_i \in \mathcal{K} = \{1, \cdots, K\}$。这一步基本都是用 RNN 或者 Transformer 完成的。

2. 对于基于 CIF 的方法：用历史信息建模强度函数 $\lambda(t|\mathcal{H}_{t-})$，然后基于此计算似然函数，最后用梯度下降法估计参数，即

$$\mathcal{L}(\theta) = \sum_i \log \lambda_\theta(t_i|\mathcal{H}_{t_{i-1}}) - \int_0^T \lambda_\theta(t|\mathcal{H}_t)\mathrm{d}t$$
$$\Rightarrow \theta^* = \arg\max_\theta \mathcal{L}(\theta)$$

3. 对于基于 CPDF 的方法：用历史信息建模事件发生时间间隔的 CPDF $f(\tau|\mathcal{H}_{t-})$，直接根据 CPDF 最大化似然函数

$$\mathcal{L}(\theta) = \sum_i \left[\log f_\theta(\tau_i|\mathcal{H}_{t_{i-1}}) + \log g_\theta(k_i|\mathcal{H}_{t_{i-1}})\right]$$
$$\Rightarrow \theta^* = \arg\max_\theta \mathcal{L}(\theta)$$

   $g_\theta(k_i|\mathcal{H}_{t_{i-1}})$ 是一个分类头，即 $g_\theta(k_i|\mathcal{H}_{t_{i-1}}) = \mathrm{softmax}(\mathbf{W}^g\mathbf{h}(t_{i-1}) + \mathbf{b}^g)$，其中 $\mathbf{W}^g \in \mathbb{R}^{K\times H}, \mathbf{b}^g \in \mathbb{R}^K$。

预测 (仿真采样) 流程可以被概括如下：在得到了训练好的模型，我们可以通过以下方式预测下一个事件的发生：已知历史信息 $\mathcal{H}_{t_N} = \{(t_i, k_i)\}_{i=1}^N$

1. 如果建模的是 CIF $\lambda_\theta(t|\mathcal{H}_{t-})$
   - 计算 $f(t|\mathcal{H}_{t-}) = \lambda(t|\mathcal{H}_{t-}) \exp\left(-\int_{t_N}^t \lambda(s|\mathcal{H}_{s-})\mathrm{d}s\right), t > t_N$
   - 采出来的下一个时间戳为 $\hat{t}_{N+1} = \int_{t_N}^\infty tp(t|\mathcal{H}_{t-})\mathrm{d}t$

- 采出来的下一个标签为 $\hat{k}_{N+1} = \arg\max_k \frac{\lambda_k(\hat{t}_{N+1}|\mathcal{H}_{t_N})}{\lambda(\hat{t}_{N+1}|\mathcal{H}_{t_N})}$

2. 如果建模的是 CPDF $f_\theta(\tau|\mathcal{H}_{t-}) = f_\theta(t - t_N|\mathcal{H}_{t_N})$

- 采出来的下一个时间戳为 $\hat{t}_{N+1} = t_N + \int_0^\infty \tau f(\tau|\mathcal{H}_{t_N})\mathrm{d}\tau = t_N + \mathbb{E}_f[\tau]$
- 采出来的下一个标签为 $\hat{k}_{N+1} = \arg\max_k g_\theta(k|\mathcal{H}_{t_N})$

4.1 RMTPP (KDD 2016)

RMTPP 是首个提出使用 RNN 聚合历史信息，用于建模 CIF 的点过程模型。具体而言，令 $\mathbf{x}_i$ 为包含事件时间戳间隔信息的向量（如相邻事件时间间隔 $\mathbf{x}_i = (\mathbf{t}_i - \mathbf{t}_{i-1})$），将其输入 RNN 中。RNN 的每个隐藏状态表示为

$$\mathbf{h}(t_i) = \mathrm{RNN}(\mathbf{h}(t_{i-1}), \mathbf{x}_i) = f(\mathbf{W}^h\mathbf{h}(t_{i-1}) + \mathbf{W}^x\mathbf{x}_i + \mathbf{b}^h)$$

然后用聚合出来的历史信息向量 $\mathbf{h}(t_i)$ 来建模条件强度函数

$$\lambda(t|\mathcal{H}_{t-}) = \phi(t - t_i|\mathbf{h}(t_i)), \phi(\cdot) \geq 0$$

在 RMTPP 中，作者使用指数函数建模条件强度函数

$$\phi(\tau|\mathbf{h}(t_i)) = \exp(w_\phi\tau + \mathbf{v}_\phi^\top\mathbf{h}(t_i) + \mathbf{b}_\phi)$$

从而对数似然函数转化为

$$\log L(\theta) = \sum_i \left[\log\phi(t_{i+1} - t_i|\mathbf{h}(t_i)) - \int_0^{t_{i+1}-t_i}\phi(\tau|\mathbf{h}(t_i))\mathrm{d}\tau\right] - \int_0^T \phi(t - T|\mathbf{h}(t_N))\mathrm{d}t$$

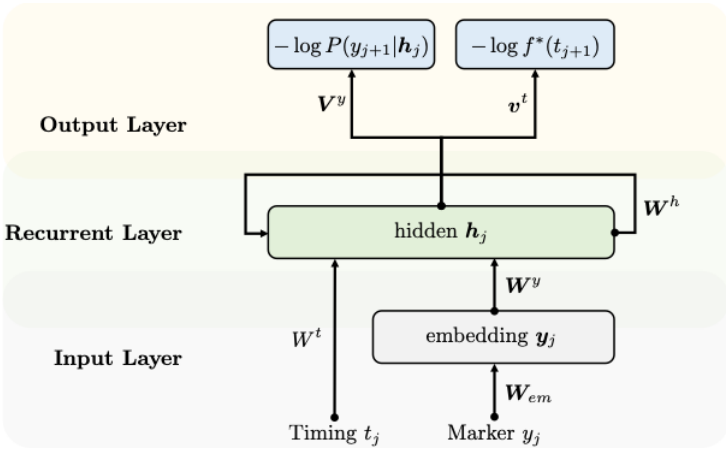


图 4: RMTPP

4.2 NHP (NIPS 2017)

4.3 FullyNN (NIPS 2019)

FullyNN 指出，RMTPP 使用指数函数建模 CIF 的假设太强，希望使用更一般化的 $\lambda^*(t) = \phi(t - t_i|\mathbf{h}(t_i)) = \phi(\tau|\mathbf{h}(t_i))$。然而，在对数似然函数中，不可避免的需要对 $\phi(\tau|\mathbf{h}(t_i))$ 进行积分，这几乎是不可能算的。所以

FullyNN 转而计算积分强度函数 $\Phi(\tau|\mathbf{h}(t_i))$

$$\Phi(\tau|\mathbf{h}(t_i)) = \int_0^\tau \phi(s|\mathbf{h}(t_i))\mathrm{d}s$$

对其进行微分立刻得到 $\phi(\tau|\mathbf{h}(t_i)) = \dfrac{\partial\Phi(\tau|\mathbf{h}(t_i))}{\partial\tau}$。这样一来，对数似然就被转化为

$$\log L(\theta) = \sum_i \left[\log \frac{\partial\Phi(\tau = t_{i+1} - t_i|\mathbf{h}(t_i))}{\partial\tau} - \Phi(t_{i+1} - t_i|\mathbf{h}(t_i))\right] - \Phi(T - t_N|\mathbf{h}(t_N))$$

这个似然函数很好算的原因是，$\Phi(\tau|\mathbf{h}_i)$ 是我们直接通过神经网络输出得到的，而 $\phi(\tau|\mathbf{h}_i)$ 可以通过 Pytorch 的自动微分轻易实现。作者使用 FFN 接受时间间隔 τ 和历史信息 $\mathbf{h}_i$ 建模 $\Phi(\tau|\mathbf{h}(t_i))$

$$\Phi(\tau|\mathbf{h}(t_i)) \triangleq Z_i^*(\tau) = \mathrm{softplus}(\mathbf{W}^{(3)}\tanh(\mathbf{W}^{(2)}\tanh(\mathbf{W}^{(1)}\tau + \tilde{\mathbf{b}}^{(1)}) + \mathbf{b}^{(2)}) + \mathbf{b}^{(3)})$$

其中 $\tilde{\mathbf{b}}^{(1)} = \mathbf{V}\mathbf{h}(t_i) + \mathbf{b}^{(1)}$，$\mathbf{h}(t_i) \in \mathbb{R}^H$ 是历史信息向量，$\mathbf{V} \in \mathbb{R}^{D\times H}$。

在具体实现上还有一些细节要注意。如图5所示，由于 $\Phi(\tau|\mathbf{h}(t_i))$ 是一个关于 τ 的非负递增函数，故

- τ 到 FNN 的第一层以及 FNN 内部层的权重矩阵都是非负的，即 $\mathbf{W}^{(1)} \in \mathbb{R}_+^{D\times 1}, \mathbf{W}^{(2)} \in \mathbb{R}_+^{D\times D}, \mathbf{W}^{(3)} \in \mathbb{R}_+^{1\times D}$。偏置项 $\mathbf{b}^{(1)} \in \mathbb{R}^D, \mathbf{b}^{(2)} \in \mathbb{R}^D, \mathbf{b}^{(3)} \in \mathbb{R}$ 并没有正负性要求。
- 最终输出 $Z_i^*(\tau) = \Phi(\tau|\mathbf{h}(t_i))$ 的激活函数是 $\mathrm{softplus}(\tau) = \log(1 + \exp(\tau))$。

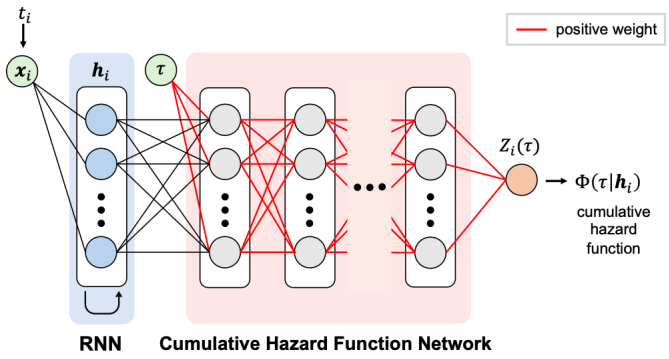


图 5: FullyNN

4.4 THP (ICML 2020)

此前的工作 (RMTPP, NHP, FullyNN) 都是用 RNN 来建模历史信息的，但这么做存在三个显著的问题：

1. **无法捕获长距依赖关系**。即使是像 LSTM/GRU 这样配备了遗忘门的 RNN，也无法捕获长距离依赖关系。某一状态的历史信息被保留的概率随着时间的增长呈指数级下降。
2. **RNN 训练难度高**。RNN 存在梯度爆炸或梯度消失的问题。
3. **不可并行**。只有在当前状态的历史信息被计算完毕后，才能计算下一个状态的历史信息。

我们很自然而然的想到用 Transformer 代替 RNN 来建模历史依赖关系。Self-attention 机制可以很好的捕获长距依赖关系，也可以并行训练；相比 RNN 只有一两层隐藏层的结构，Transformer 层可以叠的很高。**不过，由**

于点过程是定义在连续事件频域上的，而 Transformer 建模的是离散序列之间的关系，所以还需要进行一些调整才能使用。

THP 的架构如图6所示。给定包含 L 个事件的事件序列 $\mathcal{S}=\{(t_j,k_j)\}_{j=1}^{L}$，其中 k_j 为 t_j 时刻发生的事件的标签，一共有 k 个类别：$k_j\in\{1,\cdots,K\}$。

1. **时序位置编码：**对于每个时间戳 t_j，定义其位置编码 $\mathbf{z}(t_j)\in\mathbb{R}^M$ 为：

$$[\mathbf{z}(t_j)]_i=\begin{cases}\sin\left(\dfrac{t_j}{10000^{i/m}}\right), & \text{if } i \text{ is even}\\ \cos\left(\dfrac{t_j}{10000^{(i-1)/m}}\right), & \text{if } i \text{ is odd}\end{cases}$$

   其中 $[\mathbf{z}(t_j)]_i$ 表示向量 $\mathbf{z}(t_j)$ 的第 i 个元素。

2. **事件嵌入：**对于每个事件的标签 k_j，记 $\mathbf{k}_j$ 为其 one-hot 向量。我们定义可学习的嵌入矩阵为 $\mathbf{U}\in\mathrm{R}^{M\times K}$，则事件 k_j 的嵌入为 $\mathbf{U}\mathbf{k}_j\in\mathrm{R}^M$。

3. **输入矩阵：**记 $\mathbf{Y}=[\mathbf{k}_1,\cdots,\mathbf{k}_L]\in\mathrm{R}^{K\times L}$ 为事件标签的矩阵，$\mathbf{Z}=[\mathbf{z}(t_1),\cdots,\mathbf{z}(t_L)]\in\mathrm{R}^{m\times L}$ 为时间戳的位置编码矩阵，则整个事件序列 $\mathcal{S}=\{(t_j,k_j)\}_{j=1}^{L}$ 可以被表示为如下输入矩阵：

$$\mathbf{X}=(\mathbf{U}\mathbf{Y}+\mathbf{Z})^\top\in\mathrm{R}^{L\times M}$$

   输入矩阵 $\mathbf{X}$ 中每一行都代表一个事件 (包括时间戳和事件类别) 的嵌入。

4. **Multihead Self-Attention 模块:** 将输入矩阵 $\mathbf{X}$ 使用 MLP 投影到第 i 个头的查询 $\mathbf{Q}_i$、键 $\mathbf{K}_i$ 和值 $\mathbf{V}_i$，然后计算第 i 个头经注意力机制后的输出：

$$\mathbf{Q}_i=\mathbf{X}\mathbf{W}_i^Q,\ \mathbf{K}_i=\mathbf{X}\mathbf{W}_i^K,\ \mathbf{V}_i=\mathbf{X}\mathbf{W}_i^V$$
$$\mathbf{S}_i=\mathrm{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^\top+\mathrm{Mask}}{\sqrt{d_k}}\right)\mathbf{V}_i,\quad i=1,\cdots,H$$

   这里 $\mathbf{W}_i^Q\in\mathrm{R}^{M\times d_k},\mathbf{W}_i^K\in\mathrm{R}^{M\times d_k},\mathbf{W}_i^V\in\mathrm{R}^{M\times d_v}$ 是可学习的权重矩阵，d_k,d_v 是查询、键、值的维度。注意，为了防止 Self-attention 机制看到未来信息，我们给 $\mathbf{Q}_i\mathbf{K}_i^\top$ 加上一个 mask 矩阵，

$$[\mathrm{Mask}]_{ij}=\begin{cases}-\infty, & \text{if } j>i\\ 0, & \text{otherwise}\end{cases}\in\mathbb{R}^{L\times L}$$

   然后将所有的头拼接起来就得到了多头注意力机制的输出：

$$\mathbf{S}=[\mathbf{S}_1,\cdots,\mathbf{S}_H]\in\mathrm{R}^{L\times(H\times d_v)}$$

   可以设置 $M=H\times d_v$，这样 $\mathbf{S}$ 的维度就是 $L\times M$，如果不这样的话，也可以给拼接后的 $\mathbf{S}$ 再乘一个权重矩阵 $\mathbf{W}^O\in\mathrm{R}^{H\times d_v\times M}$。

$$\mathbf{S}=[\mathbf{S}_1,\cdots,\mathbf{S}_H]\mathbf{W}^O\in\mathbb{R}^{L\times M}$$

   其中 $\mathbf{W}^O\in\mathbb{R}^{(H\times d_v)\times M}$ 是可学习的权重矩阵。

5. **历史信息的输出 $\mathbf{h}(t_j)$:** 将 $\mathbf{S}$ 输入到一个 FFN 中，得到历史信息的输出 $\mathbf{H}$，然后从这个矩阵中抽取出第 j 行就得到了 t_j 时刻的历史信息 $\mathbf{h}(t_j)$：

$$\mathbf{H}=\mathrm{FFN}(\mathbf{S})=\mathrm{ReLU}(\mathbf{S}\mathbf{W}^1+\mathbf{b}^1)\mathbf{W}^2+\mathbf{b}^2\in\mathbb{R}^{L\times M}$$
$$\mathbf{h}(t_j)=\mathbf{H}(j,:)\in\mathbb{R}^M$$

   其中 $\mathbf{W}^1\in\mathbb{R}^{M\times M_h},\mathbf{b}^1\in\mathbb{R}^{M_h},\mathbf{W}^2\in\mathbb{R}^{M_h\times M},\mathbf{b}^2\in\mathbb{R}^M$ 是可学习的权重矩阵。

6. **建模 CIF:** 此前建模出的历史信息 $\mathbf{h}(t_j)$ 是离散的，而我们关心的 CIF $\lambda^*(t) = \lambda(t|\mathcal{H}_{t-})$ 是连续的，所以我们需要将 $\mathbf{h}(t_j)$ 映射到连续的 CIF 上。具体而言，假设对每种事件类型，我们都有一个 CIF $\lambda_k(t|\mathcal{H}_{t-}), k \in \{1, \cdots, K\}$，则整个事件序列的条件强度函数被定义为

$$\lambda(t|\mathcal{H}_{t-}) = \sum_{k=1}^{K} \lambda_k(t|\mathcal{H}_{t-})$$

对于每种事件类型 k，我们使用一个 FFN 将 $\mathbf{h}(t_j)$ 映射到一个标量，然后再通过一个 softplus 激活函数得到 CIF：

$$\lambda_k(t|\mathcal{H}_{t-}) = \text{softplus}_k(\underbrace{\alpha_k \cdot \frac{t-t_j}{t_j}}_{\text{current}} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t_j)}_{\text{history}} + \underbrace{b_k}_{base}), t \in [t_j, t_{j+1})$$

给定历史信息 $\mathcal{H}_{t-} = \{(t_j, k_j), t_j < t\}$，该 CIF 只定义在 $[t_j, t_{j+1})$ 上 (最后一个时间戳到下一个时间戳的区间)。从函数形式上，softplus 为 $\text{softplus}_k(x) = \beta_k \log(1 + \exp(x/\beta_k))$，保证了 $\lambda_k(t|\mathcal{H}_{t-})$ 是正的。值得注意的是，这个 CIF 中每一项的形式都有着特别的意义：

- current 项: 本质上是对离散时间戳 t_j 和 t_{j+1} 之间进行插值。α_k 衡量了差值项的重要程度。显然，这一项使得 $\lambda(t|\mathcal{H}_{t-})$ 在除了事件到达时刻 t_j 以外的时刻都是连续的。
- history 项: 把历史信息 $\mathbf{h}(t_j)$ 加权求和得到一个标量
- base 项: 表示在没有历史信息的情况下事件发生的基础强度。

7. **最大似然估计训练模型:** 对于一个观测到的从 $[t_0, t_L]$ 的事件序列 $\mathcal{S}$，根据前面建模的 CIF $\lambda(t|\mathcal{H}_{t-})$，我们可以计算似然函数，通过最大化似然函数来估计模型参数 θ：

$$\mathcal{L}(\mathcal{S};\theta) = \underbrace{\sum_i \log \lambda(t_i|\mathcal{H}_{t_{i-1}})}_{\text{event log-likelihood}} - \underbrace{\int_{t_1}^{t_L} \lambda(t|\mathcal{H}_{t-})\mathrm{d}t}_{\text{non-event log-likelihood}}$$

non-event log-likelihood 可以由 Monte Carlo 积分或者数值积分的方法计算。如果有多个观测序列 $\mathcal{S}_1, \cdots, \mathcal{S}_N$，则最大似然估计为

$$\theta^* = \arg\max_\theta \sum_{n=1}^{N} \mathcal{L}(\mathcal{S}_n;\theta)$$

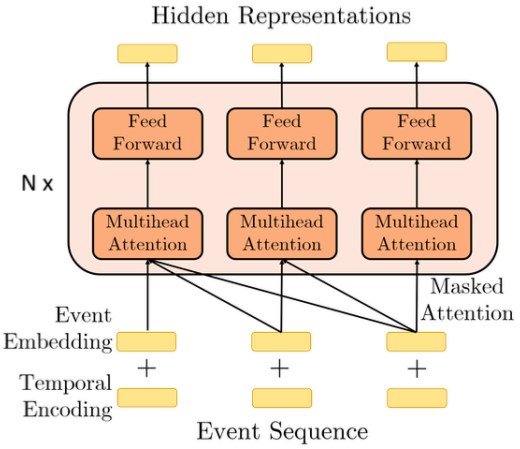


图 6: THP (Transformer Hawkes Process)

4.5 SAHP (ICML 2020)

SAHP 的思路和 THP 类似，也是使用 Transformer 来建模历史信息，但在时序位置编码和条件函数的建模上有所不同。延续 SAHP 中的记号，我们有：

1. **时序位置编码：** 对于事件 (t_i, k_i)，其时序位置编码 $\mathbf{z}(t_i) \in \mathbb{R}^M$ 为：

$$[\mathbf{z}(t_j)]_i = \begin{cases} \sin\left(\dfrac{i}{10000^{i/M}} \cdot j + w_i t_j\right), & \text{if } i \text{ is even} \\ \cos\left(\dfrac{i}{10000^{(i-1)/M}} \cdot j + w_i t_j\right), & \text{if } i \text{ is odd} \end{cases} \triangleq \begin{cases} \sin\left(\omega_i \cdot j + w_i t_j\right), & \text{if } i \text{ is even} \\ \cos\left(\omega_i \cdot j + w_i t_j\right), & \text{if } i \text{ is odd} \end{cases}$$

   其中 $\omega_i = \dfrac{i}{10000^{i/M}} \in \mathbb{R}$ 是固定的相位，$w_i \in \mathbb{R}$ 是可学习的参数。所以本质上相比 THP 的位置编码，SAHP 的位置编码多了一个可学习的向量 $\mathbf{w} = [w_1, \cdots, w_M] \in \mathbb{R}^M$。SAHP 学习事件类型嵌入的方法与 THP 相同，最终得到输入矩阵 $\mathbf{X} = (\mathbf{UY} + \mathbf{Z})^\top \in \mathbb{R}^{L \times M}$。

2. **建模 CIF:** SAHP 直接用加聚得到的历史信息 $\mathbf{h}(t_j)$ 来建模 Hawkes 过程的强度函数中的每个参数。每种事件类型 k 的 CIF 可写为

$$\lambda_k(t|\mathcal{H}_{t-}) = \text{softplus}(\mu_{k,j} + (\eta_{k,j} - \mu_{k,j}) \exp(-\gamma_{k,j}(t - t_j)), t \in [t_j, t_{j+1})$$

   其中 $\mu_{k,j} = \text{gelu}(\mathbf{w}_{k,\mu}^\top \mathbf{h}(t_j)) \in \mathbb{R}$，$\eta_{k,j} = \text{gelu}(\mathbf{w}_{k,\eta}^\top \mathbf{h}(t_j)) \in \mathbb{R}$，$\gamma_{k,j} = \text{softplus}(\mathbf{w}_{k,\gamma}^\top \mathbf{h}(t_j)) \in \mathbb{R}$。这表明 Hawkes 过程的基础强度为 $\mu_{k,j}$，随着 $t > t_j$ 的增加，强度会以指数速度向 $\eta_{k,j}$ 衰减，衰减速率由 $\gamma_{k,j}$ 控制。值得注意是，强度的变化速度 $(\eta_{k,j} - \mu_{k,j})$ 可正可负。当 $(\eta_{k,j} - \mu_{k,j}) > 0$ 时，可以认为是自激励 (exciting) 的过程，当 $(\eta_{k,j} - \mu_{k,j}) < 0$ 时，可以认为是抑制 (inhibiting) 的过程。

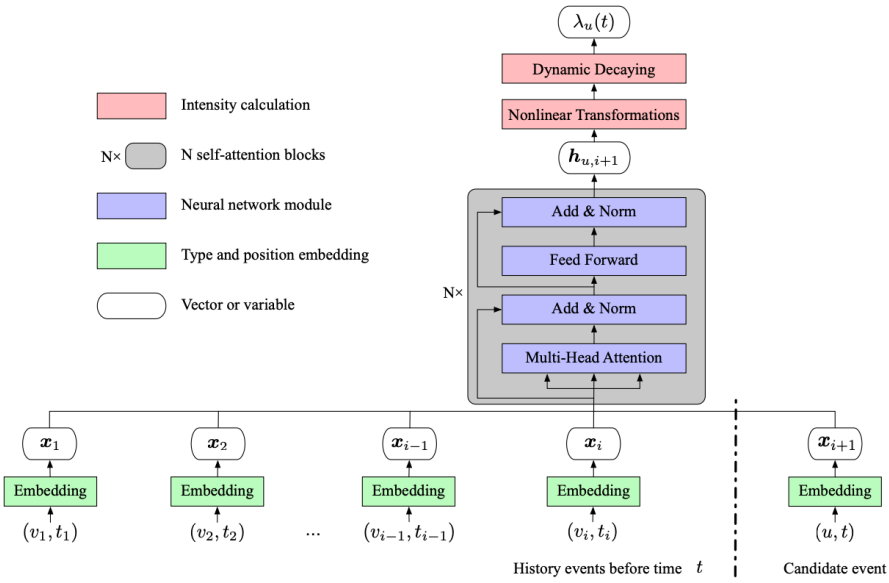


图 7: SAHP (Self-Attentive Hawkes Process), 图中记号与上述记号略有不同

4.6 IFTPP (ICLR 2020)

IFTPP 提出了一种完全**不需要依赖 CIF 建模点过程的范式**。IFTPP 的核心思想是，与其通过 CIF $\lambda(t|\mathcal{H}_{t-}$ 计算出下一个时间戳的 CPDF$f(t|\mathcal{H}_{t-})$，不如绕开 CIF 直接建模 CPDF。

1. **历史信息的输出 $\mathbf{h}(t_j)$:** IFTPP 使用 RNN 聚合历史信息：根据历史信息 $(t_1, k_1), \cdots, t_j, k_j$，先做一阶差分算出 $(\tau_1, k_1), \cdots, (\tau_j, k_j)$，然后用 RNN 聚合历史信息得到 $\mathbf{h}(t_j)$。

2. **建模 CPDF:** IFTPP 建模**事件发生时间间隔 τ 的 CPDF**。由于高斯混合模型 (GMM) 在低维密度估计方面表现良好，加上时间间隔 $\tau > 0$，因此使用**对数高斯混合模型** (Mixture of Log-normal) 来建模 $p(\tau|\mathcal{H}_{t_j})$ 是非常理想的。具体而言，$p(\tau|\mathcal{H}_{t_j})$ 可以写为

$$p(\tau|\mathcal{H}_{t_j}) = p(\tau|\mathbf{h}(t_j)) = \sum_{k=1}^{K} w_k \cdot \frac{1}{\tau s_k \sqrt{2\pi}} \exp\left(-\frac{(\log\tau - \mu_k)^2}{2s_k^2}\right)$$

其中

$$\begin{aligned}
\mathbf{w} &= [w_1, \cdots, w_K] = \text{softmax}(\mathbf{V}_w \mathbf{h}(t_j) + \mathbf{b}_w) \in \mathbb{R}^K \\
\boldsymbol{\mu} &= [\mu_1, \cdots, \mu_K] = (\mathbf{V}_\mu \mathbf{h}(t_j) + \mathbf{b}_\mu) \in \mathbb{R}^K \\
\mathbf{s} &= [s_1, \cdots, s_K] = \exp(\mathbf{V}_s \mathbf{h}(t_j) + \mathbf{b}_s) \in \mathbb{R}^K
\end{aligned}$$

这么做的好处在于 $p(\tau|\mathcal{H}_{t_j})$ 的期望是有闭式解的

$$\mathbb{E}_p[\tau] = \sum_{k=1}^{K} w_k \exp(\mu_k + \frac{s_k^2}{2})$$

从而在采样的时候，很容易就能采出下一个事件的时间戳 $t_{j+1} = t_j + \mathbb{E}_p[\tau]$。

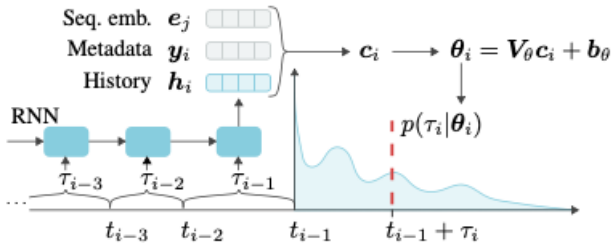


图 8: IFTPP (Intensity-Free Temporal Point Process), 图中记号与上述记号略有不同

4.7 WSM-TPP (NIPS 2024)