Inteligencia Artificial

Aprendizaje por refuerzo

Edgar Andrade, Ph.D.

Matemáticas Aplicadas y Ciencias de la computación

Última revisión: Noviembre de 2023

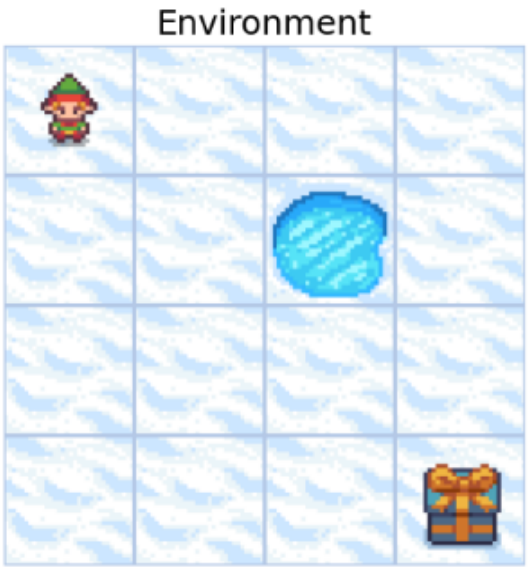
MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Contenido

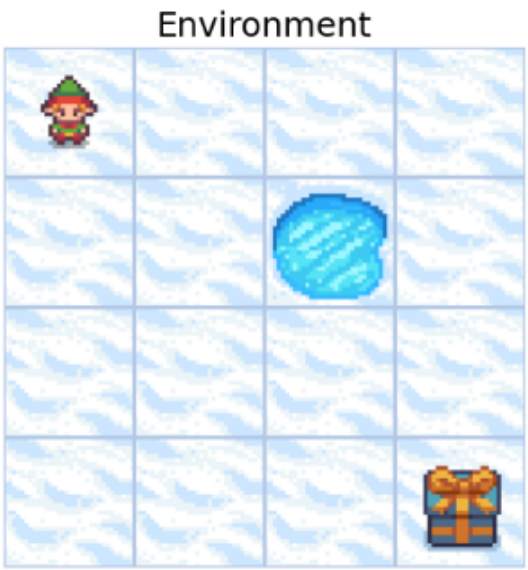
Resolviendo un entorno

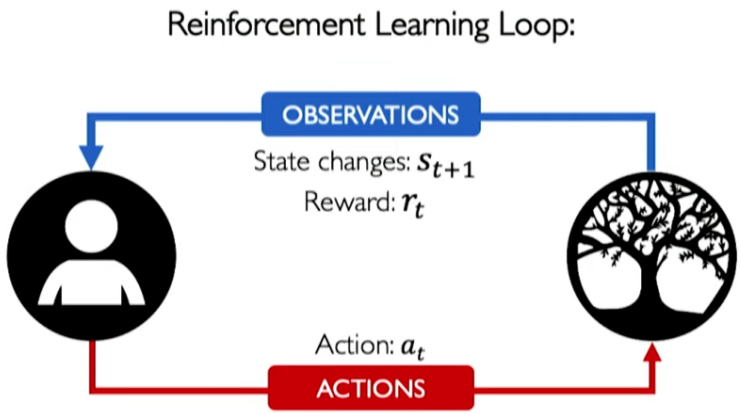
Problema: ¿Cómo podemos resolver el entorno si no conocemos el modelo subyacente?



Resolviendo un entorno

Problema: ¿Cómo podemos resolver el entorno si no conocemos el modelo subyacente?





☞ El agente debe aprender a actuar con base en su experiencia sobre un periodo prolongado de tiempo.

Componentes del aprendizaje por refuerzo

- Regla de aprendizaje
- Explorar vs. Aprovechar
- Recompensa a largo plazo

Componentes del aprendizaje por refuerzo

- Regla de aprendizaje
- Explorar vs. Aprovechar
- Recompensa a largo plazo

Aprendizaje como corrección del error del estimador de utilidad con base en la experiencia.

Componentes del aprendizaje por refuerzo

- Regla de aprendizaje
- Explorar vs. Aprovechar
- Recompensa a largo plazo

Balance entre aprovechar la información actual y explorar para conseguir mejor información.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Componentes del aprendizaje por refuerzo

- Regla de aprendizaje
- Explorar vs. Aprovechar
- Recompensa a largo plazo

El agente debe aprender a maximizar la recompensa total del episodio, no solo la recompensa por la acción actual.

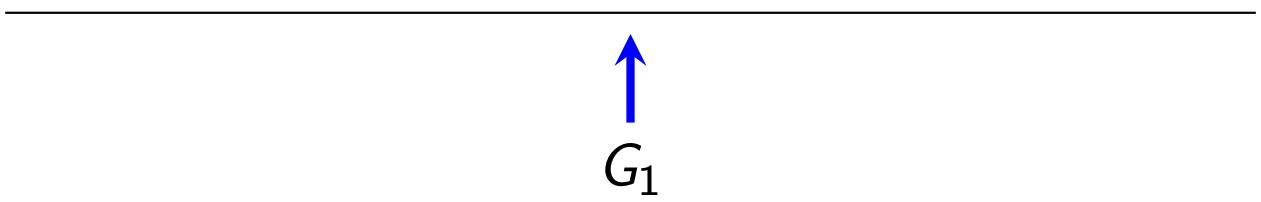
Contenido

Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.

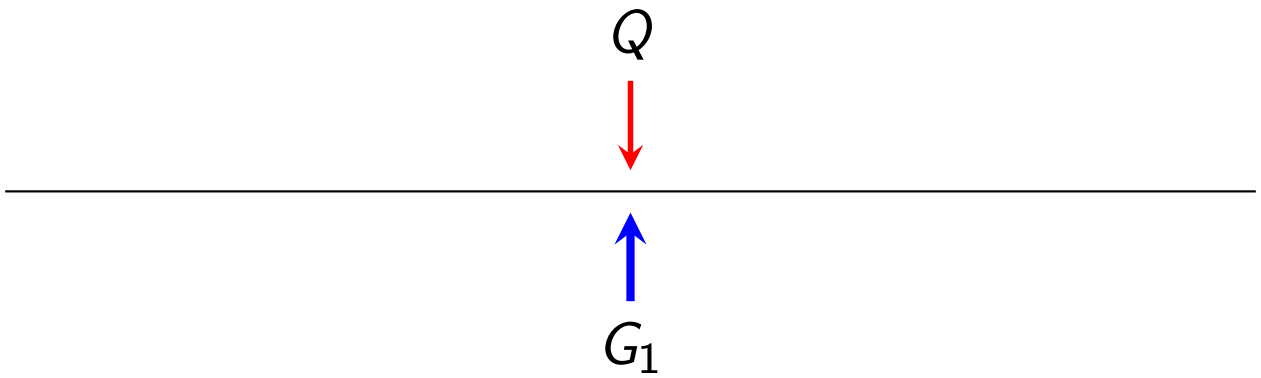
Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



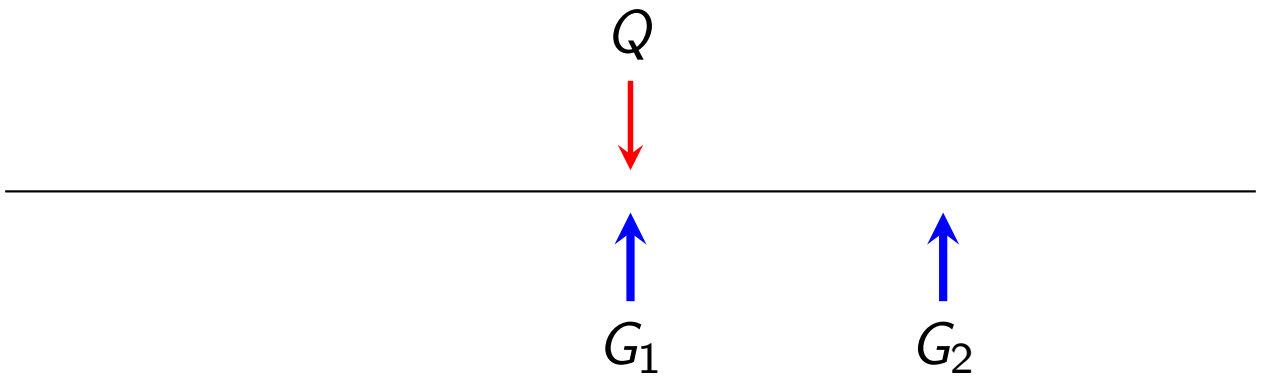
Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



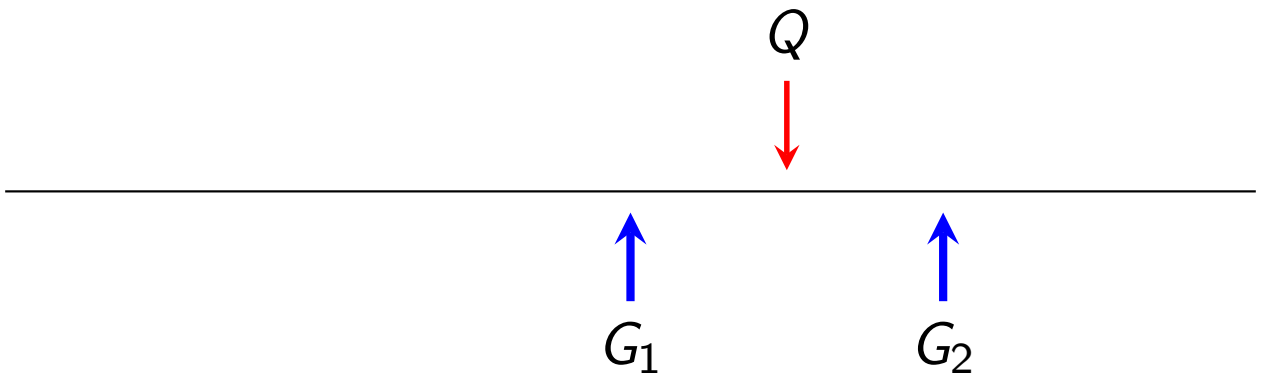
Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



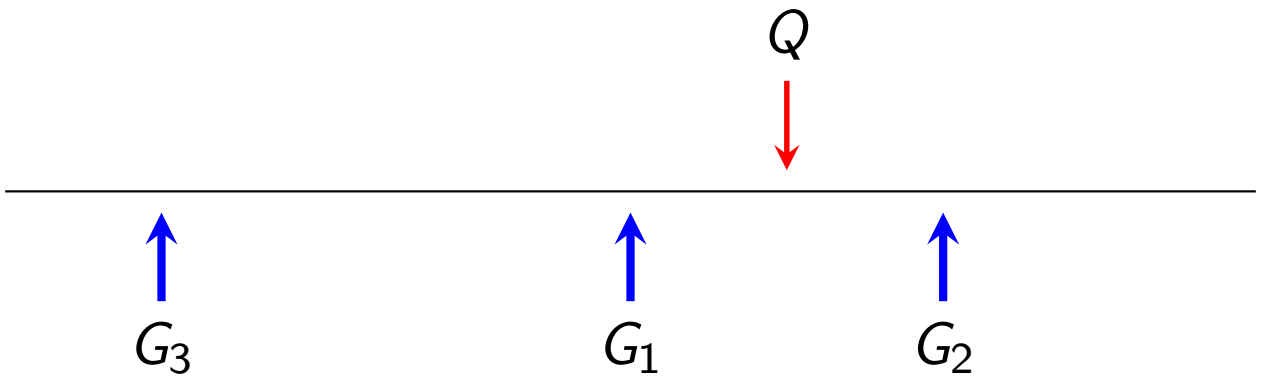
Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



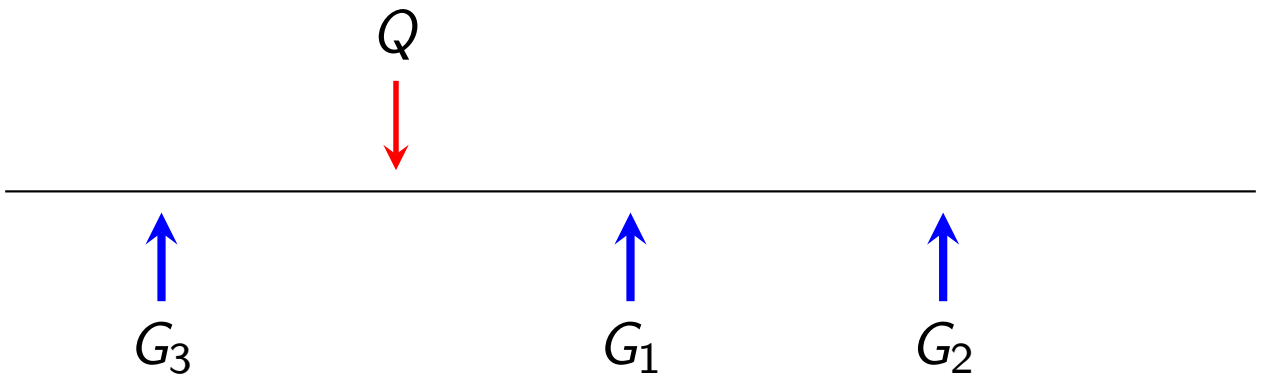
Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



Estimación

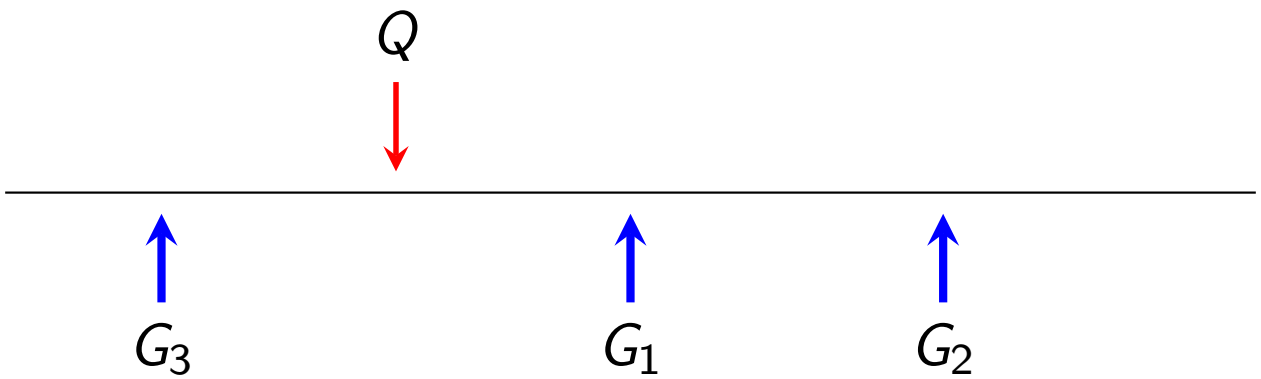
Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.



MACC
Matemáticas Aplicadas y Ciencias de la Computación

Estimación

Cada ronda queremos estimar un número natural q a partir de observaciones imperfectas $G_1, \ldots, G_T$, en donde G_i es la observación obtenida en la ronda i-ésima. Supongamos que Q es nuestra estimación actual.

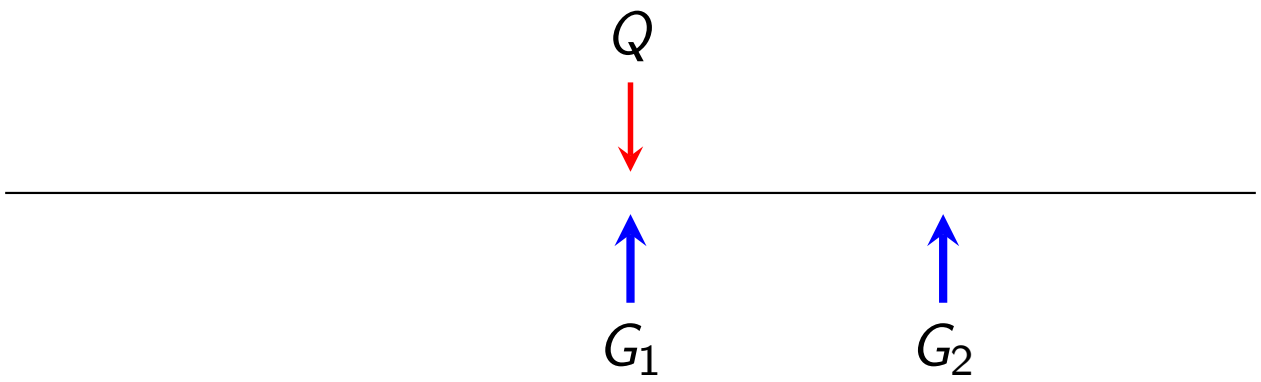


¿Qué fórmula nos sirve para medir la dirección y magnitud del cambio en la estimación?

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Corrección del error de estimación

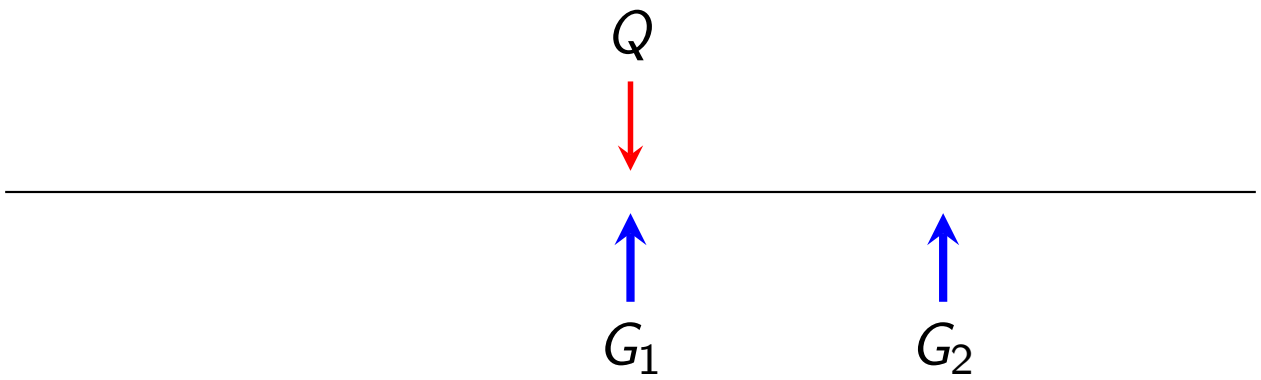
Cada ronda observamos un G_i y vemos la variación respecto a nuestra estimación actual q.



El error en la estimación es $\delta = G_2 - Q$.

Corrección del error de estimación

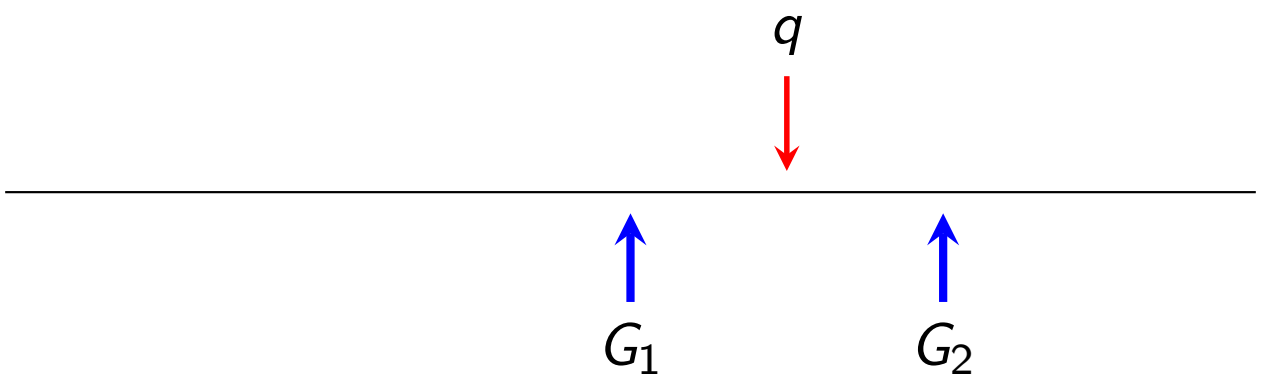
Cada ronda observamos un G_i y vemos la variación respecto a nuestra estimación actual q.



G_2 no es el dato real q, por lo que debemos ponderar el error por una tasa de aprendizaje α.

Corrección del error de estimación

Cada ronda observamos un G_i y vemos la variación respecto a nuestra estimación actual q.

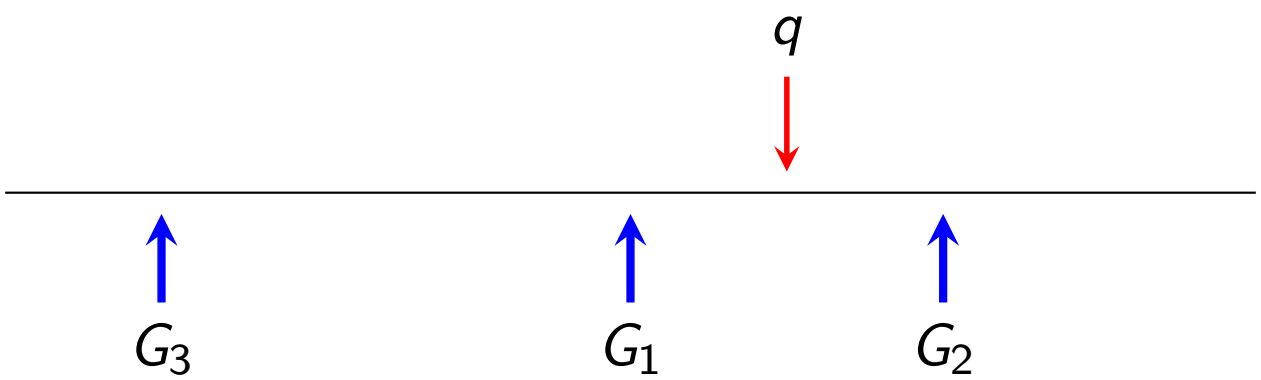


Actualizamos Q mediante la regla

$$Q \leftarrow Q + \alpha\delta$$

Corrección del error de estimación

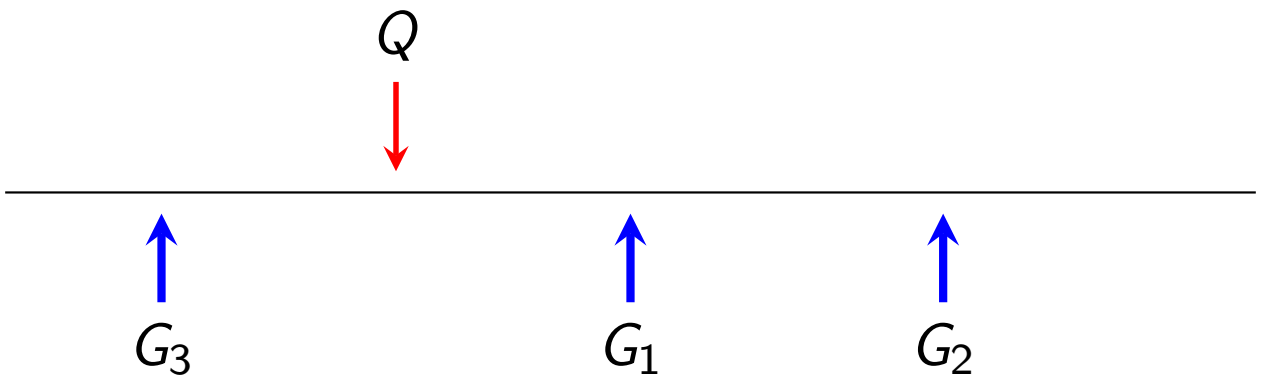
Cada ronda observamos un G_i y vemos la variación respecto a nuestra estimación actual q.



El error en la estimación es $\delta = G_3 - Q$.

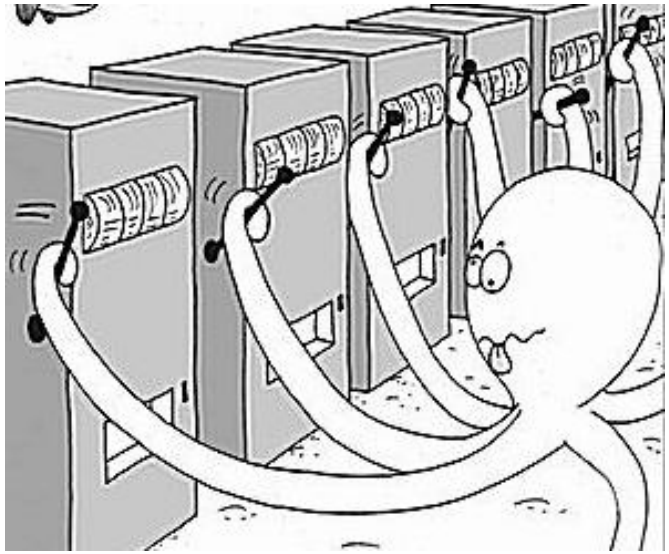
Corrección del error de estimación

Cada ronda observamos un G_i y vemos la variación respecto a nuestra estimación actual q.



$$Q \leftarrow Q + \alpha\big(G_3 - Q\big)$$

Multi-armed bandits

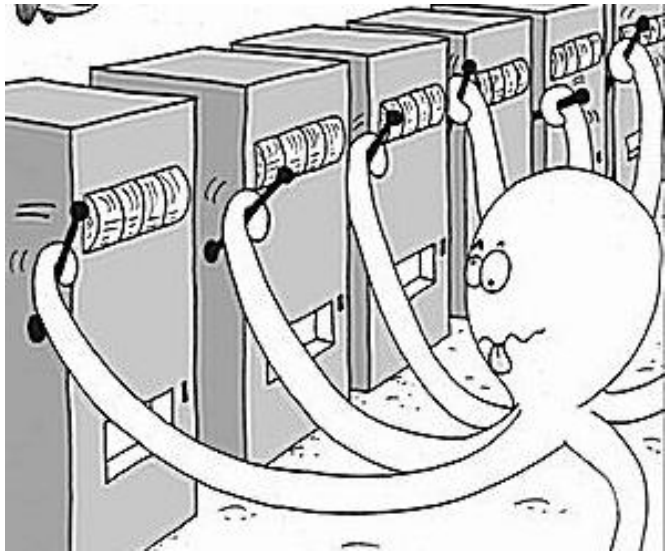


Problema de agendamiento estocástico

- El agente jala la palanca de alguna de las máquinas y obtiene una recompensa 0 o 1.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Multi-armed bandits



Problema de agendamiento estocástico

- La probabilidad de éxito en cada máquina es distinta e inicialmente desconocida por el agente.

Multi-armed bandit problem

$r(1)$ $r(2)$

Problema: ¿Cuál máquina a otorga éxito con mayor probabilidad $Q(a)$?

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Plan de solución

☞ Estimar la probabilidad de éxito $q(a)$ de cada máquina a.

Plan de solución

☞ Estimar la probabilidad de éxito $q(a)$ de cada máquina a.

☞ Cada ronda i seleccionamos una máquina a y observamos la recompensa r_i.

Plan de solución

☞ Estimar la probabilidad de éxito $q(a)$ de cada máquina a.

☞ Cada ronda i seleccionamos una máquina a y observamos la recompensa r_i.

☞ Mantenemos los estimadores $Q_i(a)$ y actualizamos mediante la regla:

$$Q_i(a) = \begin{cases} Q_{i-1}(a) + \alpha\big(r_i - Q_{i-1}(a)\big), & \text{si } a \text{ es seleccionada} \\ Q_{i-1}(a), & \text{en otro caso} \end{cases}$$

Ejemplo

Supongamos $\alpha = \frac{1}{2}$.

Máquina 1

- *Estado inicial:* $Q_0(1) = 0$

Máquina 2

- *Estado inicial:* $Q_0(2) = 0$

Ambos estimadores comienzan en 0

Ejemplo

Supongamos $\alpha = \frac{1}{2}$.

Máquina 1

- *Estado inicial:* $Q_0(1) = 0$
- *Turno 1:* $Q_1(1) = 0 + \alpha(1 - 0) = \frac{1}{2}$

Máquina 2

- *Estado inicial:* $Q_0(2) = 0$
- *Turno 1:* $Q_1(2) = 0$

Seleccionamos 1 y obtenemos éxito ($r_1 = 1$)

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo

Supongamos $\alpha = \frac{1}{2}$.

Máquina 1

- *Estado inicial:* $Q_0(1) = 0$
- *Turno 1:* $Q_1(1) = 0 + \alpha(1 - 0) = \frac{1}{2}$
- *Turno 2:* $Q_2(1) = \frac{1}{2} + \alpha(0 - \frac{1}{2}) = \frac{1}{4}$

Máquina 2

- *Estado inicial:* $Q_0(2) = 0$
- *Turno 1:* $Q_1(2) = 0$
- *Turno 2:* $Q_2(2) = 0$

Seleccionamos 1 y no obtenemos éxito ($r_2 = 0$)

Ejemplo

Supongamos $\alpha = \frac{1}{2}$.

Máquina 1

- *Estado inicial:* $Q_0(1) = 0$
- *Turno 1:* $Q_1(1) = 0 + \alpha(1 - 0) = \frac{1}{2}$
- *Turno 2:* $Q_2(1) = \frac{1}{2} + \alpha(0 - \frac{1}{2}) = \frac{1}{4}$
- *Turno 3:* $Q_3(1) = \frac{1}{4}$

Máquina 2

- *Estado inicial:* $Q_0(2) = 0$
- *Turno 1:* $Q_1(2) = 0$
- *Turno 2:* $Q_2(2) = 0$
- *Turno 3:* $Q_3(2) = 0 + \alpha(1 - 0) = \frac{1}{2}$

Seleccionamos 2 y obtenemos éxito ($r_3 = 1$)

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Explorar vs. Aprovechar (1/3)

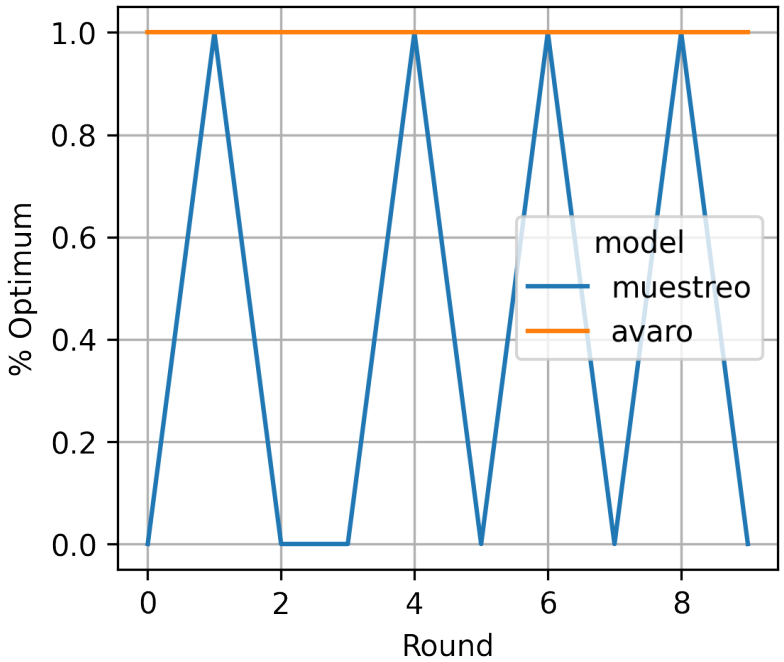
Dos enfoques extremos:

- **Explorar**: Muestrear ambos brazos.
- **Aprovechar**: Seleccionar el brazo que haya dado mejores beneficios hasta ahora (estrategia avara).

Explorar vs. Aprovechar (1/3)

Dos enfoques extremos:

- **Explorar**: Muestrear ambos brazos.
- **Aprovechar**: Seleccionar el brazo que haya dado mejores beneficios hasta ahora (estrategia avara).



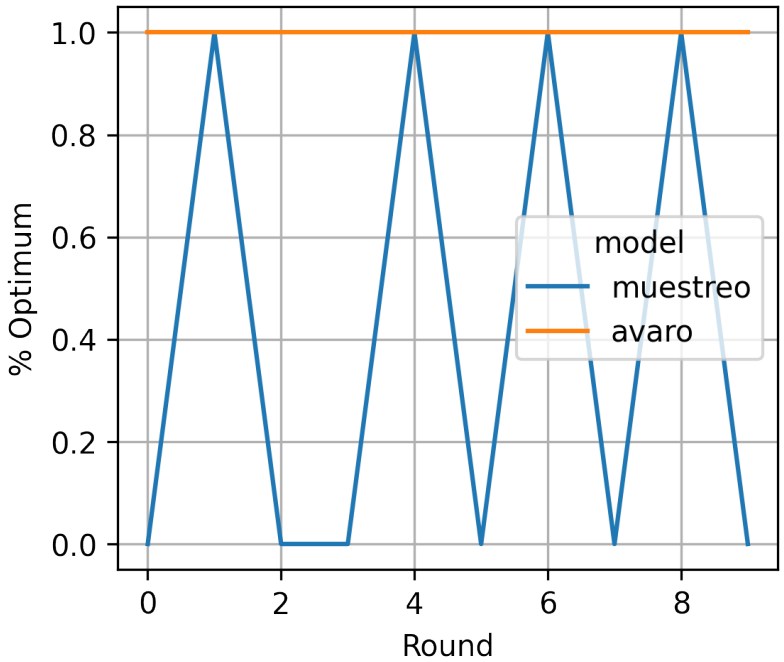
- La estrategia de muestreo exhibe un comportamiento aleatorio.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Explorar vs. Aprovechar (1/3)

Dos enfoques extremos:

- **Explorar**: Muestrear ambos brazos.
- **Aprovechar**: Seleccionar el brazo que haya dado mejores beneficios hasta ahora (estrategia avara).



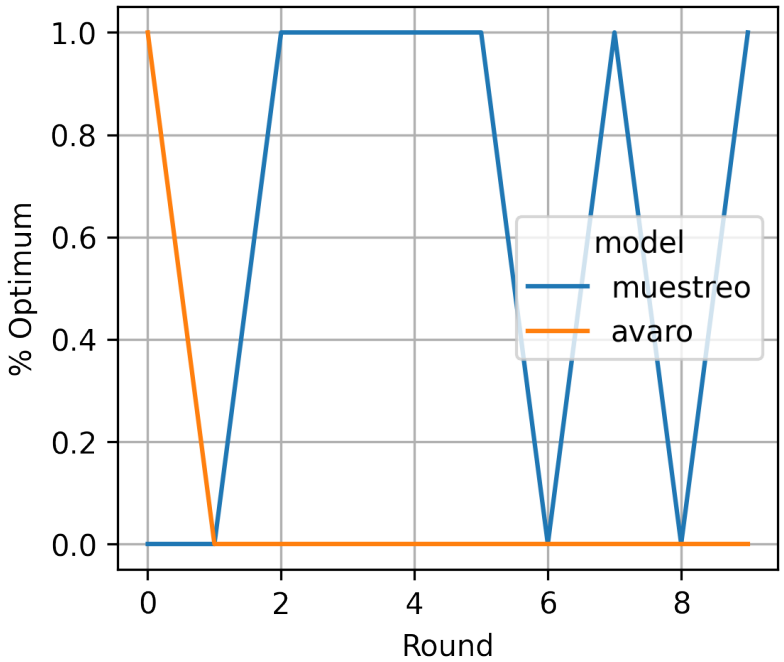
- La estrategia de muestreo exhibe un comportamiento aleatorio.
- La estrategia avara probó el brazo óptimo y tuvo éxito.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Explorar vs. aprovechar (2/3)

Dos enfoques extremos:

- **Explorar**: Muestrear ambos brazos.
- **Aprovechar**: Seleccionar el brazo que haya dado mejores beneficios hasta ahora (estrategia avara).



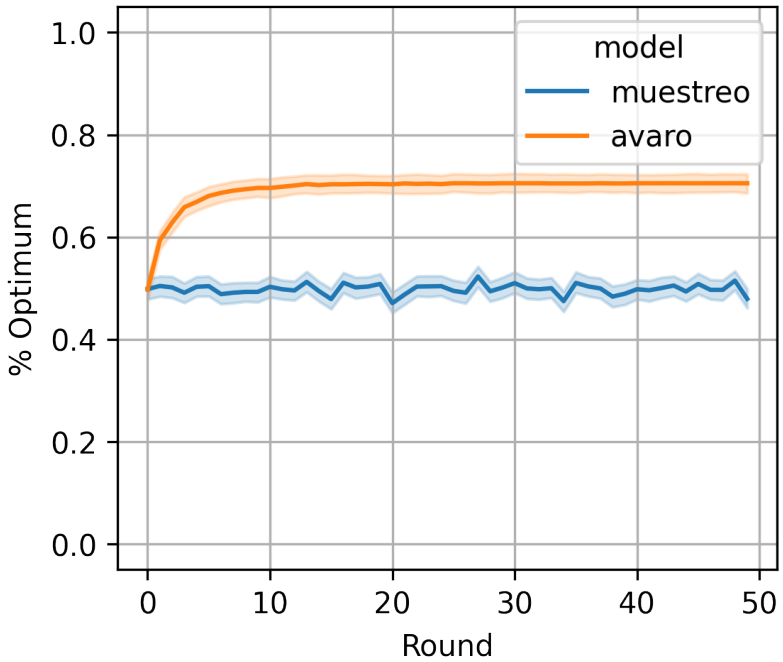
- La estrategia avara probó el brazo óptimo sin éxito.
- A continuación, la estrategia avara tuvo éxito al probar el brazo que NO es óptimo.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Explorar vs. aprovechar (3/3)

Dos enfoques extremos:

- **Explorar**: Muestrear ambos brazos.
- **Aprovechar**: Seleccionar el brazo que haya dado mejores beneficios hasta ahora (estrategia avara).



Promedio sobre 50 experimentos de 50 trials cada uno.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Posibles soluciones

Existe un abanico de maneras de enfrentar el dilema entre explorar vs. aprovechar:

- Avaro optimista inicial
- ϵ-greedy
- ϵ-greedy con recocido (annealing)
- Upper Confidence Bound
- Softmax
- Etc.

☞ Aquí solo mencionaremos la estrategia ϵ-greedy.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

ϵ-greedy (1/2)

Balance entre aprovechar (con probabilidad $1 - \epsilon$) y explorar (con probabilidad ϵ).

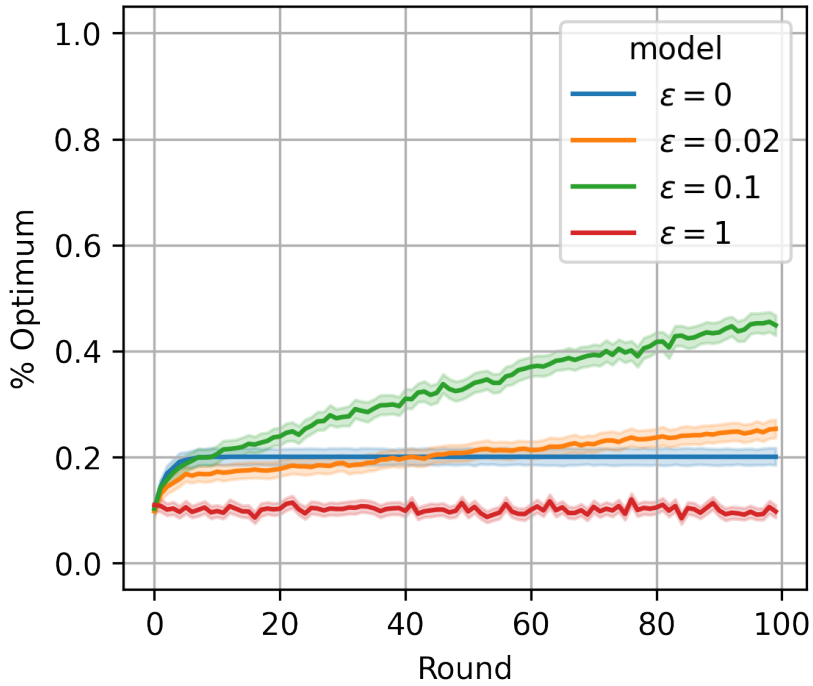
Algorithm 1: ϵ-greedy bandit algorithm

Data: una probabilidad de exploración ϵ (donde $0 \leq \epsilon \leq 1$)
Result: índice del brazo seleccionado
$Q(a) \leftarrow 0$ para cada brazo a;
while *True* **do**
  if *probabilidad* $1 - \epsilon$ **then**
    $a \leftarrow \arg\max Q(a)$;
  else
    $a \leftarrow a$ aleatoria;
  end
  Presentar la acción a al entorno y obtener la recompensa r;
  $Q(a) \leftarrow Q(a) + \alpha\left[r - Q(a)\right]$
end

MACC
Matemáticas Aplicadas y Ciencias de la Computación

ϵ-greedy (2/2)

Resultados:



Protocolo:

10 brazos. 50 experimentos de 50 trials.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Definiciones

- **Utilidad**:

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots = \sum_{t=0}^{\infty} \gamma^k r_{t+1}$$

Definiciones

- **Utilidad**:

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots = \sum_{t=0}^{\infty} \gamma^k r_{t+1}$$

- **Política**: Una función π que para cada estado s retorna una distribución de probabilidades sobre las acciones posibles, de tal manera que $\pi(a|s)$ es la probabilidad de la acción a en el estado s.

Definiciones

- **Utilidad**:

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots = \sum_{t=0}^{\infty} \gamma^k r_{t+1}$$

- **Política**: Una función π que para cada estado s retorna una distribución de probabilidades sobre las acciones posibles, de tal manera que $\pi(a|s)$ es la probabilidad de la acción a en el estado s.
- **Valor de un estado**: La utilidad esperada $v_\pi(s)$ de seguir la política π desde el estado s: $v_\pi(s) = \mathbb{E}[G|s]$.

Definiciones

- **Utilidad**:

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots = \sum_{t=0}^{\infty} \gamma^k r_{t+1}$$

- **Política**: Una función π que para cada estado s retorna una distribución de probabilidades sobre las acciones posibles, de tal manera que $\pi(a|s)$ es la probabilidad de la acción a en el estado s.
- **Valor de un estado**: La utilidad esperada $v_\pi(s)$ de seguir la política π desde el estado s: $v_\pi(s) = \mathbb{E}[G|s]$.
- **Valor de una acción**: La utilidad esperada de ejecutar una acción a en el estado s y luego actuar de acuerdo a π: $q_\pi(s, a) = \mathbb{E}[G|s, a]$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Estocasticidad de las transiciones

Después de que el agente ejecuta la acción a en el estado s, se obtiene el estado s_i con probabilidad $p(s_i|s,a)$.

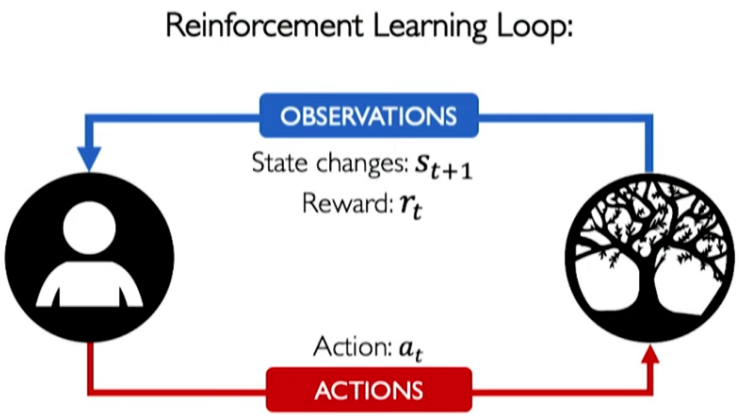
$$\{p(s_1|s,a); p(s_2|s,a); \ldots; p(s_n|s,a)\}$$

Propiedad de Markov

☞ Independencia del camino

$$p(s_{t+1}|s_0, a_0, s_1, a_1, \ldots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$$

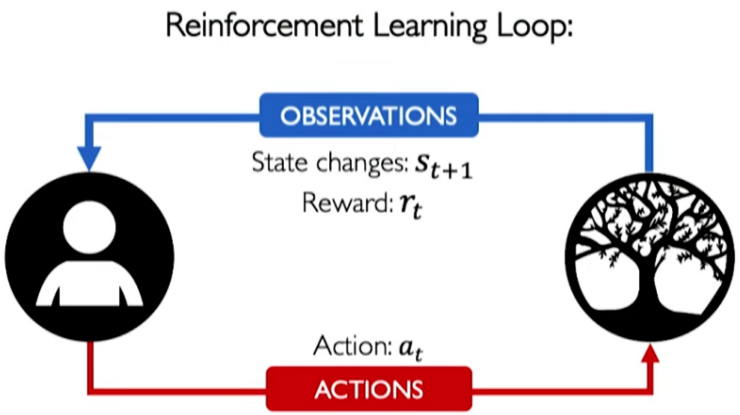
Componentes de los MDP



- Conjunto de estados
- Subconjunto de terminales
- Conjunto de acciones
- Transiciones $p(s'|s, a)$
- Recompensas $r(s, a, s')$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Componentes de los MDP

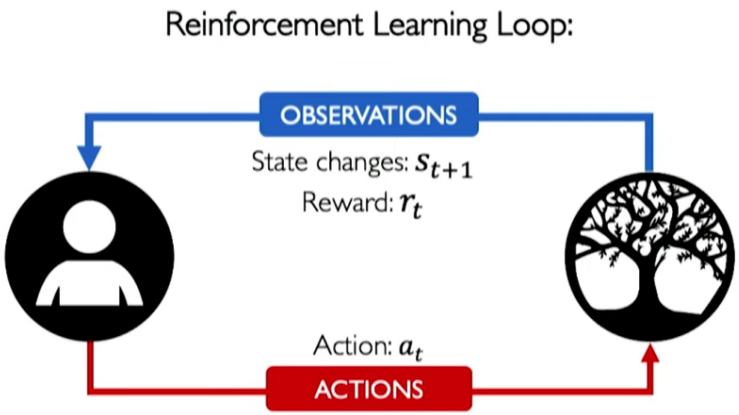


- Conjunto de estados
- Subconjunto de terminales
- Conjunto de acciones
- Transiciones $p(s'|s,a)$
- Recompensas $r(s,a,s')$

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

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Componentes de los MDP

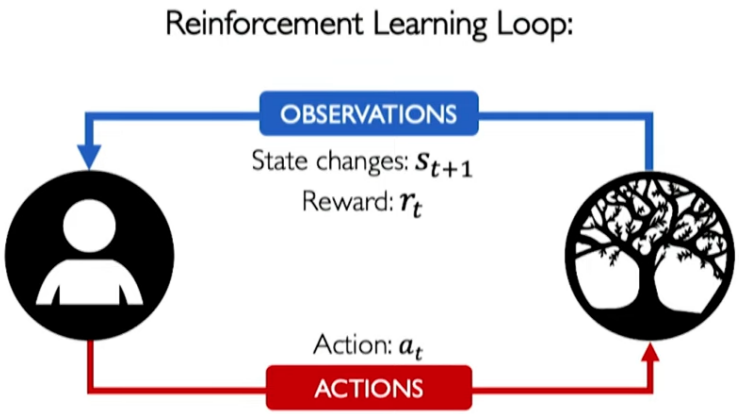


- Conjunto de estados
- Subconjunto de terminales
- Conjunto de acciones
- ~~Transiciones $p(s'|s,a)$~~
- ~~Recompensas $r(s,a,s')$~~

☞ Vamos a asumir que no conocemos el modelo del MDP.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Componentes de los MDP



- Conjunto de estados
- Subconjunto de terminales
- Conjunto de acciones
- ~~Transiciones $p(s'|s,a)$~~
- ~~Recompensas $r(s,a,s')$~~

☞ Intentamos estimar v_* y q_* directamente.

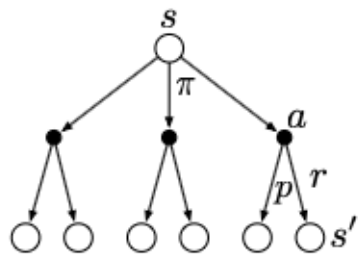
MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Usando la ecuación de Bellman

Programación dinámica:

$$V_{k+1}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \Big(p(s'|s,a) \Big[r + \gamma V_k(s') \Big] \Big)$$



Backup diagram for v_π

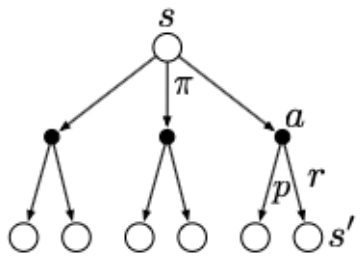
Temporal difference:

The backup diagram for TD(0)

Usando la ecuación de Bellman

Programación dinámica:

$$V_{k+1}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \left(p(s'|s,a)\left[r + \gamma V_k(s')\right]\right)$$



Backup diagram for v_π

Temporal difference:

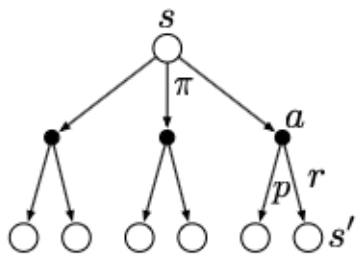
$$V_{k+1}(s) \leftarrow V_k(s) + \alpha\,(G - V_k(s))$$

The backup diagram for TD(0)

Usando la ecuación de Bellman

Programación dinámica:

$$V_{k+1}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \left(p(s'|s,a) \left[r + \gamma V_k(s') \right] \right)$$



Backup diagram for v_π

Temporal difference:

$$V_{k+1}(s) \leftarrow V_k(s) + \alpha \left(r + \gamma V_k(s') - V_k(s) \right)$$

The backup diagram for TD(0)

Regla de aprendizaje

$$V(s) \leftarrow \overbrace{V(s)}^{\text{estimado anterior}} + \underbrace{\alpha}_{\text{step size}} \Big(\overbrace{r_1 + \gamma \underbrace{V(s_1)}_{\textit{bootstrap}}}^{\text{nuevo dato}} - \underbrace{V(s)}_{\text{estimado anterior}} \Big)$$

Aprendiendo una política (SARSA)

Suponga una política π.

Aprendiendo una política (SARSA)

Suponga una política π.

- Regla para actualizar valores de pares estado acción:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma Q(s',a') - Q(s,a)\Big)$$

Donde s' es el estado al que se llegó al realizar a en s, y $a' \leftarrow$ acción al muestrear $\pi(s')$.

Aprendiendo una política (SARSA)

Suponga una política π.

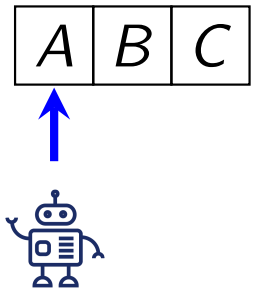
- Regla para actualizar valores de pares estado acción:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma Q(s',a') - Q(s,a)\Big)$$

  Donde s' es el estado al que se llegó al realizar a en s, y $a' \leftarrow$ acción al muestrear $\pi(s')$.
- Mejorar $\pi(s)$ con ϵ-greedy sobre Q para todo s.

Tabla Q

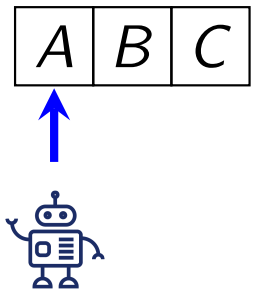
Considere el entorno del ABC, presentado hace dos clases.



	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

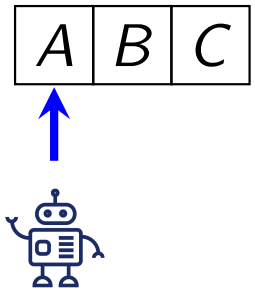


	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

Aleatoriamente, el agente escoge la acción Izquierda.

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



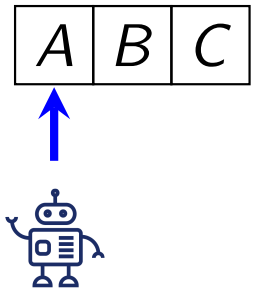
	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

El agente se queda en el estado A.

$$s = A$$
$$a = \text{Izquierda}$$
$$s' = A$$
$$a' \leftarrow \text{acción aleatoria}$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

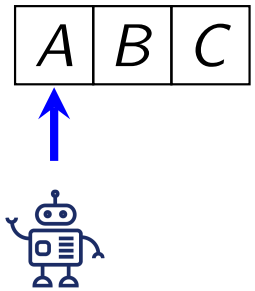
El agente se queda en el estado A.

$$s = A$$
$$a = \text{Izquierda}$$
$$s' = A$$
$$a' \leftarrow \text{Derecha}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

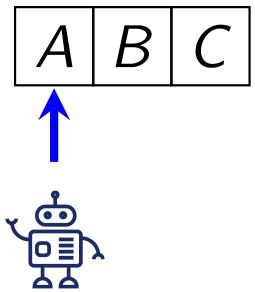


	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

$$q(A, \mathrm{I}) += \alpha\Big(-1 + \gamma q(A, \mathrm{D}) - q(A, \mathrm{I})\Big)$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



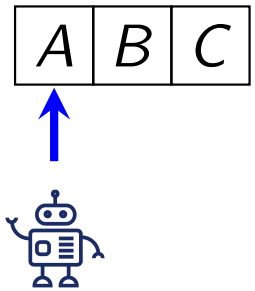
	Izquierda	Derecha
A	0	0
B	0	0
C	0	0

$$q(A, \text{I}) += 0.1\Big(-1 + 0.8 \times 0 - 0\Big)$$

Suponga $\alpha = 0.1$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

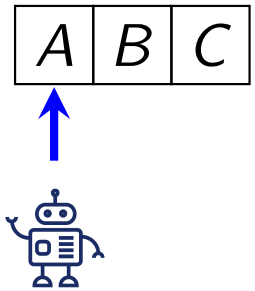


	Izquierda	Derecha
A	-0.1	0
B	0	0
C	0	0

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

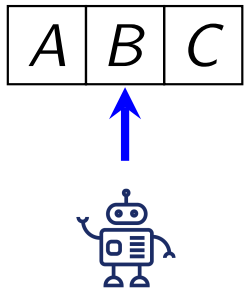


	Izquierda	Derecha
A	-0.1	0
B	0	0
C	0	0

El agente escoge la acción con mayor valor q, a saber, Derecha (esto ocurre con probabilidad ϵ).

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



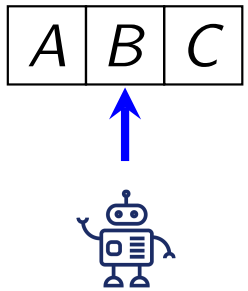
	Izquierda	Derecha
A	-0.1	0
B	0	0
C	0	0

Supongamos que el agente llega a B (esto ocurre con probabilidad 0.9).

$$s = A$$
$$a = \text{Derecha}$$
$$s' = B$$
$$a' \leftarrow \text{Derecha}$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

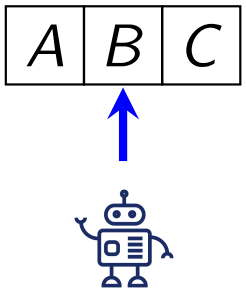


	Izquierda	Derecha
A	-0.1	0
B	0	0
C	0	0

$$q(A,\mathrm{D}) += \alpha\Big(-1+\gamma q(B,\mathrm{D})-q(A,\mathrm{D})\Big)$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

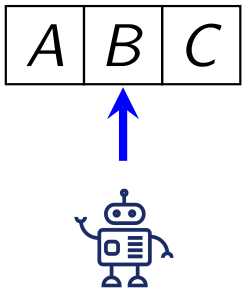


	Izquierda	Derecha
A	-0.1	0
B	0	0
C	0	0

$$q(A, \mathrm{D}) += 0.1\Big(-1 + 0.8 \times 0 - 0\Big)$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



	Izquierda	Derecha
A	-0.1	-0.1
B	0	0
C	0	0

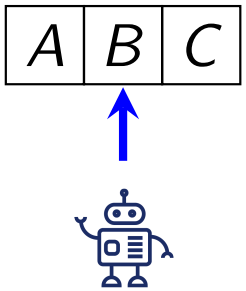
Aleatoriamente, el agente escoge la acción Derecha.

$$s = B$$
$$a = \text{Derecha}$$
$$s' = C$$
$$a' \leftarrow \text{Derecha}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

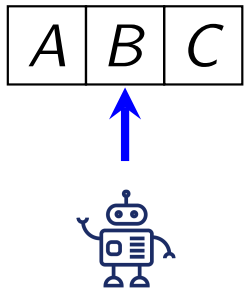


	Izquierda	Derecha
A	-0.1	-0.1
B	0	0
C	0	0

El agente llega a C.

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

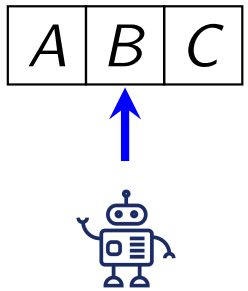


	Izquierda	Derecha
A	-0.1	-0.1
B	0	0
C	0	0

$$q(B,\mathrm{D}) += \alpha\Big(10+\gamma q(C,\mathrm{D})-q(B,\mathrm{D})\Big)$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.

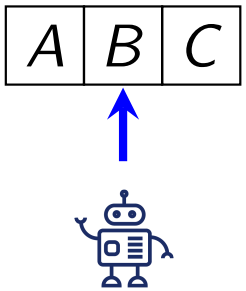


	Izquierda	Derecha
A	-0.1	-0.1
B	0	0
C	0	0

$$q(B, \mathrm{D}) += 0.1\Big(10 + 0.8 \times 0 - 0\Big)$$

Tabla Q

Considere el entorno del ABC, presentado hace dos clases.



	Izquierda	Derecha
A	-0.1	-0.1
B	0	1
C	0	0

Pseudocódigo SARSA

Algorithm 3: SARSA agent (update rule)

Data: Una acción a, un estado s' y una recompensa r

$Q(s,a) \leftarrow$ self.$Q(s,a)$ (action-value para cada (s,a));

$\pi \leftarrow$ self.π (política ϵ-greedy sobre Q);

$s \leftarrow$ self.s (estado anterior);

$a' \leftarrow$ acción dada por π en s';

self.$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma Q(s',a') - Q(s,a)\Big)$;

self.$\pi \leftarrow$ mejorar $\pi(s)$ con ϵ-greedy sobre Q;

self.$s \leftarrow s'$;

SARSA vs. Q-learning (1/2)

SARSA:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma Q(s',a') - Q(s,a)\Big)$$

Mejorar $\pi(s)$ con ϵ-greedy y Q

Q-learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big)$$

Mejorar $\pi(s)$ con ϵ-greedy y Q

MACC
Matemáticas Aplicadas y Ciencias de la Computación

SARSA vs. Q-learning (2/2)

Algorithm 3: SARSA agent (update rule)

Data: Una acción a, un estado s' y una recompensa r
$Q(s,a) \leftarrow$ self.$Q(s,a)$ (action-value para cada (s,a));
$\pi \leftarrow$ self.π (política ϵ-greedy sobre Q);
$s \leftarrow$ self.s (estado anterior);
$a' \leftarrow$ acción dada por π en s';
self.$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma Q(s',a') - Q(s,a)\Big)$;
self.$\pi \leftarrow$ mejorar $\pi(s)$ con ϵ-greedy sobre Q;
self.$s \leftarrow s'$;

Algorithm 5: Q-learning agent

Data: Una acción a, un estado s' y una recompensa r
$Q(s,a) \leftarrow$ self.$Q(s,a)$ (action-value para cada (s,a));
$\pi \leftarrow$ self.π (política ϵ-greedy sobre Q);
$s \leftarrow$ self.s (estado anterior);
self.$Q(s,a) \leftarrow Q(s,a) + \alpha\Big(r + \gamma \,\text{máx}_{a'}\, Q(s',a') - Q(s,a)\Big)$;
self.$\pi \leftarrow$ actualizar $\pi(s)$ con ϵ-greedy sobre Q;
self.$s \leftarrow s'$;

Matemáticas Aplicadas y Ciencias de la Computación

The cliff problem

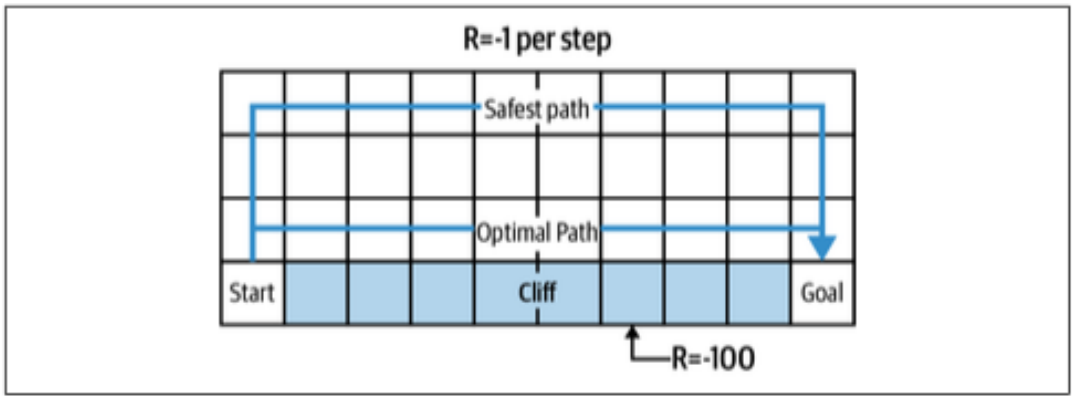


Figure 3-1. A depiction of the grid environment with a cliff along one side.[1]

Optimal policy — SARSA vs Q-learning

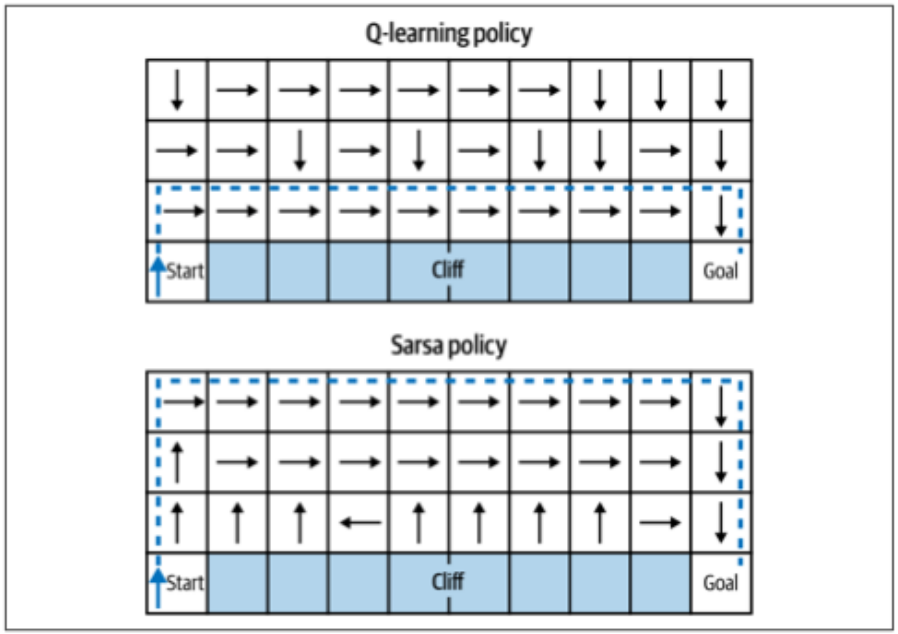


Figure 3-3. The policies derived by Q-learning and SARSA agents. Q-learning tends to prefer the optimal route. SARSA prefers the safe route.

Tabla Q

	Izquierda	Derecha	Arriba	Abajo
20	-11	-10	-11	-100
21	-10	-9	-10	-100

¿Cuál es el valor actualizado de

$$q(20, \textit{Derecha})$$

si $a' =$ Abajo, usando:

- SARSA ?
- Q-learning ?

Utility — SARSA vs Q-learning

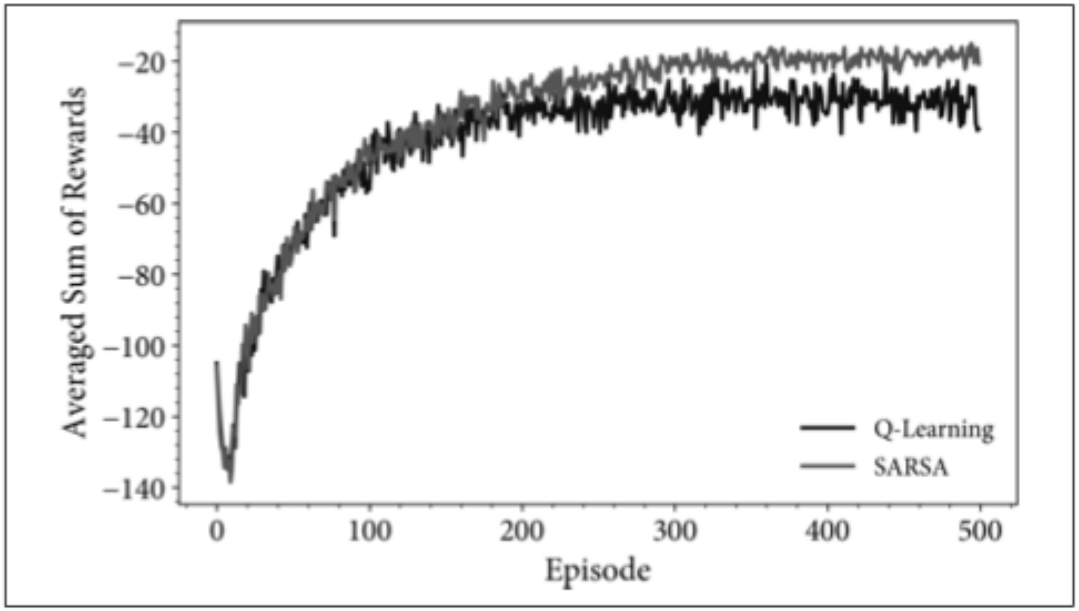


Figure 3-2. A comparison of Q-learning against SARSA for a simple grid problem. The agents were trained upon the environment in Figure 3-1. I used $\gamma \doteq 1.0$, $\epsilon \doteq 0.1$, *and* $\alpha \doteq 0.5$. *The rewards for each episode were captured and averaged over 100 trials.*

Take away

En esta sesión usted aprendió:

- Analizar el aprendizaje por refuerzo como la combinación de la estimación de la recompensa a largo plazo, la corrección del error de estimación y el balance entre aprovechar y explorar.
- Los métodos de diferencia temporal SARSA y Q-learning.