# Markov Decision Processes

by Gianluca Guglielmo

21 September 2021

# Markov Chain

## Definition

A *Markov Chain* is a stochastic model that describes a sequence of possible events $(X_t)_{t \geq 0}$ that satisfy the *Markov Property*:

$$P(X_n|X_{n-1}, \dots, X_0) = P(X_n|X_{n-1})$$

# Markov Decision Process

**Definition**

A *Markov Decision Process* is a tuple $(S, A, P_a, R_a, \gamma)$.

- $S$ is the state space.
- $A$ is the action space.
- $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability that action $a$ in state $s$ at time $t$ will lead to state $s'$ at time $t + 1$
- $R_a(s, s') = E[R_{t+1} | s_t = s, a_t = a]$ is the immediate reward received after transitioning from state $s$ to state $s'$, due to action a
- $\gamma$ is the discount factor

# Markov Decision Process

**Definition**

A policy $\pi$ is a distribution over actions given states:

$$\pi(a|s) = P(A_t = a|S_t = s)$$

**Definition**

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

$$G_t = \sum_{t=0}^{\infty} \gamma R_{a_t}(s_t, s_{t+1})$$

# Markov Decision Process

**Definition**

The *Value Function* is the expected return starting from state $S$ and following policy $\pi$: $v_\pi(s) = E[G_t|s_t = s]$

**Definition**

The *action-value function* $q_\pi(s,a)$ is the expected return starting from state $s$, taking action $a$, and then following policy $\pi$:
$q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]$

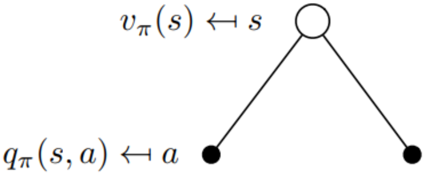
- Using the *Bellman equations*:

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v(S_{t+1})|S_t = s]$$
$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q(S_{t+1}, A_{t+1})|S_t = s]$$

## Intuition

- State $s$ leads to two possible actions following probability $\pi$, hence:

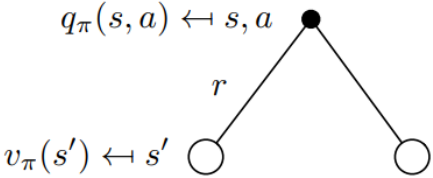
$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s, a)$$



## Intuition

- Action $a$, coming from state $s$, leads to two possible states following probability $P_a(s, s')$, hence:

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



# Optimization

- Combining the two equations:

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in A} P_a(s,s') \sum_{a \in A} \pi(a|s) q_\pi(s,a)$$

- In order to know the quality of an action in *any* given state we are interested in finding the optimal $q_*$:

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

# Partially Observable Markov Decision Process

- In a *Partially Observable Markov Decision Process* the agent can't directly observe the underlying state $s$. More formally:

**Definition**

A POMDP is a tuple $S, A, T, R, \Omega, O, \gamma$.

- $T(s'|s, a)$ is a set of conditional probabilities between states
- $\Omega$ is a set of observations
- $O(s'|s, a)$ is a set of conditional observation probabilities

# Partially Observable Markov Decision Process

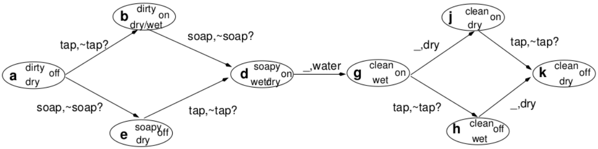
- The state isn't defined with certainty, hence data gathering is needed to update the *belief* $b(s)$ that the environment is in state $s$:

$$b(s) = \eta O(o|s,a) \sum_{s \in S} T(s'|s,a) b(s)$$

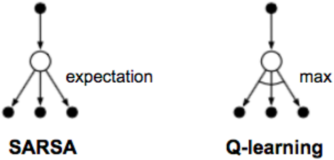
- The optimal policy is then chosen by maximizing the long-term reward:

$$\pi_* = \arg\max_{\pi} \sum_{t=0}^{\infty} \gamma^t E\left[ R(s_t, a_t) | b_0, \pi \right]$$

# Example


a
dirty
off
dry
b
dirty
on
dry/wet
e
soapy
off
dry
d
soapy
on
wet/dry
g
clean
on
wet
j
clean
on
dry
h
clean
off
wet
k
clean
off
dry
tap,~tap?
soap,~soap?
soap,~soap?
tap,~tap?
_,water
_,dry
tap,~tap?
tap,~tap?
_,dry

# SARSA Vs Q-learning



# SARSA Algorithm

**Definition**

*SARSA* is an ON-policy TD algorithm with parameters *step size* $\alpha \in (0, 1]$, *exploration rate* $\epsilon > 0$.

- Initialize $Q(s, a)$ for each state $s$ and action $a$
- For each episode:
  - Initialize $S$
  - Choose $A$ from $S$ using policy derived from $Q$ ($\epsilon$-greedy)
  - For each state of the episode:
    - Take action $A$, observe $(R, S')$
    - Choose $A'$ from $S'$ using policy derived from Q ($\epsilon$-greedy)
    - $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$
    - $S \leftarrow S'$, $A \leftarrow A'$
  - until $S$ is terminal

# Q-Learning Algorithm

**Definition**

*Q-Learning* is an OFF-policy TD algorithm with parameters *step size* $\alpha \in (0, 1]$, *exploration rate* $\epsilon > 0$.

- Initialize $Q(s, a)$ for each state $s$ and action $a$
- For each episode:
  - Initialize $S$
  - For each state of the episode:
    - Choose $A$ from $S$ using policy derived from $Q$ ($\epsilon$-greedy)
    - Take action $A$, observe $(R, S')$
    - $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
    - $S \leftarrow S'$
  - until $S$ is terminal