Reinforcement Learning: Value Functions

BIOE 498/598 PJ

Spring 2021

Review

Last time

- RL agents learn by trial and error.
- RL problems are formulated as MDPs.
- Monte Carlo methods can find policies for RL problems.

Review

Last time

- RL agents learn by trial and error.
- RL problems are formulated as MDPs.
- Monte Carlo methods can find policies for RL problems.
- **Today:** What exactly is Monte Carlo learning?

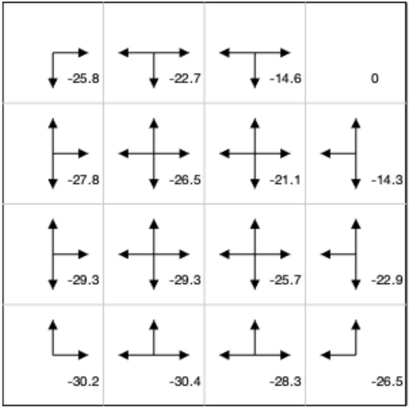
A Monte Carlo approach for Gridworld

- Each grid square is a state.
- Actions: move up, down, left, or right, but the agent cannot leave the grid.
- Reward: -1 for each step.
- Policy: Random.

Starting from a random state, make random moves until the agent reaches the end.

Repeat may times and average the total rewards from each trajectory.

The policy is to move to squares with better Monte Carlo returns.



Value functions

- We are using Monte Carlo to learn a **value function**.
- The value of a state is the expected reward from that state to the end of the trajectory.

$$V(s_i) = \mathbb{E}\left\{\sum_{k=i}^{T} r_k\right\} = \mathbb{E}\{R_i\}$$

  where R_i is the *return* starting at state s_i, i.e. the cumulative reward for the rest of the trajectory: $R_i = r_i + r_{i+1} + \cdots + r_{T-1} + r_T$.

Value functions

- We are using Monte Carlo to learn a **value function**.
- The value of a state is the expected reward from that state to the end of the trajectory.

$$V(s_i) = \mathbb{E}\left\{\sum_{k=i}^{T} r_k\right\} = \mathbb{E}\{R_i\}$$

  where R_i is the *return* starting at state s_i, i.e. the cumulative reward for the rest of the trajectory: $R_i = r_i + r_{i+1} + \cdots + r_{T-1} + r_T$.
- If we know the value function we can derive a policy: Take the action that moves to the state with the highest value.

Trajectories

- A trajectory in an MDP is a sequence of states, actions, and rewards:

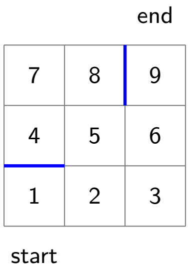
$$s_0, a_0, r_0,\ s_1, a_1, r_1,\ \ldots, s_{T-1}, a_{T-1}, r_{T-1},\ s_T, r_T$$

- The length T can vary for every trajectory.
- There is no action selected in the terminal state s_T, but there can be a terminal reward r_T.
- A reward r_i can be positive (reward), negative (penalty), or zero. Some MDPs only have a nonzero terminal reward!

From trajectories to value functions

Let's calculate $V(s)$ for a 3×3 Gridworld board.

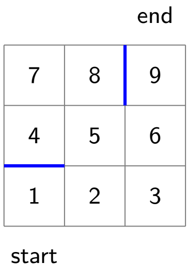
The MDP is deterministic, so knowing s_i and s_{i+1} tells us a_i. Also, $r_i = -1$ for all $0 \leq i < T$.



From trajectories to value functions

Let's calculate $V(s)$ for a 3×3 Gridworld board.

The MDP is deterministic, so knowing s_i and s_{i+1} tells us a_i. Also, $r_i = -1$ for all $0 \leq i < T$.

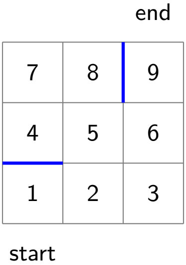


$$
\begin{aligned}
\tau_1 &: \quad 1, 2, 5, 4, 5, 6, 3, 6, 9 & R_{\tau_1} &= -8 \\
\tau_2 &: \quad 1, 2, 3, 6, 3, 2, 5, 8, 7, 8, 5, 6, 9 & R_{\tau_2} &= -12 \\
\tau_3 &: \quad 1, 2, 5, 2, 3, 6, 9 & R_{\tau_3} &= -6 \\
\tau_4 &: \quad 1, 2, 5, 4, 5, 2, 3, 6, 5, 8, 5, 6, 3, 2, 5, 6, 9 & R_{\tau_4} &= -16
\end{aligned}
$$

From trajectories to value functions

Let's calculate $V(s)$ for a 3×3 Gridworld board.

The MDP is deterministic, so knowing s_i and s_{i+1} tells us a_i. Also, $r_i = -1$ for all $0 \leq i < T$.



$$
\begin{aligned}
\tau_1 &: \quad 1, 2, 5, 4, 5, 6, 3, 6, 9 & R_{\tau_1} &= -8 \\
\tau_2 &: \quad 1, 2, 3, 6, 3, 2, 5, 8, 7, 8, 5, 6, 9 & R_{\tau_2} &= -12 \\
\tau_3 &: \quad 1, 2, 5, 2, 3, 6, 9 & R_{\tau_3} &= -6 \\
\tau_4 &: \quad 1, 2, 5, 4, 5, 2, 3, 6, 5, 8, 5, 6, 3, 2, 5, 6, 9 & R_{\tau_4} &= -16
\end{aligned}
$$

$$
V(s_1) \approx \frac{R_{\tau_1} + R_{\tau_2} + R_{\tau_3} + R_{\tau_4}}{4} = \frac{(-8) + (-12) + (-6) + (-16)}{4} = -10.5
$$

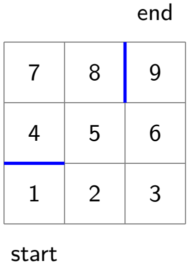
Re-using our trajectories

τ_1 : $1, 2, 5, 4, 5, 6, 3, 6, 9$

τ_2 : $1, 2, 3, 6, 3, 2, 5, 8, 7, 8, 5, 6, 9$

τ_3 : $1, 2, 5, 2, 3, 6, 9$

τ_4 : $1, 2, 5, 4, 5, 2, 3, 6, 5, 8, 5, 6, 3, 2, 5, 6, 9$



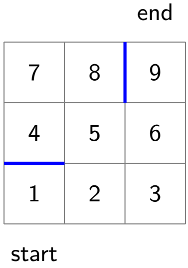
Re-using our trajectories

$\tau_1: \quad 1, 2, 5, 4, 5, 6, 3, 6, 9$

$\tau_2: \quad 1, 2, 3, 6, 3, 2, 5, 8, 7, 8, 5, 6, 9$

$\tau_3: \quad 1, 2, 5, 2, 3, 6, 9$

$\tau_4: \quad 1, 2, 5, 4, 5, 2, 3, 6, 5, 8, 5, 6, 3, 2, 5, 6, 9$



We can estimate $V(s_2)$ using the same trajectories because of the Markov Property. Every visit to s_2 is equivalent to new trajectory that begins at s_2.

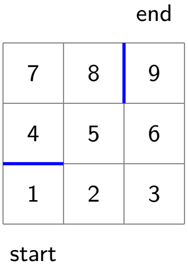
Re-using our trajectories

τ_1 : $1, 2, 5, 4, 5, 6, 3, 6, 9$

τ_2 : $1, 2, 3, 6, 3, 2, 5, 8, 7, 8, 5, 6, 9$

τ_3 : $1, 2, 5, 2, 3, 6, 9$

τ_4 : $1, 2, 5, 4, 5, 2, 3, 6, 5, 8, 5, 6, 3, 2, 5, 6, 9$



We can estimate $V(s_2)$ using the same trajectories because of the Markov Property. Every visit to s_2 is equivalent to new trajectory that begins at s_2.

Some trajectories visit s_2 more than once. For example, τ_3 has two returns $R = -5$ and $R = -3$.

Summary

- RL agents can learn by trial and error.
- MDPs provide a mathematical structure for RL problems.
- The choice of states, actions, and rewards is critical.

Summary

- RL agents can learn by trial and error.
- MDPs provide a mathematical structure for RL problems.
- The choice of states, actions, and rewards is critical.
- **Next time:** What are we learning from our random maze walks?