LECTURER: TAI LE QUY

INTRODUCTION TO REINFORCEMENT LEARNING

UNIT 2

SEQUENTIAL DECISION PROCESS

STUDY GOALS

- Identify various scenarios in which sequential decision-making is necessary
- Explain the interactions between reinforcement learning agents and their environment
- Evaluate the importance of rewards in reinforcement learning
- Apply Markov decision processes to solve reinforcement learning problems

EXPLAIN SIMPLY

1. Explain how reinforcement learning algorithms solve sequential decision problems, and how do agent actions affect the future

2. Describe the interaction between reinforcement learning agents and their environments

3. Compare model-based and model-free reinforcement learning algorithms

SEQUENTIAL DECISION PROCESS

ROBOTICS

Efficient decision-making

DATA CENTERS

Resource utilization optimization

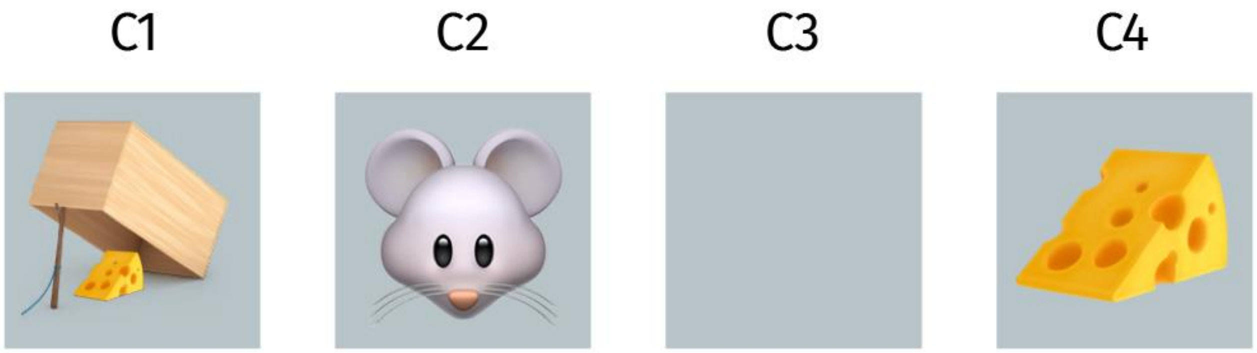
GAMES

Game strategy development

RECOMMENDER SYSTEMS

Personalized content recommendation

Making decisions over time based on environment perception

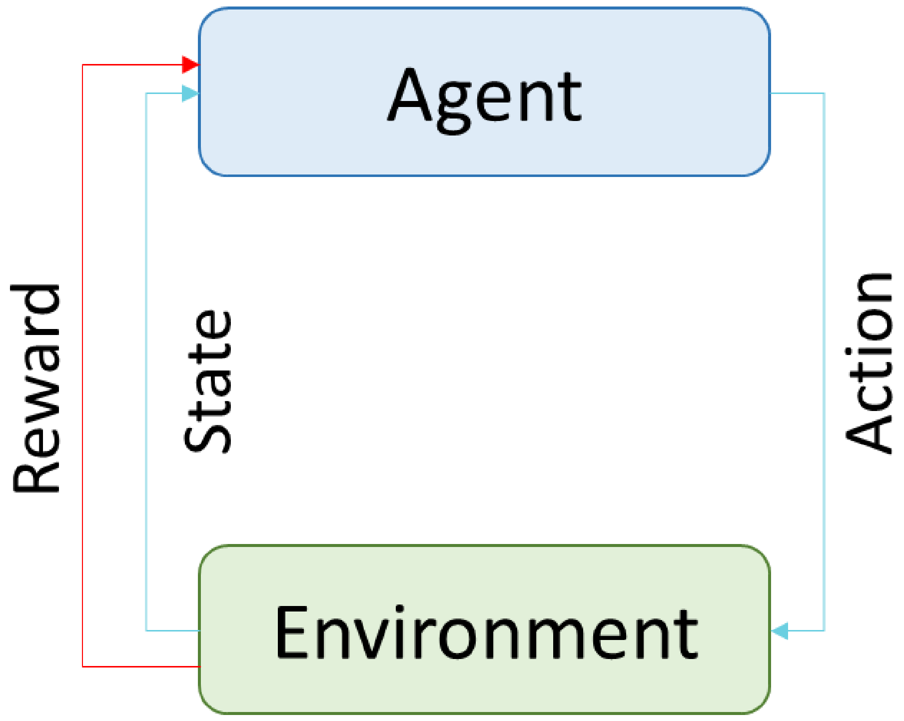


- Temporality: order of events over time
- Trajectory: sequence of states and actions $\tau = (s_0, a_0, s_1, a_1, ...)$

Source of the image: Nair, 2023.

Agent perceives environment and acts upon it

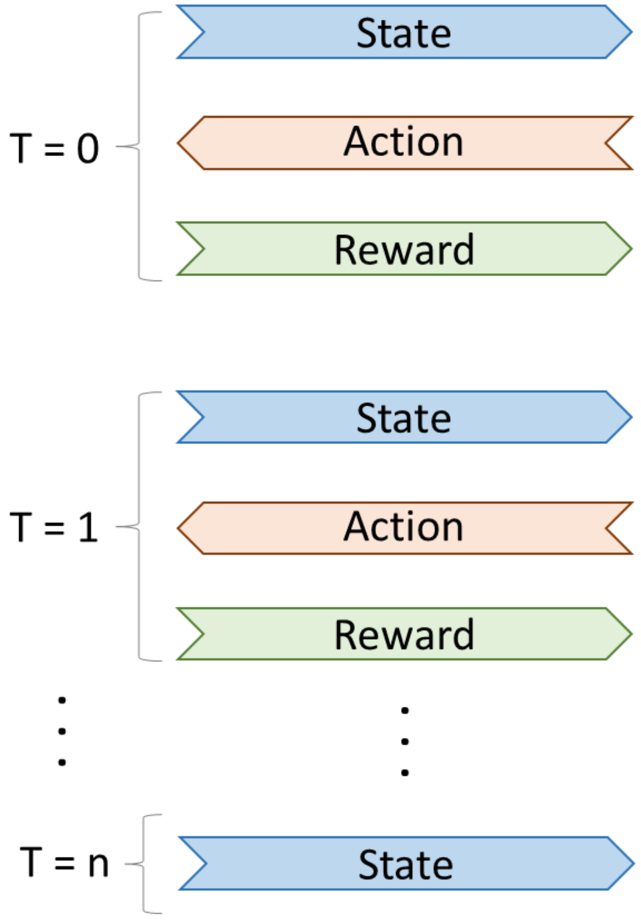
- Address uncertainty in long trajectories
- Use past interactions to adjust future encounters
- Weigh different outcomes to plan future actions



Source of the image: Plaku, 2023.

Agent uses **experience** to learn how to act

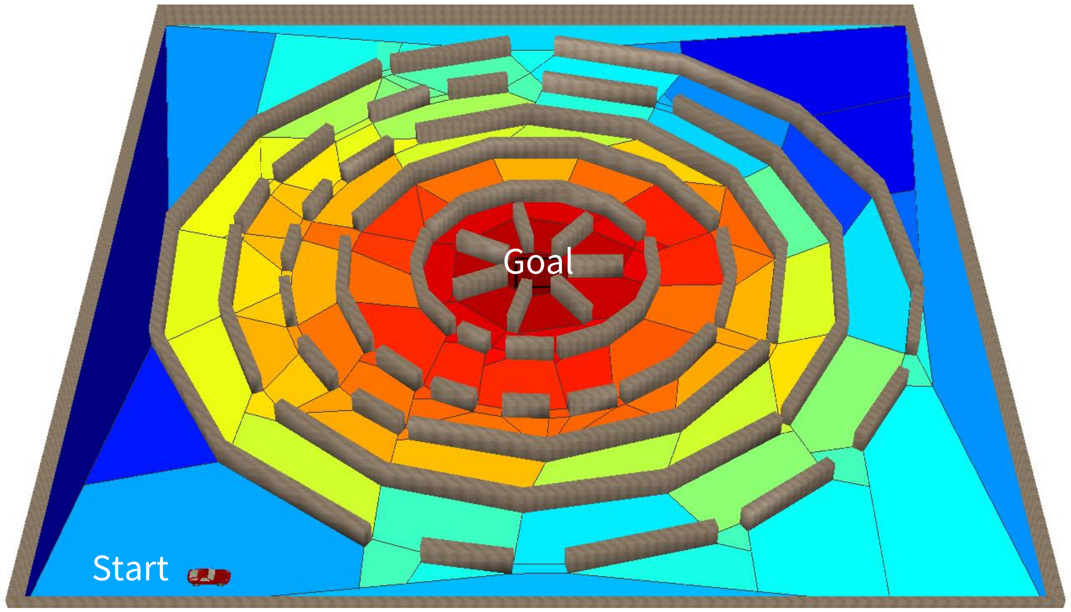
- Perceive environment state
- Select action based on state
- Generate reward based on action
- Increment time step and move to new state
- Experience < s_t, a_t, r_t, s_{t+1} >



Source of the image: Plaku, 2023, based on Nair, 2023.

State: representation of current condition of environment

- Discrete, continuous, or both
- State space is all environment configurations
- Observations are snapshots of the environment

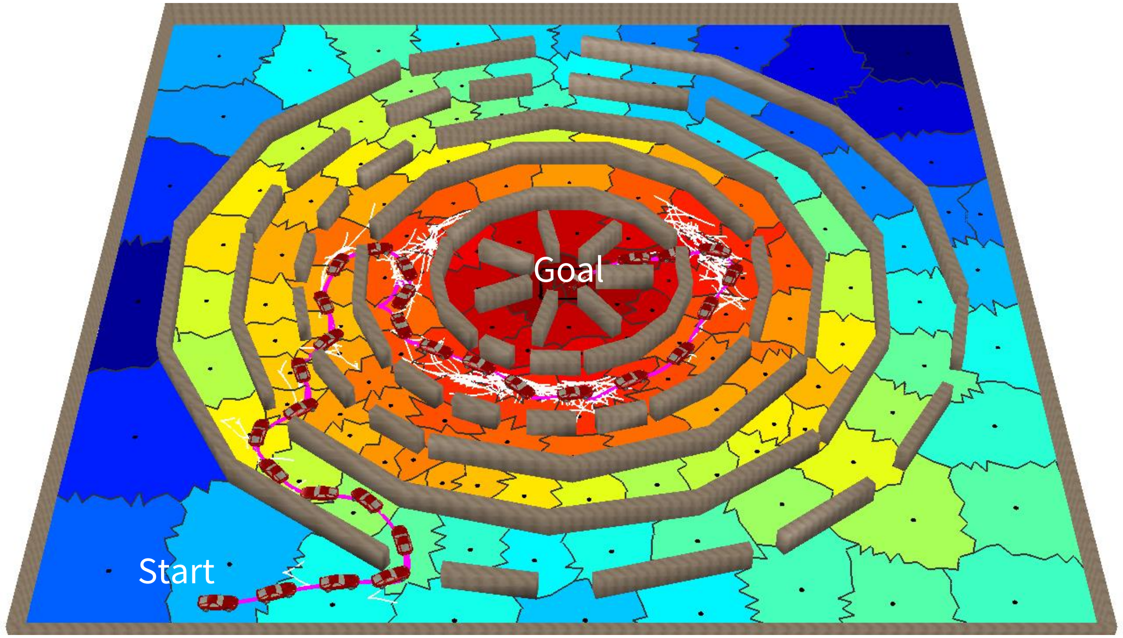


Autonomous vehicle navigating in maze-like environment.
Each region represent a state. Distance to goal is represented using color-codes
(red: close, blue: far)

Source of the text: Plaku et al., 2017
Source of the image: Plaku et.al, 2017.

Action: decision that agent executes in environment

- Influence environment state
- Discrete or continuous
- Action space is set of all possible actions
- Actions lead to next state



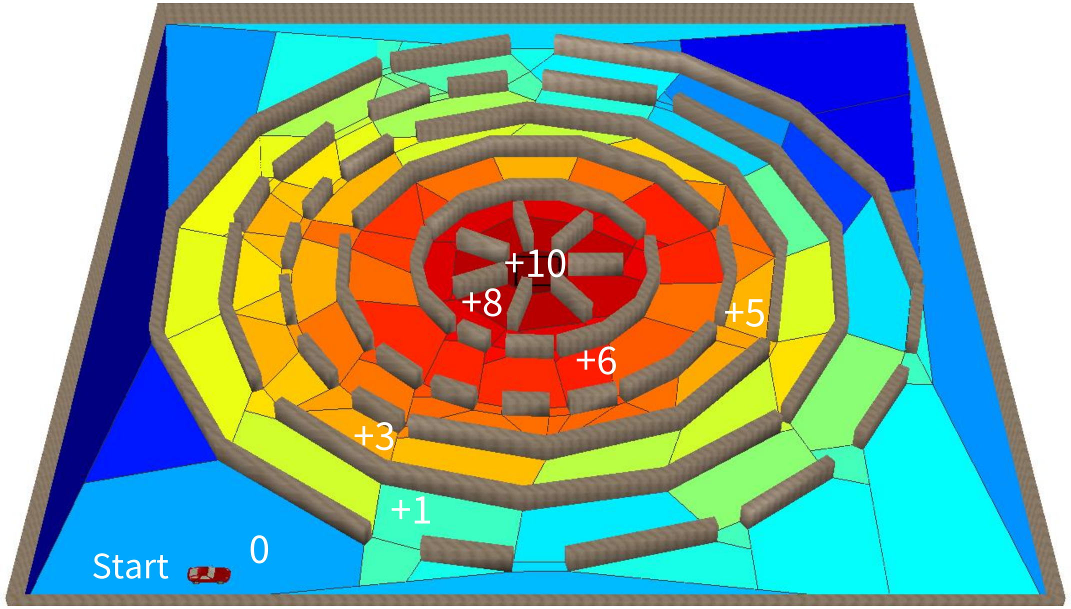
Autonomous vehicle navigating in maze-like environment.
Each action represents a possible steering command allowing the vehicle to drive in a specific direction.
Action space is shown in white. Selected action is shown by vehicle position

Source of the image: Plaku et.al, 2017.

Reward: feedback signal provided to the agent by the environment

- Goal: maximize long term reward
- Help agent discern good and bad actions
- Reward engineering to design appropriate reward functions
- Cumulative quantity

$$G = r_0 + r_1 + r_2 + \ldots = \sum_{t=0}^{T} r_t$$

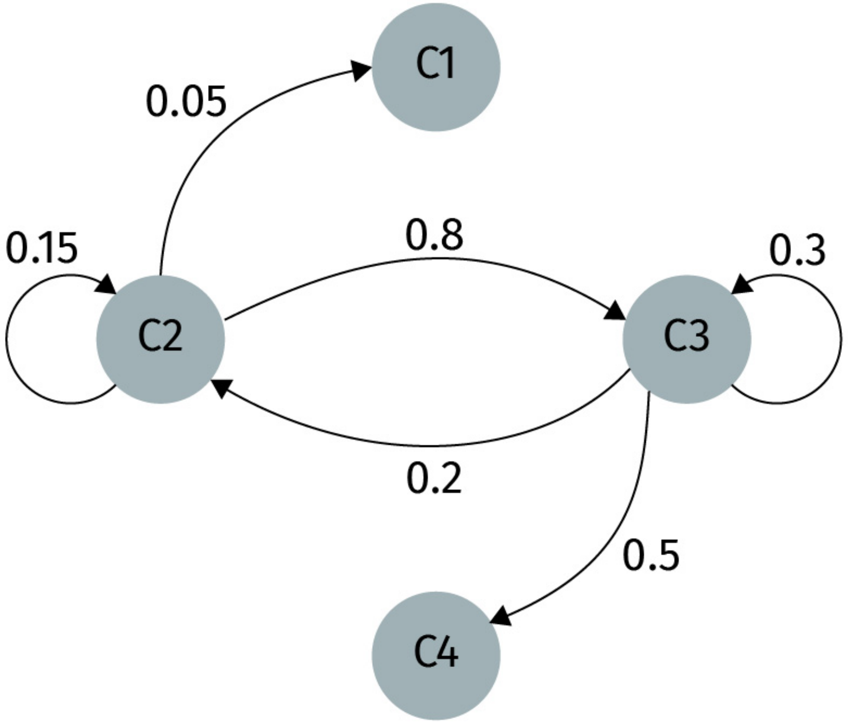


Autonomous vehicle navigating in maze-like environment.
The reward values decrease as the distance to the goal increases, with higher rewards given to regions closer to the goal.

Source of the image: Plaku et.al, 2017.

The future is independent of the past given the present

- Markov decision: model a sequence of possible events
- Markov process is memoryless and random
- Capture relevant information in current state



Markov process for the mouse grid world

Russian mathematician Andrey Markov (1856-1922)

Source of the image: Nair, 2023.

MARKOV PROPERTY

- Each state, S_t, captures all relevant information needed to predict the next state S_{t+1}. Hence, all history, S_0, S_1, …, S_{t-1} leading up to S_t is no longer required and can be discarded.
- $P(S_{t+}1|S_t) = P(St + 1|S_0, S_1, S_2, \ldots, S_t)$
- A state transition probability

$$\rho_{ss'} = \mathrm{P}\left(\mathrm{s}_{\mathrm{t}+1} = \mathrm{s}' \middle| \mathrm{s}_{\mathrm{t}} = \mathrm{s}\right)$$

MARKOV PROPERTY

- State transition matrix

$$\rho = \begin{pmatrix} \rho_{11} & \cdots & \rho_{1n} \\ \vdots & \ddots & \vdots \\ \rho_{n1} & \cdots & \rho_{nn} \end{pmatrix} \qquad \rho = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.05 & 0.15 & 0.8 & 0 \\ 0 & 0.3 & 0.2 & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- A Markov process <*S*, *P*>
- Function P defines the probability of transitioning from one state to another in a single step

$$P : S \rightarrow \rho$$

- Markov processes allow us to define the environment as a collection of states and transition probabilities

MARKOV DECISION PROCESS

State
- Possible condition of system

Action
- Choices available in each state

Reward
- Feedback from environment

Transition
- Probabilities of moving between states

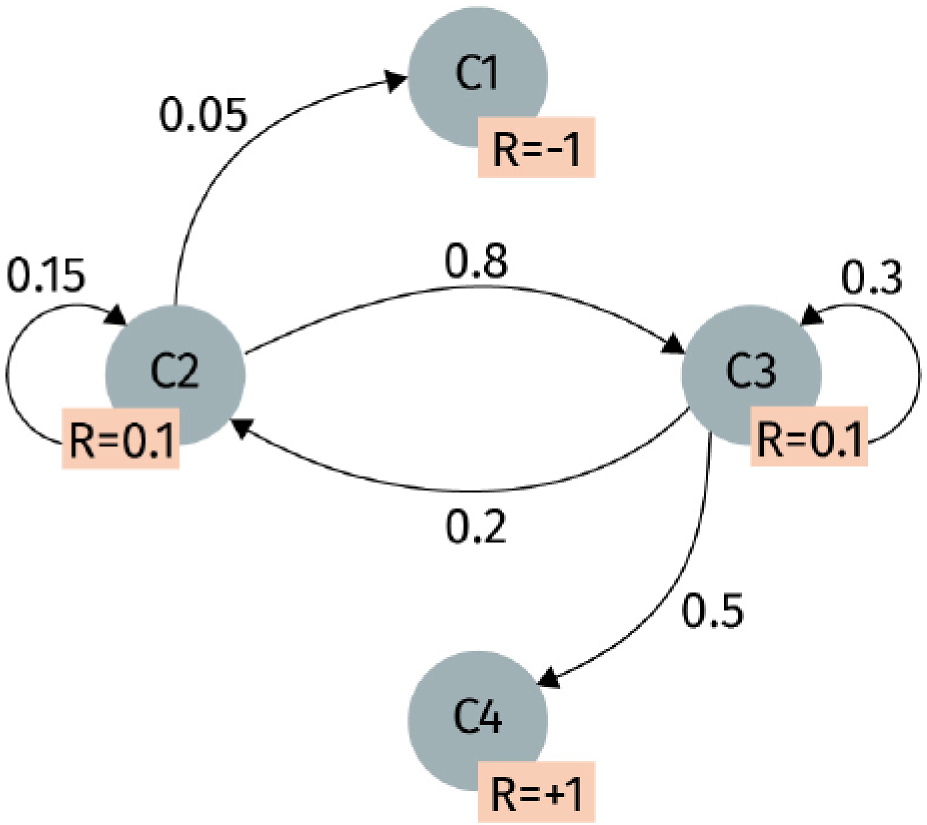
- Markov reward process < S, P, R >
- Reward function

$$R: S \rightarrow \mathbb{R}$$

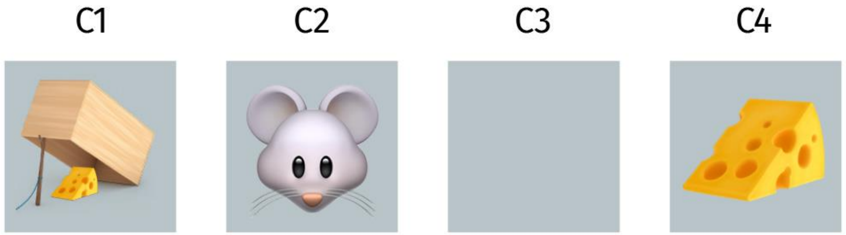
- Reward at time t

$$r_t = R(s_t = s)$$

- However, the aim of reinforcement learning is to learn to act

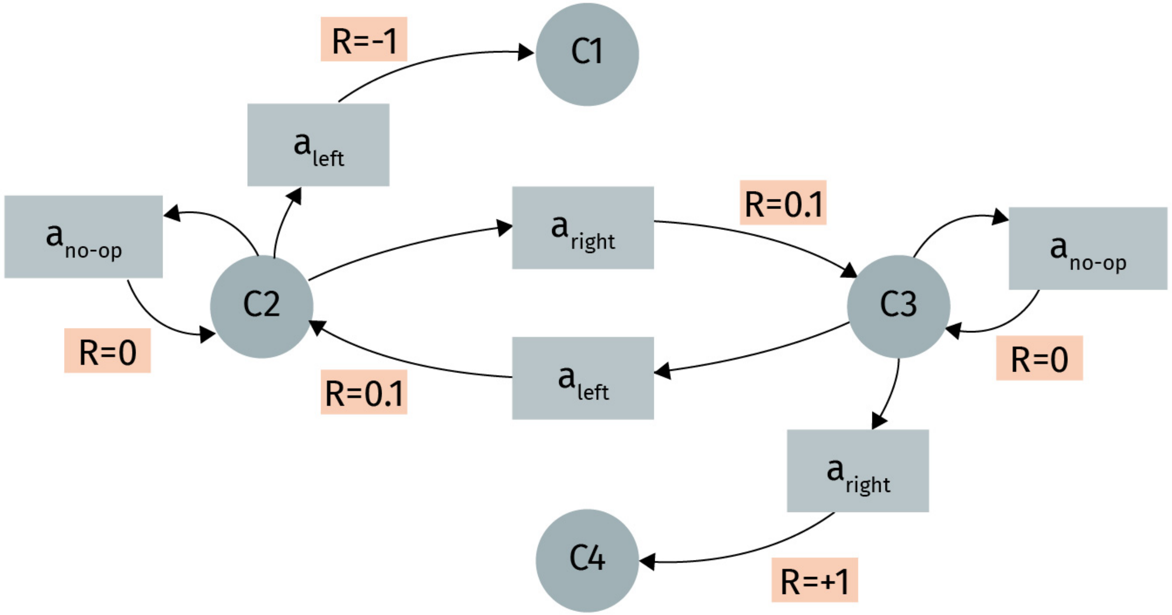


Mouse in a Markovian world: solving a decision problem



The mouse grid world

Mouse MDP: State-Action-Reward Probabilities



Source of the image: Nair, 2023.

- A Markov decision process is a 4-tuple < S, A, R, P >
- Reward function

$$R\colon S \times A \rightarrow \mathbb{R}$$

  - Reward value at time step t $$r_t = R(s_t = s, a_t = a)$$
- Transition function

$$P\colon S \times A \rightarrow \rho$$

  - State transition probability for going from state s to s′

$$\rho_{ss'} = \mathrm{P}\left(\mathrm{s}_{\mathrm{t}+1} = \mathrm{s}' \middle| \mathrm{s}_{\mathrm{t}} = \mathrm{s}, \mathrm{a}_{\mathrm{t}} = \mathrm{a}\right)$$

- Model-based RL: an agent has access to or can explicitly learn a model, i.e., the transition function, of the environment
  - Exhibit intelligent behaviors, such as planning by thinking ahead and weighing all future options
- Model-free: agents that learn directly from their experiences without explicitly building a complete model of the world
  - For most real-world problems, the ground truth model of the environment simply does not exist
  - Assumption that the environment is powered by a Markov decision process

REVIEW STUDY GOALS

- Identify various scenarios in which sequential decision-making is necessary
- Explain the interactions between reinforcement learning agents and their environment
- Evaluate the importance of rewards in reinforcement learning
- Apply Markov decision processes to solve reinforcement learning problems

SESSION 2

TRANSFER TASK

Case study

A delivery company wants to optimize its delivery process by minimizing delivery time and cost. The task involves determining the optimal delivery route, considering factors such as distance, traffic, package weight, and customer satisfaction

Task

Use MDP to model delivery as a Sequential Decision process.
Identify key components: states, actions, rewards, transitions.
Discuss how to optimize using MDP

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.

1. The sequence of states followed by actions performed by the agent in given environment is called
   a) State space
   b) Action space
   c) Reward
   d) Trajectory

2. The set of all configurations that an environment can be in is called
   a) State space
   b) Action space
   c) Reward function
   d) Transition function

LEARNING CONTROL QUESTIONS

3. Which of these formalize a Markov Decision Process?
   a) <S, P>: state space, transition function
   b) <S, P, R>: state space, action space, reward function
   c) <S, A, P, R>: state space, action space, transition function, reward function
   d) <S, P, R, τ>: state space, action space, reward function, trajectory

LIST OF SOURCES

Text

Plaku, E., Plaku, E., & Simari, P.D. (2017). Direct Path Superfacets: An Intermediate Representation for Motion Planning. *IEEE Robotics and Automation Letters, 2*, 350-357.

Images

Nair, 2023.

Plaku et al, 2017.

Plaku, 2023.

Plaku, 2023, based on Nair, 2023.