

Reinforcement Learning

Exercise 5: Model-free Control

Nico Meyer

Overview

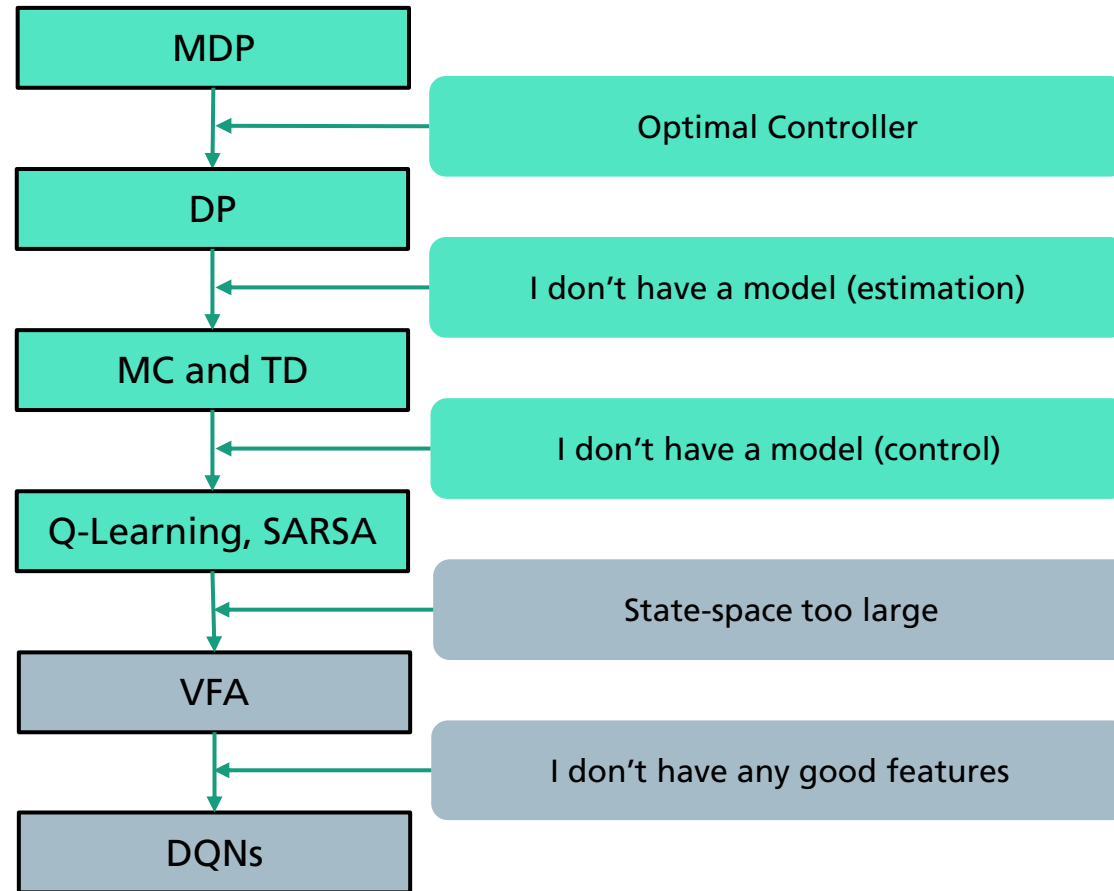
Exercise Content

<i>Week</i>	<i>Date</i>	<i>Topic</i>	<i>Material</i>	<i>Who?</i>
1	22.04.		<i>no exercises</i>	
2	29.04.	MDPs (slides)	ex1.pdf	Nico
3	06.05.	T.B.D.		
4	13.05.	Dynamic Programming (slides)	ex2.pdf, ex2_skeleton.zip	Alex
5	20.05.	OpenAI Gym, PyTorch-Intro (slides) TD-Learning (slides)		Nico
6	27.05.	TD-Control (slides)		Nico
7	03.06.	Intermediate exam		
8	10.06.		<i>no exercises</i>	
9	17.06.	DQN (slides)		Nico
10	24.06.	VPG (slides)		Alex
11	01.07.	A2C (slides)		Nico
12	08.07.	Multi-armed Bandits (slides)		Alex
13	15.07.	RND/ICM (slides)		Alex
14	22.07.	MCTS (slides)		Alex



Overview

Overall Picture



Model-free Control

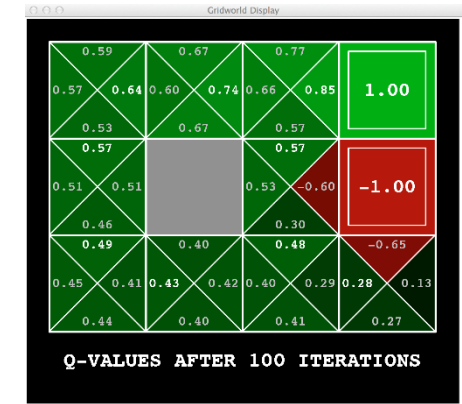
TD Methods



Recap

State-action-value function

$$s \xrightarrow{a, r_0} s_1 \xrightarrow{\pi(s_1), r_1} s_2 \xrightarrow{\pi(s_2), r_2} s_3 \dots s_{h-1} \xrightarrow{\pi(s_{h-1}), r_{h-1}} s_h$$
$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a \right]$$



Greedy Policy Improvement over Q:

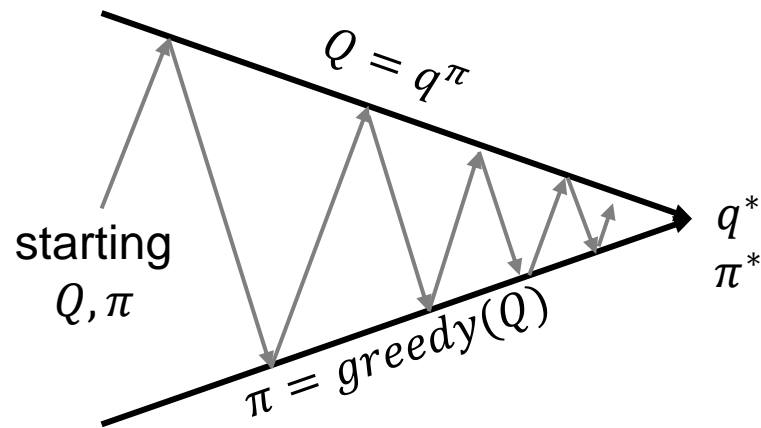
$$\pi'(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^\pi(s, a)$$

$$\forall s \in \mathcal{S}, \quad Q^{\pi'}(s, \pi'(s)) \geq Q^\pi(s, \pi(s))$$

Recap

Model-free Control

- The (model-free) control problem:
 - **Given** experience samples $s(s, a, r, s')$
 - **Learn** a close-to optimal policy π
- Simple idea:
 - If we have calculated the value function for a given policy π (e.g., from MC/TD policy evaluation from last week), we can use it for deriving a better policy π' through greedy policy improvement over $Q(s)$



Policy Evaluation: Estimate $Q = q_\pi$
e.g., Monte Carlo Policy Evaluation

Policy Improvement: Generate $\pi' \geq \pi$
e.g., Greedy Policy Improvement over Q

Recap

Q-Learning and SARSA Algorithms

Problem:

We do not know \mathcal{P} or \mathcal{R} or both of the MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$

Solution:

Model-free methods that use experience samples $s(s, a, r, s')$

In Exercise 4 we did:

Model-free Prediction: Evaluate the future, given the policy π .
(estimate the value function)

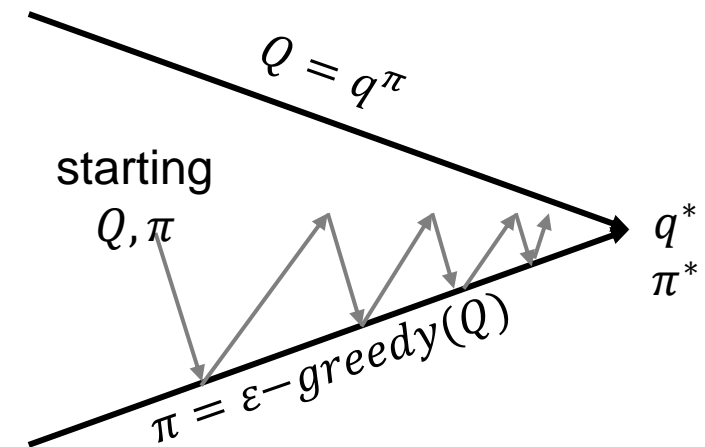
In Exercise 5 we will do:

Model-free Control: Optimize the future by finding the best policy π .
(optimize the value function)

Update every time step:

Policy Evaluation: Estimate $Q \approx q_\pi$
e.g., SARSA, Q-learning

Policy Improvement: Generate $\pi' \geq \pi$
e.g., ϵ -greedy Policy Improvement over Q



Recap

SARSA: On-policy control

- Apply TD to $Q(s, a)$
- Use ε -greedy policy improvement
- Update at every time-step

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Loop for each step of episode:

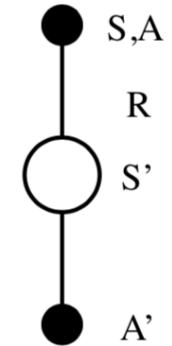
 Take action A , observe R, S'

 Choose A' from S' using policy derived from Q (e.g., ε -greedy)

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma Q(S', A') - Q(S, A)]$

$S \leftarrow S'; A \leftarrow A';$

 until S is terminal

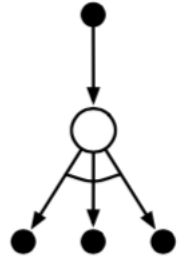


Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Recap

Q-learning: Off-policy control

- Evaluate one policy while following another
- Can re-use experience gathered from old policies



Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Take action A , observe R, S'

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$

$S \leftarrow S'$

 until S is terminal

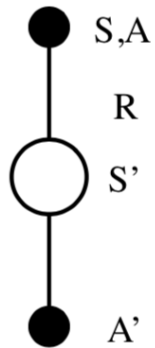
Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Recap

Q-Learning vs. SARSA

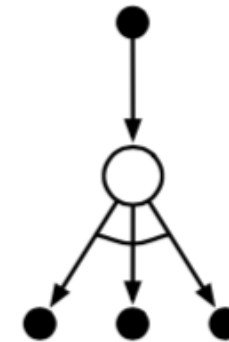
SARSA algorithm (on-policy control)

- + Processes each sample immediately
- + Minimal update cost per sample
- Requires a huge number of samples
- Requires careful schedule for the learning rate
- Makes minimal use of each sample
- The ordering of samples influences the outcome
- Exhibits instabilities under approximate representations
- Poses constraints on sample collection (on-policy)
- Requires careful handling on the policy greediness



Q-Learning algorithm (off-policy control)

- + Processes each sample immediately
- + Minimal update cost per sample
- + Poses no constraints on sample collection (off-policy)
- Requires a huge number of samples
- Requires careful schedule for the learning rate
- Makes minimal use of each sample
- The ordering of samples influences the outcome
- Exhibits (even more) instabilities under approximate representations



Epsilon-greedy policy

Why should we follow an ϵ -greedy policy? Isn't this suboptimal?

Exercise Sheet 5

Model-free Control



Thank you for your attention!