Learning with
Temporal Point Processes

RL & Control

Manuel Gomez Rodriguez
Max Planck Institute for Software Systems

ICML TUTORIAL, JULY 2018

Outline of the Seminar

TEMPORAL POINT PROCESSES (TPPS): INTRO

1. Intensity function
2. Basic building blocks
3. Superposition
4. Marks and SDEs with jumps

MODELS & INFERENCE

1. Modeling event sequences
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

RL & CONTROL

1. Marked TPPs: a new setting
2. Stochastic optimal control
3. Reinforcement learning

Next

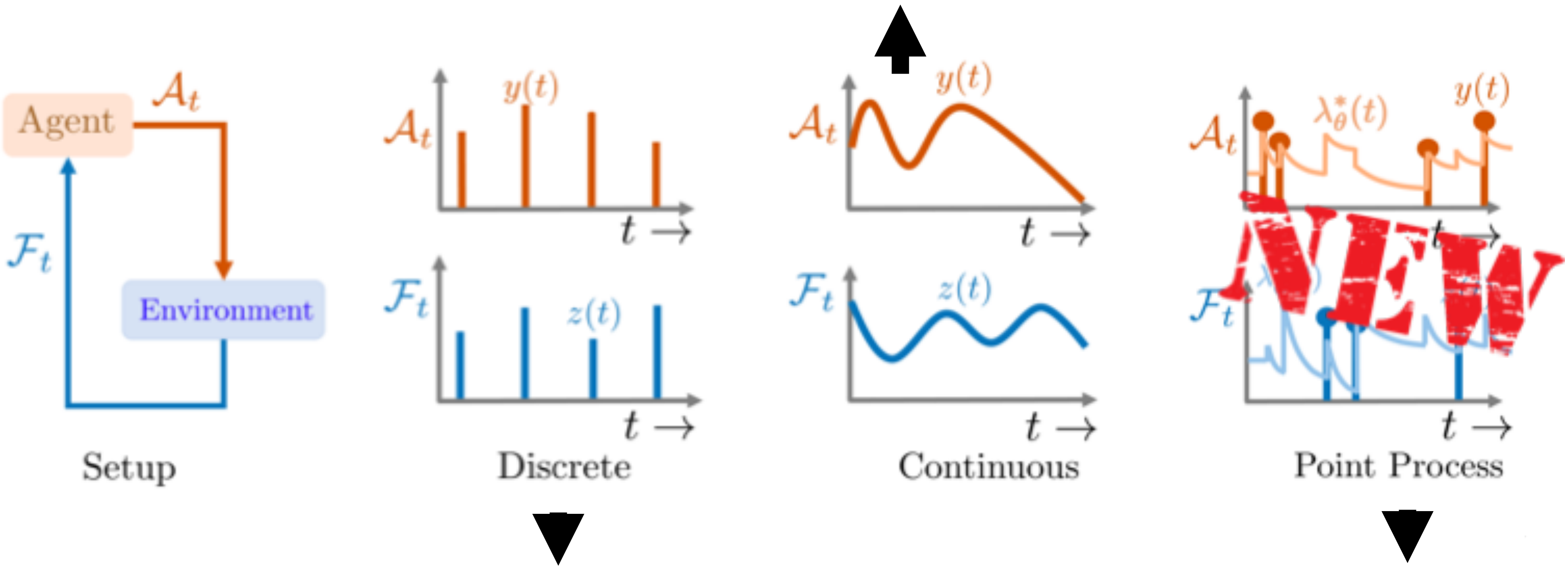
Slides/references: learning.mpi-sws.org/tpp-icml18

RL and Control

1. **Marked TPP: a new setting**
2. Stochastic optimal control
3. Reinforcement learning

MTPP: a new setting for control & RL

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

Example I: Viral marketing

Agent

Social media user

Environment

Followers' Feed

Forbes

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

THE HUFFINGTON POST

The Best Times to Post on Social Media

When to post to maximize views or likes?

$$\mu_i(t) = u(t) \;\blacktriangleright\; N_i(t)$$

Design (optimal) posting intensity

Marks (feedback) given by environment

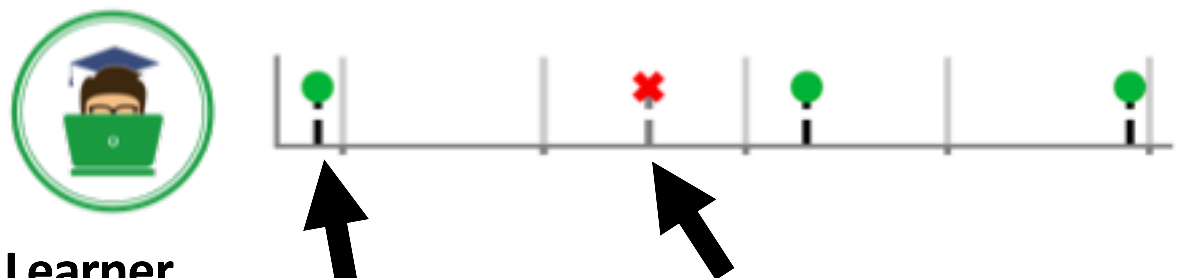
Example II: Spaced repetition

Agent

duolingo

Online learning platform

Environment



Learner

Review & successful recall

Review & unsuccessful recall

When to review to maximize recall probability?

$$\lambda_i(t) \rightarrow N_i(t)$$

Design (optimal) reviewing intensities

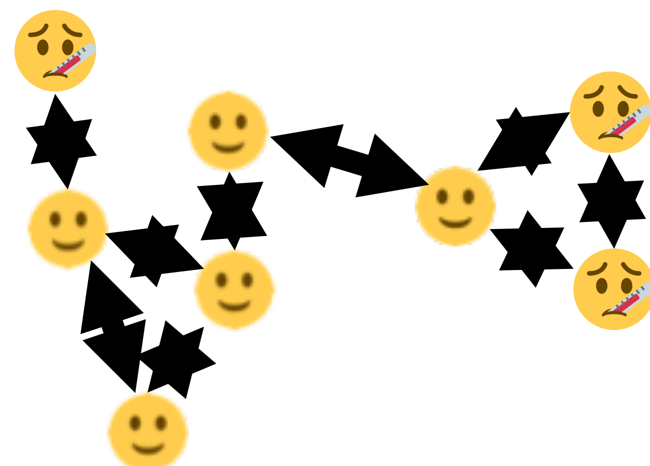
Marks

Example III: Suppressing epidemics

Agent

Health policy
(Resource allocation)

Environment



Population (social network)

Who to treat and when to reduce infections?

$\lambda_i(t) \rightarrow N_i(t)$

Design (optimal)
treatment intensities

Marks

RL & Control

1. Marked TPP: a new setting for control
2. **Stochastic optimal control**
3. Reinforcement learning

Stochastic optimal control of SDEs with jumps

If the problem dynamics can be expressed using SDEs with jumps:

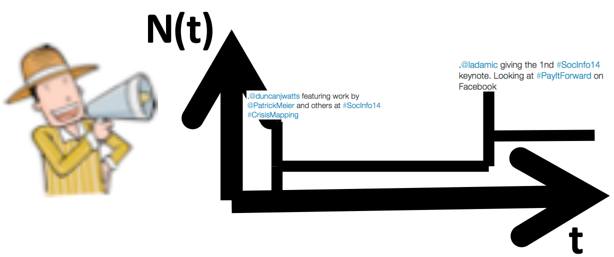
- **Op[...]al control of marked temporal po[...]**
  - [...] Kim et al. 2018;
  - [...]

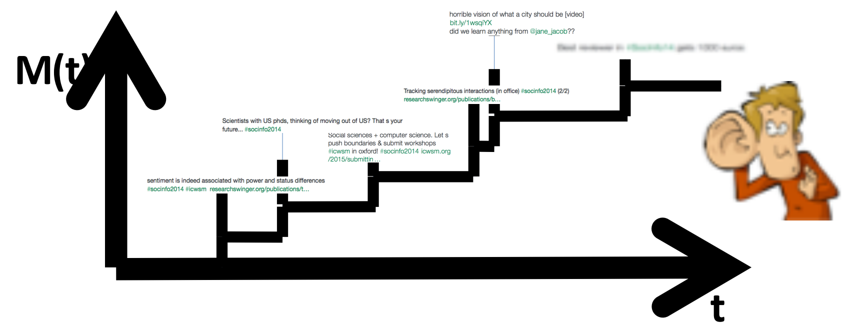
Next, details on one approach to the when to post problem

Key idea:

Policy is characterized by an intensity function!

Broadcasters and feeds





$$\mathbb{E}[d\boldsymbol{N}(t)|\mathcal{H}(t)] = \boldsymbol{\mu}(t)\, dt \quad \Rightarrow \quad \mathbb{E}[d\boldsymbol{M}(t)|\mathcal{H}(t)] = \boldsymbol{\gamma}(t)\, dt$$

Policy → **Broadcaster intensity function (tweets / hour)**

$$A^T \boldsymbol{\mu}(t)$$

Feed intensity function (tweets / hour)

Given a broadcaster i and her followers ⇒

$$\boldsymbol{M}_{\backslash i}(t) = A^T \boldsymbol{N}(t) - A_i N_i(t)$$

$$\gamma_{j\backslash i}(t) = \gamma_j(t) - \mu_i(t)$$

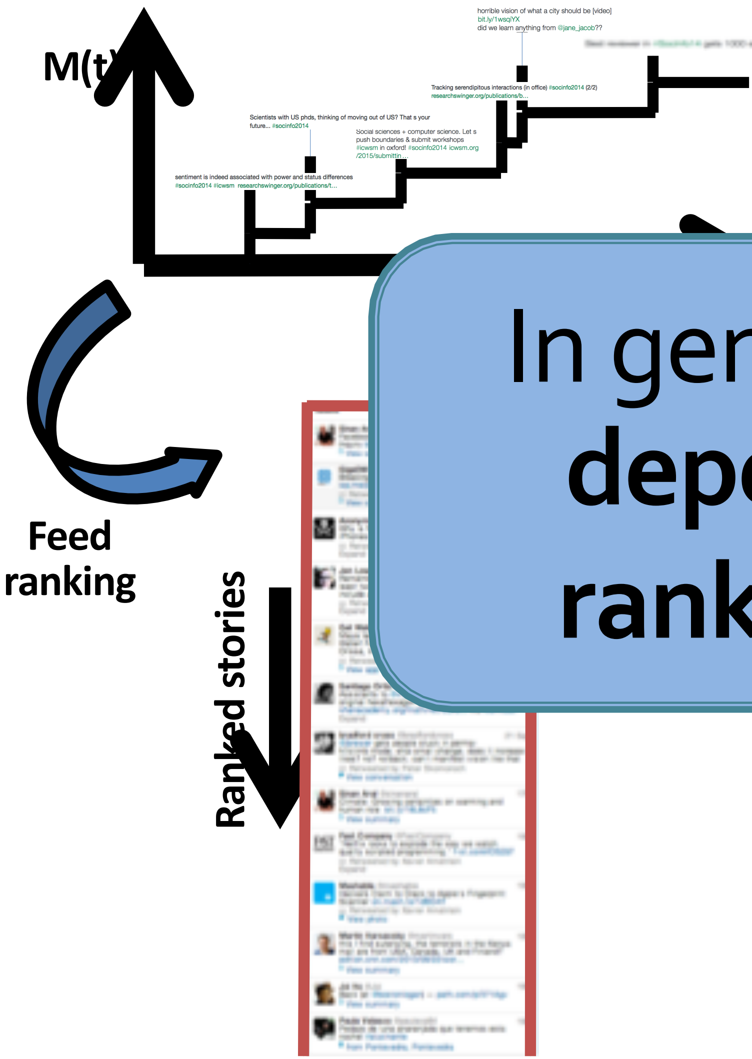
Feed due to other broadcasters

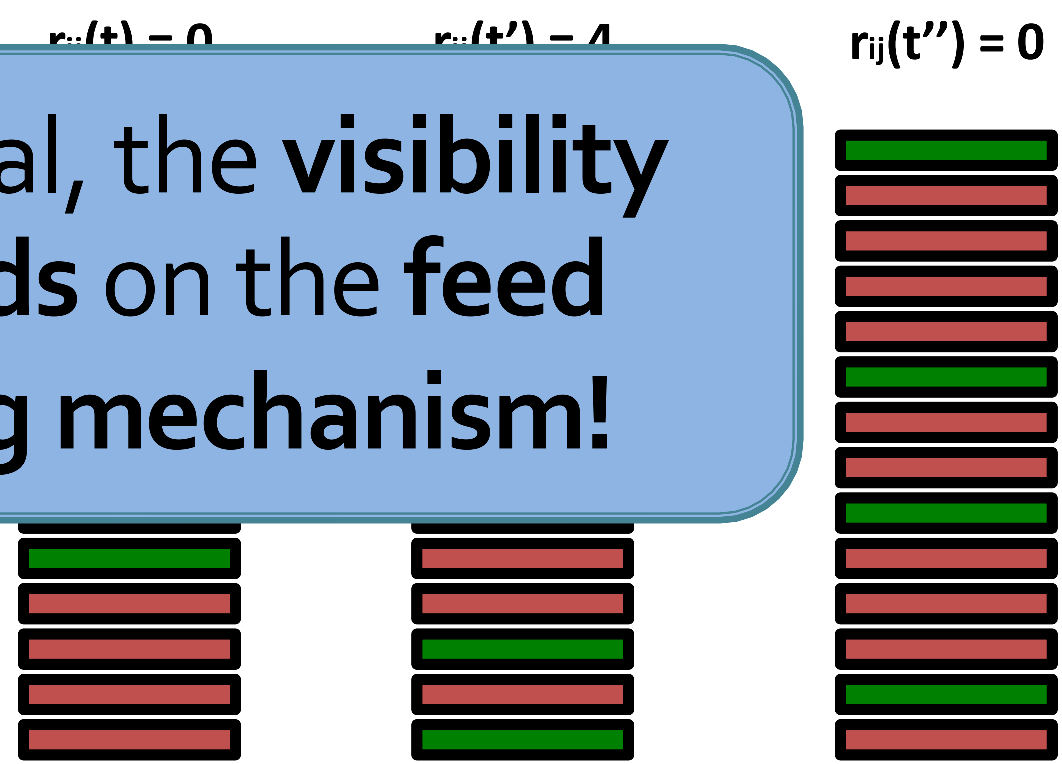
[Zarezade et al., 2017 & 2018]

Definition of visibility function

Visibility of broadcaster i at follower j

Position of the highest ranked tweet by broadcaster i in follower j's wall





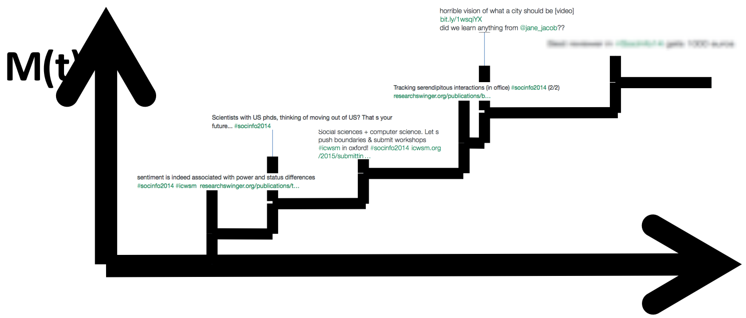
In general, the **visibility depends** on the **feed ranking mechanism!**

Post by broadcaster u

Post by other broadcasters

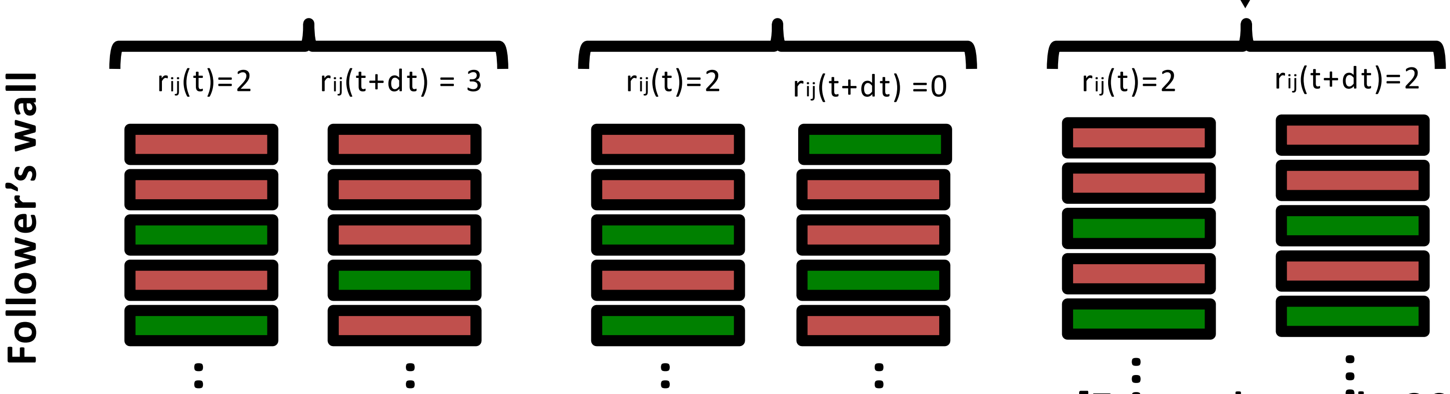
[Zarezade et al., 2017 & 2018]

Visibility dynamics in a FIFO feed (I)



Reverse chronological order

$$\underbrace{r_{ij}(t+dt)}_{\text{Rank at t+dt}} = \underbrace{(r_{ij}(t)+1)dM_{j\setminus i}(t)(1-dN_i(t))}_{\text{Other broadcasters post a story and broadcaster i does not post}} + \underbrace{0}_{\text{Broadcaster i posts a story and other broadcasters do not post}} + \underbrace{r_{ij}(t)(1-dM_{j\setminus i}(t))(1-dN_i(t))}_{\text{Nobody posts a story}}$$



[Zarezade et al., 2017 & 2018]

Visibility dynamics in a FIFO feed (II)

$$r_{ij}(t+dt) = \quad (r_{ij}(t)+1)dM_{j\backslash i}(t)(1-dN_i(t)) + \; 0 \; + \; r_{ij}(t)(1-dM_{j\backslash i}(t))(1-dN_i(t))$$

Zero-one law $dN_i(t)dM_{j\backslash i}(t) = 0$

$$dr_{ij}(t) = -r_{ij}(t)\, dN_i(t) + dM_{j\backslash i}(t)$$

$r_{ij}(t+dt) - r_{ij}(t)$

Broadcaster i posts a story

Other broadcasters posts a story

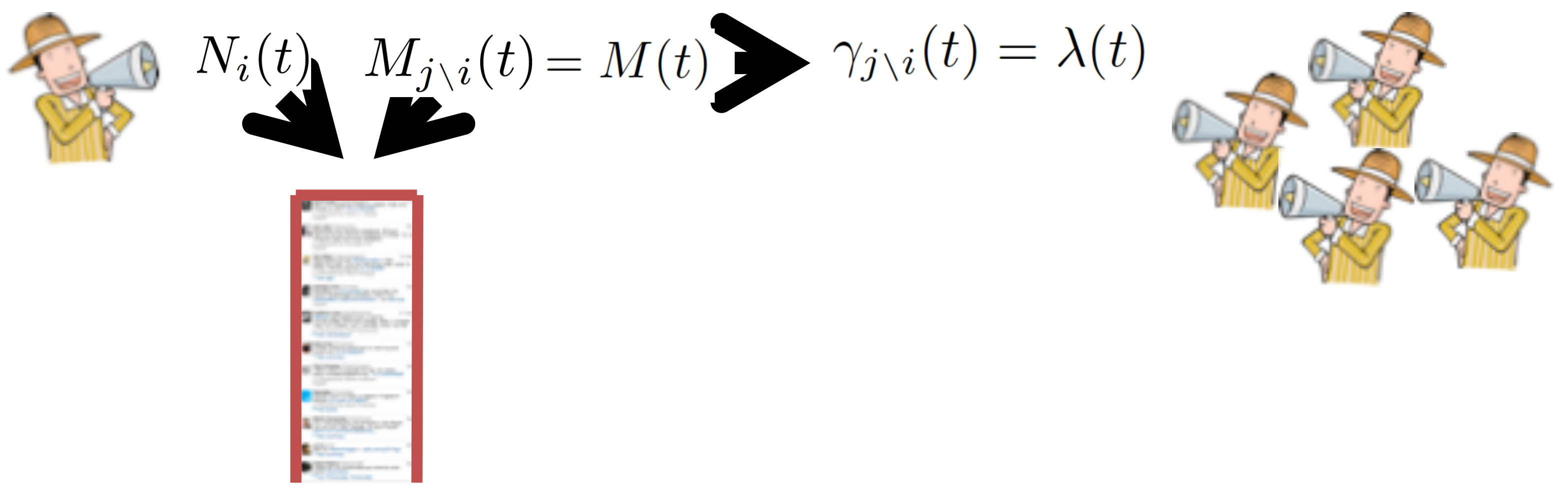
Stochastic differential equation (SDE) with jumps

OUR GOAL:

Optimize $r_{ij}(t)$ over time, so that it is small, by controlling $dN_i(t)$ through the intensity $\mu_i(t)$

[Zarezade et al., 2017 & 2018]

Feed dynamics



We consider a **general intensity:**

(e.g. Hawkes, inhomogeneous Poisson)

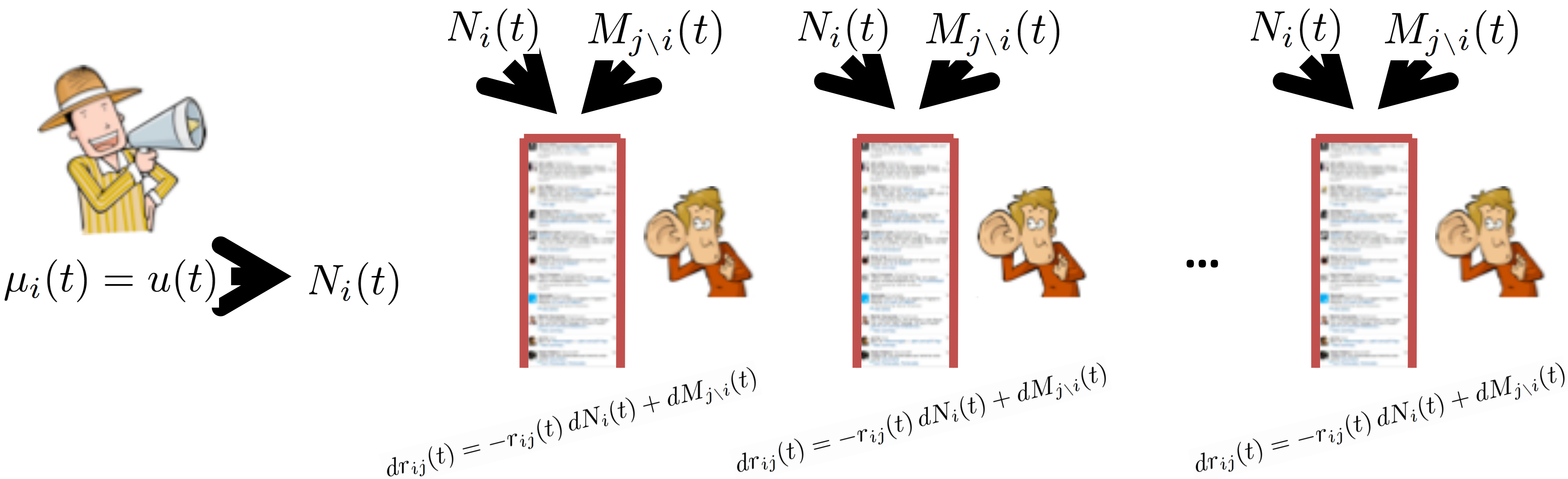
$$\lambda^*(t) = \underbrace{\lambda_0(t)}_{\substack{\textbf{Deterministic} \\ \textbf{arbitrary intensity}}} + \underbrace{\alpha \int_0^t g(t-s)dN(s)}_{\substack{\textbf{Stochastic} \\ \textbf{self-excitation}}}$$

Jump stochastic differential equation (SDE)

$$\left\{ d\lambda^*(t) = \left[\lambda_0'(t) + w\lambda_0(t) - w\lambda^*(t)\right] dt + \alpha dN_i(t) \right.$$

[Zarezade et al., 2017 & 2018]

The when-to-post problem



Optimization problem

$$\underset{u(t_0,t_f]}{\text{minimize}} \quad \mathbb{E}_{(N_i,\boldsymbol{M}_{\backslash i})(t_0,t_f]}\left[\overbrace{\phi(\boldsymbol{r}(t_f))}^{\text{Terminal penalty}} + \int_{t_0}^{t_f} \overbrace{\ell(\boldsymbol{r}(\tau),u(\tau))d\tau}^{\text{Nondecreasing loss}}\right]$$

$$\text{subject to} \quad u(t) \geq 0 \quad \forall t \in (t_0, t_f],$$

Dynamics defined by Jump SDEs

$$dr(t) = -r(t)\, dN(t) + dM(t)$$

$$d\lambda(t) = [\lambda_0'(t) + w\lambda_0(t) - w\lambda(t)]\, dt + \alpha\, dM(t)$$

[Zarezade et al., 2017 & 2018]

Bellman's Principle of Optimality

Lemma. The **optimal cost-to-go** satisfies **Bellman's Principle of Optimality**

$$J(r(t), \lambda(t), t) = \min_{u(t,t+dt]} \mathbb{E}\left[J(r(t+dt), \lambda(t+dt), t+dt)\right] + \ell(r(t), u(t))\, dt$$

$$J(r(t+dt), \lambda(t+dt), t+dt) = J(r(t), \lambda(t), t) + dJ(r(t), \lambda(t), t)$$

$$0 = \min_{u(t,t+dt]} \mathbb{E}\left[dJ(r(t), \lambda(t), t)\right] + \ell(r(t), u(t))\, dt$$

$$dr(t) = -r(t)\, dN(t) + dM(t)$$
$$d\lambda(t) = [\lambda_0'(t) + w\lambda_0(t) - w\lambda(t)]\, dt + \alpha\, dM(t)$$

Hamilton-Jacobi-Bellman (HJB) equation

Partial differential equation in J (with respect to r, λ and t)

[Zarezade et al., 2017 & 2018]

Solving the HJB equation

Consider a quadratic loss

$$\ell(r(t), u(t)) = \frac{1}{2}s(t)\, r^2(t) + \frac{1}{2}q\, u^2(t)$$

Favors some periods of times (e.g., times in which the follower is online)

Trade-offs visibility and number of broadcasted posts

Then, it can be shown that the optimal intensity is:

$$u^*(t) = q^{-1}\left[J(r(t), \lambda(t), t) - J(0, \lambda(t), t)\right]$$
$$= \sqrt{s(t)/q}\, r(t)$$

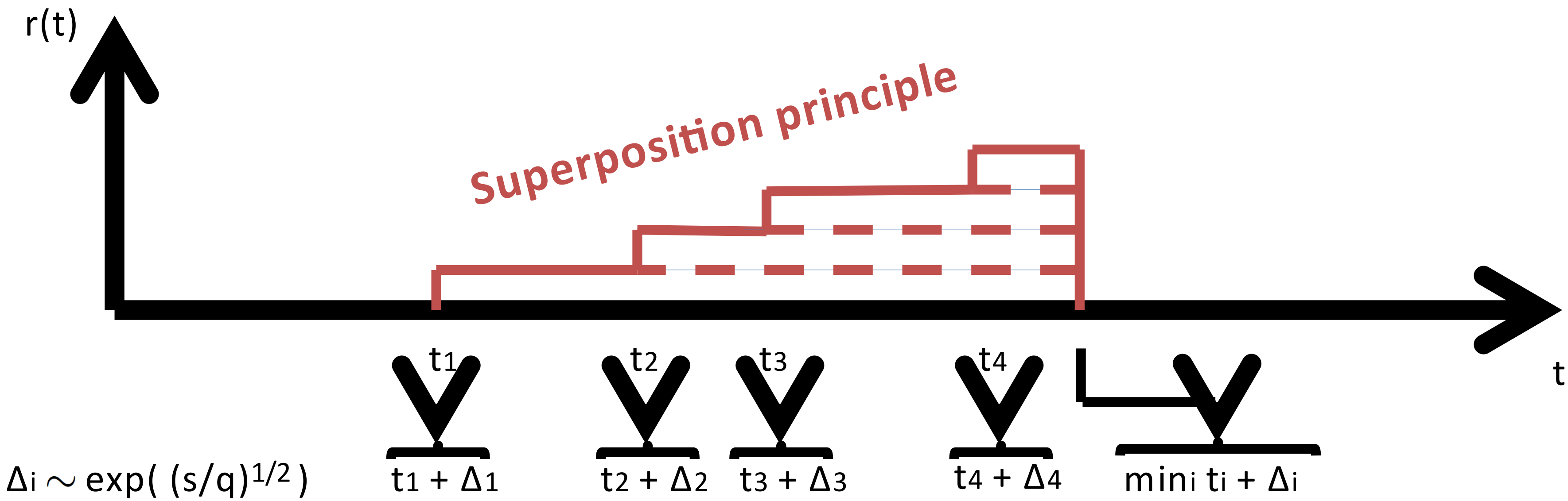
It only depends on the current visibility!

[Zarezade et al., 2017 & 2018]

The RedQueen algorithm

Consider $s(t) = s$ ➔ $u^*(t) = (s/q)^{1/2}\, r(t)$

How do we sample the next time?



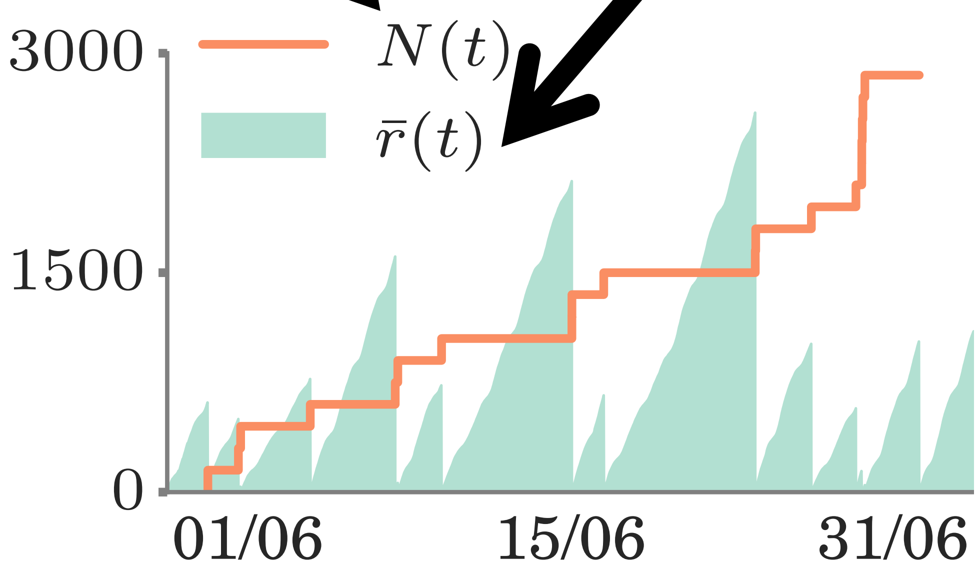
It only requires sampling $M(t_f)$ times!

[Zarezade et al., 2017 & 2018]

Example: a broadcaster in Twitter

Broadcaster's posts

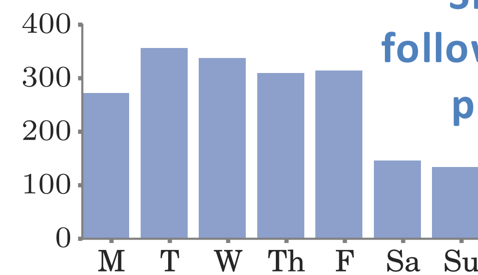
Average position over time



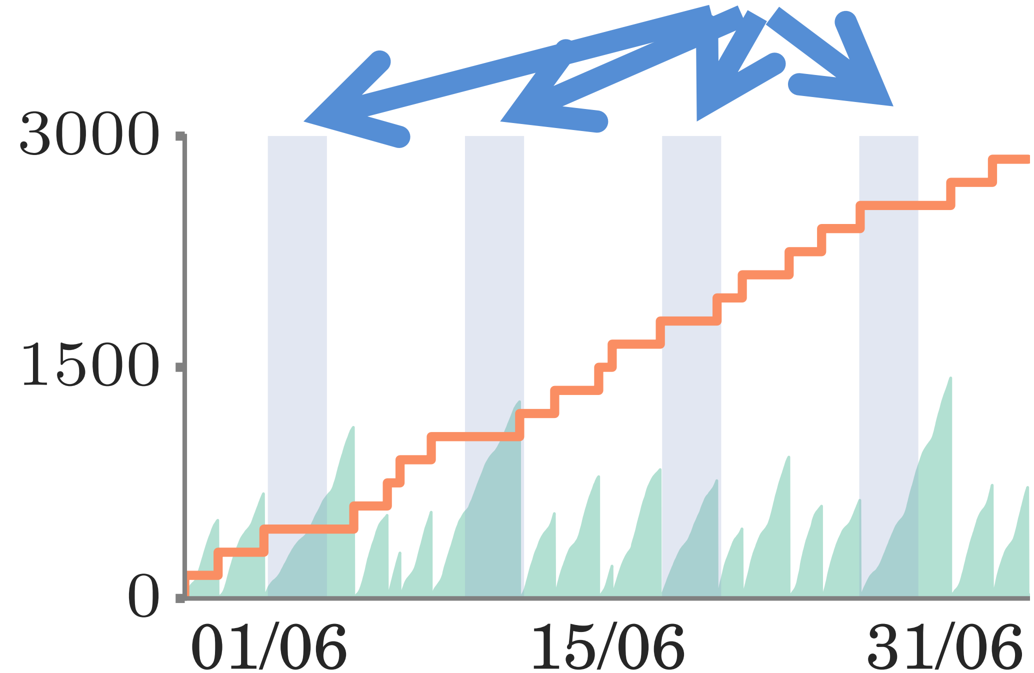
True posts

$$\frac{1}{T}\int_0^T \bar{r}(t)dt = 698.04$$

40% lower!



Significance: followers' retweets per weekday



REDQUEEN

$$\frac{1}{T}\int_0^T \bar{r}(t)dt = 425.25$$

[Zarezade et al., 2017 & 2018]

RL & Control

1. Marked TPP: a new setting
2. Stochastic optimal control
3. **Reinforcement learning**

Reinforcement learning of marked TPP

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

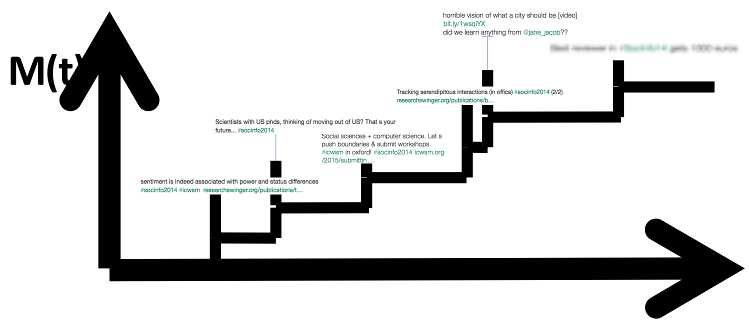
oral

Next, details on one approach to the when/what to post problem with algorithmic ranking

Similarly as with optimal control:

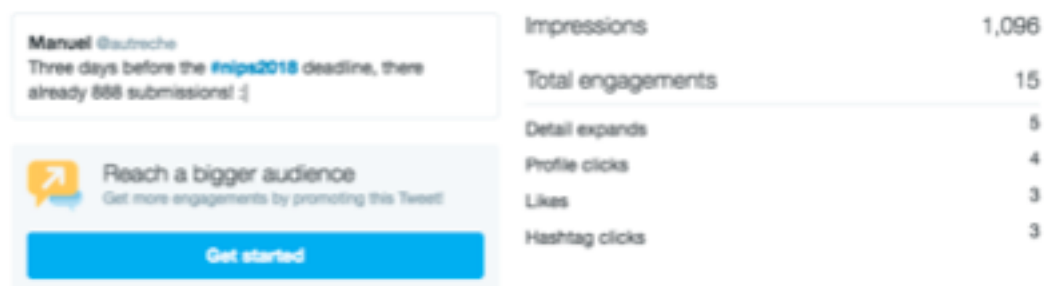
Policy is characterized by an intensity function!

Visibility dynamics are unknown



Proprietary algorithmic ranking

However, one may have access to quality metrics

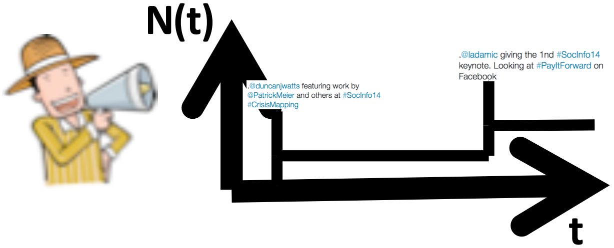


Key idea:

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

[Upadhyay et al., 2018]

Broadcasters and feedback



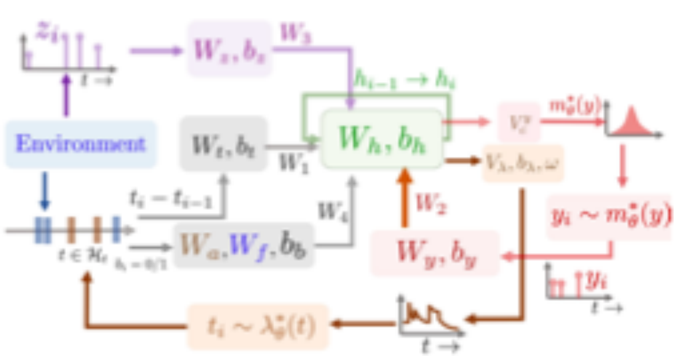
$$p^*_{\mathcal{A};\theta} = (\lambda^*_\theta, m^*_\theta)$$

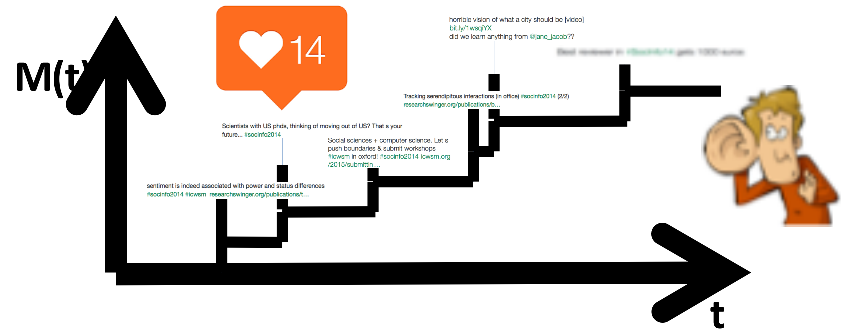
Policy

Intensity

Mark distribution

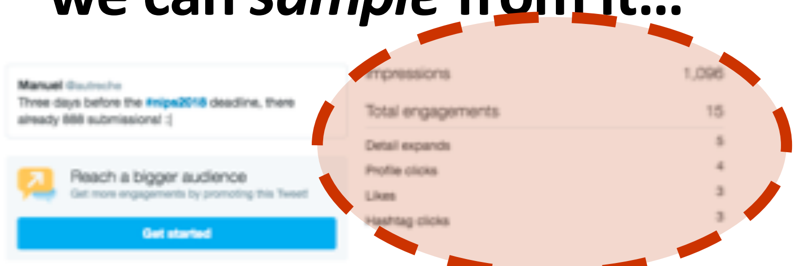
Parametrized using RNNs





$$p^*_{\mathcal{F};\phi} = (\lambda^*_\phi, m^*_\phi)$$

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



...and measure quality metrics (rewards)

[Upadhyay et al., 2018]

Policy gradient

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

$$\underset{p^*_{\mathcal{A};\theta}(\cdot)}{\text{maximize}} \quad \overbrace{\mathbb{E}_{\underbrace{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)}_{\text{Actions and environment are asynchronous!}}} \left[R^*(T) \right]}^{J(\theta)}$$

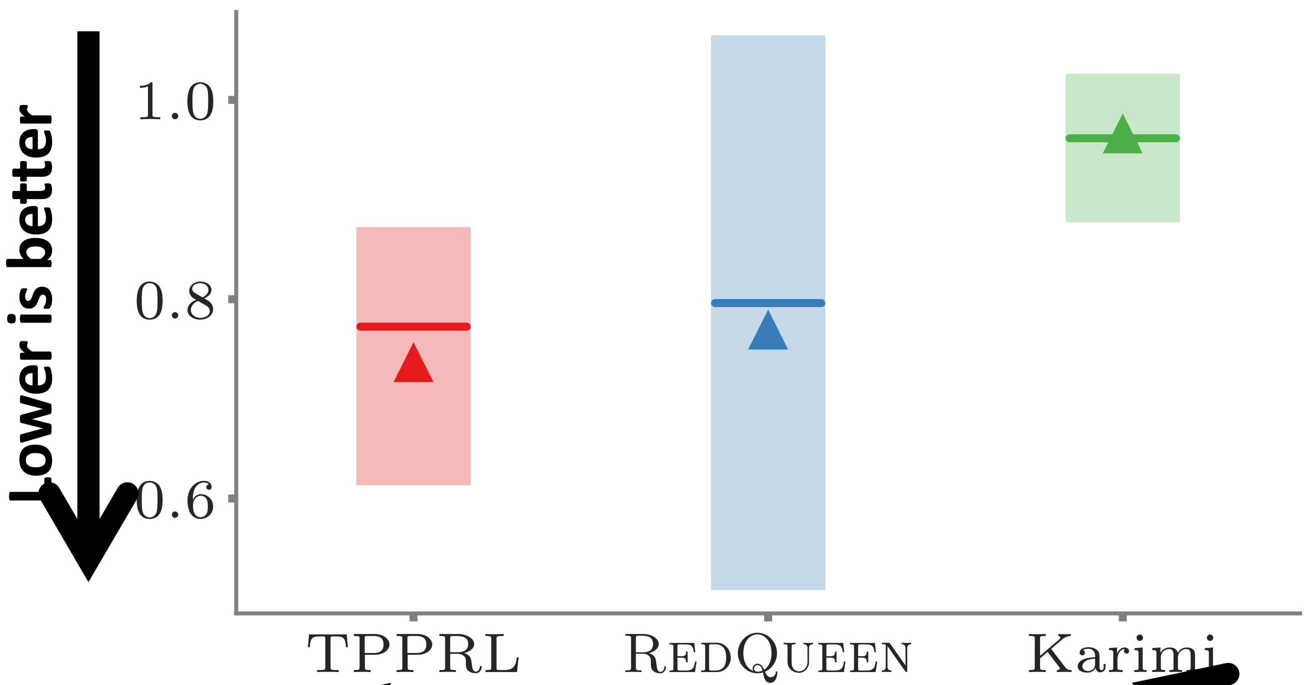
Reward
(Cumulative)

It can be shown that the reinforce trick is valid, i.e., we can compute the gradient and use SGD:

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

[Upadhyay et al., 2018]

Example: 100 broadcasters in Twitter



RL:
It *learns* the feed ranking

Optimal control:
It assumes feed in reverse chronological

Offline algorithm

[Upadhyay et al., 2018]

Many thanks!

Temporal Point processes (TPPs): Intro

1. Intensity function
2. Basic building blocks
3. Superposition
4. Marks and SDEs with jumps

Models & Inference

1. Modeling event sequences
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

RL & Control

1. Marked TPPs: a new setting
2. Stochastic optimal control
3. Reinforcement learning

Slides/references: learning.mpi-sws.org/tpp-icml18