Reinforcement Learning

Tom Vodopivec

IADS Analytics, Data Science & Decision Making Summer School 2022
2022-08-01

The Field

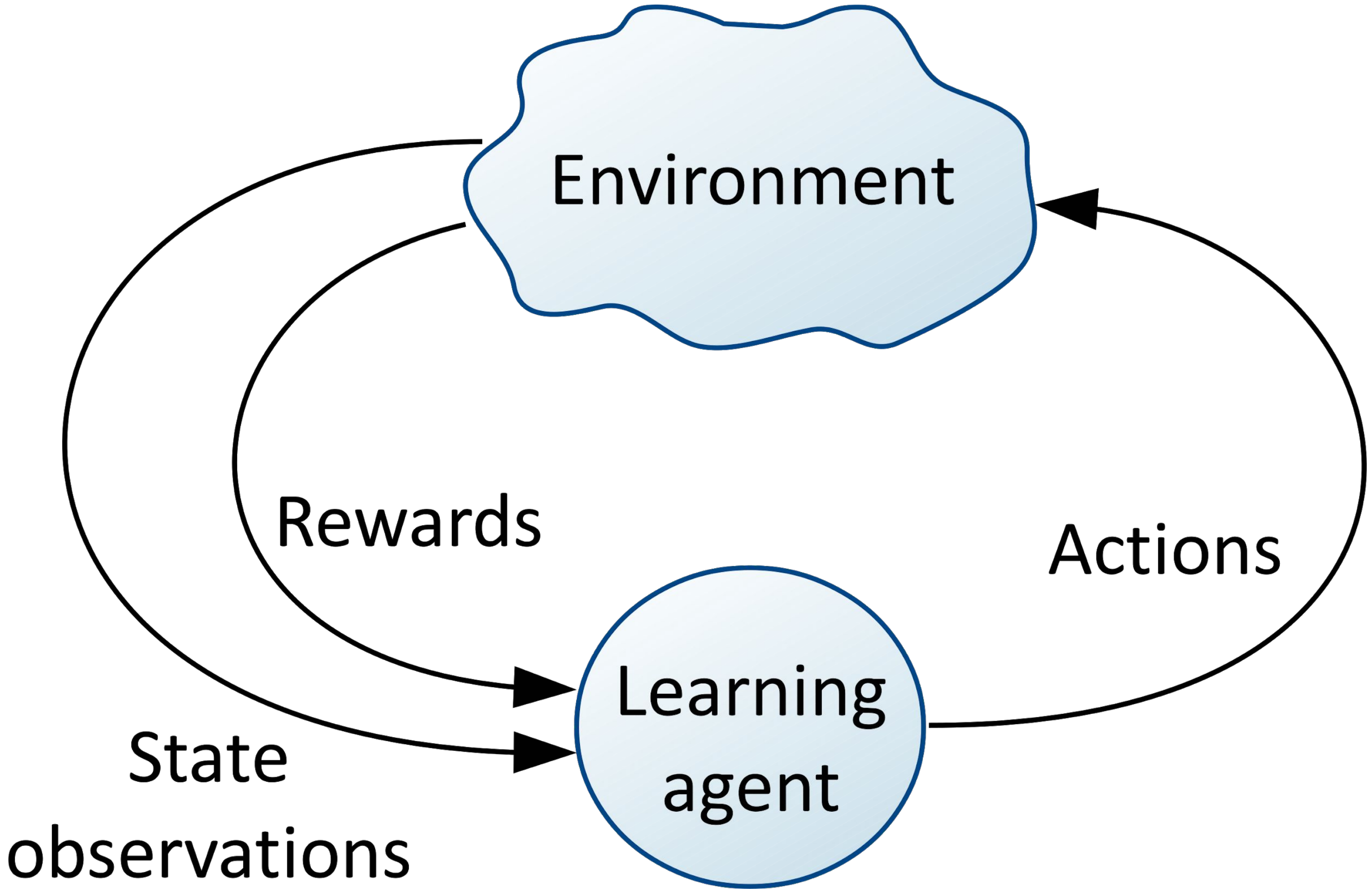
Artificial Intelligence > Machine Learning

(A special case of) supervised learning

But also has elements of unsupervised learning

//////

The Reinforcement Learning Setting



The Reinforcement Learning Setting

Agent

Behaviour > Actions

Environment

States

Rewards > define the goal of the agent

+

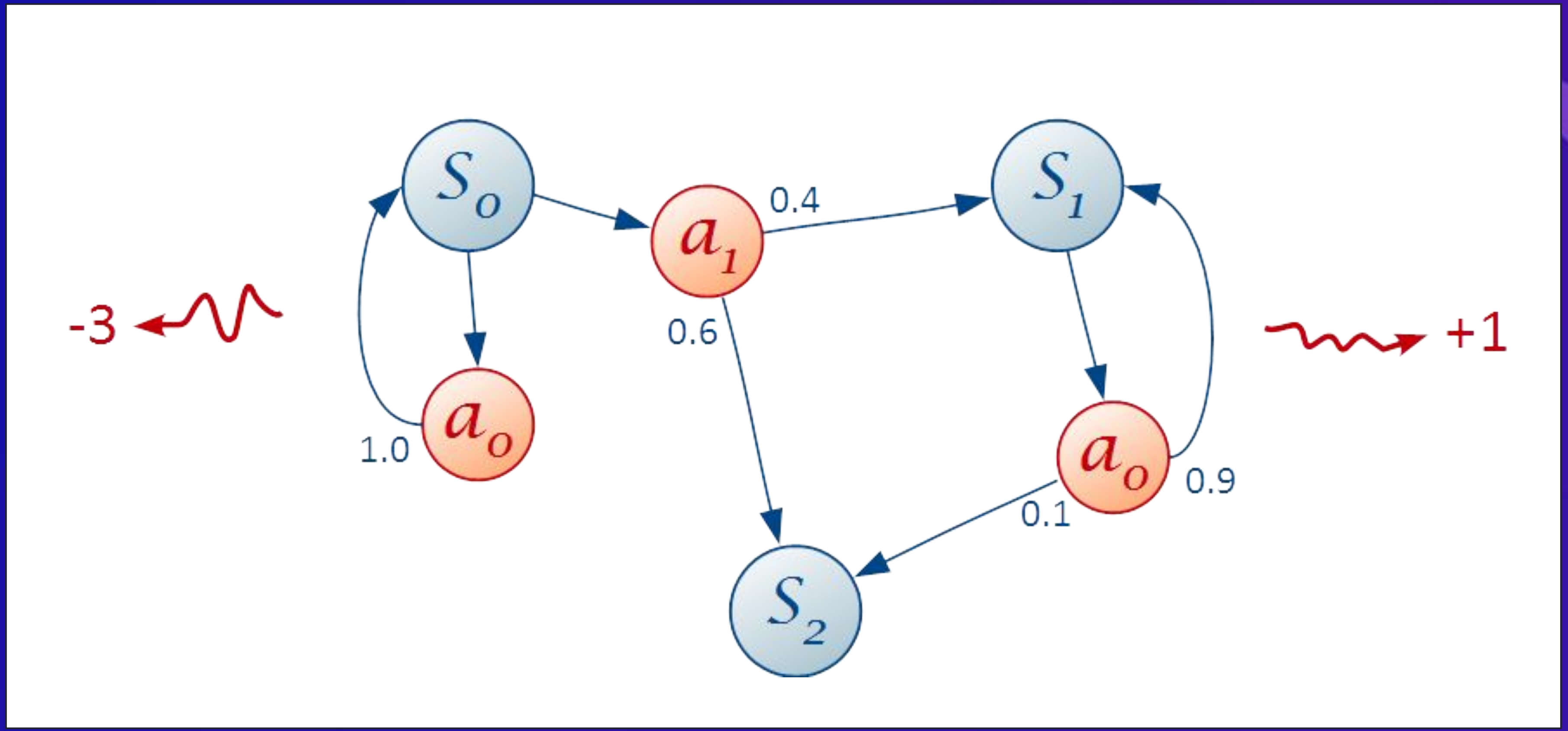
Markov Decision Processes

States

Actions

Rewards

Transition probabilities



+

Markov Decision Processes

Trajectory (concept of time)

(Expected) return

Discount rate

Continuing vs. episodic tasks

Terminal states

Non-stationary MDPs

Partially-observable MDPs

+

What to Learn From?

Experience = Samples of interaction with environment

- Real (learning)
- Simulated (planning)

Exploration-exploitation trade-off

Uncertainty

How to Store Knowledge?

State-value function

Action-value function

Behaviour policy

Model (of the environment)

+

Representations

- Tabular vs. approximate
- Fixed vs. adaptive

Updating Knowledge = Learning

Goal: learn optimal policy (and optimal value function)

Generalized policy iteration

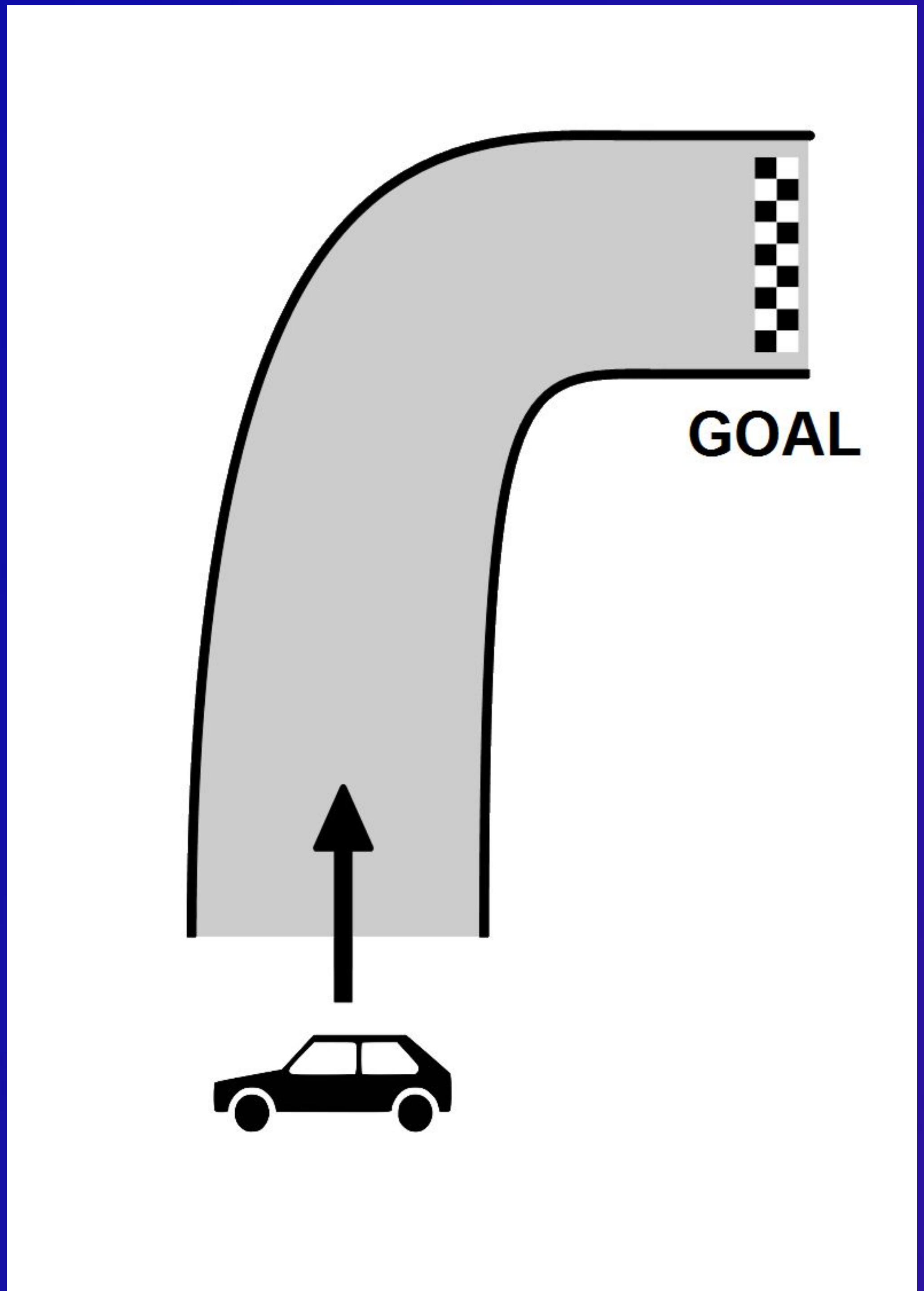
- Policy evaluation
- Policy improvement

+

Algorithms

- Model-based
- Model-free

Race Track Example



Algorithms

Dynamic programming

Monte Carlo

Temporal-difference learning

- Bootstrapping
- Eligibility traces

Algorithms

Dynamic programming: policy iteration, value iteration

Monte Carlo

Temporal-difference learning: TD(λ), Sarsa, Q-learning, Deep Q-learning

- Bootstrapping
- Eligibility traces

+

Algorithms

Dynamic programming: policy iteration, value iteration

Monte Carlo

Temporal-difference learning: TD(λ), Sarsa, Q-learning, Deep Q-learning

- Bootstrapping
- Eligibility traces

Policy gradient and Actor-Critic: REINFORCE

Applications

Games: AlphaGo, AlphaZero

Scheduling tasks: optimization of memory control

Modelling bird movement

Web services / optimization

+

Reading Material

Quick and practical state-of-the-art:

Thomas Simonini, Deep Reinforcement Learning course

Most comprehensive and best foundations:

Richard S. Sutton and Andrew G. Barto, Reinforcement Learning, An introduction, second edition

Outstanding applications:

Silver et al., AlphaGo, AlphaGo Zero, AlphaZero

Recap

Agent, actions, environnement, state, rewards

Explore and collect experience

Representation of knowledge

Update knowledge = learn

Improve behaviour

+

//////

Equations

$$V(S_t) \leftarrow V(S_t) + \alpha_n\Big[G_t - V(S_t)\Big],$$

$$V_{t+1}(S_t) = V_t(S_t) + \alpha_t\Big[(R_{t+1} + \gamma V_t(S_{t+1})) - V_t(S_t)\Big].$$

$$\delta_t = R_{t+1} + \gamma V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(s) = V_t(s) + \alpha_t \delta_t E_t(s)$$

$$E_t(s) = \begin{cases} 1 & \text{if } s = S_t \text{ (replacing)}, \\ \gamma\lambda E_{t-1}(s) + 1 & \text{if } s = S_t \text{ (accumulating)}, \\ \gamma\lambda E_{t-1}(s) & \text{if } s \neq S_t. \end{cases}$$

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

Policy Iteration (using iterative policy evaluation) for es

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
     $\Delta \leftarrow 0$
     Loop for each $s \in \mathcal{S}$:
       $v \leftarrow V(s)$
       $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\big[r + \gamma V(s')\big]$
       $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accurac

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
     old-action $\leftarrow \pi(s)$
     $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
     If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else