Statistical Data Analysis

Jana de Wiljes

Lecture 19

Idea behind Reinforcement Learning

Reach a specific target goal via sensible choice of actions (which are improved via successive feedback)

Example:

- find best strategy for games such as Tic Tac Toe, connect four, chess, ...
- find the fastest path out of a maze
- let robots learn to perform simple tasks by themselves
- train autonomous cars
- optimale treatment of patients

Alpha Go

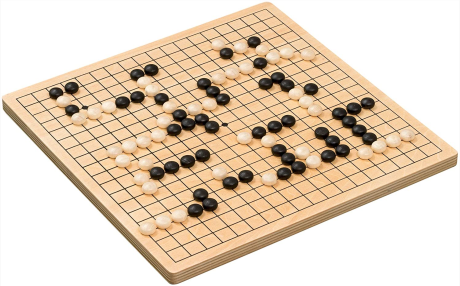
Computerprogramm trained via RL that plays board game Go

Fakts:

- developed by DeepMind
- 2015 won against Europamaster in Go
- 2016 beats World champion Lee Sedol in Go
- documentation `https://www.youtube.com/watch?v=WXuK6gekU1Y`

Go

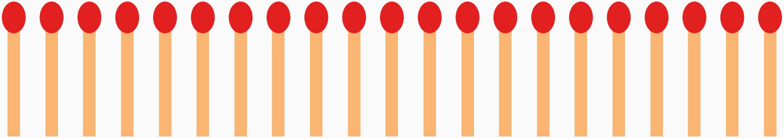
Simple rules
however very
complex strategies



Facts:

- one of the oldes gamest in the world
- popular sport in many asian countries
- there are 3^{361} combinations to place the stones on the 19x19 broad
- huge challenge for computer players due to the many combinations

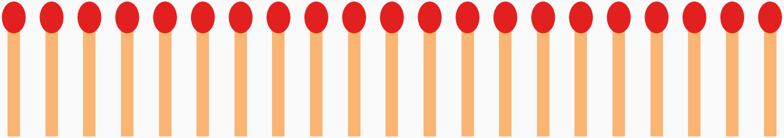
Matchstick Game



Matchstick Game

Player 1

Player 2



Matchstick Game

Player 1

Player 2

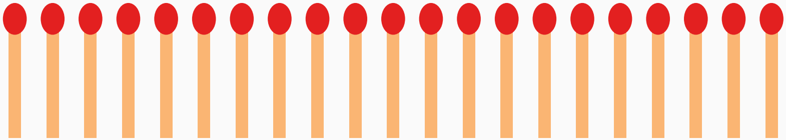
Players take turns choosing to take 1, 2 or 3 matchsticks

The Player that takes the last match looses

t = 0

Matchstick Game

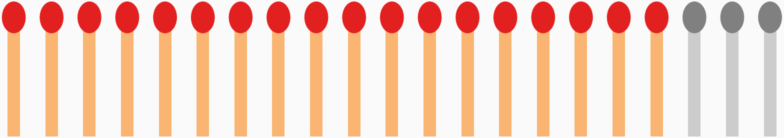
Player 1: takes 3 matchsticks



t = 0

Matchstick Game

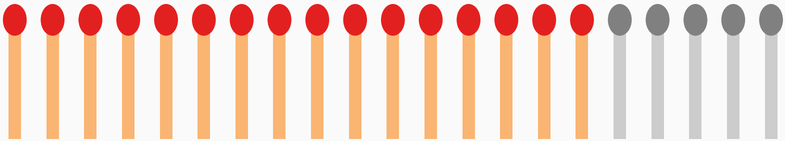
Player 1: takes 3 matchsticks



$t = 1$

Matchstick Game

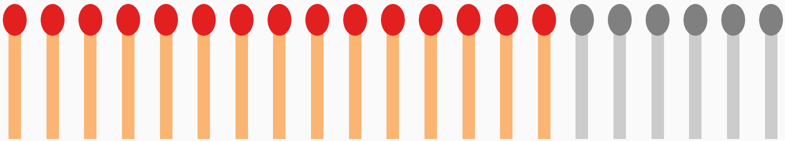
Player 2: takes 2 matchsticks



$t = 1$

Matchstick Game

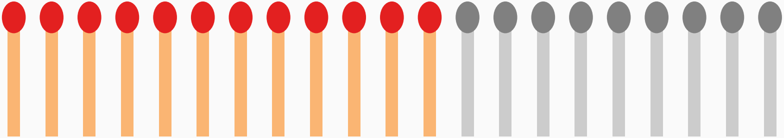
Player 1: 1 matchstick



t = 2

Matchstick Game

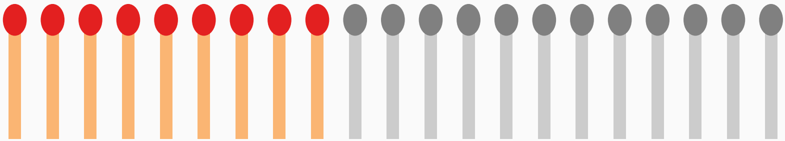
Player 2: 3 matchsticks



$t = 2$

Matchstick Game

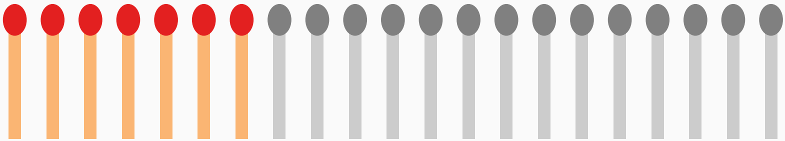
Player 1: 3 matchsticks



$t = 3$

Matchstick Game

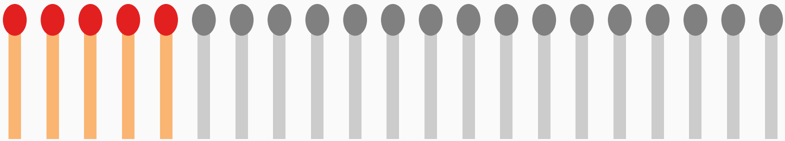
Player 2: 2 matchsticks



t = 3

Matchstick Game

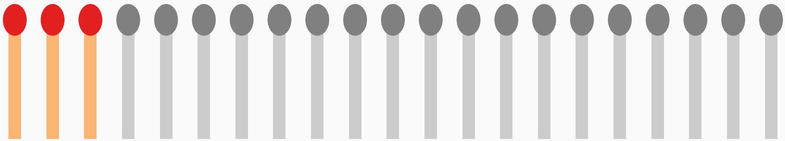
Player 1: 2 matchsticks



$t = 4$

Matchstick Game

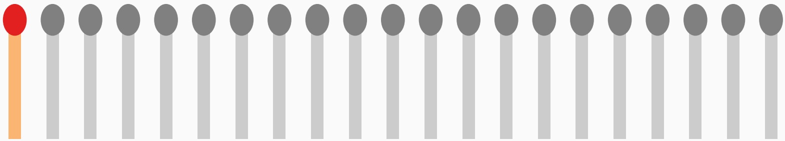
Player 2: 2 matchsticks



$t = 4$

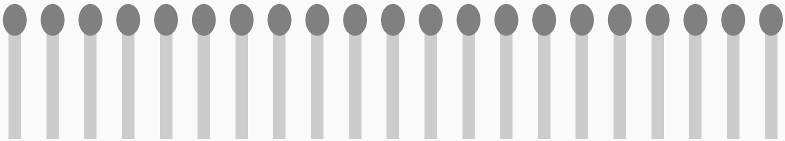
Matchstick Game

Player 1: 2 matchsticks



t = 5

Player 2: 1 matchstick

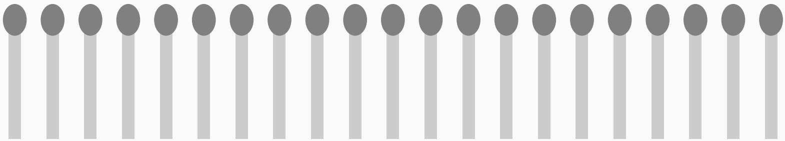


t = 5

Matchstick Game

Player 1: wins

Player 1 wins, since Player 2 took the last matchstick.



$\mathtt{t} = 6$

RL learning procedure

Interaction model of RL in the context of the matchstick game

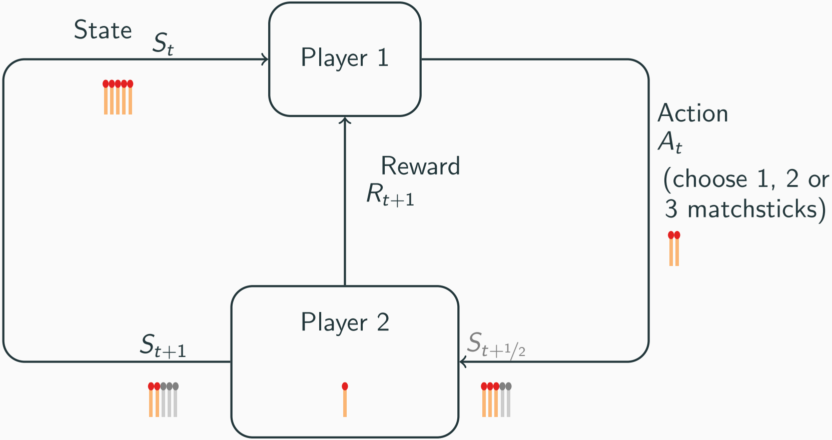


Figure adapted from Sutton & Barto (2018) Reinforcement Learning

Examples for potential rewards

- Goal: get out of a maze:
  + positive reward for finding the exit
  - negative reward if the path has been crossed before
- Goal: win in chess:
  + positive Belohnung if a figure has been beat
  - negative reward if you lost on of the figures
- Manage a power plant:
  + positive Belohnung for a specific amount of energy production
  - negative reward for core melt accident
- Goal: training a robot to walk:
  + positive reward for movement
  - negative reward for falling down
- Goal: be successful in playing Atari-games:
  + positive reward for beating the highscores
  - negative reward for loosing the game

Interaktion Modell

Agent decides which action (A_t) to take conditioned on the feedback of the environment (S_t & R_t):

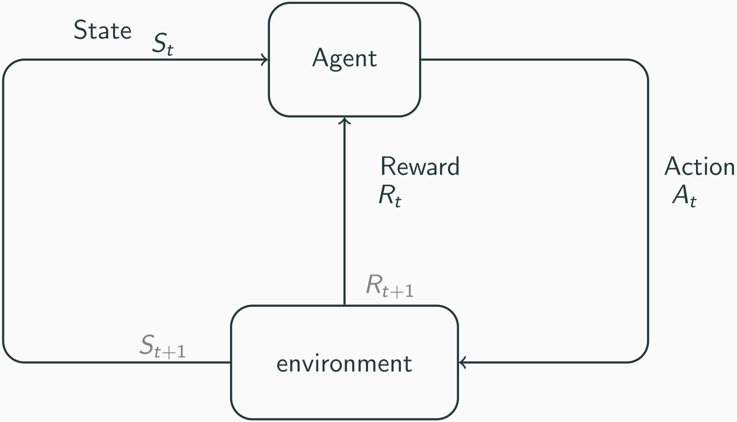
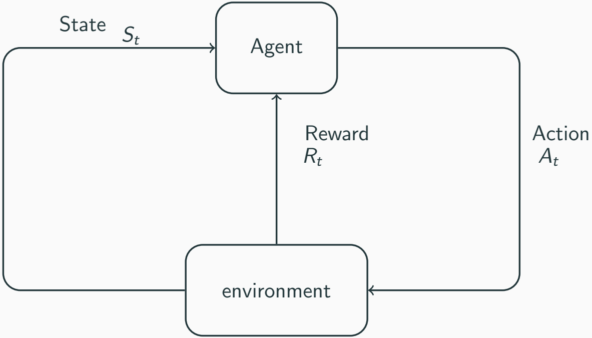


Figure adapted from Sutton & Barto (2018) Reinforcement Learning

Interaktion model

at time t:

- **Agent:**
  (1. obtains reward R_t)
  2. registers state S_t
  3. performs action A_t

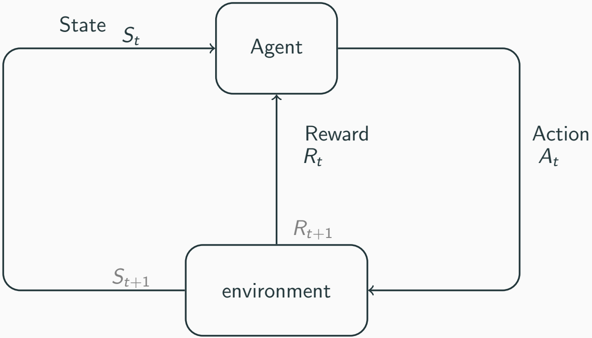


Interaktion model

at time t:

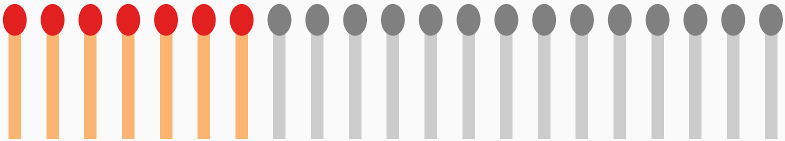
- **Agent:**
  (1. obtains reward R_t)
  2. registers state S_t
  3. performs action A_t

- **environment:**
  1. obtains aktion A_t
  2.a sends reward R_{t+1}
  2.b evolves to state S_{t+1}



Player 2: selected 2 matchsticks

$S_4 = 7$:

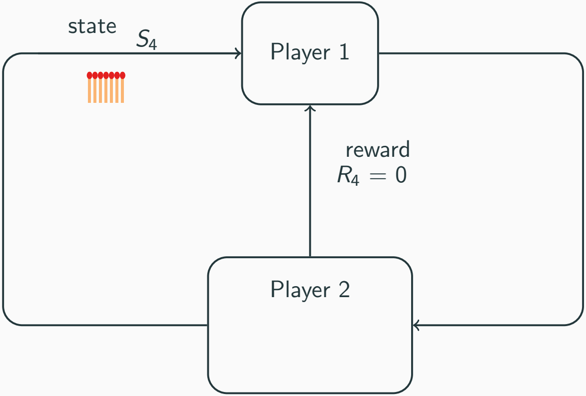


t = 4

Matchstick Game

at time t = 4:

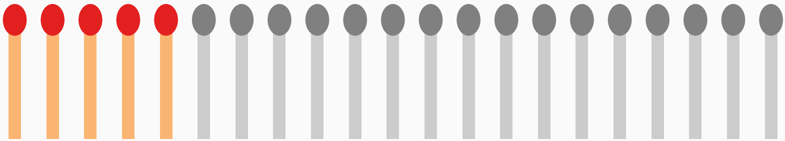
- **Agent (Player 1):**
  - registers state $S_4 = 7$: 7 matchsticks left
  - obtains reward $R_4 = 0$ (no winner or looser yet)



Matchstick Game

Player 1: take 2 matchsticks (A_4)

$S_{4+1/2} = 5$:

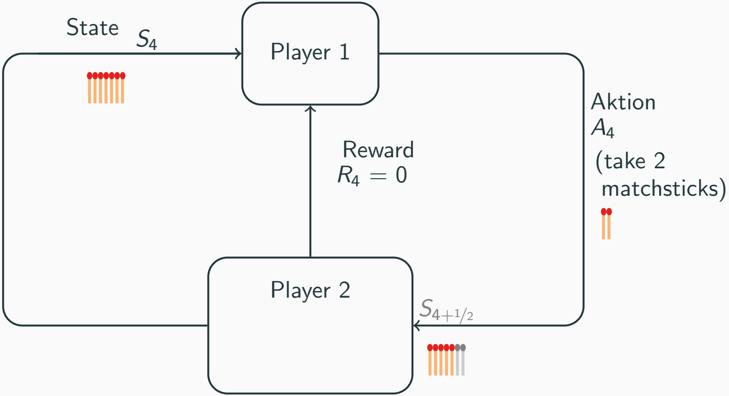


t $= 4$

Interaktionsmodell: Matchstick Game

at time t = 4:

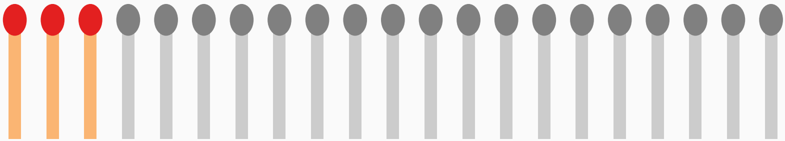
- **Agent (Player 1):**
  - registers $S_4 = 7$: 7 matchsticks left
  - ontains reward $R_4 = 0$ (no winner/looser yet)
  - performs action $A_4 = 2$: take 2 matchsticks



Interaktionsmodell: Matchstick Game

Player 2: takes 2 matchsticks

$S_5 = 3$:

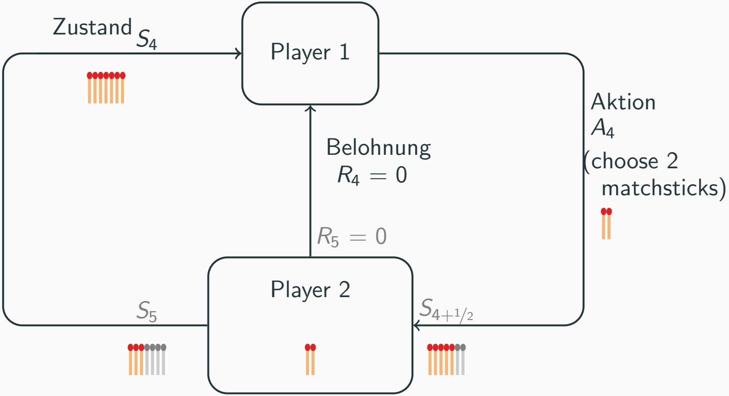


t = 4

Interaktionsmodell: Matchstick Game

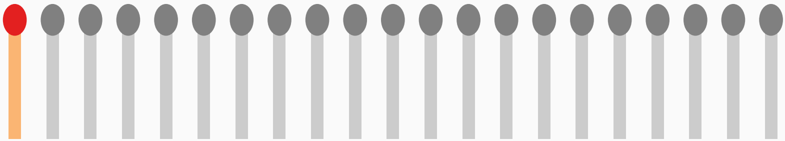
Zum Zeitpunkt t = 4:

- **environment (Player 2):**
  - obtains the action $A_4 = 2$: 5 matchsticks remaining ($S_{4+1/2}$)
  - evolves state to $S_5 = 3$: takes 2 matchsticks
  - sends reward $R_5 = 0$ (no winner/looser yet)



Player 1: takes 2 matchsticks (A_5)

$S_{5+1/2} = 1$:

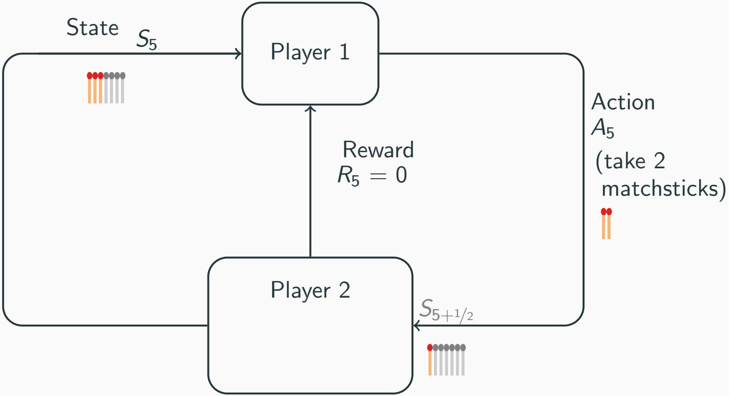


t = 5

Interaktionsmodell: Matchstick Game

Zum Zeitpunkt t = 5:

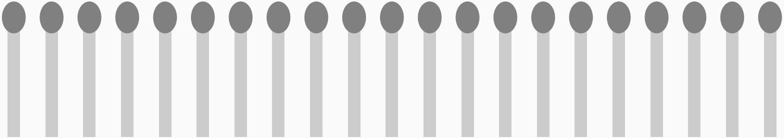
- **Agent (Player 1):**
  - registers state $S_5 = 3$: 3 matchsticks left
  - obtains reward $R_5 = 0$ (no winner yet)
  - performs actions $A_5 = 2$: takes 2 matchsticks



Matchstick Game

Player 2: takes 1 matchstick

$S_6 = 0$:

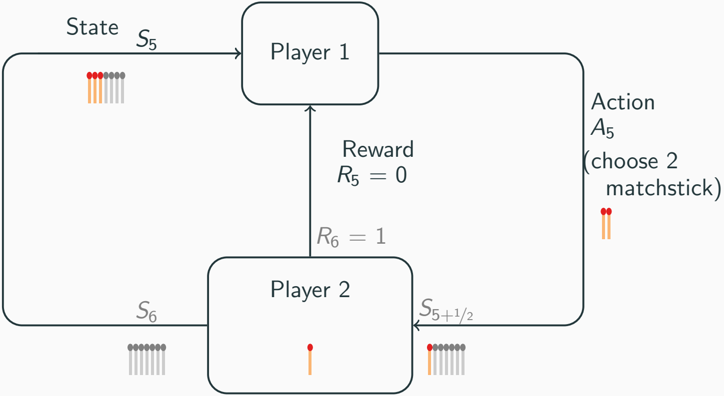


$t = 5$

Matchstick Game

at time t = 5:

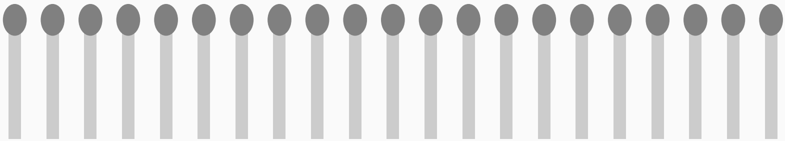
- **environment (Player 2):**
  - obtains action $A_5 = 2$: 1 matchstick left ($S_{5+1/2}$)
  - evolves the state to $S_6 = 0$: takes 1 matchstick
  - sends reward $R_6 = 1$ (Player 1 wins)



Player 1: wins

Player 1 wins because the Player 2 took the last matchstick.

$S_6 = 0$:

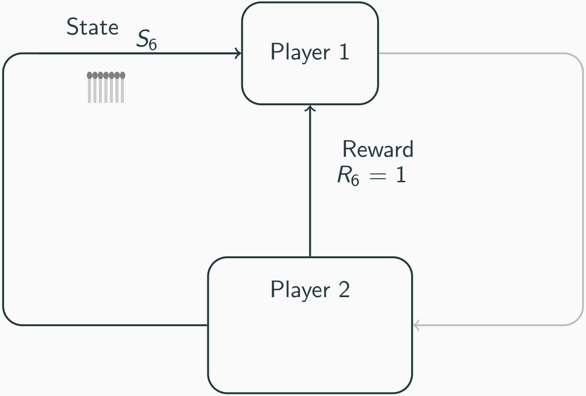


t = 6

Matchstick Game

Zum Zeitpunkt t = 6:

- **Agent (Player 1):**
  - registers state $S_6 = 0$: no matchsticks left
  - obtains reward $R_6 = 1$ (won)



Markov Decision Process

Markov Process

Defintion 1.0

Let the state space X be a bounded compact subset of the Euclidean space, the discrete-time dynamic system $(x_t)_{t\in\mathbb{N}} \in X$ is a Markov chain if it satisfies the Markov property

$$\mathbb{P}(x_{t+1} = x | x_t, x_{t-1}, \ldots, x_0) = \mathbb{P}(x_{t+1} = x | x_t)$$

Given an initial state $x_0 \in X$, a Markov chain is defined by the transition probability p

$$p(y|x) = \mathbb{P}(x_{t+1} = y | x_t = x) \tag{1}$$

Markov Decision Process

Defintion 1.0

A Markov Decision Process (MDP) $\mathcal{M}$ is defined to be a tuple $\langle \mathcal{S}, \mathcal{A}, P, P_0, q \rangle$ where

- $\mathcal{S}$ is the set of states,
- $\mathcal{A}$ is the set of actions,
- $P(\cdot|s, a) \in \mathcal{P}(\mathcal{S})$ is the probability distribution over next states, conditioned on action a being take in state s
- $P_0 \in \mathcal{P}(\mathcal{S})$ is the probability distribution according to which the initial state is selected
- $R(s, a) \sim q(\cdot|s, a) \in \mathcal{P}(\mathbb{R})$ is a random variable representing the reward obtained when action a is taken in state s

MPD assumptions

Assumptions:

- P_0, P and q are assumed to be stationary
- Assume that rewards are bounded by R_{max} and that the expected reward

$$\bar{r}(s,a) = \int rq(r|s,a)dr \leq \bar{R} \leq R_{max} \tag{2}$$

In the beginning: reward is assumed to be deterministic

Matchstick Game

- States: number of matchsticks left
  $\mathcal{S} = \{0, 1, \ldots, 21\}$
- Actions: choose 1, 2 oder 3 matchsticks
  $\mathcal{A} = \{1, 2, 3\}$
- Belohnungen:

$$\mathcal{R}(s, a) = \begin{cases} -1 & \text{, if agent takes last matchstick} \\ 0 & \text{, if agent does not take last matchstick} \\ +1 & \text{, game over and agent did not take last matchstick} \end{cases}$$

Policy

Defintion 1.0

A Markov policy is a mapping from the set of States $\mathcal{S}$ to the set of actions $\mathcal{A}$. A policy that does not change over time is called stationary. One can consider two variations of stationary Markov policies:

- a stochastic policy $\mu(\cdot|s)$ which is a probability distribution over the set of actions given a state $s \in \mathcal{S}$
- a deterministic policy which is given by map: $\mu : \mathcal{S} \to \mathcal{A}$

MDP controlled by a policy μ

Defintion 1.0

A MDP controlled by a policy μ induces a Markov chain $\mathcal{M}^\mu$ with

- reward distribution $q^\mu(\cdot|s) = q(\cdot|s, \mu(s))$ such that $R^\mu(s) = R(s, \mu(s)) \in q^\mu(\cdot|s)$
- transition kernel $P^\mu(\cdot|s) = P(\cdot|s, \mu(s))$

Path

Defintion 1.0

In a Markov chain $\mathcal{M}^{\mu}$, for action pairs $z = (s, a) \in \mathcal{Z} = \mathcal{S} \times \mathcal{A}$, we define the transition density and the initial (state-action) density as

$$P^{\mu}(z'|z) = P(s'|s, a)\mu(a'|s') \tag{3}$$

and

$$P_0^{\mu}(z_0) = P_0\mu(a_0|s_0) \tag{4}$$

respectively. Further let $\xi = \{z_0, z_1, \ldots, z_T\} \in \Xi$ with $T \in \{0, 1, \ldots, \infty\}$ denote a path (or trajectory generated by this Markov chain.)

Note: The probability density of such a path is given by

$$Pr(\xi|\mu) = P_0^{\mu}(z_0) \prod_{t=1}^{T} P^{\mu}(z_t|z_{t-1}) \tag{5}$$

Return

Defintion 1.0

The so called discounted return is a random variable $\rho : \Xi \to \mathbb{R}$

$$\rho(\xi) = \sum_{t=0}^{T} \gamma^t R(z_t) \tag{6}$$

where R is assumed to be deterministic with $\gamma \in [0, 1]$.

Expected return

Defintion 1.0

The expected return of a policy μ is defined by

$$\eta(\mu) = \mathbb{E}[\rho(\xi)] = \int_{\Xi} \rho(\xi) Pr(\xi|\mu) d\xi. \tag{7}$$

The expectations is over all possible trajectories generated by policy μ and all possible rewards collected in them.

expected return of a state s

Defintion 1.0

Analogously for a given policy μ the return of a state s is defined by

$$D^{\mu}(s) = \sum_{t=0}^{\infty} \gamma^t R(Z_t) | Z_0 = (s, \mu(\cdot|s)) \text{ with } S_{t+1} \sim P^{\mu}(\cdot|S_t) \quad (8)$$

The expected value of D^{μ} is called the value function of policy μ

$$V^{\mu}(s) = \mathbb{E}[D^{\mu}(s)] = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R(Z_t) | Z_0 = (s, \mu(\cdot|s))\Big] \quad (9)$$

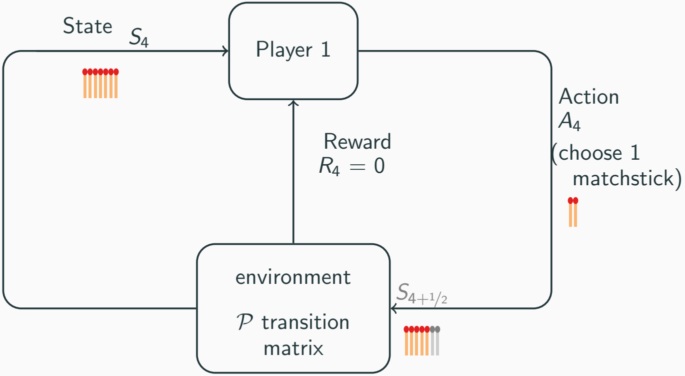
Bellman equation

The Bellmann equation for the value function allows to write the value of a state s under a policy μ in terms of its immediate reward and values of its successor states under μ

$$V^{\mu}(s) = R^{\mu}(s) + \gamma \int_{\mathcal{S}} P^{\mu}(s'|s) V^{\mu}(s') ds' \tag{10}$$

$\mathcal{P}(s'|s,a)$ for Matchstick Game

- **Agent (Player 1):** plays according to policy $\mu(s) = 1 \; \forall \; s \in \mathcal{S}$
  - registers state S_4: only 7 matchsticks left
    $\rightarrow s = (0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^T$
  - performs action: takes 1 matchstick



Action-value function

Defintion 1.0

The action-value function of a policy is the total expected (discounted reward) when it starts in state s, takes action a and then executes policy μ

$$Q^{\mu}(z) = \mathbb{E}[D^{\mu}(z)] = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R(Z_t) | Z_0 = z)\Big] \tag{11}$$

Bellman equation

Theorem 1.1

Let T be the known time horizon. The value functions of a deterministic policy μ satisfies the following equations

$$V_t^\mu(s) = R^\mu(s) + \gamma \sum_{s' \in \mathcal{S}} P^\mu(s'|s) V_{t+1}^\mu(s') \tag{12}$$

for all $t \in \{1, \ldots, T\}$ with the convention that $V_{T+1}^\mu(s) = 0$ for all $s \in \mathcal{S}$.

Complexity: for a finite state space $\mathcal{S}$ such that $|\mathcal{S}| = S$

- $V_1^\mu(s)$ can be determined using backwards induction
- $S \times T$ memory

Grid World

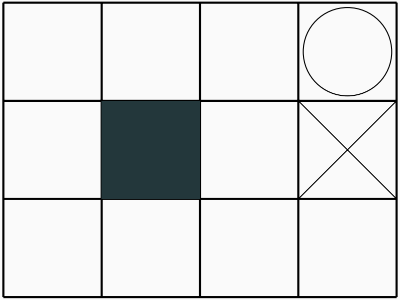


Figure adapted from Youtube: https://www.youtube.com/watch?v=bHeeaXgqVig

- $\mathcal{S} = \{(1,1),\ldots,(4,3)\} \setminus \{(2,2)\}$

Grid World

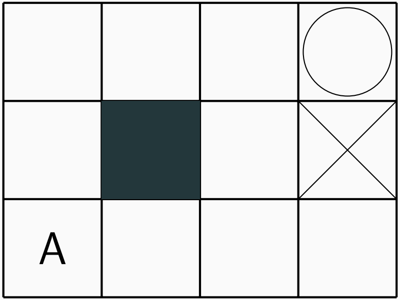


Figure adapted from Youtube: https://www.youtube.com/watch?v=bHeeaXgqVig

- $\mathcal{S} = \{(1,1),\ldots,(4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow,\downarrow,\leftarrow,\rightarrow\} = \{(0,1),(0,-1),(-1,0),(1,0)\}$

Grid World

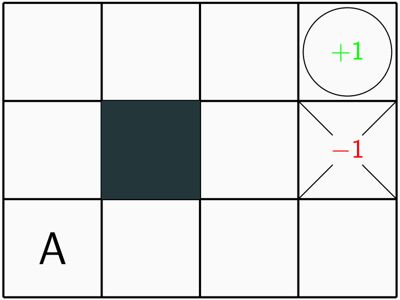
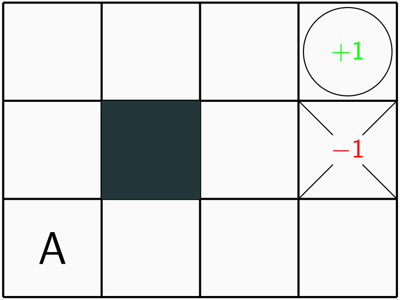


Figure adapted from Youtube: https://www.youtube.com/watch?v=bHeeaXgqVig

- $\mathcal{S} = \{(1,1),\ldots,(4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow,\downarrow,\leftarrow,\rightarrow\} = \{(0,1),(0,-1),(-1,0),(1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{touching the trap (X)} \\ +1 & \text{reaching goal (O)} \end{cases}$

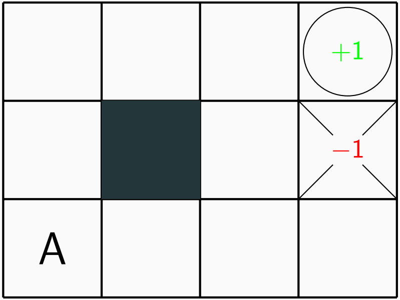
Grid World



- $\mathcal{S} = \{(1,1), \dots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$

- $\mathcal{P}^a_{ss'} = \mathbb{P}\left[s' \mid s, a\right] = \begin{cases} 0,8 & \text{target direction (a)} \\ 0,1 & \text{on the right of a} \\ 0,1 & \text{on the left of a} \end{cases}$

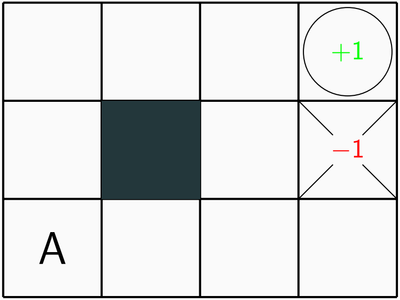
Grid World



- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$

- $\mathcal{P}^a_{ss'} = \mathbb{P}\left[s' \mid s, a\right] = \begin{cases} 0,8 & \text{target direction (a)} : s' = s + a \\ 0,1 & \text{on the right of a} : s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & \text{on the left of a} : s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$

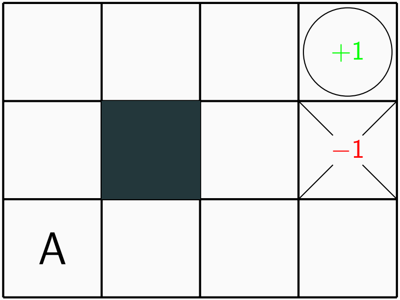
Grid World



- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$

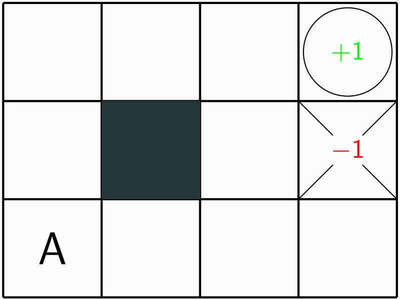
- leaving the grid: $A : \mathcal{S} \times \mathcal{A}(s) \rightarrow \mathcal{S}, (s,a) \mapsto A(s,a) = \begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Grid World



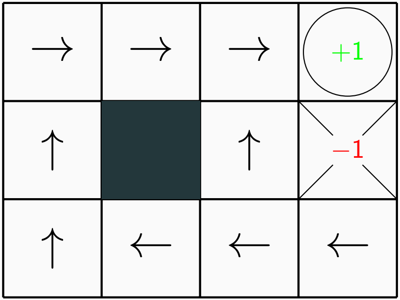
- $\mathcal{S} = \{(1,1), \dots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- A(a,s)=$\begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Grid World



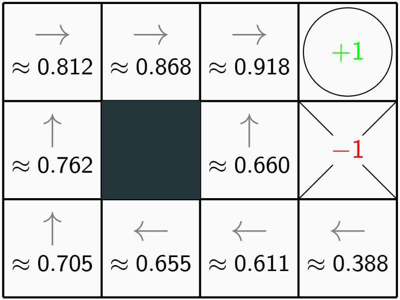
- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- A(a,s)=$\begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Grid World



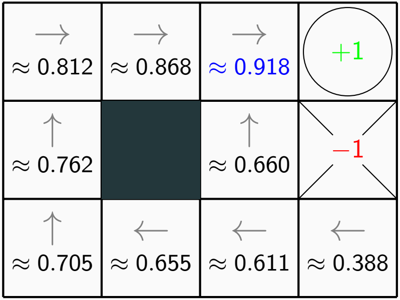
- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- A(a,s)=$\begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Grid World



- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- A(a,s)= $\begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Grid World

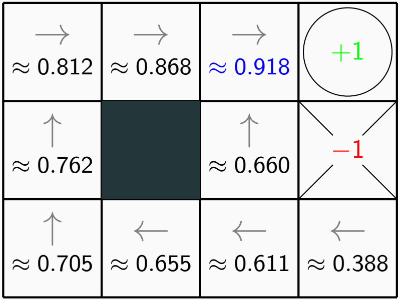


- $\mathcal{S} = \{(1,1), \dots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- A(a,s)=$\begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Example for s=(3,3), $\mu(s) = (1,0)$:

$$V^{\mu}((3,3)) = -0.04 + 0.8 \cdot V^{\mu}(s+a) + 0.1 \cdot V^{\mu}(s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}) + 0.1 \cdot V^{\mu}(s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix})$$

Grid World



- $\mathcal{S} = \{(1,1), \ldots, (4,3)\} \setminus \{(2,2)\}$
- $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\} = \{(0,1), (1,0), (0,-1), (-1,0)\}$
- $\mathcal{R}(s,a) = \begin{cases} -0.04 & \text{each step} \\ -1 & \text{on X} \\ +1 & \text{on O} \end{cases}$
- $\mathcal{P}^a_{ss'} = \begin{cases} 0,8 & s' = s + a \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ 0,1 & s' = s + a \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{cases}$
- $\text{A(a,s)} = \begin{cases} s' & s' \in \mathcal{S} \\ s & s' \notin \mathcal{S} \end{cases}$

Example for s=(3,3), $\mu(s) = (1, 0)$:

$$\begin{aligned} V^{\mu}((3,3)) &= -0.04 + 0.8 \cdot V^{\mu}((4,3)) + 0.1 \cdot V^{\mu}((3,2)) + 0.1 \cdot V^{\mu}((3,3)) \\ &= 0.8 \cdot (+1) + 0.1 \cdot 0.660 + 0.1 \cdot 0.918 - 0.04 \\ &\approx 0.918 \end{aligned}$$

Solving the Bellman equations

For a fixed stationary, deterministic policy μ we know that following equations are satisfied

$$V^{\mu}(s) = R^{\mu}(s) + \gamma \sum_{s' \in \mathcal{S}} P^{\mu}(s'|s) V^{\mu}(s') \tag{13}$$

Introducing the vectors

$$V^{\mu} = V^{\mu}(s))_{s=1}^{S} \in \mathbb{R}^{S} \tag{14}$$

$$R^{\mu} = R^{\mu}(s))_{s=1}^{S} \in \mathbb{R}^{S} \tag{15}$$

and the matrix

$$P^{\mu} = (P(s'|s, \mu(s)))_{1 \leq s \leq S \text{ and } 1 \leq s' \leq S9} \in \mathbb{R}^{S \times S} \tag{16}$$

the Bellman equation yields:

$$V^{\mu} = R^{\mu} + \gamma P^{\mu} V^{\mu} \tag{17}$$

Solving the Bellman equations

Due to

$$V^\mu = R^\mu + \gamma P^\mu V^\mu \tag{18}$$

the vector $V^\mu \in \mathbb{R}^S$ satisfies

$$(I - \gamma P^\mu) V^\mu = R^\mu \tag{19}$$

$$V^\mu = (I - \gamma P^\mu)^{-1} R^\mu \tag{20}$$

provided that the matrix $I - \gamma P^\mu$ is invertible.

Note: complexity $\mathcal{O}(S^3)$

Alternative approach to solve Bellman equations

Defintion 1.1

The Bellman operator associated to a policy μ is defined by

$$T^\mu : \mathcal{R}^S \to \mathcal{R}^S \tag{21}$$

$$V \mapsto T^\mu(V) \tag{22}$$

where

$$T^\mu(V)(s) = R^\mu(s, \mu(s)) + \gamma \sum_{s' \in \mathcal{S}} P^\mu(s'|s, \mu(s))V(s') \tag{23}$$

Idea: Solve Bellman equation via Fix Point iteration

Banach Fix Point Theorem

Theorem 1.2

Let be (X, d) a complete metric space and let $T : X \to X$ be contraction mapping on X, i.e., there exists $\gamma \in [0, 1)$ such that

$$d(T(x), T(y)) \leq \gamma d(x, y) \tag{24}$$

for all x, y in X. Then

- T admits a unique fixed-point $x*$ in X, i.e., $T(x^*) = x^*$
- for any $x_0 \in X$ the fix point iteration $x_n = T(x_{n-1})$ converges to x^* (linear convergence dependent on γ)

Note : T^μ has a unique fixed-point V_μ

Optimal value function

Goal: for a MDP find a policy μ that maximizes the value function:

Defintion 1.2

The optimal

$$V^*(s) = \sup_{\mu} V^{\mu}(s) \tag{25}$$

for all states $s \in \mathcal{S}$.

Theorem 1.3

There exists an optimal policy μ^* which satisfies

$$\mu^* \in \text{argmax}_{\mu} V^{\mu}(s) \quad \forall s \in \mathcal{S}. \tag{26}$$

Therefore we can write $V^* = V^{\mu^*}$.

Optimal policy

Defintion 1.3

A policy μ^* is referred to as optimal if it attain the optimal values at all states

$$V^{\mu^*}(s) = V^*(s) \tag{27}$$

for all $s \in \mathcal{S}$.

Theorem 1.4

$V^*(s) = \sup_\mu V^\mu(s)$ satisfy the Bellman equations :

$$V^*(s) = \max_{a \in \mathcal{A}} \left[R(s,a) + \gamma \int_{\mathcal{S}} P^\mu(s'|s,a) V^*(s') ds' \right] \tag{28}$$

Moreover, an optimal policy is given by

$$\mu^*(s) \in \mathrm{argmax}_{a \in \mathcal{A}} \left[R(s,a) + \gamma \int_{\mathcal{S}} P^\mu(s'|s,a) V^*(s') ds' \right] \quad \forall s \in \mathcal{S}. \tag{29}$$

Greedy policy

Defintion 1.4

A deterministic policy is referred to as a greedy policy if

$$\mu(s) = \text{argmax}_{a \in \mathcal{A}} \Big[R(s, a) + \gamma \int_{\mathcal{S}} P(s'|s, a) V^*(s') ds' \Big] \quad \forall s \in \mathcal{S} \tag{30}$$

Bellman equations for the optimal values

Defintion 1.4

The optimal Bellman operator is defined by

$$T^* : \mathcal{R}^S \to \mathcal{R}^S \tag{31}$$

$$V \mapsto T^*(V) \tag{32}$$

where

$$T^*(V)(s) = \max_{a \in \mathcal{A}} \left[R^\mu(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s') \right] \tag{33}$$

Optimal action-value function

Goal: for a MDP find a policy μ that maximizes the value function:

$$Q^*(z) = \sup_{\mu} Q^{\mu}(z) \tag{34}$$

for all states $s \in \mathcal{S}$.

Bellman equation for action-value function

The Bellmann equation for the action-value function under a policy μ in terms of its immediate reward and values of its successor states and actions under μ

$$Q^{\mu}(s,a) = R(s,a) + \gamma \int_{\mathcal{S}} P^{\mu}(s'|s,a)\Big(\int_{a'\in\mathcal{A}} \mu(a'|s')Q^{\mu}(s',a')da'\Big)ds' \tag{35}$$

Link between action-value and value function

$$V^{\mu}(s) = \int_{a \in \mathcal{A}} Q^{\mu}(s, a) \tag{36}$$

$$Q^{*}(s, a) = R(s, a) + \gamma \int_{\mathcal{S}} P^{\mu}(s'|s, a)\Big(\max_{a \in \mathcal{A}} Q^{*}(s', a')\Big) ds' \tag{37}$$