Reinforcement Learning of Marked Temporal Point Processes

HUMAN-CENTERED MACHINE LEARNING

http://courses.mpi-sws.org/hcml-ws18/
Reinforcement learning on different settings

Actions and feedback are real-valued functions in continuous time

Actions and feedback occur in discrete time

Actions and feedback are asynchronous events localized in continuous time
Reinforcement learning of marked TPP

If the problem dynamics cannot be expressed using SDEs with jumps or the objective is intractable:

Reinforcement learning of marked temporal point processes

→ Policy gradient [Upadhyay, 2018]

→ Policy iteration [Farajtabar et al., 2017]

Similarly as with optimal control:
Policy is characterized by an intensity function!
If the problem dynamics cannot be expressed using SDEs with jumps or the objective is intractable:

Next, details on the approach based on policy gradient

Similarly as with optimal control:
Policy is characterized by an intensity function!
Viral marketing

Agent

Environment

When to post to maximize views or likes?

$$\mu_i(t) = u(t) \rightarrow N_i(t)$$

Design (optimal) posting intensity

Marks (feedback) given by environment

Forbes

For Brands And PR: When Is The Best Time To Post On Social Media?

The Best Times to Post on Social Media
Visibility dynamics are unknown

However, one may have access to quality metrics

Key idea:
Think of these metrics as rewards in a reinforcement learning setting!

[Upadhyay et al., 2018]
Broadcasters and feedback

We do not know the feedback distribution but we can sample from it.

\[ p_{\mathcal{A};\theta}^* = (\lambda_{\theta}^*, m_{\theta}^*) \]

\[ p_{\mathcal{F};\phi}^* = (\lambda_{\phi}^*, m_{\phi}^*) \]

We do not know the feedback distribution but we can sample from it...

...and measure quality metrics (rewards)

[Upadhyay et al., 2018]
What is the goal in reinforcement learning?

We aim to maximize the average reward in a time window $[0, T]$:

$$J(\theta) \quad \mathbb{E}_{A_T \sim p_A^*; \theta(\cdot), F_T \sim p_F^*; \phi(\cdot)}[R^*(T)]$$

Actions and environment are asynchronous!

Reward (Cumulative)

Connection to optimal control:

$$J(r(t), \lambda(t), t) = \min_{u(t,t_f)} \mathbb{E}_{(N,M)(t,t_f)} \left[ \phi(r(t_f)) + \int_t^{t_f} \ell(r(\tau), u(\tau)) d\tau \right]$$

[Upadhyay et al., 2018]
We use gradient descent to improve the policy, i.e., the intensity, over time:

$$\theta_{l+1} = \theta_l + \alpha_l \nabla_\theta J(\theta) \bigg|_{\theta = \theta_l}$$

We need to compute the gradient of an average. But the average depends on the parameters!

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{A_T \sim p^*_A; \theta(\cdot), F_T \sim p^*_F; \phi(\cdot)} [R^*(T)]$$

[Upadhyay et al., 2018]
The reinforce trick allows us to overcome this implicit dependence:

\[
\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{A_T \sim p_A^*; \theta(\cdot), \mathcal{F}_T \sim p_\mathcal{F}^*; \phi(\cdot)} \left[ R^*(T) \right]
\]

Appendix A in Upadhyay et al., 2018

\[
\nabla_\theta J(\theta) = \mathbb{E}_{A_T \sim p_A^*; \theta(\cdot), \mathcal{F}_T \sim p_\mathcal{F}^*; \phi(\cdot)} \left[ R^*(T) \nabla_\theta \log \mathbb{P}_\theta(\mathcal{A}_T) \right]
\]
Likelihood of action events

$$\nabla_\theta J(\theta) = \mathbb{E}_{A_T \sim p_{A;\theta}(\cdot), F_T \sim p_{F;\phi}(\cdot)} \left[ R^*(T) \nabla_\theta \log \mathbb{P}_\theta (A_T) \right]$$

\[ \mathbb{P}(A_T) := \left( \prod_{e_i \in A_T} \lambda^*_\theta(t_i) \right) \exp \left( - \int_0^T \lambda^*_\theta(s) \, ds \right) \]

The key remaining question is how to parametrize the intensity $\lambda^*_\theta(t)$

[Upadhyay et al., 2018]
Policy parametrization

Output layer:
\[
\lambda^*_\theta(t) = \exp(b_\lambda + w_t(t - t') + V_\lambda h_i)
\]

Hidden layer:
\[
h_i = \tanh(W_h h_{i-1} + W_1 \tau_i + W_4 b_i + b_h)
\]

Input layer:
\[
\tau_i = W_t(t_i - t_{i-1}) + b_t
\]
\[
b_i = W_a(1 - e_i) + W_f e_i + b_b
\]

Last time the broadcaster posted

Parameters

[i-th event] (i-1)-th event

[Upadhyay et al., 2018]
Sampling from the policy

\[ \lambda_{\theta}^*(t) = \exp(b_\lambda + w_t(t - t') + V_\lambda h_i) \]

The intensity can increase or decrease every time an event by the other broadcasters take place:

→ We cannot apply just superposition

→ We can use inversion sampling: The CDF is a function by parts, where each part is defined once an event by the other broadcasters happens

Appendix C in Upadhyay et al., 2018

[Upadhyay et al., 2018]
Average Rank

\[ R^*(T) = \int_0^T r(t) dt \]

- **TPPRL**
- **RQ\(^*\)**
- **RQ**
- **Karimi**

**Relative decrease**

- **RL**: It *learns* the feed ranking
- **True visibility**
- **RedQueen** (feed in reverse chronological)

Lower is better

[Upadhyay et al., 2018]
Time at the top

\[ R^*(T) = \int_0^T \mathbb{I}(r(t) < 1) dt \]

Higher is better

Relative increase

<table>
<thead>
<tr>
<th>Method</th>
<th>Relative Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPPRL</td>
<td>1.15</td>
</tr>
<tr>
<td>RQ(^*)</td>
<td>1.10</td>
</tr>
<tr>
<td>RQ</td>
<td>1.05</td>
</tr>
<tr>
<td>Karimi</td>
<td>1.00</td>
</tr>
</tbody>
</table>

It learns the feed ranking

RedQueen (true visibility)

RedQueen (feed in reverse chronological)

Offline algorithm

[Upadhyay et al., 2018]