Algorithms and Uncertainty, Summer 2021 Lecture 16 (4 pages)

Markov Decision Processes with Infinite Time Horizon

Thomas Kesselheim Last Update: June 10, 2021

After having seen many examples of a Markov decision process with a finite time horizon, we will turn today to infinite time horizons. That is, one considers an eternal process but future rewards are less valuable than current ones. Such processes play a very important role in machine learning in the context of reinforcement learning.

1 Model

We again have a Markov decision process, defined by states $\mathcal{S}$, actions $\mathcal{A}$, rewards $r_a(s)$, and state transition probabilities $p_a(s, s')$.

We start from a state $s_0 \in \mathcal{S}$. A policy π is again a function, which defines which action $\pi(s_0, \dots, s_{t-1}) \in \mathcal{A}$ to take in step t when the states so far have been $s_0, \dots, s_{t-1}$. So, again a random sequence of states $s_0^\pi, s_1^\pi, \dots$ and actions $a_0^\pi, a_1^\pi, \dots$ evolves.[1]

Given a discount factor γ, $0 < \gamma < 1$, the expected reward of policy π when starting at s_0 is

$$V(\pi, s_0) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{a_t^\pi}(s_t^\pi)\right] \ .$$

One motivation for this discounted reward is a less strict time horizon. After each step, we toss a biased coin. If it comes up heads (probability γ), we continue, if it comes up tails (probability $1-\gamma$), we stop right here.

We can use the same arguments as for finite time horizons to see that the optimal policy only depends on the current state. For such a Markovian policy, we have

$$V(\pi, s) = r_{\pi(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi(s)}(s, s') \cdot V(\pi, s') \ .$$

Naturally, defining $V^*(s) = \max_\pi V(\pi, s)$, we have

$$V^*(s) = \max_{a \in \mathcal{A}} \left(r_a(s) + \gamma \sum_{s' \in \mathcal{S}} p_a(s, s') \cdot V^*(s') \right) \ .$$

This equation is called *Bellman equation*.

2 Computing Optimal Policies via Linear Programming

The key observation to derive (approximations of) optimal policies is that we only need to know (an approximation of) the vector $V^*(s)$.

Lemma 16.1. *Let $(W_s)_{s \in \mathcal{S}}$ be a vector such that $|W_s - V^*(s)| \le \epsilon$ for all s for an $\epsilon \ge 0$, then the policy π that in state s chooses the action a that maximizes $r_a(s) + \gamma \sum_{s' \in \mathcal{S}} p_a(s, s') \cdot W_{s'}$ fulfills $V(\pi, s) \ge V^*(s) - \frac{2\epsilon}{1-\gamma}$ for all s.*

Note that in particular the case $\epsilon = 0$ tells us that the policy π will be optimal if $W_s = V^*(s)$.

[1] Note that we start indexing the sequences at 0.

Proof. Note that W_s approximates the expected reward of an optimal policy π^* starting from s. Our policy might be suboptimal; its expected reward is $V(\pi, s)$. We will show that nonetheless, W_s is also a good approximation.

To this end, let $\hat{s}$ be the state s for which $W_s - V(\pi, s)$ is largest. Let $\delta = W_{\hat{s}} - V(\pi, \hat{s})$. We will show that $\delta \leq \frac{1+\gamma}{1-\gamma}\epsilon$. This then implies that for every state s we have $V(\pi, s) \geq W_s - \delta \geq V^*(s) - \epsilon - \delta \geq V^*(s) - \frac{2\epsilon}{1-\gamma}$. In order to derive the desired bound for δ, let us consider the following.

$$\begin{aligned} V(\pi, \hat{s}) &= r_{\pi(\hat{s})}(\hat{s}) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi(\hat{s})}(\hat{s}, s') \cdot V(\pi, s') \\ &\geq r_{\pi(\hat{s})}(\hat{s}) + \gamma \left(\sum_{s' \in \mathcal{S}} p_{\pi(\hat{s})}(\hat{s}, s') \cdot W_{s'} - \delta \right) \end{aligned}$$

Observe that our choice of $\pi(\cdot)$ ensures that $\pi(\hat{s})$ maximizes $r_{\pi(\hat{s})}(\hat{s}) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi(\hat{s})}(\hat{s}, s') \cdot W_{s'}$. Hence, we can replace $\pi(\hat{s})$ by the optimal policy $\pi^*(\hat{s})$ and only decrease the value of the expression. In addition, using that W_s and $V^*(s)$ can only differ by at most ϵ, we get:

$$\begin{aligned} V(\pi, \hat{s}) &\geq r_{\pi^*(\hat{s})}(\hat{s}) + \gamma \left(\sum_{s' \in \mathcal{S}} p_{\pi^*(\hat{s})}(\hat{s}, s') \cdot W_{s'} - \delta \right) \\ &\geq r_{\pi^*(\hat{s})}(\hat{s}) + \gamma \left(\sum_{s' \in \mathcal{S}} p_{\pi^*(\hat{s})}(\hat{s}, s') \cdot V^*(s') - \epsilon - \delta \right) \\ &= \left(r_{\pi^*(\hat{s})}(\hat{s}) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi^*(\hat{s})}(\hat{s}, s') \cdot V^*(s') \right) - \gamma(\epsilon + \delta) \\ &= V^*(\hat{s}) - \gamma(\epsilon + \delta) \ . \end{aligned}$$

We also have $V(\pi, \hat{s}) = W_{\hat{s}} - \delta \leq V^*(\hat{s}) + \epsilon - \delta$ by the definition of δ. In combination, this gives us $\epsilon - \delta \geq -\gamma(\epsilon + \delta)$ and therefore $\delta \leq \frac{1+\gamma}{1-\gamma}\epsilon$. □

So, all we would need to know are the values of $V^*(s)$. Unfortunately, unlike in the finite horizon case, there is no simple base of the recursion. Therefore, computing them is more complicated here.

One way is by linear programming: We treat the entries $V^*(s)$ as variables, which have to fulfill the Bellman equations. More precisely, the LP reads

$$\begin{aligned} &\text{minimize} \sum_{s \in \mathcal{S}} V^*(s) \\ &\text{subject to } r_a(s) + \gamma \sum_{s' \in \mathcal{S}} p_a(s, s') \cdot V^*(s') \leq V^*(s) \qquad \text{for all } s \in \mathcal{S},\ a \in \mathcal{A} \end{aligned}$$

Note that the constraints actually only require that the left-hand side of each Bellman equation is at least as large as the respective right-hand side. The objective function ensures that an optimal solution to this LP fulfills them indeed with equality: If for any s, there is some slack with respect to all a, one can reduce $V^*(s)$ by the smallest slack and improve the solution.

3 Value Iteration

In usual applications, solving the LP is too slow and not necessary. One can find an approximate solution vector much faster using algorithms, which iteratively improve the solution.

Given a vector $(W_s)_{s\in\mathcal{S}}$, let $T(W)$ be the vector defined by

$$(T(W))_s = \max_{a\in\mathcal{A}} \left(r_a(s) + \gamma \sum_{s'\in\mathcal{S}} p_a(s,s') \cdot W_{s'} \right) \ .$$

The vector V^* is a fixed point of the function T, called the *Bellman operator*. In order to find V^*, we therefore repeatedly apply function T, starting from an arbitrary vector $W^{(0)}$. This method is called *value iteration*.

Theorem 16.2. *For any starting point $W^{(0)}$, the sequence $W^{(0)}, W^{(1)}, \ldots$ defined by value iteration converges to V^*. More precisely, for every $\epsilon > 0$, there is a $t_0 \in \mathbb{N}$ such that for all $t \geq t_0$ we have $|W_s^{(t)} - V^*(s)| < \epsilon$.*

For two vectors W, W', define the distance $d(W,W') = \|W - W'\|_\infty$. So, it is the maximum amount that the two vectors differ by in one component.

Lemma 16.3. *For any vectors W and W', we have $d(T(W), T(W')) \leq \gamma d(W,W')$.*

Proof. To this end, consider any component $s \in \mathcal{S}$. We have to show that $|(T(W))_s - (T(W'))_s| \leq \gamma d(W,W')$.

Let $a^* \in \mathcal{A}$ be an action attaining the maximum in the definition of $T(W)_s$. That is, we have

$$T(W)_s = r_{a^*}(s) + \gamma \sum_{s'\in\mathcal{S}} p_{a^*}(s,s') \cdot W_{s'}$$

The action a^* might not be the optimal choice for $T(W')_s$ but it is a feasible one, so

$$T(W')_s \geq r_{a^*}(s) + \gamma \sum_{s'\in\mathcal{S}} p_{a^*}(s,s') \cdot W'_{s'}$$

In combination:

$$T(W)_s - T(W')_s \leq \gamma \sum_{s'\in\mathcal{S}} p_{a^*}(s,s') \cdot (W_{s'} - W'_{s'}) \ .$$

For any $s' \in \mathcal{S}$, we have $W_{s'} - W'_{s'} \leq \max_{s''\in\mathcal{S}} |W_{s''} - W'_{s''}| = d(W,W')$, so

$$T(W)_s - T(W')_s \leq \gamma \sum_{s'\in\mathcal{S}} p_{a^*}(s,s') \cdot d(W,W') = \gamma d(W,W') \ ,$$

because the probabilities sum up to 1.

The same argument holds if we swap the roles of W and W'. Therefore $|(T(W))_s - (T(W'))_s| \leq \gamma d(W,W')$. □

Now, we can continue to the proof of Theorem 16.2.

Proof of Theorem 16.2. By Lemma 16.3, we know that $d(W^{(t)}, V^*) \leq \gamma^t d(W^{(0)}, V^*)$. As $d(W^{(0)}, V^*)$ is finite and independent of t, for each $\epsilon > 0$ there has to be a t_0 such that $\gamma^t d(W^{(0)}, V^*) < \epsilon$ for $t \geq t_0$. □

4 Policy Iteration

An alternative to value iteration is *policy iteration*. We start from an arbitrary policy $\pi^{(0)}$ and improve it iteratively in a sequence $\pi^{(1)}, \pi^{(2)}, \dots$ until in one iteration the policy does not change.

Given policy $\pi^{(t)}$, we can compute an improved policy as follows. First compute all values $V(\pi^{(t)}, s)$ by solving a system of linear equations. Now set $\pi^{(t+1)}(s)$ to the action a that maximizes $r_a(s) + \gamma \sum_{s' \in \mathcal{S}} p_a(s, s') \cdot V(\pi^{(t)}, s')$. Note that this quantity is actually the expected reward of a different, non-Markovian policy, namely the one that starts from state s by choosing action a and chooses actions according to $\pi^{(t)}$ afterwards.

Theorem 16.4. *Policy iteration converges in finitely many steps to an optimal policy.*

Proof. Note that if $\pi^{(t+1)} = \pi^{(t)}$, then this policy fulfills the Bellman equation. Therefore, any fixed point is an optimal policy.

It remains to prove that the sequence converges. Because there are only finitely many Markovian policies, the only way it could possibly not converge is a cycle. We show that there is no cycle in the iteration by showing that $V(\pi^{(t+1)}, s) \geq V(\pi^{(t)}, s)$ for all t and all $s \in \mathcal{S}$. Note that if $V(\pi^{(t+1)}, s) = V(\pi^{(t)}, s)$ for all s, then we have found a fixed point.

So, let us fix t and show that $V(\pi^{(t+1)}, s) \geq V(\pi^{(t)}, s)$ for all $s \in \mathcal{S}$. To this end, define an auxiliary sequence of policies $\pi'_0, \pi'_1, \dots$. We define π'_i as the policy that in the first i steps uses $\pi^{(t+1)}$ and then afterwards uses $\pi^{(t)}$. By this definition $V(\pi^{(t)}, s) = V(\pi'_0, s)$ and $V(\pi^{(t+1)}, s) = \lim_{i \to \infty} V(\pi'_i, s)$. It is therefore enough to show that

$$V(\pi'_i, s) \geq V(\pi'_{i-1}, s) \qquad \text{for all } i \in \mathbb{N} \text{ and all } s \in \mathcal{S} \ .$$

We show this claim by induction on i. The base case is $i = 1$. For this case, we have

$$V(\pi'_0, s) = r_{\pi^{(t)}(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi^{(t)}(s)}(s, s') V(\pi^{(t)}, s')$$

and

$$V(\pi'_1, s) = r_{\pi^{(t+1)}(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi^{(t+1)}(s)}(s, s') V(\pi^{(t)}, s') \ ,$$

because policy π'_1 does the first step according to $\pi^{(t+1)}$ and then uses $\pi^{(t)}$. Our definition of policy iteration was exactly that $\pi^{(t+1)}(s)$ maximizes this expression. Therefore, the claim holds.

For $i > 1$, we have

$$V(\pi'_{i-1}, s) = r_{\pi^{(t+1)}(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi^{(t+1)}(s)}(s, s') V(\pi'_{i-2}, s')$$

and

$$V(\pi'_i, s) = r_{\pi^{(t+1)}(s)}(s) + \gamma \sum_{s' \in \mathcal{S}} p_{\pi^{(t+1)}(s)}(s, s') V(\pi'_{i-1}, s') \ .$$

By induction hypothesis, we know that $V(\pi'_{i-2}, s') \leq V(\pi'_{i-1}, s')$ for all $s' \in \mathcal{S}$. So, this immediately implies that $V(\pi'_{i-1}, s) \leq V(\pi'_i, s)$ because every term in the expression for $V(\pi'_i, s)$ is at least as large as the respective term in the expression for $V(\pi'_{i-1}, s)$. □