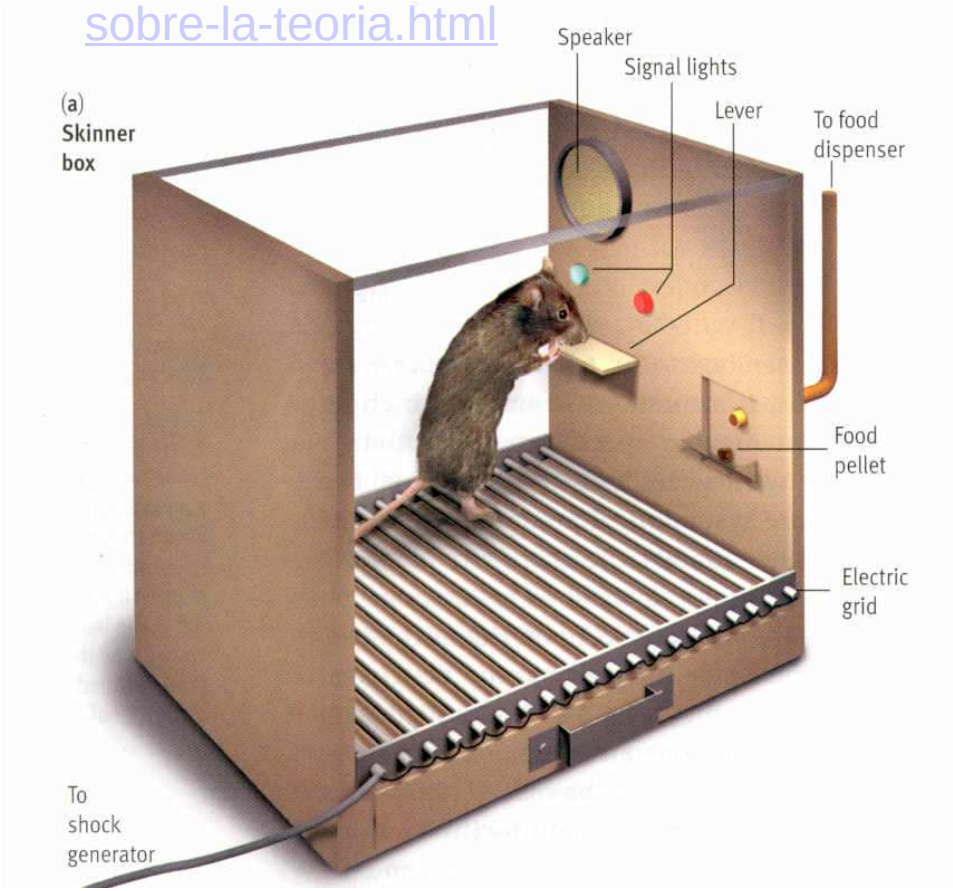
Aprendizaje por refuerzo (II)

RL en otras disciplinas

- El aprendizaje por refuerzo se ha estudiado en psicología desde hace más de 60 años.
  - Educación: http://aprendizajeducacion.blogspot.com/2011/03/exposicion-sobre-la-teoria.html



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

Ejemplo: Aprendizaje animal

- Ejemplo: comida
  - Las abejas terminan aprendiendo planes de comida óptimos 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.

Aprendizaje por refuerzo

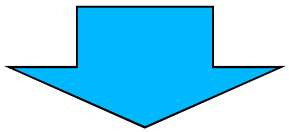
- Aprendizaje por refuerzo
- Aprendizaje por refuerzo pasivo
  - Estimación directa de la utilidad
  - Aprendizaje basado en modelo
  - Aprendizaje basado en diferencia temporal
- Aprendizaje por refuerzo activo
  - Q-aprendizaje
  - Selección explorativa de acciones

Aprendizaje por refuerzo

- Seguimos modelando nuestro mundo mediante un PDM:
  - S - conjunto de estados
  - A - conjunto de acciones
  - T: S x A x S $\rightarrow \Re$ - función de transición
  - R: S x A x S $\rightarrow \Re$ – 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)
- Similar a la evaluación de políticas, pero sin conocer T ni R.



¿Cómo evaluamos V?

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)
- Similar a la evaluación de políticas, pero sin conocer T ni R.

¿Cómo evaluamos V?

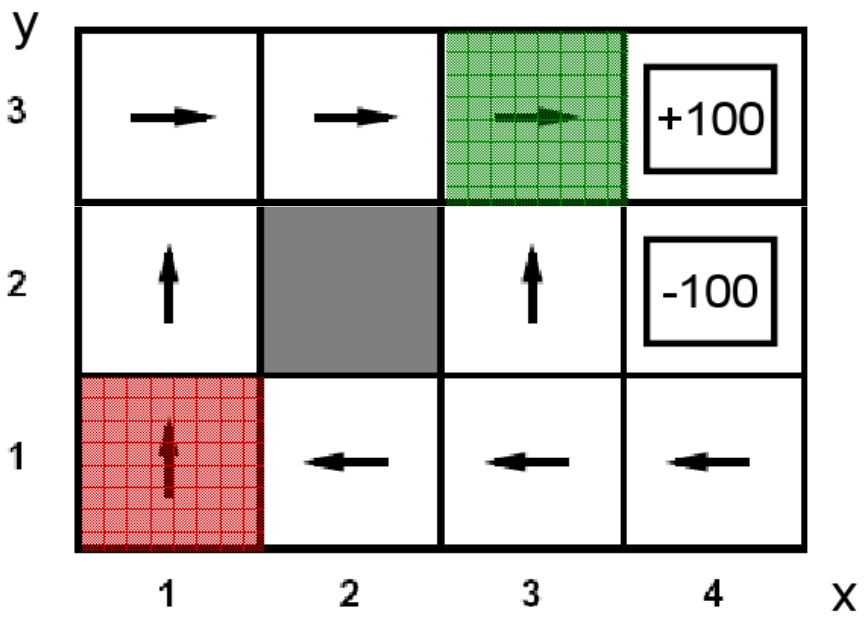
a partir de la experiencia

Estimación directa de la utilidad

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

- Episodes:

(1,1) up -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) ?

V(3,3) ?

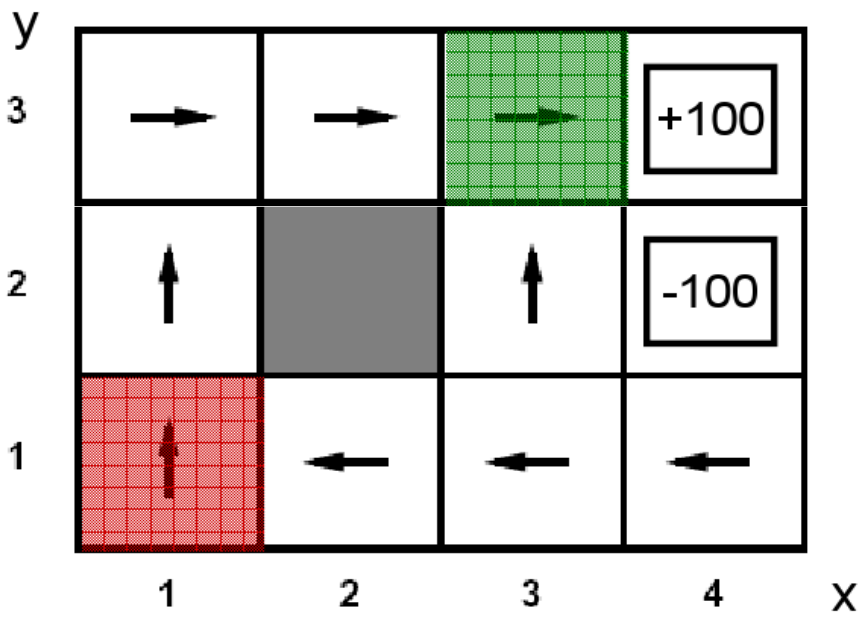
A l hacer estos dos recorridos ¿Qué experiencia/recompensa se ha acumulado sobre cada uno de estos estados?

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

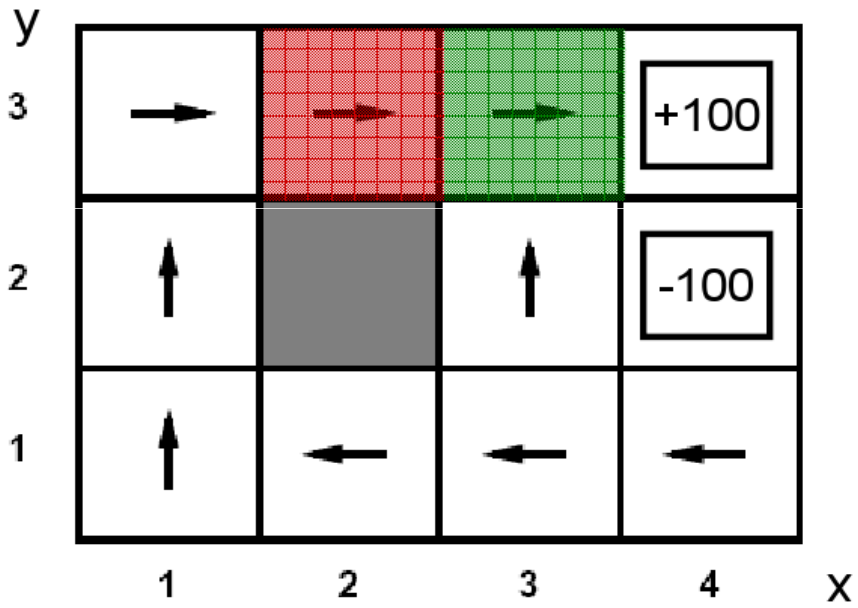
- Alternativa:
  - Estimar el MDP a partir de las observaciones
  - Determinar el valor de cada estado resolviendo el MDP estimado
- Caso más sencillo
  - Contar los estados a los que se llega para cada s,a
  - Estimar T(s,a,s')
  - Descubrir R(s,a,s') cada vez que ocurra

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>) ?

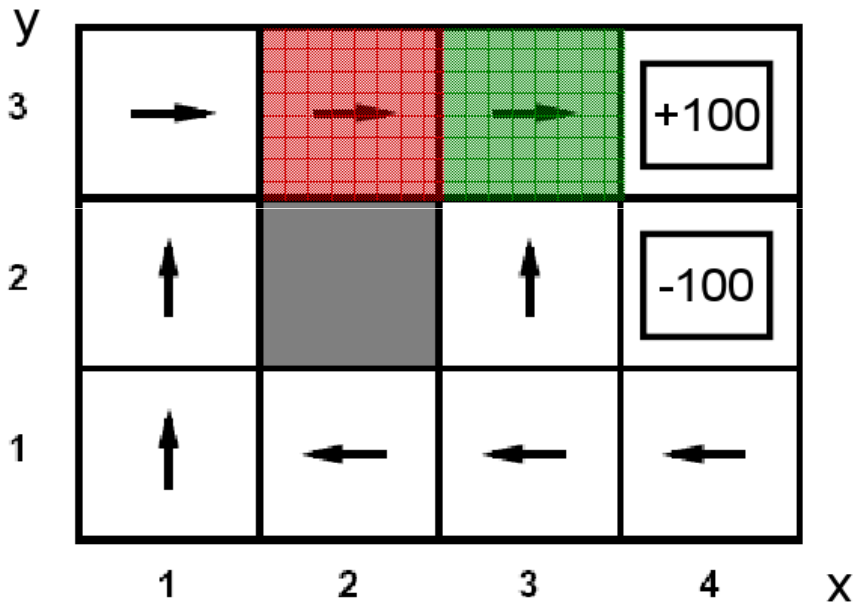
T(<2,3>, right, <3,3>) ?

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

Aprendizaje basado en modelo

- Episodes:

(1,1) up -1	(1,1) 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>) = 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)*

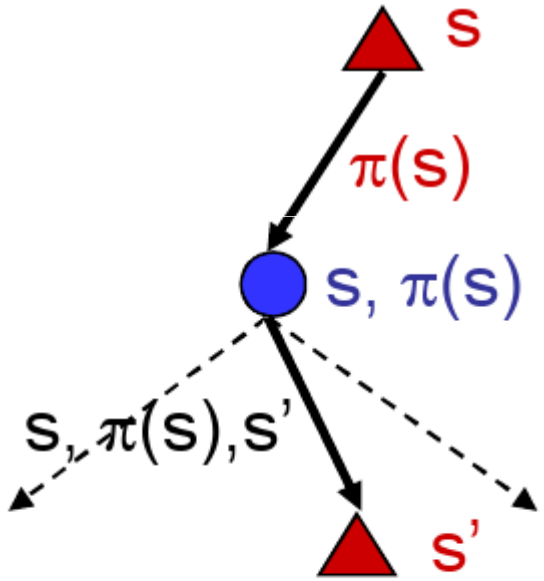
  si s' es nuevo **entonces hacer** $U[s'] \leftarrow r'$; $R[s'] \leftarrow r'$
  si s no es nulo (null) **entonces hacer**
    incrementar $N_{sa}[s, a]$ y $N_{sas'}[s, a, s']$
    para cada t tal que $N_{sas'}[s, a, t]$ no sea cero **hacer**
      $T[s, a, t] \leftarrow N_{sas'}[s, a, t] / N_{sa}[s, a]$
  $U \leftarrow$ 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

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.



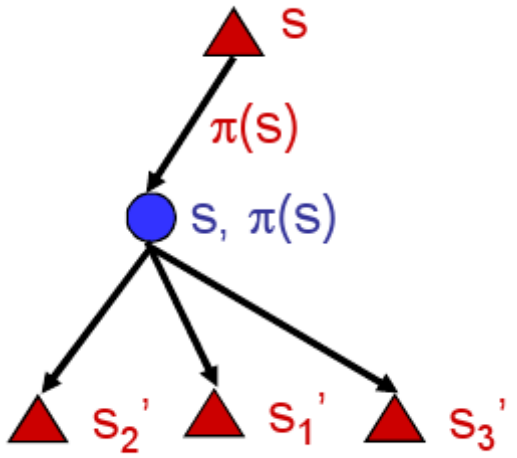
$$V_0^{\pi}(s) = 0$$

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

¿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



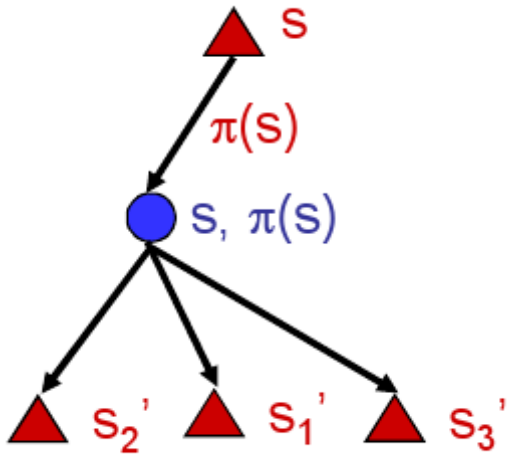
$$sample_1 = R(s, \pi(s), s'_1) + \gamma V_i^{\pi}(s'_1)$$
$$sample_2 = R(s, \pi(s), s'_2) + \gamma V_i^{\pi}(s'_2)$$
$$\dots$$
$$sample_k = R(s, \pi(s), s'_k) + \gamma V_i^{\pi}(s'_k)$$

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

¿Podemos reemplazar la esperanza con la media?

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

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



$$sample_1 = R(s, \pi(s), s'_1) + \gamma V^{\pi}_{i}(s'_1)$$
$$sample_2 = R(s, \pi(s), s'_2) + \gamma V^{\pi}_{i}(s'_2)$$
$$\dots$$
$$sample_k = R(s, \pi(s), s'_k) + \gamma V^{\pi}_{i}(s'_k)$$

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

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

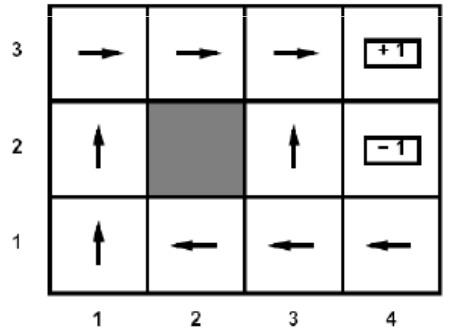
α : factor de aprendizaje

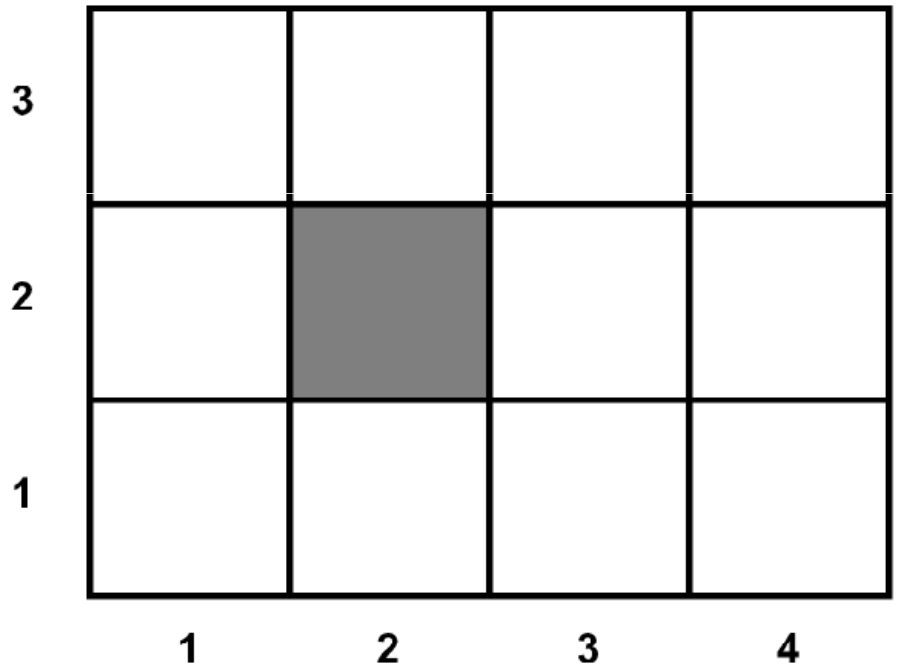
Aprendizaje TD: Ejemplo

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	(1,1) 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)	





$\gamma = 1$, $\alpha(n) = 1/n$

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

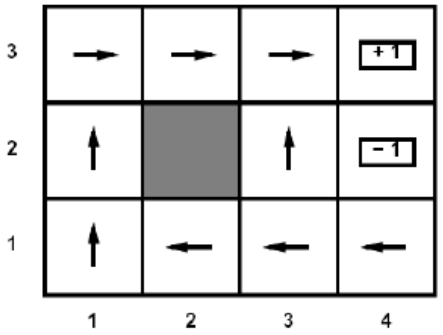
Aprendizaje TD: Ejemplo

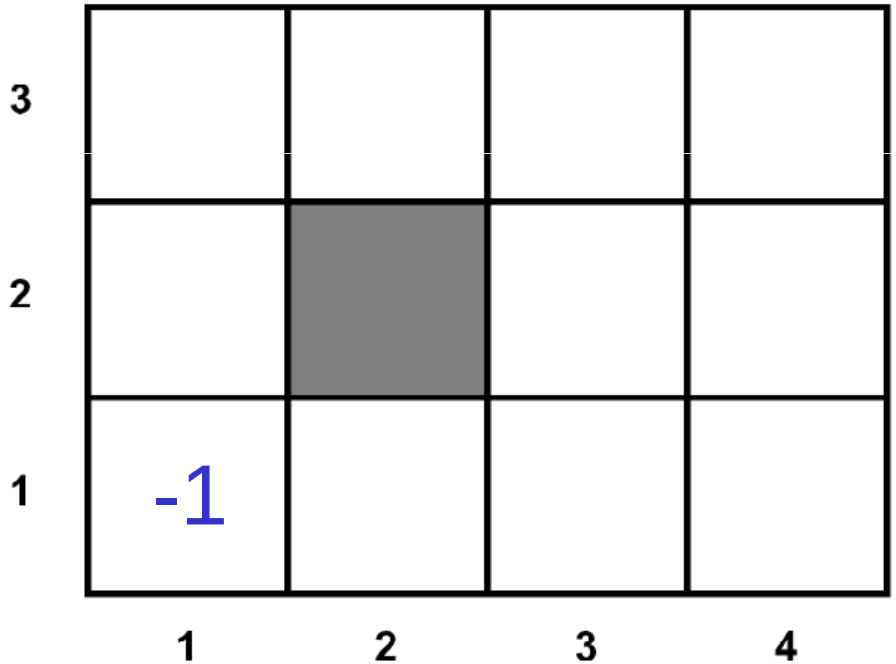
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi\,[s'], r'$

s' → (1,1) up -1

(1,2) up -1

s‘=(1,1) nuevo: U(1,1)= -1
s null





$\gamma = 1$, $\alpha(N_s[s]) = 1/N_s[s]$

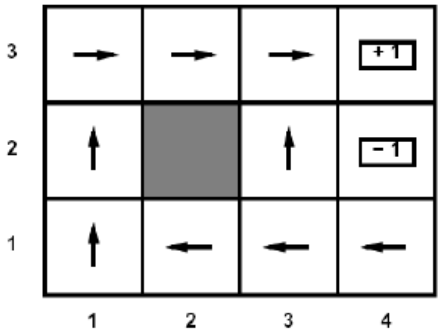
Ejercicio: **1er** paso del primer episodio

Aprendizaje TD: Ejemplo

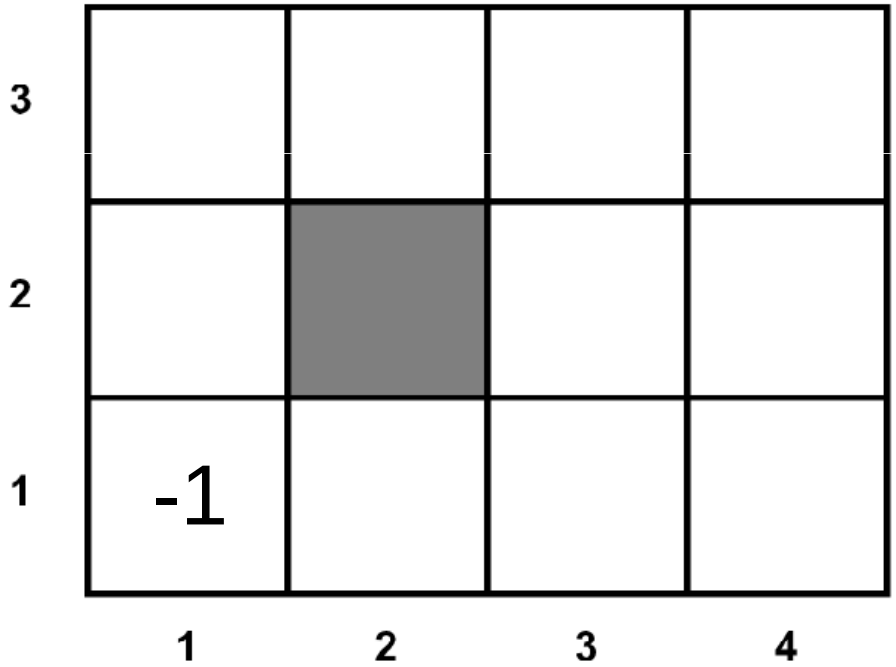
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi\,[s'], r'$

s →(1,1) up -1

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



$\gamma = 1$, $\alpha(N_s[s]) = 1/N_s[s]$

Ejercicio: 2º paso del primer episodio

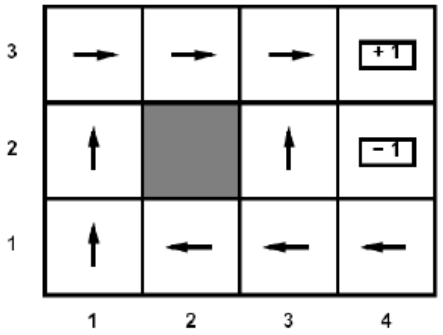
Aprendizaje TD: Ejemplo

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** $s, a, r \leftarrow s', \pi\,[s'], r'$

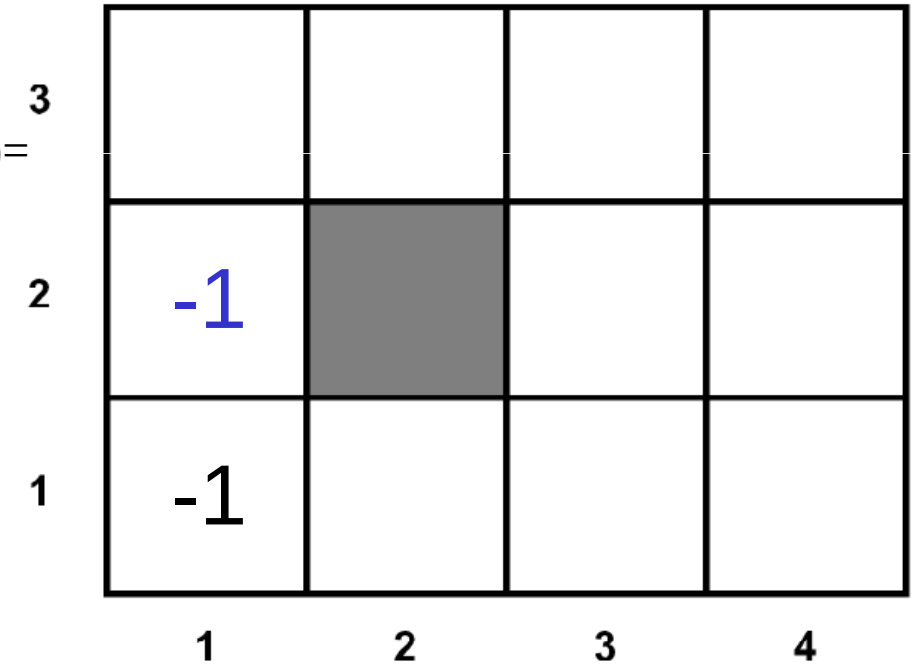
s →(1,1) up -1

s' →(1,2) up -1

(1,2) up -1



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



$\gamma = 1$, $\alpha(N_s[s])=1/N_s[s]$

Ejercicio: 2º paso del primer episodio

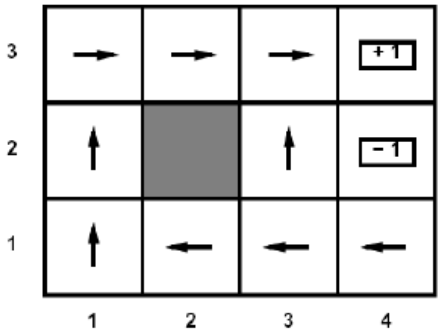
Aprendizaje TD: Ejemplo

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** $s, a, r \leftarrow s', \pi\,[s'], r'$

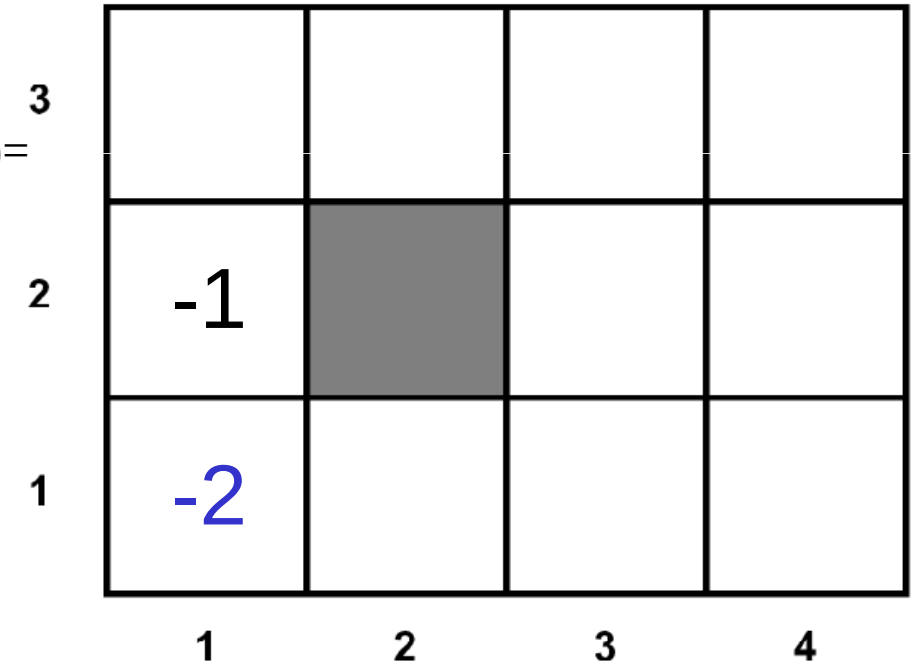
(1,1) up -1

s →(1,2) up -1

s' →(1,2) up -1



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



$\gamma = 1$, $\alpha(N_s[s])=1/N_s[s]$

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

Aprendizaje TD: Ejemplo

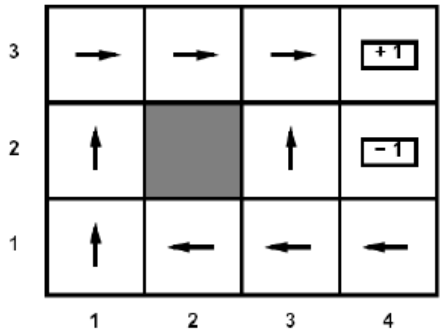
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi\,[s'], r'$

(1,1) up -1

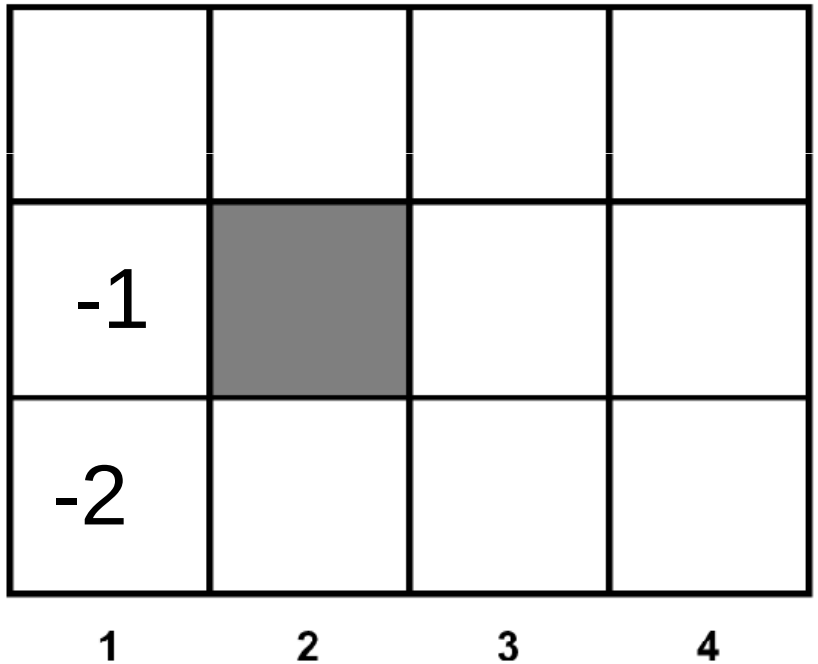
s →(1,2) up -1

s' →(1,2) up -1

(1,3) right -1



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



$\gamma = 1$, $\alpha(N_s[s])=1/N_s[s]$

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

Aprendizaje TD: Ejemplo

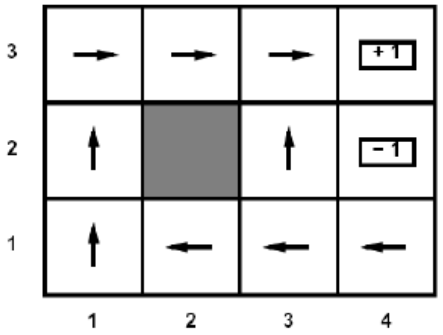
si s' es nuevo **entonces** $U[s'] \leftarrow r'$
si s no es nulo (null) **entonces hacer**
  incrementar $N_s[s]$
  $U[s] \leftarrow U[s] + \alpha(N_s[s])(r + \gamma\, U[s'] - U[s])$
si TERMINAL$[s']$ **entonces** $s, a, r \leftarrow$ nulo (null) **si no** $s, a, r \leftarrow s', \pi\,[s'], r'$

(1,1) up -1

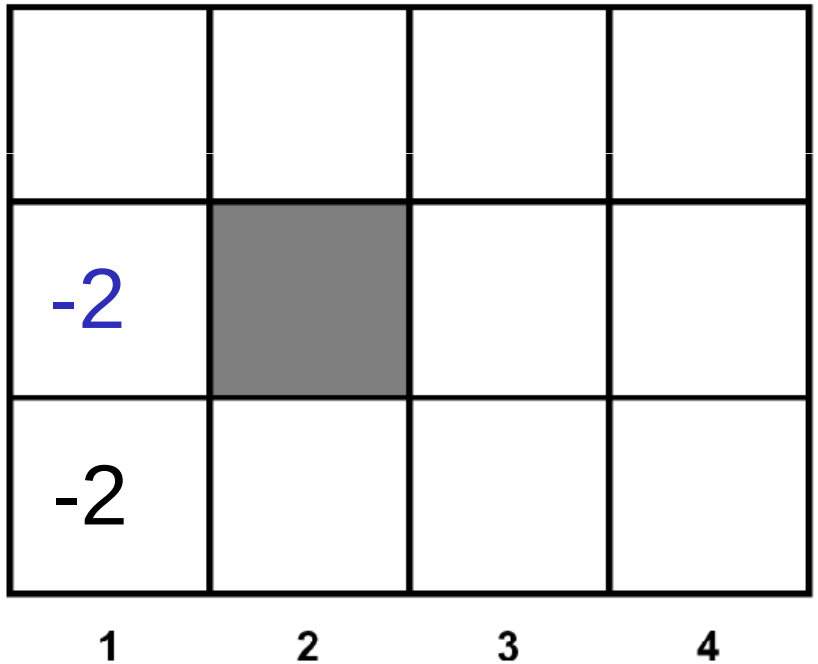
(1,2) up -1

s →(1,2) up -1

s' →(1,3) right -1



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



$\gamma = 1$, $\alpha(N_s[s])=1/N_s[s]$

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

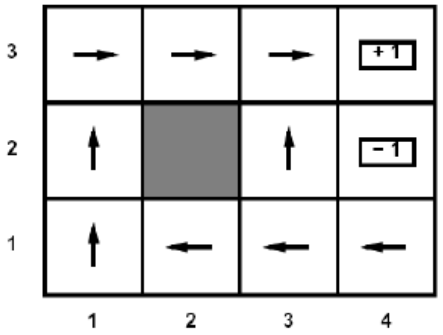
(1,1) up -1

(1,2) up -1

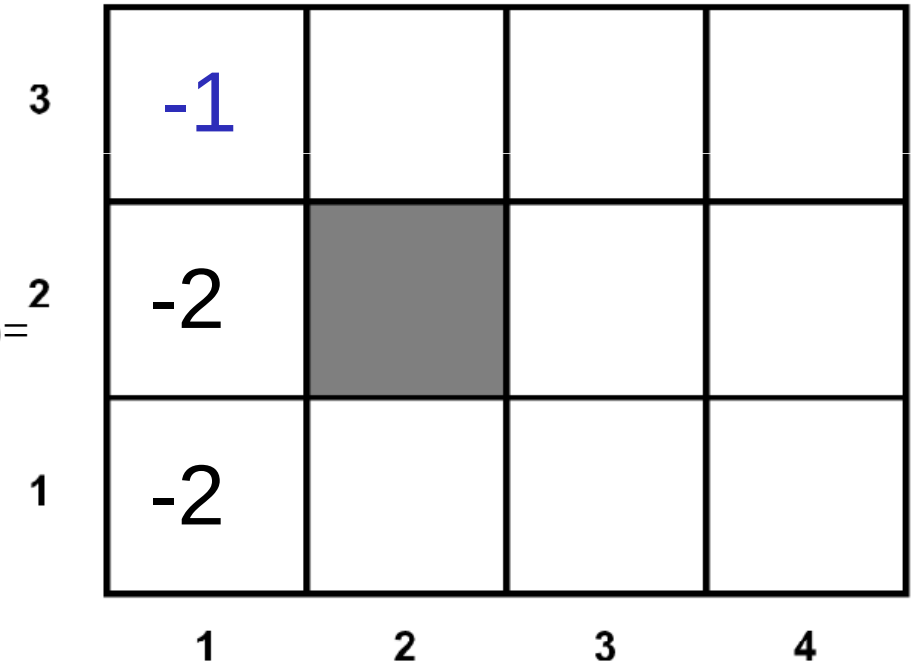
s → (1,2) up -1

s' → (1,3) right -1

(2,3) right -1



s‘=(1,3) nuevo U(1,3)=-1
s =(1,2) no nulo:
$N_s(1,2)=2$
$U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,3)-U(1,2))=$
$= -2 + 1/2 \cdot (-1 + 1 \cdot -1 - (-2))=$
$= -2 + 0.5 \cdot (-1 -1 + 2)= -2 +0= -2$



$\gamma = 1$, $\alpha(N_s[s])=1/N_s[s]$

Ejercicio: 4º paso del primer episodio

Aprendizaje TD: Ejemplo

UNIVERSITAT DE BARCELONA

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** $s, a, r \leftarrow s', \pi\,[s'], r'$

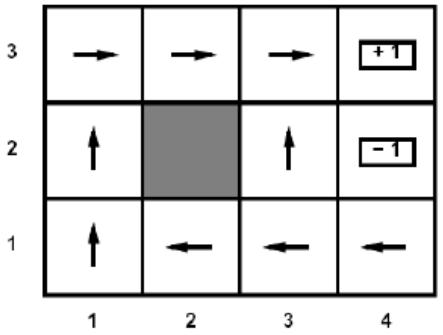
(1,1) up -1

(1,2) up -1

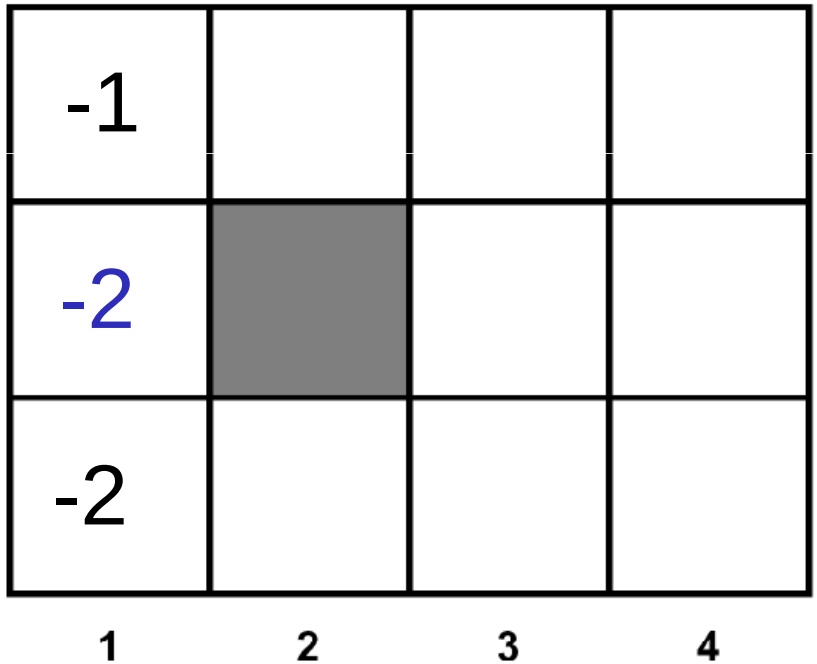
(1,2) up -1

s →(1,3) right -1

s' →(2,3) right -1



s‘=(1,3) nuevo U(1,3)=-1
s =(1,2) no nulo:
  $N_s(1,2)=2$
  $U(1,2)=U(1,2)+\alpha(N_s(1,2)) \cdot (r+\gamma U(1,3)-U(1,2))=$
    $= -2 + 1/2 \cdot (-1 + 1 \cdot -1 - (-2))=$
    $= -2 + 0.5 \cdot (-1 -1 + 2)= -2 +0= -2$
s‘ no terminal: s=(1,3), a=π(1,3)=right, r = -1



$\gamma = 1$, $\alpha(N_s[s]) = 1/N_s[s]$

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

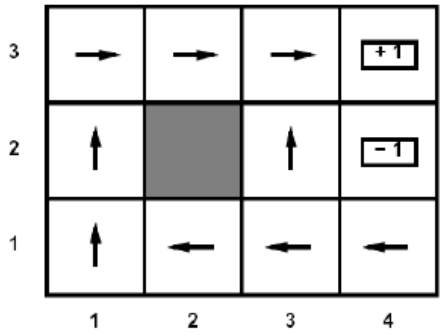
(1,2) up -1

(1,2) up -1

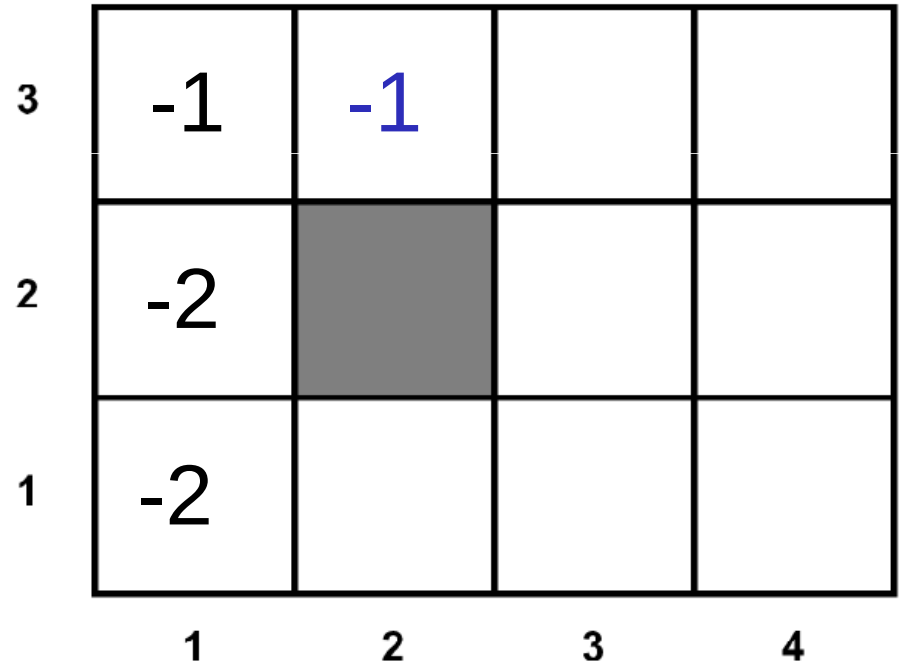
s →(1,3) right -1

s' →(2,3) right -1

(3,3) right -1



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



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

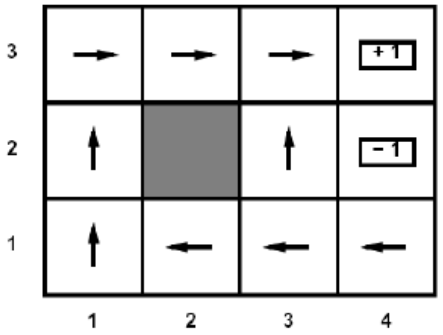
(1,2) up -1

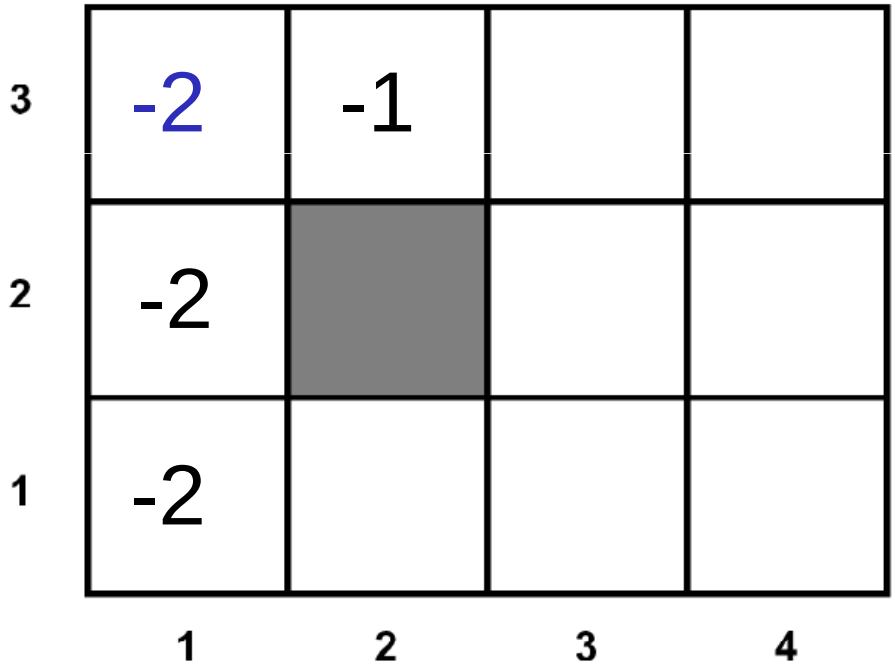
(1,2) up -1

(1,3) right -1

s →(2,3) right -1

s' →(3,3) right -1





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

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

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

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

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...

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 $Q^*(s,a)=0$

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

Q-aprendizaje

- Q-aprendizaje: Iteración de Q-valores basada en muestreo.
- 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:

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

Q-aprendizaje
(libro Russell and Norvig)

UNIVERSITAT DE BARCELONA

función AGENTE-APRENDIZAJE-Q (*percepción*) **devuelve** una acción

entradas: *percepción*, una percepción indica el estado actual s' y la señal de recompensa r'

estática: Q, una tabla de valores de acción indexada por el estado y la acción

N_{sa}, una tabla de frecuencias de los pares estado-acción

s, a, r, el estado, la acción y la recompensa previa, inicialmente nulos

si s no es nulo **entonces hacer**

incrementar $N_{sa}[s, a]$

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

si TERMINAL?[s'] **entonces** s, a, r ← nulo

si no $s, a, r \leftarrow s', \text{argmax}_{a'} f(Q[a', s'], N_{sa}[a', s']), r'$

devolver a

Figura 21.8 Un agente de aprendizaje-Q exploratorio. Es un aprendizaje activo que aprende el valor $Q(a, s)$ de cada acción en cada situación. Usa la misma función de exploración f que el agente ADP exploratorio, pero evita tener que aprender el modelo de transiciones ya que el valor-Q de un estado se puede relacionar directamente con los de sus vecinos.

Q-learning
(e-libro Sutton and Barto)

UNIVERSITAT DE BARCELONA

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'
    $Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$
    $s \leftarrow s'$;
  until s is terminal

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

Propiedades del Q-aprendizaje

- 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

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

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

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

Recapitulemos

- Cosas que sabemos hacer:
  - Resolver pequeños MDPs exactamente, offline
  - Estimar los valores de los estados para una política determinada
  - Estimar Q*(s,a) para la política óptima ejecutando una política de exploración
- Técnicas:
  - Iteración de valores e iteración de políticas
  - Aprendizaje TD
  - Q-aprendizaje
  - Selección exploratoria de acciones