Regret bounds for kernel-based reinforcement learning

Omar D. Domingues¹, Pierre Ménard¹, Matteo Pirotta², Emilie Kaufmann¹,³, Michal Valko¹,⁴

¹Inria Lille - Nord Europe
²Facebook AI Research, Paris
³CNRS & Université de Lille
⁴DeepMind, Paris
Reinforcement Learning

Framework that models several learning problems, for instance

- Controlling robots to reach a goal
- Playing games
- Recommendation systems
- Self-driving cars

Very hard to solve

- Requires a lot of data
- How to collect data efficiently?
Mathematical model

- The environment is modeled as a Markov decision process
  - State and action sets \( S, A \)
  - Transition probabilities \( P(s' | s, a) \)
  - Reward function \( r(s, a) \)

- The agent follows a policy
  - Action to take in state \( s \) at time \( h \): \( \pi_h(s) \)

- Goal: maximize the sum of rewards

\[
\max_{\pi} \mathbb{E}_\pi \left[ \sum_{h=1}^{H} r(S_h, A_h) \right]
\]

- If we have a simulator, use approximate dynamic programming (ADP).
Kernel-based reinforcement learning

- Approximate DP technique introduced by Ormoneit & Sen (2002)*
- Simulate **N independent samples** from the MDP
  - i-th sample = (state, action, reward, next state) = \((s_i, a_i, r_i, s'_i)\)
- **Estimate a model**

\[
\hat{r}(s, a) = \frac{\sum_i w_i(s, a) r_i}{\beta + \sum_i w_i(s, a)}, \quad \hat{P}(s'|s, a) = \frac{\sum_i w_i(s, a) \delta_{s'_i}(s')}{\beta + \sum_i w_i(s, a)}
\]

- Run **dynamic programming on the estimated model**, complexity = O(N²)
- **Asymptotic convergence guarantee**

Our contribution: Kernel-based RL with exploration*

- Exploration strategy to collect data online (not independent)
- Finite time-guarantee, "error per sample" converges to zero

\[
\frac{\text{Regret}(N)}{N} \lesssim \left( \frac{1}{N} \right)^{\frac{1}{2d+1}}
\]

- Assumption: model is Lipschitz continuous with respect to a given metric
- \( d = \) covering dimension of the state-action space

Example
Thank you!
• How to avoid the curse of dimensionality?
• How to improve the runtime?
• Extension to non-stationary MDPs*