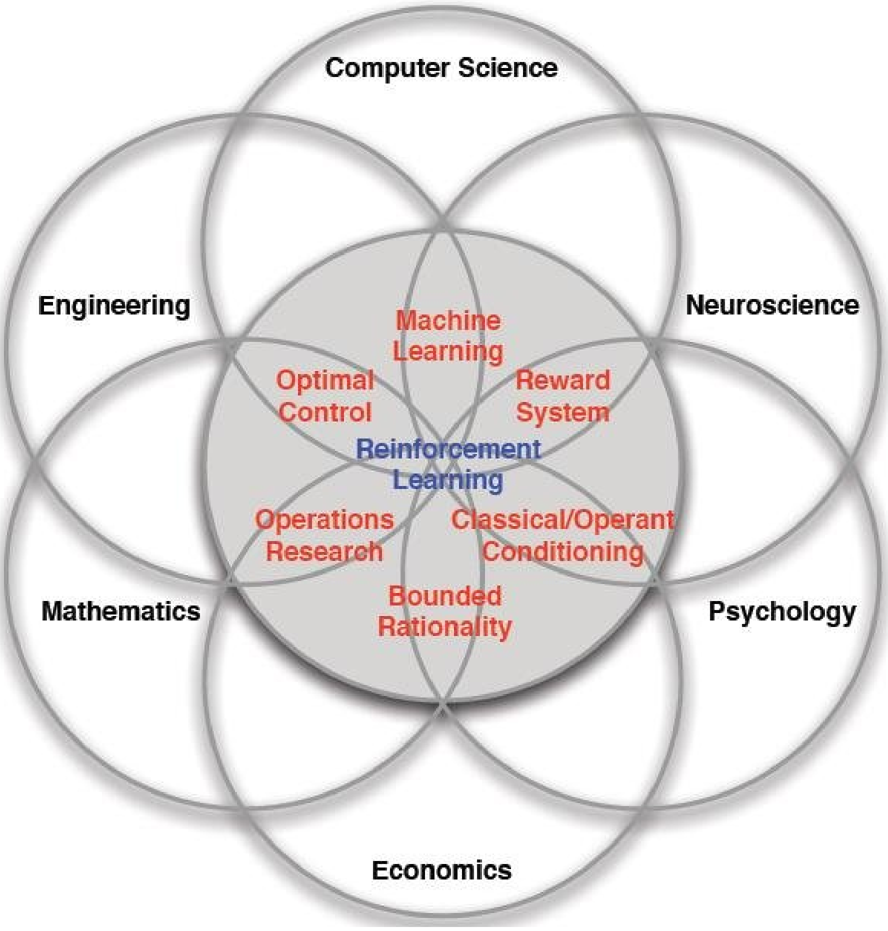
Universidad Zaragoza

Introduction to Reinforcement Learning

Machine Learning (69152)

Rubén Martínez Cantín
Dpto. Informática e Ingeniería de Sistemas
Universidad de Zaragoza

What is reinforcement learning



Credit: David Silver

Reinforcement learning vs (un)supervised machine learning

Reinforcement learning is both a subfield and a problem definition.

- There is an agent that takes actions.
- No supervision, labels or oracle. Just a reward (good vs bad).
- Dynamic/sequential problem. Time is always involved.
- Data and actions are interconected.
- Noisy movement. Non-deterministic dynamics.
- The target is a behaviour, policy or controller.
- Optimal policy: maximize the reward.

Examples of reinforcement learning

- Fly stunt manoeuvres in a helicopter
- Defeat the world champion at Go
- Manage an investment portfolio
- Control a power station
- Make a humanoid robot walk
- Autonomous driving and parking
- Play many different Atari games better than humans
- Design realistic computer simulations.

Videos

What is a reward?

Reward hypothesis [Sutton and Barto]

That all of what we mean by **goals** and purposes can be well thought of as the **maximization of the expected value** of the cumulative sum of a received scalar signal (called **reward**).

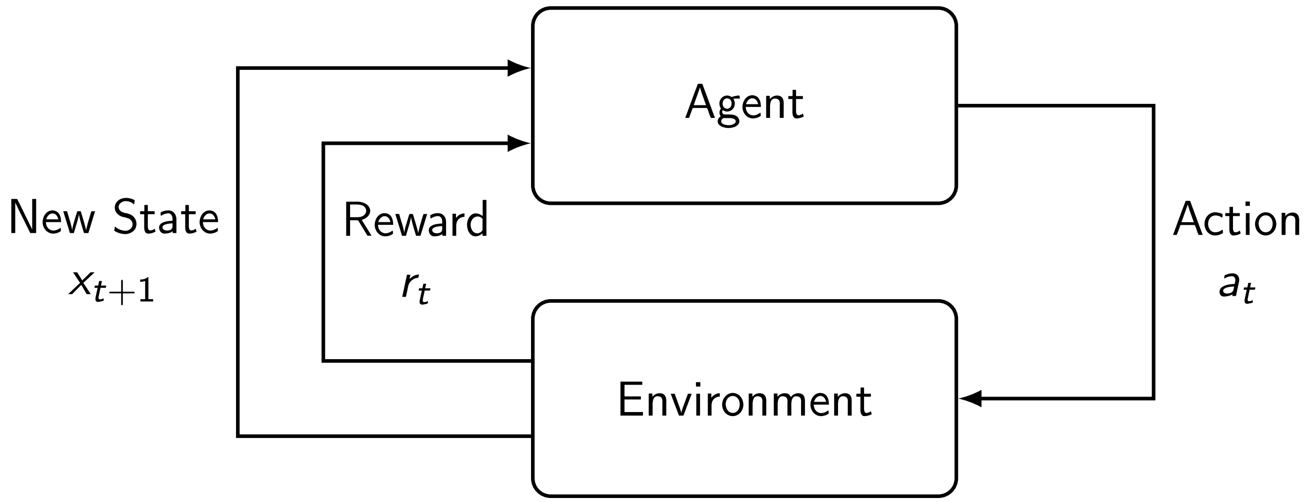
- A reward R_t is a scalar feedback signal (e.g.: score in a game)
- Indicates how well agent is doing at step t
- Objective: maximize cumulative (past, present and future) reward

- In reinforcement learning, people usually work with rewards
- In control theory/robotics/engineering, people usually work with costs
- *Maximize reward* vs *minimize cost* $\rightarrow$ reward = -cost

Examples of rewards/costs

- Fly stunt manoeuvres in a helicopter/Autonomous driving and parking
  - + reward for following desired trajectory
  - - reward for crashing/energy/fuel
- Defeat the world champion at Go
  - +/- reward for winning/losing a game
- Manage an investment portfolio
  - + reward for each $ in bank
- Control a power station
  - + reward for producing power
  - - reward for exceeding safety thresholds
- Make a humanoid robot walk
  - + reward for forward motion/smoothness
  - - reward for falling over
- Play many different Atari games better than humans
  - +/- reward for increasing/decreasing score
- Design realistic computer simulations.
  - +/- reward for similarity to real videos.

Markov Decision Processes

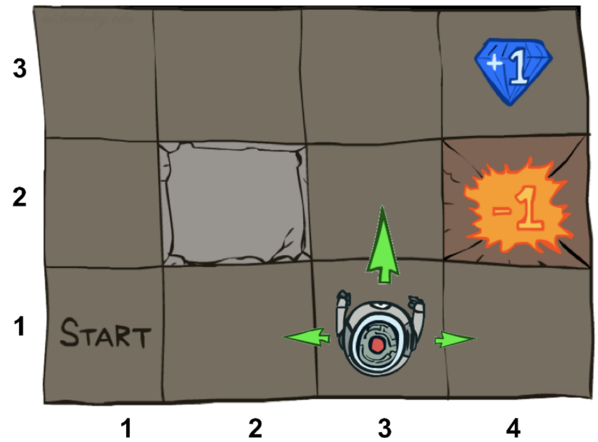


For a Markov Decision Processes (MDPs):

- Markov property $x_t = f(x_{t-1}, a_t)$ (State as memory)
- Action comes from policy and current state $a_t \Leftarrow \pi(x_t)$

Example. Gridworld

- Discrete problem (maze-like):
  - The agent lives in a grid. Discrete states = grid cells.
  - Discrete actions = *N*, *E*, *W*, *S*. Walls block motion.
  - Terminal states. The *game* ends in those cells.
- Noisy movement. For example, if the agent moves North:
  - 80% of the time, the agent goes North, if there is free space.
  - 10% of the time goes East and 10% goes West.
  - Walls block movement. Agent does not move.

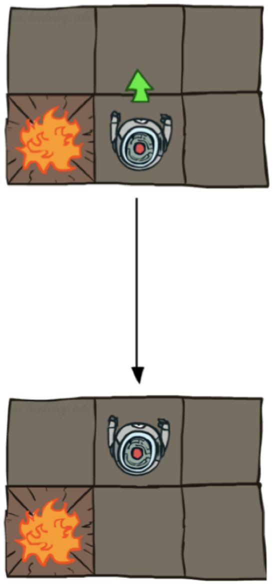


- Rewards at each step:
  - Small "living" rewards each step (positive or negative).
  - Large rewards at the end.

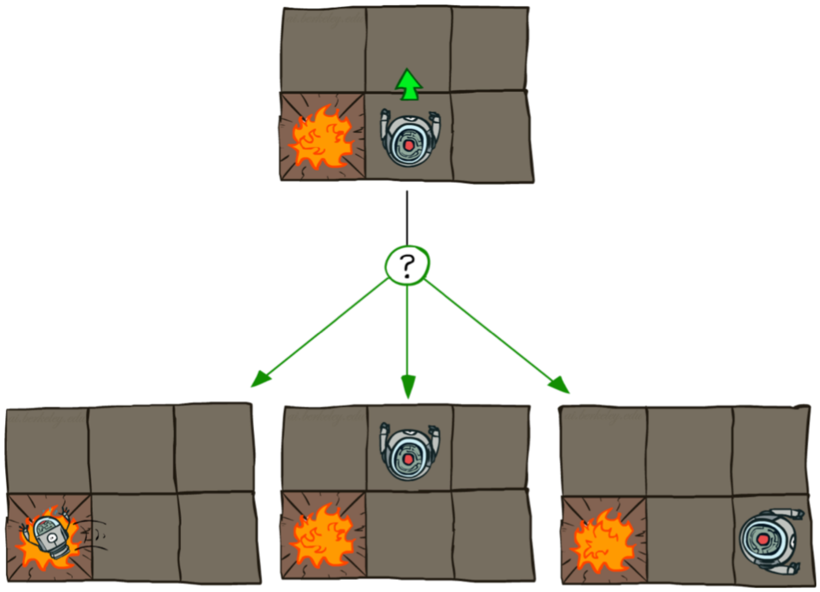
Credit: Dan Klein, Pieter Abbeel

Example. Gridworld

Deterministic motion

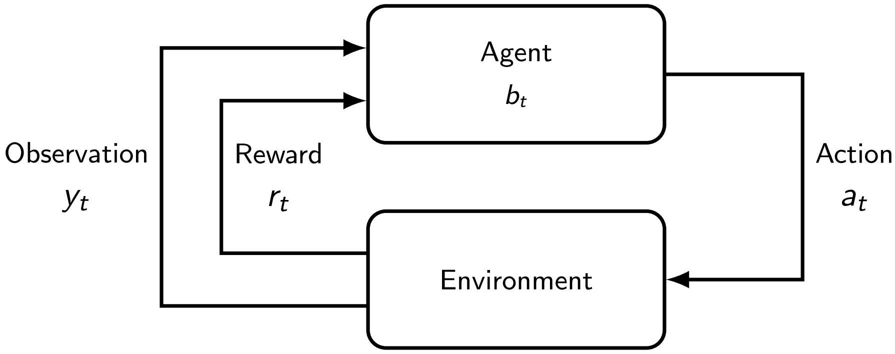


Stochastic motion



Credit: Dan Klein, Pieter Abbeel

Generalized Decision Processes

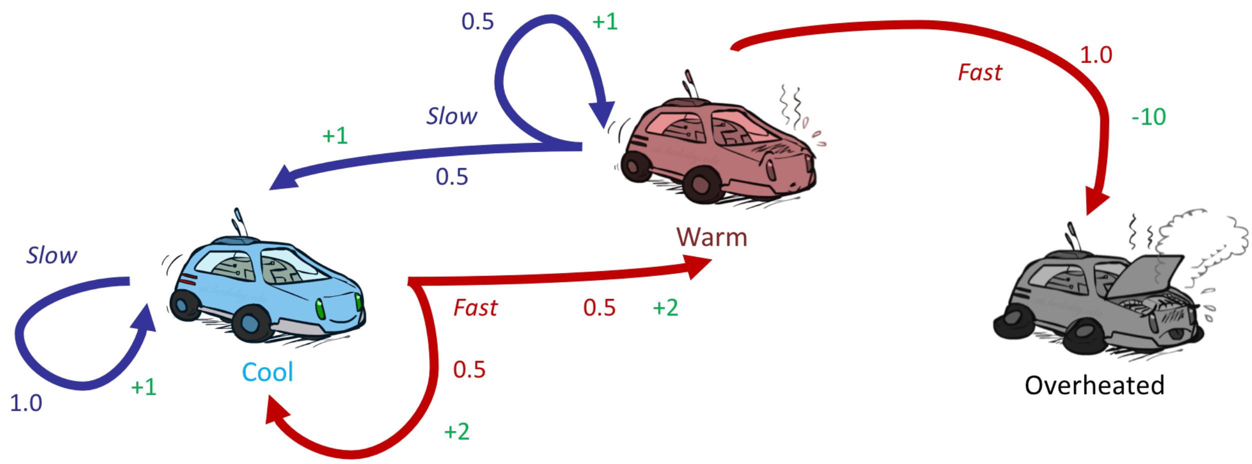


- The agent indirectly observes the environment.
- Partially Observable Markov Decision Processes (POMDPs):
  - State is *hidden*.
  - State can be estimated with beliefs/probabilities $b_t = p(x_t | y_{1:t}, a_{1:t})$.
    - Kalman filter, HMM, Monte Carlo, etc.
  - Action $a_t \Leftarrow \pi(b_t)$
- Non-Markov Decision Processes:
  - Action $a_t \Leftarrow \pi(y_t)$
  - Policy much more complicated (e.g.: deep neural networks).
  - Maybe ill-posed.

How is everything connected?

Models: environment predictions

- Transition model $p(x_{t+1}|x_t, a_t)$
- Observation model $p(y_t|x_t)$ $\leftarrow$ We assume perfect observations $x_t = y_t$
- Reward model $p(r_{t+1}|x_t, a_t)$ $\leftarrow$ We assume deterministic rewards $r_t = R(x_t, a_t, x_{t+1})$

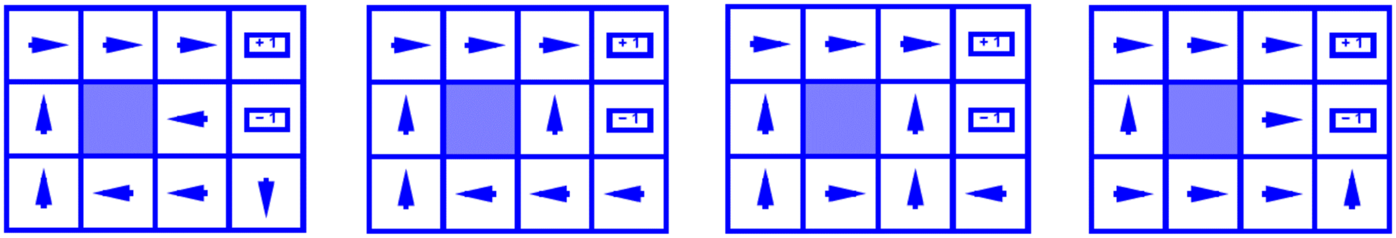


Credit: Dan Klein, Pieter Abbeel

How is everything connected?

Policy: agent behavior

- Mapping from states x_t to actions a_t. We want the optimal policy.
- Deterministic $a_t = \pi(x_t)$ or stochastic $\pi(a_t|x_t) \Rightarrow a_t \sim p(a_t|x_t)$



$R = -0.01$ $R = -0.03$ $R = -0.4$ $R = -2.0$

Credit: Dan Klein, Pieter Abbeel

How is everything connected?

Value function

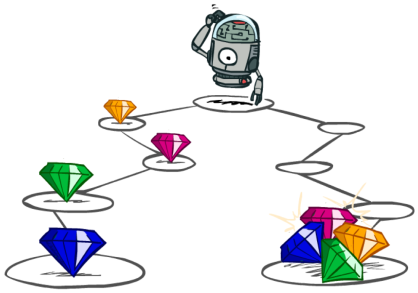
- Prediction of future rewards for a given policy.
- Informs about how good/bad is to reach a state.

$$V^{\pi}(x) = \mathbb{E}_{\pi}\left(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | x\right)$$

$$Q^{\pi}(x, a) = \mathbb{E}_{\pi}\left(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | x, a\right)$$

Discounting factor:

- Do you prefer 5$ now or 10$ in a week?
- $0 < \gamma \leq 1$
- Penalizes procrastination.

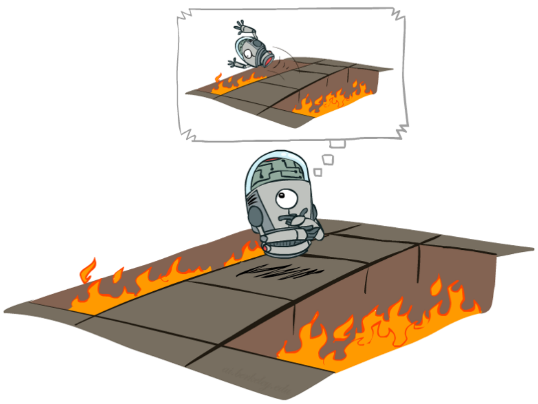


Credit: Dan Klein, Pieter Abbeel

Sequential decision making

Planning in MDPs (Offline)

- The agent has a good model of the environment.
- Everything is computed with the model. No real interaction.



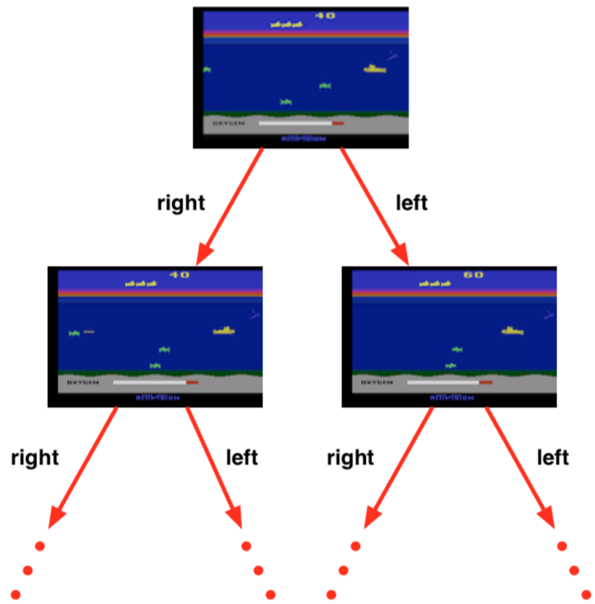
Reinforcement learning (Online)

- The agent has minimal information of the environment.
- Learning by interaction.
- Exploration and exploitation

Credit: Dan Klein, Pieter Abbeel

Video game example: Planning

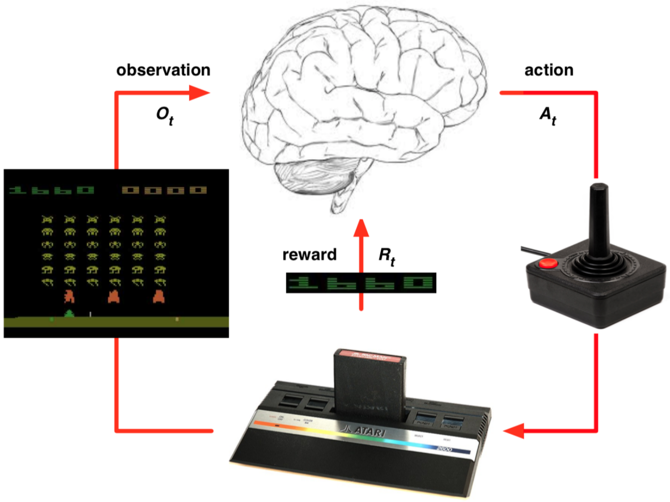
- Rules of the game are known
- We can emulate o simulate $\Rightarrow$ Access to perfect model.
- Given a state x_t and an action a_t:
  - We can predict the next state x_{t+1}
  - We can predict the reward of the step r_t.
- Find optimal policy by planning. For example: tree search for discrete systems.



Credit: David Silver

Video game example: Reinforcement learning

- Rules of the game are unknown
- Learn the rules by interaction (playing)
- Move joystick (action) $\Rightarrow$ check pixels (state) and score (reward)
- Choose good actions to improve score (exploitation).
- Choose new actions to learn about game (exploration).
- The agent must combine both.



- Example:
  - Exploitation: Go to your favorite restaurant.
  - Exploration: Try the newly open restaurant.

Credit: David Silver