Machine Learning

Reinforcement Learning

Karol Przystalski

April 6, 2022

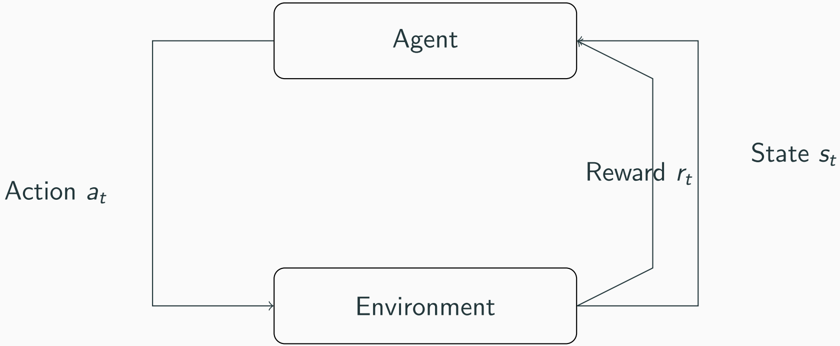
Department of Information Technologies, Jagiellonian University

Agenda

1. Introduction
2. Markov Decision Process
3. Reinforcement learning methods

Introduction

Reinforcement learning cycle



$$R_t = r_{t+1} + \gamma r_{t+1} + \gamma^2 r_{t+3} + \ldots + \gamma^{k-1} r_k + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (1)$$

where γ is closer to 0 than the distance we look into the future is smaller.

State

	Next state			
Current state	A	B	C	D
A	-1	2	3	-
B	1	-1	1	2
C	0	1	-1	0
D	1	-	1	-1

Terms

A few terms to remember:

- $V(s)$ – value of state,
- $Q(s, a)$ – action-value function.

$$V(s) = E(r_t | s_t = s) = E\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s\} \tag{2}$$

$$Q(s, a) = E(r_t | s_t = s, a_t = a) = E\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s, a_t = a\} \tag{3}$$

Action selection

There are many strategies how to select the next action. The most popular:

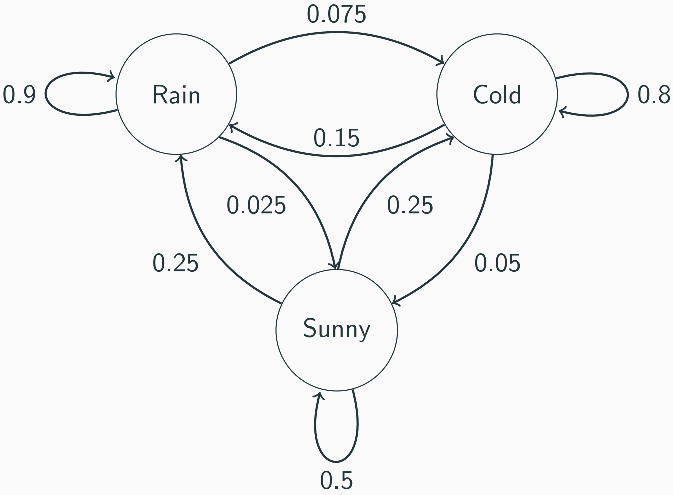
- Greedy – just pick the highest value of $Q_{s,t}(a)$,
- ε-greedy – we have a small probability ε that allow us to pick some other action at random,
- soft-max – instead of ε we have have a more sophisticated solution for alternative paths; the selection can be made by:

$$P(Q(_{s,t}(a)) = \frac{\exp(Q_{s,t}(a)/\tau}{\sum_b \exp(Q_{s,t}(b)/\tau}, \tag{4}$$

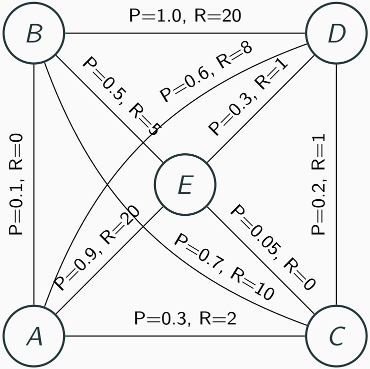
  where τ is the temperature. When τ is high, all actions have similar probability.

Markov Decision Process

Markov Chain



Markov Decision Process



$$Pr(r_t = r', s_{t+1} = s' | s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, \ldots, r_1, s_1, a_1, r_0, s_0, a_0) \quad (5)$$

Reinforcement learning methods

RL methods

There many RL methods, but the most popular are:

- Q-learning,
- SARSA,
- Deep Q-Netowrk,
- Deep Deterministic Policy Gradient.

Q-learning

The q-learning method consist of steps:

1. init the $Q(s, a)$ to small random values for all s and a,
2. select action a using the ε-greedy strategy,
3. take action a and receive reward r,
4. sample new state s',
5. update $Q(s, a) \leftarrow Q(s, a) + \mu(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$,
6. set $s \leftarrow s'$, $a \leq$
7. repeat from step 2 until there no more episodes.

SARSA

SARSA is acronym for State-Action-Reward-State-Action. It consist of following steps:

1. init the $Q(s, a)$ to small random values for all s and a,
2. select action a using the best strategy,
3. take action a and receive reward r,
4. sample new state s',
5. update $Q(s, a) \leftarrow Q(s, a) + \mu(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$,
6. set $s \leftarrow s'$, $a \leftarrow a'$,
7. repeat from step 2 until there no more episodes.

Questions?