Solving the inverted pendulum as a Markov decision process

1 Markov decision processes

A Markov decision process (MDP) is defined by a 5-tuple $(\mathcal{S}, \mathcal{A}, P.(\cdot, \cdot), R.(\cdot, \cdot), \gamma)$ where

(i) $\mathcal{S}$ is a finite set of states;

(ii) $\mathcal{A}$ is a finite set of actions;

(iii) $P_a(s, s')$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$;

(iv) $R_a(s, s')$ is the immediate reward received after transitioning from state s to state s' due to action a; and

(v) $\gamma \in [0, 1]$ is a discount factor which represents the difference in importance between future rewards and present rewards.

The problem is to find a policy for a decision-making agent that navigates the states of this world via actions. The policy is a function π that specifies the action $\pi(s)$ that the agent will choose when in state s. Our goal is to choose a policy that will maximize a cumulative function of the random rewards; in particular, we wish to maximize the discounted sum of expected rewards over a potentially infinite horizon:

$$\sum_{t=0}^{\infty} \gamma^t \mathbb{E}[R_{a_t}(s_t, s_{t+1})] \quad \text{where } a_t = \pi(s_t). \tag{1}$$

1.1 Policies and value functions

Imagine that we have some magical function $V(\cdot)$ which tells us the inherent value of a state. It accepts a state as its argument, and its output value tells us *how characteristically good* it is to be in that state for the purposes of achieving our goal (that is, maximizing reward). A bit philosophical, eh?

If we had this function, then choosing a policy would be trivial: the optimal policy is one which always selects the action that maximizes the expectation of $V(s')$ based on the present state s. The expectation of $V(s')$ under a given action a in state s is a sum

of $V(s')$ over all possible s', with each term weighted by the probability of transitioning from s to s' due to action a:

$$\mathbb{E}[V(s')] = \sum_{s' \in \mathcal{S}} P_a(s, s')V(s'). \tag{2}$$

So the optimal policy is

$$\pi^*(s) = \arg\max_a \mathbb{E}[V(s')] = \arg\max_a \sum_{s' \in \mathcal{S}} P_a(s, s')V(s'). \tag{3}$$

But, the inherent value of a state $V(\cdot)$ depends on what policy we are following. In particular, it is the sum of the expected reward over all future times if we act according to a policy π. Intuitively, there are two factors which influence the value of state s:

(i) the immediate expected reward of acting on our policy in state s and

(ii) the expected value of the state which follows from acting on our policy in state s.

Note that the second of these is defined in terms of the expected value of the state which follows. Thus, our description of the value of state s includes the value of state s', and in this way recursively captures something about the value of all future states.

Let's work out each of the two factors above, starting with the first piece. The expected reward of taking action a in a given state s is a sum of $R_a(s, s')$ over all possible s', with each term weighted by the probability of transitioning from s to s' due to action a:

$$\mathbb{E}[R_a(s)] = \sum_{s' \in \mathcal{S}} P_a(s, s')R_a(s, s'). \tag{4}$$

The action a is selected by policy $\pi(s)$, so we can rewrite our equation above as the expected reward of acting under policy π in a given state s:

$$\mathbb{E}[R_{\pi(s)}(s)] = \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s')R_{\pi(s)}(s, s'). \tag{5}$$

This is our expression for the immediate expected reward of acting on our policy in state s (the first of the two factors mentioned above). But the value of state s depends not only on this immediately available reward; it also depends on the expected value of the state which follows from acting on our policy (the second of the two factors above):

$$\mathbb{E}[V(s')] = \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s')V(s'). \tag{6}$$

The full value of state s is a sum of equations (5) and (6), with the second weighted by our discount factor to reflect the fact that this is the value of a state (s') which is one step ahead in the future from the state (s) whose value we are presently considering. So, to get the value of state s under policy π, we sum the expectation of the immediate reward for acting on our policy from state s and the future-discounted expectation of the value of the state s' which follows from acting on our policy in state s:

$$V(s) = \mathbb{E}[R_{\pi(s)}(s)] + \gamma \mathbb{E}[V(s')] \tag{7}$$

$$= \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s') R_{\pi(s)}(s, s') + \gamma \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s') V(s') \tag{8}$$

$$= \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s') \Big(R_{\pi(s)}(s, s') + \gamma V(s') \Big) \tag{9}$$

Let's take stock. We have an expression for the optimal policy with respect to any given value function, and we have an expression for a value function under a given policy. To reiterate:

$$\pi^*(s) = \arg\max_a \mathbb{E}[V(s')] = \arg\max_a \sum_{s' \in \mathcal{S}} P_a(s, s') V(s') \tag{10}$$

$$V(s) = \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s') \Big(R_{\pi(s)}(s, s') + \gamma V(s') \Big) \tag{11}$$

1.2   The Bellman optimality equations

But now we have a bit of a conundrum. We know the optimal policy with respect to a given value function, but our value function itself depends on what policy we're acting under! How do we find, at once, the optimal value function and its optimal policy?

There exists a large space of possible value functions due to all possible policies we could use. However, consider for a moment that we are handed from on high the optimal policy for our task. Under this policy, the output of the value function will be greater than or equal to the output of the value function under any other policy. This is because under the optimal policy, the value of any given state is always maximized. So, the maximum of all possible value functions for a state coincides with the value function we have when acting under an optimal policy.

With this in mind, we can readily define a notion of the optimal value function. The optimal value function for a state must equal the value function for that state under the best possible action for that state:

$$V^*(s) = \max_a V(s). \tag{12}$$

Note that in this case, the action a is the optimal policy $\pi^*(s)$. This enables us to write out $V(s)$ with no reference to $\pi(s)$ at all, instead substituting $\pi(s) = a$:

$$V^*(s) = \max_a \sum_{s' \in \mathcal{S}} P_{\pi(s)}(s, s') \Big(R_{\pi(s)}(s, s') + \gamma V^*(s') \Big). \tag{13}$$

This is a set of $|\mathcal{S}|$ nonlinear recurrence relations in $|\mathcal{S}|$ variables[1], each one requiring a maximum over $|\mathcal{A}|$ expressions ($|\mathcal{A}|$ and $|\mathcal{S}|$ are the number of elements in the sets $\mathcal{A}$ and $\mathcal{S}$, respectively). These are called the *Bellman equations* for the optimal value function.

Intuitively, solving each one amounts to finding the value of trying each available action in a state, and then selecting the value which corresponds to the best action in that state (the value of that state under an optimal policy). Solving them all simultaneously amounts to finding the value of all states at once when acting under an optimal policy, yielding the optimal value function.

Once we have $V^*(s)$, we can substitute this back into our expression for $\pi^*(s)$:

$$\pi^*(s) = \arg\max_a \sum_{s' \in \mathcal{S}} P_a(s, s') V^*(s'). \tag{14}$$

This is an optimal policy, and our solution to the MDP.

2 The inverted pendulum problem as an MDP

To set up the inverted pendulum problem as an MDP, we need to define for our agent

  (i) a world of available states,

 (ii) a set of actions it can take,

(iii) the transition probabilities which describe the physical dynamics of this world,

(iv) a reward function which incentivizes remaining balanced upright, and

 (v) a discount factor which reflects how long- or short-sighted our agent is.

2.1 The agent's world

First we will build the world of our agent by choosing (i) and (ii): the states it can visit and the actions it can take. The computational complexity of solving the Bellman equations explodes rapidly as the combinations of possible states and actions grow. So, our choices of (i) and (ii) ultimately depend on how long we are willing to wait, and unfortunately it doesn't take much to have to wait an astronomically period time. Since we're not willing to wait the age of the universe, our state and action spaces will have to be pretty small indeed.

Let's begin with a state space for our agent. The state of the pendulum-cart system can be fully characterized by four variables: angular position of the pendulum, angular velocity of the pendulum, position of the cart, and velocity of the cart. In order to avoid our state space becoming overly large, we will have to discretize each of these continuous variables into distressingly course-grained substates.

---

[1]Such systems generally cannot be characterized analytically and lack a closed-form solution. In fact, solving nonlinear recurrence relations is often viewed as a hopeless endeavor: to solve *every* nonlinear recurrence relation would imply that one could solve the Halting problem, as one could encode a program as initial states and the workings of the Turing machine as the recurrence relations (see this Stack Exchange discussion by Rex Kerr). But these equations can be solved numerically (at least in principle), and with sufficiently small action and state spaces this computation is tractable. This where we're headed.
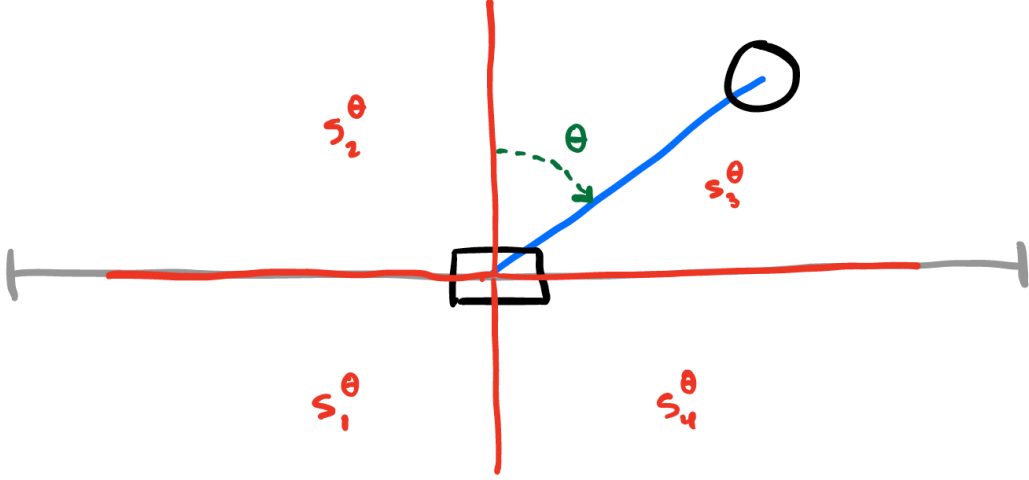
Figure 1: Visualization of the substate space for angular position.

First let's consider the pendulum. For the angular position subspace $\mathcal{S}^\theta$, we choose the four substates

$$
s^\theta = \begin{cases}
s_1^\theta & : -\pi \le \theta < -\frac{\pi}{2} \\
s_2^\theta & : -\frac{\pi}{2} \le \theta < 0 \\
s_3^\theta & : 0 \le \theta < \frac{\pi}{2} \\
s_4^\theta & : \frac{\pi}{2} \le \theta < \pi
\end{cases} \quad . \tag{15}
$$

For the angular velocity subspace $\mathcal{S}^\omega$, we similarly choose four substates

$$
s^\omega = \begin{cases}
s_1^\omega & : \omega < -\alpha \\
s_2^\omega & : -\alpha \le \omega < 0 \\
s_3^\omega & : 0 \le \omega < \alpha \\
s_4^\omega & : \alpha \le \omega
\end{cases} \tag{16}
$$

where α is a parameter deciding the 'width' of our state bins.
Now on to the cart. For its position subspace $\mathcal{S}^x$, we choose the four substates

$$
s^x = \begin{cases}
s_1^x & : x < -\beta \\
s_2^x & : -\beta \le x < 0 \\
s_3^x & : 0 \le x < \beta \\
s_4^x & : \beta \le x
\end{cases} \tag{17}
$$

and for its velocity subspace $\mathcal{S}^v$ we choose four substates

$$
s^v = \begin{cases}
s_1^v & : v < -\eta \\
s_2^v & : -\eta \le v < 0 \\
s_3^v & : 0 \le v < \eta \\
s_4^v & : \eta \le v
\end{cases} \tag{18}
$$

5

where β and η are again parameters deciding the 'width' of the state bins.

The full state space $\mathcal{S}$ is $\mathcal{S}^\theta \times \mathcal{S}^\omega \times \mathcal{S}^x \times \mathcal{S}^v$, or the set of all ordered pairs $(s^\theta, s^\omega, s^x, s^v)$. By our choices above, this space has 256 states.

Our agent acts by applying horizontal forces to the cart. We'll allow our agent to choose among five actions, each applying a different, discrete-valued force to the cart:

$$\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\} \text{ where } \begin{cases} a_1 & : f = -\kappa_2 \\ a_2 & : f = -\kappa_1 \\ a_3 & : f = 0 \\ a_4 & : f = \kappa_1 \\ a_5 & : f = \kappa_2 \end{cases}. \tag{19}$$

κ_1 and κ_2 are parameters deciding the force of the action.

2.2 How the world works

Now our agent needs a model of the physical dynamics of its world and how its actions affect those dynamics. This model comes in the form of a set of transition probabilities $P_a(s, s')$—one for every possible action a—describing the probability of transitioning from state s to state s' due to that action.

In our case the transition probabilities represent the physics of the pendulum-cart system. However, forming these transition probabilities is not totally straightforward: the states which describe our system each contain a wide, continuous range of actual physical configurations, leaving a great range of physical variability over which state will proceed from which. So, we will produce the transition probabilities via a Monte Carlo estimate of randomly simulated dynamics. For each state a and state s, we'll randomly sample an angular position and velocity uniformly from within the state s, and run the simulation one step forward to end up in state s_{next}. We'll repeat this a large number of times N and estimate the transition probabilities as

$$P_a(s, s') = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(s_{next} = s'). \tag{20}$$

Since the most extreme states for velocity and angular velocity cover an unbounded range of physical possibilities (off to velocities of positive and negative ∞), we'll just have to choose physically realistic boundaries for the sake of sampling.

This is going to be an inherently flawed estimate. We're estimating the dynamics by sampling uniformly from the continuum of available positions and velocities within each state; yet in reality, there's no reason the physical system should visit all possible (continuous) positions and velocities within a state with equal probability. Nonetheless, let's consider this satisfactory, reasoning that our estimate of the transition probabilities should encode no finer-grained statistical assumptions about our system than our agent's representation of its state allows.[2]

---

[2]Note that if we *could* choose arbitrarily fine-grained states, this issue would go away: as the state "slices" grew thinner, the physical dynamics of the system would become far less variable within each

2.3 What the agent wants

Next we need to construct a reward function that incentivizes our agent to do the right thing. We would like our agent to balance the pendulum and remain near the center of the track. This means there are two things we wish to avoid:

(i) allowing the pendulum to fall below the level of the track and

(ii) moving the cart too close to the ends of the track.

So, we will let the reward function equal -1 for any transition into a state that satisfies either of these conditions, and 0 otherwise:

$$R_a(s, s') = \begin{cases} -1 & : s^{\theta\prime} \in \left\{s_1^\theta, s_4^{\theta\prime}\right\} \text{ or } s^x \in \left\{s_1^{x\prime}, s_4^{x\prime}\right\} \\ 0 & : \text{otherwise} \end{cases} . \tag{21}$$

Finally, we make sure to choose a discount factor γ as well. Having fleshed out $\mathcal{S}$, $\mathcal{A}$, $P.(\cdot, \cdot)$, $R.(\cdot, \cdot)$, and γ, we have fully set up our inverted pendulum problem as an MDP. The rest is up to the Bellman equations.

---

one, and our estimate of their transition probabilities would increasingly-well approximate the dynamics of the physical system. Furthermore, these transition probabilities would become increasingly sparse, reflecting the fact that in a causally deterministic system each state may transition to only one other state with unity probability. But alas, our agent must content itself to inhabit a simple—and unsettlingly noisy—world.