Learning Temporal Point Processes via Reinforcement Learning

Shuang Li¹, Shuai Xiao², Shixiang Zhu¹, Nan Du³, Yao Xie¹, Le Song¹,²

¹Georgia Institute of Technology
²Ant Financial
³Google Brain
Motivation

- **Event data**: tweets/retweets, crime events, earthquakes, patient visits to hospital, finance transactions, ...

- Learn temporal pattern of event data.
  - Event time is random
  - Complex dependency structure
Point Process Model

- **Intensity function**

\[
\lambda(t|\mathcal{H}_t)dt = \mathbb{E}[N[t, t + dt]|\mathcal{H}_t]
\]

where \( N[t, t + dt] \) is the number of events falling in the set \([t, t + dt)\).

| Point Process          | \( \lambda_\theta(t|\mathcal{H}_t) \) | Temporal Pattern |
|------------------------|-------------------------------------|------------------|
| Poisson                | constant                            | ![Poisson Temporal Pattern] |
| Inhomogeneous Poisson  | \( \lambda_\theta(t) \)             | ![Inhomogeneous Poisson Temporal Pattern] |
| Hawkes                 | \( \mu + \alpha \sum_{t_i \in \mathcal{H}_t} \exp(-|t - t_i|) \) | ![Hawkes Temporal Pattern] |
Traditional Maximum-Likelihood Framework

- Model conditional intensity \( \lambda_{\theta}(t|\mathcal{H}_t) \)
as a parametric/non-parametric form.

- Learn model by maximizing likelihood

\[
P(t_1, t_2, \ldots, t_n) = \exp \left( - \int_{(0,T)} \lambda_{\theta}(t|\mathcal{H}_t) \, dt \right) \prod_i \lambda_{\theta}(t_i|\mathcal{H}_t)
\]
New Reinforcement Learning Framework

- **Learn policy**
  \[
  \pi_\theta(a|s_t) = p(t_i \mid t_{i-1}, \ldots, t_1)
  \]
  where \(a \in \mathbb{R}^+\) is the next event time, to maximize cumulative reward.
- **Learn reward**
  \[ r(a) \]
  to guide policy to imitate observed event data (expert).
Inverse Reinforcement Learning

\[ r^* = \max_{r \in \mathcal{F}} \left( \mathbb{E}_{\text{expert}} \left[ \sum_i r(t_i) \right] - \max_{\pi_\theta} \mathbb{E}_{\pi_\theta} \left[ \sum_i r(a_i) \right] \right) \]

Choose \( r \in \mathcal{F} \) be unit ball in Reproducing Kernel Hilbert Space (RKHS). We obtain \textit{analytical} optimal reward

Given \( L \) expert trajectories, \( M \) generated trajectories by policy

\[ \hat{r}^*(a) \propto \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1} \kappa \left( t_i^{(l)}, a \right) - \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1} \kappa \left( a_i^{(m)}, a \right) \]

where \( \kappa(t, t') \) is a universal RKHS kernel.
Modeling Framework

\( \pi_\theta (a|s_t) \)

Policy Gradient

expert

\[ r^*(a_1) r^*(a_2) r^*(a_3) r^*(a_4) \]

update optimal reward
Numerical Results

- **Our method:** RLPP
- **Baselines:**
  - State-of-the-art methods: RMTPP (Du et al. 2016 KDD), WGAN (Xiao et al. 2017 NIPS)
  - Parametric baselines: Inhomogeneous Poisson (IP), Hawkes (SE), Self-correcting (SC)
- **Comparison of learned empirical intensity**

![Graphs showing intensity comparison](image)

- **Comparison of runtime**

<table>
<thead>
<tr>
<th>Method</th>
<th>RLPP</th>
<th>WGAN</th>
<th>RMTPP</th>
<th>SE</th>
<th>SC</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>80 m</td>
<td>1560m</td>
<td>60m</td>
<td>2m</td>
<td>2m</td>
<td>2m</td>
</tr>
<tr>
<td>Ratio</td>
<td>40x</td>
<td>780x</td>
<td>30x</td>
<td>1x</td>
<td>1x</td>
<td>1x</td>
</tr>
</tbody>
</table>
Poster

- Tue Dec 4th 05:00 -- 07:00 PM
- @ Room 210 & 230 AB #124