UNIVERSITAT DE BARCELONA

Inteligencia Artificial

Aprendizaje por refuerzo

Aprendizaje por refuerzo

- Juguemos:
  - Veis una luz encendida, dos botones (uno a la derecha y otro a la izquierda) y un feriante con mala cara que os pide un euro por cada vez que presionéis un botón y os dice que juguéis. No parece que haya otra opción que jugar a su juego... así que le dais el primer euro y os pregunta qué botón queréis apretar.

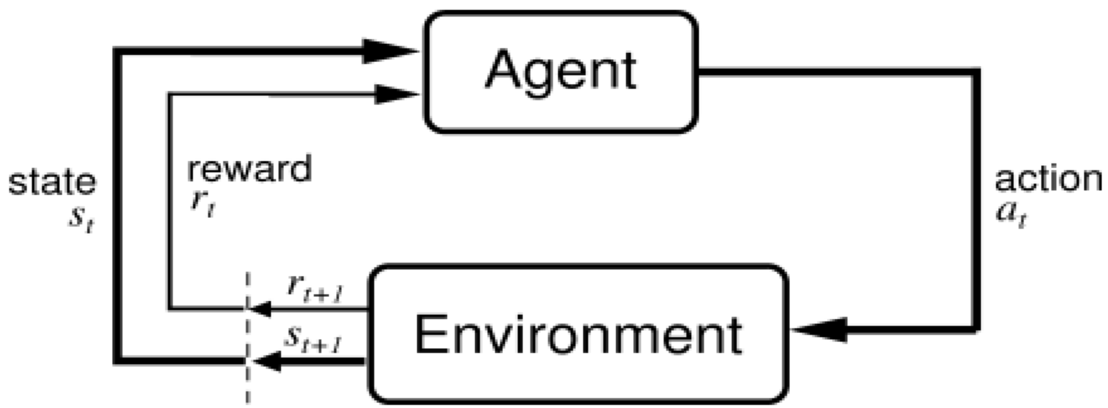
Aprendizaje por refuerzo

- Juguemos:
  - Veis una luz encendida, dos botones (uno a la derecha y otro a la izquierda) y un feriante con mala cara que os pide un euro por cada vez que presionéis un botón y os dice que juguéis. No parece que haya otra opción que jugar a su juego... así que le dais el primer euro y os pregunta qué botón queréis apretar.

Pag. 840 libro (4^{th} ed.):
Imagine you are playing a new game whose rules you do not know;
After a hundred or so moves, the referee tells you “You lose”.
That is RL in a nutshell.

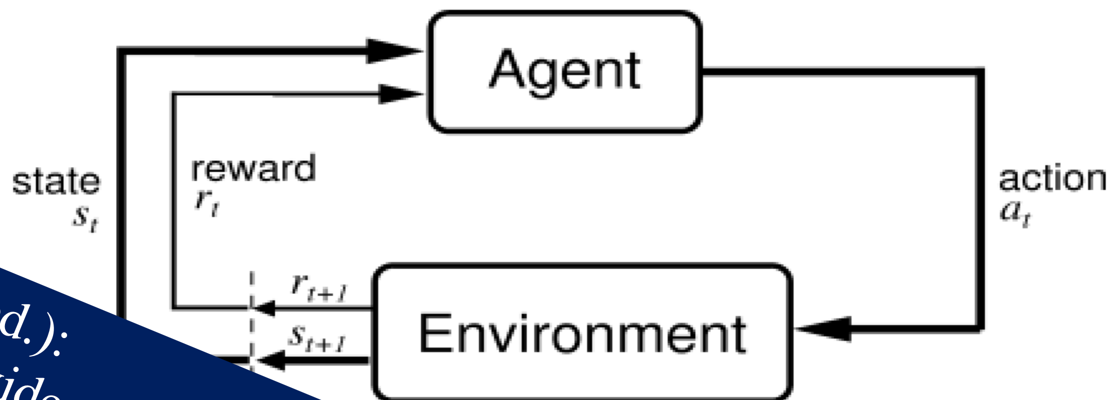
Aprendizaje por refuerzo

- Idea fundamental:
  - Recibimos información en forma de recompensa.
  - La utilidad del agente viene dada por la función de recompensa.
  - El agente debe aprender a actuar para maximizar la recompensa esperada.



Aprendizaje por refuerzo

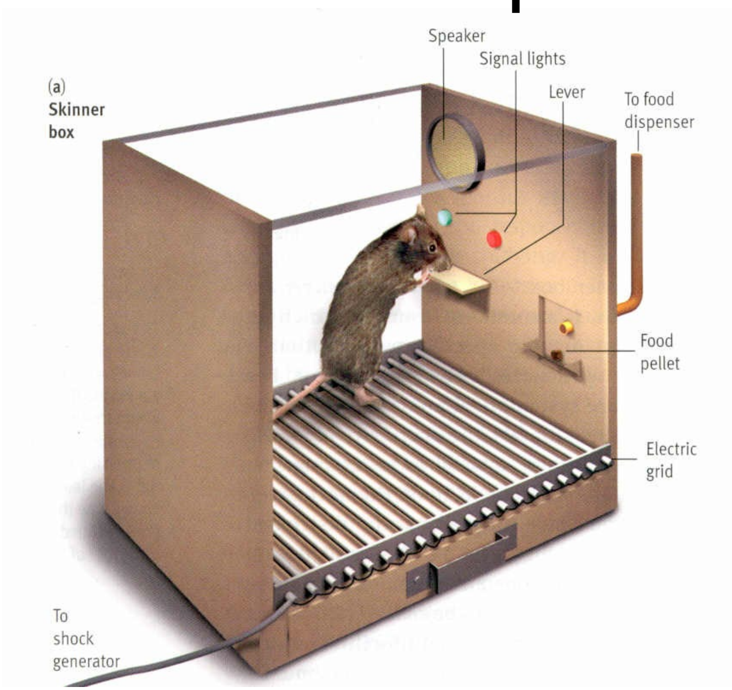
- Idea fundamental:
  - Recibimos información en forma de recompensa.
  - La utilidad del agente viene dada por la función de recompensa.
  - El agente debe aprender a actuar para maximizar la recompensa esperada.



Pag. 840 libro (4th ed.):
Racing or chess provide sparse rewards

RL en otras disciplinas

- El aprendizaje por refuerzo se ha estudiado en psicología desde hace más de 60 años.
- Aprendizaje por refuerzo (antiguo): http://www.dailymotion.com/video/xhcilv_aprendizaje-el-refuerzo_school
- Video dopamina http://www.dailymotion.com/video/x28uxgc



- Recompensas: comida, hambre, dolor, drogas, etc.

UNIVERSITAT DE BARCELONA

Ej: Aprendizaje animal

- Ejemplo: comida
  - Las abejas terminan aprendiendo trayectorias de comida óptimas en campos de flores artificiales con suministro de néctar controlado.
  - Las abejas tienen un conexión neuronal directa del sistema de medición de la ingesta de néctar al sistema de planificación motor.

UNIVERSITAT DE BARCELONA

Aprendizaje por refuerzo

- Aprendizaje por refuerzo
- Aprendizaje por refuerzo pasivo
  - Estimación directa de la utilidad
  - Aprendizaje basado en modelo
  - Aprendizaje basado en diferencia temporal
- Aprendizaje por refuerzo activo
  - Q-aprendizaje
  - Selección exploratoria de acciones

UNIVERSITAT DE BARCELONA

Aprendizaje por refuerzo

- Seguimos modelando nuestro mundo mediante un PDM:
  - S - conjunto de estados
  - A - conjunto de acciones
  - T: S x A x S $\rightarrow \mathfrak{R}$ - función de transición
  - R: S x A x S $\rightarrow \mathfrak{R}$ – función de recompensa
- Seguimos buscando una política óptima π^*
- Pero ahora: No conocemos **a priori** ni T ni R.
  - No sabemos ni qué estados son buenos ni lo que hacen las acciones.
  - Tenemos que intentar cosas para aprender sus consecuencias.

Aprendizaje por refuerzo pasivo

- No conocemos las transiciones T(s,a,s')
- No conocemos las recompensas R
- Nos dan una política **fija** π(s)
- Objetivo:

  Aprender los valores de los estados V(s)

¿Cómo evaluamos V?

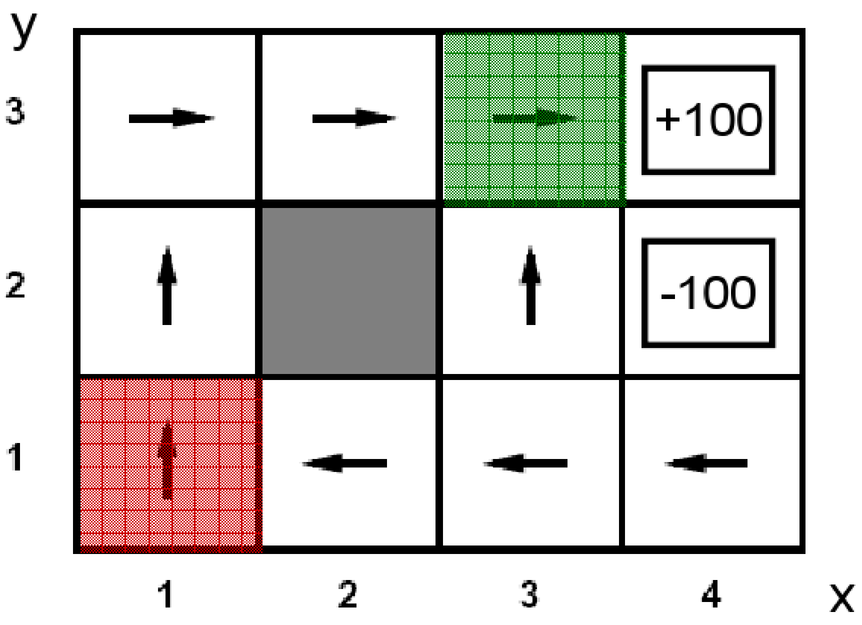
a partir de la experiencia

Estimación directa de la utilidad

La **percepción** determina el estado y la recompensa actual

- Episodes:

(1,1) up



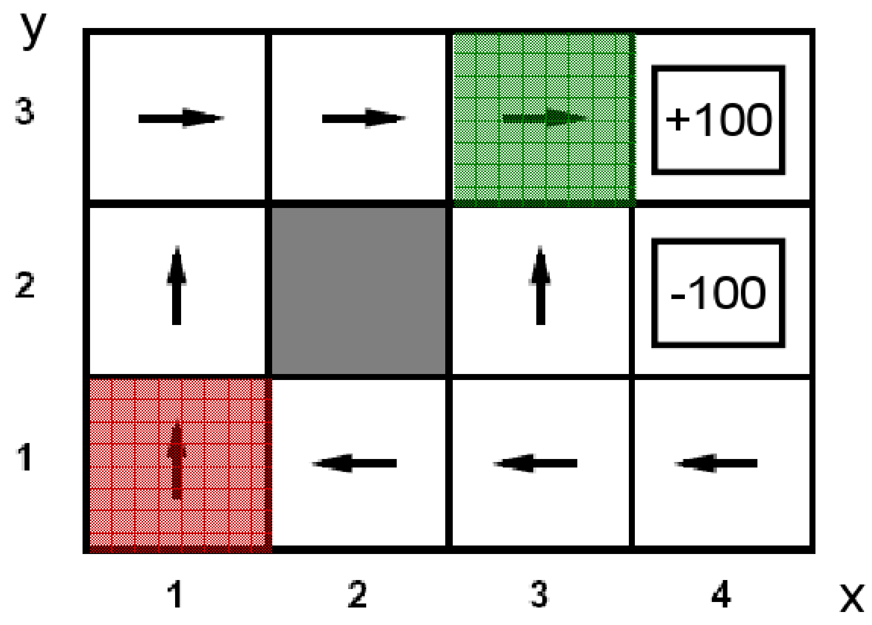
Estimación directa de la utilidad

La **percepción** determina el estado y la recompensa actual

- Episodes:

(1,1) up -1

(1,2)



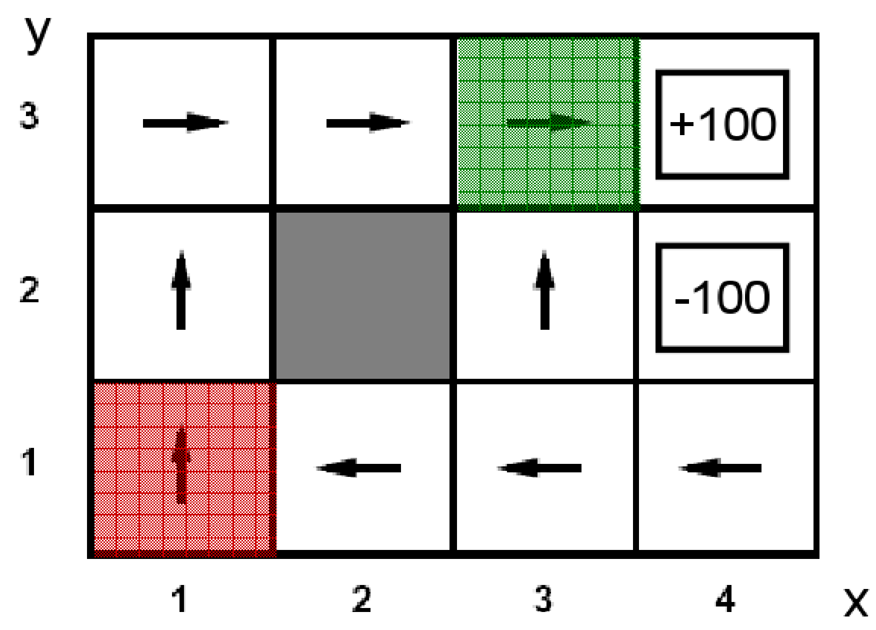
Estimación directa de la utilidad

La **percepción** determina el estado y la recompensa actual

- Episodes:

(1,1) up -1
(1,2) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(3,3) right -1
(4,3) exit +100
(done)

(1,1) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(4,2) exit -100
(done)



V(1,1) ?

V(3,3) ?

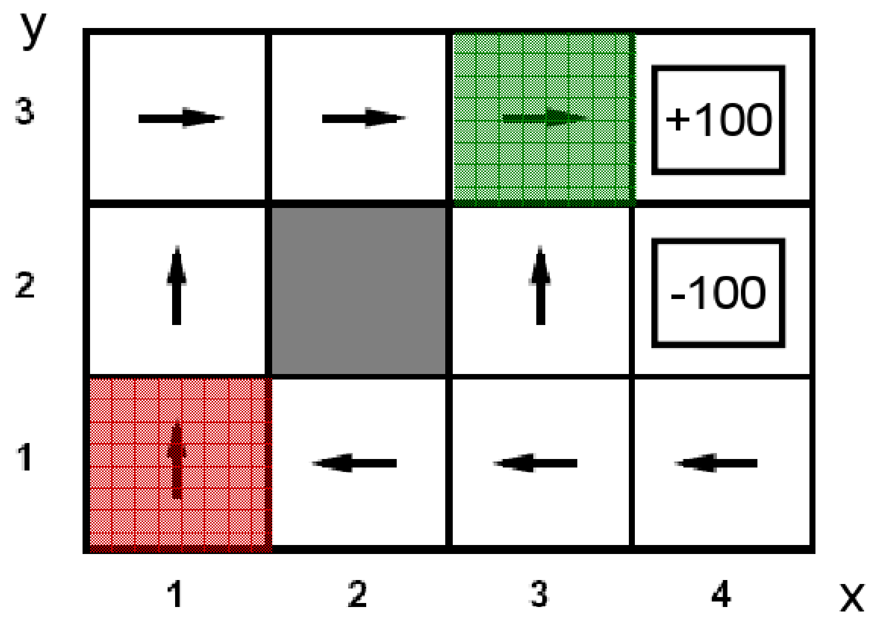
A l hacer estos dos recorridos ¿Qué experiencia/recompensa se ha acumulado sobre cada uno de estos estados?

Estimación directa de la utilidad

La **percepción** determina el estado y la recompensa actual

- Episodes:

(1,1) up -1	(1,1) up -1
(1,2) up -1	(1,2) up -1
(1,2) up -1	(1,3) right -1
(1,3) right -1	(2,3) right -1
(2,3) right -1	(3,3) right -1
(3,3) right -1	(3,2) up -1
(3,2) up -1	(4,2) exit -100
(3,3) right -1	(done)
(4,3) exit +100	
(done)	



V(1,1) ~ (92 + -106) / 2 = -7

V(3,3) ~ (99 + 97 + -102) / 3 = 31.3

Aprendizaje basado en modelo

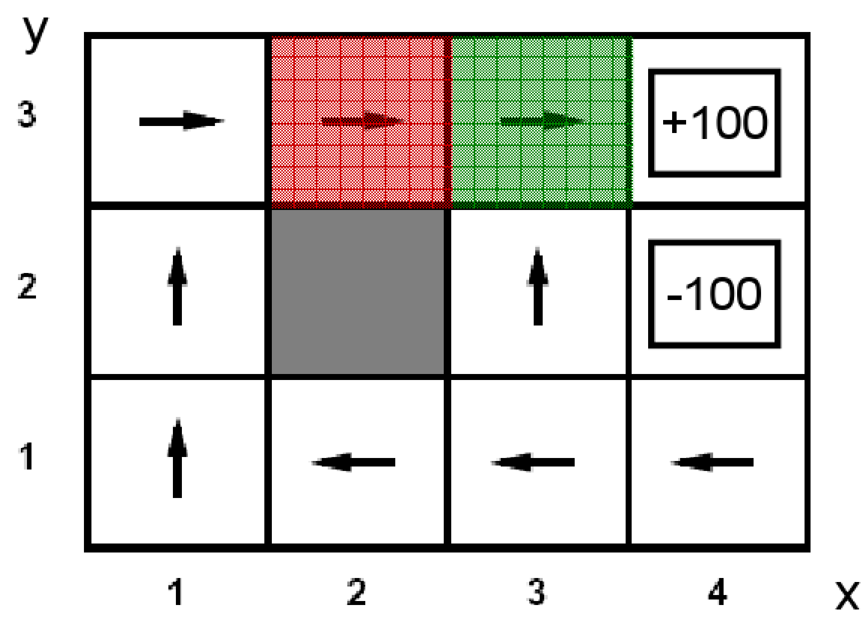
- Alternativa:
  - Estimar el MDP a partir de las observaciones
  - Determinar el valor de cada estado resolviendo el MDP estimado
- Caso más sencillo
  - Contar los estados a los que se llega para cada s,a
  - Estimar T(s,a,s')
  - Descubrir R(s,a,s') cada vez que ocurra

ADP: Adaptive Dynamic Programming

Aprendizaje basado en modelo

- Episodes:

(1,1) up -1	(1,1) up -1
(1,2) up -1	(1,2) up -1
(1,2) up -1	(1,3) right -1
(1,3) right -1	(2,3) right -1
(2,3) right -1	(3,3) right -1
(3,3) right -1	(3,2) up -1
(3,2) up -1	(4,2) exit -100
(3,3) right -1	(done)
(4,3) exit +100	
(done)	



$\gamma = 1$

T(<3,3>, right, <4,3>) **?**

T(<2,3>, right, <3,3>) **?**

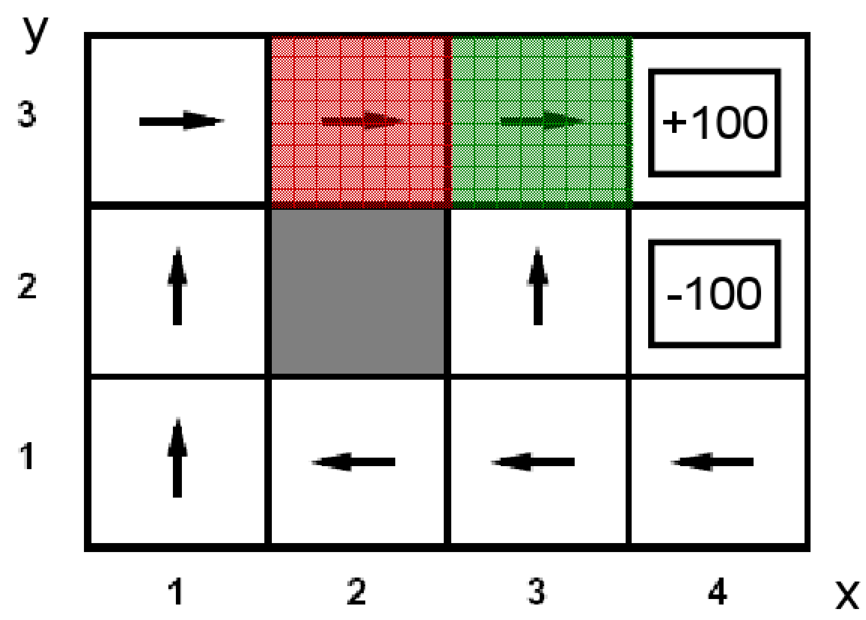
A l hacer estos dos recorridos ¿Qué proporción de veces se ha realizado la transición?

Aprendizaje basado en modelo

- Episodes:

(1,1) up -1
(1,2) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(3,3) right -1
(4,3) exit +100
(done)

(1,1) up -1
(1,2) up -1
(1,3) right -1
(2,3) right -1
(3,3) right -1
(3,2) up -1
(4,2) exit -100
(done)



$\gamma = 1$

T(<3,3>, right, <4,3>) = 1 / 3

T(<2,3>, right, <3,3>) = 2 / 2

Aprendizaje basado en modelo

función AGENTE-PASIVO-ADP (percepción) **devuelve** una acción
  entradas: *percepción*, indica el estado actual s' y la señal de recompensa r'
  estática: π, una política fija
    mdp, un MDP con modelo T, recompensas R, descuento γ
    U, una tabla de utilidades, inicialmente vacía
    N_{sa}, una tabla de frecuencias para pares estado-acción, inicialmente a cero
    $N_{sas'}$, una tabla de frecuencias para tripletas estado-acción-estado, inicialmente a cero
    s, a, el estado y la acción previa, inicialmente a nulo *(null)*
  si s' es nuevo **entonces hacer** $U[s'] \leftarrow r'$; $R[s'] \leftarrow r'$
  si s no es nulo (null) **entonces hacer**
    incrementar $N_{sa}[s, a]$ y $N_{sas'}[s, a, s']$
    para cada t tal que $N_{sas'}[s, a, t]$ no sea cero **hacer**
      $T[s, a, t] \leftarrow N_{sas'}[s, a, t] \,/\, N_{sa}[s, a]$
  $U \leftarrow$ DETERMINACIÓN-VALOR(π, U, *mdp*)
  si TERMINAL?[s'] **entonces** $s, a \leftarrow$ nulo *(null)* **si no** $s, a \leftarrow s', \pi\,[s']$
  devolver a

Figura 21.2 Un agente de aprendizaje por refuerzo pasivo basado en programación dinámica adaptativa. Para simplificar el código, hemos asumido que cada percepción puede dividirse en un estado percibido y una señal de recompensa.

Aprendizaje basado en modelo

función AGENTE-PASIVO-ADP (percepción) **devuelve** una acción
  entradas: *percepción*, indica el estado actual s' y la señal de recompensa r'
  estática: π, una política fija
    mdp, un MDP con modelo T, recompensas R, descuento γ
    U, una tabla de utilidades, inicialmente vacía
    N_{sa}, una tabla de frecuencias para pares estado-acción, inicialmente a cero
    $N_{sas'}$, una tabla de frecuencias para tripletas estado-acción-estado, inicialmente a cero
    s, a, el estado y la acción previa, inicialmente a nulo *(null)*
  si s' es nuevo **entonces hacer** $U[s'] \leftarrow r'$; $R[s'] \leftarrow r'$
  si s no es nulo (null) **entonces hacer**
    incrementar $N_{sa}[s, a]$ y $N_{sas'}[s, a, s']$
    para cada t tal que $N_{sas'}[s, a, t]$ no sea cero **hacer**
      $T[s, a, t] \leftarrow N_{sas'}[s, a, t] \,/\, N_{sa}[s, a]$
  $U \leftarrow$ DETERMINACIÓN-VALOR(π, U, *mdp*)
  si TERMINAL?[s'] **entonces** $s, a \leftarrow$ nulo *(null)* **si no** $s, a \leftarrow s', \pi[s']$
  devolver a

Figura 21.2 Un agente de aprendizaje por refuerzo pasivo basado en programación dinámica adaptativa. Para simplificar el código, hemos asumido que cada percepción puede dividirse en un estado percibido y una señal de recompensa.

Se actualiza el modelo de **todos** los estados **t** sucesores de s

Aprendizaje basado en modelo (4 ed)

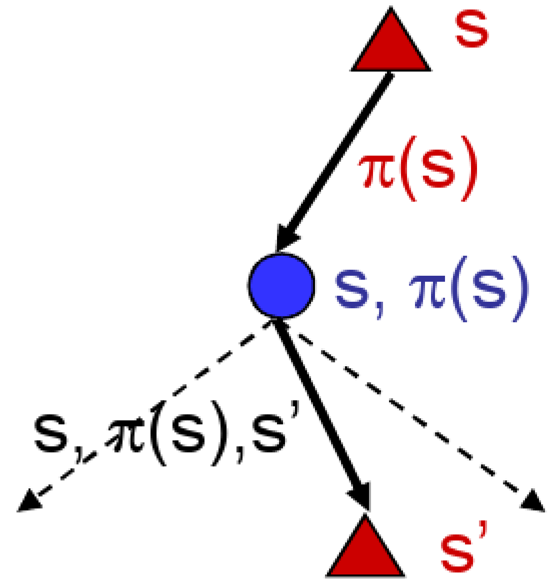
function PASSIVE-ADP-LEARNER(*percept*) **returns** an action
  inputs: *percept*, a percept indicating the current state s' and reward signal r
  persistent: π, a fixed policy
    mdp, an MDP with model P, rewards R, actions A, discount γ
    U, a table of utilities for states, initially empty
    $N_{s'|s,a}$, a table of outcome count vectors indexed by state and action, initially zero
    s, a, the previous state and action, initially null

  if s' is new **then** $U[s'] \leftarrow 0$
  if s is not null **then**
    increment $N_{s'|s,a}[s,a][s']$
    $R[s, a, s'] \leftarrow r$
    add a to $A[s]$
    $\mathbf{P}(\cdot \mid s,a) \leftarrow$ NORMALIZE($N_{s'|s,a}[s, a]$)
    $U \leftarrow$ POLICYEVALUATION(π, U, mdp)
    $s, a \leftarrow s', \pi[s']$
    return a

Figure 23.2 A passive reinforcement learning agent based on adaptive dynamic programming. The agent chooses a value for γ and then incrementally computes the P and R values of the MDP. The POLICY-EVALUATION function solves the fixed-policy Bellman equations, as described on page 567.

Recordat.:Evaluación de políticas

- Las actualizaciones de Bellman simplificadas nos permiten calcular V para una política preestablecida.
  - La nueva V es la esperanza asumiendo la V anterior como cierta
  - Desafortunadamente necesitamos T y R.



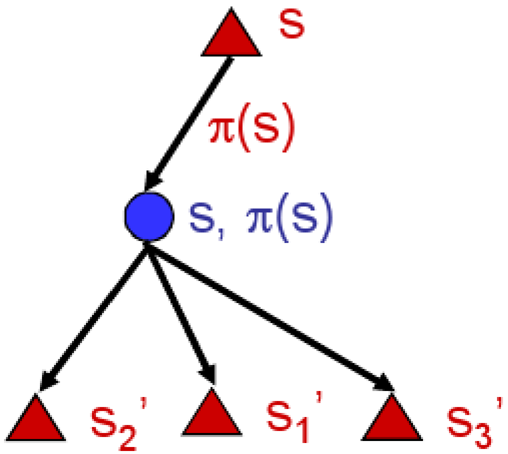
$$V_0^{\pi}(s) = 0$$

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_i^{\pi}(s')]$$

¿Podemos reemplazar la esperanza con la media?

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_i^{\pi}(s')]$$

- Podemos estimar a partir de las muestras que tenemos sin necesidad de construir un modelo



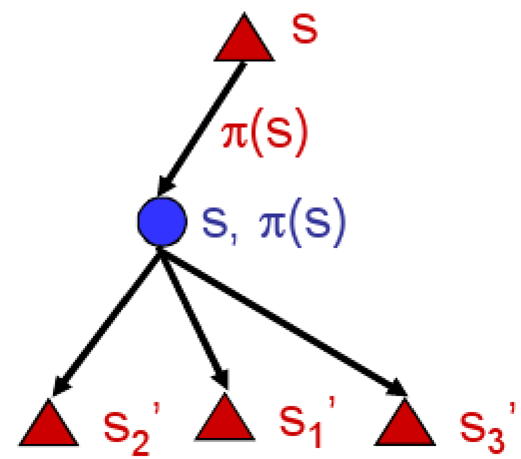
$$sample_1 = R(s, \pi(s), s_1') + \gamma V_i^{\pi}(s_1')$$
$$sample_2 = R(s, \pi(s), s_2') + \gamma V_i^{\pi}(s_2')$$
$$\dots$$
$$sample_k = R(s, \pi(s), s_k') + \gamma V_i^{\pi}(s_k')$$

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{k} sample_k$$

¿Podemos reemplazar la esperanza con la media?

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} T(s,\pi(s),s')[R(s,\pi(s),s') + \gamma V_i^{\pi}(s')]$$

- Podemos estimar a partir de las muestras que tenemos sin necesidad de construir un modelo



$$sample_1 = R(s,\pi(s),s_1') + \gamma V_i^{\pi}(s_1')$$

$$sample_2 = R(s,\pi(s),s_2') + \gamma V_i^{\pi}(s_2')$$

$$\cdots$$

$$sample_k = R(s,\pi(s),s_k') + \gamma V_i^{\pi}(s_k')$$

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{k} sample_k$$

Sample of V(s): $sample = R(s,\pi(s),s') + \gamma V^{\pi}(s')$

Update to V(s): $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample$

Same update: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

α : factor de aprendizaje

Aprendizaje TD

función AGENTE-PASIVO-TD (percepción) **devuelve** una acción
  entradas: percepción, una percepción indica el estado actual s' y la señal de recompensa r'
  estática: π, una política fijada
    U, una tabla de utilidades, inicialmente vacía
    N_s, una tabla de frecuencias por estados, inicialmente a cero
    s, a, r, el estado, la acción y la recompensa previa, inicialmente a nulo (null)

  si s' es nuevo **entonces** $U[s'] \leftarrow r'$
  si s no es nulo (null) **entonces hacer**
    incrementar $N_s[s]$
    $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
  si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$
  devolver a

Figura 21.4 Un agente de aprendizaje por refuerzo pasivo que aprende estimaciones de la utilidad usando diferencias temporales.

TD actualiza el modelo a partir de los estados sucesores de s **visitados**
(aproxima ADP)

Aprendizaje TD (0)

(e-libro Sutton and Barto, 2nd ed.)

Input: the policy π to be evaluated
Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0, \forall s \in \mathcal{S}^+$)
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        $A \leftarrow$ action given by π for S
        Take action A; observe reward, R, and next state, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until S is terminal

Figure 6.1: Tabular TD(0) for estimating v_π.

Aprendizaje TD: Ejemplo

Ejercicio: hacer los cálculos para los primeros 5 pasos del primer episodio

$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

(1,1) up -1

(1,2) up -1

(1,2) up -1

(1,3) right -1

(2,3) right -1

(3,3) right -1

(3,2) up -1

(3,3) right -1

(4,3) exit +100

(done)

(1,1) up -1

(1,2) up -1

(1,3) right -1

(2,3) right -1

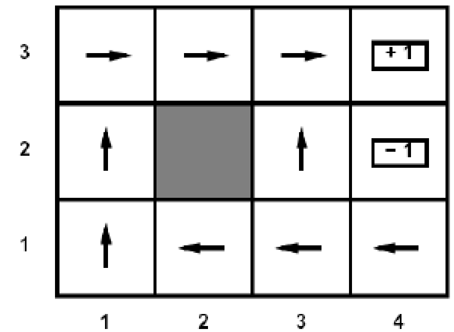
(3,3) right -1

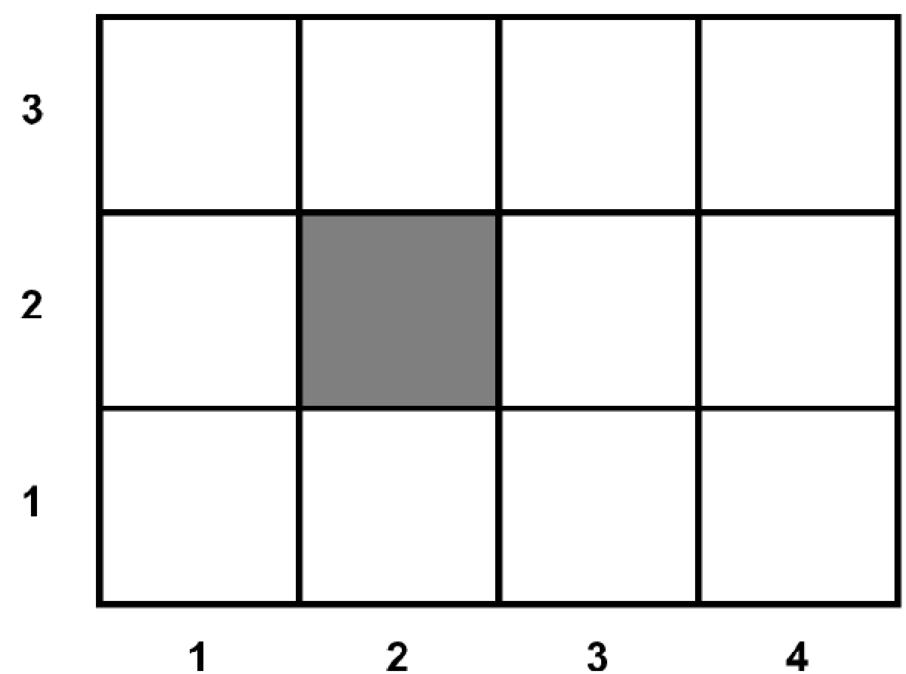
(3,2) up -1

(4,2) exit -100

(done)

Take $\gamma = 1$, $\alpha = 0.5$





Ejercicio: **1er** paso del primer episodio

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$

si s no es nulo (null) **entonces hacer**

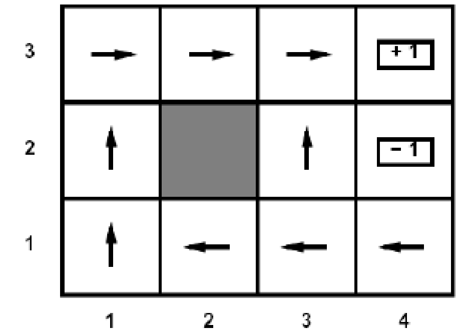
  incrementar $N_s[s]$

  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$

si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

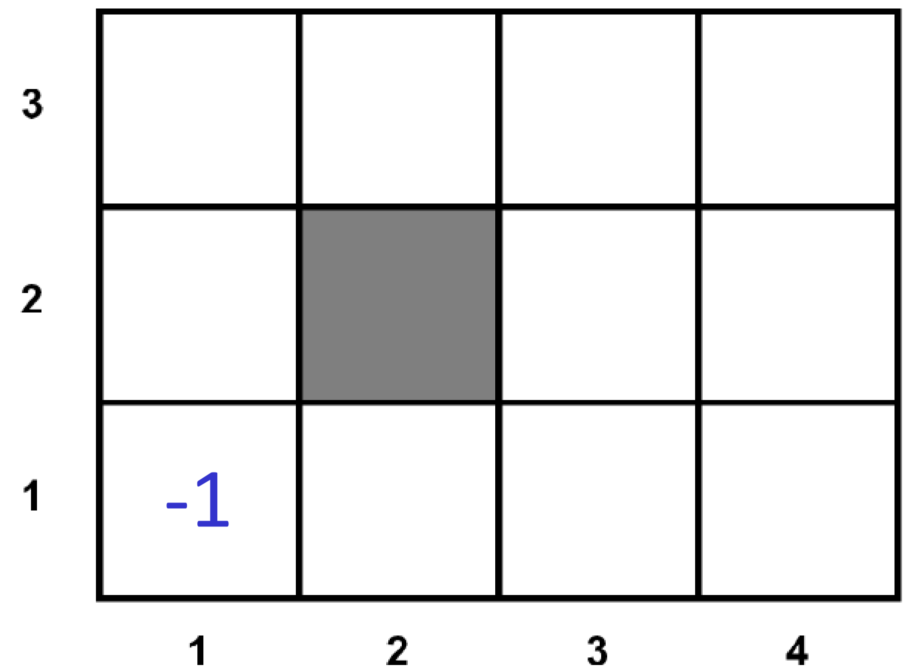
s' → (1,1) up -1

(1,2) up -1



s'=(1,1) nuevo: U(1,1)= -1

s null



Take $\gamma = 1$, $\alpha = 0.5$

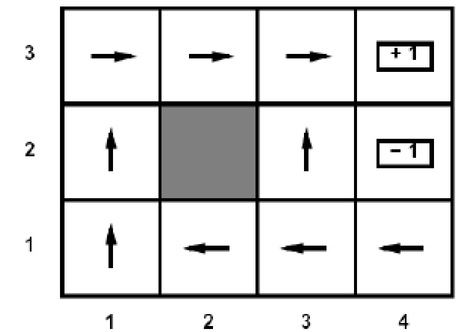
Ejercicio: **1er** paso del primer episodio

Aprendizaje TD: Ejemplo

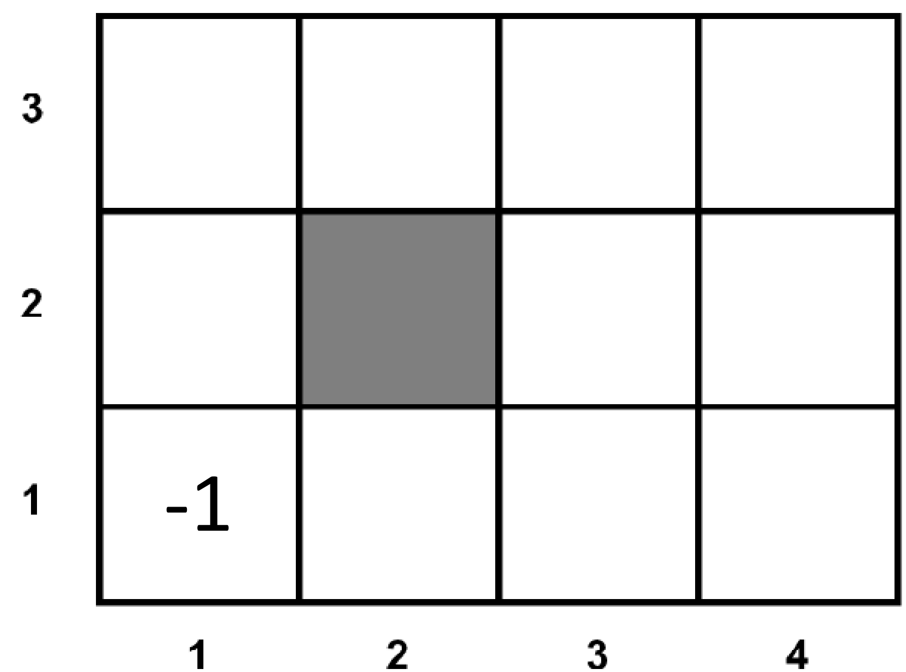
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si $\text{TERMINAL}[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

s →(1,1) up -1

s' →(1,2) up -1



s‘=(1,1) nuevo: U(1,1)= -1
s null
s‘ no terminal: s=(1,1), a=π(1,1)=up, r = -1



Take $\gamma = 1$, $\alpha = 0.5$

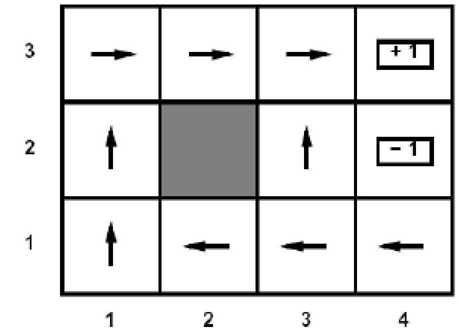
Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

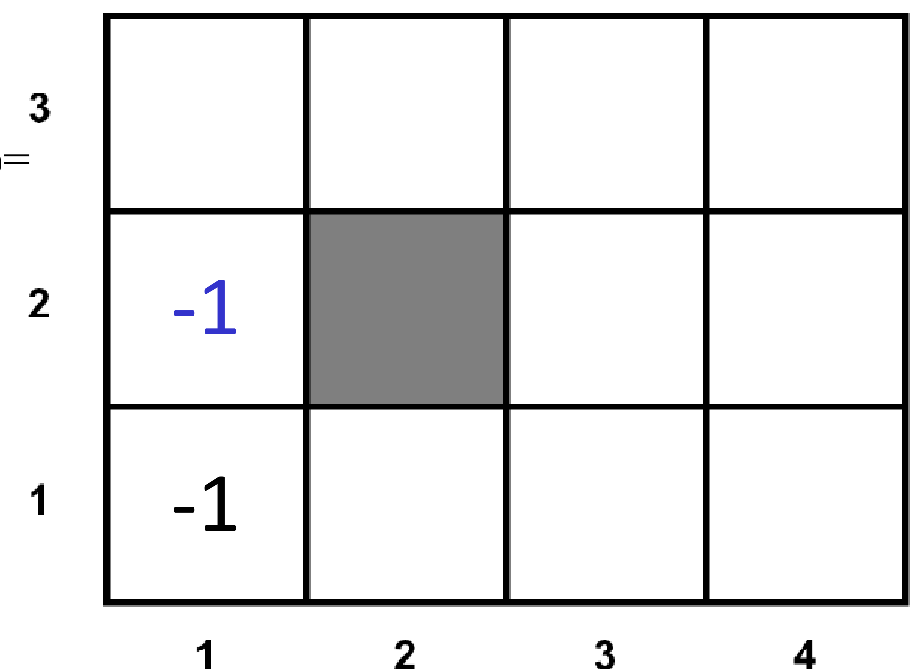
s → (1,1) up -1

s' → (1,2) up -1

(1,2) up -1



s‘=(1,2) nuevo: U(1,2)= -1
s =(1,1) no nulo:
  $N_s(1,1)=1$
  $U(1,1)=U(1,1)+\alpha(N_s(1,1)) \cdot (r+\gamma U(1,2)-U(1,1))=$
  $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 -(-1))=$
  $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$



Take $\gamma = 1$, $\alpha = 0.5$

Ejercicio: **2º** paso del primer episodio

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$

si s no es nulo (null) **entonces hacer**

  incrementar $N_s[s]$

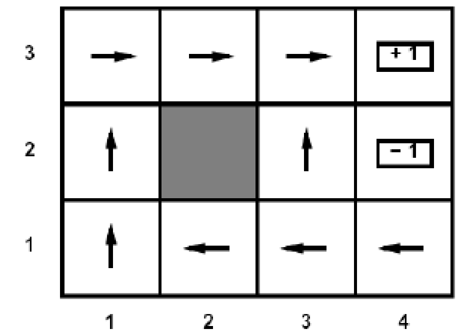
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$

si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

s →(1,2) up -1

s' →(1,2) up -1



s'=(1,2) nuevo: U(1,2)= -1

s =(1,1) no nulo:

  $N_s(1,1)=1$

  $U(1,1)=U(1,1)+\alpha(N_s(1,1)) \cdot (r+\gamma U(1,2)-U(1,1))=$

  $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 -(-1))=$

  $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$

s' no terminal: s=(1,2), a=π(1,2)=up, r = -1

	1	2	3	4
3				
2	-1			
1	-1.5			

Take γ = 1, α = 0.5

Ejercicio: **3er** paso del primer episodio

Aprendizaje TD: Ejemplo

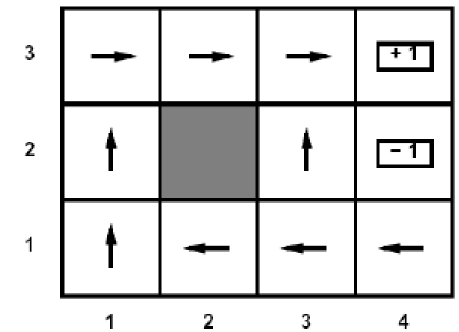
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

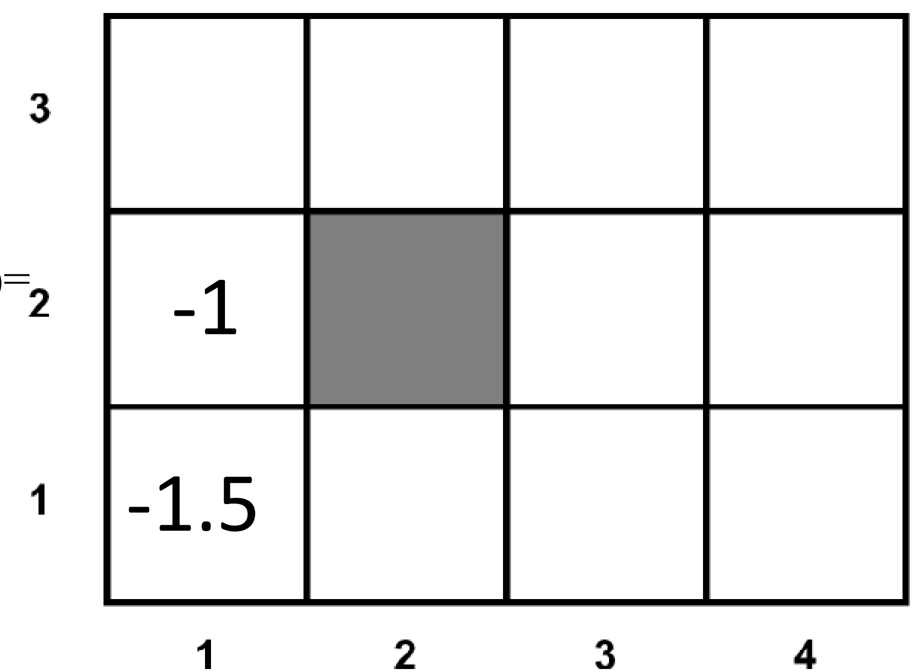
s → (1,2) up -1

s' → (1,2) up -1

(1,3) right -1



s'=(1,2) no nuevo
s =(1,2) no nulo:
  $N_s(1,2)=1$
  $U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,2)-U(1,2))=$
  $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 - (-1))=$
  $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$



Take $\gamma = 1$, $\alpha = 0.5$

Ejercicio: 3er paso del primer episodio

Aprendizaje TD: Ejemplo

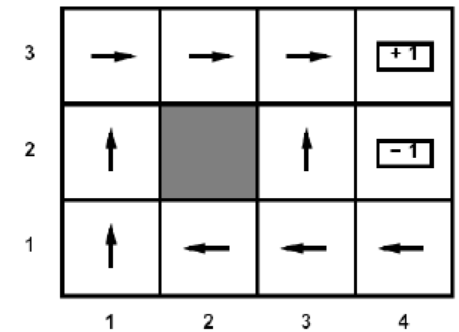
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

(1,2) up -1

s → (1,2) up -1

s' → (1,3) right -1



s‘=(1,2) no nuevo
s =(1,2) no nulo:
  $N_s(1,2)=1$
  $U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,2)-U(1,2))=$
  $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 - (-1))=$
  $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$
s‘ no terminal: s=(1,2), a=π(1,2)=up, r = -1

3				
2	-1.5			
1	-1.5			
	1	2	3	4

Take $\gamma = 1$, $\alpha = 0.5$

Ejercicio: **4º** paso del primer episodio

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si $\text{TERMINAL}[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

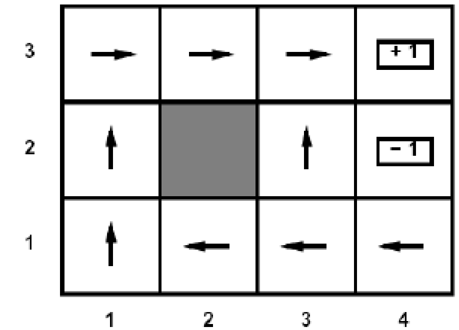
(1,1) up -1

(1,2) up -1

s → (1,2) up -1

s' → (1,3) right -1

(2,3) right -1



s'=(1,3) nuevo U(1,3)=-1
s =(1,2) no nulo:
  $N_s(1,2)=2$
  $U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,3)-U(1,2))=$
  $= -1.5 + 0.5 \cdot (-1 + 1 \cdot -1 - (-1.5))=$
  $= -1.5 + 0.5 \cdot (-1 -1 + 1.5) = -1.5 -0.25 = -1.75$

3	-1			
2	-1.5			
1	-1.5			
	1	2	3	4

Take $\gamma = 1$, $\alpha = 0.5$

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$

si s no es nulo (null) **entonces hacer**

  incrementar $N_s[s]$

  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$

si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

(1,2) up -1

(1,2) up -1

s → (1,3) right -1

s' → (2,3) right -1

	1	2	3	4
3	→	→	→	+1
2	↑		↑	-1
1	↑	←	←	←

s'=(1,3) nuevo U(1,3)=-1

s =(1,2) no nulo:

  $N_s(1,2)=2$

  $U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,3)-U(1,2))=$

  $= -1.5 + 0.5 \cdot (-1 + 1 \cdot -1 - (-1.5))=$

  $= -1.5 + 0.5 \cdot (-1 -1 + 1.5)= -1.5 -0.25= -1.75$

s' no terminal: s=(1,3), a=π(1,3)=right, r = -1

	1	2	3	4
3	-1			
2	-1,75			
1	-1.5			

Take $\gamma = 1$, $\alpha = 0.5$

Ejercicio: 5º paso del primer episodio

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

(1,2) up -1

(1,2) up -1

s → (1,3) right -1

s' → (2,3) right -1

(3,3) right -1

3	→	→	→	+1
2	↑		↑	-1
1	↑	←	←	←
	1	2	3	4

s'=(2,3) nuevo U(2,3)=-1
s =(1,3) no nulo:
  $N_s(1,3)=1$
  $U(1,3)=U(1,3)+\alpha(N_s(1,3)) \cdot (r+\gamma U(2,3)-U(1,3))=$
  $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 - (-1))=$
  $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$

3	-1	-1		
2	-1,75			
1	-1.5			
	1	2	3	4

Ejercicio: **5º** paso del primer episodio

Aprendizaje TD: Ejemplo

si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi[s'], r'$

(1,1) up -1

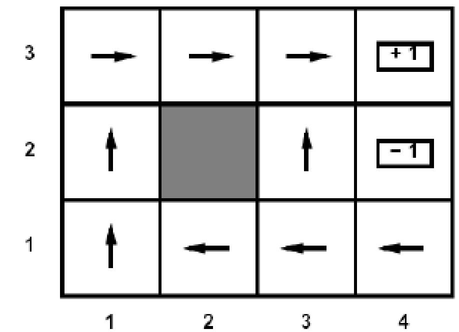
(1,2) up -1

(1,2) up -1

(1,3) right -1

s →(2,3) right -1

s' →(3,3) right -1



	1	2	3	4
3	-1.5	-1		
2	-1,75			
1	-1.5			

s'=(2,3) nuevo U(2,3)=-1
s =(1,3) no nulo:
  $N_s(1,3)=1$
  $U(1,3)=U(1,3)+\alpha(N_s(1,3)) \cdot (r+\gamma U(2,3)-U(1,3))=$
    $= -1 + 0.5 \cdot (-1 + 1 \cdot -1 -(-1))=$
    $= -1 + 0.5 \cdot (-1 -1 + 1)= -1 -0.5= -1.5$
s' no terminal: s=(2,3), a=π(2,3)=right, r = -1

Aprendizaje TD: Inconveniente

- El aprendizaje TD es libre de modelo para aprendizaje pasivo
- No podemos utilizar los valores de los estados obtenidos para generar un política óptima

$$\pi(s) = \arg\max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s') \right]$$

- Quizá podemos aprender los Q-valores directamente y sin modelo...

Links

- Robot que le da la vuelta a un pancake

http://www.youtube.com/watch?v=W_gxLKSsSIE

- Robot autónomo que aprende a caminar:

http://www.youtube.com/watch?v=RZf8fR1SmNY

- Robot que mueve una pala de ping-pong
  http://www.youtube.com/watch?v=CA5IpqNg-hY

Aprendizaje por refuerzo activo

- No conocemos las transiciones T(s,a,s')
- No conocemos las recompensas R(s,a,s')
- Podemos escoger las acciones que queramos
- Objetivo:
  - Aprender la política óptima
- Similar a la iteración de valores o de políticas, pero sin conocer T ni R
- En este caso:
  - El alumno decide qué acciones tomar
  - Tradeoff fundamental: Explotación vs. Exploración
  - La planificación ocurre mientras se está inmerso en el entorno

Rodeo: Iteración de Q-valores

- Iteración de valores: busca aproximaciones sucesivas a los valores óptimos
  - Comenzar con $V_0(s)=0$

$$V_{i+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_i(s')\right]$$

- Pero los Q-valores son más útiles
  - Comenzamos con $Q_0(s,a)=0$

$$Q_{i+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_i(s',a')\right]$$

Q-aprendizaje

- Q-aprendizaje: Iteración de Q-valores basada en muestreo (experiencia).
- Para aprender los valores Q*(s,a):
  - Cada vez que se reciba una muestra (s,a,s',r)
    - Considerar nuestra anterior estimacion Q(s,a)
    - Incorporar la información recibida:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

    - Incoporar la nueva estimación en la media:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\left[sample\right]$$

UNIVERSITAT DE BARCELONA

Consideraciones sobre alfa

- Muestra: [3,5,7,100] $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$

  | [100,7,5,3] Media: 28.75

α	Q_1	Q_2	Q_3	Q_4
Fija α=0.1	0.3 | 10.0	0.77 | 9.7	1.39 | 9.23	11.25 | 8.607
Fija α=0.25	0.75 | 25.0	1.81 | 20.5	3.11 | 16.63	27.33 | 13.22
Fija α=0.5	1.5 | 50.0	3.25 | 28.5	5.13 | 16.75	52.56 | 9.88
Fija α=0.75	2.25 | 75.0	4.31 | 24.0	6.33 | 9.75	76.58 | 4.69
Fija α=1	3 | 100	5 | 7	7 | 5	100 | 3
α=1/N	(α=1) 3.0 | 100.0	(α=1/2) 4.0 | 53.5	(α=1/3) 5.0 | 37.33	(α=1/4) 28.75 28.75

UNIVERSITAT DE BARCELONA

Consideraciones sobre alfa

- Si α=1/N calculemos la muestra $[n_1, n_2, n_3]$:
- $Q_1=(1-1/1)*Q_0 + 1/1\ n_1$
- $Q_2=(1-1/2)*Q_1 + 1/2\ n_2 = 1/2\ n_1 + 1/2\ n_2 = 1/2\ (n_1 + n_2)$
- $Q_3=(1-1/3)*Q_2 + 1/3\ n_3 = (2/3)(1/2)(n_1+n_2)+ 1/3\ n_3 = 1/3\ (n_1 + n_2 +n_3\ )$

Consideraciones sobre alfa

- Prueba por inducción de que Si $\alpha=1/N$ se hace la media: asumimos que la muestra es de N números n_i
- Tomamos $s_n = \Sigma_{(i=1..N)} n_i /N$
- $Q_1=(1-1/1)*Q_0 + 1/1\ n_1 = n_1$
- $Q_2=(1-1/2)*Q_1 + 1/2\ n_2 = 1/2\ n_1 + 1/2\ n_2 = 1/2\ (n_1 + n_2)$
- $Q_{N+1}= (1-1/(N+1))*Q_N + 1/(N+1)\ n_{N+1} =$
  $= (N+1-1)/(N+1))(\Sigma_{(i=1..N)} n_i /N)+ 1/(N+1)\ n_{N+1} =$
  $= (N/(N+1))(\Sigma_{(i=1..N)} n_i /N)+ n_{N+1} /(N+1) =$
  $= (1/(N+1))(\Sigma_{(i=1..N)} n_i)+ n_{N+1} /(N+1)=$
  $= (1/(N+1))((\Sigma_{(i=1..N)} n_i)+ n_{N+1})=$
  $= \Sigma_{(i=1..N+1)} n_i /(N+1)$

Q-aprendizaje
(libro Russell and Norvig 4th Ed)

Chapter 23 Reinforcement Learning

function Q-LEARNING-AGENT(*percept*) **returns** an action
  inputs: *percept*, a percept indicating the current state s' and reward signal r
  persistent: Q, a table of action values indexed by state and action, initially zero
    N_{sa}, a table of frequencies for state–action pairs, initially zero
    s, a, the previous state and action, initially null

  if s is not null **then**
    increment $N_{sa}[s,a]$
    $Q[s,a] \leftarrow Q[s,a] + \alpha(N_{sa}[s,a])(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$
  $s, a \leftarrow s', \text{argmax}_{a'} f(Q[s',a'], N_{sa}[s',a'])$
  return a

Figure 23.8 An exploratory Q-learning agent. It is an active learner that learns the value $Q(s,a)$ of each action in each situation. It uses the same exploration function f as the exploratory ADP agent, but avoids having to learn the transition model.

Q-learning
(e-libro Sutton and Barto)

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
    Initialize s
    Repeat (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

Figure 6.12: Q-learning: An off-policy TD control algorithm.

Propiedades del Q-learning

- Resultados sorprendentes: Q-aprendizaje converge a la política óptima
  - Si exploras bastante
  - Si la ratio de aprendizaje es suficientemente baja
  - Básicamente es idependiente de cómo se seleccionan las acciones.

Exploración / Explotación

- Existen diferentes esquemas para forzar la exploración
  - El más simple, la selección aleatoria de acciones (ε voraz):
    - Lanzamos una moneda antes de elegir qué acción realizar.
    - Con probabilidad 1-ε escogemos la mejor opción según los Q-valores actuales.
    - Con probabilidad ε escogemos una acción al azar.
  - Problemas de las acciones aleatorias:
    - Exploramos todo el espacio pero lo seguimos haciendo una vez que ya hemos aprendido.
    - Solución: disminuir ε con el tiempo.
    - Otra solución: funciones de exploración (ej. Intentar acciones que se han probado poco evitando acciones que parezcan tener poca utilidad)

Funciones de exploración

- Cuándo explorar:
  - Acciones aleatorias: podemos ir a parar a acciones que ya sepamos que son malas
  - Mejor idea: explorar áreas de las que aún no tenemos información
- Función de exploración:
  - Recibe una estimación del valor del estado y un contador del número de veces que hemos estado.
  - f(u,n) = u+k/n

$$Q_{i+1}(s,a) \leftarrow_\alpha R(s,a,s') + \gamma \max_{a'} Q_i(s',a')$$

$$Q_{i+1}(s,a) \leftarrow_\alpha R(s,a,s') + \gamma \max_{a'} f(Q_i(s',a'), N(s',a'))$$

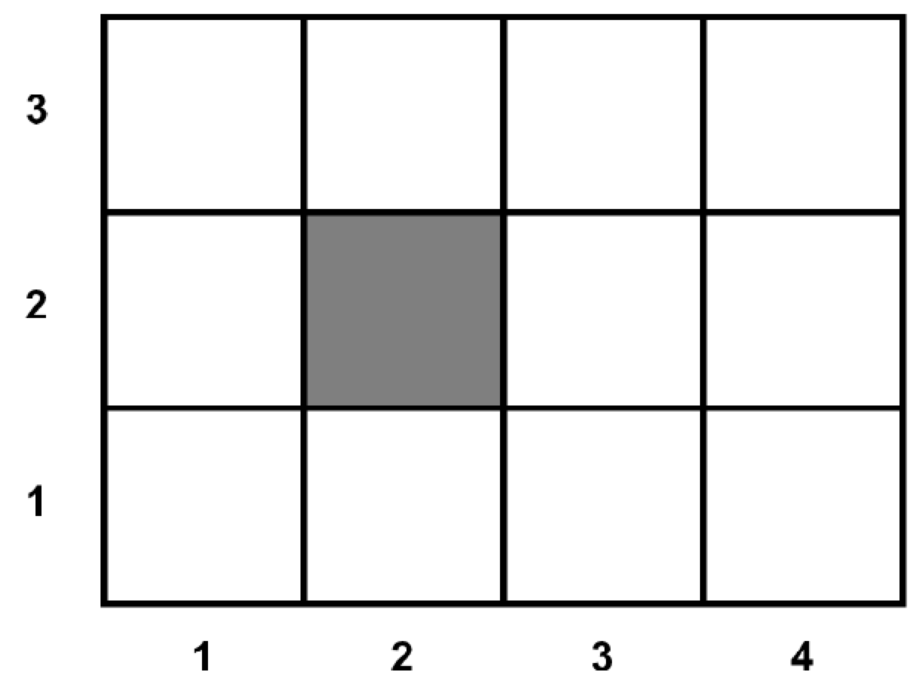
Q-learning: Ejemplo simplificado

Hacer las 5 primeras iteraciones de Q-learning considerando estado inicial (1,1), $\gamma=1$, $\alpha=1$, y donde a cada iteración se nos da la acción elegida, el estado al que se llega tras hacer la acción y la recompensa puntual obtenida. **$\gamma=1$, $\alpha=1$ por simplificar, $\gamma<1$, $\alpha=1/N$ para converger**

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

$\gamma=1$, $\alpha=1$

(1,1)



Q-learning: Ejemplo simplificado

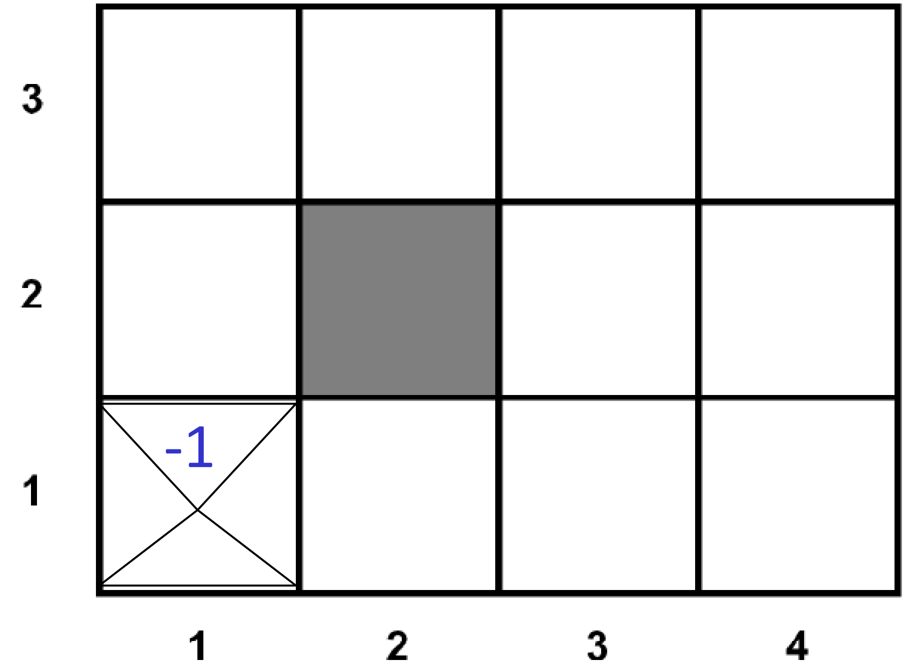
$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r + \gamma \max a'\ Q(s', a') - Q(s, a)\ ]$ $\gamma = 1,\ \alpha = 1$

1ª iteración: acción elegida a=up, estado al que se llega s'=(1,2), recompensa obtenida r=-1

s →(1,1) up -1

s' →(1,2)

s=(1,1), s'=(1,2) Q((1,1),up)=0+[-1+0-0]=-1



Q-learning: Ejemplo simplificado

$Q(s,a) \leftarrow Q(s,a) + \alpha\ \mathbf{[r + \gamma\ max\ a'\ Q(s', a') - Q(s, a)}\ ]$ $\gamma = 1,\ \alpha = 1$

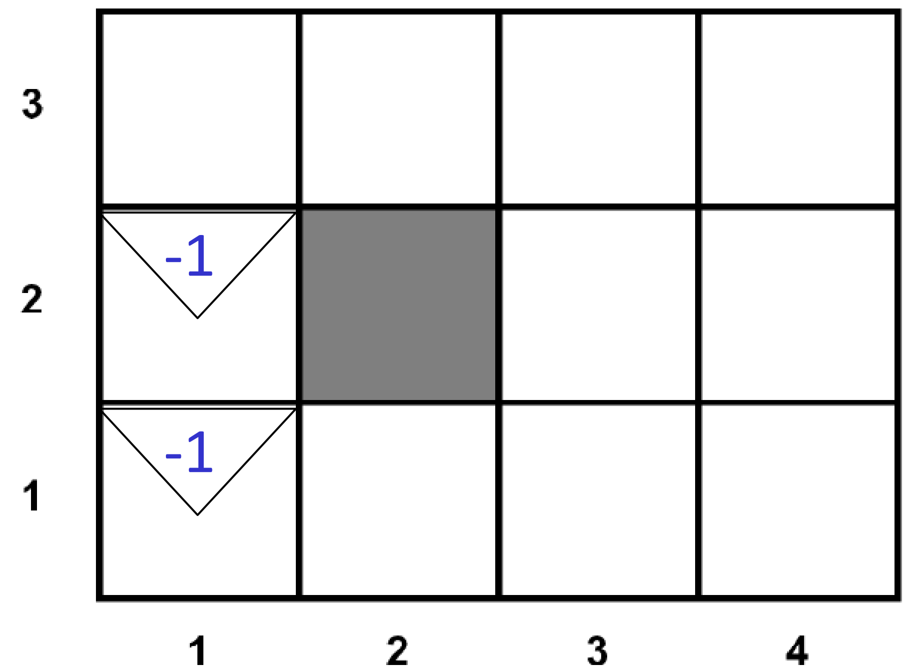
2ª iteración: acción elegida a=up, estado al que se llega s'=(1,2), recompensa obtenida r=-1

(1,1) up -1

s →(1,2) up -1

s' →(1,2)

s=(1,2), s'=(1,2) Q((1,2),up)=0+[-1+0-0]=-1



Q-learning: Ejemplo simplificado

Q(s,a) ← Q(s,a) + α **[r + γ max a' Q(s', a') – Q(s, a)**]   γ =1, α=1

3ª iteración: acción elegida a=up, estado al que se llega s'=(1,3), recompensa obtenida r=-1

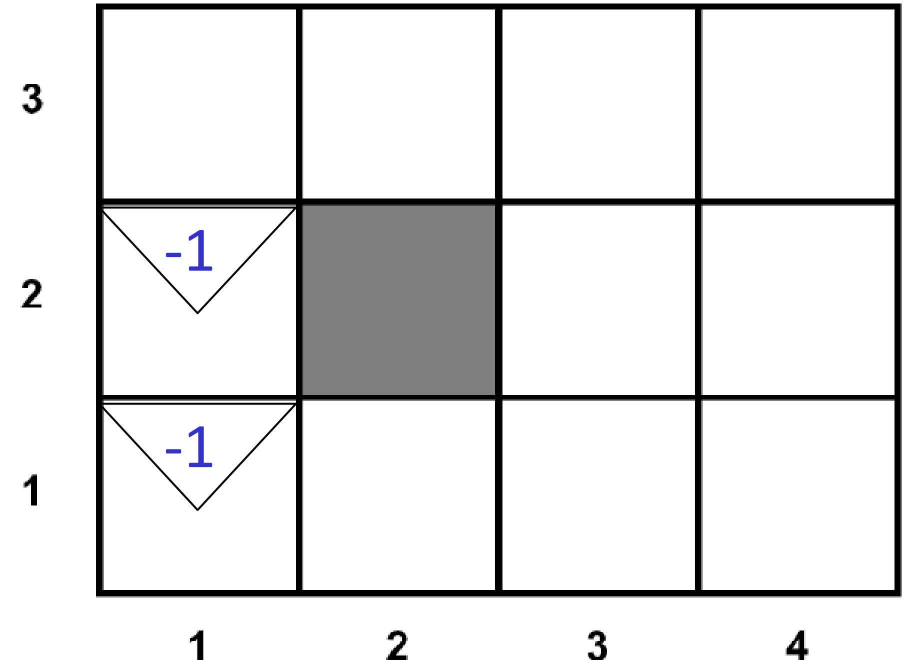
(1,1) up -1

(1,2) up -1

s →(1,2) up -1

s' →(1,3)

s=(1,2), s'=(1,3) Q((1,2),up)=-1+[-1+0+1]=-1



Q-learning: Ejemplo simplificado

$Q(s,a) \leftarrow Q(s,a) + \alpha \, [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ $\gamma = 1, \alpha = 1$

4ª iteración: acción elegida a=right, estado al que se llega s'=(2,3), recompensa obtenida r=-1

(1,1) up -1

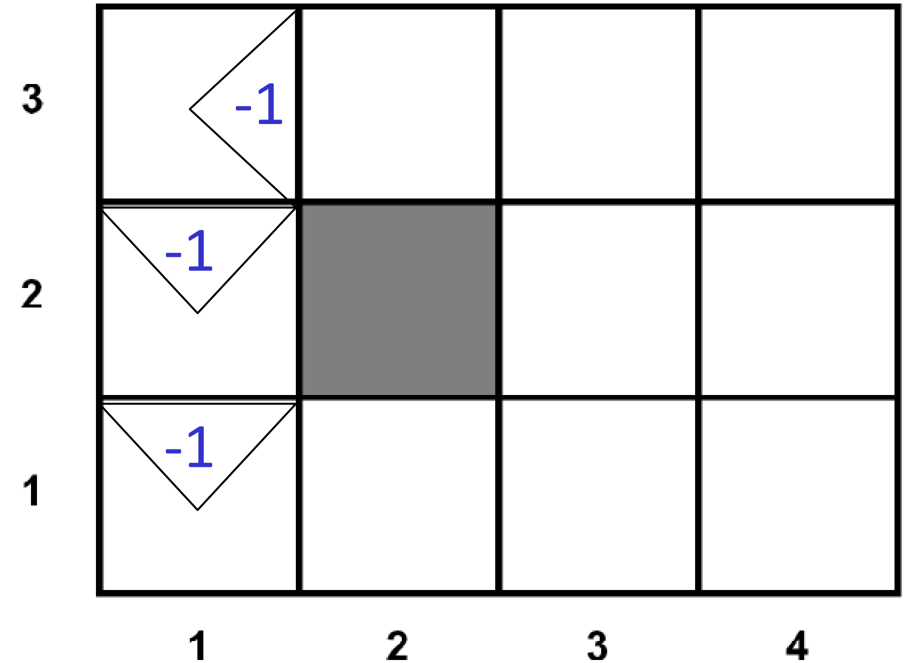
(1,2) up -1

(1,2) up -1

s → (1,3) right -1

s' → (2,3)

s=(1,3), s'=(2,3) Q((1,3),right)=0+[-1+0-0]=-1



Q-learning: Ejemplo simplificado

$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ $\gamma = 1,\ \alpha = 1$

5ª iteración: acción elegida a=right, estado al que se llega s'=(3,3), recompensa obtenida r=-1

(1,1) up -1

(1,2) up -1

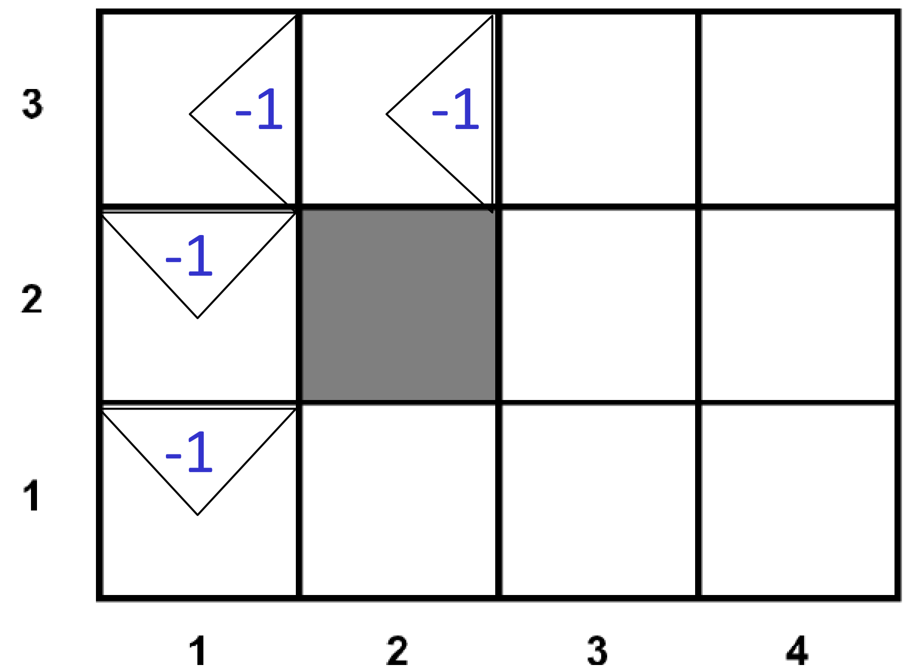
(1,2) up -1

(1,3) right -1

s →(2,3) right -1

s' →(3,3)

s=(2,3), s'=(3,3) Q((2,3),right)=0+[-1+0-0]=-1



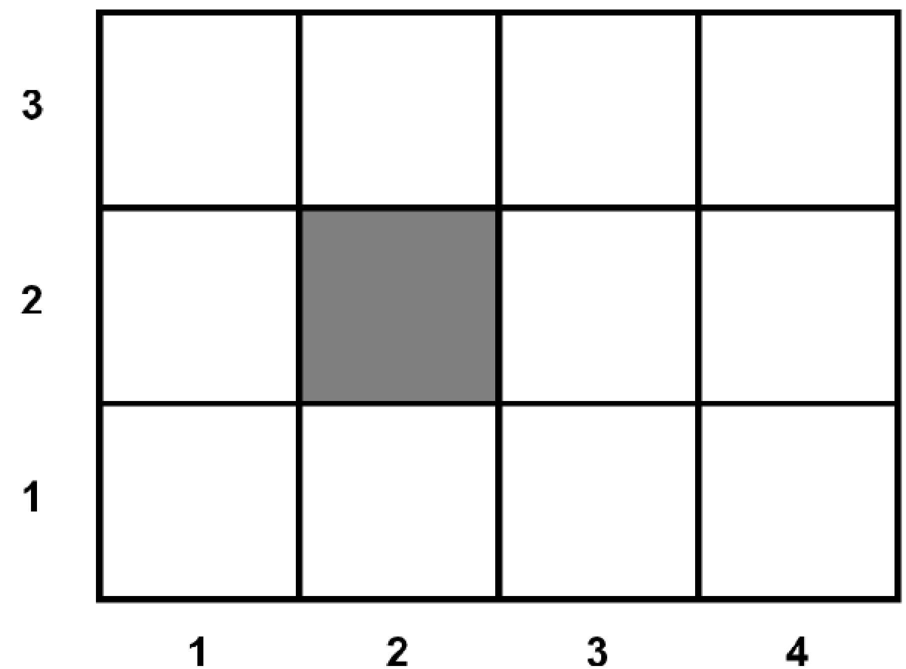
Q-learning: Ejercicio

Ejemplo con estado inicial (1,1), γ =0.9, α=0.1, pero a cada iteración elegimos una acción en función de ε=0.4, los valores de Q(s,a) y las veces que se han visitado y vemos el estado al que se llega tras hacer la acción y la recompensa puntual obtenida.

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

γ =0.9, α=0.1, ε =0.4

(1,1)



Q-learning: Ejercicio

$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r + \gamma \max_{a'} Q(s', a') - Q(s, a)\ ]$ $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.4$

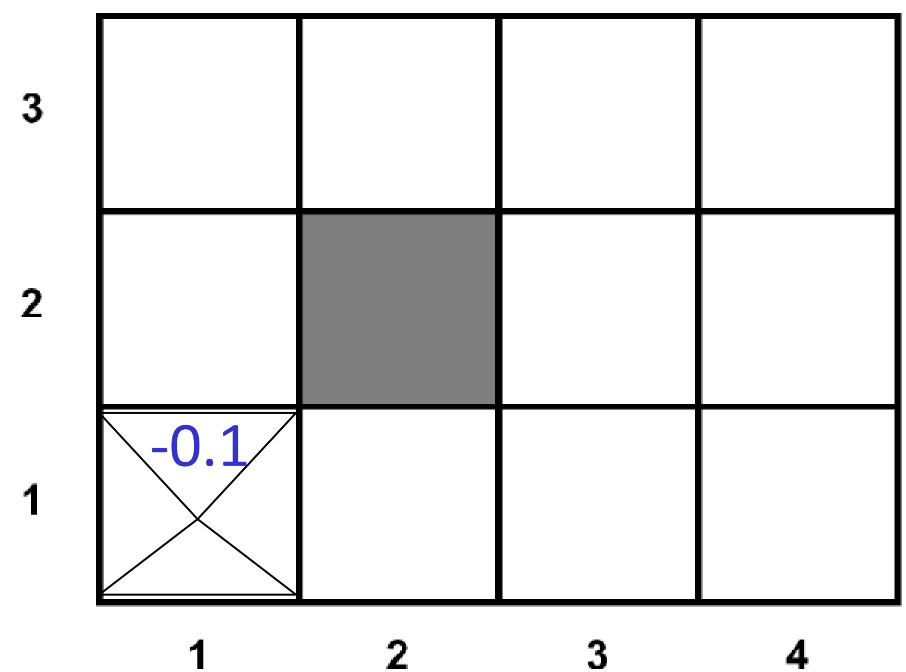
1ª iteración: estado actual s=(1,1)

s → (1,1), up, -1

s' → (1,2)

s=(1,1), s'=(1,2)
Q((1,1),up)=0+0.1*[-1+0-0]=-0.1

- Calcular Aleatorio(0..1)=0.6 > ε =0.4 sale explotación, dado que $\forall a \in$ {up,down, right, left} tenemos que Q(s, a)=0 , $N_{s,a}=0$ elegimos la acción de forma aleatoria: a=up
- Estado al que se llega: s'=(1,2)
- Recompensa obtenida: r=-1
- Ahora: Q((1,1),up)=-0.1, y $N_{(1,1),up}=1$



Q-learning: Ejercicio

$Q(s,a) \leftarrow Q(s,a) + \alpha\,[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.4$

2ª iteración: estado actual s=(1,2)

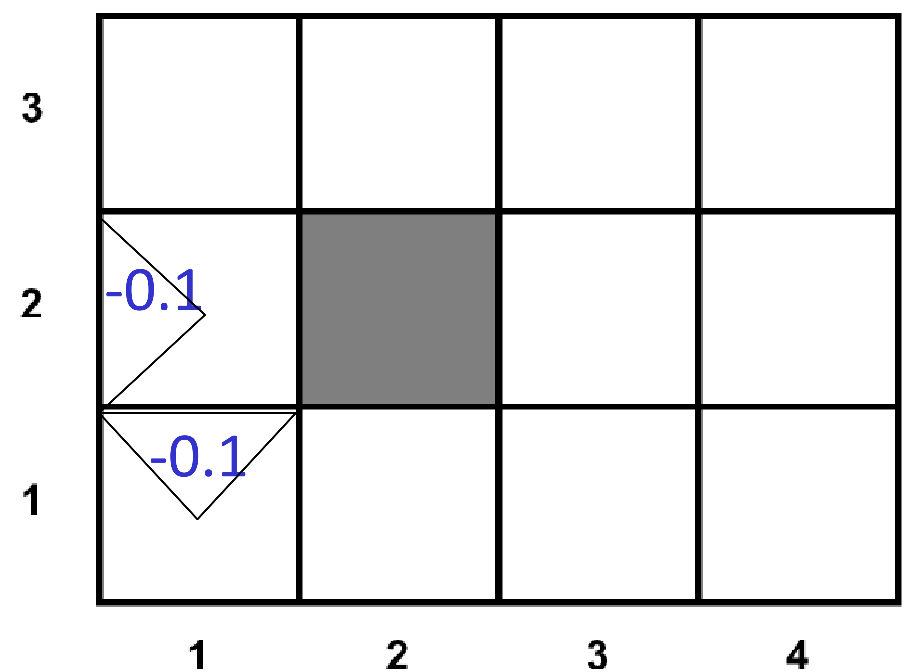
(1,1), up, -1

s → (1,2), left, -1

s' → (1,2)

- Calcular Aleatorio(0..1)=0.8 > ε =0.4 sale explotación, dado que $\forall a \in$ {up,down, right, left} tenemos que Q(s, a)=0 , $N_{s,a}=0$ elegimos la acción de forma aleatoria: a=left
- Estado al que se llega: s'=(1,2)
- Recompensa obtenida: r=-1
- Ahora: Q((1,2),left)=-0.1, y $N_{(1,2),left}=1$

s=(1,2), s'=(1,2)
Q((1,2),left)=0+0.1*[-1+0-0]=-0.1



UNIVERSITAT DE BARCELONA

Q-learning: Ejercicio

$Q(s,a) \leftarrow Q(s,a) + \alpha\ [\mathbf{r + \gamma \max a'\ Q(s', a') - Q(s, a)}\ ]$ $\gamma = 0.9,\ \alpha = 0.1,\ \varepsilon = 0.4$

3ª iteración: estado actual s=(1,2)

(1,1), up, -1

(1,2), left, -1

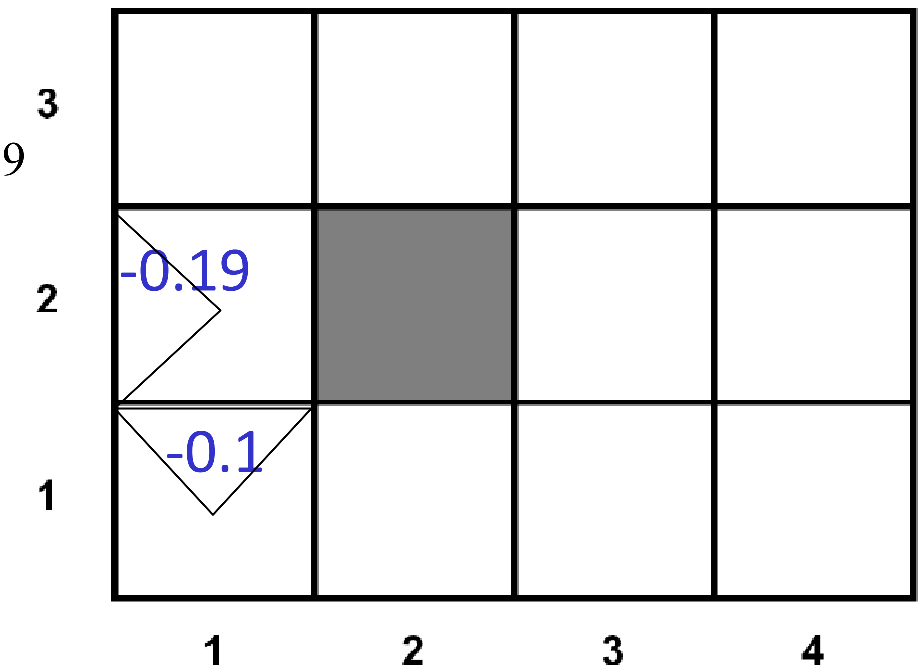
s → (1,2), left, -1

s' → (1,3)

- Calcular Aleatorio(0..1)=0.3 < ε =0.4 sale exploración, elegimos la acción de forma aleatoria entre a$\in${up,down, right, left}: a=left
- Estado al que se llega s'=(1,3)
- Recompensa obtenida r=-1
- Ahora Q((1,2),left)=-0.19, y $N_{(1,2),left}=2$

s=(1,2), s'=(1,3)

Q((1,2),left)=-0.1+0.1*[-1+0+0.1]= -0.1+0.1* [-0.9]=-0.19



Notar que si calcular Aleatorio(0..1)=0.7 > ε =0.4 sale explotación, entonces $\forall$a$\in${up, down, right} tenemos que Q(s, a)=0 , $N_{s,a}=0$
pero Q(s,left)=-0.1, y $N_{s,left}=1$ y por tanto elegimos una acción de entre las acciones para las que Q es mayor (up, down, right) considerando también las frequencias $N_{s,a}$

UNIVERSITAT DE BARCELONA

Q-learning: Ejercicio

$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r + \gamma \max_{a'} Q(s', a') - Q(s, a)\ ]$ $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.4$

4ª iteración: estado actual s=(1,3)

(1,1), up, -1

(1,2), left, -1

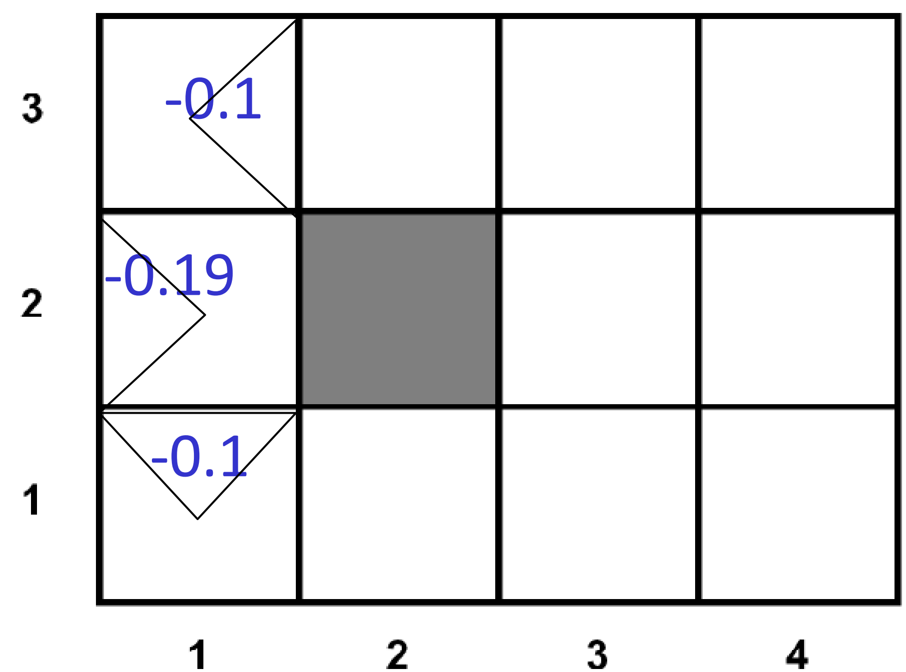
(1,2), left, -1

s → (1,3), right, -1

s' → (2,3)

s=(1,3), s'=(2,3)
Q((1,3),right)=0+0.1*[-1+0-0]=-0.1

- Calcular Aleatorio(0..1)=0.5 > ε =0.4 sale explotación, dado que $\forall a \in$ {up,down, right, left} tenemos que Q(s, a)=0 , $N_{s,a}=0$
elegimos la acción de forma aleatoria: a=right
- Estado al que se llega: s'=(2,3)
- Recompensa obtenida: r=-1
- Ahora: Q((1,3),right)=-0.1, y $N_{(1,3),right}=1$



Q-learning: Ejercicio

$Q(s,a) \leftarrow Q(s,a) + \alpha\, [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ $\gamma = 0.9$, $\alpha = 0.1$, $\varepsilon = 0.4$

5ª iteración: estado actual s=(2,3)

(1,1), up, -1

(1,2), left, -1

(1,2), left, -1

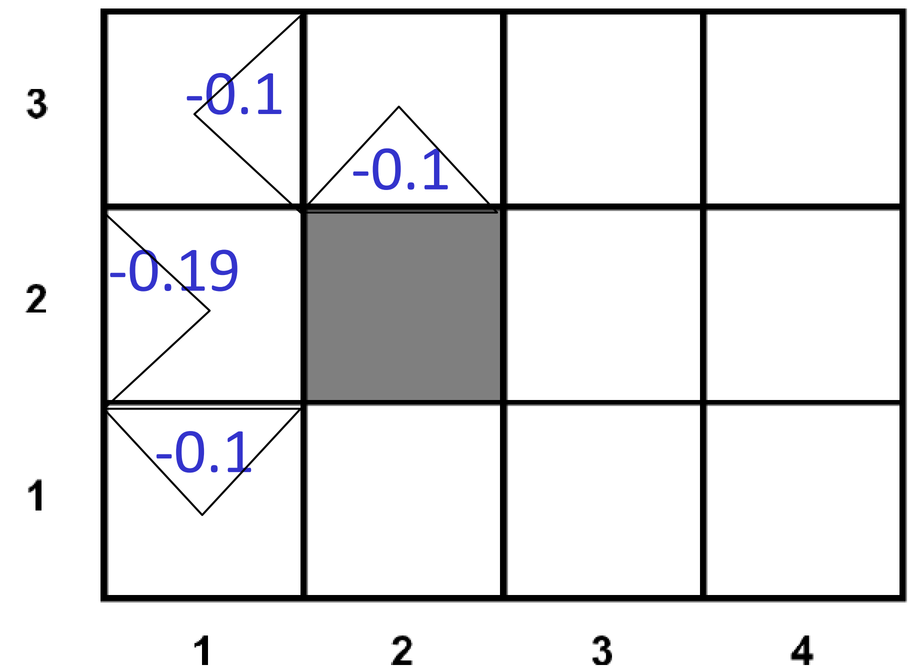
(1,3), right, -1

s → (2,3), down, -1

s' → (2,3)

s=(2,3), s'=(2,3)
Q((2,3),down)=0+0.1*[-1+0-0]=-0.1

- Calcular Aleatorio(0..1)=0.1> ε =0.4 sale exploración, elegimos la acción de forma aleatoria entre a$\in${up,down, right, left}: a=down
- Estado al que se llega s'=(2,3)
- Recompensa obtenida r=-1
- Ahora Q((2,3),down)=-0.1, y $N_{(2,3),down}$=1



Recapitulemos: Aprendizaje

- Cosas que sabemos hacer:
  - Resolver pequeños MDPs exactamente, offline
  - Estimar Q*(s,a) para la política óptima ejecutando una política de exploración

- Técnicas:
  - Iteración de valores
  - Q-aprendizaje
  - Selección exploratoria de acciones