Notes on Reinforcement Learning course taught by David Silver in 2015

compiled by D. Gueorguiev, 12/25/2025

Lecture 1: Introduction to Reinforcement Learning

Definition *History*

The history is the sequence of observations O_i, actions A_i, rewards R_i for $i = 1, \ldots, t$:

$$H_t = A_1, O_1, R_1, \ldots, A_t, O_t, R_t$$

These are all observable variables up to time t.

Definition Information State

An *information state* (aka *Markov state*) contains all useful information from the history

A state S_t is Markov iff

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \ldots, S_t] \quad \text{(mkv.0)}$$

In words, the future is independent of the past given the present

$$H_{1:t} \to S_t \to H_{t+1:\infty}$$

Definition *Partially Observable Environment*

Agent indirectly observes the environment. Thus the agent state ≠ environment state

Definition *Partially Observable Markov Decision Process (POMDP)*

Agent must construct its own state representation S_t^a. Examples of such state representation are:

- Complete history: $S_t^a \equiv H_t$
- Beliefs of environment state: $S_t^a = (\mathbb{P}[S_t^e = s^1], \ldots, \mathbb{P}[S_t^e = s^n])$
- Recurrent neural network: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

Components of RL agent

- *Policy*: agent's behavior function which defines how the agent picks his action. It prescribes what action the agent should take given its current state.
- *Value Function*: how good is each state and/or action; how much reward do we expect to get if we get that particular action.
- *Model*: agent's representation of the environment; how the agent thinks the environment works.

Definition *Policy*

Policy is a map from state to action – a function $\pi : \mathcal{S} \to \mathcal{A}$ which represents mapping from states to probabilities of selecting each possible action.

If the agent is following policy π at time t, then $\pi(a|s)$ is the probability that $A_t = a$ if $S_t = s$. Note that $\pi(a|s)$ is an ordinary function which defines a probability distribution over $a \in \mathcal{A}(s)$ for each $s \in \mathcal{S}$.

Lecture 2: Markov Decision Processes

Definition *Markov Process (MP) / Markov chain (MC)*
Markov Process / Markov Chain is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$ such that
$\mathcal{S}$ is a finite set of states
$\mathcal{P}$ is a state transition probability matrix
$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$
Markov Property is in place: the transitional probabilities depend only on the most recent previous state

$$\mathcal{P} = from \overset{to}{\begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix}} \quad \text{where } \sum_{j=1}^{n} \mathcal{P}_{ij} = 1 \qquad \text{(trn.1)}$$

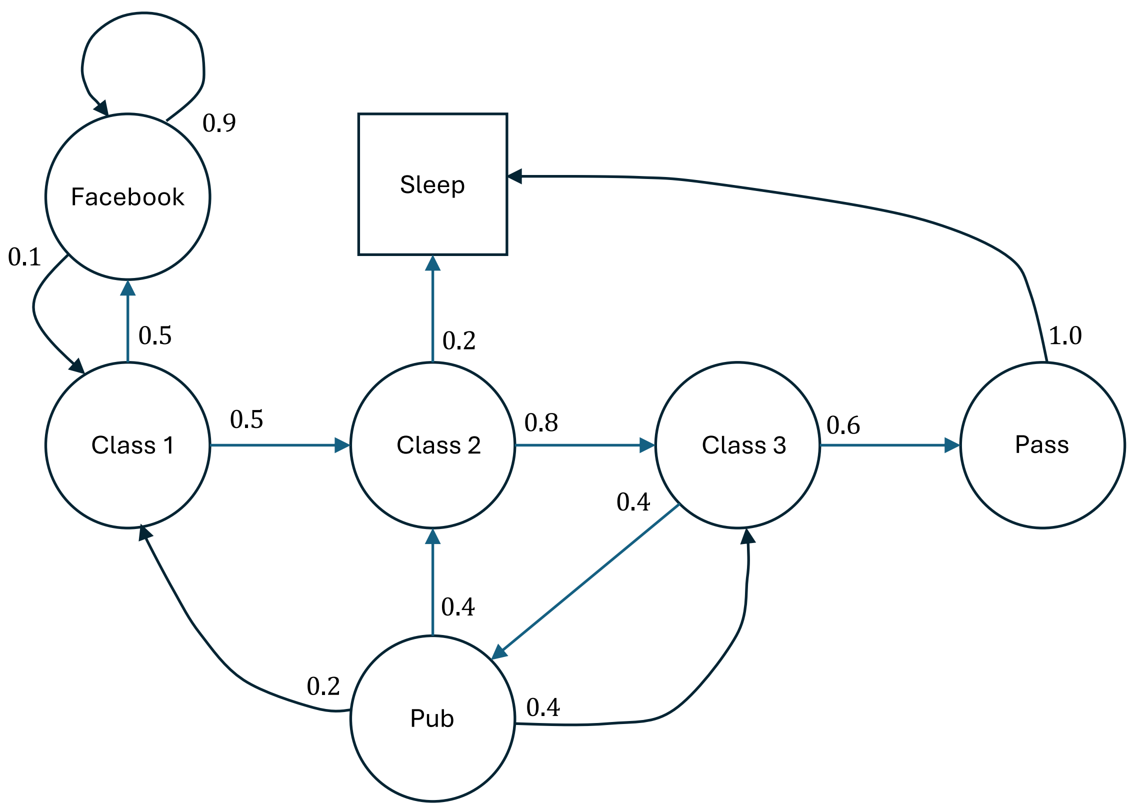

Figure : Example MP

$$\mathcal{P} = \begin{array}{rc} & \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \begin{array}{r} C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} & \begin{bmatrix} 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.0 & 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.6 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.2 & 0.4 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \end{array}$$

Definition *Markov Reward Process (MRP)*
Markov Reward Process (MRP) is the tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
$\mathcal{S}$ is finite set of states
$\mathcal{P}$ is a state transition probability matrix
$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$
$\mathcal{R}$ is the reward function
$\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$ (rwd.1)
$\gamma \in [0,1]$ is the discount factor
Markov Property is in place: the transitional probabilities depend only on the most recent previous state

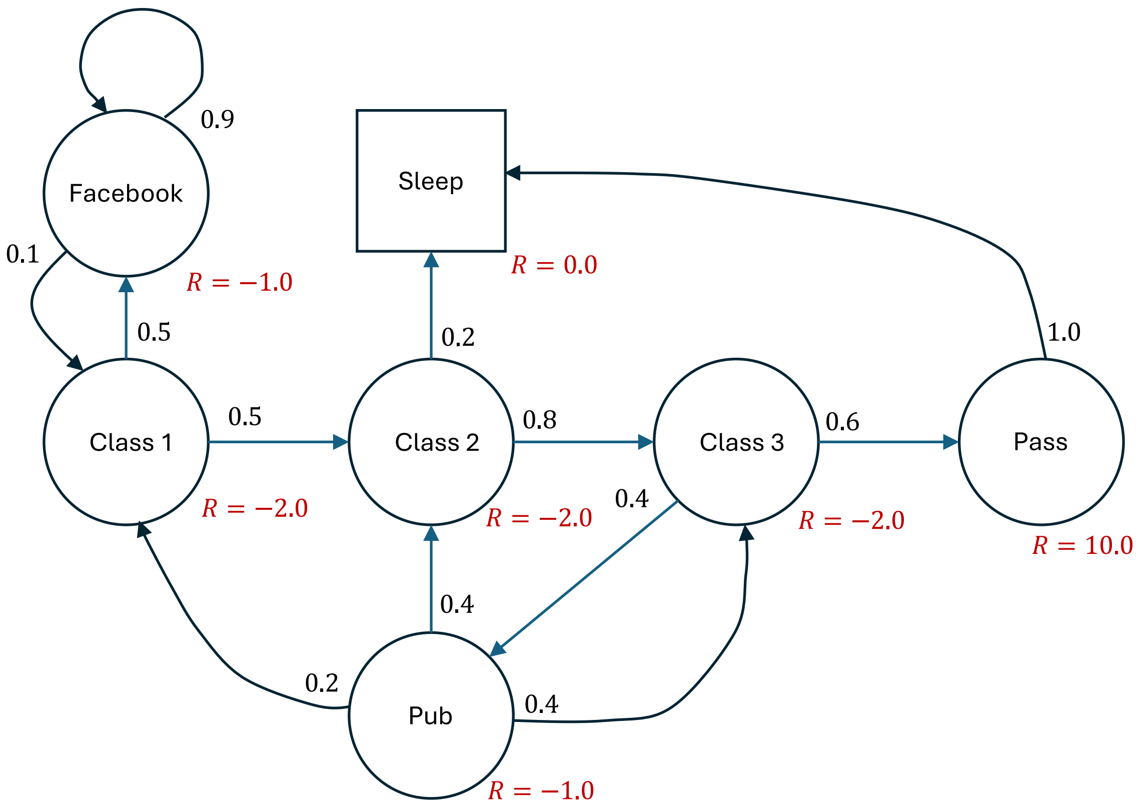

Figure : example MRP

Definition *Return*
The *return* G_t is the total discounted reward from time-step t.
$G_t = R_{t+1} + \gamma R_{t+1} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ (ret.1)

The Bellman's equation for MRP

$v(s) = \mathbb{E}[G_t | S_t = s]$ (expect.1)
$= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \cdots) | S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s]$ (expr.1)
$= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$ /* by the law of iterated expectations */ (bel.1)

Note on deriving (bel.1) from (expret.1)
We have $v(s) = \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s]$ given by (expr.1). Notice that the expectation on the RHS of (expr.1) is computed over all states which follow in time after the given state s at time t. Per (expect.1) we have $v(s') = \mathbb{E}[G_{t+1} | S_{t+1} = s']$. From the last two facts it follows that $[R_{t+1} + \gamma G_{t+1} | S_t = s]$ can be rewritten as $\mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$.

//TODO: finish the discussion on MRP

Definition *Markov Decision Process (MDP)*

Lecture 3: Planning by Dynamic Programming

Lecture 4: Model-Free Prediction

Lecture 5:

References

[1] Lecture 1: Introduction to Reinforcement Learning, David Silver, DeepMind x UCL 2015
[2] Lecture 2: Markov Decision Process, David Silver, DeepMind x UCL 2015
[3] Lecture 3: Planning By Dynamic Programming, David Silver, DeepMind x UCL 2015
[4] Lecture 4: Model-Free Prediction, David Silver, DeepMind x UCL 2015
[5] Lecture 5: Model-Free Control, David Silver, DeepMind x UCL 2015
[6] Lecture 6: Value Function Approximation, David Silver, DeepMind x UCL 2015
[7] Lecture 7: Policy Gradient Methods, David Silver, DeepMind x UCL 2015
[8] Lecture 8: Integrating Learning and Planning, David Silver, DeepMind x UCL 2015
[9] Lecture 9: Exploration and Exploitation, David Silver, DeepMind x UCL 2015
[10] Lecture 10: Classic games, David Silver, DeepMind x UCL 2015