Markov Decision Process

1. **Limitations of k-Armed bandit problem**

The k-bandit problem only introduces the decision with the present resources. However, it has 2 limitations:

- They do not account for the fact that different situations call for different actions
- They can only be concerned about the immediate reward, so the agent will always take the action with the best immediate outcome regardless of their impacts in the future.
-

2. **The Markov Decision Process (MDP) framework**

The Markov Decision Process can be formalized with a general framework, the agent and the environment interact at discrete time steps.

1. Each time, the agent receives a state S_t from the environment.
2. The agent gives an action A_t corresponding to the state.
3. The environment responds back with another state and the reward R_t from the last action.

When an agent chooses an action in one state, there are many states and rewards that will happen later

$$p(s', r|s, a)$$

The value of the probability p must be non-negative and it's sum over all possible next states and rewards must be one

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$$

The goal of Reinforced Learning

In the k-armed bandit problem, we maximize the immediate outcome of the state. However, in MDP framework, an action on this time step might yield large reward because the agent of transition into a state that yields low reward. Therefore, what looked good in the short-term might not be the best in the long-term.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots R_T$$

The reward hypothesis

1. Understanding intelligent behavior

The three levels of intelligence of the model can be represented by the quotes:

- “Give a man a fish and he'll eat for a day” => Programming AI
- “Teach a man to fish and he’ll eat for a lifetime” => Supervised Learning
- “Give a man a taste of fish and he’ll figure out how to fish even if the details change” => Reinforced Learning

2. Defining rewards

The reward can be defined in various ways depending on the problem. Sometimes this task can be challenging because of some reasons such as the lack of currency, complex goals, dynamic environments, risk and balance, etc.

3. Research Directions

The lecture suggests two main research areas: determining what rewards agents should optimize and designing algorithms to maximize those rewards.
Various methods for specifying rewards are explored, including programming, on-the-fly adjustments by humans, and inverse reinforcement learning, where desired behaviors inform reward structures.

Continuing tasks

1. Episodic tasks vs Continuing tasks

Episodic Tasks	Continuing Tasks
• Interaction breaks naturally into episodes • Each episode ends in a terminal state • Episodes are independent $G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots R_T$	• Interaction goes on continually • No terminal state $G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots = \infty?$

2. Discounting

In the continuing tasks, to make the sum of the return become finite, we add discount rate γ for the rewards in the future where $0 \leq \gamma < 1$

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{k-1} R_{t+k} + \cdots$$
$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

3. Recursive nature of returns

The equation for the return can be recursively written below:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$
$$= R_{t+1} + \gamma(R_{t+2} + yR_{t+3} + \gamma^2 R_{t+k} + \cdots)$$
$$= R_{t+1} + \gamma G_{t+1}$$