- Markov decision process formally describe an environment for reinforcement learning.
- where the environment is fully observable.
- i.e. The current state completely characterizes the process.

Almost all RL problems can be formalized as MDPs eg.
- Optimal control primarily deals with continuous MDPs
- Partially observable problems can be converted to MDPs
- Bandits are MDPs with one state.

Markov Property

The future is independent of the past given the present.

Def: The state S_t is Markov iff —

$$P[S_{t+1} | S_t] = P[S_{t+1} | S_1, \ldots, S_t]$$

The state captures all relevant information from history. Once the state is known, history may be thrown away. i.e., state is a sufficient statistic of the future.

State Transition Matrix:

For a Markov state s and a successor state s', the state transition probability is defined by —

$$P_{ss'} = P[S_{t+1} = s' | S_t = s]$$

State transition matrix P defines transition probabilities from all states s to all successor states s'.

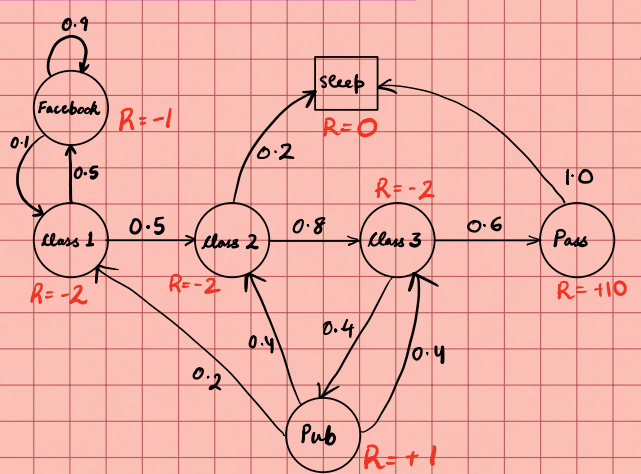
$$P = \text{from} \begin{bmatrix} & \text{to} & \\ P_{11} & \cdots & P_{1n} \\ \vdots & & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$

where each row of the matrix sums to 1.

Markov Process — It is a memoryless random process, i.e., a sequence of random states $S_1, S_2, \ldots$ with Markov property.

Def: — A Markov process or a Markov chain is a tuple $\langle S, P \rangle$:
- S is a finite set of states
- P is a state transition probability matrix, $P_{ss'} = P[S_{t+1} = s' | P_t = s]$

Example: Student Markov Chain:



Sample episodes for student Markov Chain starting from $S_1 = C_1$

Markov Reward Process: It is a Markov chain with values.

Def: Markov Reward process is a tuple $\langle S, P, R, \gamma \rangle$

- S is a finite set of states
- P is a state transition probability matrix: $P_{ss'} = P[S_{t+1} = s' \mid S_t = s]$
- R is a reward function, $R_s = E[R_{t+1} \mid S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$

Def: The return G_t is the total discounted reward from time-step t.

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

The discount $\gamma \in [0, 1]$ is the present value of future rewards.

Why discount?

- There's more uncertainity in the future.
- Mathematical convinience - avoids infinite rewards in cyclic processes
- Sometimes undiscounted rewards are used if all sequences terminate.

Value Function

Def: The state value function $v(s)$ of an MRP is the expected return starting from state s.

$$v(s) = E[G_t \mid S_t = s]$$

Bellman Equation for MRPs

The value function can be decomposed into two parts.

- immediate reward R_{t+1}
- discounted value of successor state $\gamma v(S_{t+1})$

$$\begin{aligned} v(s) &= E[G_t \mid S_t = s] \\ &= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s] \\ &= E[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \cdots \mid S_t = s] \\ &= E[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= E[R_{t+1}] + \gamma E[G_{t+1} \mid S_t = s] \\ &= R_s + \gamma \sum_{s' \in S} P_{ss'} v(s') \end{aligned}$$

Bellman eq. can be written in matrix form.

$$\begin{bmatrix} v(1) \\ \vdots \\ v(2) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_2 \end{bmatrix} + \gamma \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

$$v = R + \gamma P v$$
$$(1 - \gamma P) v = R$$
$$v = (1 - \gamma P)^{-1} R$$

- Computational complexity $\sim O(n^3)$
- Direct solution only possible for small MRPs
- More iterative processes to solve these
  - Dynamic Programming
  - Monte-Carlo Simulations
  - Temporal-Difference learning

Markov Decision Process:

A Markov Decision Process (MDP) is a MRP with decisions. It is an environment in which all states are Markov.

Definition: A MDP is a tuple $\langle S, A, P, R, \gamma \rangle$ where:

- S is a finite set of states
- A is a finite set of actions.
- P is a state transition probability matrix.
- $P^a_{ss'} = P[S_{t+1} = s' \mid S_t = s, A_t = a]$
- R is a reward function $R^a_s = E[R_{t+1} \mid S_t = s, A_t = a]$
- γ is a discount factor $\gamma \in [0, 1]$

Policy: A policy π is a distribution over actions given states,

$$\pi(a \mid s) = P[A_t = a \mid S_t = s]$$

A policy fully defines the behaviour of the model

- MDP policies depend on the current state and not the history.
- Policies are stationary (time-independent)

$$A_t \sim \pi(\cdot \mid S_t), \forall t > 0.$$

Given an MDP $M = \langle S, A, P, R, \gamma \rangle$ and a policy π.
The state sequences $S_1, S_2, S_3, \ldots$ is a Markov Process $\langle S, P^\pi \rangle$
The state and reward sequence $S_1, R_2, S_2, \ldots$ is a MRP $\langle S, P^\pi, R^\pi, \gamma \rangle$
where –

$$P^\pi_{ss'} = \sum_{a \in A} \pi(a \mid s) P^a_{ss'}$$
$$R^\pi_s = \sum_{a \in A} \pi(a \mid s) R^a_s$$

Value Function

The state-value function $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π.

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$

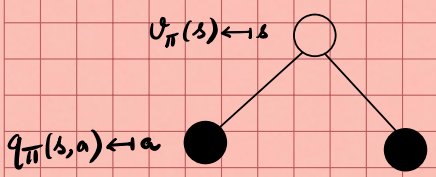
The action-value function $q_\pi(s,a)$ is the expected return starting from state s and taking action a, and then following π.

$$q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a]$$

Bellman Expectation Equation:

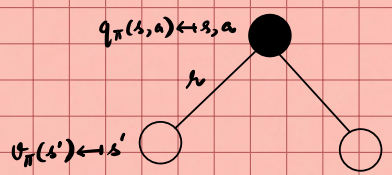
$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$



$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) q_\pi(s,a)$$

avg. over actions that we might take



we start of taking an action

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s')$$

v: how good is it to be in a particular state

q: how good is it to take a particular action

From both equations:

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) \left[R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right]$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a' \mid s') q_\pi(s', a')$$

Optimal Value Function:

Def: The optimal state value function is the max. value function over all the policies.

$$v_*(s) = \max_\pi v_\pi(s)$$

The optimal action-value function $q_\pi(s,a)$ is the maximum action-value function over all policies.

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

The optimal value function specifies the best possible performance in the MDP.

An MDP is "solved" when we know the optimal value function.

Optimal Policy:

Define a partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \forall s$$

Theorem:

For any Markov Decision Process –

- There exists an optimal policy π_* that is better than or equal to all other policies, $\pi_* \geq \pi, \forall \pi$
- All optimal policies reach the optimal value function $v_{\pi_*}(s) = v_*(s)$
- All optimal policies achieve the optimal action-value function $q_{\pi_*}(s,a) = q_*(s,a)$

Finding an Optimal Policy –

An optimal policy can be found by maximizing over $q_*(s,a)$,

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in A}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

- There is always a deterministic optimal policy for any MDP.