AdTrans project, 2023-1-PL01-KA220-HED-000158917

ADTRANS
ADVANCED DATA ANALYSIS FOR TRANSPORTATION AND MOBILITY ACADEMIC TEACHERS

Session 4

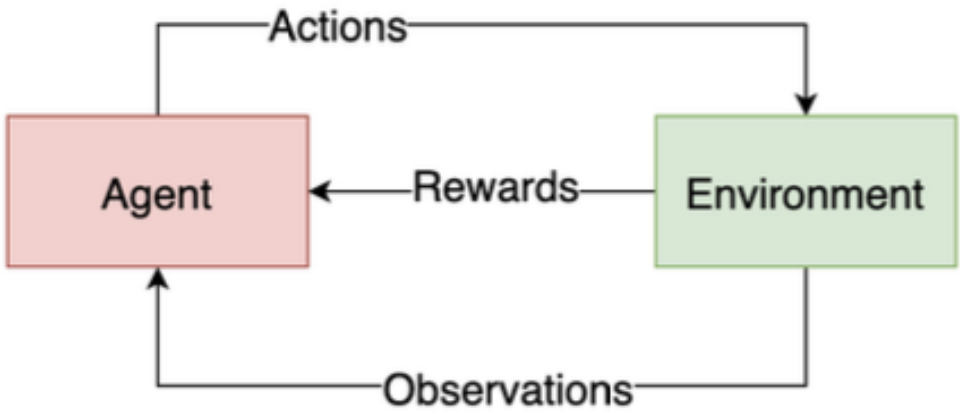
Reinforcement Learning Basics

Paraskevi Fasouli

Co-funded by the European Union

RL Introduction

- Limitations of supervised learning: Required labelled data
- Reinforcement learning addresses the cases where data is not available

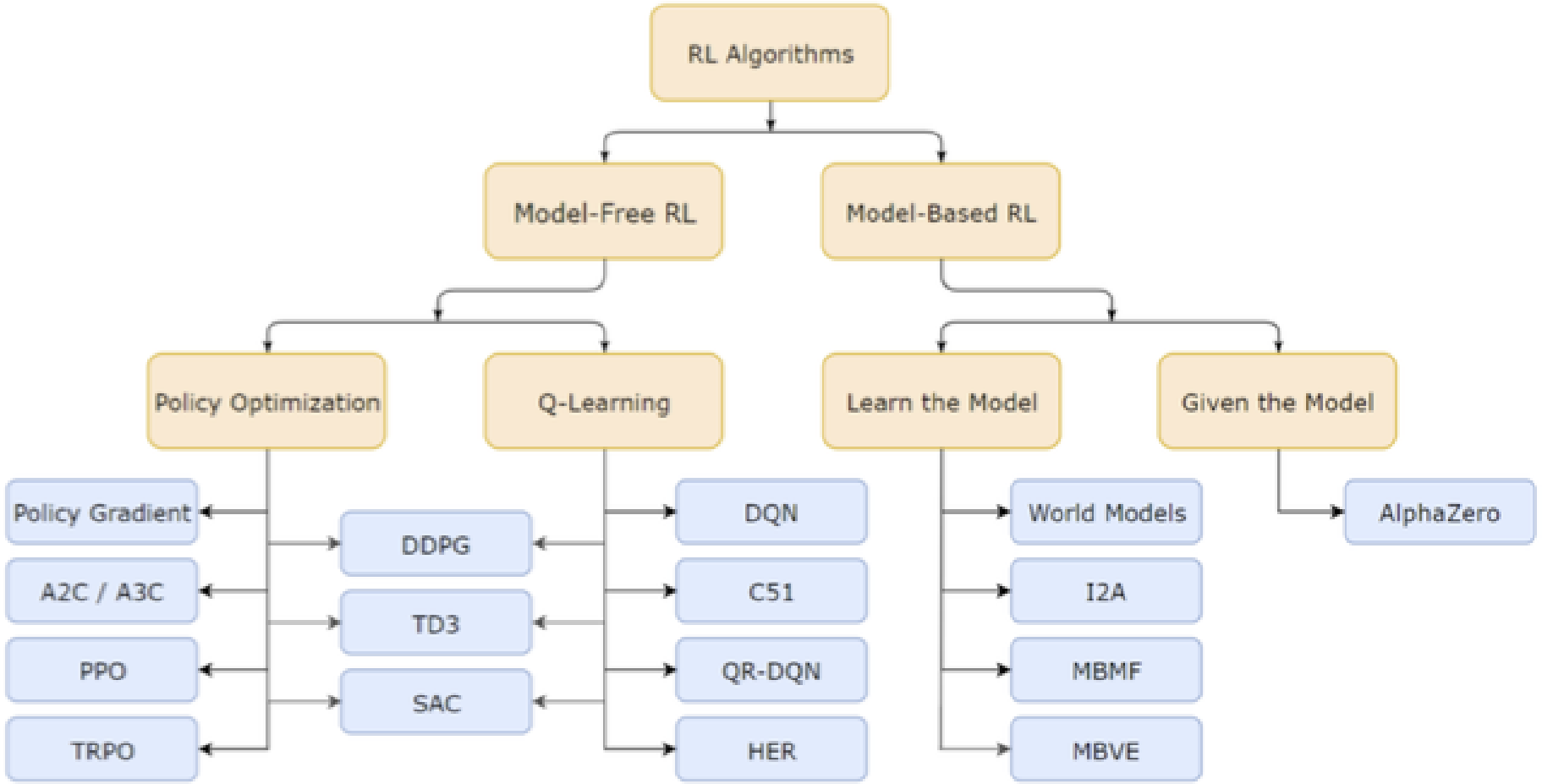


In RL: An agent performs an action on an environment and obtains a reward.
Based on the reward, agent should reinforce its behaviour for better rewards

RL Introduction

Rewards:	Agent:	Environment:	Actions:	Observations:
• A scalar obtained from the environment • Reflects how well the agent behaved • The reward should reinforce the agent in a negative or positive way	• Observe and Performs actions on the environment obtain rewards	• A space that changes when actions are performed. The environment provides observations and rewards to the agent	• The moves allowed by the environment and depends on the rules of the game • Actions can be discrete or continuous	• The second input channel to the agent • Observations help agent to decide actions

RL Methods


RL Algorithms
Model-Free RL
Model-Based RL
Policy Optimization
Q-Learning
Learn the Model
Given the Model
Policy Gradient
A2C / A3C
PPO
TRPO
DDPG
TD3
SAC
DQN
C51
QR-DQN
HER
World Models
I2A
MBMF
MBVE
AlphaZero

RL Methods

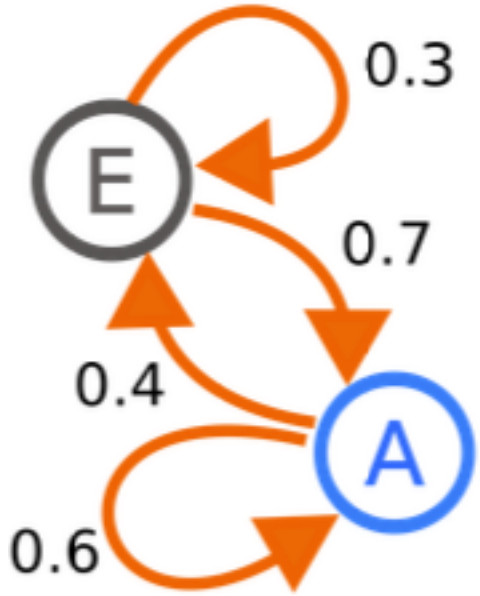
Model-Based:	• They must build a model of the environment to predict the rewards • Based on the predicted rewards, agent takes an action
Model-Free:	• They do not require a model of the environment to obtain rewards • Rather they get the reward and observation directly by interacting with the environment
Examples:	• Model based: Board game with rules • Model free: Cart pole system, robot manipulator environment

RL Methods

Policy-Based Methods:	• Policy is a probability distribution over available actions • This method approximates what action agent should take for a given state
On-Policy:	• Requires fresh samples from the environment to train for the best policy prediction
Off-Policy:	• Can train on historical data from a previous version of the agent
Value-Based Methods:	• Find the value of the state for all the actions possible • Perform the best action that maximises the value

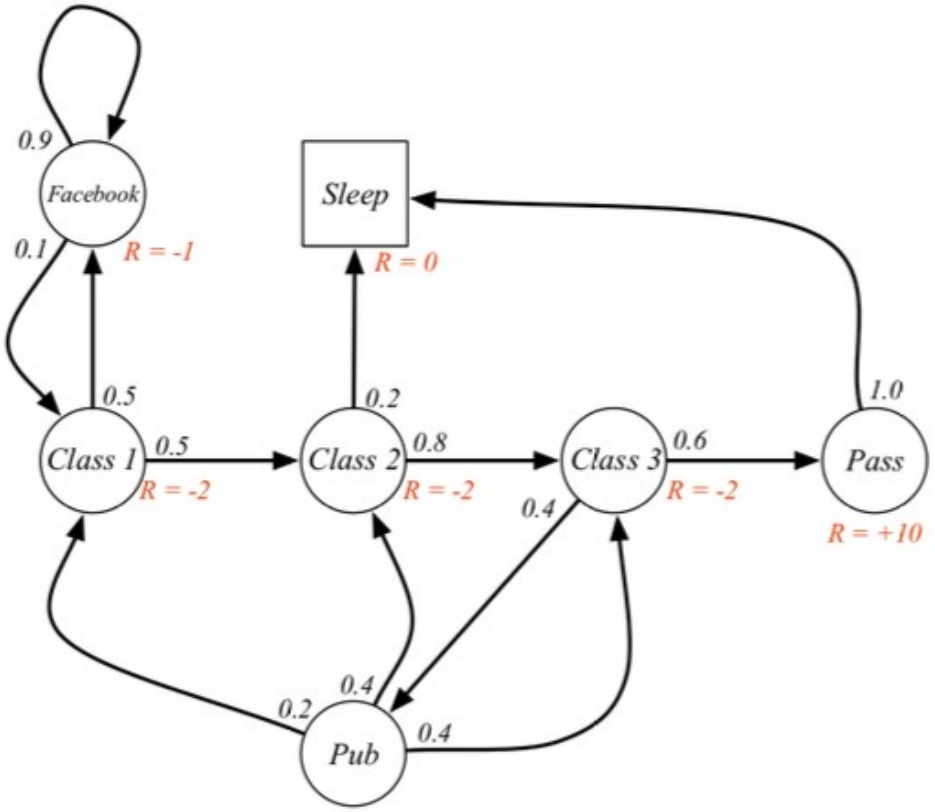
Markov Process

- Markov process (MP) is the simplest case of MDP
  - Defined by states and the transition between them
  - States: A complete observation of the environment
  - All possible states are called state space
  - Sequence of states is a Markov chain:
    - e.g: [sunny, rainy, rainy, sunny, sunny….] a history
- Extend the MP with rewards, then it becomes a Markov Reward Process (MRP)
- MRP extended with actions called the Markov Decision Process (MDP)

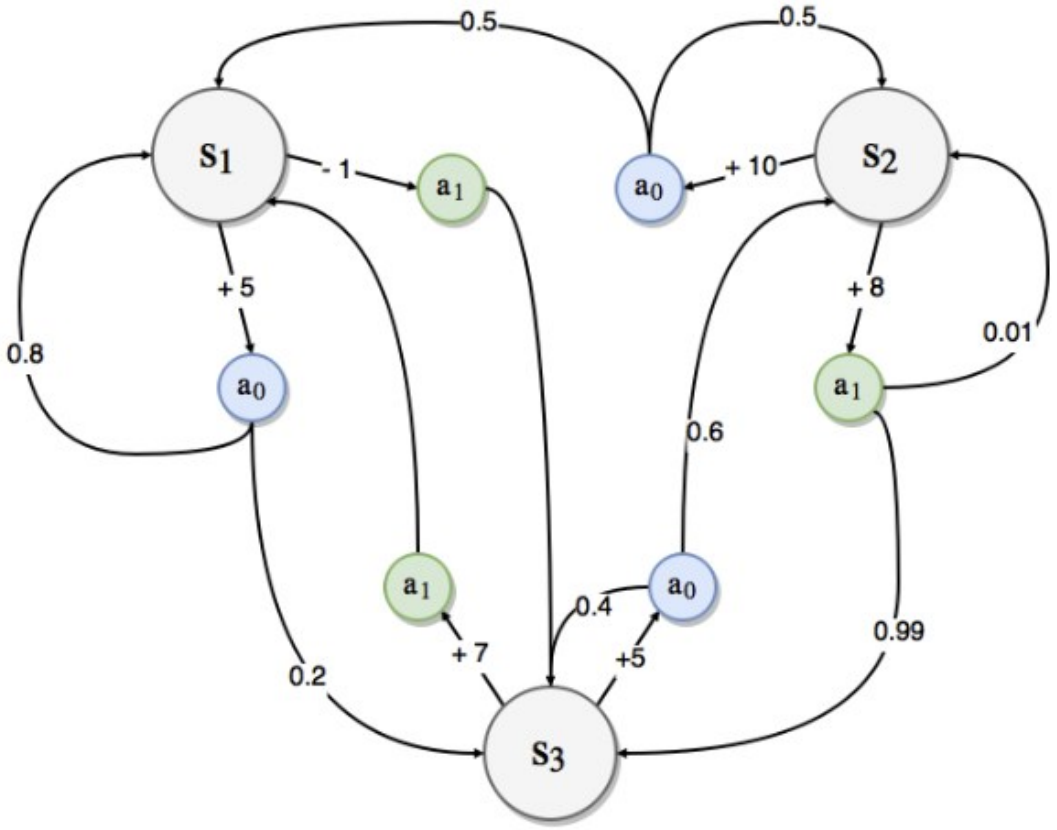


A Markov Process with states and State transition probabilities

Markov Process



A Markov Reward Process. The probability of transition and the reward for each transition is necessary for a MRP



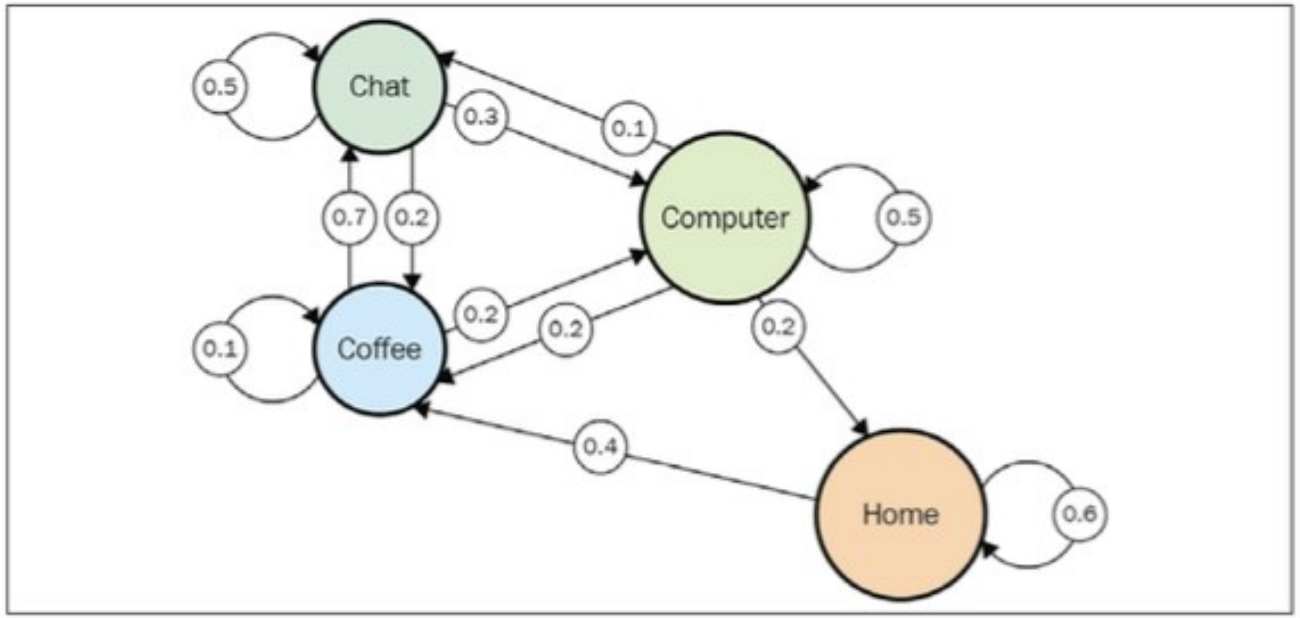
A Markov Decision Process. In this case, the transition happens according to a given action

Markov Process

EXAMPLE

- The state transition in matrix form

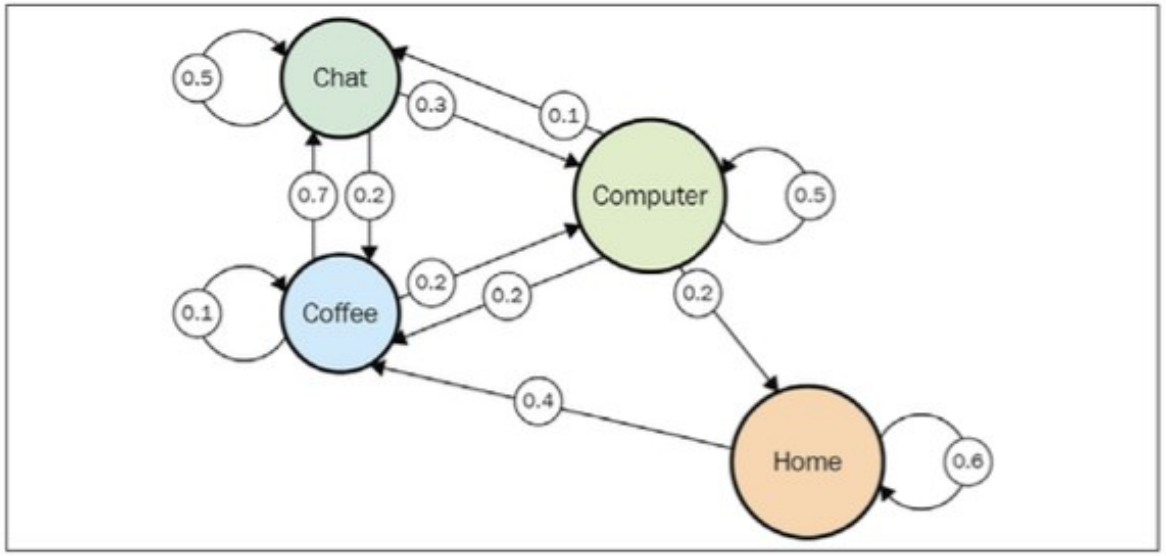
	Home	Coffee	Chat	Computer
Home	60 %	40 %	0	0
Coffee	0	10 %	70 %	20 %
Chat	0	20 %	50 %	30 %
Computer	20 %	20 %	10 %	50 %



- Sequence of system states makes an episode:
  - Home -> Coffee -> Computer -> Computer -> Chat -> Coffee -> Computer -> Home
  - Home -> Coffee -> Computer -> Chat -> Coffee -> Computer

Markov Reward Process

EXAMPLE



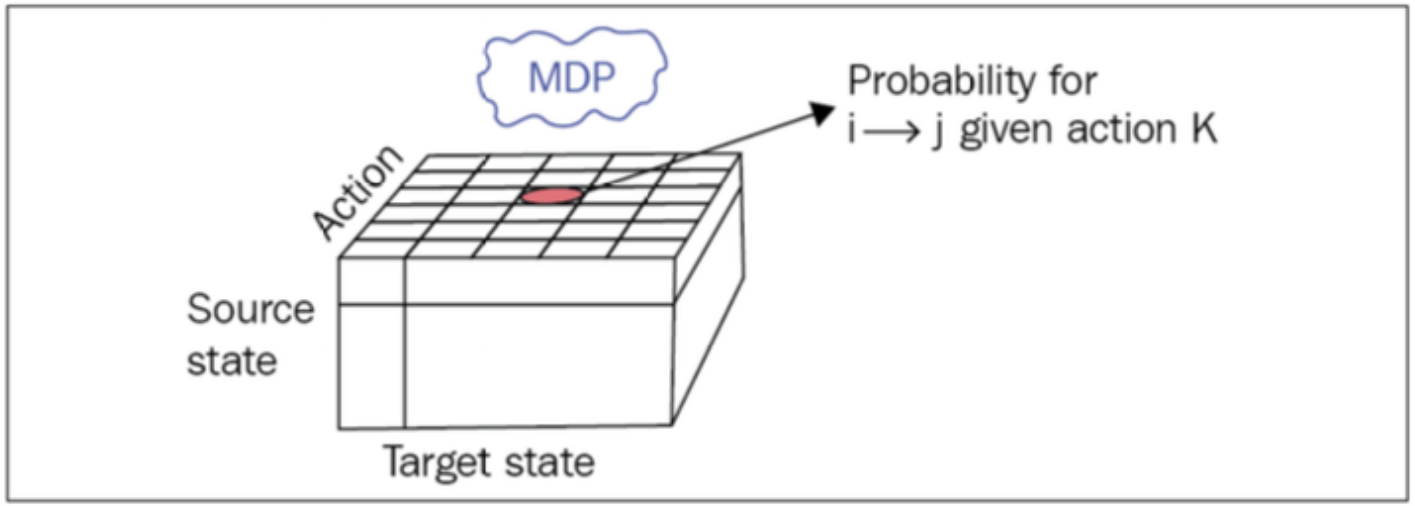
	Home	Coffee	Chat	Computer
Home	60 / 1	40 / 1	0	0
Coffee	0	10 / 1	70 / 2	20 / 3
Chat	0	20 / 1	50 / -1	30 / 2
Computer	20 / 2	20 / 1	10 / -3	50 / 5

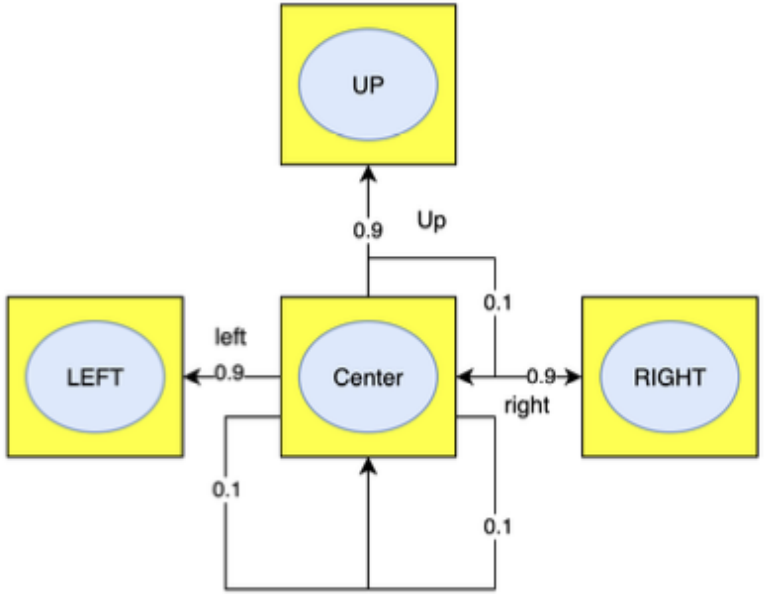
- We can get the value of each state as follows:
  - *V(coffee)* = 0.2 * 3 + 0.7 * 2 + 0.1 * 1 = 2.1
  - *V(computer)* = 0.5 * 5 + 0.1 * −3 + 0.2 * 1 + 0.2 * 2 = 2.8

Markov Decision Process

EXAMPLE

- Adding the actions to a MRP makes it a cube where now the actions determine the transitions

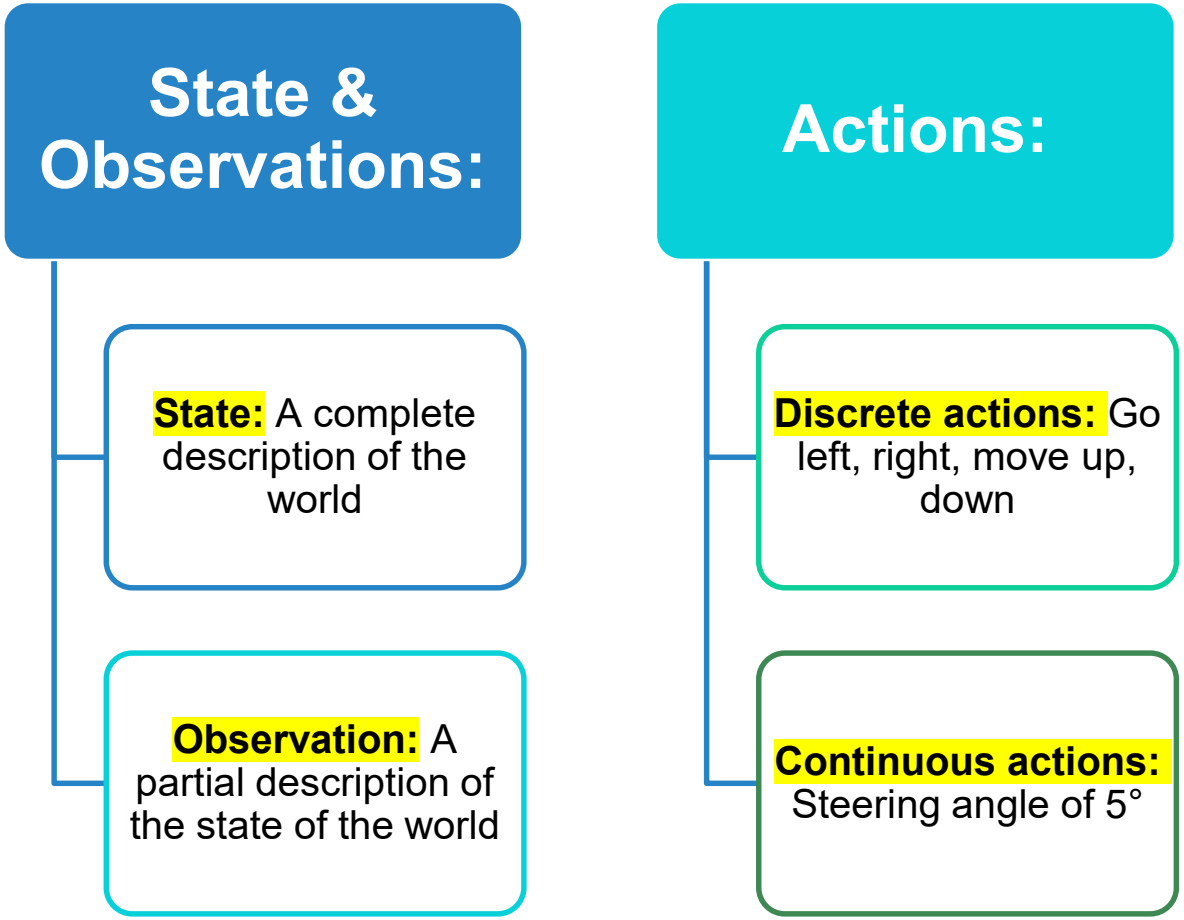




- E.g: A robot in a grid world can execute discrete actions; go left, right, or up
- Each action has a transition probability of 90% and 10% stay in the same place
- Goal is to find the best action for a given state

Terminology

RL Terminology



RL Terminology

RL Policy:

- A policy is a set of rules that governs the behavior of an agent
- An agent can perform different policies and end up in different states (from the current state)
- With different policies agent will gather different rewards
- Goal of the agent is to find the best policy that maximizes the reward

Deterministic Policy Definition:

$$a = \mu(s_t)$$

Stochastic Policy Definition:

where the A_t, S_t action and state at step t

$$a \sim \pi(\,.\,|\,s) = P(A_t = a\,|\,S_t = s)$$

For discrete actions: Categorical Policies (sample from a categorical distribution)

RL Terminology

Trajectory:

- A trajectory is a sequence of state action pairs
- $\tau = (s_0, a_0, s_1, a_1, \ldots)$
- Deterministic transition: $s_{t+1} = f(s_t, a_t)$
- Stochastic transition: $s_{t+1} \sim P(\,.\,|\, s_t, a_t)$

Rewards & Returns:

- Reward function: $r_t = R(s_t, a_t, s_{t+1})$
- Finite horizon undiscounted return: $R(\tau) = \sum_{t=0}^{T} r_t$
- Infinite horizon discounted return: $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r^t$; γunt faction [0, 1.0]

Mathematical Formulas

State value function for a given policy π	$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau \sim \pi}[R(\tau) \mid s]$
Action value function for a given policy π	$Q^{\pi}(s,a) = \mathop{\mathbb{E}}_{\tau \sim \pi}[R(\tau) \mid s,a]$
Optimal value function for a given state	$V^{*}(s) = max_{\pi} \mathop{\mathbb{E}}_{\tau \sim \pi}[R(\tau) \mid s]$
Optimal action value function for a given state action	$Q^{*}(s,a) = max_{\pi} \mathop{\mathbb{E}}_{\tau \sim \pi}[R(\tau) \mid s,a]$
Optimal action α* for an optimal value function Q*(s,α)	$a^{*} = argmax_{a} Q^{*}(s,a)$

Mathematical Formulas

Bellman equation:

- The initial reward you get for being in the state + the discounted future rewards
- State and action value function for a given policy π:
  -
  - $$V^{\pi}(s) = \mathop{\mathbb{E}}_{a\sim\pi, s'\sim P}[r(s,a) + \gamma V^{\pi}(s')]$$
- Optima $$Q^{\pi}(s,a) = \mathop{\mathbb{E}}_{s'\sim P}[r(s,a) + \gamma \mathop{\mathbb{E}}_{a'\sim\pi}[Q(s',a')]]$$
  -
  - $$V^{*}(s) = max_{a} \mathop{\mathbb{E}}_{s'\sim P}[r(s,a) + \gamma V^{*}(s')]$$

    $$Q^{*}(s,a) = \mathop{\mathbb{E}}_{s'\sim P}[r(s,a) + \gamma max_{a'}[Q^{*}(s',a')]]$$

End of Session 4