Reinforcement Learning
IA318
Markov Decision Process
Dynamic Programming

Thomas Bonald

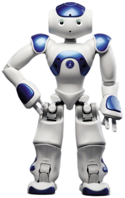
2022 – 2023

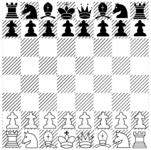
Reinforcement learning

Techniques for sequential decision making:

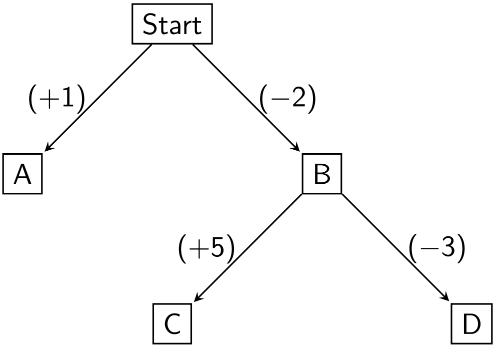
- How to **explore** the environment?
- How to **exploit** it?
- How to leverage **experience**?

Need for **feedback**!
The objective is to learn the **optimal policy** sequentially (possibly after multiple episodes)

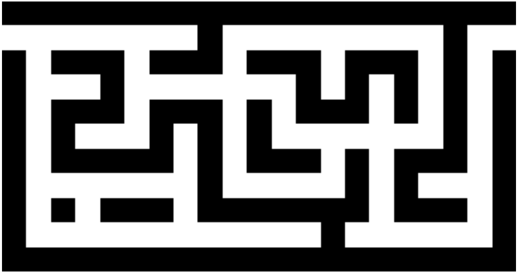


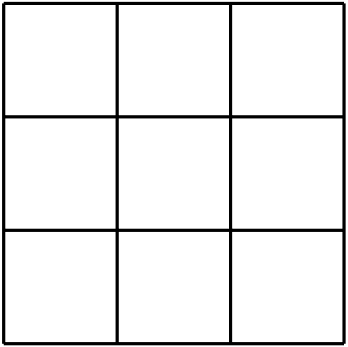
Example 1: ABCD



Example 2: Maze



Example 3: Tic-Tac-Toe



Outline

- **Markov decision process**
- Value function
- Policy iteration
- Value iteration

Markov decision process

At time $t = 0, 1, 2, \ldots$, the agent in **state** s_t takes **action** a_t and:

- receives **reward** r_t
- moves to **state** s_{t+1}

The reward and new state are **stochastic** in general.
Some states may be **terminal** (e.g., games).

Definition

A **Markov decision process** (MDP) is defined by:

- the initial state distribution, $p(s_0)$
- the reward distribution, $p(r_t | s_t, a_t)$
- the transition probabilities, $p(s_{t+1} | \, s_t, a_t)$

We denote by S the set of **non-terminal** states.

Policy

Definition

Given a Markov decision process, a **policy** defines the action taken in each non-terminal state:

$$\forall s \in S, \quad \pi(a|s) = P(a_t = a|\ s_t = s)$$

- A policy is **stochastic** in general.
- When **deterministic**, we use the simple notation $\pi(s)$ for the action taken in state s.

Remark

Given some policy π, the sequence of states $s_0, s_1, s_2, \ldots$ defines a **Markov chain**.

Objective function

Definition

Given the rewards $r_0, r_1, r_2, \ldots$, we refer to the **gain** as:

$$G = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots = \sum_{t=0}^{+\infty} \gamma^t r_t$$

This sum might be **truncated** in the presence of terminal states.

The parameter $\gamma \in [0, 1]$ is the **discount factor**:

- $\gamma = 0 \longrightarrow$ immediate reward
- $\gamma = 1 \longrightarrow$ cumulative reward

In the absence of terminal states, we assume $\gamma < 1$.

Outline

- Markov decision process
- **Value function**
- Policy iteration
- Value iteration

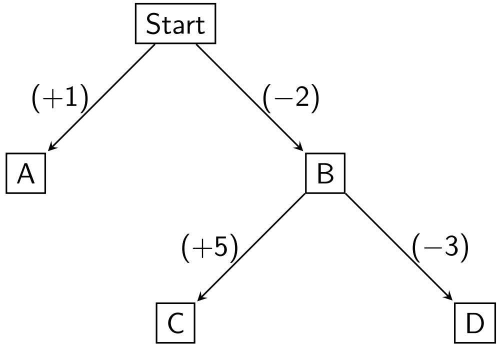
Value function

Definition

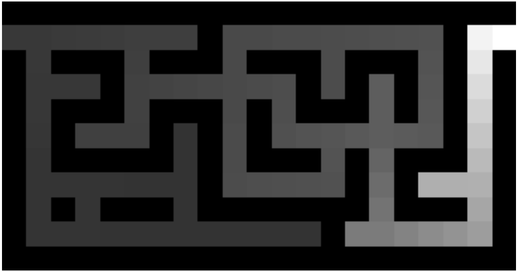
The **value function** of a policy π is the **expected gain** from each state:

$$\forall s, \quad V_\pi(s) = \mathrm{E}_\pi(G | s_0 = s)$$

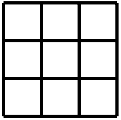
ABCD (random policy)



Maze (random policy)



Tic-Tac-Toe (random players)



$t = 0$

0.34	0.2	0.34
0.2	**0.5**	0.2
0.34	0.2	0.34

$t = 2$

X	0.16	**0.27**
0.16	O	0.02
0.27	0.02	-0.12

Bellman's equation

Recall the definition:

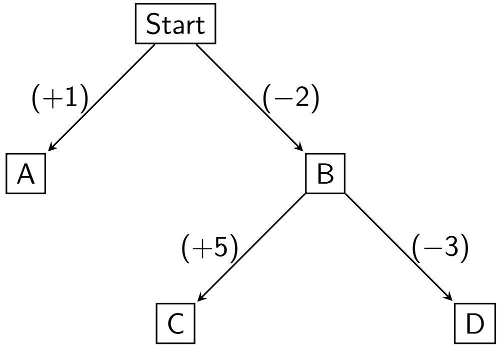
$$\forall s, \quad V_\pi(s) = \mathrm{E}_\pi(G|s_0 = s)$$

Proposition

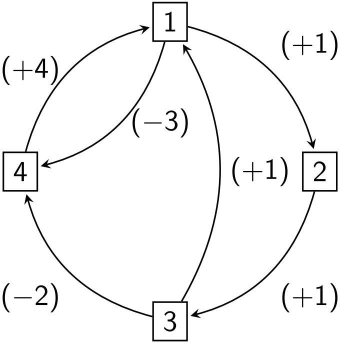
The value function V_π of any policy π is the **unique solution** to the equation:

$$\forall s \in S, \quad V(s) = \mathrm{E}_\pi(r_0 + \gamma V(s_1)|s_0 = s)$$

ABCD (random policy)



Quiz



Consider the following policy π: $\pi(1) = 2, \pi(3) = 4$
The discount factor is $\gamma = \frac{1}{2}$
What is the value function V_π?

Solution to Bellman's equation

Write Bellman's equation as the **fixed-point** equation:

$$\boxed{V = T_\pi(V)}$$

with $T_\pi(V)(s) = \mathrm{E}_\pi(r_0 + \gamma V(s_1) |\ s_0 = s)$ for all $s \in S$.

Proposition

If $\gamma < 1$, then:

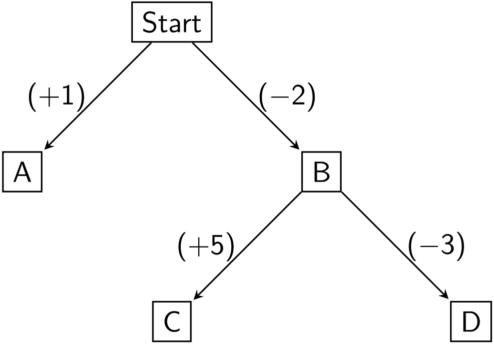
$$\forall V, \quad \lim_{n \to +\infty} T_\pi^n(V) = V_\pi$$

Proof: The mapping T_π is contracting:

$$\forall U, V, \quad ||T_\pi(V) - T_\pi(U)||_\infty \leq \gamma ||V - U||_\infty$$

$\rightarrow$ Banach fixed-point theorem

ABCD (fixed-point iteration)



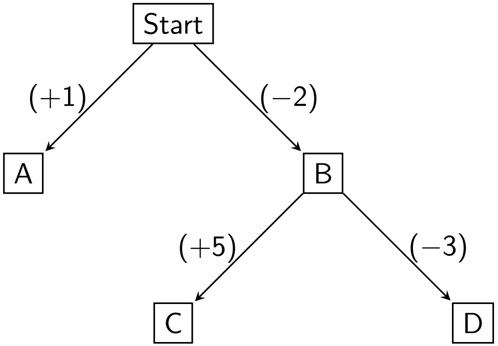
Optimal policy

Definition

A policy $\pi^\star$ is **optimal** if and only if

$$\forall s, \quad V_{\pi^\star}(s) \geq V_\pi(s)$$

ABCD (optimal policy)



Bellman's optimality equation

Recall Bellman's equation for a policy π:

$$\begin{aligned}\forall s \in S, \quad V(s) &= \mathrm{E}_\pi(r_0 + \gamma V(s_1) | s_0 = s) \\ &= \sum_a \pi(a|s) \mathrm{E}(r_0 + \gamma V(s_1) | s_0 = s, a_0 = a)\end{aligned}$$

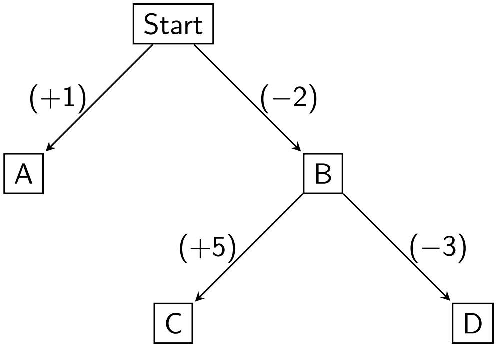
Let's replace π by the best action in each state.

Proposition

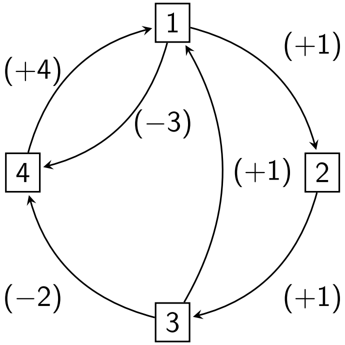
There is a **unique solution** $V^\star$ to the equation:

$$\forall s \in S, \quad V(s) = \max_a \mathrm{E}(r_0 + \gamma V(s_1) |\ s_0 = s, a_0 = a)$$

ABCD (optimal value function)



Quiz



The discount factor is $\gamma = \frac{1}{2}$
What is the optimal value function $V^{\star}$?

Solution to Bellman's optimality equation

Write Bellman's optimality equation as the **fixed-point** equation:

$$V = T^\star(V)$$

with $T^\star(V)(s) = \max_a \mathrm{E}(r_0 + \gamma V(s_1)|\ s_0 = s, a_0 = a), \forall s \in S$.

Proposition

If $\gamma < 1$, then:

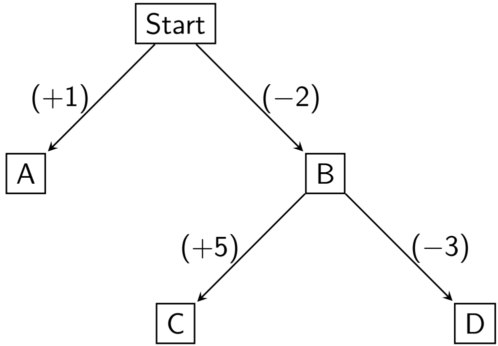
$$\forall V, \quad \lim_{n\to+\infty} (T^\star)^n(V) = V^\star \geq \max_\pi V_\pi$$

Proof:

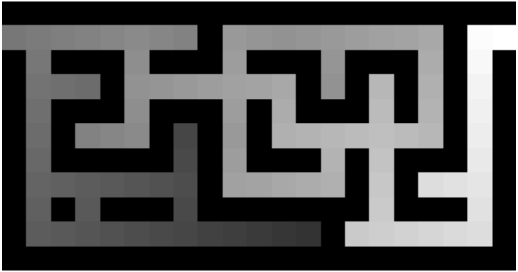
- $T^\star$ is contracting $\to$ Banach fixed-point theorem
- $T^\star \geq T_\pi$ for each policy π, so that:

$$V^\star = \lim_{n\to+\infty} (T^\star)^n(V) \geq \lim_{n\to+\infty} T_\pi^n(V) = V_\pi$$

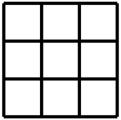
ABCD (fixed-point iteration)



Maze (optimal value function)



Tic-Tac-Toe
(optimal value function against a random player)



$t = 0$

0.995	0.987	**0.995**
0.987	0.980	0.987
0.995	0.987	**0.995**

$t = 2$

X	**0.96**	0.92
0.96	O	0.89
0.92	0.89	0.92

Optimal policy

An **optimal policy** follows from the optimal value function:

$$\forall s \in S, \quad \pi^\star(s) = a^\star \in \arg\max_a \mathrm{E}(r_0 + \gamma V^\star(s_1) |\ s_0 = s, a_0 = a)$$

Bellman's optimality theorem

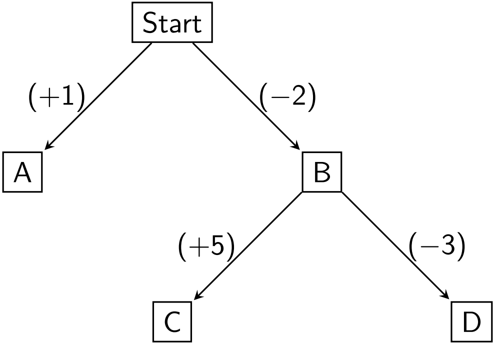
The policy $\pi^\star$ is optimal:

$$\forall s, \quad V_{\pi^\star}(s) = \max_\pi V_\pi(s)$$

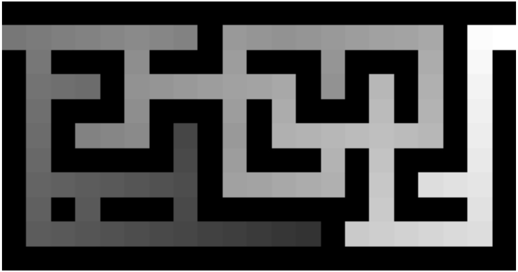
Note that:

- The optimal policy is **not unique** in general.
- There always exists a **deterministic** optimal policy.
- Any policy π whose value function V_π solves **Bellman's optimality equation** is optimal.

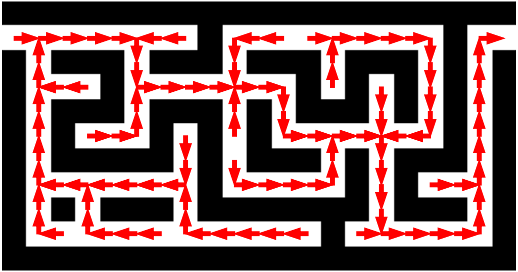
ABCD (optimal policy)



Maze (optimal policy)



Maze (optimal policy)



Outline

- Markov decision process
- Value function
- **Policy iteration**
- Value iteration

Policy improvement

Assume the value function V_π of policy π is known.
Let π' the policy defined by:

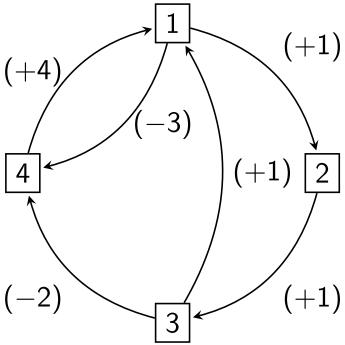
$$\pi'(s) = a^\star \in \arg\max_a \mathrm{E}(r_0 + \gamma V_\pi(s_1)|\ s_0 = s, a_0 = a)$$

Proposition

The policy π' is better than π:

$$\forall s, \quad V_{\pi'}(s) \geq V_\pi(s)$$

Quiz



Consider the following policy π: $\pi(1) = 2, \pi(3) = 4$
The discount factor is $\gamma = \frac{1}{2}$
What is the new policy π' after policy improvement?

Policy iteration

Algorithm

Starting from some arbitrary **policy** $\pi = \pi_0$, iterate until convergence:

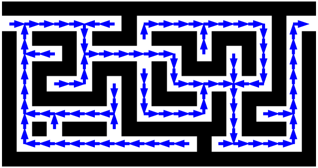
1. **Evaluate** the policy (by solving Bellman's equation)
2. **Improve** the policy:

$$\forall s, \quad \pi(s) \leftarrow \arg\max_a \mathrm{E}(r_0 + \gamma V_\pi(s_1) |\ s, a)$$

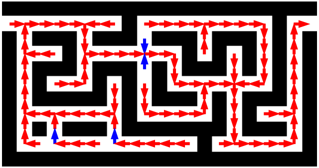
- The sequence $\pi_0, \pi_1, \pi_2, \ldots$ is **monotonic** (in value) and converges in **finite time** (for finite numbers of states and actions).
- The limit is an **optimal policy**.
- These results assume **perfect** policy evaluation.

Maze (policy iteration)

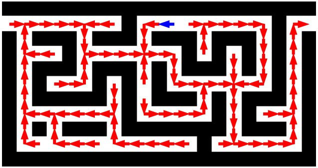
$n = 1$



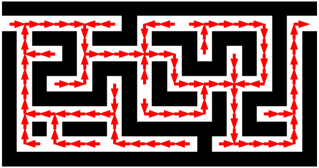
$n = 2$



$n = 5$



$n = 6$



Practical considerations

- The step of **policy evaluation** is time-consuming (solution of Bellman's equation)
- Do we need the **exact** solution?
  No, since it is used only to **improve** the policy!
- Why not directly improving the **value function**?
  This is value iteration!

Outline

- Markov decision process
- Value function
- Policy iteration
- **Value iteration**

Value Iteration

Algorithm

Starting from some arbitrary **value** function $V = V_0$,
iterate until convergence:

$$\forall s, \quad V(s) \leftarrow \max_a \mathrm{E}(r_0 + \gamma V(s_1) | \, s, a)$$

- The sequence $V_0, V_1, V_2, \ldots$ converges (provided $\gamma < 1$) but not in finite time in general $\rightarrow$ need a **stopping** condition
- The **limit** solves Bellman's optimality equation.
- The corresponding policy is **optimal**.

Summary

Key concepts

- **Markov decision process**
  A general model for reinforcement learning
- **Value function**
  Expected gain in each state
- **Policy iteration**
  Based on successive Bellman's equations
- **Value iteration**
  Based on Bellman's optimality equation

Next steps

- What if the state space is **too large**?
- What if the model is **unknown**?