1 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 grid is denoted by s_{ij} where i and j are the row and column positions respectively.
- The initial state is s_{11}.
- The robot cannot occupy s_{22} because there is a wall in it.
- The goal states are s_{24} and s_{34}. When the robot reaches a goal state, it has to escape this world.

The environment is full 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.

2 examples:

- The robot is in s_{13} and tries to go down. It successfully moves to s_{23} with probability 0.8, moves into s_{12} with probability 0.1 and moves into s_{14} with probability 0.1.

- (CQ) The robot is in s_{14} and tries to move right. It stays in s_{14} with probability 0.9 and moves to s_{24} with probability 0.1.

The transitions are Markovian: The future is independent of the past given the present. (Every day, we can wipe our slate clean and start over.)

The reward function:

- The reward at s_{24} is -1.
- The reward at s_{34} is 1.
- The reward in any other square is -0.04. (An incentive for the agent to reach s_{34} as quickly as possible. The agent wants to escape this world as soon as possible.)

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 future is independent of the past given the present.
- A set of states, a set of actions in each state, a transition model, and a reward function.

2 What does a solution to a MDP look like?

- If we look at this grid, it's tempted to say that going down, down, right, right, and right will get us to s_{34}. What would happen if we stick to a fixed sequence of actions?

  It could pretty much take us anywhere. For example, the sequence down, down, right, right, right could take us to any state with positive probability. It could take us to s_{34} with probability $0.8^5 \approx 0.33$. It could take us to s_{24} with probability $0.1 * 0.1 * (0.8 * 0.1 + 0.1 * 0.8)$.

- Can the solution be a fixed sequence of actions? No. a fixed sequence of actions could take us to any state with positive probability. However, after every action, we may not end up where we intended to. Thus, it is not enough to have a fixed plan. We need to keep updating our plan to adapt to the unexpected consequences. **A solution must specify what we need to do in every possible state.**

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

How do we evaluate a policy? The expected utility of a robot given a policy. Sum up the product of the probability of each sequence of states and the expected utility of visiting the sequence of states. The expected utility of a policy is.

$$\sum_{\text{state sequence}} P(\text{state sequence}) U(\text{state sequence})$$

If the number of state sequences is infinite, it may not be possible to calculate this directly.

An optimal policy, denoted by π^* is the one that yields the highest expected utility.

3 How does the reward affect the optimal policy?

	1	2	3	4
1	↓	←	←	←
2	↓	X	↓	-1
3	→	→	→	+1

The penalty of not reaching a goal state (-0.04) is pretty small. Thus, the optimal policy when we are in state s_{13} is conservative. We prefer to take the long way around to avoid reaching s_{24} by accident.

The optimal policy changes depending on the cost of taking a step. The optimal policy will carefully balance the risk and the reward.

If $R(s) < -1.6284$, life is so painful that the agent heads straight for the nearest exit, even if the exit is worth -1.

	1	2	3	4
1	→	→	→	↓
2	↓	X	→	-1
3	→	→	→	+1

If $R(s) > 0$, life is so pleasant and the agent avoids both goal states.

	1	2	3	4
1				↑
2		X	←	-1
3			←	+1

4 Modeling Utilities Over Time

Is there a finite or an infinite horizon?

- 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.

  Should we behave differently in a state at different times? No. The optimal action in each state stays the same. The optimal policy is stationary.

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 $0 \leq \gamma \leq 1$. How much does an agent prefer a reward today versus the same reward tomorrow? Everyday, there is a chance that tomorrow will not come. With an infinite sequence of states, the total additive rewards is infinite whereas the total discounted rewards is finite.

We will use discounted rewards.