Preference based Reinforcement Learning – Finite-time guarantees

Aarti Singh
Associate Professor

ICERM workshop
Aug 2, 2021
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**
Is a part 3D printable?
Two types of queries:

- \( n \) labels
  \[ Y \sim f_y(X) \]

- \( m \) comparisons
  \[ Z \sim f_y(X) - f_y(X') \]

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

**Goal:** Minimize the number of labels using preference feedback in the form of comparisons to achieve error \( \varepsilon \)

Label complexity, \( n \sim f(\varepsilon) \)  
Comparison complexity, \( m \sim g(\varepsilon) \)
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]
How much can preference feedback help?

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

Taking away message:

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

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.
Markov Decision Process

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

- Deterministic unobserved reward $r : S \times A \rightarrow \mathbb{R}$ \quad $r \in [0,1]$
- Random state transition $p : S \times A \rightarrow S$

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

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

Goal – Find policy $\hat{\pi}$ such that with probability $> 1-\delta$, $v^\hat{\pi}(s_0) \geq v^{\pi^*}(s_0) - \varepsilon$
Preferences: comparison between trajectories $\tau$ and $\tau'$
can compare partial trajectories

Even with perfect preferences, might not recover the optimal policy:
E.g., $\pi$ has reward 1 w.p. 0.4 and 0 w.p. 0.6
$\pi'$ has reward 0.1
$\tau'$ beats $\tau$ w.p. 0.6, but $\pi$ has a higher expected reward

There exists an MDP and policies $\pi_1, \pi_2, \pi_3$ such that
$\pi_1 > \pi_2 > \pi_3 > \pi_1$.

Stochastic comparisons:
Let $\tau$ and $\tau'$ be two (random) trajectories by executing $\pi$ and $\pi'$
from state $s$, then
$$\Pr[\tau \succcurlyeq \tau'] \geq C_0 (v^\pi(s) - v^\pi'(s)).$$
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 $(\varepsilon, \delta)$ 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]

Step 1

\[ 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \]

Step \( H - 1 \) Step \( H \)

\[ S \]

For \( h = H, H-1, \ldots, 1 \)

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]

Step 1

\[
\begin{align*}
&\begin{array}{c}
\text{Step } H - 1 \\
\text{Step } H \\
\text{For } h = H, H-1, \ldots, 1
\end{array}
\end{align*}
\]

For each \( s_h \) in \( S_h \)

\[
\begin{align*}
r &= \begin{cases} 
1 & \text{if reached } s_h \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

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]

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

<table>
<thead>
<tr>
<th>Error on each state</th>
<th>Step complexity</th>
<th>Comparison Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O} \left( \frac{\varepsilon}{H} \right)$</td>
<td>$\tilde{O} \left( \frac{H^3 S^2 A}{\varepsilon^3} + \frac{S^4 A H^3}{\varepsilon} \right)$</td>
<td>$\tilde{O} \left( \frac{H^2 S A}{\varepsilon^2} \right)$</td>
</tr>
</tbody>
</table>

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
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

<table>
<thead>
<tr>
<th>Error on each state</th>
<th>Step complexity</th>
<th>Comparison Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{O} \left( \frac{\varepsilon}{H} \right)$</td>
<td>$\tilde{O} \left( \frac{H^3 S^2 A}{\varepsilon^3} + \frac{S^4 A H^3}{\varepsilon} \right)$</td>
<td>$\tilde{O} \left( \frac{H^2 S A}{\varepsilon^2} \right)$</td>
</tr>
<tr>
<td>unconstrained</td>
<td>$\tilde{O} \left( \frac{H^2 S^2 A}{\varepsilon^2} + \frac{S^4 A H^3}{\varepsilon} \right)$</td>
<td>$\tilde{O} \left( \frac{H S^2 A}{\varepsilon^2} \right)$</td>
</tr>
</tbody>
</table>

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

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

We assume $S > H$ for simplicity.
Open Questions

- **Theory:** Comparison complexity bounds
  - Can RL complexity be independent of horizon $H$,
  - Improved step complexity if non-uniform error over states ($\propto$ reachability) but worse comparison complexity

- **Adaptive algorithms:**
  - Hybrid reward-preference feedback
  - Limited rounds of interaction

- **Other forms of feedback:**
  - Comparisons of features
  - Causal relations, knowledge graphs, demos, instructions, ...
Acknowledgements & References

Artur Dubrawski
Ruosong Wang
Lin F. Yang
Sivaraman Balakrishnan
Xi Chen
Hariank Muthakana
Aparna Joshi
Kyle Miller

Yichong Xu


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.