"*The first principle is that you must not fool yourself and you are the easiest person to fool.*" R. Feynman

Eigen Letters

Location: Mumbai, India

VOL.I. . . No.2 | *JANUARY 1, 2022* | Written & edited by: Puneet Panwar

Hey friends, this is the second edition of the Eigen Letters. I am happy to report that the first edition was a success; it could reach nearly 6K people on *LinkedIn*. I am thankful to all those who gave their valuable time to provide feedback for improving future editions. In this edition, we will talk about Markov Decision Process (MDP) for Reinforcement Learning (RL).

Markov Decision Process for RL

In mathematics, a Markov decision process (MDP) is a discrete-time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. The name of MDPs comes from the Russian mathematician *Andrey Markov* as they are an extension of Markov chains.

Terminology: before diving deep into MDP, we will familiarize ourselves with RL terminology.

1. **Agent:** agent is a software program which learns to make intelligent decisions.
2. **Environment:** environment is the model of the surrounding with which agent interacts.
3. **State:** the state of a system describes enough about the system to determine its future behaviour in the absence of any external forces affecting the system.
4. **Policy:** function to map agent's state to actions
5. **Action:** agent interacts with the environment and moves from one state to another state by performing action, **A**.
6. **Reward:** reward is the feedback the agent receives from environment for action **A**.

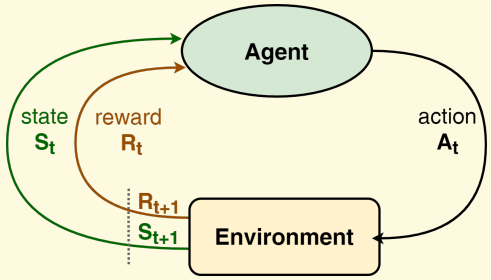

Fig. 1: Agent-Environment Interaction

In control theory, agent, environment and actions are called controller, controlled system and control signal respectively. To understand MDP, first, we need to learn about *Markov Property*, *Markov chain* and *Markov Reward Process*.

• **Markov Property**: the term Markov property refers to the memoryless property of a stochastic process, it means that the state captures all the relevant information from the history. A state **S** is Markov if and only if:

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, S_2, ..., S_t] \tag{1}$$

This means that the future is independent of the past given the present.

• **Markov Chain/Process**: Markov chain consists of a sequence of states that strictly obey the Markov Property (1). The probability of moving from one state to another state is called *transition probability* denoted by $P(S'|S)$, where S is the previous state and S' denotes the next state. Markov chain consists of a set of state and their transition matrix; its a tuple $\langle S, P \rangle$. For example, consider a Markov chain with three states A, B and C and their transition probabilities shown in Fig. 2.

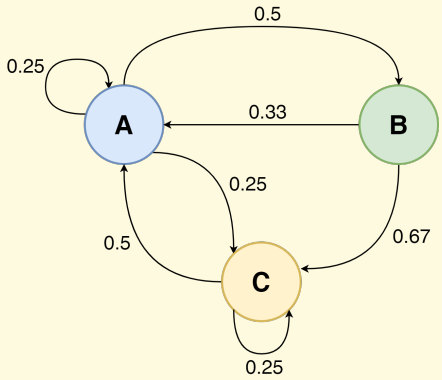

Fig. 2: State diagram of Markov chain

Fig. 2 shows the state transition diagram, in this diagram there are three possible states A, B and C and arrows from each state to other states shows the transition probabilities p_{ij}. We often list the transition probabilities in a matrix. This matrix is called the **state transition matrix** and usually shown by P. The state transition matrix for the above example is:

$$\mathrm{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.33 & 0 & 0.67 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

• **Markov Reward Process (MRP):** Markov reward process is an extension to the Markov chain with the reward function.
Markov chain = states + transition probability
MRP = Markov chain + Reward Function

$$\mathrm{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s] \tag{2}$$

$$\mathrm{R}_{ss'} = \mathbb{E}[R_{t+1} \mid S_t = s] \tag{3}$$

where P is a state transition matrix and R is the reward. This reward is received when the agent leaves the state S_t and denoted by R_{t+1}. $\mathbb{E}$ is the expected value of reward in the next time step.

The expectation or expected values is the average value of the random variable where each value is weighted according to its probability. The expectation of a random variable X with range $\{x_1, x_2, ..., x_N\}$ can be defined as:

$$\mathbb{E}(X) = \sum_{i=1}^{N} x_i p(X = x_i) \tag{4}$$

For more on expectation refer this *link*. To summarize, we can say that MRP is a tuple $\langle S, P, R, \gamma \rangle$, where γ is a discount factor; which will be discussed later.

• **Markov Decision Process (MDP):** A Markov decision process is a Markov reward process with Actions.
MDP = MRP + Actions

$$P^{a}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a] \tag{5}$$
$$R^{a}_{ss'} = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] \tag{6}$$

Eq. (4) shows the probability of moving from state s to s' while performing an action a and Eq. (5) shows the expected reward agent receives for moving from state s to s' while performing an action a. MDP is a tuple $\langle S, A, P, R, \gamma \rangle$.

with this familiarity it's time to see how MDP can be used to model Reinforcement Learning problems. Markov property states that the next state is only dependent on present state and not on the history. The major question here is, *Is Markov Property applicable to RL setting?* The answer to this is YES! The agent makes decisions only based on current state and not based on the past states.

In RL problems, we want to maximize the total rewards we receive by taking action over time. We are not told what actions lead to higher rewards. Instead, we learn it from experience. We repeat try-and-error attempts and observe which action gives us a higher reward and which gives a lower reward. Moreover, we are not even told when rewards are given in the beginning. They might be given immediately or might be given after a few time steps after we take action, called *delayed reward*. Therefore, we need a dynamic framework that captures those two features, "try-and-error search" and "delayed rewards" and MDP successfully models this. MDP captures the dynamism of RL problems by defining the dynamics function of the MDP, which is a probability distribution of pairs of a state and a subsequent reward, conditional on its previous state and the taken action there. The MDP framework is important because it accounts for a change in situations that might lead to change in optimal action, which is not considered in the k-armed bandit framework which is another framework to model RL problem (*source*). So, we can model an RL environment as an MDP. Now we will explain everything with have discussed so far using an example and also talk about other terms used in RL: *Return, discount factor, state-value function and Bellman Expectation Equation*.

Example: Student Markov Process

In this example, a student starts with class 1 and if she can make it till class 3, Tadaaa! she will pass the examination. BUT this journey is not that easy and she has multiple distractions on her way. Suppose, she is attending her first class and finds the class boring; midway shes decides to open Facebook and keeps scrolling through and ends up missing the second class or she may find a class so boring and end up falling asleep or after doing a few classes she feels very proud of herself and wants to celebrate your achievements by visiting a nearby pub.

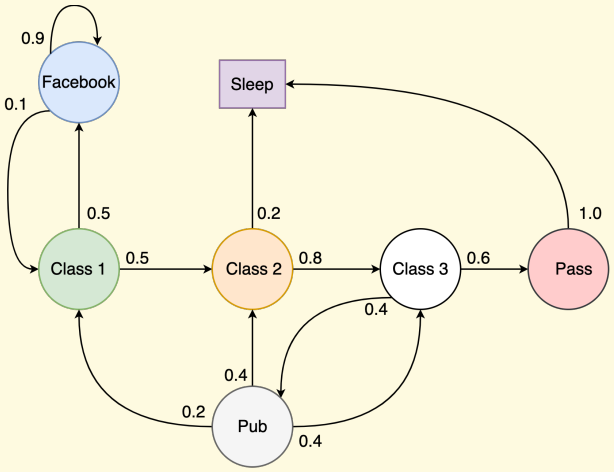

Fig. 3: Student Markov chain

Fig. 3, shows *Student Markov Process*. ***Episode*** means sequence of all the states that come in traversing from an initial-state to the terminal-state. Sample episodes for Student Markov Chain starting from $S_1 = C_1$:

1. C1 → C2 → C3 → Pass → Sleep
2. C1 → FB → FB → C2 → C3 → Pass → Sleep
3. C1 → C2 → C3 → Pub → C2 → C3 → Pass → Sleep

Now we will assign reward for every state and Fig. 4 shows how our Student MRP looks like.

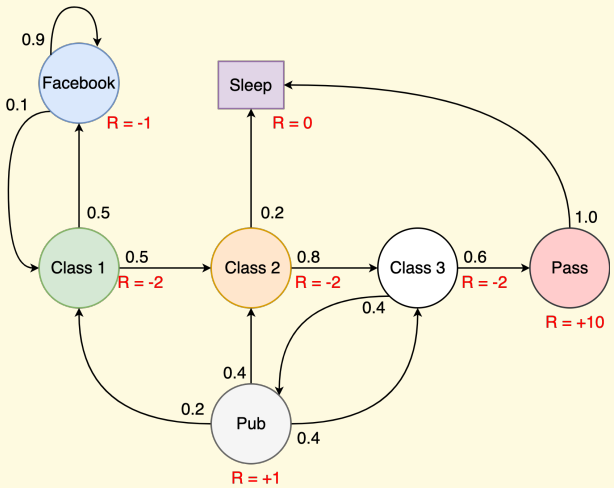

Fig. 4: Student MRP

Maybe the student doesn't like it so much in the class so we give a reward, $R = -2$ but when she passes the exam she is very happy and we assign this state a reward of +10. While scrolling Facebook feed she may feel little better and we have assigned this a reward of -1 and for Pub a reward of +1. The value judgements for reward values are subjective. What we are more interested in is the total reward in following a sequence of MRP; so we will talk about *Return* which is defined as:

• **Return (G_t):** the return is the total discounted reward from

time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (7)$$

where $\gamma \in [0, 1]$ is discount factor. If $\gamma \approx 0$, we are interested in immediate rewards and this leads to "myopic" evaluation. On the other hand, if $\gamma \approx 1$, we are more interested in delayed/future rewards and leads to "far-sighted" evaluation. When we set $\gamma = 0$, the agent will never learn as it considers only the immediate reward and when $\gamma = 1$, the agent learns forever, looking for the future rewards that lead to infinity. The value of γ is usually lies between 0.2 and 0.8. Going back to Student MRP, Starting from $S_1 = C_1$ with $\gamma = 0.5$, sample returns are calculated as follows:

$$G_1 = R_2 + \gamma R_3 + ... + \gamma^{T-2} R_T$$

The return for episode $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow Pass \rightarrow Sleep$ is calculated as:

$$V_1 = -2 - 2 * 0.5 - 2 * 0.5^2 + 10 * 0.5^3 = -2.25$$

Similarly, the return value for other episodes can be calculated. Now we will discuss about, the value function which gives long-term value of the state s.

• **State Value Function** ($v(s)$)**:** state-value function of an MRP is the expected return starting from state s.

$$v(s) = \mathbb{E}[G_t \mid S_t = s] \quad (8)$$

Fig. 5 shows the value of state-value function for different values of γ. The state function value for every state summarizes that how good it is to be in that state.

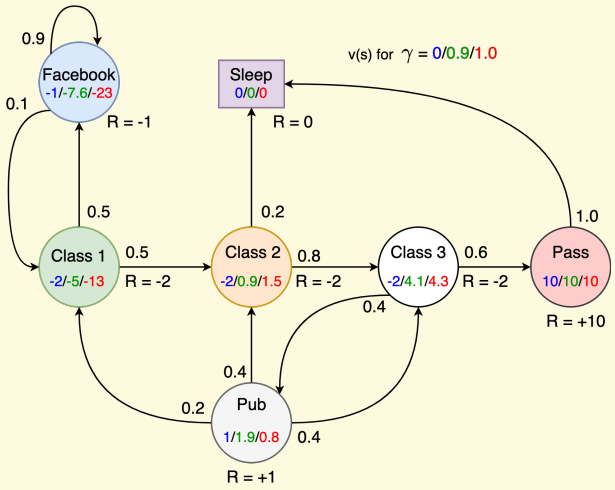

Fig. 5: SVF student MRP

• **Bellman Equation for MRPs:** Bellman equation decomposes the value function into two components: *the immediate reward*, R_{t+1} and *discounted value of the future state*, $\gamma v(S_{t+1})$.

$$\begin{aligned} v(s) &= \mathbb{E}[G_t \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + ...) \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ v(s) &= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s] \end{aligned}$$

As shown in Fig. 6, the value of state value function can be validated using Bellman equation. For example if you are in state C_3, you can either choose to go to pub or get pass.

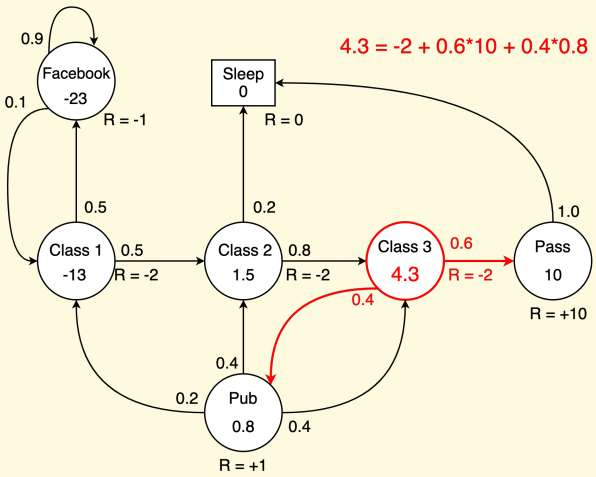

Fig. 6: Bellman Eq. for student MRP

Now we will redo our student MRP with introduction to action space. As now we have decision to make and introduction of decisions makes it: Student MDP. The decisions here can be: a student may choose to study *study*, *sleep*, *Facebook or Pub*. When we talk about taking decisions we need to formalize what it means to take decision. To do that we define a new term called *Policy*:

• **Policy** (π)**:** Policy gives the probability of picking an action a at state s. Mathematically, it may be represented as:

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s] \quad (9)$$

Once we have a policy, it completely defines how the agent is going to behave.
In this edition of Eigen Letters, we covered MDP and how it is used in RL with an example. In the next edition, I shall complete this topic by discussing how an optimal policy is achieved.

The good stuff!

Music: *Aaj Din Chadheya*, to all lovers out there :)

Quote: *"It's easier to change yourself than to change the world" - Naval Ravikant*

Tech Video: *Risking My Life To Settle A Physics Debate* on Veritasium youtube channel.

RL Resources

- UCL Course on RL by David Silver
- Reinforcement Learning: An Introduction

I hope that you liked the content of this edition and would request you to provide your valuable feedback and suggestions to improve future editions. To subscribe to *Eigen Letters*, visit *here*. Thanks for reading.

with love: Puneet Panwar