Reinforcement Learning
IA318
TD Learning

Thomas Bonald

2022 – 2023

TELECOM Paris

IP PARIS

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**.

Definition

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

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

Objective function

Definition

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

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

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

Value function

Consider some policy π.

Definition

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

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

Bellman's equation

The value function V_π is the unique solution to the **fixed-point equation**:

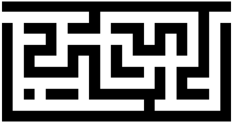
$$\forall s, \quad V(s) = \mathrm{E}_\pi(r_0 + \gamma V(s_1) |\, s_0 = s)$$

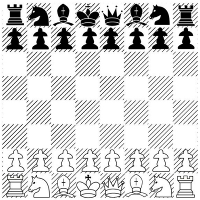
Online policy evaluation

How to evaluate the value function V_π online, while **interacting** with the environment?

Useful when:

- The environment is **unknown** (e.g., robot, maze)
- The state space is **too large** (e.g., games)





Data stream

How to estimate the **mean** M of some data stream $x_1, x_2, \ldots$?

Two options:

1. Store the **sum**:

$$S \leftarrow S + x_t \quad M \leftarrow \frac{S}{t}$$

2. Update with the **difference**:

$$M \leftarrow M + \alpha(x_t - M) \quad \alpha = \frac{1}{t}$$

We use the notation:

$$M \overset{\alpha}{\leftarrow} x_t - M$$

Note: α is often set to some (small) value to account for **non-stationarity** (e.g., $\alpha = 0.01$)

Outline

1. **Monte-Carlo learning**
2. TD learning

MC learning

Idea: Evaluate the **value function** of some policy π using **complete episodes** $s_0, s_1, \ldots, s_T$ (assuming the presence of terminal states, or some fixed time horizon T)

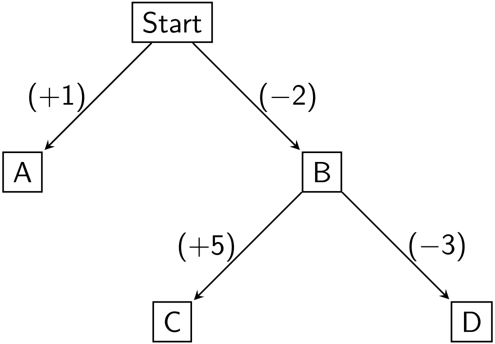
Gain G_t at time t:

$$
\begin{aligned}
G_0 &= r_0 + \gamma r_1 + \ldots + \gamma^{T-1} r_{T-1} \\
G_1 &= r_1 + \gamma r_2 + \ldots + \gamma^{T-2} r_{T-1} \\
&\ldots \\
G_{T-1} &= r_{T-1}
\end{aligned}
$$

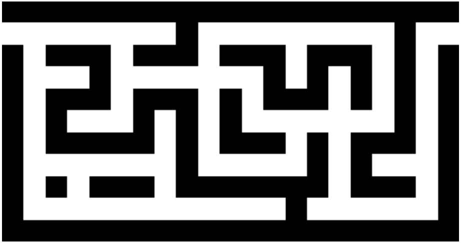
MC updates

$$\forall t, \quad V(s_t) \overset{\alpha}{\leftarrow} G_t - V(s_t)$$

Example

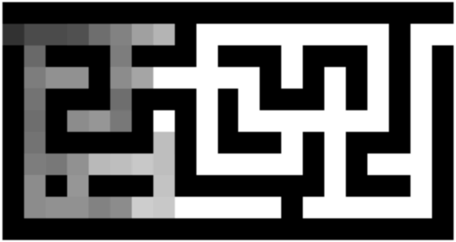


Maze

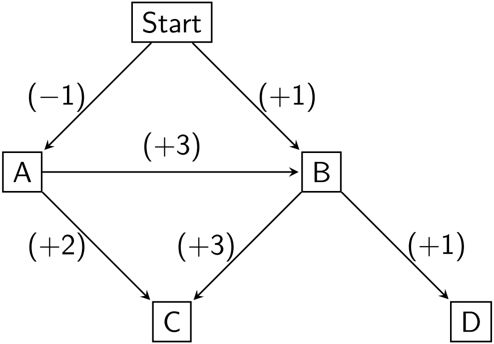


Maze

MC learning on 10 episodes with time horizon $T = 200$



Exercise



What is the value function after MC learning over the episodes [Start, A, B, C] and [Start, B, C]?

Outline

1. Monte-Carlo learning
2. **TD learning**

TD learning

Idea: Online estimation of the **value function** of some policy π (no need for **complete episodes**)

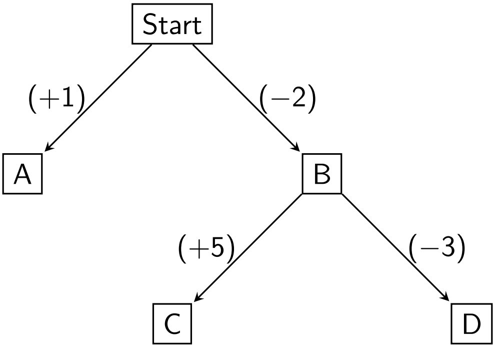
TD updates

$$\forall t, \quad V(s_t) \stackrel{\alpha}{\leftarrow} r_t + \gamma V(s_{t+1}) - V(s_t)$$

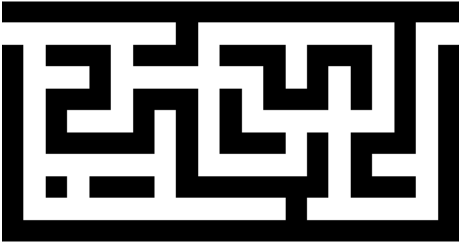
cf. Bellman's equation

$$\forall s, \quad V(s) = \mathrm{E}_\pi(r_t + \gamma V(s_{t+1}) | \, s_t = s)$$

Example

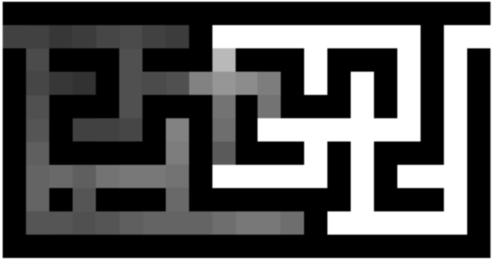


Maze

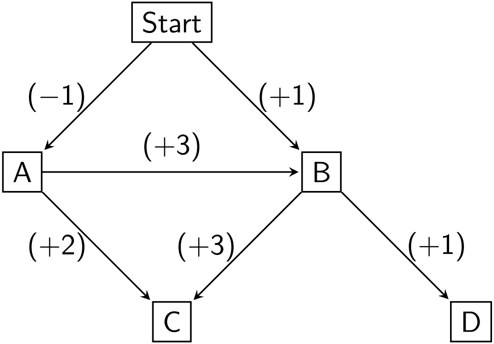


Maze

TD learning on 10 episodes with time horizon $T = 200$



Exercise



What is the value function after TD learning over the episodes [Start, A, B, C] and [Start, B, C]?

MC learning vs TD learning

MC

- requires **complete** episodes
- requires **memory**
- has **high variance** but **low bias**

TD

- learns **continuously**
- is **memory-less** (cf. Markov property)
- has **low variance** but potentially **high bias** (depending of the initial value of V)

From TD to MC: *n*-step TD

Estimation of the gain at time t after n time steps:

$$
\begin{aligned}
G_t^{(1)} &= r_t + \gamma V(s_{t+1}) \\
G_t^{(2)} &= r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \\
&\dots \\
G_t^{(n)} &= r_t + \gamma r_{t+1} + \ldots + \gamma^n V(s_{t+n})
\end{aligned}
$$

n-step TD

$$\forall t, \quad V(s_t) \xleftarrow{\alpha} G_t^{(n)} - V(s_t)$$

Summary

Key concepts

- **MC learning**
  Learning from complete episodes
- **TD learning**
  Online learning
- Both useful for **policy improvement**

Next steps

- Online control by **Q-learning**
- Opportunistic exploration by **Monte-Carlo Tree Search**