Machine Learning for Graphs and Sequential Data Exercise Sheet 03

Temporal Point Processes

Problem 1: Consider a temporal point process, where all the inter-event times $\tau_i = t_i - t_{i-1}$ are sampled i.i.d. from the distribution with the survival function

$$S(\tau) = \exp\Big(-(e^{b\tau} - 1)\Big)$$

with a parameter $b > 0$.

a) Write down the closed-form expression for the conditional intensity function $\lambda^*(t)$ of this TPP. Simplify as far as you can.

b) Write down the closed-form expression for the log-likelihood of a sequence $\{t_1, ..., t_N\}$ generated from this TPP on the interval $[0, T]$. Simplify as far as you can.

Problem 2: Consider an inhomogeneous Poisson process (IPP) on $[0, 1]$ with the intensity function $\lambda(t) = 2t$. We simulate a sample from this IPP using thinning. For this, we first simulate a *homogeneous* Poisson process (HPP) with intensity $\mu = 4$ and apply the thinning procedure described in the lecture. What is the expected number of events from the HPP that will be rejected when using this procedure?

Problem 3: Consider an inhomogeneous Poisson process on $[0, 4]$ with the intensity function $\lambda(t) = \beta t$, where $\beta > 0$ is a parameter that has to be estimated. You have observed a single sequence $\{1, 2.1, 3.3, 3.8\}$ generated from this IPP. What is the maximum likelihood estimate of the parameter β?

Problem 4: Consider a *neural* temporal point process where the conditional intensity function is defined with a neural network. In particular, for a time point t_i, we represent the history $\{t_1, t_2, \ldots, t_{i-1}\}$ with a fixed-sized vector $\boldsymbol{h}_i \in \mathbb{R}^d$. The conditional intensity function $\lambda^*(t)$ is defined as a function of $\boldsymbol{h}_i$. We will use the transformer architecture (see previous lecture). We propose the following implementation.

Given the full sequence $\{t_1, t_2, \ldots, t_n\}$, we calculate all $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$ in parallel. We first calculate vectors $\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i \in \mathbb{R}^d$ as a function of t_i. We stack these vectors into matrices $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n \times d}$. The output of the transformer is: $\boldsymbol{H} = \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}$, then $\boldsymbol{h}_i$ is the ith row of $\boldsymbol{H}$.

Identify the errors in this implementation compared to the original definition of $\boldsymbol{h}_i$. Propose a solution.