Chapter 4

Computational Complexity

In the previous chapter, we showed how to find an approximately optimal policy in a sample-efficient manner—that is, by interacting with the transition and reward functions at most polynomially many times in the feature dimension d and horizon H. In this chapter, we show that despite the statistical problem being easy, the computational problem is hard: no efficient algorithm exists for solving the same problem under standard complexity assumptions.

4.1 Complexity Problems

Our proof is based on a reduction from the classical 3-SAT problem:

Definition 4.1 (3-SAT)**.** *Given a Boolean formula φ in conjunctive normal form (CNF) with v variables and $O(v)$ clauses, the goal is to determine whether φ is satisfiable—that is, whether there exists an assignment $w \in \{0,1\}^v$ such that every clause in φ evaluates to `True`.*

For example, the formula $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$ is satisfiable. We now formally define the interaction model for the linear reinforcement learning (RL) problem:

Definition 4.2 (Linear RL and Interaction Model)**.** *An algorithm is given access to a deterministic finite-horizon MDP M with horizon H, and the following oracles:*

- ***Reward oracle:*** *Given a state s and action a, returns a sample from $R(s,a)$.*
- ***Transition oracle:*** *Given a state s and action a, returns the next state $T(s,a)$.*
- ***Feature oracle:*** *Given a state s (or a state-action pair (s,a)), returns the corresponding d-dimensional feature vector $\phi(s)$ or $\phi(s,a)$.*

Each oracle call has constant runtime, and all input/output sizes are polynomial in the feature dimension d.

We assume that the optimal value functions Q^* and V^* are linear in these features (Definition 3.1). Our goal is to prove the following computational lower bound for finding an approximate optimal policy:

Theorem 4.3. *Unless $\mathsf{NP} = \mathsf{RP}$, there is no polynomial-time algorithm which, given access to a deterministic MDP M with at least two actions and horizon $H = O(d)$, where the optimal value functions Q^* and V^* are linear in d-dimensional features ϕ (Definition 3.1), outputs a policy π satisfying $V^\pi > V^* - 1/4$ with constant probability.*

We construct such MDPs from 3-SAT formulas in a way that an efficient algorithm for solving the MDP would yield an efficient algorithm for solving 3-SAT.

4.2 Linear Infinite-Horizon MDP

To build intuition, we begin with an infinite-horizon deterministic MDP derived from a CNF formula. Consider the example:

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$

We define an infinite-horizon MDP (whose unrolling is illustrated in Figure 4.1) as follows:

- Each state corresponds to a partial assignment $w \in \{0,1\}^v$ of the SAT variables. The root corresponds to the all-zero assignment.
- The process terminates when the agent reaches a pre-decided satisfying assignment w^*.
- From any state w, an unsatisfied clause is selected, and the agent chooses among three actions: flipping one of the three variables in that clause.
- Each action incurs a reward of -1.

Because each action has a reward of -1, the optimal policy minimizes the path length to the satisfying assignment w^*. This leads to the following:

Lemma 4.4. *The optimal value functions Q^* and V^* are linear in w and w^*. Specifically, for a state s with assignment w:*

$$V^*(s) = -D(w, w^*),$$

where $D(w, w^*)$ is the Hamming distance between w and w^*. Since $D(w, w^*) = \frac{1}{2}(v - \langle w, w^* \rangle)$, it is linear in both w and w^*. Moreover, $Q^*(s, a) = V^*(T(s, a)) - 1$.

While conceptually clean, this construction is infinite-horizon. Truncating the tree to make it finite-horizon would require inserting leaf rewards that depend on w^*, which cannot be simulated efficiently.

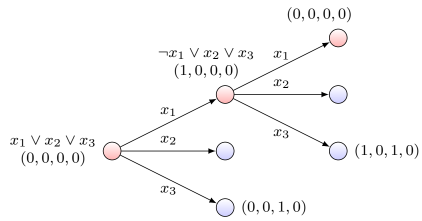

Figure 4.1: Unrolling of an infinite-horizon MDP based on SAT clauses. Each node corresponds to an assignment; actions flip variables in an unsatisfied clause.

4.3 Linear Finite-Horizon MDP

To address this, we define a finite-horizon MDP with carefully constructed leaf rewards. Rewards are zero everywhere except at the leaves. At a leaf with assignment w and depth l, the expected reward is:

$$\mathbb{E}[R(w,l)] = \left(1 - \frac{l + D(w,w^*)}{H+v}\right)^r.$$

This form ensures that the optimal value function has the same structure:

Lemma 4.5. *The optimal value function at a state s with assignment w and depth l is*

$$V^*(s) = \left(1 - \frac{l + D(w,w^*)}{H+v}\right)^r.$$

This function is a degree-r polynomial in the inner product $\langle w, w^*\rangle$, and thus linear in $d = v^r$-dimensional features.

Sketch. See [11] for details. The greedy policy that flips variables to reduce $D(w,w^*)$ by 1 per step (and thus increases l by 1) maintains $D(w,w^*) + l$ as a constant. Since the reward is decreasing in this quantity, the greedy policy is optimal, and the value function inherits the same functional form as the reward. □

Suppose we truncate the tree at depth $H = d = v^r$. Then the maximum reward at the final level is:

$$\left(1 - \frac{H}{H+v}\right)^r = \left(\frac{v}{H+v}\right)^r = v^{-O(r^2)}.$$

Since any algorithm restricted to polynomial time in d and H can reach only $\mathrm{poly}(v^r)$ states, it will observe negligible reward at the leaves and gain no useful information.

In conclusion, while statistically efficient algorithms exist under the assumption that V^* and Q^* are linear in known features, computationally efficient algorithms do not—unless $\mathsf{NP} = \mathsf{RP}$.

Bibliography

[1] Farama Foundation. Cartpole – gymnasium classic control environment, 2023. `https://gymnasium.farama.org/environments/classic_control/cart_pole/`.

[2] Farama Foundation. Pong – ale atari environment, 2023. `https://ale.farama.org/environments/pong/`.

[3] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

[5] Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. AAAI Press, 2005.

[6] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

[7] Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. *Advances in Neural Information Processing Systems*, 26, 2013.

[8] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

[9] Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

[10] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2256–2264, Red Hook, NY, USA, 2013. Curran Associates Inc.

[11] Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pages 1282–1302. PMLR, 2022.