Preference based Reinforcement Learning – Finite-time guarantees

Aarti Singh
Associate Professor

ICERM workshop
Aug 2, 2021

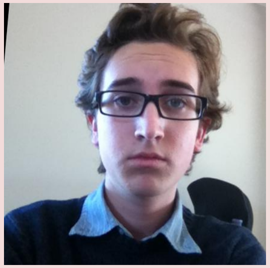
Preference vs Label feedback

Preference feedback can be easier and more accurate

Optimization

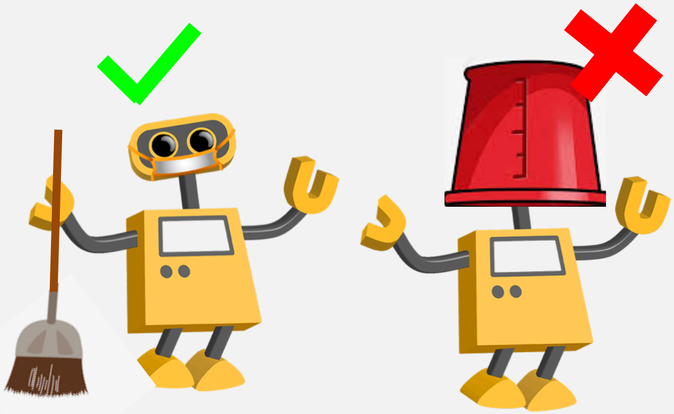
Evaluate fit vs compare

Regression



What's his age? Who looks older?

Reinforcement Learning



Clean if you see mess

Classification

Jet Engine Bracket

Compressor blades

Is a part 3D printable?

Setup

Two types of queries:

n labels

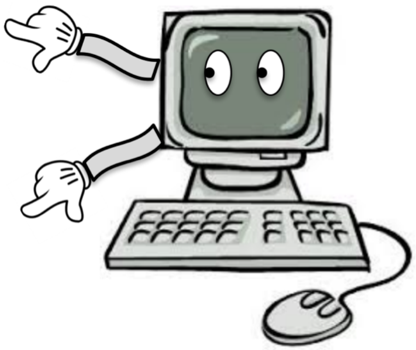
$$X$$

$$Y \sim f_y(X)$$

m comparisons

$$X, X'$$

$$Z \sim f_y(X) - f_y(X')$$



Algorithm that decides which type of data to collect, when and how much

<u>Goal:</u> Minimize the number of labels using preference feedback in the form of comparisons to achieve error ε

Label complexity, $n \sim f(\varepsilon)$ Comparison complexity, $m \sim g(\varepsilon)$

Preference feedback

How much can preference feedback help?

- Convex optimization [Jamieson+'13, Kumagai'17,Ailon+'14, Sui+'17]
- Non-convex GP optimization [UAI'20]
- **Reinforcement learning**
  - **Policy optimization** [NeurIPS'20]

- Classification [NIPS'17, Kane+'17]
- Threshold bandits [AISTATS'20]
- Regression [JMLR'20, ICML'18, Asilomar'18]

Take away message

How much can preference feedback help?

- Convex optimization
- Non-convex GP optimization
- Reinforcement learning
  - Policy optimization

Comparisons only suffice and rate same as labels only*

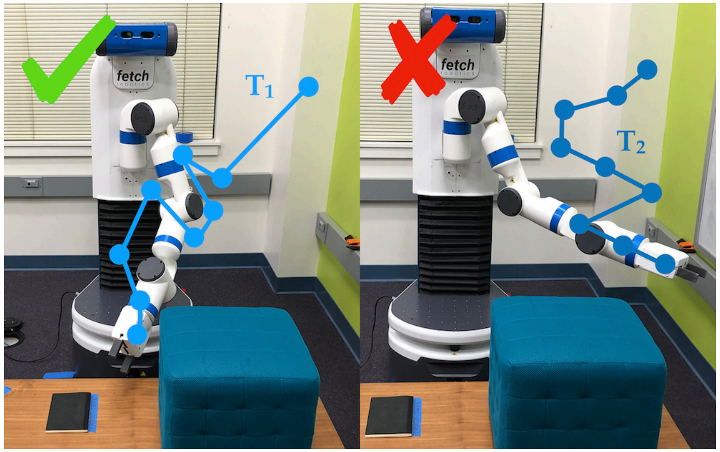
- Classification
- Threshold bandits
- Regression

Comparisons can reduce label complexity to 1-dimension*

*comparison noise no worse than labels

Preference based Reinforcement Learning

Rewards are hard to design in complex problems; poor rewards lead to unexpected and unsafe behaviors



[Palan et al'19]

[Amodei-Clark'16]

Preferences are easier to specify: comparisons of trajectories

[Novoseller et al '19] – asymptotic convergence of Thompson sampling for GP models of state transitions and trajectory reward comparisons

RL set up

Markov Decision Process

|S| States, |A| Actions, horizon H

- Deterministic unobserved reward r : S x A $\longrightarrow$ R $\quad r \in [0,1]$
- Random state transition p : S x A $\longrightarrow$ S

Non-stationary policy π : S $\longrightarrow$ A $\quad \pi = (\pi_1, \ldots, \pi_H)$

Value function $v_h^\pi(s) = E\left[\sum_{t=h}^{H} r(s_t, \pi(s_t)) \,|s_h = s\right]$ $\quad v_h^\pi(s) \in [0,1]$

Goal – Find policy $\widehat{\pi}$ such that with probability > 1-δ,

$$v^{\hat{\pi}}(s_0) \geq v^{\pi^*}(s_0) - \varepsilon$$

Assumptions

Preferences: comparison between trajectories τ and τ'
can compare partial trajectories

Even with perfect preferences, might not recover the optimal policy:

E.g., π has reward 1 w.p. 0.4 and 0 w.p. 0.6

π' has reward 0.1

τ' beats τ w.p. 0.6, but π has a higher expected reward

There exists an MDP and policies π_1, π_2, π_3 such that $\pi_1 \succ \pi_2 \succ \pi_3 \succ \pi_1$.

<u>Stochastic comparisons</u>:

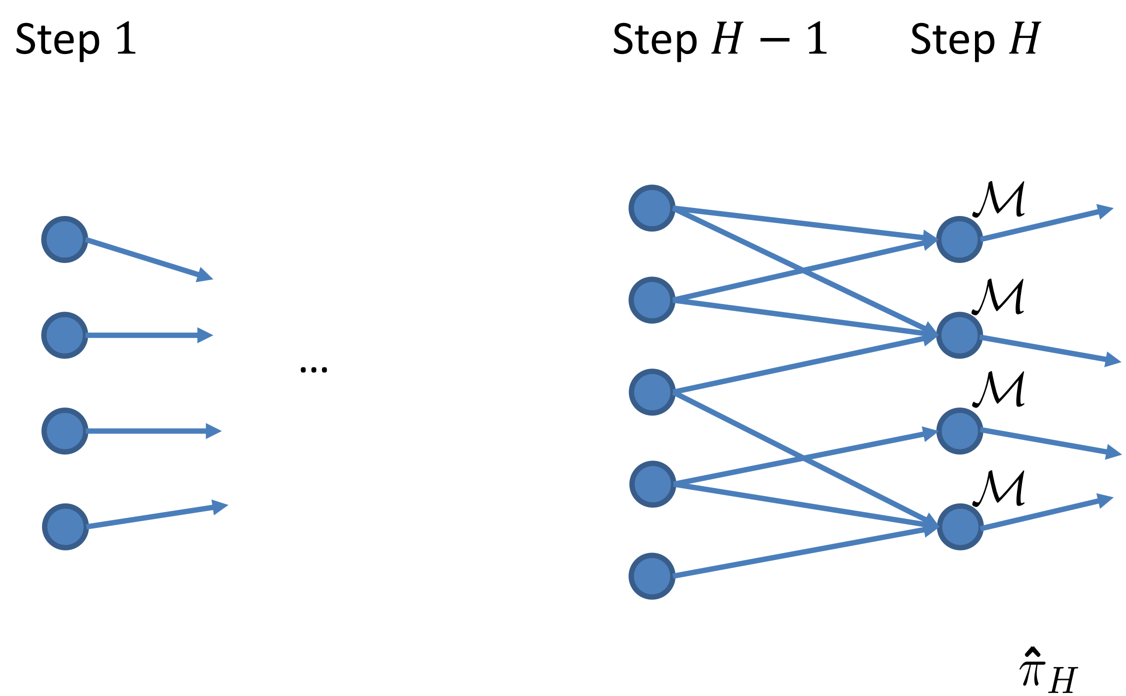
Let τ and τ' be two (random) trajectories by executing π and π' from state s, then

$$\Pr[\tau \succ \tau'] \geq C_0(v^{\pi}(s) - v^{\pi'}(s)).$$

PbRL with a simulator

Simulator: can start in any state

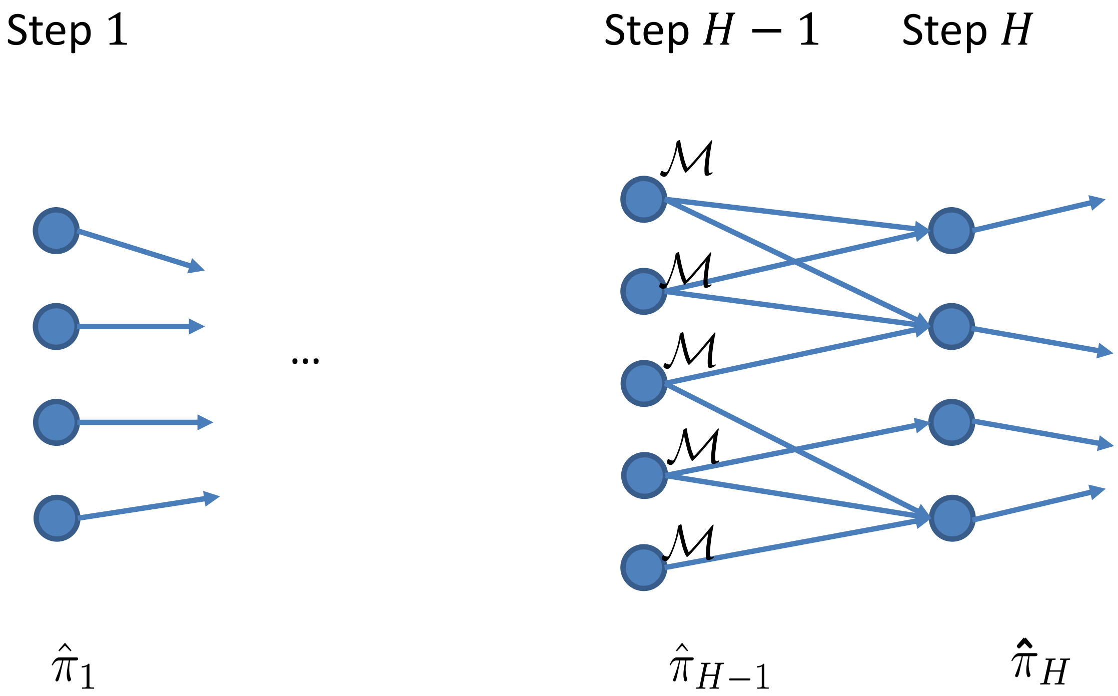
Dynamic programming to find best action for each state using a dueling bandit subroutine $\mathcal{M}$ (can't use value function since reward not known)



PbRL with a simulator

Simulator: can start in any state

Dynamic programming to find best action for each state using a dueling bandit subroutine $\mathcal{M}$ (can't use value function since reward not known)



PbRL with a simulator

Simulator: can start in any state

Dynamic programming to find best action for each state using a dueling bandit subroutine $\mathcal{M}$ (can't use value function since reward not known)

If we run an $(\varepsilon/H, \delta/S)$ optimal dueling bandit algorithm $\mathcal{M}$ on every state s, the algorithm finds an (ε, δ) correct policy using $\tilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right)$ simulator steps and $\tilde{O}\left(\frac{H^2SA}{\varepsilon^2}\right)$ comparisons.

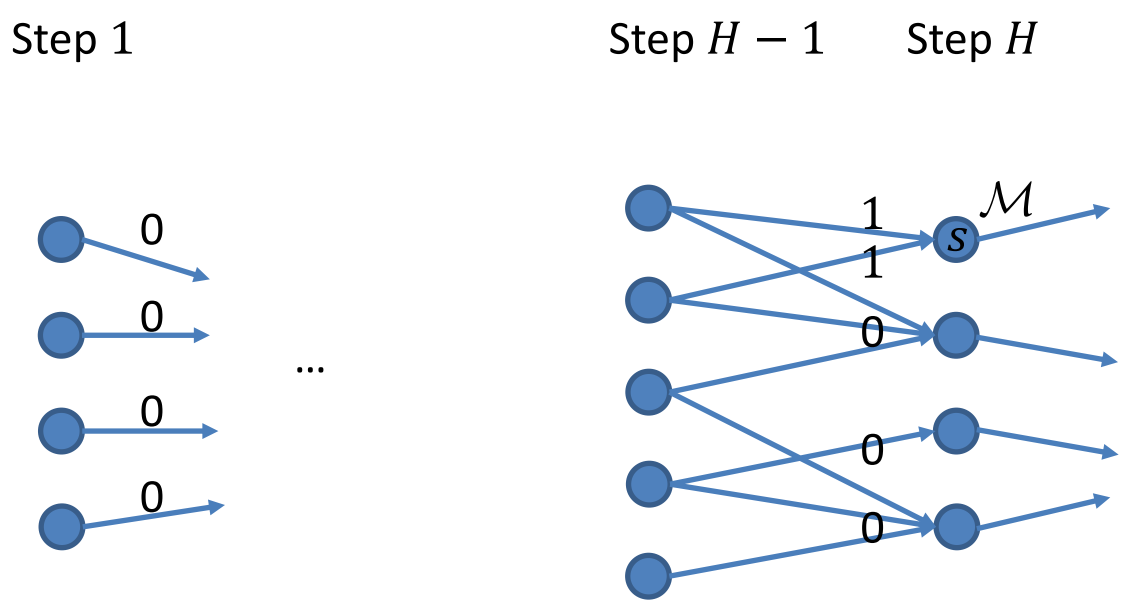
- Same number of steps and episodes as reward-based RL [Azar et al'13]
- We use OPT-Maximize [Falahatgar et al'17] as $\mathcal{M}$ - requires Strong Stochastic Transitivity (SST) of policy preference, implied by

$$\Pr[\tau \succ \tau'] \geq C(r(\tau) - r(\tau'))$$

PbRL without a simulator

Dynamic programming to find best action for each state using a dueling bandit subroutine + Synthetic reward function

reward-free exploration [Du et al'19, Jin et al'20, Misra et al'19]



For h = H, H-1, …, 1

For each s_h in S_h

r = { 1 if reached s_h; 0 otherwise }

Run any value based tabular RL to optimize r

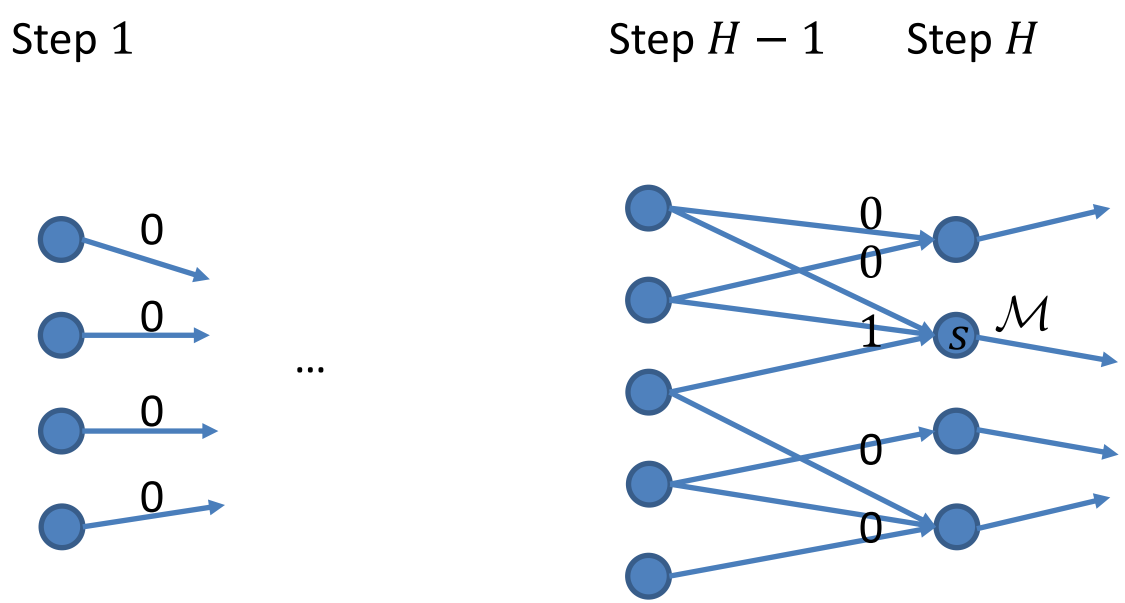
If reach s, generate trajectory & use dueling bandit $\mathcal{M}$ to find best action

- We use EULER [Zanette-Brunskill'20] as value based tabular RL algorithm

PbRL without a simulator

Dynamic programming to find best action for each state using a dueling bandit subroutine + Synthetic reward function

reward-free exploration [Du et al'19, Jin et al'20, Misra et al'19]



For h = H, H-1, …, 1

For each s_h in S_h

$r = \begin{cases} 1 \text{ if reached } s_h \\ 0 \text{ otherwise} \end{cases}$

Run any value based tabular RL to optimize r

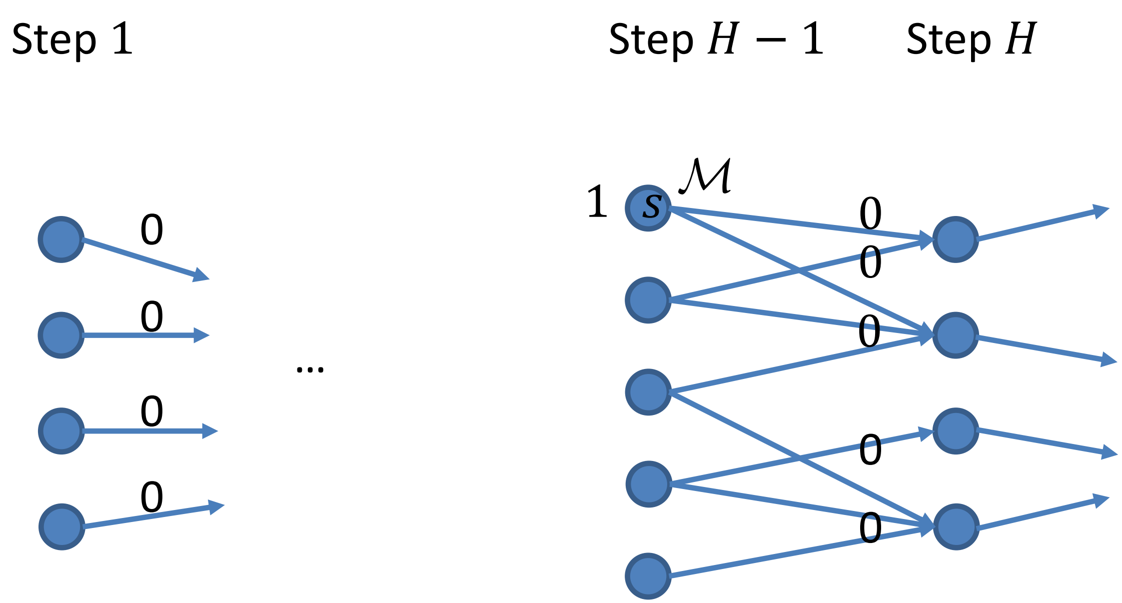
If reach s, generate trajectory & use dueling bandit $\mathcal{M}$ to find best action

➢ We use EULER [Zanette-Brunskill'20] as value based tabular RL algorithm

PbRL without a simulator

Dynamic programming to find best action for each state using a dueling bandit subroutine + Synthetic reward function

reward-free exploration [Du et al'19, Jin et al'20, Misra et al'19]



For h = H, H-1, ..., 1

For each s_h in S_h

r = { 1 if reached s_h; 0 otherwise }

Run any value based tabular RL to optimize r

If reach s, generate trajectory & use dueling bandit $\mathcal{M}$ to find best action

- We use EULER [Zanette-Brunskill'20] as value based tabular RL algorithm

PbRL without a simulator

Sample complexity depends on how we distribute errors over states in $\mathcal{M}$

- Uniform error over all states: better comparison complexity

Error on each state	Step complexity	Comparison Complexity
$\tilde{O}\left(\frac{\varepsilon}{H}\right)$	$\tilde{O}\left(\frac{H^3 S^2 A}{\varepsilon^3} + \frac{S^4 A H^3}{\varepsilon}\right)$	$\tilde{O}\left(\frac{H^2 S A}{\varepsilon^2}\right)$

We assume $S > H$ for simplicity

Value based RL: $\tilde{O}\left(\frac{H^3 S A}{\varepsilon^2}\right)$ steps and $\tilde{O}\left(\frac{H^2 S A}{\varepsilon^2}\right)$ episodes [Azar et al. 2017]

➢ Comparisons match number of episodes in reward based RL

PbRL without a simulator

Sample complexity depends on how we distribute errors over states in $\mathcal{M}$

- Uniform error over all states: better comparison complexity
- Varied acc to reachability of states: better step complexity

Error on each state	Step complexity	Comparison Complexity
$\tilde{O}\left(\frac{\varepsilon}{H}\right)$	$\tilde{O}\left(\frac{H^3S^2A}{\varepsilon^3}+\frac{S^4AH^3}{\varepsilon}\right)$	$\tilde{O}\left(\frac{H^2SA}{\varepsilon^2}\right)$
unconstrained	$\tilde{O}\left(\frac{H^2S^2A}{\varepsilon^2}+\frac{S^4AH^3}{\varepsilon}\right)$	$\tilde{O}\left(\frac{HS^2A}{\varepsilon^2}\right)$

We assume $S > H$ for simplicity

Corresponds to error $O\left(\frac{\varepsilon}{\sqrt{\mu(s)SH}}\right)$

$\mu(s)$ is the maximum probability to reach s using any policy

Open Questions

- <u>Theory:</u> Comparison complexity bounds
  - can pref RL complexity be independent of horizon H,
  - improved step complexity if non-uniform error over states (∝ reachability) but worse comparison complexity

- <u>Adaptive algorithms:</u>

  Hybrid reward-preference feedback

  Limited rounds of interaction

- <u>Other forms of feedback:</u>

  Comparisons of features

  Causal relations, knowledge graphs, demos, instructions, …

Acknowledgements & References

Yichong Xu

Artur Dubrawski
Ruosong Wang
Lin F. Yang

Sivaraman Balakrishnan
Xi Chen
Hariank Muthakana
Aparna Joshi
Kyle Miller

Preference-based Reinforcement Learning with Finite-Time Guarantees, *NeurIPS'20.*

Zeroth Order Non-convex optimization with Dueling-Choice Bandits, *UAI'20.*

Thresholding Bandit Problem with both Duels and Pulls, *AISTATS'20.*

Interactive Linear Regression with Pairwise Comparisons, *Asilomar'18.*

Nonparametric Regression with Comparisons: Escaping the Curse of Dimensionality with Ordinal Information, *ICML'18, JMLR'20.*

Noise-Tolerant Interactive Learning Using Pairwise Comparisons, *NIPS 2017*.