CSE 250A. Principles of AI

Probabilistic Reasoning and Decision-Making

Lecture 17 – Markov decision processes

Lawrence Saul
Department of Computer Science and Engineering
University of California, San Diego

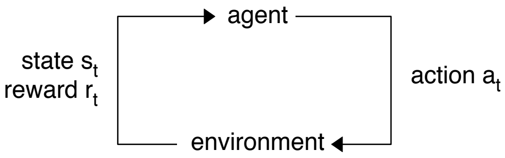
Fall 2021

Outline

1. Review and example
2. Value functions
3. Algorithms

Reinforcement learning (RL)

- **Learning from experience in the world**



- **Formalization as Markov decision process**

$\mathcal{S}$	state space
$\mathcal{A}$	action space
$P(s'\|s,a)$	transition probabilities
$R(s)$	reward function

MDP $\{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s)\}$

Policies and their expected returns

- **Decision-making in MDPs**

  A **policy** $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is a mapping of states to actions. There are combinatorially many policies:

  $$\#\text{ policies} \;=\; |\mathcal{A}|^{|\mathcal{S}|}$$

- **Experience under policy π**

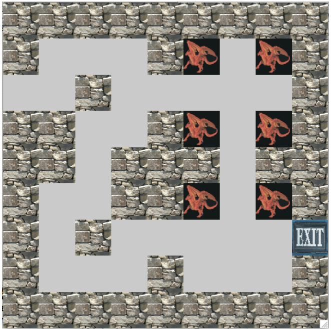
  $$\begin{array}{lccccccl} \text{state} & s_0 & \xrightarrow{\;\pi(s_0)\;} & s_1 & \xrightarrow{\;\pi(s_1)\;} & s_2 & \cdots \\ \text{reward} & r_0 & & r_1 & & r_2 & \cdots \end{array}$$

- **Expected returns**

  $$\mathrm{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s\right] \;=$$

  the expected value of the discounted infinite-horizon return, starting in state s at time $t=0$, and following policy π.

Example

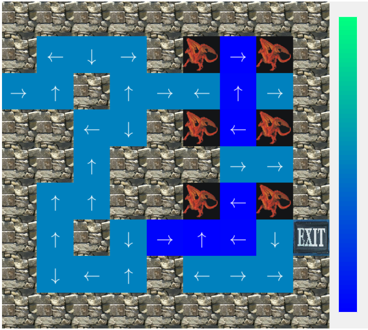


How to exit the maze with high probability?

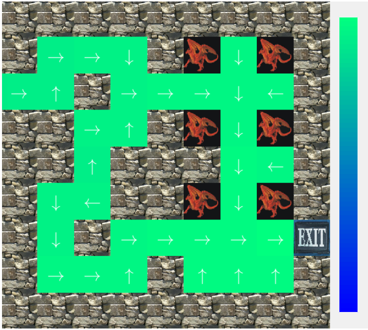
$\mathcal{S}$	location in maze
$\mathcal{A}$	$\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
$P(s' \mid s, a)$	move *with some probability* in direction of arrow
$R(s)$	$+1$ (exit), -1 (dragon), 0 (otherwise)
γ	0.99 (close to one)

Example — no uncertainty

$\mathcal{S}$	location in maze
$\mathcal{A}$	$\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
$P(s'\|s, a)$	move **deterministically** in direction of action
$R(s)$	+1 (exit), -1 (dragon), 0 (otherwise)
γ	0.99



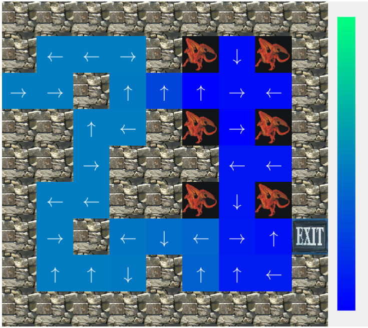
random policy



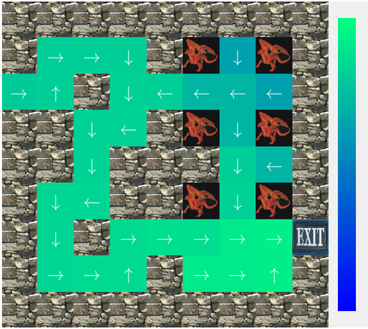
optimal policy

Example — low uncertainty

$\mathcal{S}$	location in maze
$\mathcal{A}$	$\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
$P(s'\|s,a)$	move **80%** in direction of action
$R(s)$	+1 (exit), -1 (dragon), 0 (otherwise)
γ	0.99



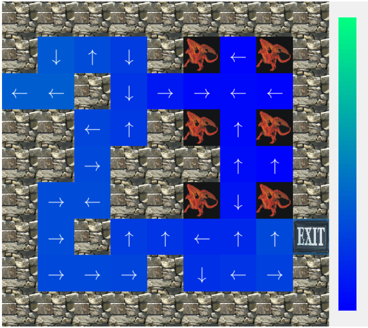
random policy



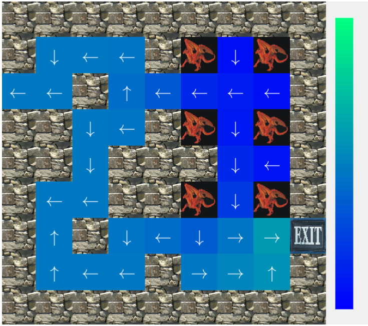
optimal policy

Example — high uncertainty

$\mathcal{S}$	location in maze
$\mathcal{A}$	$\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$
$P(s'\|s,a)$	move **20%** in direction of action
$R(s)$	+1 (exit), -1 (dragon), 0 (otherwise)
γ	0.99



random policy



optimal policy

Outline

1. Review and example
2. $\boxed{\text{Value functions}}$
3. Algorithms

State value function

$$V^\pi(s) \;=\; \mathrm{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s\right]$$

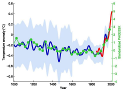
expected return,
starting in state s,
following policy π

- **Values versus rewards:**

  The reward $R(s)$ give **immediate** feedback to the agent.
  The value $V^\pi(s)$ computes the expected **long-term** return.

- **Types of behaviors:**

  Sacrifice now for long-term gain: $R(s) < 0,\ V^\pi(s) > 0$.
  Win now at the expense of later: $R(s) > 0,\ V^\pi(s) < 0$.



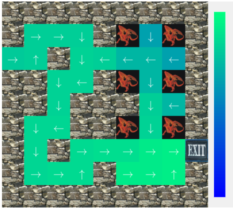
Properties of the state value function

- **Experience under policy π**

$$\begin{array}{lccccc} \text{state} & s_0 & \xrightarrow{\pi(s_0)} & s_1 & \xrightarrow{\pi(s_1)} & s_2 & \cdots \\ \text{reward} & r_0 & & r_1 & & r_2 & \cdots \end{array}$$

- **Adjacent states**

  States (s, s') can be visited in succession if $P(s'|s, \pi(s)) > 0$. The values $V^\pi(s)$ and $V^\pi(s')$ should be related, but how?



The **Bellman equation** tells us how.

Bellman equation

$$
\begin{aligned}
V^\pi(s) &= \mathrm{E}^\pi\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \,\middle|\, s_0 = s\right] \\
&= R(s) \;+\; \gamma\, \mathrm{E}^\pi\left[R(s_1) + \gamma R(s_2) + \cdots \,\middle|\, s_0 = s\right] \\
&= R(s) \;+\; \gamma \sum_{s'} P(s'|s,\pi(s))\, \mathrm{E}^\pi\left[R(s_1) + \gamma R(s_2) + \cdots \,\middle|\, s_1 = s'\right] \\
&= R(s) \;+\; \gamma \sum_{s'} P(s'|s,\pi(s))\, V^\pi(s')
\end{aligned}
$$

The Bellman equation is the basis for much that will follow:

$$
\boxed{V^\pi(s) \;=\; R(s) \;+\; \gamma \sum_{s'} P(s'|s,\pi(s))\, V^\pi(s')}
$$

Action value function

$$Q^\pi(s, a) = \mathrm{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s, a_0 = a\right]$$

expected return,
starting from state s,
taking action a,
then following policy π

- **Motivation**

  Useful to imagine how small changes affect expected outcomes. What if (just once) the agent acted differently in state s?

- **Analogous to the Bellman equation:**

$$Q^\pi(s, a) = R(s) + \gamma \sum_{s'} P(s'|s, a)\, V^\pi(s')$$

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s))\, V^\pi(s')$$

Optimality

- **Theorem**

  There exists at least one policy π^* (and perhaps many) such that $V^{\pi^*}(s) \geq V^{\pi}(s)$ for all policies π and states s of the MDP.

- **Proof**

  Later — by construction.

- **Notation**

$$
\begin{aligned}
V^*(s) &= V^{\pi^*}(s) \\
Q^*(s,a) &= Q^{\pi^*}(s,a)
\end{aligned}
$$

  These optimal value functions are **unique**.
  (All optimal policies share the same value functions.)

Relations at optimality

- **From the optimal action value function:**

$$\begin{aligned} V^*(s) &= \max_a \left[Q^*(s,a)\right] \\ \pi^*(s) &= \operatorname*{argmax}_a \left[Q^*(s,a)\right] \end{aligned}$$

- **From the optimal state value function:**

$$\begin{aligned} Q^*(s,a) &= R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \\ \pi^*(s) &= \operatorname*{argmax}_a \left[R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s')\right] \end{aligned}$$

- **Why are these relations useful?**

  Sometimes it can be easier to estimate $Q^*(s,a)$ or $V^*(s)$ (which are continuous) than to learn $\pi^*(s)$ (which is discrete).

Outline

1. **Review**
2. **Value functions**
3. **Algorithms**

Planning in MDPs

Given a complete model of the agent and its environment as a Markov decision process, namely

$$\text{MDP} \;=\; \{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s), \gamma\},$$

how can we *efficiently* compute (i.e., in time *polynomial in the number of states*) any of the following:

1. an optimal policy $\pi^*(s)$?
2. the optimal state value function $V^*(s)$?
3. the optimal action value function $Q^*(s,a)$?

This is the problem of **planning** in MDPs.

Three algorithms

Today we'll describe three basic algorithms.
Each of them solves a core problem in MDPs.

1. **Policy evaluation**

   How to compute $V^{\pi}(s)$ for some fixed policy π?

2. **Policy improvement**

   How to compute a policy π' such that $V^{\pi'}(s) \geq V^{\pi}(s)$?

3. **Policy iteration**

   How to compute an optimal policy $\pi^*(s)$?

Policy evaluation

For a fixed policy π, how to compute the state value function

$$V^{\pi}(s) \;=\; \mathrm{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \,\middle|\, s_0 = s\right]?$$

From the Bellman equation:

$$V^{\pi}(s) \;=\; R(s) + \gamma \sum_{s'} P(s'|s,\pi(s))\, V^{\pi}(s').$$

This is a system of n linear equations for n unknowns;
the unknowns are $V^{\pi}(s)$ for $s \in \{1, 2, \ldots, n\}$ where $n = |\mathcal{S}|$.

Solving the linear system

- **From the Bellman equation:**

$$V^{\pi}(s) \;=\; R(s) + \gamma \sum_{s'} P(s'|s,\pi(s))\, V^{\pi}(s').$$

- **Rearranging terms:**

$$\begin{aligned} R(s) &= V^{\pi}(s) - \gamma \sum_{s'} P(s'|s,\pi(s))\, V^{\pi}(s') \\ &= \sum_{s'} \Big[\underbrace{I(s,s')}_{\text{identity matrix}} - \gamma P(s'|s,\pi(s)) \Big] V^{\pi}(s') \end{aligned}$$

- **In matrix-vector form:**

$$R \;=\; \Big[I - \gamma P^{\pi} \Big] V^{\pi}$$

$$\left[\begin{array}{c} \text{column vector of} \\ n \text{ known rewards} \end{array} \right] = \left[\begin{array}{c} n \times n \text{ matrix} \\ \text{(known)} \end{array} \right] \left[\begin{array}{c} \text{column vector of} \\ n \text{ unknown values} \end{array} \right]$$

Solving the linear system (con't)

- **Solution**

$$R = \left[I - \gamma P^\pi\right] V^\pi \quad \Longrightarrow \quad V^\pi = \underbrace{(I - \gamma P^\pi)^{-1}}_{\text{matrix inverse}} R$$

- **Complexity**

  It takes $O(n^2)$ operations to solve this system of equations.
  (For very large n, HW 9.5 develops an iterative solution.)

- **Example**

  Let $\mathcal{S} = \{1, 2\}$ and $P(s'|s, \pi(s)) = 0.5$ for all (s, s').

$$\begin{bmatrix} V^\pi(1) \\ V^\pi(2) \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \gamma \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \right)^{-1} \begin{bmatrix} R(1) \\ R(2) \end{bmatrix}.$$

Policy improvement

- **Problem statement**

  Given a policy π and its state value function $V^{\pi}(s)$, how to compute a policy π' such that

$$V^{\pi'}(s) \ \geq V^{\pi}(s) \quad \text{for all states } s?$$

- **Definition**

  Given the action value function $Q^{\pi}(s,a)$ for policy π, we define the **greedy policy** π' by

$$\pi'(s) \ = \ \operatorname*{argmax}_a \Big[Q^{\pi}(s,a) \Big].$$

  Why *greedy*? Because we change the action in state s to whatever appears to improve the expected return.

Greedy policies

- **In terms of the state value function:**

$$\begin{aligned}
\pi'(s) &= \operatorname*{argmax}_a \Big[Q^\pi(s,a) \Big] \\
&= \operatorname*{argmax}_a \Big[R(s) + \gamma \textstyle\sum_{s'} P(s'|s,a)\, V^\pi(s') \Big] \\
&= \operatorname*{argmax}_a \Big[\textstyle\sum_{s'} P(s'|s,a)\, V^\pi(s') \Big]
\end{aligned}$$

- **Test your understanding:**

$\pi'(s) = \pi(s)$ for some $s \in \mathcal{S}$? **not necessarily**

$\pi'(s) \neq \pi(s)$ for some $s \in \mathcal{S}$? **not necessarily**

$Q^\pi(s, \pi'(s)) \geq Q^\pi(s, \pi(s))$ for all $s \in \mathcal{S}$? **TRUE**

Policy improvement

- **Theorem**

  The greedy policy π' everywhere dominates the policy π from whose value functions it was derived:

$$V^{\pi'}(s) \;\geq V^{\pi}(s) \quad \text{for all states } s \in \mathcal{S}.$$

- **Proof:** *next lecture.*

- **Is this surprising?**

  Yes — greedy methods (in general) often have shortcomings.

  No — the MDP lends itself to this approach.

Policy iteration

How to compute π^*?

1. **Choose an initial policy $\pi : \mathcal{S} \to \mathcal{A}$.**
2. **Repeat until convergence:**

   Compute the action value function $Q^\pi(s, a)$.
   Compute the greedy policy $\pi'(s) = \operatorname{argmax}_a Q^\pi(s, a)$.
   Replace π by π'.

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s, a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_1}(s) \\ Q^{\pi_1}(s, a) \end{matrix} \xrightarrow{\text{improve}} \cdots$$

Convergence of policy iteration

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \cdots$$

- **Why must this converge?**

  Recall that $V^{\pi'}(s) \geq V^{\pi}(s)$ for all $s \in \mathcal{S}$.
  Thus we cannot cycle back to old policies.

  Also the number of policies is finite: $|\mathcal{A}|^{|\mathcal{S}|} < \infty$.
  Thus we cannot improve indefinitely.

- **Why does this converge to an optimal policy?**

  We will prove this in the next lecture.

Convergence of policy iteration

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \cdots$$

- **Why must this converge?**

  Recall that $V^{\pi'}(s) \geq V^{\pi}(s)$ for all $s \in \mathcal{S}$.
  Thus we cannot cycle back to old policies.

  Also the number of policies is finite: $|\mathcal{A}|^{|\mathcal{S}|} < \infty$.
  Thus we cannot improve indefinitely.

- **Why does this converge to an optimal policy?**

  We will prove this in the next lecture.
  Also there will be **DEMOS** ...