



Aprendizaje Reforzado

Maestría en Data Mining, Universidad Austral

Javier Kreiner



Plan de la clase

Teoría:

- Exploración vs Explotación
- Bandidos Multi-Brazo
- 10-armed Test Bed
- Métodos: ϵ -greedy, optimista, cota superior de confianza
- Cadenas de Markov
- Procesos de Decisión de Markov
- Procesos de Recompensa de Markov
- Función de Valor
- Función de Acción-Valor
- Ecuación de Esperanza de Bellman
- Ecuación de Optimalidad de Bellman
- Evaluación de una política

Exploración vs Explotación



- **Explotación:** Tomar la acción que es **más conveniente** *en el momento*.
- **Exploración:** Tomar **decisiones sub-óptimas** con el propósito de *obtener más información*.

Ejemplos:

¿Qué publicidad nuestro? ¿Dónde realizó pozos de petróleo? ¿Qué restorán elijo?

Bandidos Multi-brazo

Sólo hay acciones y recompensas.

$$A_t \rightarrow R_t$$



$$q_*(a) = E[R_t | A_t = a]$$

$a = 1, \dots, k$

Métodos basados en la función de valor

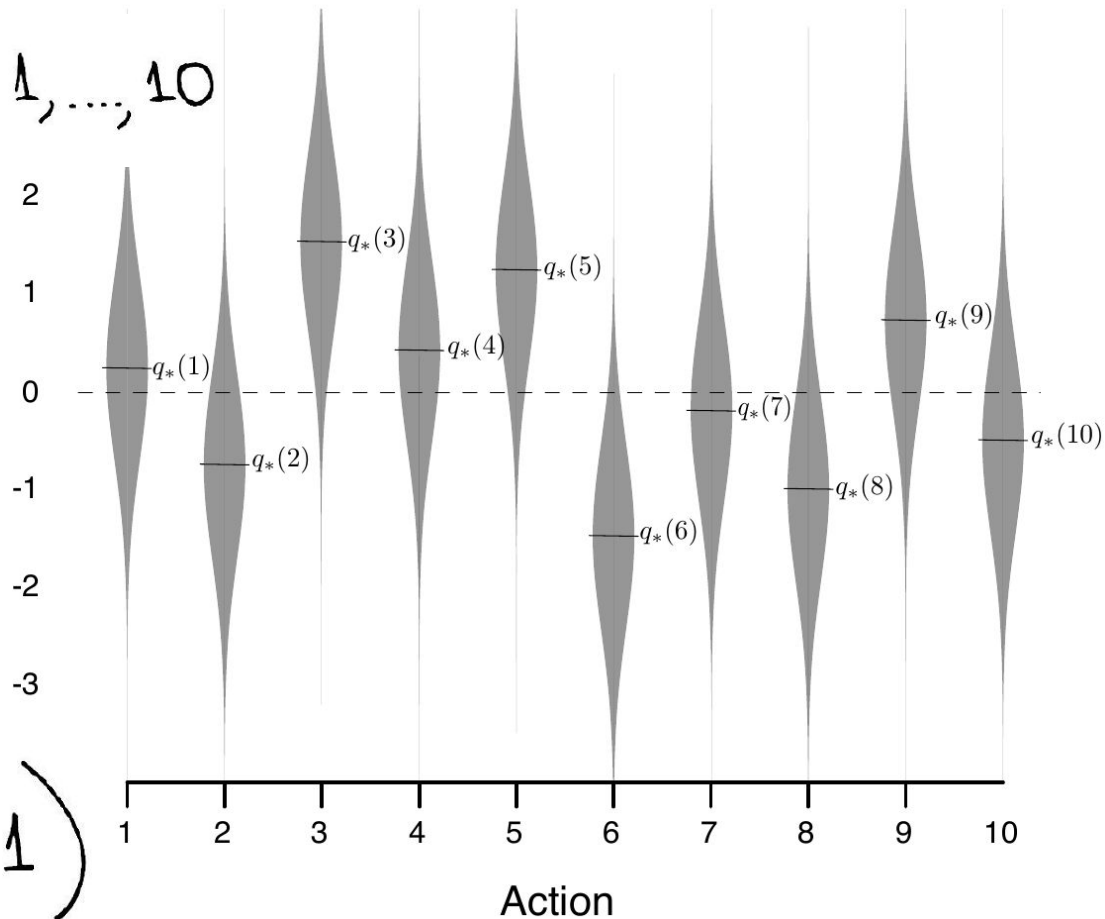
$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \mathbb{1}_{A_i=a}}{N_t(a)}$$

$$A_t = \underset{a}{\operatorname{argmax}} Q_t(a)$$

10-armed Testbed

$$q^*(a) \sim \mathcal{N}(0, 1), \quad a = 1, \dots, 10$$

Reward
distribution

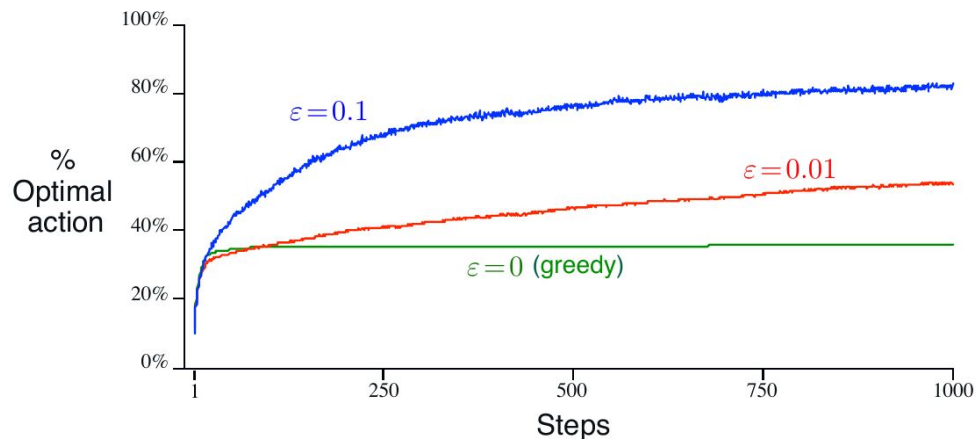
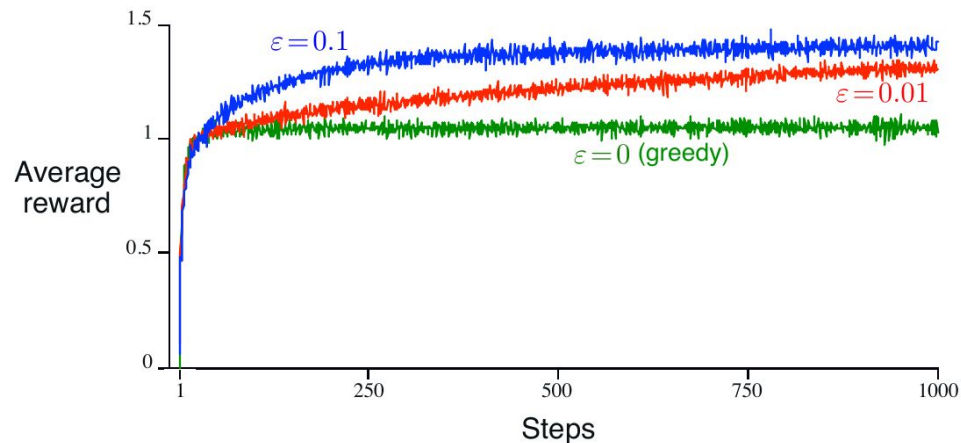


$$R_t | A_t = a \sim \mathcal{N}(q^*(a), 1)$$

10-armed Testbed - ϵ -greedy



- Dependiendo del ruido de la recompensa, se modifica el rendimiento del ϵ -greedy.
- Inclusive, en casos donde la recompensa es determinística (en función de la acción) puede convenir ϵ -greedy (caso no estacionario).



Algoritmo y caso no estacionario

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \operatorname{argmax}_a Q(a) & \text{with probability } 1 - \varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases} \quad (\text{breaking ties randomly})$$

$$R \leftarrow \text{bandit}(A)$$

$$N(A) \leftarrow N(A) + 1$$

$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$$

Caso no estacionario

$$Q_{n+1} \doteq Q_n + \alpha [R_n - Q_n] = (1 - \alpha)^n Q_1 + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i.$$

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty$$

Controla la fluctuaciones del comienzo

$$\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty.$$

Garantiza la convergencia

Método “Optimístico”

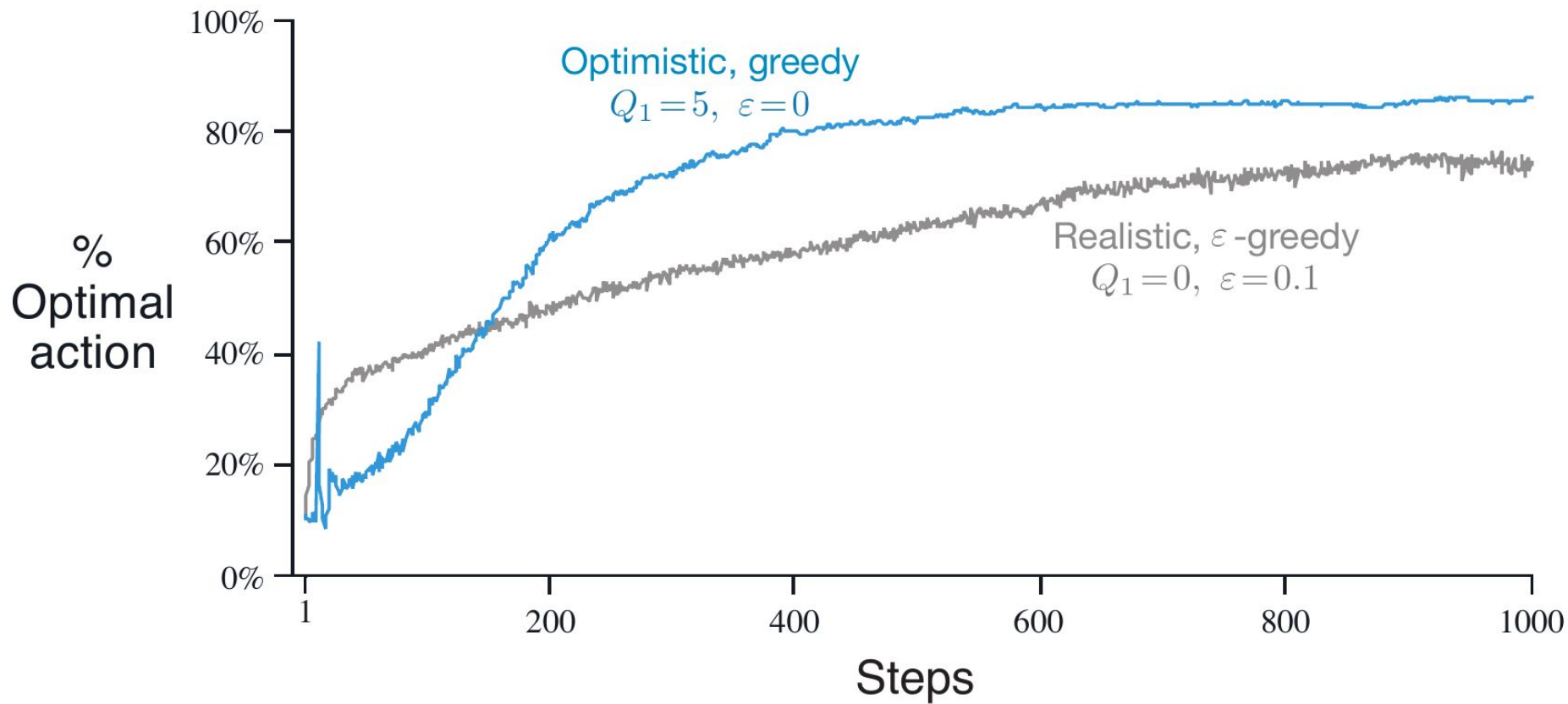
El método anterior es muy dependiente de las *condiciones iniciales*,

$$Q_1(a)$$

$$Q_2(a) = Q_1(a) + \frac{1}{N_1(a)} [R_1 - Q_1(a)]$$

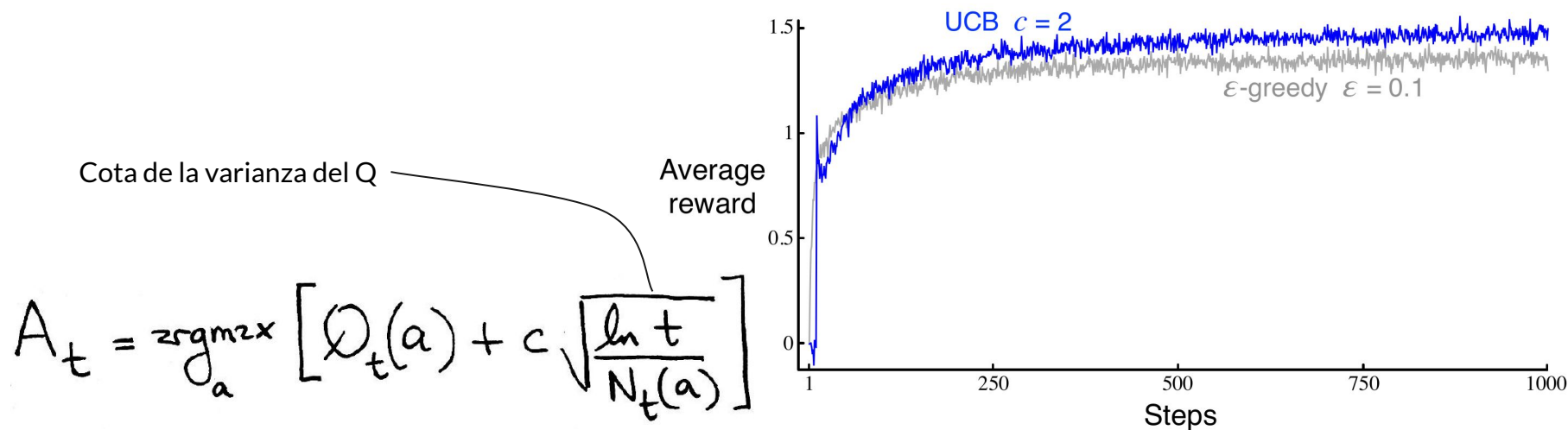
Si defino todas las condiciones iniciales de manera **optimista**, la recompensa no va a cumplir con las expectativas por lo que *muchas veces voy a cambiar de acción*.

Es decir, **voy a explorar!**



Selección de acción basada en Cota Superior de Confianza (UCB)

ϵ -greedy selecciona las acciones no óptimas *sin utilizar **nada** de información* sobre las mismas.





Programación



Propiedad de Markov

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

Proceso o Cadena de Markov



A Markov process is a memoryless random process, i.e. a sequence of random states S_1, S_2, \dots with the Markov property.

Definition

A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- \mathcal{S} is a (finite) set of states
- \mathcal{P} is a state transition probability matrix,
$$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$$

Proceso de Recompensa de Markov

A Markov reward process is a Markov chain with values.

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

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

Proceso de recompensa de Markov


Función de recompensa

$$\mathcal{R}_s := E[R_{t+1} | S_t = s]$$

Retorno

$$G_t := R_{t+1} + \gamma R_{t+2} + \dots$$

Función de Valor

$$v(s) = E[G_t | S_t = s]$$

Ecuación de Bellman

Podemos descomponer la función de valor en dos partes:

- La recompensa inmediata
- La función de valor en el próximo paso descontada

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

Ecuación de Bellman



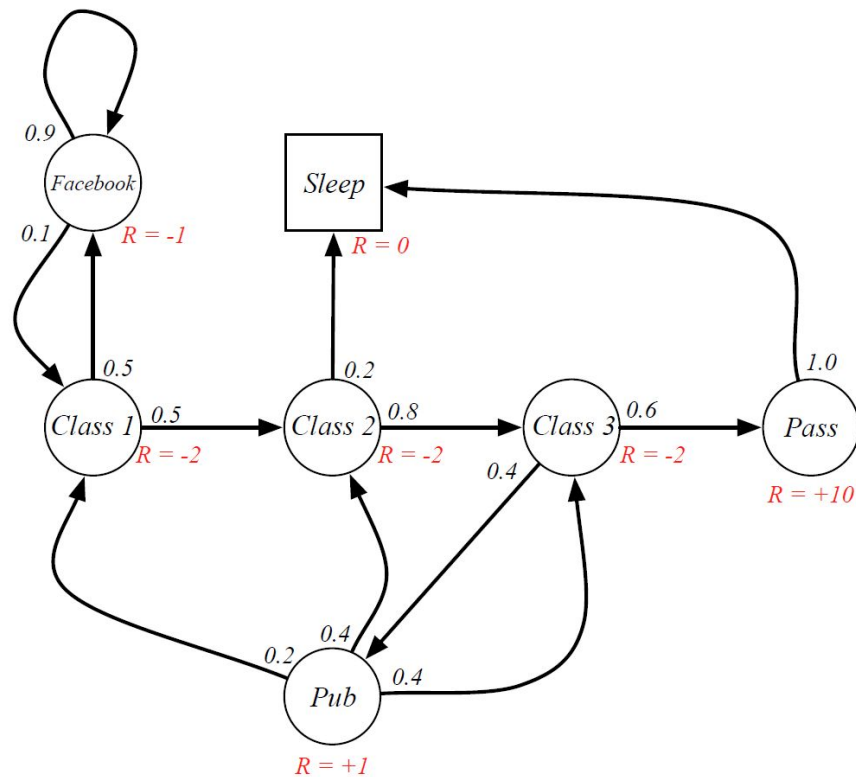
$$v(s) = \mathcal{R}_s + \gamma \sum_{s'} p_{s,s'} v(s')$$

$$v = (v(s_1), \dots, v(s_K))$$

Un sistema de K ecuaciones!

- Con matrices:
$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$
- Escrito en una línea: $v = \mathcal{R} + \gamma p v$

Un ejemplo



Proceso de Decisión de Markov



Se agrega un *acción* la cual modifica el ambiente.

$$p_{s,s'}^a := P(S_{t+1} = s' | S_t = s, A_t = a)$$

$$\mathcal{R}_s^a := E[R_{t+1} | S_t = s, A_t = a]$$

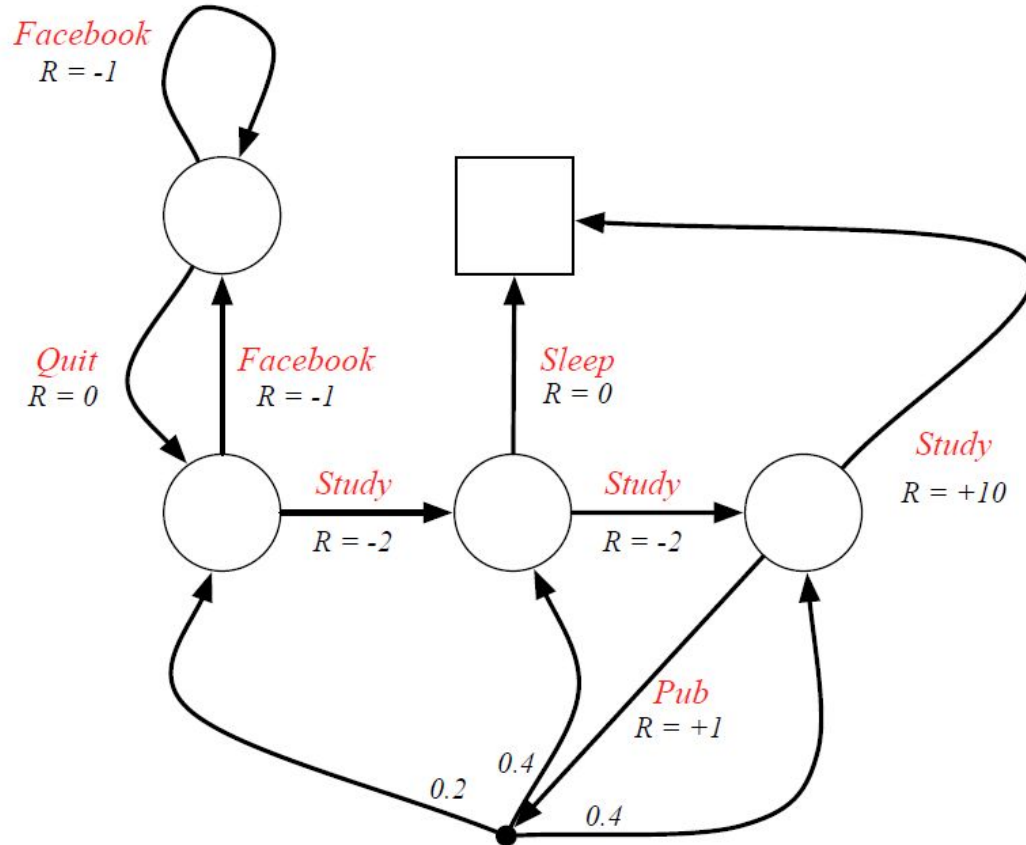
Proceso de Decisión de Markov

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

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

Ejemplo



Política

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

- Es una distribución de probabilidad sobre las acciones dado un estado
- Define el comportamiento del agente
- Dado un proceso de decisión de Markov y una política fija, la secuencia

S_1, R_2, S_2, \dots es un proceso de recompensa de Markov $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$

Donde:

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



Funciones de Valor:

- Función de valor de estado:

$$v_{\pi}(s) = \mathbb{E}_{\pi} [G_t \mid S_t = s]$$

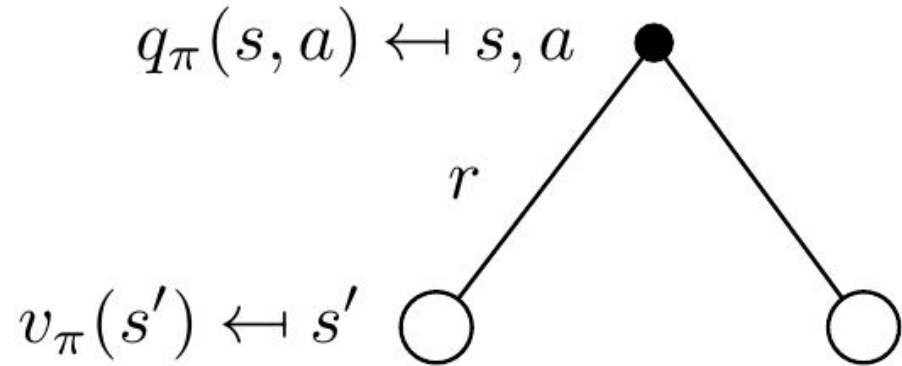
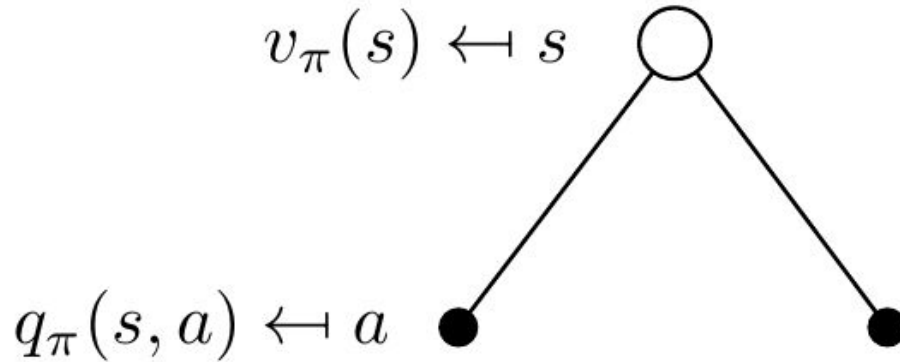
$$v_{\pi}(s) = \mathbb{E}_{\pi} [R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s]$$

- Función de valor de estado-acción:

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} [G_t \mid S_t = s, A_t = a]$$

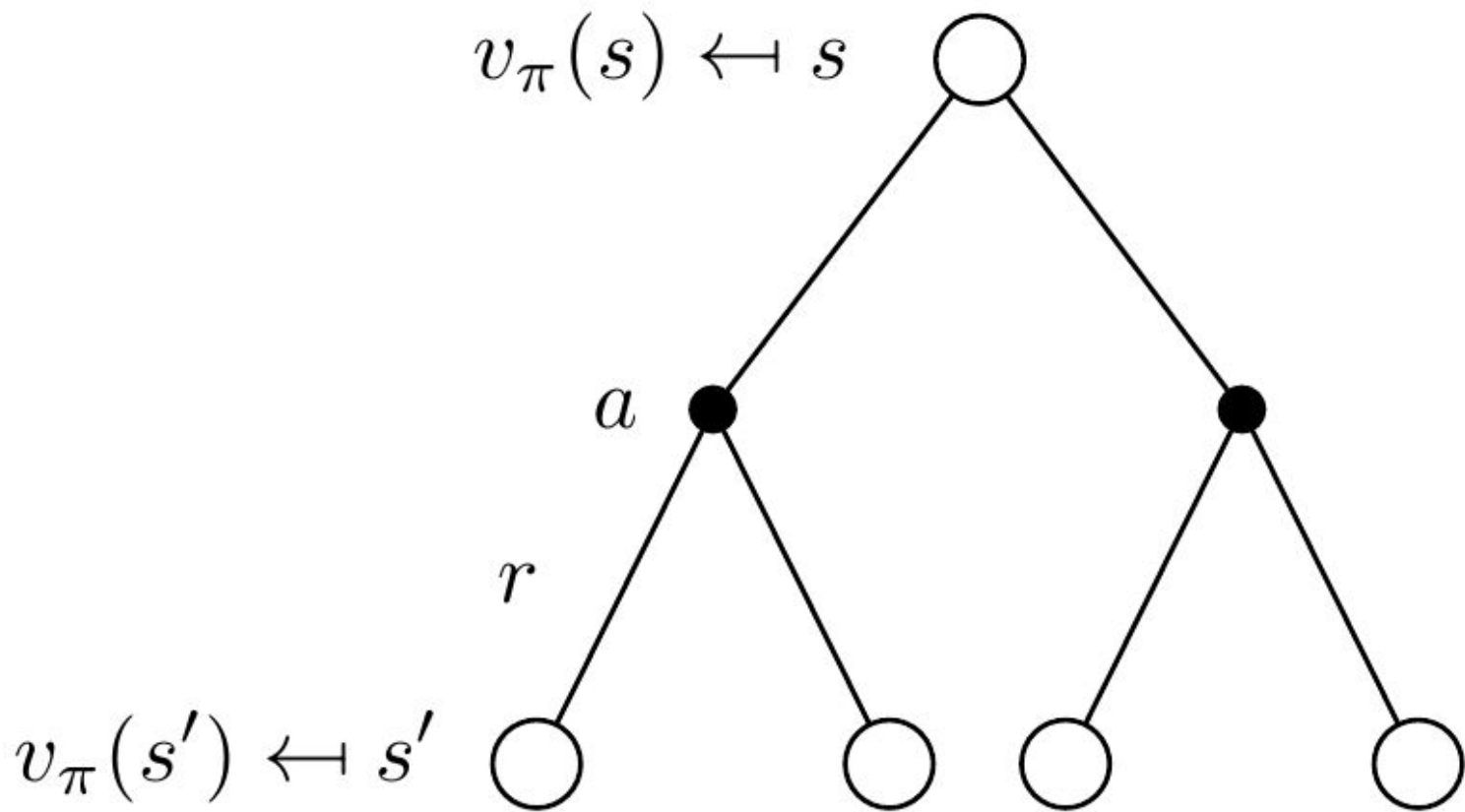
$$q_{\pi}(s, a) = \mathbb{E}_{\pi} [R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

Ecuación de Bellman (bis)




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

$$q_{\pi}(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi}(s')$$



Evaluación de Política


$$\begin{aligned} v_{\pi}(s) &= \sum_a [\mathcal{R}_s^a + \gamma \sum_{s'} v_{\pi}(s') p_{s,s'}^a] \pi(a|s) \\ &= \mathcal{R}_s^{\pi} + \gamma \sum_{s'} v_{\pi}(s') p_{s,s'}^{\pi} \end{aligned}$$

Método Iterativo

$$v_{\pi}^{k+1}(s) = \mathcal{R}_s^{\pi} + \gamma \sum_{s'} v_{\pi}^k(s') p_{s,s'}^{\pi}$$

Función de Valor Óptima



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

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

Optimalidad de MDP

$\exists \pi_* / \pi_* \geq \pi \forall \pi$ such that

$$v_*(s) = v_{\pi_*}(s), \quad q_*(s, a) = q_{\pi_*}(s, a) \quad \forall s, a$$

$$\pi_*(s) = \operatorname{argmax}_a q_*(s, a)$$

$$v_*(s) = \max_a q_*(s, a)$$

Ecuación de optimalidad de Bellman



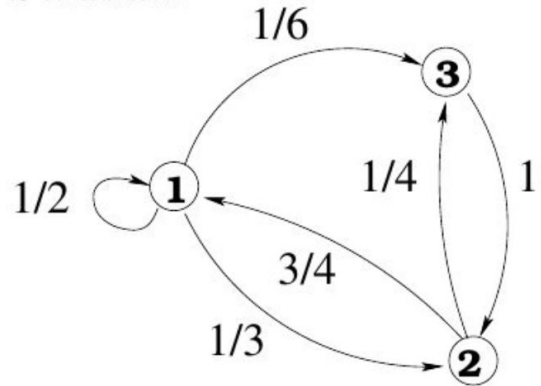
$$v_*(s) = \max_a [\mathcal{R}_s^a + \gamma \sum_{s'} p_{s,s'}^a v_*(s')]$$

Son ecuaciones NO lineales!

¿Cómo obtener v_* y π_* ?

Ejercicios para entregar

1. Simular el proceso de recompensa del slide 20 y obtener el retorno esperado con $\gamma = \frac{1}{2}$
2. ~~Calcular el retorno esperado~~



$$R(1) = -2, \quad R(2) = 3, \quad R(3) = 5$$

$$G(S_1, S_2) = R(S_1) + R(S_2)$$

Calcular $E_{e1}[G(S_1, S_2)]$ de manera analítica y por monte carlo.



Lectura recomendada:

- Introducción a Contextual Bandits: <https://towardsdatascience.com/contextual-bandits-and-reinforcement-learning-6bdfeaece72a>
- Capítulos 2 y 3 de Sutton and Barto
- Desafíos de Reinforcement Learning: <https://www.alexirpan.com/2018/02/14/rl-hard.html>