

# Value and Policy Iteration

DOROZHKO Anton

Novosibirsk State University

May 16, 2020

# Outline

- 1 Markov decision processes
- 2 Policy Iteration
- 3 Value Iteration
- 4 Other

# Introduction to MDPs

- MDP describes environment
- Fully observable - state completely characterises the process
- Almost all RL problems can be formalised as MDPs

## Definition

A state  $S_t$  is Markov iff

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \dots, S_t]$$

- captures all relevant information
- can throw away the history if we know state

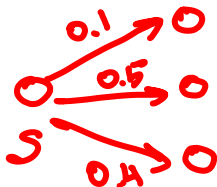
MDP:  $S \checkmark$   
 $A \times$   
 $P \checkmark$   
 $R \times$

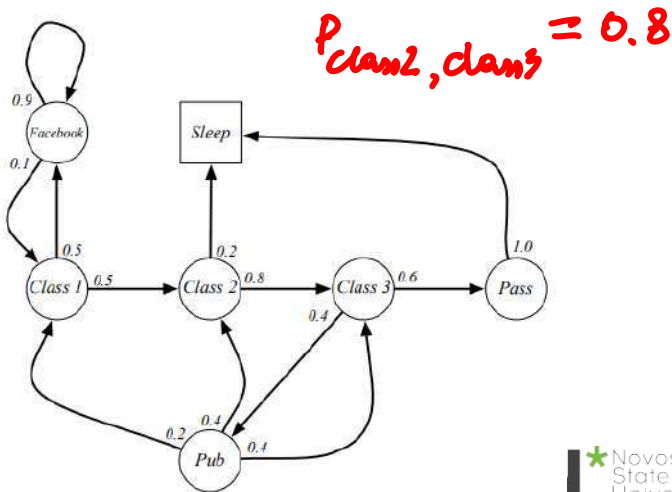
## Definition

A Markov Process (or Markov Chain) is a tuple  $(\mathcal{S}, \mathcal{P})$

- $\mathcal{S}$  is a (finite) set of states
- $\mathcal{P}$  is a transition probability matrix

$$P_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$$



Student's MDP<sup>1</sup><sup>1</sup>from David Silver course

# Markov Reward Process

MDP S ✓  
 P ✓  
 A X  
 B ~  
 γ ✓

## Definition

A Markov Reward Process is a tuple  $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$

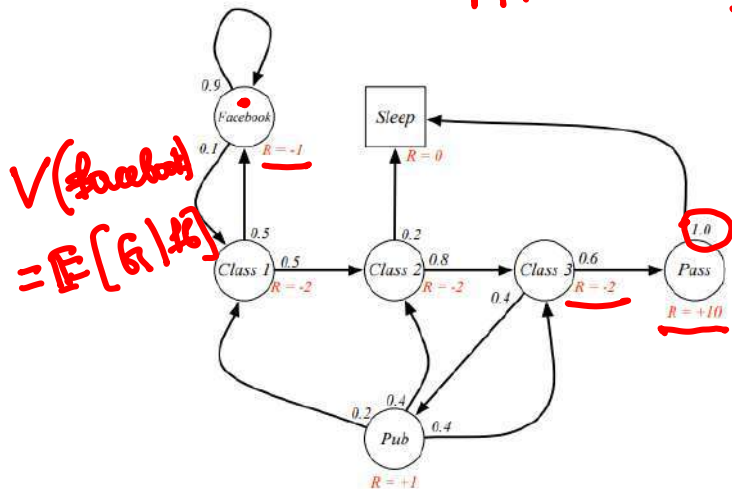
- $\mathcal{S}$  is a (finite) set of states
- $\mathcal{P}$  is a transition probability matrix

$$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$$

- $\mathcal{R}$  is a reward function :  $\mathcal{R}_s = \mathbb{E}[R_{t+1} | \underline{S_t = s}]$
- $\gamma$  is a discount factor,  $\gamma \in [0, 1]$

Student's MRP <sup>2</sup>

MP MRP



<sup>2</sup>from David Silver course

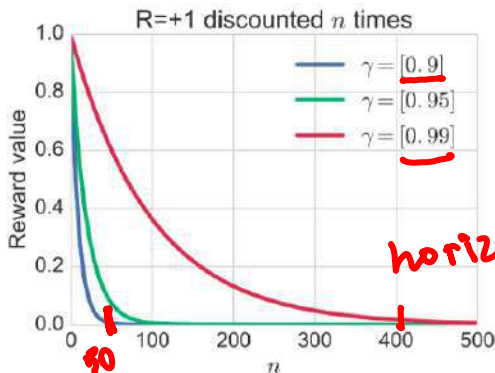
# Discounted Reward

*cumulative reward*

$\gamma$

$$\underline{G_t} = R_t + \gamma R_{t+1} \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

If  $\max R = 1$  then  $G_0 = \sum \gamma^k = \frac{1}{1-\gamma}$





# Discounted Reward

$$\underline{G_t} = R_t + \underbrace{\gamma R_{t+1} \dots}_{\text{discounted future rewards}} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$\underline{G_t} = R_t + \gamma \underline{G_{t+1}}$$

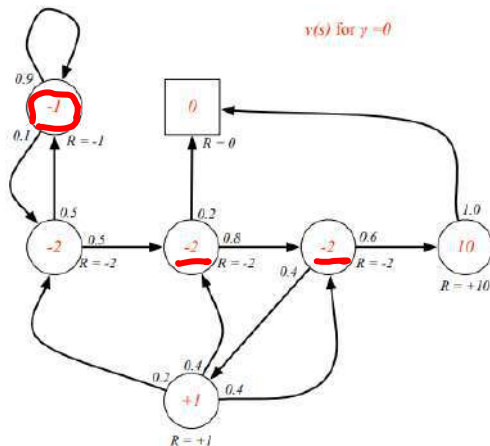
# Value Function

## Definition

The state value function  $V(s)$  of an MRP is the expected return starting from state  $s$

$$V(s) = \mathbb{E}[G_t | S_t = s]$$

$V(s)$  gives the **long-term value of state  $s$**

State value function's for Student MRP <sup>3</sup>

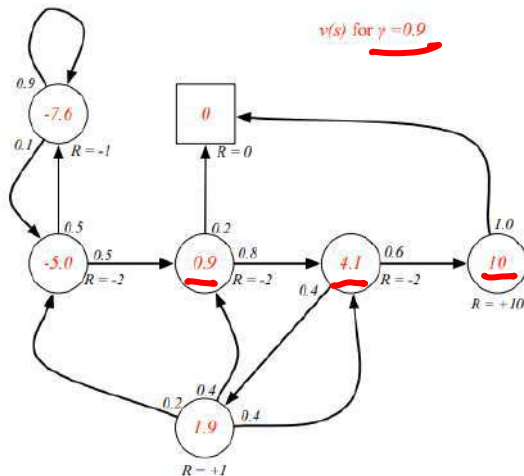
$$\underline{\gamma = 0}$$

$$V(s) =$$

$$E[G | s]$$

$$\underline{G = R_{\frac{1}{2}} + \gamma V_{\frac{1}{2}}}$$

<sup>3</sup>from David Silver course


State value function's for Student MRP <sup>4</sup>

<sup>4</sup>from David Silver course

# Bellman Equation for Value Function

Decomposition:

- immediate reward  $R_{t+1}$
- discounted value of the next state  $\gamma V(S_{t+1})$

$$\begin{aligned}
 \underline{V(s)} &= \mathbb{E}[\underline{G_t} | S_t = s] \\
 &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s] \\
 &= \mathbb{E}[R_{t+1} + \underline{\gamma}(R_{t+2} + \gamma R_{t+3} + \dots) | S_t = s] \\
 &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\
 &= \mathbb{E}[\underline{R_{t+1}} + \gamma \underline{V_{t+1}} | S_t = s]
 \end{aligned}$$


## Bellman equation MRP

current  $r$  / value at next state

$$\underline{V}(s) = \underline{\mathbb{E}}[R_{t+1} + \gamma V(S_{t+1} | S_t = s)]$$

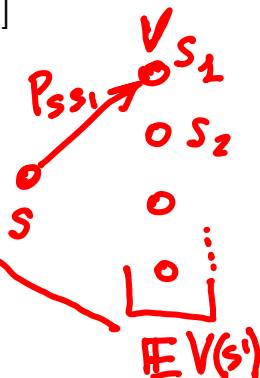
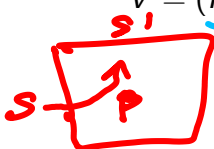
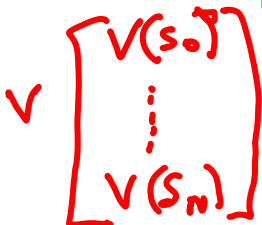
$$V(s) = r + \gamma \sum_{s' \in \mathcal{S}} P_{ss'} V(s')$$

↓ matrices

$$\underline{V} = \underline{R} + \gamma \underline{P} \underline{V}$$

$$(I - \gamma \underline{P}) \underline{V} = \underline{R}$$

$$\underline{V} = (I - \gamma \underline{P})^{-1} \underline{R}$$



$$V = \mathcal{R} + \gamma \mathcal{P}V$$

$$(I - \gamma \mathcal{P})V = \mathcal{R}$$

$$V = (I - \gamma \mathcal{P})^{-1} \mathcal{R}$$

- $\mathcal{O}(n^3)$  for  $n$  states
- small MDPs
- Other options:
  - Dynamic programming (DP)
  - Monte-Carlo evaluation (MC) ←
  - Temporal-Difference learning (TD) ←

# Markov Decision Process



## Definition

A Markov **Decision Process** is a tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$

- $\mathcal{S}$  is a (finite) set of states
- $\mathcal{A}$  is a finite set of actions
- $\mathcal{P}$  is a transition probability matrix

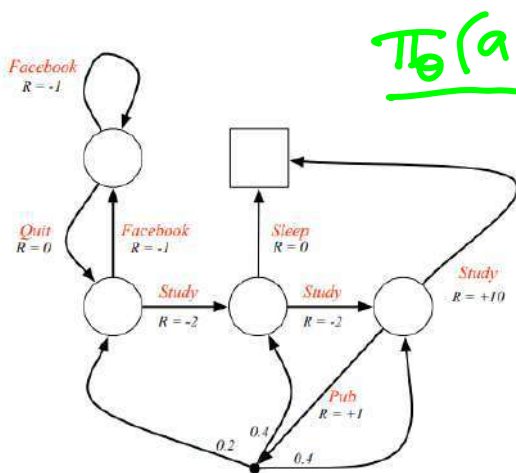
$$\mathcal{P}_{ss'}^a = \mathbb{P}[\mathcal{S}_{t+1} = s' | \mathcal{S}_t = s, \mathcal{A}_t = a]$$

- $\mathcal{R}$  is a reward function :

$$\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | \mathcal{S}_t = s, \mathcal{A}_t = a]$$

- $\gamma$  is a discount factor,  $\gamma \in [0, 1]$



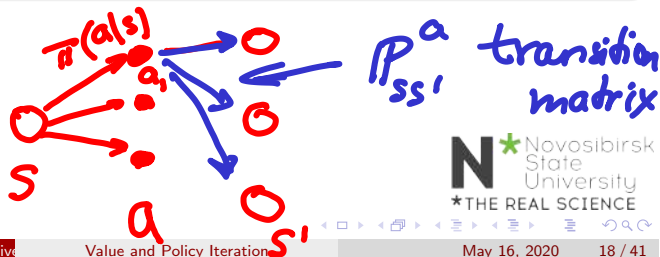
Student's MDP <sup>5</sup><sup>5</sup>from David Silver course

# Policy

## Definition

A policy  $\pi$  is a distribution over actions given states

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$$



# Value Function

## State-value function

The state value function  $V_\pi(s)$  of an MDP is the expected return starting from state  $s$ , and then following policy  $\pi$

$$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$



## Action-value function

The action-value function  $Q_\pi(s, a)$  is expected return starting from state  $s$ , taking action  $a$ , and then following policy  $\pi$

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$



$$\pi \doteq \operatorname{argmax}_a Q(s, a)$$



# Notation variants

$$\begin{aligned}
 \underline{\mathbb{E}[G_0]} &= \mathbb{E}[R_0 + \gamma R_1 + \dots + \gamma^T R_T] \\
 &= \mathbb{E}[G_0 | \underline{\pi_\theta}] \\
 &= \underline{\mathbb{E}_{\pi_\theta}[G_0]} \\
 &= \sum_{t=0}^T \mathbb{E}_{(s_t, a_t) \sim p_\theta} [\gamma^t R_t] \\
 &= \mathbb{E}_{\tau \sim p_\theta(\tau)} [G(\tau)]
 \end{aligned}$$

rollout / trajectory

- $\tau = (s_0, a_0, s_1, a_1, \dots, a_{T-1}, s_T)$
- $\underline{p_\theta(\tau)} = p(s_0) \prod_{t=0}^{T-1} \underline{\pi_\theta(a_t | s_t)} p(s_{t+1} | s_t, a_t)$

choice of transition action

# Bellman Expectation Equation

Decomposition into immediate reward plus discounted value in next state

$$V_{\pi}(s) = \mathbb{E}_{\pi}[\underline{R_{t+1}} + \gamma V_{\pi}(S_{t+1}) | S_t = s]$$

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s, \underline{A_t = a}]$$

# Optimal Value Functions

## Definition

The optimal state-value function  $V_*(s)$  is the maximum value function over all policies

$$V_*(s) = \max_{\pi} V_{\pi}(s)$$

The optimal action-value function  $Q_*(s, a)$  is the maximum action-value function over all policies

$$Q_*(s, a) = \max_{\pi} Q_{\pi}(s, a)$$

# Optimal Policy

Partial ordering over policies

$$\pi \geq \underline{\pi'} \quad \text{if} \quad V_\pi(s) \geq V_{\pi'}(s) \quad (\forall s)$$

Theorem

For any Markov Decision Process (MDP)

- There exists an optimal policy  $\pi_*$  that is better than or equal to all other policies  $\pi_* \geq \pi, \forall \pi$
- All optimal policies achieve the optimal values function  $V_{\pi_*}(s) = V_*(s)$
- All optimal policies achieve the optimal action-value function  $Q_{\pi_*}(s, a) = Q_*(s, a)$

# Iterative algorithm

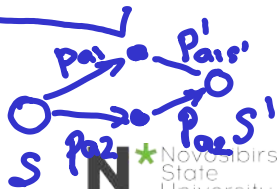
$$V(s)$$

- Initialize  $V_0(s) = 0$  for all  $s$
- for  $k = 1$  until convergence
  - for all  $s \in \mathcal{S}$

curr reward

$$V_k(s) = R(s) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s) V_{k-1}(s')$$

- $\mathcal{O}(|\mathcal{S}|^2)$  for each iteration

$$P(s'|s)$$




## MDP + Policy

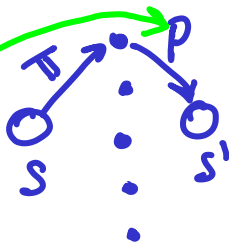
policy evaluation

fix

- MDP +  $\pi(a|s)$  = Markov Reward Process
- MRP(  $\mathcal{S}, \mathcal{R}^\pi, \mathcal{P}^\pi, \gamma$  ), where

$$R^\pi(s) = \sum_{a \in A} \pi(a|s) R(s, a)$$

$$P^\pi(s'|s) = \sum_{a \in A} \pi(a|s) P(s'|s, a)$$



- We can reuse iterative algorithm

$V(s)$

# Iterative algorithm

$$\pi: \mathcal{S} \rightarrow \mathcal{A}$$

MDP

$$r(s, a)$$

$$P(s' | s, a)$$

- Initialize  $V_0(s) = 0$  for all  $s$
- for  $k = 1$  until convergence
  - for all  $s \in \mathcal{S}$

$$\checkmark \quad V_k^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) \underline{V_{k-1}^\pi(s')}$$

- **Bellman backup** for particular policy

# MDP Control

- Compute optimal policy

$$\underline{\pi^*(s)} = \arg \max_{\pi} \underline{V^{\pi}(s)}$$

- There **exists a unique value function**

## Gridworld example

$$|A|=4$$



•	1	2	3
4	5	6	7
8	9	10	11
12	13	14	X

$r = -1$   
on all transitions

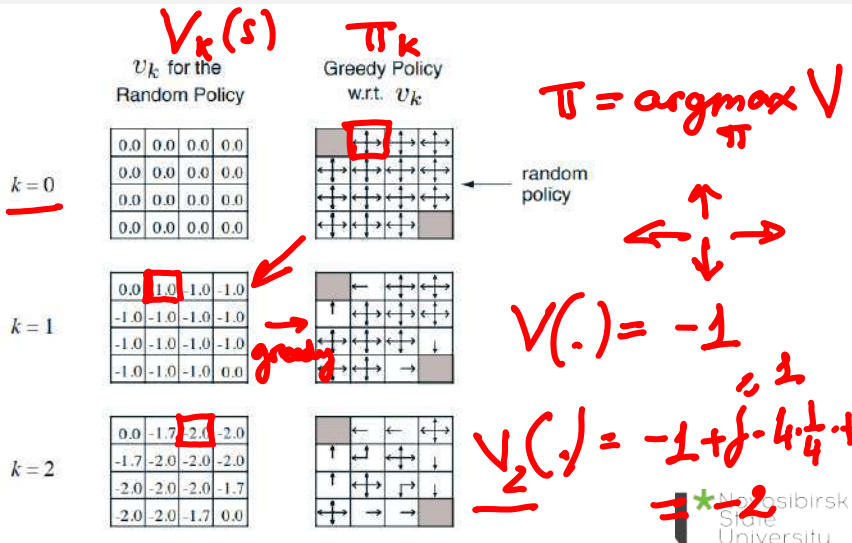
- Undiscounted episodic MDP ( $\gamma = 1$ )
- Nonterminal states 1, ..., 14
- One terminal state (shown twice as shaded squares)
- Actions leading out of the grid leave state unchanged
- Reward is  $-1$  until the terminal state is reached
- Agent follows uniform random policy

$$G = R_t + R_{t+1} + \dots$$

$$\pi(n|\cdot) = \pi(e|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = 0.25$$

evaluate  $\pi$ ?  $V(s)$

## Gridworld example



# Policy Iteration(PI)

- $i = 0$
- Initialize  $\pi_0(s)$  randomly for all  $s$
- While  $i == 0$  or  $\|\pi_i - \pi_{i-1}\| > \epsilon$ 
  - $V^\pi \leftarrow$  MDP policy evaluation of  $\pi_i$
  - $\pi_{i+1} \leftarrow$  Policy improvement
  - $i = i + 1$

greedy of  $V^\pi$

fix  $\pi \rightarrow V$

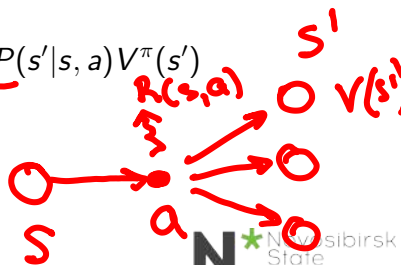
# Q function

Action value or State-Action value or Q-function

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

$$Q^{\pi}(s, a) = \underline{R(s, a)} + \gamma \sum_{s' \in \mathcal{S}} \underline{P(s'|s, a)} V^{\pi}(s')$$

Take action  $a$ , then follow policy  $\pi$



# Policy Improvement

- Compute Q function of  $\pi_i$ 
  - For  $s \in S$  and  $a \in A$  :

$$Q^{\pi_i}(s, a) = R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^{\pi_i}(s')$$

- Compute new policy  $\pi_{i+1}$

$$\pi_{i+1}(s) = \arg \max_a Q^{\pi_i}(s, a) \quad \forall s \in S$$



# Monotonic Improvement in Policy

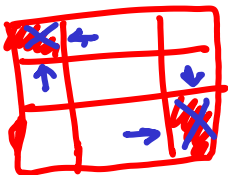
$$\pi_{i+1} = \operatorname{argmax}_a Q^{(\pi_i)}(s, a)$$

- 1  $V^{\pi_i}(s) \leq \max_a Q^{\pi_i}(s, a)$
- 2  $= \max_a (R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^{\pi_i}(s'))$
- 3  $= R(s, \pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{i+1}(s)) V^{\pi_i}(s')$
- 4  $\leq R(s, \pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{i+1}(s)) \max_{a'} Q^{\pi_i}(s', a')$
- 5 continue to expand  $a' = \pi_{i+1}(s')$

$$= V^{\pi_{i+1}}(s)$$

# Value Iteration

- Policy iteration : computes optimal value and policy
- Value Iteration :
  - Optimal value for state  $s$  if  $k$  episodes left
  - Iterate to consider longer episodes



# Gridworld example

## Example: Shortest Path

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

 $V_1$ 

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

 $V_2$ 

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

 $V_3$ 

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

 $V_4$ 

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

 $V_5$ 

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

 $V_6$ 

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

 $V_7$ 



# Contraction Operator

- Let  $O$  be an operator, and  $\|\cdot\|$  denote any norm of  $x$
- if  $\|OV - OV'\| \leq \|V - V'\|$  then  $O$  is a contraction operator
- $O$  has fixed point  $x$
- $Ox = x$

sketch  
idea

Proof: Bellman Backup is a Contraction on  $V$  for  $\gamma < 1$

$$V \begin{bmatrix} \cdot \end{bmatrix} \longleftrightarrow V' \begin{bmatrix} \cdot \end{bmatrix}$$

$$\|V - V'\| = \max_s \|V(s) - V'(s)\|$$

Exercise

$$V(s) = R(s, \pi(s)) + \gamma \sum P(s') V(s')$$

# POMDP

state



Figure: Doom Classic

A Partially Observable Markov Decision Process is an MDP with hidden states. It is HMM with actions.

# POMDP

## Definition

A POMDP is a tuple  $(S, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \gamma)$

- $S$  is a (finite) set of states ← hidden
- $\mathcal{A}$  is a finite set of actions
- $\mathcal{O}$  is a finite set of observations
- $\mathcal{P}$  is a transition probability matrix  

$$\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$
- $\mathcal{R}$  is a reward function :  $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- $\mathcal{Z}$  is an observation function  $\mathcal{Z}_{s'o}^a = \mathbb{P}[O_{t+1} = o | S_t = s, A_t = a]$
- $\gamma$  is a discount factor,  $\gamma \in [0, 1]$



# Questions ?

*The only stupid question is the one you were afraid to ask but never did*  
- Rich Sutton