Fully Neural Network based Model for General Temporal Point Processes

Takahiro Omi\(^1\), Naonori Ueda\(^2\), 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:
  \[
  \lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{P[\text{one event occurs in } [t, t + \Delta t)|H_t]}{\Delta t}
  \]
- Parametric models of the conditional intensity function:
  - Poisson process
    \[
    \lambda(t|H_t) = \lambda
    \]
  - Hawkes process
    \[
    \lambda(t|H_t) = \lambda + \sum_{s < t} \psi(t-s)
    \]
  - Self-correcting process
    \[
    \lambda(t|H_t) = \exp[\alpha - \sum_{s < t} \psi(t-s)]
    \]

**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
  \[
  \lambda(t|H_t) = \phi(t - t_c|H_t).
  \]

**Method**

Our novel approach
- Instead of the hazard function, we model the cumulative hazard function,
  \[
  \Phi(t|H_t) = \int_0^t \phi(s|H_s)ds.
  \]
- The hazard function is given by differentiating the cumulative hazard function,
  \[
  \phi(t|H_t) = \frac{d}{dt} \Phi(t|H_t).
  \]
- The log-likelihood function of the model is reformulated as
  \[
  \log L(t|\theta) = \sum_{i=1}^n \left[ \log \left( \frac{\phi(t_i - t_{i-1}|H_{t_{i-1}})}{\phi(t_i - t_{i-1}|H_{t_{i-1}})} \right) - \Phi(t_i - t_{i-1}|H_{t_{i-1}}) \right]
  \]

Proposed model
- **The feedforward neural network model of the cumulative hazard function**
  The cumulative hazard function is a monotonically increasing function of \(t\), 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**

<table>
<thead>
<tr>
<th></th>
<th>Flexibility</th>
<th>Closed-form likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential hazard function [2]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Constant hazard function [3]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Piecewise constant hazard function [4]</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>(Proposed) Neural cumulative hazard function</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Continuous-time LSTM model [5]</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

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

References:

**Experiments**

Synthetic datasets
- Seven simulated sequences, stationary Poisson, non-stationary Poisson
- Trend function
- Self-correcting
- Hawkes1
- Hawkes2

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).