Reinforcement Learning from Small Data

Mengdi Wang
RL is Markov decision process

- A finite set of states $S$
- A finite set of actions $A$
- Reward is given at each state-action pair $(s,a)$:
  \[ r(s,a) \in [0,1] \]
- State transits to $s'$ with prob. $P(s'|s,a)$
- Find a best policy $\pi:S\rightarrow A$ such that
  \[
  \max_\pi v^\pi = \mathbb{E}^\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]
  \]
- $\gamma \in (0,1)$ is a discount factor

When the full model is known, MDP can be solved using Dynamic Programming
Why RL is not only about optimization?

*Probabilities $P(s'| s, a)$ not unknown*
Reinforcement learning achieves phenomenal empirical successes
What if the data/trial is limited and costly
Statistically proven reinforcement learning?

How many samples are needed to learn an 90%-optimal policy?
What does a sample mean?

Samples are state-transition triplets \((s,a,s')\).
Sharp sample complexities for MDP

- **Information-theoretical limit (Azar et al. 2013):** Any method finding an \( \epsilon \)-optimal policy with probability 2/3 needs at least sample size

\[
\Omega \left( \frac{|SA|}{(1 - \gamma)^3 \epsilon^2} \right)
\]

- **The optimal sampling-based algorithm (with Sidford, Yang, Ye, 2018):** it finds an \( \epsilon \)-optimal policy with probability 1-\( \delta \) using sample size

\[
O \left( \frac{|SA|}{(1 - \gamma)^3 \epsilon^2 \log \frac{1}{\delta}} \right)
\]

- **Q-Learning is provably efficient (Jin et al 2018):** For H-horizon MDP, Q learning achieves regret

\[
\Omega \left( \sqrt{|SA| H^3 T} \right)
\]

Statistical complexity of RL (in this basic setting) is finally well understood
S is too big
Bellman equation and Duality

**Bellman equation is the optimality principal for MDP** (in the average-reward case, where $\gamma=1$)

$$
\bar{v}^* + v^*(s) = \max_a \left\{ \sum_{s' \in \mathcal{S}} P_a(s, s')v^*(s') + r_a(s) \right\}, \quad \forall s \in \mathcal{S}
$$

- The $\max$ operation applies to every state & actions -&gt; nonlinearity + high dim

**Bellman equation is equivalent to a bilinear saddle point problem** (Wang 2017)

$$
\min_v \max_{\mu \in \Delta} \left\{ L(v, \mu) = \sum_a (\mu_a^T((I - P_a)v + r_a)) \right\}
$$

- Value function
- Stationary state-action distribution

- Strong duality between value function and invariant measure
- **SA x S LP**
State Feature Map

- Suppose that we are given a state feature map

\[ \text{state} \mapsto [\phi_1(\text{state}), \ldots, \phi_N(\text{state})] \in \mathbb{R}^N \]

- Can we do better?

- Tetris can be solved well using 22 features
  - Feature 1: Height of wall
  - Feature 2: Number of holes
Representing value function using linear combination of features

- The value function of a policy is the expected cumulative reward as the initial state varies:

\[ V^\pi : \mathcal{S} \rightarrow \mathbb{R}, \quad V^\pi(s) = \mathbb{E}^\pi \left[ \sum_{t=0}^{H} r(s_t, a_t) \mid s_0 = s \right] \]

- Suppose that the high-dimensional value vector admits a linear model:

\[ V^\pi(s) \approx w_1 \phi_1(s) + \ldots + w_N \phi_N(s) \]

- Example:

\[ \text{Score of } = w_1 \times \text{Height of Wall} + w_2 \times \# \text{ Holes} + \ldots \]
Reducing Bellman equation using features

$\tilde{v}^* + v^*(s) = \max_a \left\{ \sum_{s' \in S} P_a(s, s')v^*(s') + r_a(s) \right\}, \forall s$

Bellman eq:
\{ High-dim Nonlinear \}

Bellman saddle point:
\{ High-dim \}

$v(\cdot) \approx \sum_{i=1}^{r_s} w_i \phi_i(\cdot)$

$\mu(s, a) \approx \sum_{i=1}^{r_s} \sum_{j=1}^{r_a} u_{ij} \phi_i(s) \psi_j(a)$

\{ Low-dim Convex-concave Strong duality Parametric \}
RL complexity reduces with given features

Suppose that good state and action features are known

- For average-reward RL, a primal-dual policy learning method finds the optimal policy using sample size (with YC, LL, 2018)

\[ \Theta \left( C \cdot \frac{|N_S N_A|}{\epsilon^2} \right) \]

where C is polynomial in mixing and ergodicity parameters

- Sample-Optimal Parametric Q-Learning for discounted RL (with LY, 2019)

\[ \Theta \left( \frac{|N_S N_A|}{\epsilon^2(1 - \gamma)^3} \right) \]

- Reduced S to # features

- Not very surprising … Can we do well without known features?
What are good state features?

- Given a stationary Markov chain with transition operator $P$ and one-step reward function $r$, the average-reward difference-of-value function is given by

$$v = \lim_{T \to \infty} \left( r + Pr + P^2r + \cdots + P^Tr - (T\bar{r}) \cdot 1 \right).$$

- Suppose that $P$ admits the decomposition

$$P = \Phi \tilde{P} \Phi^T$$

- Both the value $v$ and the invariant measure $\xi$ lie in the span of $\phi$:

$$v, \xi \in \text{Span}(\Phi)$$

Good features $\phi$ shall span the column spaces of $P$
Consider an ergodic Markov process

\[ X_1, X_2, \ldots, X_t, \ldots \]

Spectral decomposition of the transition operator

\[
\mathbb{P}(X_{t+1} \mid X_t) \approx \sum_{i} u_i(X_t) v_i(X_{t+1})
\]

Markov features

- \( u_i(\cdot) \)'s \( v_i(\cdot) \)'s are natural features for RL
- Reward-independent

Objective: Learn the optimal state embedding from data

\[
\max_{\Psi : X \to \mathbb{R}^r, \Psi_j \in H} \text{Tr} \left( \int \Psi(x)p(x, y)\Psi(y)^T dxdy \right)
\]

- (w A Zhang, 2018, w X L, A Zhang, 2018, 2019)

Kernelized state embedding from random features

**Data:** A high-dimensional time series and a kernel space

\[ X_1, X_2, \ldots, X_t, \ldots, \quad \text{where} \quad X_t \in \mathbb{R}^d \]

**Solution:**

1. Approximating the kernel function \( K \) using random features
   
   \[ K(x, y) \approx \phi(x)^T \phi(y) \quad \phi(\cdot) = [\phi_1(\cdot), \ldots, \phi_N(\cdot)]^T \]

2. Estimate a Galerkin projection matrix

   \[ \hat{Q} = \frac{1}{T} \sum_{t=1}^{T} \phi(X_t)\phi(X_{t+1})^T \]

3. Find the best rank-\( r \) approximation

   \[ \hat{Q} = \hat{U} \Lambda \hat{V}^T, \quad \hat{Q}_r = \hat{U}_r \Lambda_r \hat{V}_r^T \]

**Output:** Low-dim state embedding (a kernelized diffusion map)

\[ X \mapsto P(\cdot | X) \mapsto \phi(X)^T \hat{U}_r \in \mathbb{R}^r \]
Finding Metastable State Clusters

• We want to find a partition of the state space such that that states within the same set shares similar future paths

\[
\min_\Omega \, \min_1 \sum_{i=1}^m \int_{\Omega_i} \pi(x) \| p(\cdot \mid x) - q_i(\cdot) \|^2_{L^2} dx,
\]

• If the MC is reversible, the problem finds the optimal metastable partition [E 2008]

\[
(A_1^*, \ldots, A_m^*) = \arg\max_{A_1, \ldots, A_m} \sum_{k=1}^m p(A_k \mid A_k)
\]

• Solution: 1. Estimate state embedding; 2. Solve

\[
\min_{(\Omega_1, \ldots, \Omega_m)} \sum_{i=1}^m \sum_{i \in [N]} \| \hat{\Psi}(x_i) - s_i \|^2 dx
\]
Example: stochastic diffusion process

Potential Function

True Invariant Measure $p(x_1, x_2)$
Learning state embeddings from sample path

State embeddings in 3D

Clusters from State aggregation learning
Clustering of state embedding learned from $P^t$

Learning metastable sets from state trajectories
An example of NYC taxi data

Data: NYC taxi trips, 10e6 records
Each trip is a state-transition sample of a city wide random walk

Aggregation partition of NYC learned from data
An interpretable model: Soft State Aggregation

Soft state aggregation reduces the complexity of control and RL [Singh 94, Bertsekas 97]

In the matrix case:

\[ P^* = U^{*T}V^* \]

\[ U^* \geq 0, V^* \geq 0, U^*1 = 1, V^{*T}1 = 1 \]
Unsupervised state aggregation learning

\[ P^* = U^{*T}V^* \]

\[ U^* \geq 0, V^* \geq 0, U^*1 = 1, V^*T1 = 1 \]

- Nonnegative factorization
- Generally nonidentifiable
- Need anchors (like topic modeling)
- Exploit the polytope

estimate Markov features  \( \hat{U}, \hat{V} \)  vertex hunting states in feature space \( s \) \( \rightarrow [\hat{U}_1(s), \hat{U}_2(s), \hat{U}_3(s)] \)

anchor states reveals meta-states
**Theoretical results for aggregation learning**

**Theory:** Assuming \( r \) meta-states and existence of anchor states. Given a sample state trajectory of length \( n \) and under some ergodicity/mixing assumptions, the aggregation learning method estimates individual aggregation/disaggregation distributions to the accuracy

\[
\mathbb{E} \left[ \max_s \| \tilde{U}(s, \cdot) - U^*(s, \cdot) \|_{TV} + \max_k \| \tilde{V}_k - V_k^* \|_{TV} \right] \leq C \cdot \left( \sqrt{\frac{rS_{mix}}{n}} \right)
\]

(\text{w. Y Duan, T Ke, 2018})
Soft state aggregation for NYC taxi data

State aggregation learning identifies “anchor regions” of the meta-states - representing leading traffic modes
Soft state aggregation for NYC taxi data

Estimated aggregation and disaggregation distributions
Example: State Trajectories of Demon Attack

(a) Before Embedding

(b) After Embedding

Visualization of game states before and after embedding in t-SNE plots.
Game states that are close after embedding

- **About to score; both moving to the left**
- **New demons appearing**
- **Waiting for new targets; moving to center from opposite ends**

State embedding identifies states as similar if they share similar future paths.
Provably-efficient RL with small data is possible, both in theory and in practice

Thank you!