Deep Reinforcement Learning of Marked Temporal Point Processes

Max Planck Institute for Software Systems – Networks and Machine Learning

Utkarsh Upadhyay, Abir De, Manuel Gomez-Rodriguez

{utkarshu,ade,manuelgr}@mpi-sws.org; Networks-Learning/TPPRL

Introduction

Humans *take actions* and *receive feedback* using
- asynchronous and stochastic
- discrete events in continuous time

to achieve *their goals*.

Events as Marked Temporal Point Processes

$t_i \in \mathbb{R}$: Times $\quad z_i \in \mathbb{Z}$: Marks

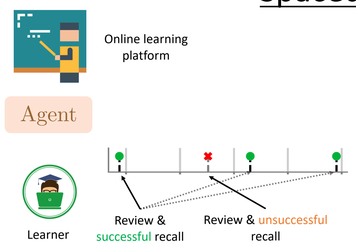
$\mathcal{H}_t = \{e_0 = (t_0, z_0), \ldots, e_n = (t_n, z_n)\}$

$\mathcal{H}_t$: History $\quad \mathcal{A}_t$: Actions $\quad \mathcal{F}_t$: Feedback

Given the above representation based on MTPPs, can we design online interventions to help them achieve their *goals*?

Example Applications

Spaced Repetition

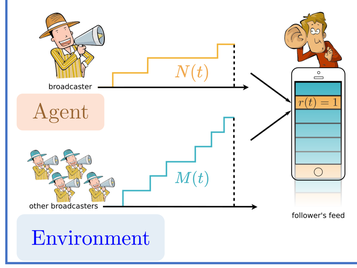


When to review to maximize recall?

t_i Goal

Reward:
- To *maximize* final test score

Viral Marketing



When to post to maximize views?

t_i Goal

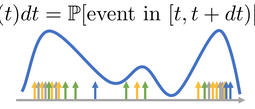
Reward:
- To *minimize* average rank
- To *maximize* time at the top

MTPP based Policy and Control

Characterization of MTPPs

Intensity Function

$\lambda^*(t)dt = \mathbb{P}[\text{event in } [t, t+dt) | \mathcal{H}_t]$



Mark Distribution

$z_i \sim m(z|\mathcal{H}_t) = m^*(z)$

MTPP based Policy: $p^*_{\mathcal{A};\theta}(\bullet) = (\lambda^*_\theta(\bullet), m^*_\theta(\bullet))$

Previous MTPP Control Methods

Chronological sorted feeds ✓
Algorithmic feeds ✗

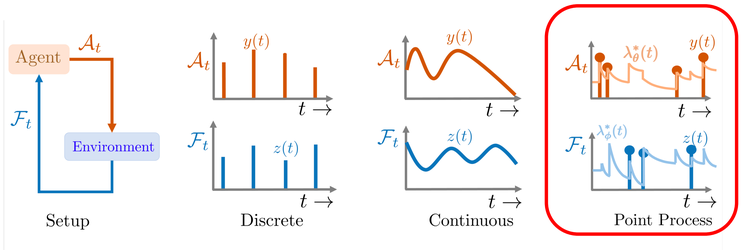
- Rely on known dynamics of the environment

Improve continuous retention ✓
Improve test scores ✗

- Carefully choose reward functions (*i.e.,* goals)
- Cannot leverage advances in deep learning

Reinforcement Learning of MTPPs

Novel RL Setting



TPPRL: Policy Gradient for MTPPs

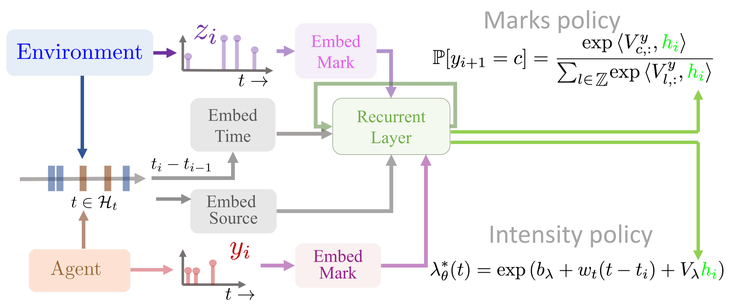
Aim: for *any* reward $R^*(T)$ $\quad \underset{p^*_{\mathcal{A};\theta}(\cdot)}{\text{maximize}} \quad \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} [R^*(T)]$

Solution: Develop REINFORCE trick for *intensities.*

- Can handle arbitrary reward functions!
- No model for the Feedback process needed!

Handling Asynchronicity

Embedding Asynchronous History



Taking Actions Asynchronously

Next event time is sampled from intensity $\lambda^*_\theta(t)$, but:

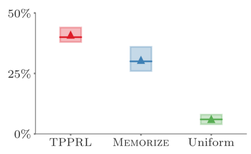
More feedback may arrive in the meanwhile

Solution: Efficient & unbiased resampling algorithm.

Evaluation

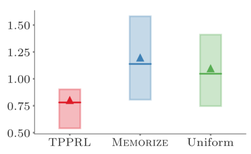
Spaced Repetition

- Exponential memory model
- Model trained on data from



(a) Recall

Better test score

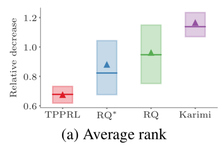


(b) Items' difficulty

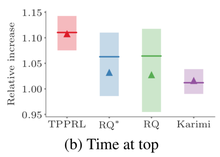
Learns to exploit easy items/lessons

Viral Marketing

- Realistic priority based feeds
- Data from 1000 tweeters



(a) Average rank



(b) Time at top

Can optimize for algorithmic feeds and different rewards