Fully Neural Network based Model for General Temporal Point Processes

Takahiro Omi[1,3], Naonori Ueda[2,3], Kazuyuki Aihara[1]
[1] The University of Tokyo, [2] NTT Communication Science Laboratories, [3] RIKEN AIP

Temporal Point Process

✓ **A temporal point process** is a mathematical model of **temporally discrete events** such as earthquakes, financial transactions, communication in a social network, user activity at a web site, and so on.

✓ **Conditional Intensity function** characterizes the temporal point process:

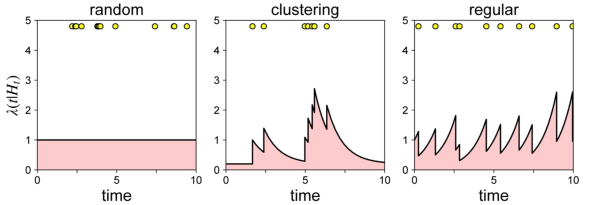
$$\lambda(t|H_t) = \lim_{\Delta\to 0}\frac{P(\text{one event occurs in } [t, t+\Delta)|H_t)}{\Delta}.$$

✓ **Parametric models of the conditional intensity function:**

Poisson process: $\lambda(t|H_t) = \mu$

Hawkes process: $\lambda(t|H_t) = \mu + \sum_{t_i<t} g(t-t_i)$

Self-correcting process: $\lambda(t|H_t) = \exp\left[\mu t - \sum_{t_i<t}\alpha\right]$

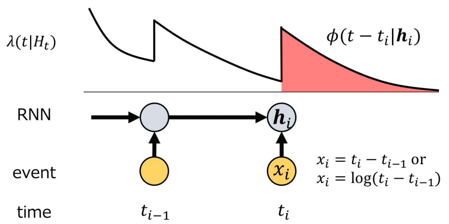


Background: RNN based Approach to Point Processes

✓ The RNN based model aims at flexibly modeling the dependence of the event occurrence on the event history.

✓ The RNN is used to encode the event history. Then the conditional intensity function is formulated via **the hazard function** as

$$\lambda(t|H_t) = \phi(t - t_i|\boldsymbol{h}_i).$$



✓ The parametric model of the hazard function $\phi(t - t_i|\boldsymbol{h}_i)$:

$$\phi(\tau|\boldsymbol{h}_i) = \exp(w\tau + \boldsymbol{v}\cdot\boldsymbol{h}_i + b) \quad \text{(Exponential hazard function [2])}$$

$$\phi(\tau|\boldsymbol{h}_i) = \exp(\boldsymbol{v}\cdot\boldsymbol{h}_i + b) \quad \text{(Constant hazard function [3])}$$

Problem

✓ The hazard function is usually modeled by a specific parametric function. However such an assumption can limit the expressive power of the model.

✓ If we use a complex model for the hazard function, the log-likelihood function cannot be exactly evaluated😢. This is because the loglikelihood function of the model,

$$\log L(\{t_i\}) = \sum_i \left\{\log \phi(t_i - t_{i-1}|\boldsymbol{h}_i) - \int_0^{t_i - t_{i-1}} \phi(\tau|\boldsymbol{h}_i)\, d\tau\right\},$$

includes the integral of the hazard function.

Method

Our novel approach

✓ Instead of the hazard function, we model **the cumulative hazard function**,

$$\Phi(\tau|\boldsymbol{h}_i) = \int_0^{\tau}\phi(s|\boldsymbol{h}_i)ds.$$

✓ The hazard function is given by differentiating the cumulative hazard function,

$$\phi(\tau|\boldsymbol{h}_i) = \frac{\partial}{\partial\tau}\Phi(\tau|\boldsymbol{h}_i).$$

✓ The log-likelihood function of the model is reformulated as

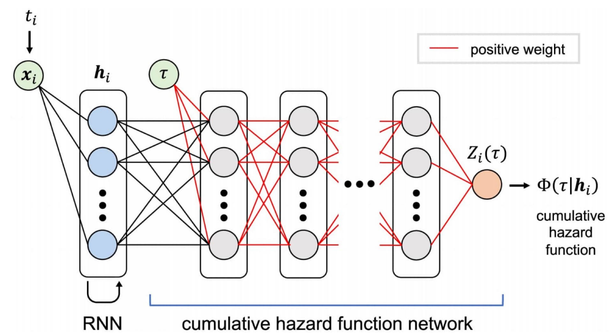
$$\log L(\{t_i\}) = \sum_i\left[\log\left\{\frac{\partial}{\partial\tau}\Phi(\tau = t_i - t_{i-1}|\boldsymbol{h}_i)\right\} - \Phi(t_i - t_{i-1}|\boldsymbol{h}_i)\right]$$

😀This can be exactly evaluated even for a complex model of the cumulative hazard function!!

Proposed model

✓ **The feedforward neural network model of the cumulative hazard function**

The cumulative hazard function is a monotonically increasing function of τ, which can be reproduced by constraining the particular network connections to be positive [1].



Our contribution

✓ (Flexibility) The hazard function can be flexibly modeled.

✓ (Efficiency) The log-likelihood function can be exactly evaluated without any numerical approximations, so that the model can be efficiently trained.

Related works

	Flexibility	Closed-form likelihood
Exponential hazard function [2]		✓
Constant hazard function [3]		✓
Piecewise constant hazard function [4]	?	✓
(Proposed) Neural cumulative hazard function	✓	✓
Continuous-time LSTM model [5]	✓	

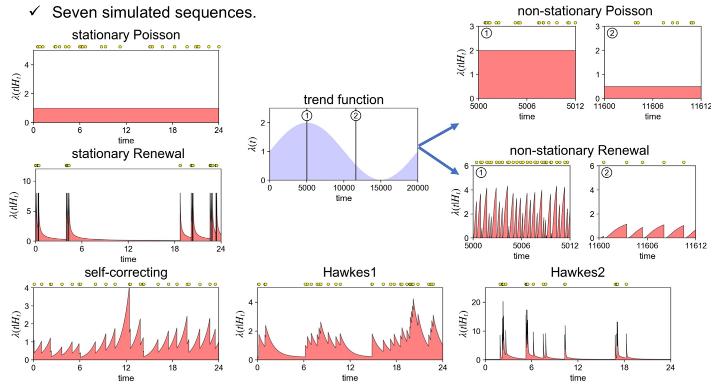
* The continuous-time model employs a quite different network architecture than the other RNN based models.

Experiments

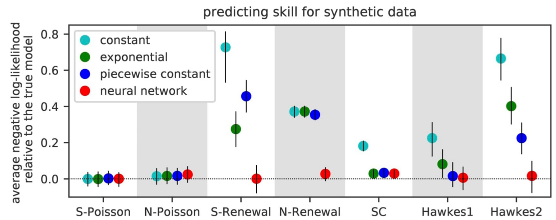
Synthetic datasets

* The mean inter-event interval is 1 for all the sequences.

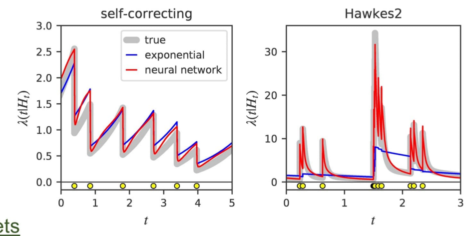
✓ Seven simulated sequences.



✓ The proposed model is better or similar than the other models.

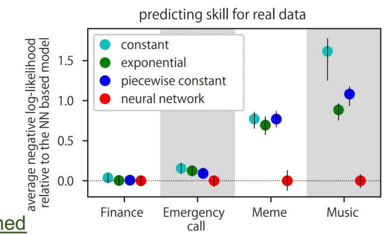


✓ The constant or exponential hazard function is sensitive to model misspecification.



Real datasets

✓ Our model also performs better than the other models for the real datasets.



We also confirmed

✓ Our model also outperforms the other models for a timing prediction task where the predictive performance is evaluated by the mean absolute error.

✓ Our model outperforms the state-of-the-art continuous-time LSTM model (see the paper for the details).

References: [1] J. Sill, Monotonic Networks, NIPS 1998. [2] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. KDD 2016. [3] H. Huang, H. Wang, and Br. Mak. Recurrent poisson process unit for speech recognition. AAAI 2019. [4] H. Jing and A. J. Smola. Neural survival recommender. WSDM 2017. [5] H. Mei and J. M. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. NIPS 2017.