1 Introduction

- Making sequential decisions in an environment with uncertainty
- Sequential (not episodic)
- Fully (not partially) observable
- Stochastic (not deterministic)

2 Defining a Markov Decision Process

A robot is situated in a grid world with 4 columns and 3 rows.

	1	2	3	4
1	Start			
2		X		-1
3				+1

The states

- Each square is denoted by s_{ij} where i and j are the row and column positions respectively.
- The initial state is s_{11}.
- There is a wall in s_{22} and the robot cannot occupy it.
- The goal states are s_{24} and s_{34}. When the robot reaches a goal state, it escapes this world.

The environment is fully observable – The agent knows where it is.

The environment is stochastic – An action does not always achieve its intended effect.

The actions: up, down, left, right. All four actions are possible in every state.

The transition model $P(s'|s, a)$:

- An action achieves its intended effect with probability 0.8.
- An action leads to a 90-degree left turn with probability 0.1.

- An action leads to a 90-degree right turn with probability 0.1.
- If the robot bumps into a wall, it stays in the same square.

The transitions are Markovian: The probability of reaching state s' from state s depends only on state s and not on the history of earlier states.

The reward function $R(s)$ denotes the reward of entering a state s.

- The reward of entering s_{24} is -1.
- The reward of entering s_{34} is 1.
- The reward of entering any other square is -0.04.

For now, the total utility for a sequence of states is just the sum of the rewards received.

To sum up, what is a Markov decision process?

- A sequential decision problem
- The environment is fully observable - The agent knows the state it is in.
- The environment is stochastic - An action may not have its intended effect.
- A Markovian transition model: the transition only depends on the current state and does not depend on the history of states.
- A set of states, a set of actions in each state, a transition model, and a reward function.

3 What does a solution to a MDP look like?

- If the environment is deterministic, is "down, down, right, right, and right" an optimal policy?
- What would happen if we follow a fixed sequence of actions, say down, down, right, right, and right?
- Can a fixed sequence of actions be the optimal solution to a MDP?

We call a solution of this kind a policy, denoted by π. $\pi(s)$ denotes the recommended action when we are in state s.

How do we compare different policies and find the optimal policy? We can calculate the expected reward/utility of each policy. (You will see that we won't need to calculate the expected reward/utility directly.) The optimal policy, denoted by π^*, is the one that yields the highest expected utility.

4 The optimal policy of a MDP

The optimal policy of a MDP shows a careful balancing of risk and reward. It changes depending on the rewards for the non-goal states (-0.04).

When $R(s) = -0.04$, the optimal policy is as follows.

When $R(s) < -1.6284$, what does the optimal policy look like?

When $-0.4278 < R(s) < -0.0850$, what does the optimal policy look like?

When $-0.0221 < R(s) \leq 0$, what does the optimal policy look like?

When $R(s) > 0$, what does the optimal policy look like?

5 Modeling Utilities Over Time

5.1 Is there a finite or an infinite horizon for decision making?

- Finite horizon: There is a fixed number of time periods left. After that, game is over and nothing matters.

  If there are 3 days left, at state s_{13}, we need to aggressively move towards s_{34} to have a shot of getting there. If there are 100 days left, at state s_{13}, we can safely take the longer route to avoid s_{24}.

  With a finite horizon, the optimal action in a state may change over time. The optimal policy is non-stationary.

- Infinite horizon: There is no end time/deadline.

  There is always an infinite amount of time left. We should NOT behave differently in a state at different times. The optimal action in each state stays the same. The optimal policy is stationary.

We will model the problem as having an infinite horizon.

5.2 How should we calculate the utility of a sequence of states?

- Additive rewards:

$$U(s_0, s_1, s_2, ..s) = R(s_0) + R(s_1) + R(s_2) + ...$$

- Discounted rewards:

$$U(s_0, s_1, s_2, ..s) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + ...$$

  where the discount factor $0 \leq \gamma \leq 1$.

  Why should we use discounted rewards instead of additive rewards?

We will use discounted rewards.