Aalto University
School of Electrical Engineering

ELEC-E8125 Reinforcement Learning Reinforcement learning in discrete domains

Joni Pajarinen

19.9.2023

Today

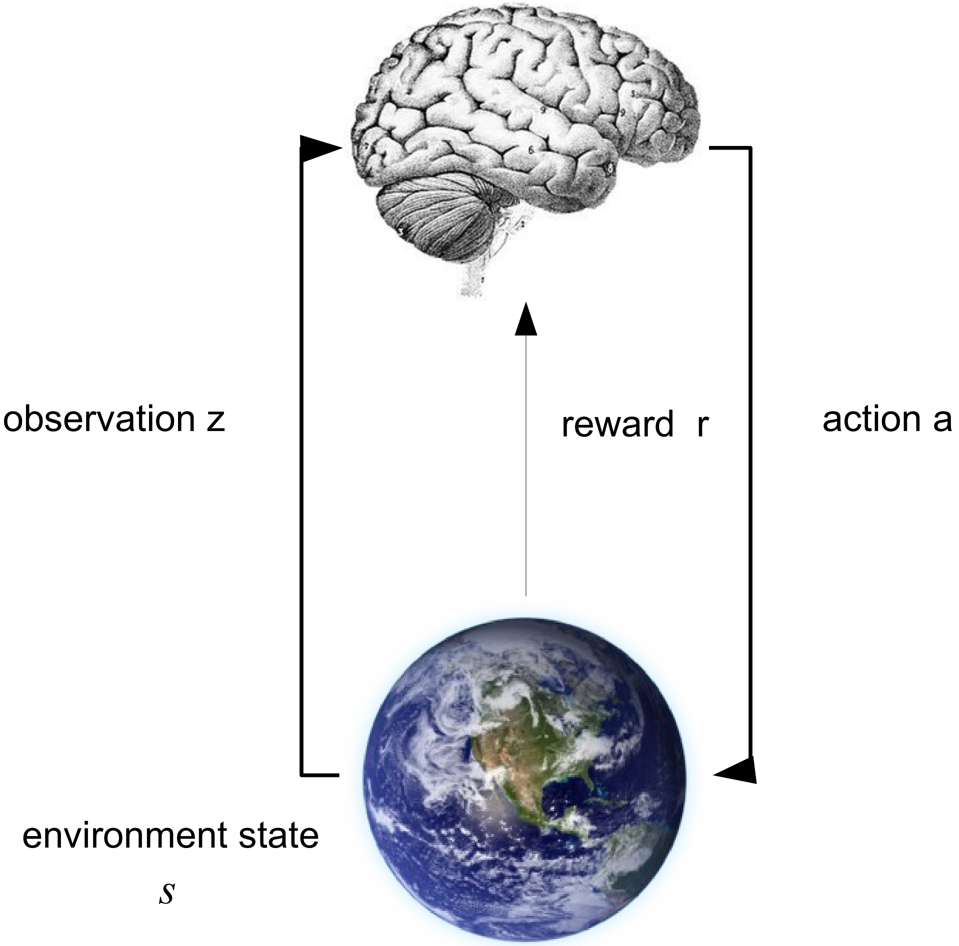
- Reinforcement learning
- Policy evaluation vs control problems
- Monte-Carlo and Temporal difference

Learning goals

- Understand basic concepts of RL
- Understand Monte-Carlo (MC) and temporal difference (TD) approaches for policy evaluation and control

- Be able to implement MC and TD

Reinforcement learning



RL
Markov decision process (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, e.g.
$a^*_{1,\dots,T} = max_{a_1,\dots,a_T} \sum_{t=1}^{T} r_t$

Learning must **explore** policies

Reinforcement learning

- MDP with unknown dynamics (*P*) and reward function (*r*)
- Model based RL: Estimate MDP, apply MDP methods
  - Estimate MDP dynamics and reward functions from data
- Can we do without *P* and *r* ?
  - Can we evaluate a policy (construct value function) if we have multiple episodes (in episodic tasks) available?

Monte-Carlo policy evaluation

- Complete episodes give us samples of return *G*
- Learn value of particular policy from episodes under that policy:

$$V_\pi(s) = E_\pi\left[G_t \middle| s_t = s\right] \qquad G_t = \sum_{k=0}^{H} \gamma^k r_{t+k}$$

- Estimate value as empirical mean return
  - For each visited state *s* in an episode,

$$N(s) = N(s) + 1 \quad S(s) = S(s) + G_t \quad V(s) \approx S(s)/N(s)$$

- When number of episodes approaches infinity, *V(s)* converges: $V(s) \rightarrow V_\pi(s)$

Empirical mean approaches true mean.

Can we do without episodes?

Temporal difference (TD) – learning without episodes

- For each state transition, update a guess towards a guess:

$$V(s_t) = V(s_t) + \alpha\left(\underline{r_t + \gamma V(s_{t+1})} - V(s_t)\right)$$

- Approach called TD(0)

Estimated return

- Compare to MC

$$V(s_t) = V(s_t) + \alpha\left(G_t - V(s_t)\right)$$

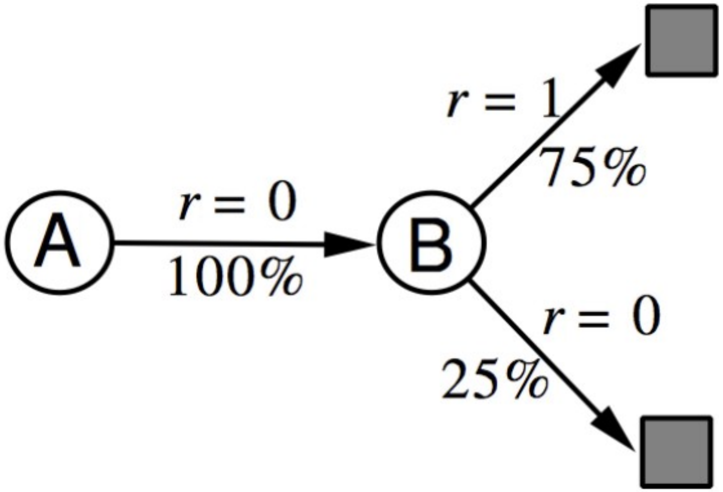
True return

What if we have limited data?

Batch learning

- For a limited number of trials available:
  - Sample episode *k*
  - Apply MC or TD(0) to episode *k.*

A, 0, B, 0
B, 1
B, 1
B, 1
B, 1
B, 1
B, 1
B, 0



What is *V(A)*?

Beware small sample numbers.

MC vs TD

- MC
  - Needs full episodes. Only works in episodic environments
  - High variance, zero bias → good but slow convergence
  - Does not exploit Markov property → often better in non-Markov environments

- TD (esp. TD(0))
  - Can learn from incomplete episodes and online after each time step
  - Works in continuing non-episodic environments
  - Low variance, some bias → often more efficient than MC, discrete state TD(0) converges, more sensitive to initial value
  - Exploits Markov property → often more efficient in Markov environments

Can we combine these / find an intermediate solution?

λ-return

- Combine returns in different horizons

$$G_t^{\lambda} = (1-\lambda)\sum_{k=0}^{\infty} \lambda^{k} G_t^{(k)}$$

k-step return: $G_t^{(k)} = \sum_{i=0}^{k-1} \gamma^{i} r_{t+i} + \gamma^{k} V(s_{t+k})$

- State value function update becomes

$$V(s_t) = V(s_t) + \alpha\left(G_t^{\lambda} - V(s_t)\right)$$

- This is called TD(λ) based on the *forward view*

Requires complete episodes! Can we survive without?
First: an alternative viewpoint!

Causes and effects – eligibility traces

- Which state is the “cause” of a reward?
- Frequency heuristic: most frequent states likely
- Recency heuristic: most recent states likely

- *Eligibility trace* for a state combines these:

$$E_t(s) = \gamma \lambda E_{t-1}(s) + \mathbf{1}(s_t = s)$$

“How often a particular state was visited recently on average?”

Not exactly, also considers discount.

Backward-TD(λ)

- Extend TD time horizon with decay (λ)
- After episode, update

$$V(s) = V(s) + \alpha E_t(s)\left(r_t + \gamma V(s_{t+1}) - V(s_t)\right)$$

- TD(1) equal to MC

$$E_t(s) = \gamma \lambda E_{t-1}(s) + \mathbf{1}(s_t = s)$$

- Eligibility traces way to implement *backward* TD(λ), *forward* TD(λ) requires episodes

Slightly different in online case.

What if λ=0?

Control / decision making?

- So far we only found out how to estimate value functions for a particular policy

- Can we use this to optimize a policy?

Policy improvement and policy iteration

- Given a policy π , it can be improved by
  - Evaluating its value function
  - Forming a new policy by acting greedily with respect to the value function

- This always improves the policy

- Iterating multiple times called *policy* iteration
  - Converges to optimal policy

Monte-Carlo Policy iteration

- Can we choose action using value function $V(s)$?

- Greedy policy improvement using action-value function *Q(s,a)* does not require a model:

$$\pi'(s) = arg\, max_a Q(s, a)$$

- Estimate *Q(s,a)* using MC (empirical mean = “calculate average”)

Note: calculate frequencies for all state-action pairs.

Exploration-exploitation trade-off:
How can we ensure that we try different actions?

Ensuring exploration

Note: in exercise 3 task 1, we will use epsilon greedy exploration

- Simple approach: ε-greedy exploration:
  - Explore: Choose action at random with probability ε
  - Exploit: Be greedy with probability 1 - ε

$$\pi(a|s) = \begin{cases} 1-\epsilon & \textit{if } a = \arg\max_{a'} Q(s,a') \\ \epsilon/(N_{actions}-1) & \textit{for any other action} \end{cases}$$

Number of different actions

- How to converge to optimal policy?
  - Idea: reduce ε over time.
  - For example, for *k*:th episode $\epsilon = \frac{b}{b+k}$

"Greedy in the Limit with Infinite Exploration" (GLIE)

constant

Can we use TD instead of MC for control?

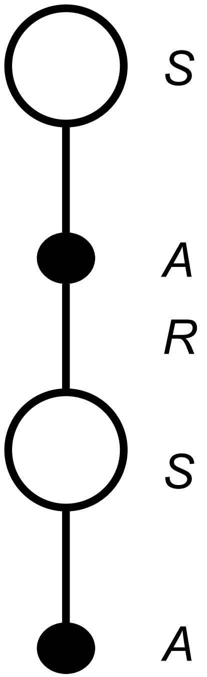
SARSA

- Idea: Apply TD to *Q(S,A)*
  - With ε-greedy policy improvement
  - Update each time step

$$Q(s,a)=Q(s,a)+\alpha\left(r+\gamma Q(s',a')-Q(s,a)\right)$$

Compare with

$$V(s_t)=V(s_t)+\alpha\left(r_t+\gamma V(s_{t+1})-V(s_t)\right)$$



- SARSA converges under
  - GLIE policy (greedy in the limit of infinite exploration),
  - $\sum_{t=0}^{\infty}\alpha_t=\infty \quad \sum_{t=0}^{\infty}\alpha_t^2<\infty$

For example $\alpha_t=1/t$

SARSA(λ)

- Instead of TD(0) update in SARSA, use TD(λ) update
- Backward SARSA(λ)

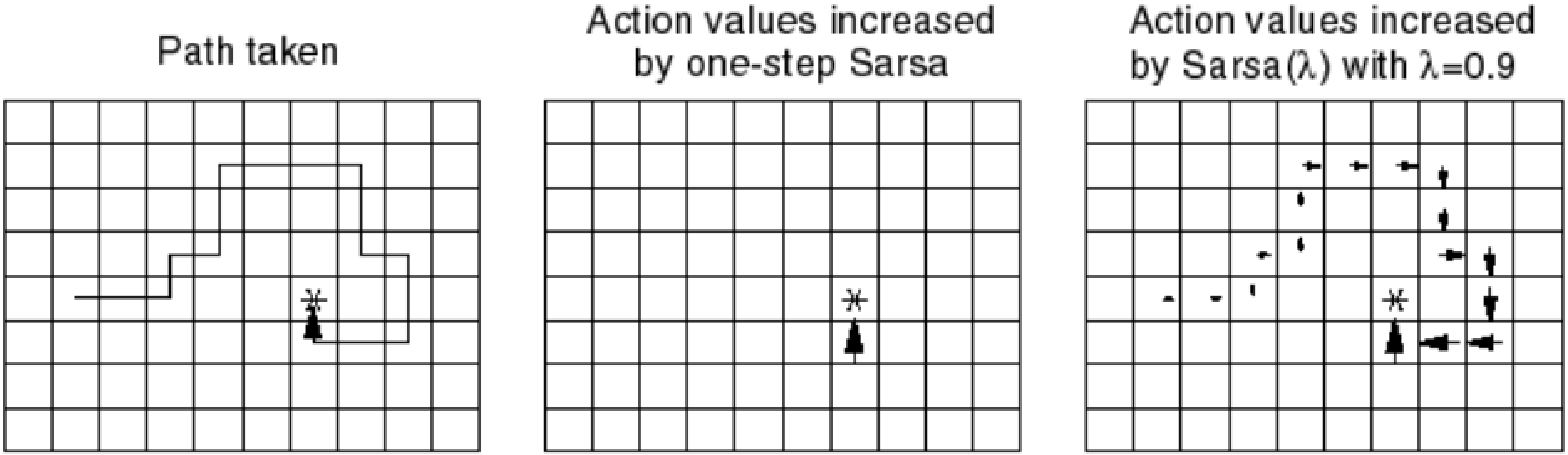
$$E_t(s,a)=\gamma\lambda E_{t-1}(s,a)+\mathbf{1}(s_t=s,a_t=a)$$

$$Q(s,a)=Q(s,a)+\alpha E_t(s,a)\left(r_t+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right)$$

Compare to

$$E_t(s)=\gamma\lambda E_{t-1}(s)+\mathbf{1}(s_t=s)$$

$$V(s)=V(s)+\alpha E_t(s)\left(r_t+\gamma V(s_{t+1})-V(s_t)\right)$$



Path taken

Action values increased by one-step Sarsa

Action values increased by Sarsa(λ) with λ=0.9

On-policy vs off-policy learning

- *On-policy learning* (methods so far)
  - Use a policy while learning how to optimize it
  - “Learn on the job”
- *Off-policy learning*
  - Use another policy while learning about optimal policy
  - Can learn from observation of other agents
  - Can learn about optimal policy when using exploration policy

Q-learning

Note: in exercise 3 task 1,
we will implement Q-learning

- Use ε-greedy *behavior policy* to choose actions
- *Target policy* is greedy with respect to Q

$$\pi(s) = arg\,max_a Q(s,a)$$

- Update target policy greedily:

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma\, max_{\hat{a}} Q(s', \hat{a}) - Q(s,a)\right)$$

Assume we take greedy action at next step.

- Q converges to Q*

Summary

- In reinforcement learning, dynamics and reward function of the MDP are in general unknown

- MC (Monte-Carlo) approaches sample returns from full episodes
- TD (temporal difference) approaches sample estimated returns (biased)
- Returns can be used to update a policy or value function

Next: Extending state spaces

- What to do if
  - discrete state space is too large?
  - state space is continuous?

- Readings
  - Sutton & Barto, ch. 9-9.3, 10-10.1