Aalto University
School of Electrical Engineering

ELEC-E8125 Reinforcement learning Overview

Joni Pajarinen

4.9.2023

Today

- Introduction to planning in sequential problems
- Overview of course contents

Let’s talk about planning

- Name planning problems from your daily life

Let’s talk about planning

- Name planning problems from your daily life

- Design a plan to solve your problem

Let’s talk about planning

- Name planning problems from your daily life
- Design a plan to solve your problem
- What is a plan?

Planning and surprises

- Does your plan allow for surprises or unknowns?

Planning and surprises

- Does your plan allow for surprises or unknowns?

- How would you modify the plans to allow surprises?

Planning and surprises

- Does your plan allow for surprises or unknowns?

- How would you modify the plans to allow surprises?

- A plan can be conditional on current observation

Mapping from observation to action

$$\pi : O \rightarrow A$$
$$a = \pi(o)$$

Information needs

- Are there cases when current observation is not sufficient to make decisions? If yes, when does that happen?

Information needs

- Are there cases when current observation is not sufficient to make decisions? If yes, when does that happen?

- Sometimes history of observations is needed
- Information used for decision can be abstracted as *state*

- Give examples of state for different problems

Plan as policy

- Let’s consider that everything can be observed at time of each decision
- Plan is then a policy function from state to action

$$\pi : O \to A$$
$$a = \pi(o)$$

Plan as policy

- Let’s consider that everything can be observed at the time of each decision
- Plan is then a policy function from state to action

- Can all plans (purposeful decision strategies) be represented like this?

$$\pi : O \rightarrow A$$
$$a = \pi(o)$$

Plan as policy

- Let’s consider that everything can be observed at time of each decision
- Plan is then a policy function from state to action

- Can all plans (purposeful decision strategies) be represented like this?
  - Many can, but sometimes it’s useful to be random: some games with simultaneous moves (game theory), when we do not know the best action (exploring actions), ...

$$\pi(a|o)$$

Success

- How can you define success in planning?

Success

- How can you define success in planning?

- Reaching a particular state
- Making particular state transitions

Success

- How can you define success in planning?

- Reaching a particular state
- Making particular state transitions

- Are all plans that reach a goal equally good?

Success

- How can you define success in planning?

- Reaching a particular state
- Making particular state transitions

- Are all plans that reach a goal equally good?
- Give an example of a good and a bad plan

Objective(s)

- How can you formulate goal(s) in planning to take into account plan quality?

Objective(s)

- How can you formulate goal(s) in planning to take into account plan quality?
- Immediate reward vs cumulative return

Objective(s)

- How can you formulate goal(s) in planning to take into account plan quality?
- Immediate reward vs cumulative return
- Design rewards for your own problem

Evaluating policy quality

- Assuming that:
  - we have a policy
  - know the associated reward function
  - the system can be tested

  how can the quality of the policy be evaluated?

Planning as optimization

- Planning (sequential decision making) can be understood as *optimization of a policy with respect to expected return*

Planning as optimization

- Planning (sequential decision making) can be understood as *optimization of a policy with respect to expected return.*

- To automatically solve such problems, which information is needed? Where can the information come from?

Information for planning

- Effects of actions in different states
  - Which state I may end up to if I do X now?
- Rewards of state-action pairs
  - What's the reward if I now do X?

Why is reinforcement learning hard?

- Effects of actions (state dynamics)
  - need to be learned
  - are often stochastic
- Rewards
  - (may) need to be learned
  - may be delayed (“sparse rewards”)
  - may be difficult to choose/formulate
- Trade-off between learning (*exploration*) and maximizing rewards (*exploitation*)

Reinforcement learning problem

- Determine policy

$$a = \pi(s)$$

such that expected cumulative return is maximized

$$\pi^* = \arg\max_{\pi} \mathbb{E}[G]$$

$$G = \sum_t r_t$$

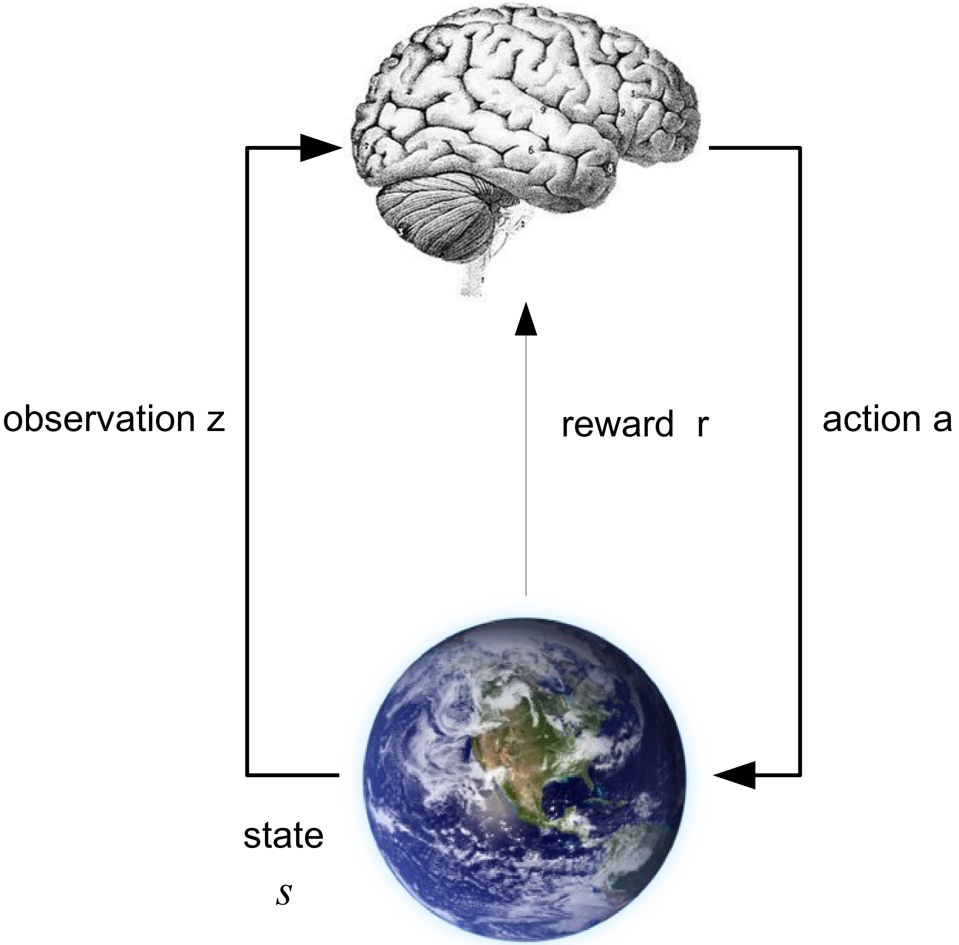
Why is reinforcement learning hard?

- States
  - Need to be defined
  - Actions' effect on states (state dynamics)
    - need to be learned
    - can be stochastic
- Rewards
  - (may) need to be learned
  - may be delayed ("sparse rewards")
  - may be difficult to choose/formulate
- Optimizing the policy: trade-off between learning (exploration) and maximizing rewards (exploitation)

Summary so far

- Can you
  - explain what is reinforcement learning
  - define a problem as a reinforcement learning problem
  - explain why reinforcement learning is difficult

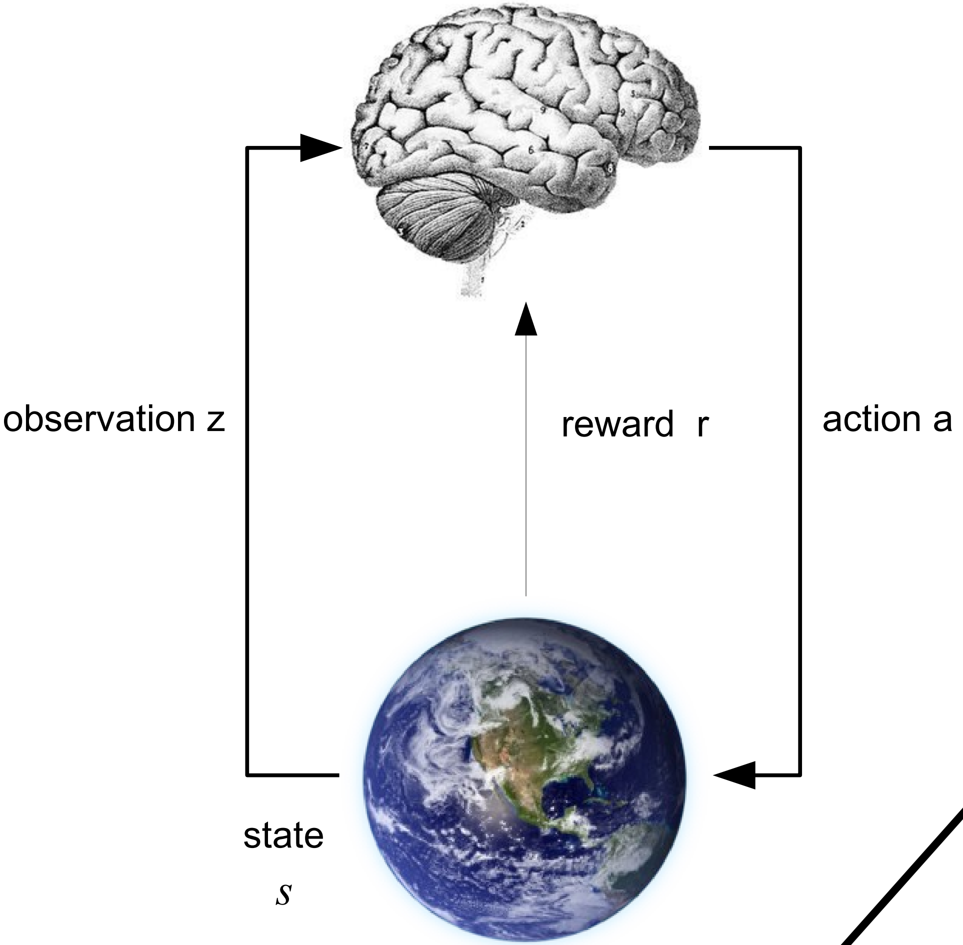
Setting



Task
Choose a sequence of actions that maximizes cumulative reward.

Note: observations are often denoted with o or z

Markov decision process



Can you explain what Markovian means?

MDP
Environment observable
$z = s$

Defined by dynamics
$P(s_{t+1} | s_t, a_t)$

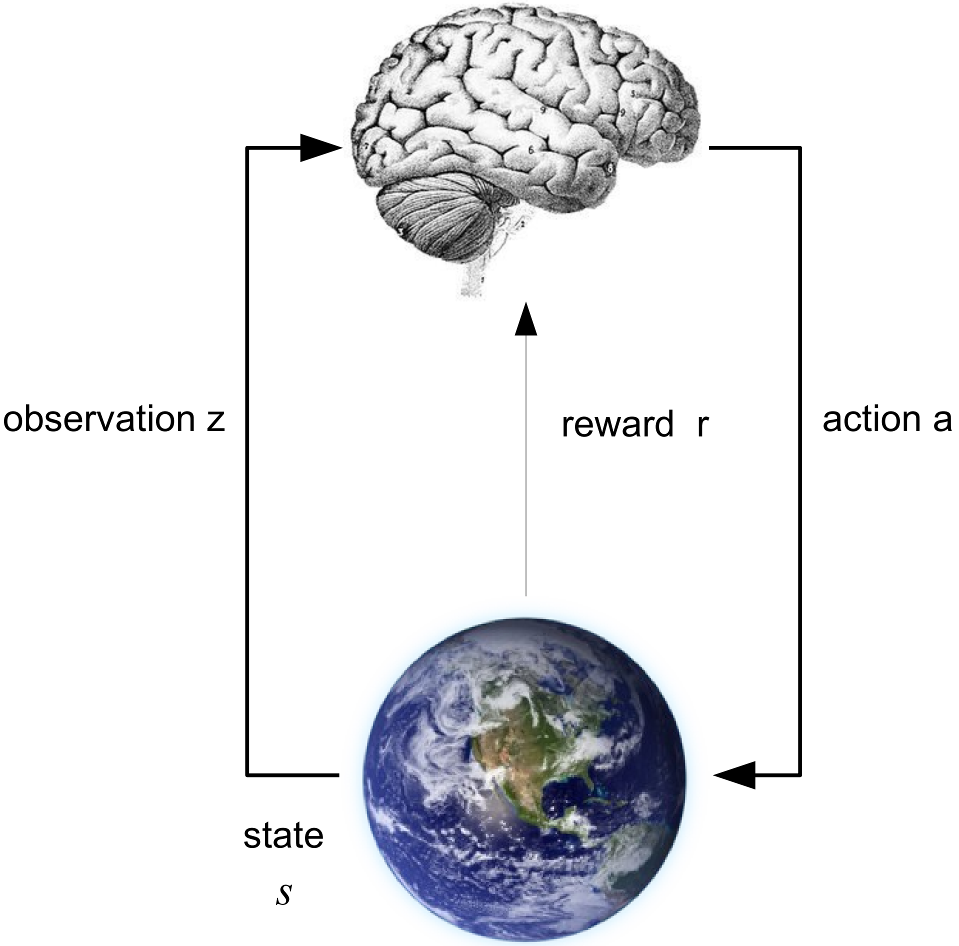
And reward function
$r_t = r(s_t, a_t)$

Solution, for example
$a^*_{1,\dots,T} = \arg\max_{a_1,\dots,a_T} \sum_{t=1}^{T} r_t$

Represented as policy
$a = \pi(s)$

[Sutton & Barto, 2018] uses also $r_{t+1} = r(s_t, a_t, s_{t+1})$ *.*
We use $r_t = r(s_t, a_t)$ *. There is no fundamental difference.*
Both can be used.

Reinforcement learning



RL (reinforcement learning)
MDP with **unknown**
Markovian dynamics

$P(s_{t+1}|s_t, a_t)$

Unknown reward function

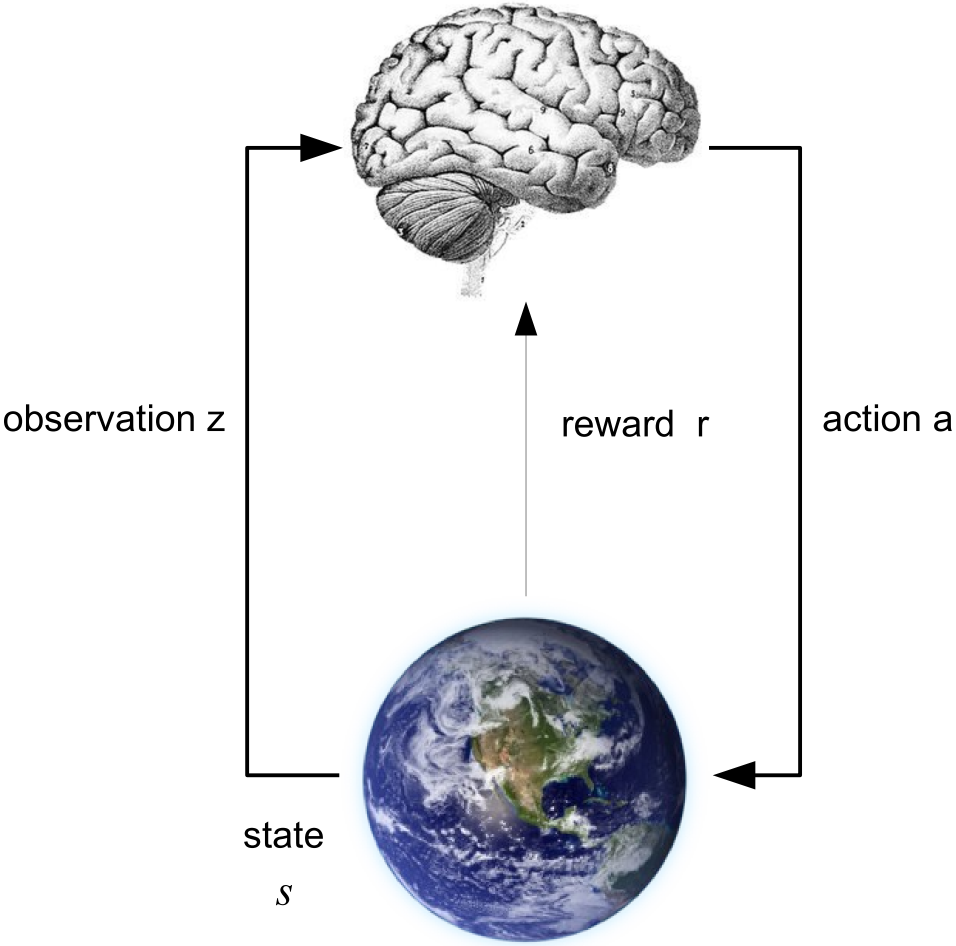
$r_t = r(s_t, a_t)$

Solution similar, for example

$a^*_{1,\ldots,T} = arg\ max_{a_1,\ldots,a_T} \sum_{t=1}^{T} r_t$

Learning must **explore** policies

Partially observable MDP (POMDP)



POMDP
Environment not directly observable

Defined by dynamics
$P(s_{t+1}|s_t, a_t)$

Reward function
$r_t = r(s_t, a_t)$

Observation model
$P(z_{t+1}|s_{t+1}, a_t)$

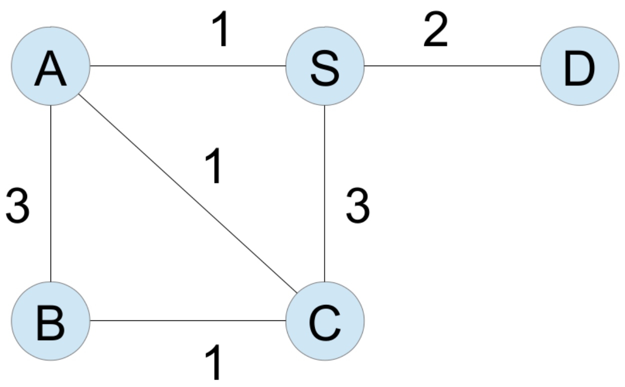
Solution similar, for example
$a^*_{1,\dots,T} = arg\ max_{a_1,\dots,a_T} \sum_{t=1}^{T} r_t$

Course outline

- Markov decision processes

- Reinforcement learning
  - Value-based, policy-based, model-based
  - Exploitation and exploration
  - Partially observable Markov decision processes

Toward optimal planning in time series: Shortest paths



Shortest path from S to B?

- Associate each edge of graph with nonnegative cost
- Cost of plan is the sum of costs

Optimal planning (fixed-length plans)

- Cost functional

$$L(\tau_K) = \sum_{k=1}^{K} l(s_k, a_k) + l_F(s_{K+1})$$

$$\tau_K = (a_1, \ldots, a_K)$$

$$l_F(s) = \begin{matrix} 0, s \in S_G \\ \infty, s \notin S_G \end{matrix}$$

Goal set

- Goal: $min_{\tau} L(\tau)$

Solving optimal planning

- Principle of Optimality (Bellman, 1957): An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

- Value-function:
  - cost-to-go $G_k^*(s_k)$

$$G_k^*(s_k) = min_{a_k,\dots,a_K}\left\{\sum_{i=k}^{K} l(s_i,a_i)+l_F(s_F)\right\}$$

Backward value iteration

- Assume we know $G^*_{k+1}(s)$

  how to compute $G^*_k(s_k)$?

Richard Bellman
https://mathshistory.st-andrews.ac.uk/Biographies/Bellman/poster/lived/

$$G^*_k(s_k) = min_{a_k,\ldots,a_K}\left\{\sum_{i=k}^{K} l(s_i, a_i) + l_F(s_F)\right\}$$

$$\begin{aligned} G^*_k(s_k) &= min_{a_k}\left\{l(s_k, a_k) + G^*_{k+1}(s_{k+1})\right\} \\ &= min_{a_k}\left\{l(s_k, a_k) + G^*_{k+1}(f(s_k, a_k))\right\} \end{aligned}$$

$$G^*_{K+1}(s) = l_F(s)$$

Value iteration, unknown length plans

- Iterating recursion until value function stationary: optimal cost plans have been found from all states that can reach a goal state

$$G^*(s) = min_a \left\{ l(s,a) + G^*(f(s,a)) \right\}$$

Bellman equation

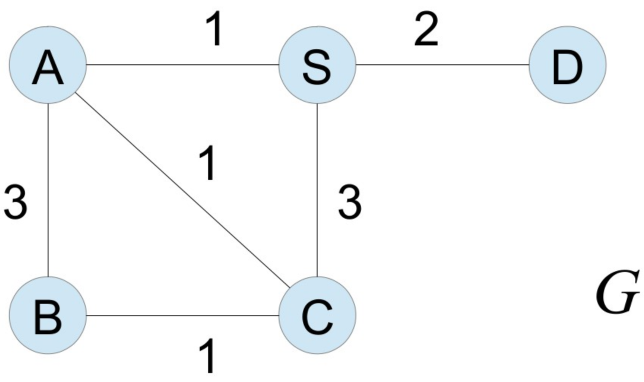
- Using G^*, optimal actions can be found from

$$a^* = argmin_{a \in A(s)} \left\{ l(s,a) + G^*(f(s,a)) \right\}$$

Exercise

- Use backward value iteration for

$$s_I = S$$
$$S_G = \{B\}$$



Reminder:

$$G^*(s) = min_a \{ l(s,a) + G^*(f(s,a)) \}$$

Aalto University
School of Electrical
Engineering

Next time: Markov decision processes

- Sutton & Barto, chapters 2-2.3, 2.5-2.6, 3-3.8 due one week from now
- Complete Quiz 1 before next week's lecture (quiz will open on Tuesday)