
1.a

Iteration $i = 0$:

$$V_{\text{opt}}(-2) = 0, \quad V_{\text{opt}}(-1) = 0, \quad V_{\text{opt}}(0) = 0, \quad V_{\text{opt}}(1) = 0, \quad V_{\text{opt}}(2) = 0$$

Iteration $i = 1$:

$$\begin{aligned}
V_{\text{opt}}(-2) &= \max\big(0.8(10+0) + 0.2(-5+0),\ 0.7(10+0) + 0.3(-5+0)\big) \\
&= \max(7,\ 5.5) = 7 \\
V_{\text{opt}}(-1) &= \max(7,\ 5.5) = 7 \\
V_{\text{opt}}(0) &= \max\big(0.8(-5+0) + 0.2(-5+0),\ 0.7(-5+0) + 0.3(-5+0)\big) = -5 \\
V_{\text{opt}}(1) &= \max\big(0.8(-5+0) + 0.2(50+0),\ 0.7(-5+0) + 0.3(50+0)\big) \\
&= \max(6,\ 11.5) = 11.5 \\
V_{\text{opt}}(2) &= 11.5
\end{aligned}$$

Iteration $i = 2$:

$$\begin{aligned}
V_{\text{opt}}(-2) &= \max\big(0.8(10+0) + 0.2(-5+7),\ 0.7(10+0) + 0.3(-5+7)\big) \\
&= \max(8.4,\ 7.6) = 8.4 \\
V_{\text{opt}}(-1) &= \max\big(0.8(10+0) + 0.2(-5-5),\ 0.7(10+0) + 0.3(-5-5)\big) \\
&= \max(6,\ 4) = 6 \\
V_{\text{opt}}(0) &= \max\big(0.8(-5+7) + 0.2(-5+11.5),\ 0.7(-5+7) + 0.3(-5+11.5)\big) \\
&= \max(2.9,\ 3.35) = 3.35 \\
V_{\text{opt}}(1) &= \max\big(0.8(-5-5) + 0.2(50+0),\ 0.7(-5-5) + 0.3(50+0)\big) \\
&= \max(2,\ 8) = 8 \\
V_{\text{opt}}(2) &= \max\big(0.8(-5+11.5) + 0.2(50+0),\ 0.7(-5+11.5) + 0.3(50+0)\big) \\
&= \max(15.2,\ 19.55) = 19.55
\end{aligned}$$

1.b

The optimal policy is:

$$\pi^*(-1) = a_1, \quad \pi^*(0) = a_2, \quad \pi^*(1) = a_2$$

2.a

To convert an MDP with discount factor $\lambda < 1$ into one compatible with a solver that assumes a learning rate of 1, we introduce an absorbing state o.

The absorbing state satisfies:

$$T(o, a, o) = 1, \quad R(o, a, o) = 0$$

For a simple MDP with one state s and one action a, the Bellman equation is:

$$V_{\text{opt}}(s) = T(s, a, s)R(s, a, s) + \lambda V_{\text{opt}}^{t-1}(s)$$

We modify the transition and reward functions as follows:

$$T'(s, a, o) = 1 - \lambda, \quad T'(s, a, s) = \lambda T(s, a, s)$$

$$R'(s, a, o) = 0, \quad R'(s, a, s) = R(s, a, s)$$

This preserves the expected reward while allowing the Bellman update to use a learning rate of 1. The same construction generalizes to larger MDPs.

3.c

The value iteration training plot shows the agent learning to avoid negative-reward states and converging toward regions with positive rewards.

During training, the agent achieves an average of two positive reward states and eventually avoids rewards below -400. However, during evaluation it encounters states with rewards as low as -600, indicating that some transitions were not adequately modeled during training.

This highlights a weakness of value iteration when the MDP model is incomplete or contains inaccurate transition probabilities.

4.d

Tabular Q-learning achieves the optimal policy around iteration 750. Function approximation reaches its peak moving average earlier, around iteration 600.

However, function approximation attains a lower peak reward (approximately -250) compared to tabular Q-learning (approximately -150). This is likely because tabular Q-learning can converge to the true Q-function with sufficient exploration, while function approximation generalizes across states and may smooth out optimal values.

4.e

Function approximation is beneficial in the Mountain Car problem because it generalizes across continuous state spaces and can find a good-enough policy faster than tabular Q-learning.

It is also more space-efficient since it does not require storing values for every discrete state. When full exploration is infeasible, function approximation often outperforms tabular methods.

5.a

The `max_speed` parameter clips the car's velocity, limiting the range of positions the car can reach and constraining the dynamics of the environment.

5.b

Removing the velocity constraint does not affect the optimal behavior because it does not introduce more optimal states.

5.c

The two scenarios differ because increasing the maximum speed expands the range of velocities that can reach the goal, requiring greater control effort.

5.d

An MDP for exploring residential neighborhoods can be defined with:

$$\text{state} = (\text{position}, \text{velocity})$$

$$\text{actions} = \{\text{accelerate}, \text{brake}, \text{turn angle}\}$$

$$R(s, a, s') = \begin{cases} 100 & \text{if isEnd}(s') \\ -1 & \text{otherwise} \end{cases}$$

The policy is:

$$\pi(s) = \arg\max_a Q(s, a)$$

This formulation could cause harm because it does not account for road constraints or pedestrians. As a result, it may violate the beneficence principle by enabling unsafe behavior.