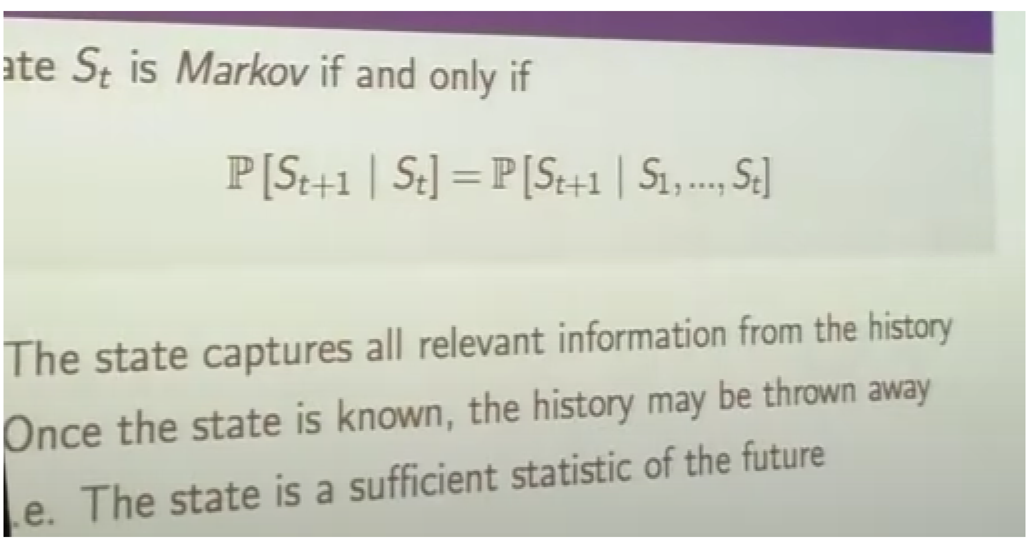
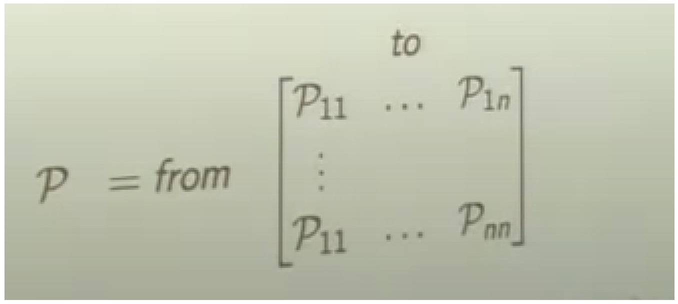
Markov Decision Process

Markov's Decision Process

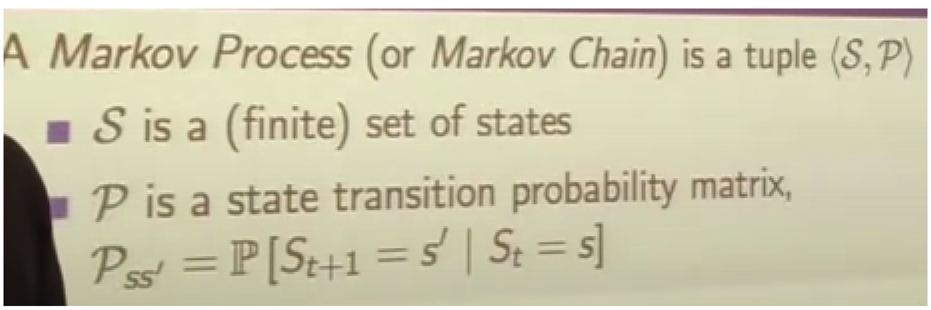
- MDP formally describes an environment for reinforcement learning.
- Almost all RL problems can be formalised as MDPs
- "The future is independent of the past given the present"



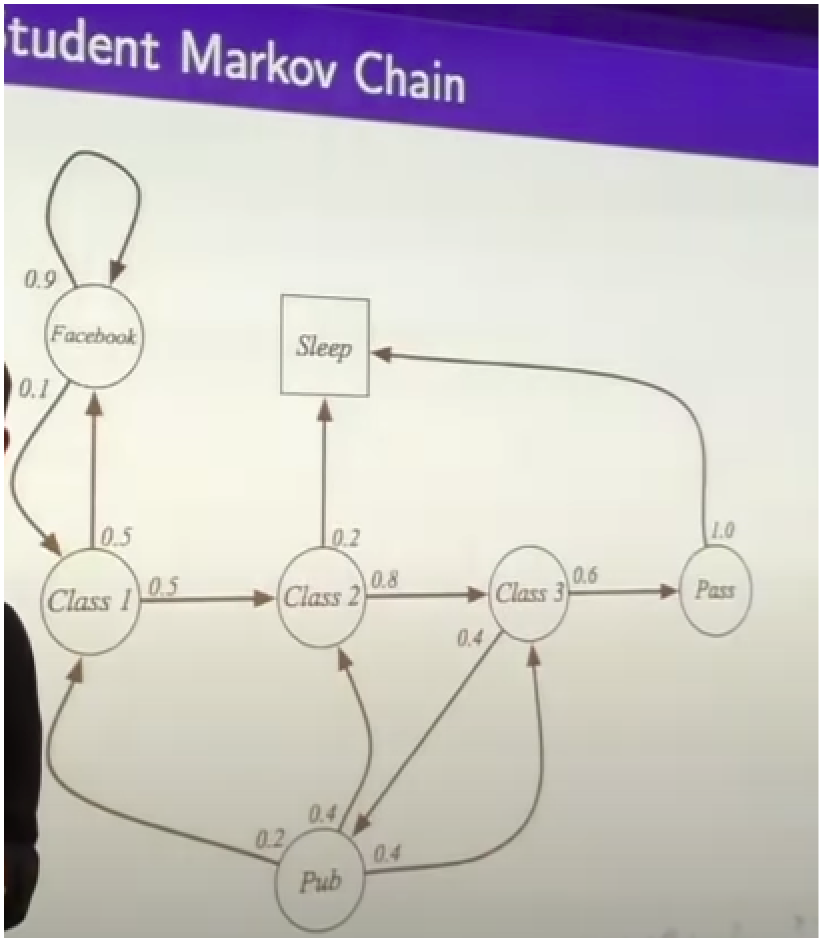
- The transition matrix defines the probabilities from all states s to all successor states s'.

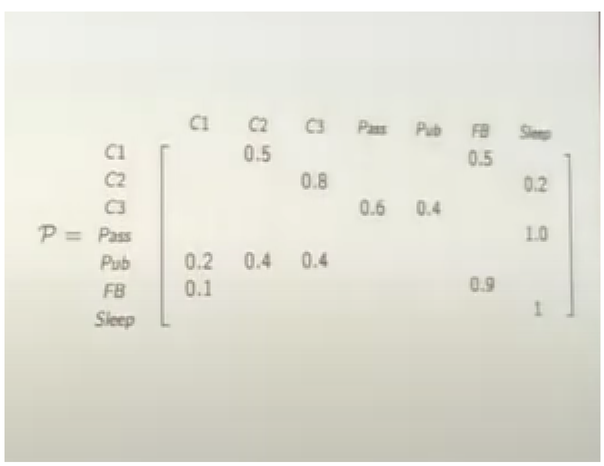


- Each row of the matrix sums to 1.
- A Markov process is a memoryless random process, i.e. a sequence of random states S1, S2, … with the Markov property.



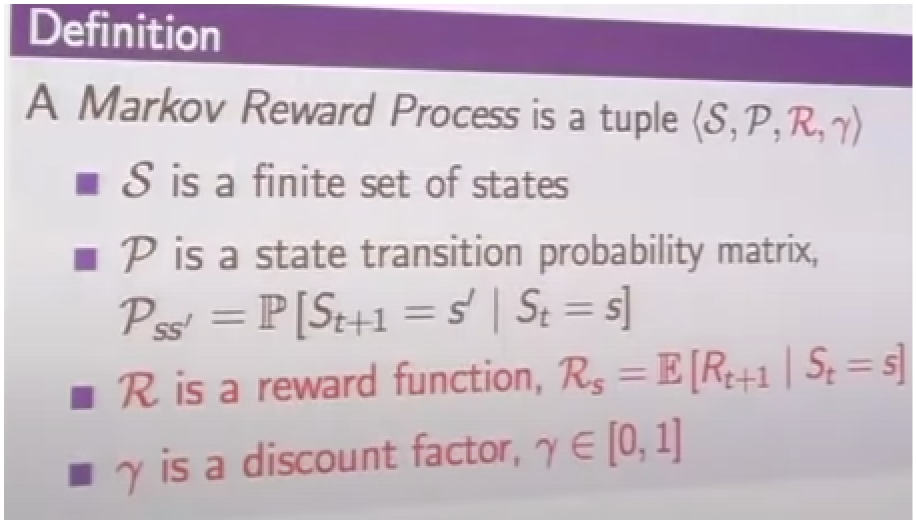
- Example:



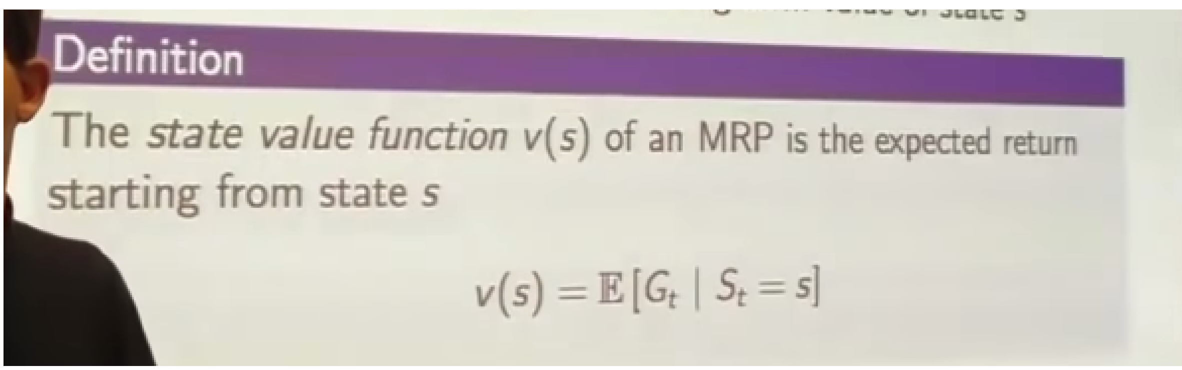


$$
\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\left[\begin{array}{ccccccc}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1.0 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{array}\right]
\end{array}
$$

- A Markov reward process is a Markov chain with values.



- The discount factor is brought about the balance the imperfect nature of the model. No model i.e. environment is perfect, it's not possible to account for all features and possibilities.
- The value function v(s) gives the long-term value of state s.



- Bellman Equation
  - Immediate Reward + Value Function(G) of the next state.

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

- Bellman equation can be expressed concisely using matrices.

The Bellman equation can be expressed concisely using matrices,

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

where v is a column vector with one entry per state

$$
\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}
$$

- Computational Complexity is O(n^3) for n states, not suitable for large MRPs.
- **Markov Reward Process (MRP)**
- Other iterative process for large MRPs,
    - Dynamic Programming
    - Monte-Carlo evaluation
    - Temporal-Difference Learning
- A policy π is a distribution over actions given states

$$\pi(a|s) = P[At \ = a|St \ = s]$$

- Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- The state sequence $S_1, S_2, ...$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- The state and reward sequence $S_1, R_2, S_2, ...$ is a Markov reward process $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
- where

$$\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'}$$

$$\mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s$$

- The optimal state-value function v*(s) is the maximum value function over all policies.

$$v * (s) = max v\pi(s)$$

- **The optimal action-value function q*(s,a) is the maximum action-value function over all policies.**

$$q * (s,\ a) = max\ q\pi(s,\ a)$$

- The optimal action-value function basically tells you the most optimal path that you can take after considering the probability of all actions the agent could take.

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

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

- There is always a deterministic optimal policy for any MDP
- If we know $q_*(s, a)$, we immediately have the optimal policy

> The only stupid question is the one you were afraid to ask but never did. -Rich Sutton