Algorithms and Uncertainty, Winter 2017/18 Lecture 10 (4 pages)

Markov Decision Processes

Instructor: Thomas Kesselheim

As a motivation, consider the following game: There are n envelopes. Envelope i contains prize v_i with probability p_i. With probability $1 - p_i$ it is empty. You make open envelopes and keep the prizes as long as you do not open an empty envelope.

What is the best strategy to play this game? One might be tempted to act myopically: Open the envelope of highest expected reward $p_i \cdot v_i$. This is, for example, a bad idea in the following setting:

$$v_1 = 1000, \quad p_1 = \frac{1}{100}, \qquad v_i = p_i = 1 \text{ for } i > 1 \ .$$

Here, we would open envelope 1 first but with 99 % chance, we do not get anything. It is much better to first open envelopes $2, \dots, n$ and only then to take the chance and open envelope 1.

Today's goal will be to introduce a general model for such stochastic decision problems, to describe optimal policies and give algorithms to compute them.

1 Markov Decision Processes

A Markov Decision Process is defined by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function that defines a reward $r_a(s)$ for taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and a random transition function, which is defined by probabilities $p_a(s, s')$: If we are in state $s \in \mathcal{S}$ and we take action $a \in \mathcal{A}$, then we move on to state $s' \in \mathcal{A}$ with probability $p_a(s, s') \in [0, 1]$.

The process works as follows. We start from state $s_0 \in \mathcal{S}$, choose one action $a \in \mathcal{A}$. We immediately get reward $r_a(s) \in \mathbb{R}$ and then continue to a random state s', which is given by the probability distribution $p_a(s_0, \cdot)$. This way, a sequence $s_0, s_1, \dots$ evolves. We move from s_t to s_{t+1} by the probability distribution $p_a(s_t, \cdot)$. So, the probabilities only depend on the current state and the current action but not on which states we have seen. This makes the process *Markovian*.

Generally, rewards may also be random, just as in our example above. To capture this, set $r_a(s)$ to the *expected* reward that you get when taking action a in state s.

On the one hand, this generalizes a deterministic finite automaton. Here, for each a and s, there is exactly one s' for which $p_a(s, s') = 1$ and $p_a(s, s') = 0$ otherwise. On the other hand, it is also a generalization of a Markov chain. Here $\mathcal{A}$ has only one element (an action like "continue") and the we move through states without having a real choice.

A *policy* π assigns to each sequences of states $s_0, \dots, s_{t-1} \in \mathcal{S}$ an action $\pi(s_0, \dots, s_{t-1}) \in \mathcal{A}$. So, if we run policy π starting from s_0, we pass through random sequence of states $s_0^\pi, s_1^\pi, \dots$, a random sequence of actions $a_0^\pi, a_1^\pi, \dots$.

Generally, we can move through a Markov decision process for unbounded time. Then, of course, the sum of reward is unbounded, too. Therefore, one considers two different models of time horizons.

2 Finite Time Horizon

When dealing with a finite time horizon, there is some T such that we do not care what happens after time T. In this case, we can write the expected reward of policy π when starting at s_0 as

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

We also define $V^*(s_0, T)$ as the highest expected reward that one can achieve starting from s_0 in T steps, that is, $V^*(s_0, T) = \max_{\text{policy } \pi} V(\pi, s_0, T)$. (Note that there are only finitely many histories and therefore only finitely many different policies, so the maximum is well-defined.)

Consider an optimal policy π, that is $V(\pi, s_0, T) = V^*(s_0, T)$. As a_0^π is deterministic, we might as well write

$$V(\pi, s_0, T) = r_{a_0^\pi}(s_0) + \mathbf{E}\left[\sum_{t=1}^{T} r_{a_t^\pi}(s_t^\pi)\right] = r_{a_0^\pi}(s_0) + \sum_{s' \in \mathcal{S}} p_{a_0^\pi}(s_0, s') \mathbf{E}\left[\sum_{t=1}^{T} r_{a_t^\pi}(s_t^\pi) \,\middle|\, s_1^\pi = s'\right] \ .$$

Let us inspect the expectation on the right-hand side. We claim that

$$\mathbf{E}\left[\sum_{t=1}^{T} r_{a_t^\pi}(s_t^\pi) \,\middle|\, s_1^\pi = s'\right] = V^*(s', T-1) \ .$$

The reason is simple: Both is the maximum expected reward that we would receive from a Markov decision process running for $T-1$ steps, starting from s. On the left-hand side, we actually start from s_0 but this does not make a difference for the remaining steps. Importantly, rewards in the current step only depend on the current state and action, not on the past ones.

We skip the fleshed out formal argument here. One possible way is to assume that either side is strictly larger than the other and observe that one could either add or remove s_0 from the beginning of the history.

Consequently, we can define $V^*(s, T)$ recursively as

$$V^*(s, T) = \max_{a \in \mathcal{A}} \left(r_a(s) + \sum_{s' \in \mathcal{S}} p_a(s, s') V^*(s', T-1) \right) \ .$$

These observations directly lead to the following theorem:

Theorem 10.1. *For finite time horizons, there is an optimal policy that is* Markovian. *That is, actions only depend on the current state and the number of remaining steps. An optimal policy for a time horizon of T steps can be computed in time $O(T \cdot |\mathcal{S}|^2 \cdot |\mathcal{A}|)$.*

Proof. We can compute an optimal policy by dynamic programming. We have to compute $T \cdot |\mathcal{S}|$ values of V^* in total, each computation takes $|\mathcal{S}| \cdot |\mathcal{A}|$ steps. By tracing back the generation of V^*, we get a policy that is Markovian. □

3 Discounted Rewards

Instead of considering a finite time horizon it is at least as common to consider an eternal process but future rewards are less valuable than current ones. 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 tales (probability $1-\gamma$), we stop right here.

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

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

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

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

This equation is called *Bellman equation*.

Observe that if we know the vector $V^*(s)$, then we could reconstruct the optimal policy. Unfortunately, we don't and unlike in the finite horizon case, there is no simple base of the recursion. However, there is simple way to approximately find such a vector

Given a vector $(W_s)_{s \in \mathcal{S}}$, $W_s \geq 0$, let $T(W)$ be the vector defined by

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

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. This method is called *value iteration*.

Theorem 10.2. *Value iteration is well-defined, i.e., it converges to the unique fixed point of T.*

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 10.3. *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

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

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

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

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 10.2.

Proof of Theorem 10.2. Let V^* be the fixed point of T that is induced by the optimal policy. Let V^{**} be any other fixed point. Then, we have $d(V^*, V^{**}) = d(T(V^*), T(V^{**})) \leq \gamma \cdot d(V^*, V^{**})$. As $\gamma \in (0, 1)$, this means that $d(V^*, V^{**}) = 0$. So the two fixed points have to be identical.

Furthermore, starting from any $W^{(0)}$, 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, the sequence has to converge on V^*. □