Ayudantía 12

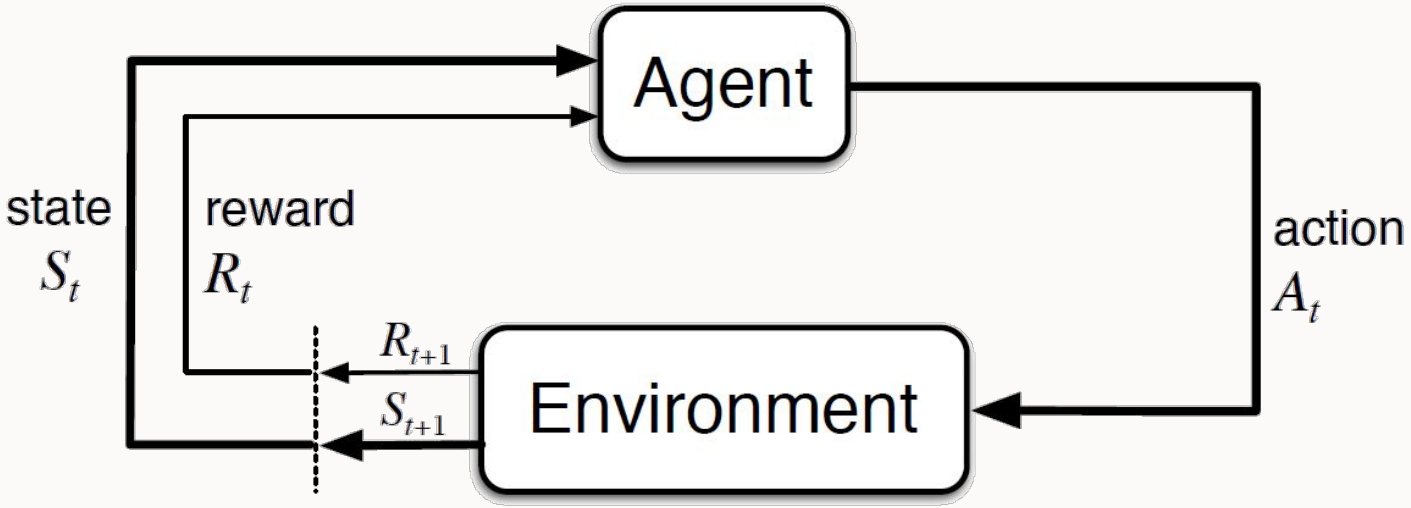
Reinforcement Learning

Martín Castillo & Felipe Villagrán

15 de noviembre 2024

¿Qué es?

Aprender a elegir acciones en un entorno tal que se maximice la **recompensa esperada**



¿Y el entorno?

Markov Decision Process (MDP)

$$\mathcal{M} = \langle S, A, p, \gamma \rangle$$

S → Conjunto de estados

***A*(s)** → Conjunto de acciones

***p*(s', r | s, a)** → Función de dinámica del entorno

γ → Factor de descuento

Markov Decision Process (MDP)

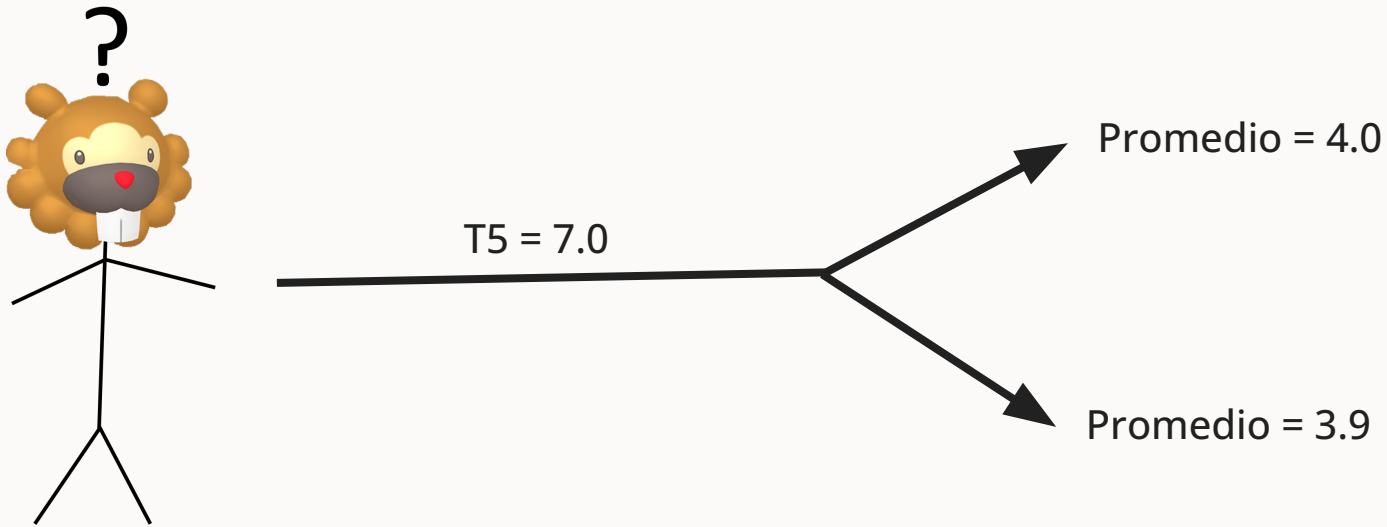
Para aplicar reinforcement learning, realizamos un supuesto:

Propiedad Markoviana → **s'** y **r** dependen únicamente de **s** y **a**

Markov Decision Process (MDP)

Para aplicar reinforcement learning, realizamos un supuesto:

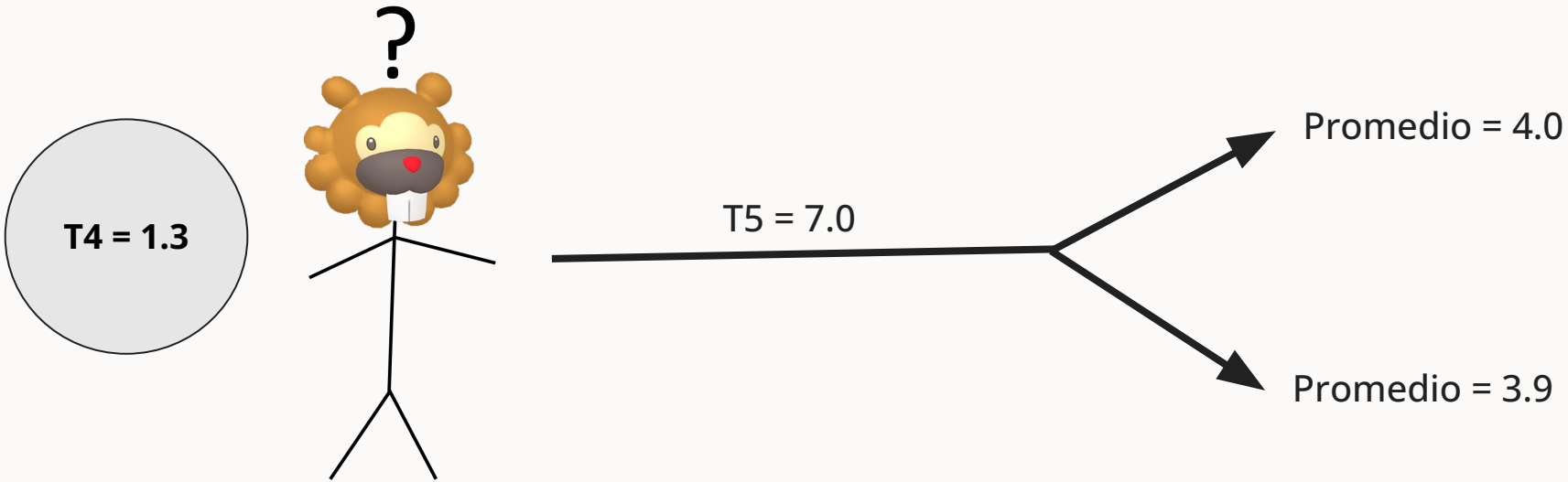
Propiedad Markoviana → **s'** y **r** dependen únicamente de **s** y **a**



Markov Decision Process (MDP)

Para aplicar reinforcement learning, realizamos un supuesto:

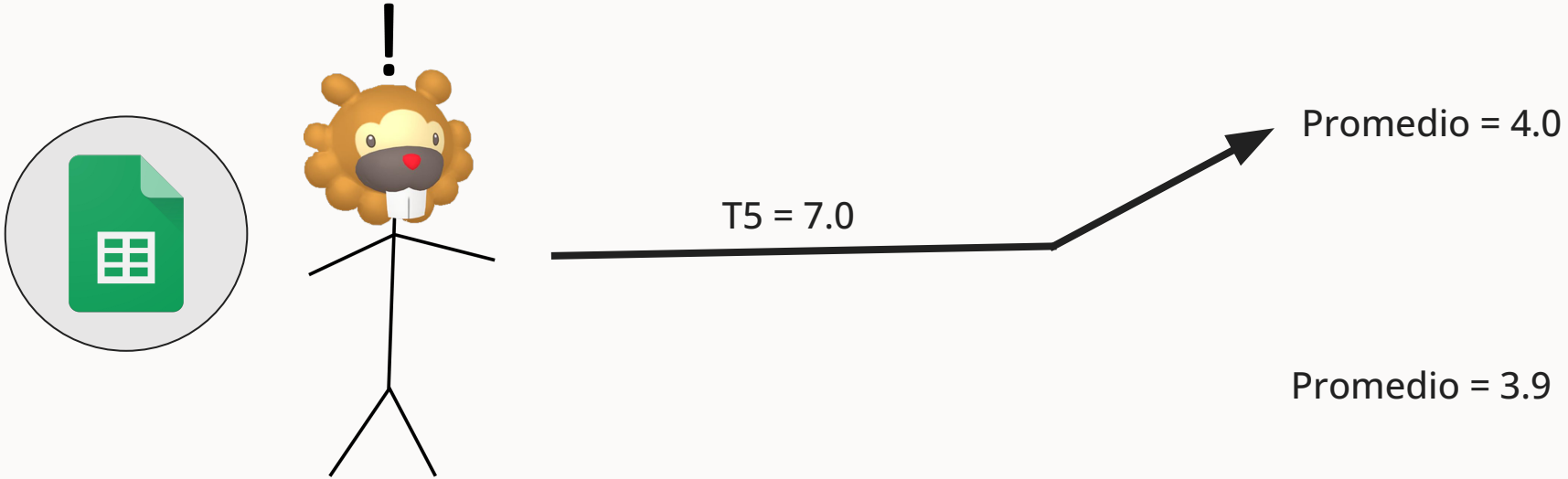
Propiedad Markoviana → **s'** y **r** dependen únicamente de **s** y **a**



Markov Decision Process (MDP)

Para aplicar reinforcement learning, realizamos un supuesto:

Propiedad Markoviana → **s'** y **r** dependen únicamente de **s** y **a**



Dinámica del entorno

Un MDP es gobernado por una función $p(s', r \mid s, a)$

Si estoy en **s** y tomo la acción **a** ¿Cuál es la probabilidad de llegar a **s'** con recompensa **r**?

Política de acción (π)

El objetivo del agente es encontrar una política de acción óptima

$$\pi(s_t|a_t)$$

Es decir, una que maximice la recompensa esperada

¿Cuánto vale un estado?

Función de Valor

El valor de un estado es la recompensa que esperamos obtener desde ese estado

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

Función de Valor

El valor de un estado es la recompensa que esperamos obtener desde ese estado

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

¡Depende de la política!

Función de Valor

De forma análoga, el valor de una acción es la recompensa que esperamos obtener desde que tomamos esa acción

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

¿Y cómo calculamos el valor?

La Ecuación de Bellman

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big]$$

La ecuación de Bellman nos permite estimar el valor de un estado bajo una política de acción π.

En otras palabras, buscamos la esperanza matemática de la ganancia total que obtendremos partiendo desde ese estado hasta el final de la simulación.

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

Si tenemos un problema de decisión de Markov bien definido (MDP), podemos expresar esta ecuación en términos de la función de probabilidad p que define un MDP.

El valor de un estado es la suma del valor de las acciones que es posible tomar en ese estado ponderadas por la probabilidad de tomar dicha acción.

$$v_\pi(s) = \underbrace{\sum_a}_{\text{Suma sobre las acciones}} \underbrace{\pi(a \mid s)}_{\text{Probabilidad de elegir la acción}} \underbrace{\sum_{s', r} p(s', r \mid s, a)\,[r + \gamma v_\pi(s')]}_{\text{Valor de la acción en ese estado}}$$

El valor de una acción dado un estado está incluido en la ecuación de Bellman.

Si al elegir una acción a en un estado s obtenemos como resultado una recompensa r y un estado s', entonces el valor esperado de la recompensa que obtendremos a partir de ese punto es $[r + \gamma v_\pi(s')]$, que corresponde a la recompensa obtenida mas el valor esperado descontado de lo que obtendremos a partir del nuevo estado.

En palabras más simples, si vemos los resultados de una acción a en un estado s como una tupla (s', r), entonces el valor de esa tupla es $[r + \gamma v_\pi(s')]$.

El valor de una acción en un estado es la suma del valor de los posibles resultados de esa acción ponderados por la probabilidad de que ocurran dichos resultados.

$$q_\pi(a \mid s) = \underbrace{\sum_{s', r}}_{\text{Suma sobre las tuplas resultado de } (s,a)} \underbrace{p(s', r \mid s, a)}_{\text{Probabilidad de que } (s,a) \text{ lleve a un resultado } (s',r)} \underbrace{[r + \gamma v_\pi(s')]}_{\text{Valor del resultado}}$$

Valor de un Estado - Policy Evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Queremos mejorar este valor

Mejorando la Política - Value Iteration

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$

Mejorando la Política - Value Iteration

Cuando $V_\pi(s)$ ya no mejora, es que encontramos su valor máximo $V_*(s)$

…y este valor es el que se tiene bajo una política óptima

Pero no siempre conocemos

$$p(s', r \mid s, a)$$

¿Qué hacemos?

Algoritmo Q-Learning

En lugar de estados usamos pares acción-estado

$v_\pi(s)$ → $Q(s_t, a_t)$

Algoritmo Q-Learning

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Algoritmo Q-Learning

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

A diferencia de value iteration se actualizan los valores de ejecutar una acción siguiendo una política a partir de un estado (Q-value) en lugar de el valor de un estado.

Hiperparámetros

- Learning rate: Velocidad de aprendizaje
- Exploration rate: Proporción de acciones con sample
- Discount rate: Ponderación de recompensa inmediata y aprendida

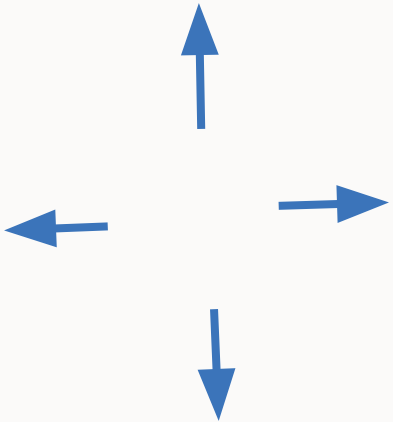
Pseudo Código

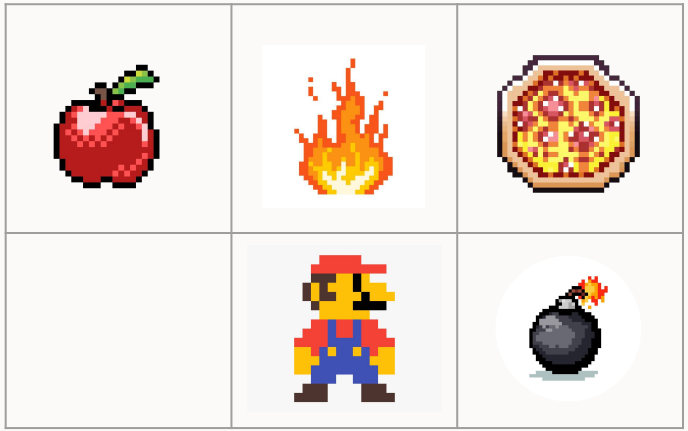
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

Ejemplo





Ejemplo

- Rewards

+1	-1	+10
0	0	-10

- Indices

0	1	2
3	4	5

Ejemplo

	0	1	2	3	4	5
up						
down						
left						
right						

Algoritmo Q-learning

Definición

Para aquellos que les guste el tema

31470	IIC3675	SI	NO	1	NO		Presencial		Aprendizaje Reforzado	Toro Rodrigo	San Joaquín	10	40	18