Efficient Planning in Large MDPs with Weak Linear Function Approximation

Roshan Shariff
roshan.shariff@ualberta.ca
University of Alberta ◆ Amii

Csaba Szepesvári
szepesva@ualberta.ca
DeepMind ◆ University of Alberta ◆ Amii

Alberta Machine Intelligence Institute

Large Markov Decision Process (MDP)

- Large state space of size S
- Action space of size A
- Infinite horizon
- Discounted by factor γ

Weak Linear Function Approximation

- Feature representation $\varphi(s) \in \mathbb{R}^d$ for each state s
- Small approximation error for *optimal* value function:

$$|\varphi(s)^\top \theta^* - v^*(s)| \le \varepsilon_{\text{approx}} \text{ for some } \theta^* \in \mathbb{R}^d$$

Weak: only optimal value function need be representable!

The Planning Problem

- Local planning: for any *given* state s_0, output random action a
- Uses simulator to sample next state and reward for any state and action
- Goal is to be close-to-optimal:

$$\mathbb{E}[q^*(s_0, a)] \ge v^*(s_0) - \varepsilon(1-\gamma)$$

- Resulting policy is almost optimal:

$$v_\pi(s) \ge v^*(s) - \varepsilon \qquad \text{for all states } s$$

Planning in Large MDPs

Avoid scaling with number of states, or exponential scaling in horizon ($H = 1/(1-\gamma)$ is the effective horizon)

✘ **Impossible** without additional assumptions!
Need $(1/\varepsilon)^H$ samples for ε-suboptimal policy [Kearns, et al., 2002]

✘ **Impossible** with weak function approximation, if policy must be $\varepsilon_{\text{approx}}$-suboptimal [Du, et al., 2020]

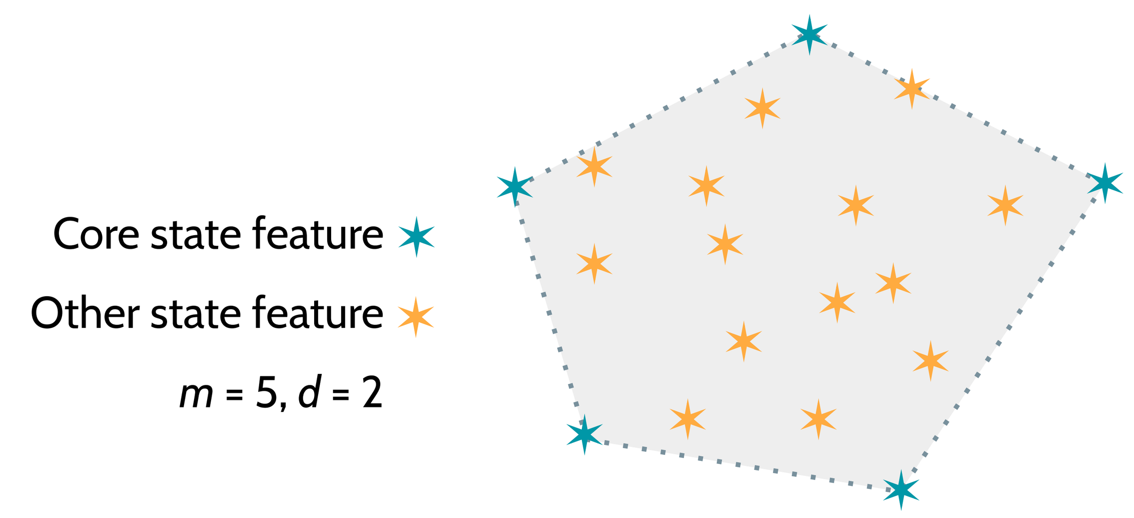
✔ **Possible** for $(\varepsilon_{\text{approx}} H^2 \sqrt{d})$-suboptimal policies, but requires value functions of all policies to be representable with low error [Lattimore, et al., 2020; Van Roy & Dong, 2019]

✔ **Possible** with strong assumptions on MDP dynamics (linear MDPs, low Bellman rank, etc.)

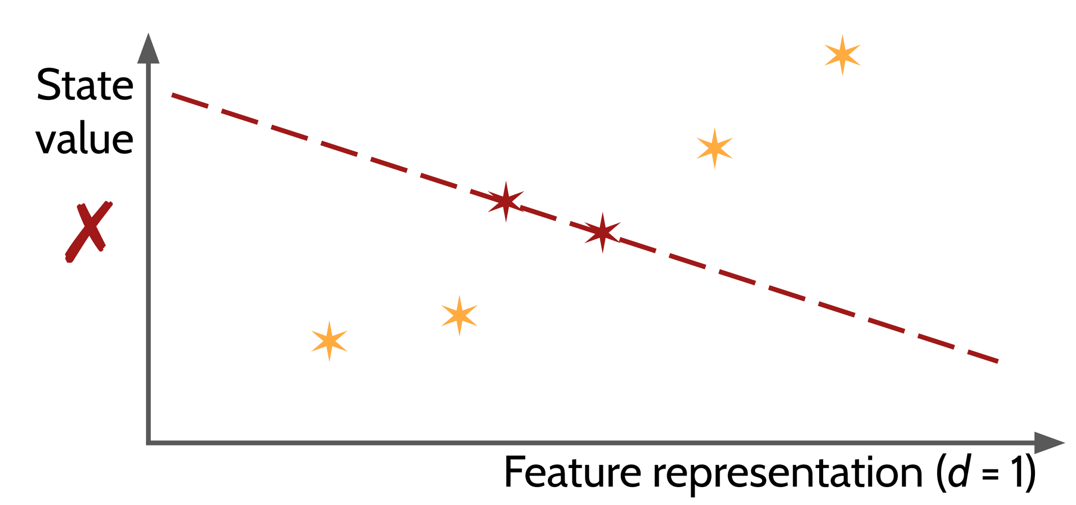
Can we plan efficiently in large MDPs with only weak linear function approximation and no restrictions on MDP dynamics?

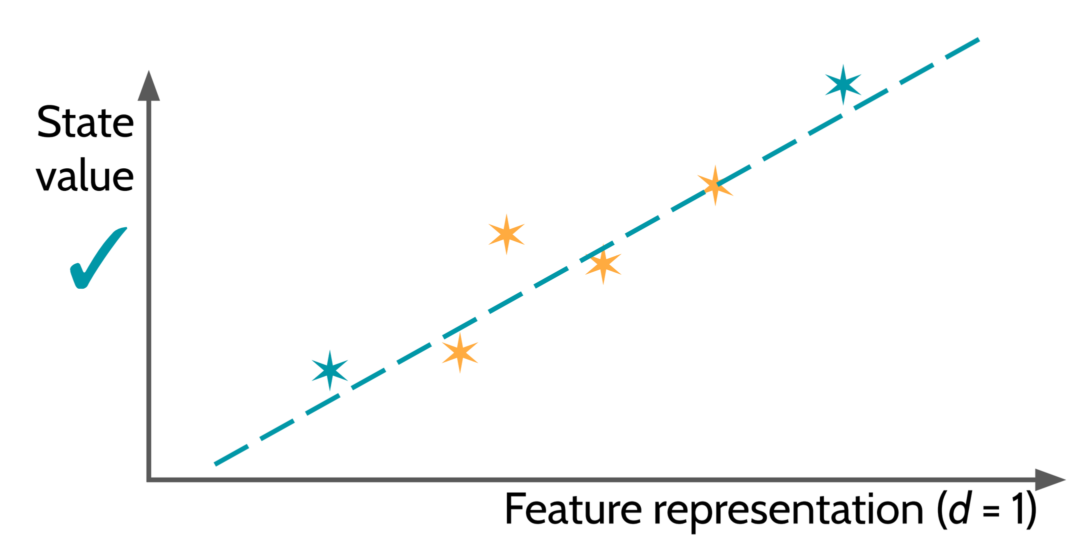
Assumption: Core States

A small subset of states (of size m)
whose features' convex hull covers all other state features



- Purely **geometric** condition on feature representation
- Use feature representation to generalize value function from core states to other states
- Intuition: core states with "extreme" features avoid extrapolation





CoreStoMP

A Saddle-Point Algorithm for Planning with Core States

- Based on Relaxed Approximate Linear Program [Lakshminarayanan, et al., 2018]
- Uses Stochastic Mirror-Prox to approximately solve saddle-point formulation of problem
- Gradient estimates come from simulator

Main Result

Running CoreStoMP on state s for T iterations:

- Uses the simulator $O(mAT)$ times
- Outputs random action a with

$$\mathbb{E}_a[v^*(s) - q^*(s,a)] \le O\left(\frac{\varepsilon_{\text{approx}}}{1-\gamma}\right) + \tilde{O}\left(\frac{1}{(1-\gamma)^2}\sqrt{\frac{m}{T}}\right)$$

- Results in policy π with value loss

$$\max_{s\in\mathcal{S}} v^*(s) - v_\pi(s) \le O\left(\frac{\varepsilon_{\text{approx}}}{(1-\gamma)^2}\right) + \tilde{O}\left(\frac{1}{(1-\gamma)^3}\sqrt{\frac{m}{T}}\right)$$

Algorithm 1 CoreStoMP: Stochastic Mirror-Prox for Planning with Core States

Parameters: T, B, η
Initialization: $\boldsymbol{\theta}_0 \leftarrow \mathbf{0} \in \mathbb{R}^d,\ \lambda_{0,(0,a)} \leftarrow 1/A,\ \lambda_{0,(s,a)} \leftarrow \gamma/((1-\gamma)mA) \quad \forall\, s \in \mathcal{S}_*,\ a \in \mathcal{A}$
for $\tau = 1, 2, \ldots, T$ **do**
  $(\boldsymbol{\theta}'_\tau, \boldsymbol{\lambda}'_\tau) \leftarrow \textsc{ProxUpdate}(B, \eta, (\boldsymbol{\theta}_{\tau-1}, \boldsymbol{\lambda}_{\tau-1}), (\boldsymbol{\xi}, \boldsymbol{\rho}))$ where $\boldsymbol{\xi} \sim \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{\lambda}_{\tau-1}), \boldsymbol{\rho} \sim \hat{f}_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_{\tau-1})$
  $(\boldsymbol{\theta}_\tau, \boldsymbol{\lambda}_\tau) \leftarrow \textsc{ProxUpdate}(B, \eta, (\boldsymbol{\theta}_{\tau-1}, \boldsymbol{\lambda}_{\tau-1}), (\boldsymbol{\xi}', \boldsymbol{\rho}'))$ where $\boldsymbol{\xi}' \sim \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{\lambda}'_\tau), \boldsymbol{\rho}' \sim \hat{f}_{\boldsymbol{\lambda}}(\boldsymbol{\theta}'_\tau)$
end for
return $\left(\sum_{\tau=1}^T \boldsymbol{\lambda}_\tau\right)/T$

function $\textsc{ProxUpdate}(B, \eta, (\boldsymbol{\theta}, \boldsymbol{\lambda}), (\boldsymbol{\xi}, \boldsymbol{\rho}))$
  $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta} - \eta\boldsymbol{\xi}$
  $\boldsymbol{\theta}' \leftarrow \tilde{\boldsymbol{\theta}}/\max\{1, \|\boldsymbol{\Phi}_*\boldsymbol{\theta}\|_2/B\}$
  $\tilde{\boldsymbol{\lambda}} \leftarrow \exp(\log\boldsymbol{\lambda} + \eta\boldsymbol{\rho})$
  $\boldsymbol{\lambda}'_0 \leftarrow \tilde{\boldsymbol{\lambda}}_0/\|\tilde{\boldsymbol{\lambda}}_0\|_1$ where $\tilde{\boldsymbol{\lambda}}_0 := [\tilde{\lambda}_{0,a}]_{a\in\mathcal{A}}$ and similarly for $\boldsymbol{\lambda}'$.
  $\boldsymbol{\lambda}'_* \leftarrow (\gamma/(1-\gamma))\tilde{\boldsymbol{\lambda}}_*/\|\tilde{\boldsymbol{\lambda}}_*\|_1$ where $\tilde{\boldsymbol{\lambda}}_* := [\tilde{\lambda}_{i,a}]_{i\in[m],a\in\mathcal{A}}$ and similarly for $\boldsymbol{\lambda}'$.
  return $(\boldsymbol{\theta}', \boldsymbol{\lambda}')$
end function

References

- Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? In ICLR (2020).
- Kearns, M., Mansour, Y., and Ng, A. Y. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. Machine Learning, 49:193–208 (2002).
- Lakshminarayanan, C., Bhatnagar, S., and Szepesvári, Cs. A linearly relaxed approximate linear program for Markov decision processes. IEEE Transactions on Automatic Control, 63(4):1185–1191 (Apr. 2018).
- Lattimore, T., Szepesvári, Cs., and Weisz, G. Learning with good feature representations in bandits and in RL with a generative model. In ICML (2020).
- Van Roy, B. and Dong, S. Comments on the Du-Kakade-Wang-Yang lower bounds (2019). arXiv:1911.07910 .