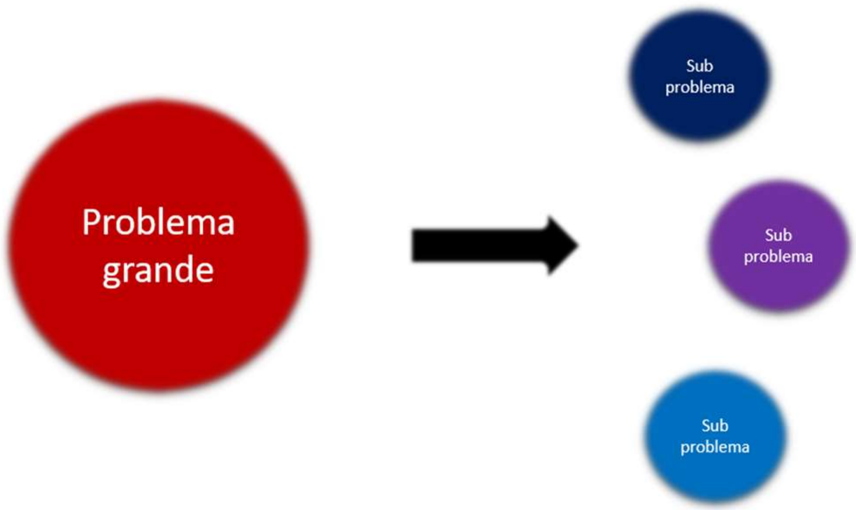
Evaluación de la política

Evaluación de la Política – Introducción a la Programación Dinámica



Evaluación de la Política – Introducción a la Programación Dinámica

- Estos algoritmos requieren de conocimientos previos obtenidos en el Proceso de Decisión de Markóv, que describe la interacción del agente y entorno

- Conocimientos previos: $\{S, A, R, p, \gamma\}$, donde R, p son el modelo del entorno

Evaluación de la Política – Introducción a la Programación Dinámica

- Nos centraremos en el algoritmo de Evaluación de la Política

¿En qué se diferencia este algoritmo de los otros 3?

- La política es la que le indica al agente, partiendo de un estado S qué acción A debe tomar
- Entrenar el agente implica encontrar la mejor política posible $\pi = (a|s)$
- Si decimos "MEJOR política", estamos indicando que hay algunas que funcionan mejor que otras. Eso quiere decir que se **cuantifica la política**
- Debemos asignar por medio de un método, un puntaje a dicha política

Evaluación de la Política – Introducción a la Programación Dinámica

¿En qué se diferencia este algoritmo de los otros 3?

Entra 1 política $\pi = (a|s)$ → **Sale** 1 Evaluación de la Política

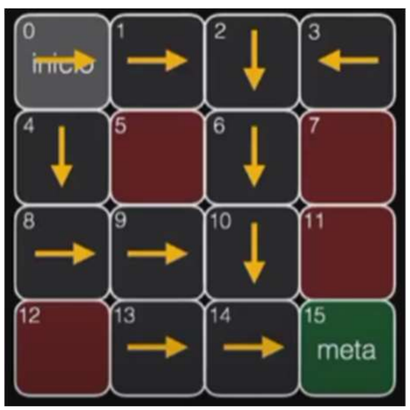
Entran n políticas $\pi = (a|s)$ → **Sale** la mejor política $\pi *= (a|s)$

Entran n políticas $\pi = (a|s)$ → **Sale** la mejor política $\pi *= (a|s)$

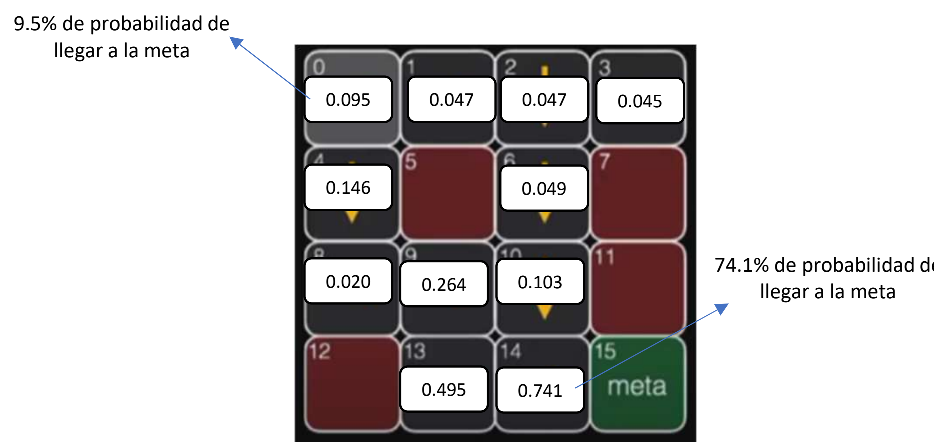
Entran n políticas $\pi = (a|s)$ → **Sale** la mejor política $\pi *= (a|s)$

Evaluación de la Política – Introducción a la Programación Dinámica

Utilizando la función estado-valor, se cuantifica qué tan bueno o qué tan malo es un estado, dependiendo de la política que siga un agente, esos son los valores que están en los recuadros blancos.

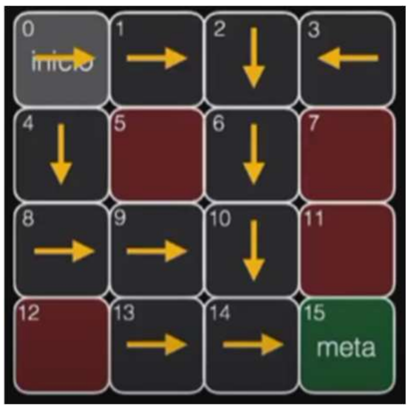


Política

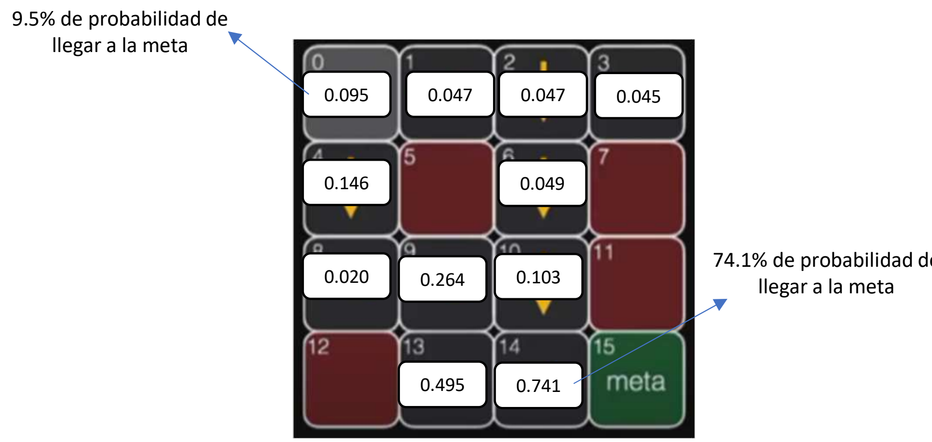


Predicción (valores de la función estado-valor

Evaluación de la Política – Introducción a la Programación Dinámica



Política



Predicción (valores de la función estado-valor

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

Evaluación de la Política – Introducción a la Programación Dinámica

En conclusión, la Evaluación de la Política va a resolver la Ecuación de Bellman, para la función estado-valor.

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

Antes de resolver la ecuación, tener en cuenta:

- La **política** $\pi(a|s)$: Entrada del algoritmo
- La **función de transición** $p(s', r|s, a)$

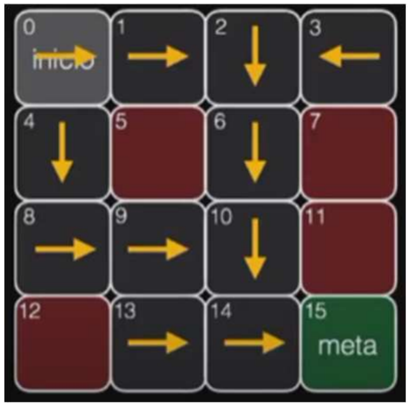
Evaluación de la Política – Introducción a la Programación Dinámica

- Las **recompensas obtenidas** r
- El **factor de descuento** γ
- Los **espacios de estados** $S = \{S_0\ , S_1, S_2, S_3, \dots , S_m\}$
- Los **espacios de acciones** A= $\{A_0\ , A_1, A_2, A_3, \dots , A_n\}$

Evaluación de la Política – Principio de Funcionamiento

¿Cómo se va a resolver la ecuación?

De manera iterativa, inicializando todos los estados en cero.

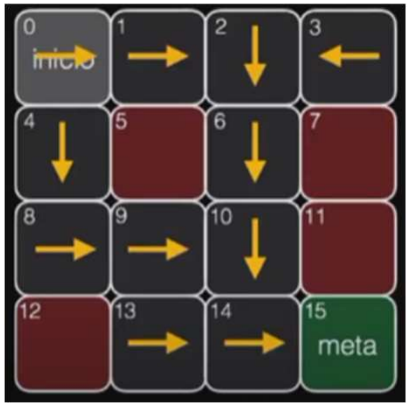


Política

Inicialización

Evaluación de la Política – Principio de Funcionamiento

Primera iteración

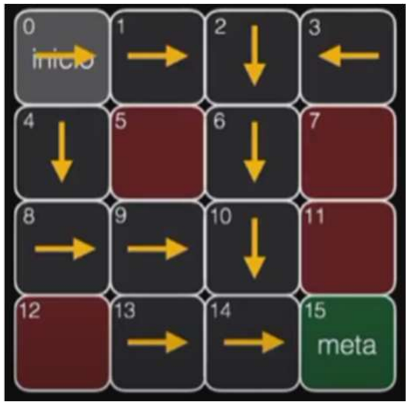


Política

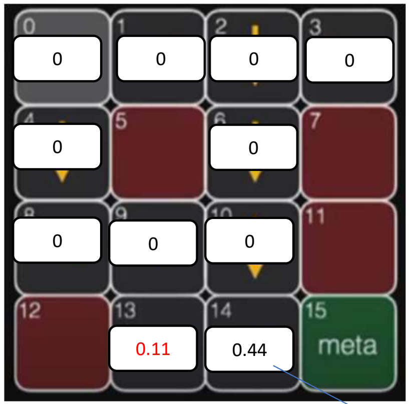
i=1

Evaluación de la Política – Principio de Funcionamiento

Segunda iteración



Política

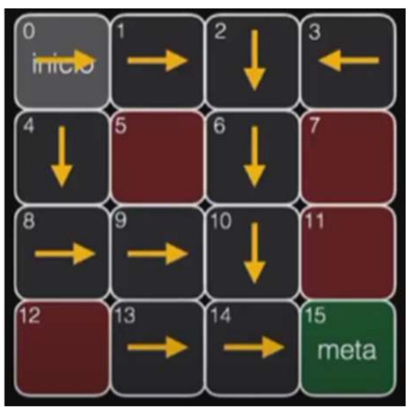


i=2

Se actualiza el valor teniendo en cuenta un nuevo estado

Evaluación de la Política – Principio de Funcionamiento

Tercera iteración



Política

i=3

Se actualiza el valor teniendo en cuenta un nuevo estado

Se actualiza el valor teniendo en cuenta un nuevo estado

Evaluación de la Política – Principio de Funcionamiento

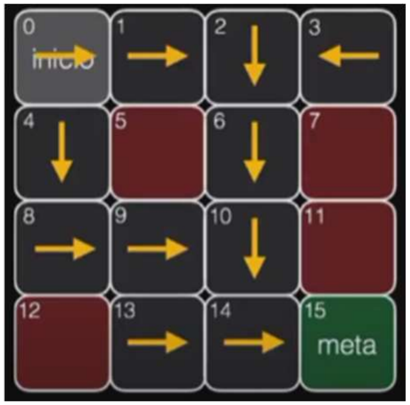
A medida que pasan las iteraciones, los valores para cada estado o casilla se van estabilizando, los cambios no son tan bruscos…

Por ello, en teoría se podría decir que se requiere un número infinito de iteraciones, para tener el valor del estado de manera EXACTA

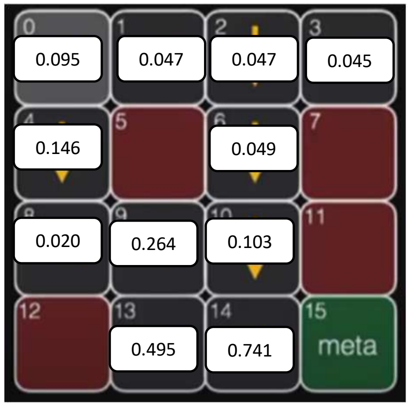
Pero, en la práctica no es posible, tenemos un número finito.

Evaluación de la Política – Principio de Funcionamiento

En la iteración 218…



Política



…i=218

Evaluación de la Política – Las Ecuaciones

Se procede a convertir la ecuación de Bellman, en algo susceptible para aplicar iteraciones

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

$$\Downarrow$$

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Siguiente iteración

Anterior iteración

Evaluación de la Política – Pseudocódigo del algoritmo

- Entrada: π, la política a evaluar
- Parámetro: $\theta > 0$, umbral que permitirá detener la ejecución del algoritmo en un número finito de iteraciones

Valor de referencia para saber si los cambios son significativos o no

- Inicializar:
  - Todos los valores a 0. Los estados terminales siempre se mantendrán en 0
  - $\Delta \leftarrow 0$: variable que contendrá la máxima variación de estados entre una iteración y otra

Qué tanto variaron los valores después de cada iteración

- Iterar

Repetir mientras $\Delta > \theta$:

Mientras valga la pena seguir iterando

por cada s en S:

temp $\leftarrow$ v(s)

Por cada estado, se almacena su valor de manera temporal

$$v(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v(s')]$$

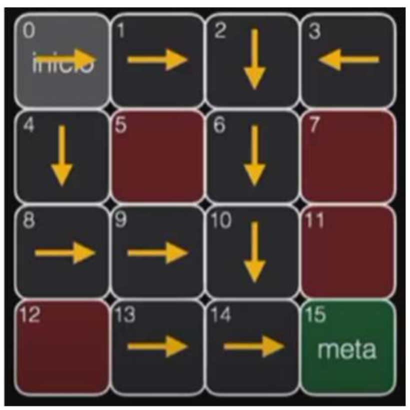
Se calcula el estado actualizado k+1

$\Delta \leftarrow \max(\Delta, |$ v(s)-temp $|)$

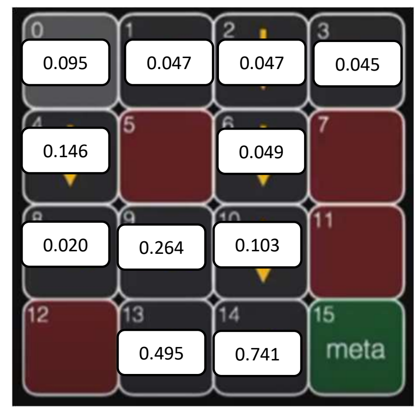
Se actualiza Δ

Evaluación de la Política – Utilidad

Su utilidad es evaluar la política



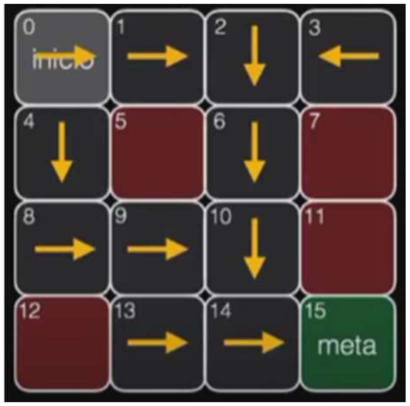
Política 1



Evaluación 1

Evaluación de la Política – Utilidad

Su utilidad es evaluar la política... La mejor sería la política 2, debido a sus probabilidades



Política 2

Evaluación 2

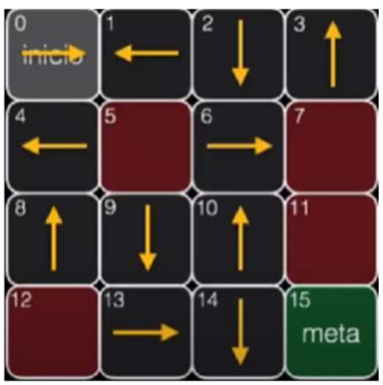
Evaluación de la Política – Ejemplo práctico

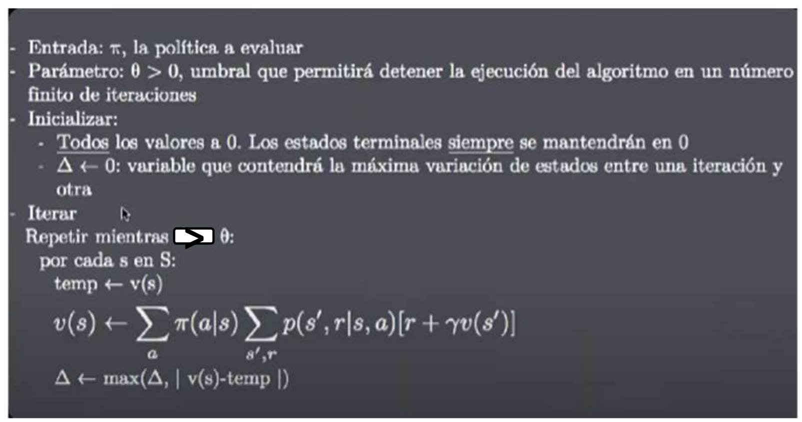
Recordar que…

- El juego cuenta con 16 casillas
- Las casillas 5, 7, 11, 12 y 15 son estados terminales. Ya sea porque el agente cayó en el hueco o porque ganó el juego
- Sólo se obtendrá recompensa al pasar de la casilla 14 a la 15. Cualquier otro transición, dará una recompensa de 0.
- Como hay un componente estocástico, el agente se puede mover en 4 direcciones, pero la política le indica al agente que se mueva hacia la derecha, la probabilidad de que esto pase será de 33% y el 66% restante, está dividido en las direcciones ortogonales.
- Tenemos un factor de descuento que empezará en γ=1

Evaluación de la Política – Ejemplo práctico

Se evaluará la siguiente política. Dada esa política, evaluaremos cada uno de los estados en los que está el agente, teniendo en cuenta el algoritmo que tenemos en pseudocódigo





- Entrada: π, la política a evaluar
- Parámetro: $\theta > 0$, umbral que permitirá detener la ejecución del algoritmo en un número finito de iteraciones
- Inicializar:
  - Todos los valores a 0. Los estados terminales siempre se mantendrán en 0
  - $\Delta \leftarrow 0$: variable que contendrá la máxima variación de estados entre una iteración y otra
- Iterar

Repetir mientras $\Delta > \theta$:
  por cada s en S:
    temp ← v(s)

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

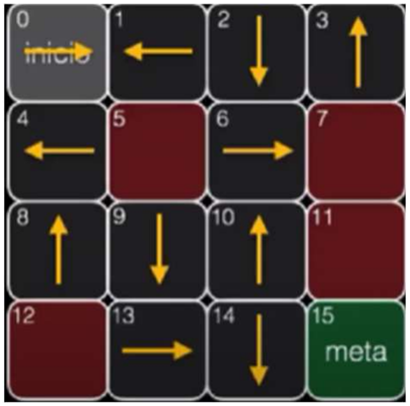
$\Delta \leftarrow \max(\Delta, |\ v(s)\text{-temp}\ |)$

Evaluación de la Política – Ejemplo práctico

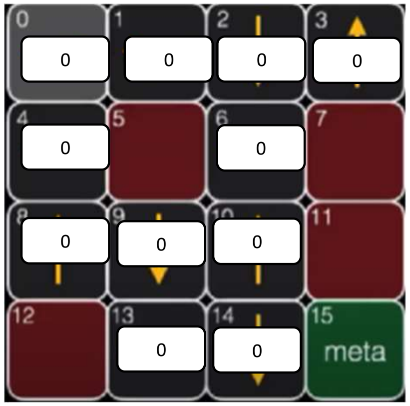
Inicializamos variables…

$\Theta = 1x10^{-10}$

$\Delta=0$



Política



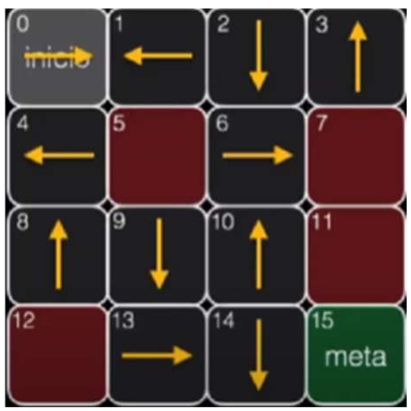
Inicialización

Evaluación de la Política – Ejemplo práctico

Iteración 1

$$v_{k+1}(s) = \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Partiendo de la **casilla 14**, pues es la cual genera una recompensa, de resto los términos se anularán, pues siempre serán r=0.



Política

Inicialización

Evaluación de la Política – Ejemplo práctico

Iteración 1

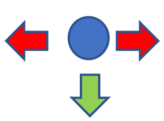
Se anula, pues es sólo 1 acción

$$v_{k+1}(s) = \cancel{\sum_a \pi(a|s)} \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Política

33% s'=13

33% s'=15



33%
s'=14

Estado 14 ($s = 14$), teniendo en cuenta el movimiento ↓ de la política

$$v_1(14) = p(\overset{0.33}{\cancel{s' = 14, r = 0 | s = 14, a = \downarrow}})[0 + \overset{0}{\cancel{1 v_0(14)}}]$$

Se anula, no hay iteración anterior, estamos en la primera

$$+$$

$$p(\overset{0.33}{\cancel{s' = 13, r = 0 | s = 14, a = \leftarrow}})[0 + \overset{0}{\cancel{1 v_0(13)}}]$$

$$+$$

$$p(\overset{0.33}{\cancel{s' = 15, r = 1 | s = 14, a = \rightarrow}})[1 + \overset{1}{\cancel{1 v_0(15)}}]$$

$$\boldsymbol{v_1(14) = 0.33}$$

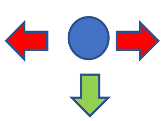
Evaluación de la Política – Ejemplo práctico

Iteración 1

Se anula, pues es sólo 1 acción

$$v_{k+1}(s) = \cancel{\sum_a \pi(a|s)} \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Política



33% s'=13

33% s'=15

33% **s'=14**

Estado 14 ($s = 14$), teniendo en cuenta el movimiento ↓ de la política

$$v_1(14) = \overset{0.33}{p(s' = 14, r = 0|s = 14, a =\downarrow)}[0 + 1\overset{0}{v_0(14)}]$$

Se anula, no hay iteración anterior, estamos en la primera

+

$$\overset{0.33}{p(s' = 13, r = 0|s = 14, a =\leftarrow)}[0 + 1\overset{0}{v_0(13)}]$$

+

$$\overset{0.33}{p(s' = 15, r = 1|s = 14, a =\rightarrow)}[1 + 1\overset{1}{v_0(15)}]$$

$$\boldsymbol{v_1(14) = 0.33}$$

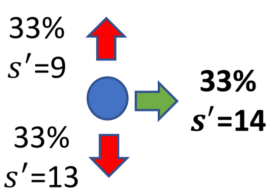
Evaluación de la Política – Ejemplo práctico

Iteración 1

Se anula, pues es sólo 1 acción

$$v_{k+1}(s) = \sum_{a} \cancel{\pi(a|s)} \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Política



Estado 13 ($s = 13$), teniendo en cuenta el movimiento → de la política

$$v_1(13) = \overset{0.33}{p(s' \cancel{= 14, r = 0 | s = 13, a =} \rightarrow)}[0 + \overset{0}{\cancel{1v_0(14)}}]$$

Se anula, no hay iteración anterior, estamos en la primera

$$+$$

$$\overset{0.33}{p(s' \cancel{= 9, r = 0 | s = 13, a =} \uparrow)}[0 + \overset{0}{\cancel{1v_0(9)}}]$$

$$+$$

$$\overset{0.33}{p(s' \cancel{= 13, r = 0 | s = 13, a =} \downarrow)}[0 + \overset{0}{\cancel{1v_0(13)}}]$$

$$\boldsymbol{v_1(13) = 0}$$

Evaluación de la Política – Ejemplo práctico

Actualizamos el valor de Δ, observamos que la única variación fue del estado 14, de resto todas en esa primera iteración, fueron 0, entonces:

$\Delta \leftarrow 0$

$\Delta \leftarrow 0{,}33-0$

$\Delta \leftarrow 0{,}33 > 1x10^{-10}$ (ϴ)

Como se cumple lo anterior, seguimos iterando.

Evaluación de la Política – Ejemplo práctico

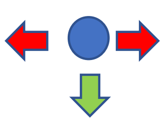
Iteración 2

Se anula, pues es sólo 1 acción

$$v_{k+1}(s) = \sum_{a} \cancel{\pi(a|s)} \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Política

33%
s'=13



33%
s'=15

33%
s'=14

Estado 14 ($s = 14$), teniendo en cuenta el movimiento ↓ de la política

$$v_2(14) = p(\overset{0.33}{\cancel{s'=13, r=0|s=14, a=}}\leftarrow)[0 + 1\overset{0}{\cancel{v_1(13)}}]$$

$v_1(13) = 0$ En la 1era iteración

+

$$p(\overset{0.33}{\cancel{s'=14, r=0|s=14, a=}}\downarrow)[0 + 1\overset{0.33}{\cancel{v_1(14)}}]$$

+

$$p(\overset{0.33}{\cancel{s'=15, r=1|s=14, a=}}\rightarrow)[1 + 1\overset{0}{\cancel{v_0(15)}}]$$

$\boldsymbol{v_2(14)}$ ***= 0.33x0 + 0.33x0.33+ 0.33x1 = 0.4389***

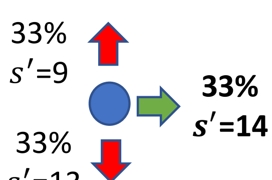
Evaluación de la Política – Ejemplo práctico

Iteración 2

Se anula, pues es sólo 1 acción

$$v_{k+1}(s) = \sum_a \cancel{\pi(a|s)} \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

Política



Estado 13 ($s = 13$), teniendo en cuenta el movimiento → de la política

$$v_2(13) = p(\overset{0.33}{\cancel{s'=14, r=0|s=13, a=\rightarrow}})[0 + \overset{0.33}{\cancel{1v_1(14)}}]$$

+

$$p(\overset{0.33}{\cancel{s'=9, r=0|s=13, a=\uparrow}})[0 + \overset{0}{\cancel{1v_1(9)}}]$$

+

$$p(\overset{0.33}{\cancel{s'=13, r=0|s=13, a=\downarrow}})[0 + \overset{0}{\cancel{1v_1(13)}}]$$

$$v_2(13) = 0.33x0.33 + 0.33x0 + 0.33x0 = \mathbf{0.1089}$$

Evaluación de la Política – Ejemplo práctico

Actualizamos el valor de Δ, observamos que hubo variación en el estado 13 y 14, de resto todas en esa segunda iteración, fueron 0, entonces debido a que hay 2 variaciones, tomamos el valor máximo (el peor de los casos) pues queremos que se estabilice:

Para el estado 13:

Δ←0,1089-0

Δ←0,1089

Para el estado 14:

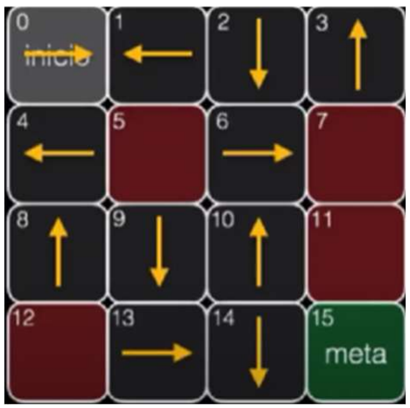
Δ←0,4389-0.33

Δ←0,1089

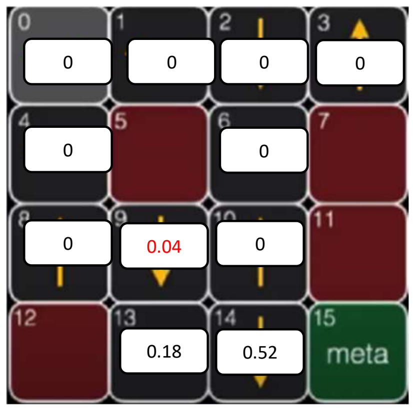
Entonces, Δ←0,1089> $1x10^{-10}$ (ϴ). Como se cumple lo anterior, seguimos iterando.

Evaluación de la Política – Ejemplo práctico

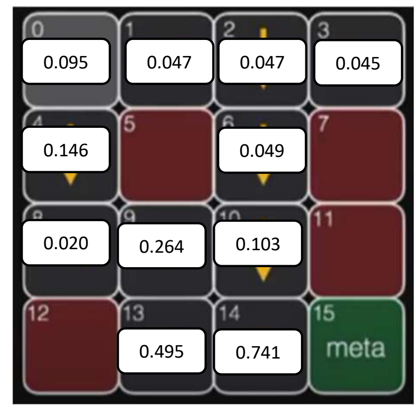
Si continuáramos iterando…



Política



k=3



…k=218

Δ<Θ Por lo tanto ya no se itera más

Mejora de la política

Mejora de la política – Ejemplo práctico

Una desventaja de la evaluación de la política es que sólo indica qué tan buena o mala es. Pero el objetivo principal del **aprendizaje por refuerzo**, es que el agente ejecute la **MEJOR política**

Así que para ir un paso más adelante, se usará la **función acción-valor+algoritmo de evaluación de la política** para **MEJORARLA**

Mejora de la política – Ejemplo práctico

Recordando…

Entra 1 política $\pi = (a|s)$ → 

Entra 1 política $\pi = (a\|s)$	Evaluación de la política	**Sale** 1 Evaluación de la Política
Entran n políticas $\pi = (a\|s)$	Mejora de la política	**Sale** la mejor política $\pi *= (a\|s)$
Entran n políticas $\pi = (a\|s)$	Iteración de la política	**Sale** la mejor política $\pi *= (a\|s)$
Entran n políticas $\pi = (a\|s)$	Iteración de valores	**Sale** la mejor política $\pi *= (a\|s)$

Mejora de la política – Ejemplo práctico

Limitaciones…

En el caso de los 3 últimos algoritmos, al entrar muchas políticas, hay mucho gasto computacional, pues se deben evaluar cada uno de los estados en las k iteraciones, así sea teniendo un tablero de 16 casillas, en otros casos, el panorama de gasto computacional sería PEOR.

Mejora de la política – Ejemplo práctico

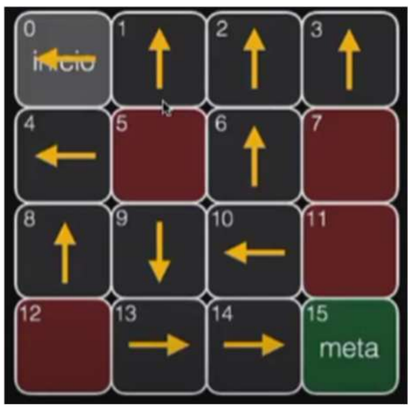
¿Cuál sería la mejor manera para encontrar la política óptima?

Introduciendo la función acción-valor: Nos permite evaluar de manera completa el comportamiento del agente, pues estado-valor, se limita solo a estados del agente.

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

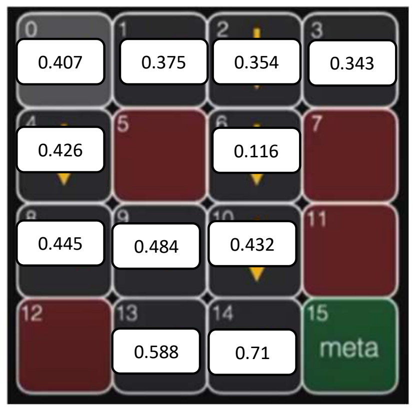
Mejora de la política – Ejemplo práctico

¿Cuál es la dinámica?



Política inicial

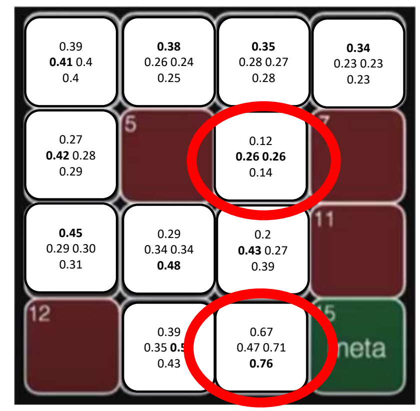
Evaluación de la política



$v\pi(s)$

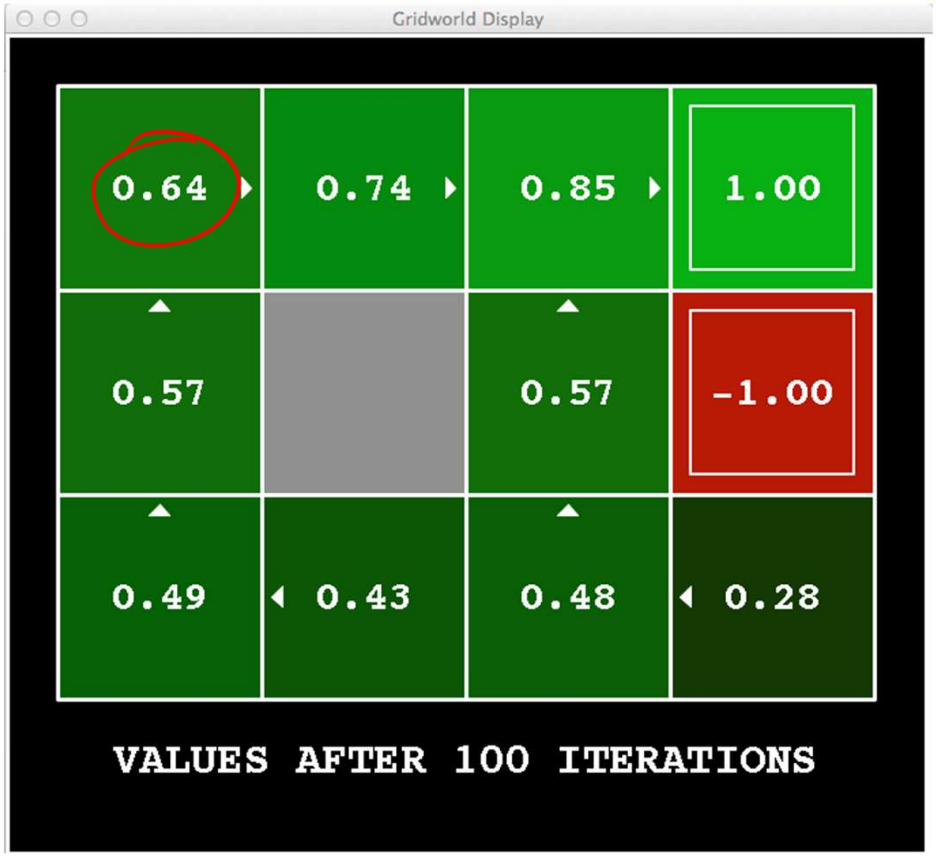
Cálculo función Q

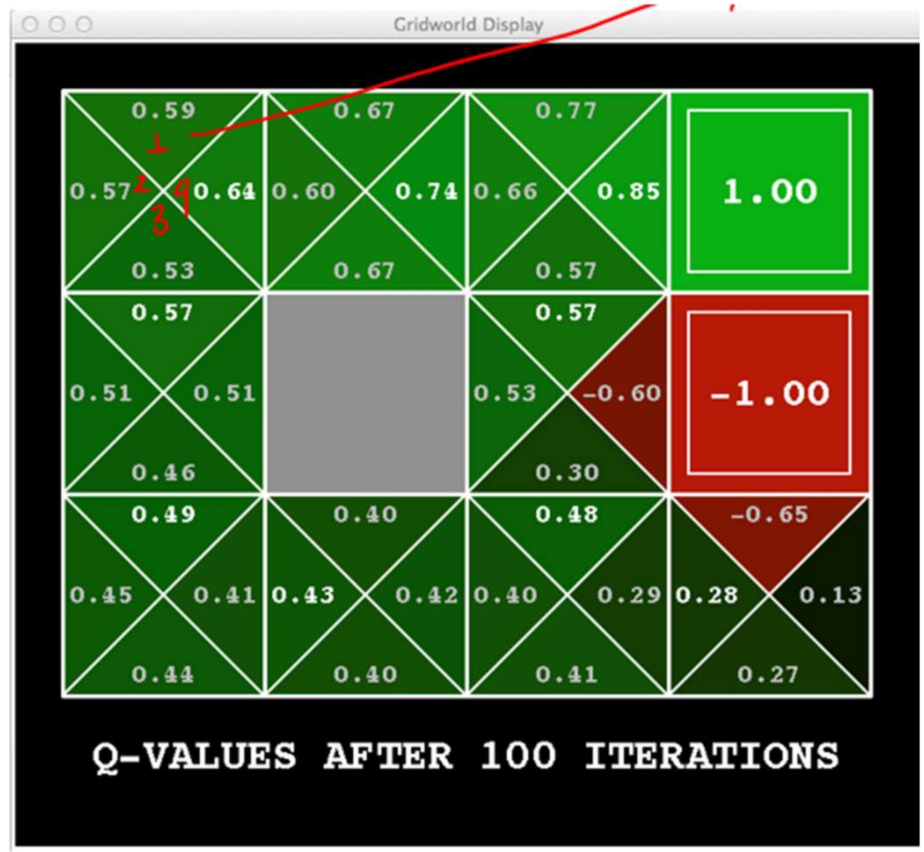
Para mejorar la política, en cada estado, tendremos 3 posibles valores. Y nos fijamos en el mejor (más alto)



$q\pi(s, a)$

En negrilla está la dirección que según la evaluación de la acción, tendrá mejor éxito, podemos observar que algunas acciones se mantienen de acuerdo a la política, pero otras se actualizan como MEJORES





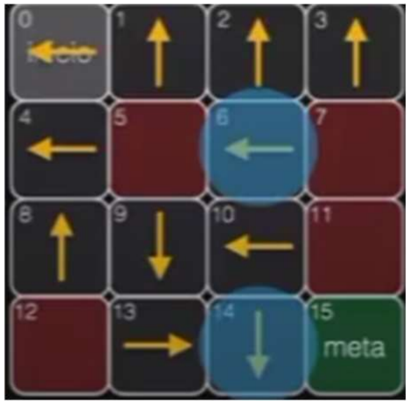
Esta técnica implica obtener la política óptima, que se refiere a la mejor acción para un estado determinado, seleccionando la acción que maximiza la función de valor de estado óptima para ese estado. Esta función de valor de estado óptima se calcula mediante un proceso iterativo. El algoritmo se denomina iteración de valor debido a este enfoque.

El método inicializa la función de valor de estado (V) con valores aleatorios y luego mejora iterativamente su estimación hasta la convergencia. Durante cada iteración, se actualizan tanto los valores Q(s,a) como V(s). La iteración de valores garantiza los mejores resultados posibles al optimizar la función de valor de estado hasta que converge a una solución óptima.

Mejora de la política – Ejemplo práctico

¿Cuál es la dinámica?

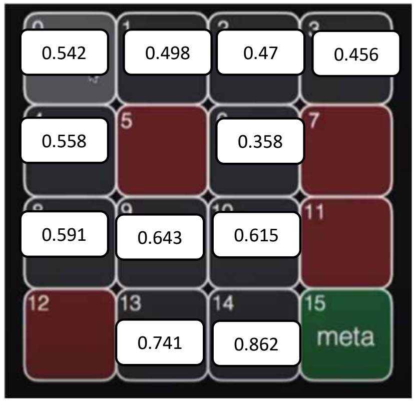
Mejor acción por cada estado



Política mejorada

Cambiaron las acciones para las casillas 6 y 14 con respecto a la política original

Nueva Evaluación de la política



$v'\pi(s)$

¡Versión mejorada de la política original!

Mejora de la Política – Pseudocódigo del algoritmo

Entra la política a mejorar, estados evaluados y el factor γ (es el mismo)

- Entradas: $\pi(a|s)$ (la política a evaluar), estados asociados ($v_\pi(s)$) y γ (debe ser el mismo usado en la evaluación de la política)

Calcular la función acción-valor, partiendo de la ecuación de Bellman. Conocemos TODOS lo valores

- Para cada estado (s) y cada posible acción (a), calcular la función Q a partir de la Ecuación de Bellman:

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

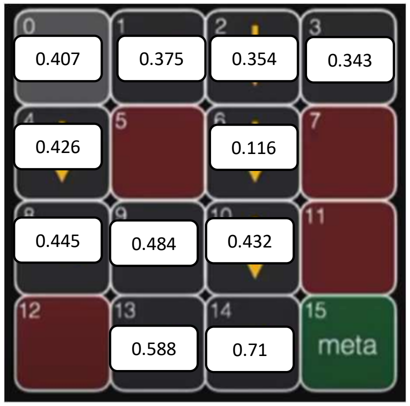
Teniendo en cuenta la función q anterior, tomamos el valor máximo del valor de la acción por casilla (acción con mayor retorno posible)

- La política mejorada será simplemente el resultado de tomar la mejor acción indicada por la función Q en cada uno de los estados:

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

Mejora de la Política – Ejemplo práctico

Política inicial



$v\pi(s)$

Se pretende realizar una mejora a la política inicial, a partir de la evaluación de la política. Partiendo de:

- Una política inicial y su respectiva evaluación
- Casilla 0
- Movimiento hacia la izquierda
- γ=0.99
- Componente estocástico, con sus direcciones ortogonales
- Función acción-valor

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

Mejora de la política – Ejemplo

Estado: 0, acción: ←

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

Valor de la política evaluada en esa casilla

$$q_\pi(s=0, a=\leftarrow) = \overset{0.33}{\cancel{p(s'=0, r=0|s=0, a=\leftarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.4079}{\cancel{v_\pi(s'=0)}}]$$

Valor de la política evaluada en esa casilla

$$+\overset{0.33}{\cancel{p(s'=0, r=0|s=0, a=\uparrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.4079}{\cancel{v_\pi(s'=0)}}]$$

Valor de la política evaluada en esa casilla

$$+\overset{0.33}{\cancel{p(s'=4, r=0|s=0, a=\downarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.4263}{\cancel{v_\pi(s'=4)}}]$$

$$q_\pi(s=0, a=\leftarrow) = 0.33 * 0.99 * 0.4079 + 0.33 * 0.99 * 0.4079 + 0.33 * 0.99 * 0.4263$$

$$\boldsymbol{q_\pi(s=0, a=\leftarrow) = 0.4057}$$

Mejora de la política – Ejemplo

Estado: 0, acción: →

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

Valor de la política evaluada en esa casilla

$$q_\pi(s=0,a=\rightarrow) = \overset{0.33}{\cancel{p(s'=1,r=0|s=0,a=\rightarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.3754}{\cancel{v_\pi(s'=1)}}]$$

Valor de la política evaluada en esa casilla

$$+\overset{0.33}{\cancel{p(s'=0,r=0|s=0,a=\uparrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.4079}{\cancel{v_\pi(s'=0)}}]$$

Valor de la política evaluada en esa casilla

$$+\overset{0.33}{\cancel{p(s'=4,r=0|s=0,a=\downarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} \overset{0.4263}{\cancel{v_\pi(s'=4)}}]$$

$$q_\pi(s=0,a=\rightarrow) = 0.33*0.99*0.3754 + 0.33*0.99*0.4079 + 0.33*0.99*0.4263$$

$$\boldsymbol{q_\pi(s=0,a=\rightarrow) = 0.3951}$$

Mejora de la política – Ejemplo

Estado: 0, acción: ↑

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

Valor de la política evaluada en esa casilla

$$q_\pi(s=0, a=\uparrow) = p(s' \overset{0.33}{\cancel{=0, r=0|s=0, a=\uparrow}})*[\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} v_\pi(s' \overset{0.4079}{\cancel{=}} 0)]$$

Valor de la política evaluada en esa casilla

$$+p(s' \overset{0.33}{\cancel{=0, r=0|s=0, a=\leftarrow}})*[\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} v_\pi(s' \overset{0.4079}{\cancel{=}} 0)]$$

Valor de la política evaluada en esa casilla

$$+p(s' \overset{0.33}{\cancel{=1, r=0|s=0, a=\rightarrow}})*[\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}} v_\pi(s' \overset{0.3754}{\cancel{=}} 1)]$$

$$q_\pi(s=0, a=\uparrow) = 0.33 * 0.99 * 0.4079 + 0.33 * 0.99 * 0.4079 + 0.33 * 0.99 * 0.3754$$

$$\boldsymbol{q_\pi(s=0, a=\uparrow) = 0.389}$$

Mejora de la política – Ejemplo

Estado: 0, acción:↓

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

Valor de la política evaluada en esa casilla

$$q_\pi(s=0, a=\downarrow) = \overset{0.33}{\cancel{p(s'=4, r=0|s=0, a=\downarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}}\, \overset{0.4263}{\cancel{v_\pi(s'=4)}}]$$

Valor de la política evaluada en esa casilla

$$+\overset{0.33}{\cancel{p(s'=0, r=0|s=0, a=\leftarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}}\, \overset{0.4079}{\cancel{v_\pi(s'=0)}}]$$

Valor de la política evaluada en esa casilla

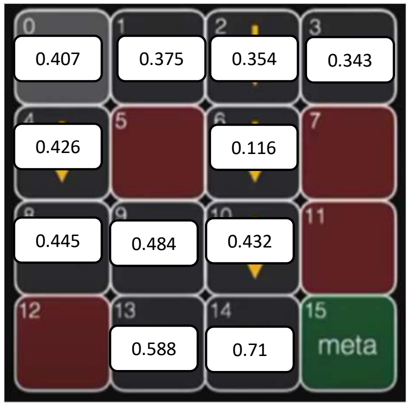
$$+\overset{0.33}{\cancel{p(s'=1, r=0|s=0, a=\rightarrow)}} * [\overset{0}{\cancel{r}} + \overset{0.99}{\cancel{\gamma}}\, \overset{0.3754}{\cancel{v_\pi(s'=1)}}]$$

$$q_\pi(s=0, a=\downarrow) = 0.33 * 0.99 * 0.4263 + 0.33 * 0.99 * 0.4079 + 0.33 * 0.99 * 0.3754$$

$$\boldsymbol{q_\pi(s=0, a=\downarrow)} = \mathbf{0.3951}$$

Mejora de la Política – Ejemplo práctico

Política inicial



$v\pi(s)$

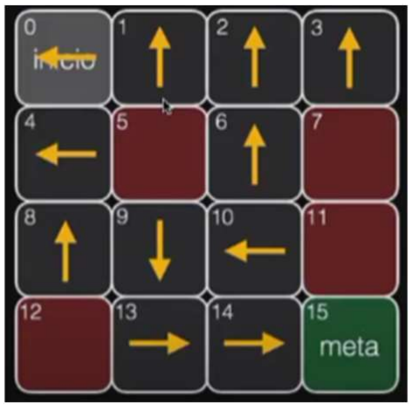
Si nos fijamos, la acción que dio el mejor resultado, fue cuando se ejecutó el movimiento hacia la izquierda, siendo el mismo para la política inicial planteada

$$q_\pi(s = 0, a = \leftarrow) = 0.4057$$

$$q_\pi(s = 0, a = \leftarrow) = 0.4057$$
$$q_\pi(s = 0, a = \rightarrow) = 0.3951$$
$$q_\pi(s = 0, a = \uparrow) = 0.389$$
$$q_\pi(s = 0, a = \downarrow) = 0.3951$$

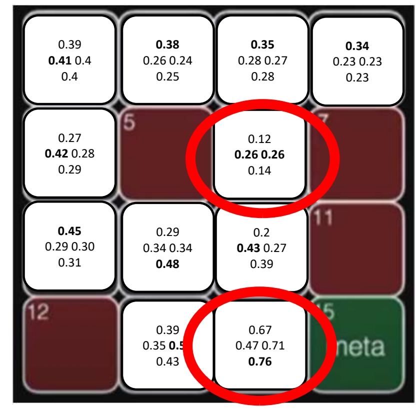
Mejora de la política – Ejemplo práctico

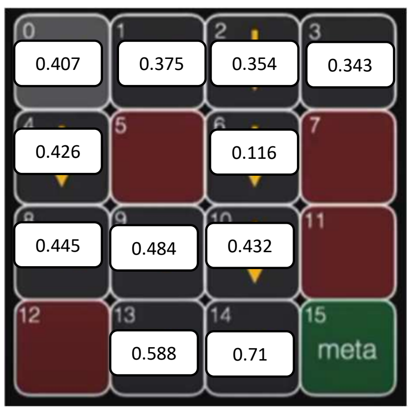
Si se repitiera para cada estado…



Política inicial

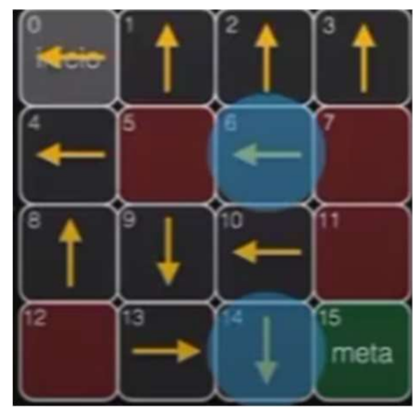
$q\pi(s, a)$





$v\pi(s)$

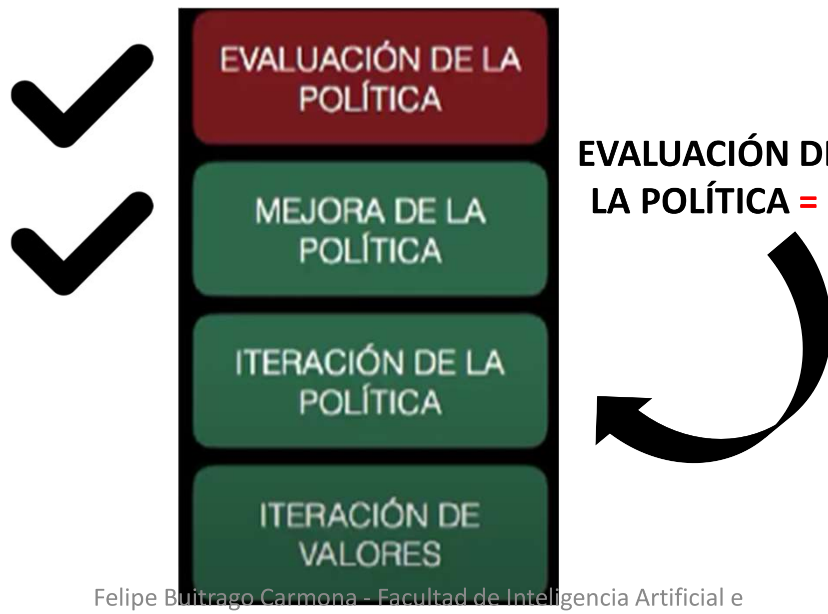
Política mejorada



Iteración de la política

Iteración de la política

Evaluar la política y mejorar la política hasta una x cantidad de veces, es el proceso de iteración de la política



EVALUACIÓN DE LA POLÍTICA + MEJORA DE LA POLÍTICA = ITERACIÓN DE LA POLÍTICA

Iteración de la política - Pseudocódigo

No importa qué tan buena o mala sea esa política inicial, pues al finalizar el algoritmo, tendremos una política **óptima**

Valor de los estados

En el momento en que la política **anterior**, sea igual a política **mejorada** después de n iteraciones, ya no es necesario seguir iterando

- Entradas: una política arbitraria ($\pi(a|s)$); todos los elementos del Proceso de Decisión de Markov (incluyendo γ); el parámetro θ de la Evaluación de la Política
- Iterar indefinidamente y en cada iteración:
  - política_anterior $\longleftarrow \pi(a|s)$
  - $v_\pi(s) \longleftarrow$ Evaluación de la Política
  - $q_\pi(s,a) \longleftarrow$ Calcular valores Q
  - $\pi(a|s) \longleftarrow$ Mejora de la Política
  - Si $\pi(a|s)$ == política_anterior:
    - Detener las iteraciones

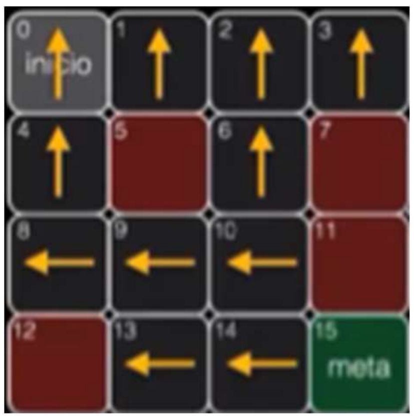
Entre evaluación y mejora de la política, realmente se detendrá en un punto

Valor de las acciones

Por lo tanto, se acaba el proceso y se obtiene la política óptima

Iteración de la política – Ejemplo práctico

Iteración 1



Política arbitraria

Evaluación de la política arbitraria

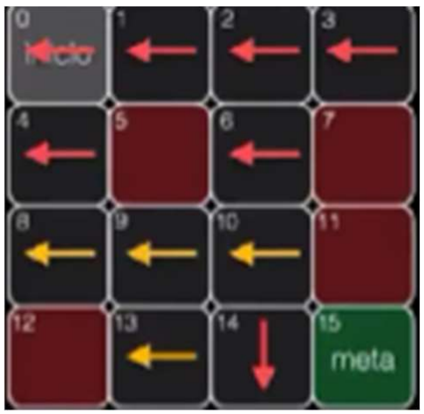
De hecho, es una MUY MALA política, pues en la evaluación de sus estados, todos son 0

Mejora de la política

7 cambios

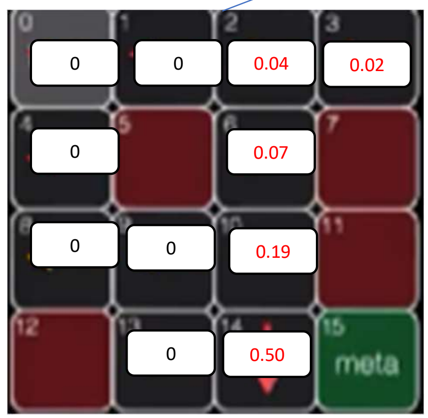
Iteración de la política – Ejemplo práctico

Iteración 2



Política mejorada 1

Evaluación de la política mejorada 1



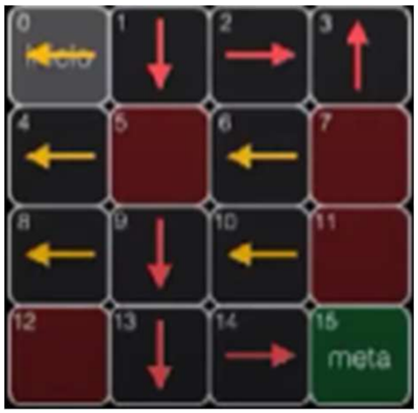
Esta política efectivamente mejoró la situación

Mejora de la política

6 cambios

Iteración de la política – Ejemplo práctico

Iteración 3



Política mejorada 2

Evaluación de la política mejorada 2

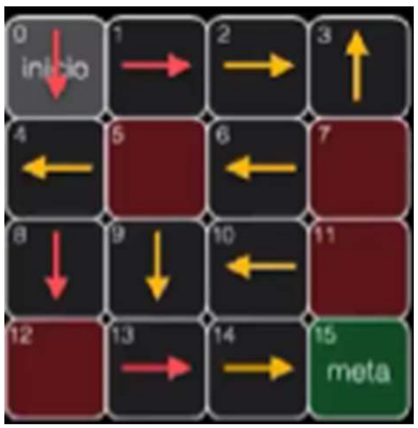
Esta política efectivamente mejoró la situación

Mejora de la política

4 cambios

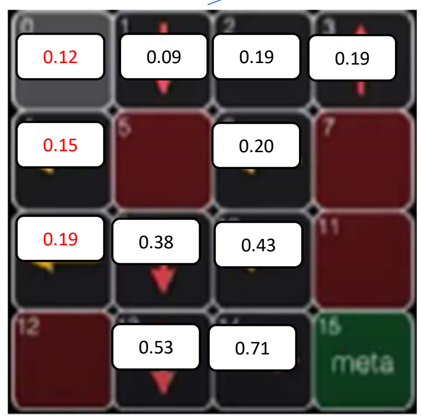
Iteración de la política – Ejemplo práctico

Iteración 4



Política mejorada 3

Evaluación de la política mejorada 3



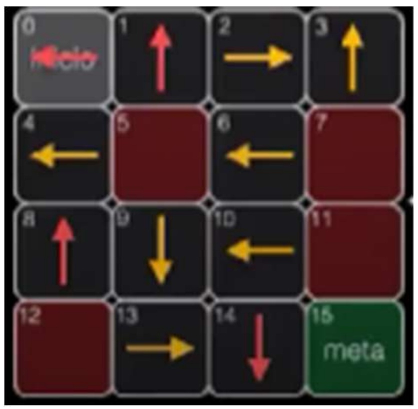
Esta política efectivamente mejoró la situación

Mejora de la política

4 cambios

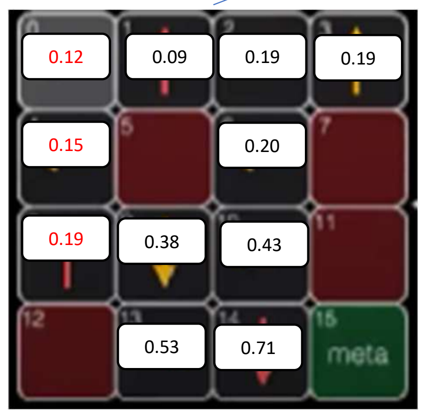
Iteración de la política – Ejemplo práctico

Iteración 5



Política mejorada 4

Evaluación de la política mejorada 4



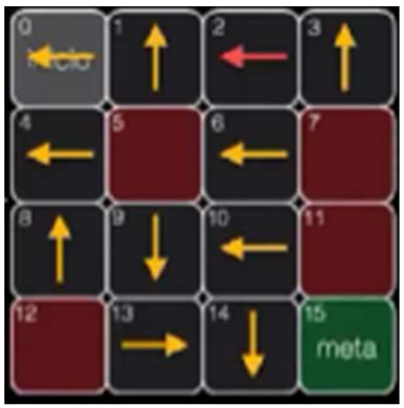
Esta política efectivamente mejoró la situación

Mejora de la política

1 cambio

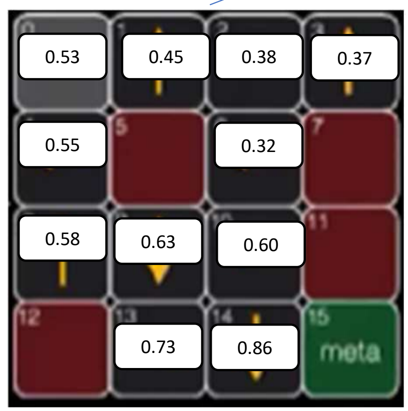
Iteración de la política – Ejemplo práctico

Iteración 6



Política mejorada 5

Evaluación de la política mejorada 5



Esta política muestra 1 cambio en la misma casilla, no hay cambios contundentes

Mejora de la política

Se llegó a 1 de las políticas ÓPTIMAS

¡PARAMOS DE ITERAR...!

Iteración de valores

Iteración de valores

- Manera alternativa de mejorar la política
- Utilizando las funciones acción-valor y estado-valor

Recordando que…

- Los algoritmos de programación dinámica buscan tener una política ÓPTIMA
- La **evaluación de la política** nos permite obtener la función estado-valor: qué tan buena o mala es una política
- La **mejora de la política** nos permite mejorar la política evaluada utilizando la función acción-valor
- La **iteración de la política** es una combinación de la evaluación de la política y mejora de la política.
- La iteración de la política posee un inconveniente, pues se debe repetir muchas veces (múltiples iteraciones) para llegar a una política óptima. Allí aparece la **iteración de valores**, no hay que esperar que terminen TODAS las iteraciones, pues va evaluando simultáneamente al aparecer nuevas políticas.

Iteración de valores – Pseudocódigo de la mejora de política con modificaciones

Entra la política arbitraria, y los demás elementos del Proceso de Decisión de Márkov

- Entradas: una política arbitraria ($\pi(a|s)$); todos los elementos del Proceso de Decisión de Markov (incluyendo γ); el parámetro θ de la Evaluación de la Política
- Iterar indefinidamente y en cada iteración:
  - política_anterior $\longleftarrow \pi(a|s)$
  - $v_\pi(s) \longleftarrow$ Evaluación de la Política
  - $q_\pi(s,a) \longleftarrow$ Calcular valores Q
  - $\pi(a|s) \longleftarrow$ Mejora de la Política
  - Si $\pi(a|s)$ == política_anterior:
    - Detener las iteraciones

Recordar que no es literalmente indefinidamente, pues hay parámetros que hacen que pare las iteraciones

Se requieren múltiples iteraciones

Se calculan el simultáneo al proceso de evaluación de la política

Ahora SÍ se puede entrar en un proceso de mejora de la política

Si ya no hay más mejoras por hacer, se detienen las iteraciones

NOTA: Lo que nos indica el algoritmo de iteración de valores como tal, es que no se debe esperar a que se evalúe la política, para luego calcular los valores Q, sino que estos 2 pasos se hacen en simultáneo

Iteración de valores : Ecuaciones asociadas

Evaluación $$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_k(s')]$$

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

Dado que ambos se van a ejecutar en una misma iteración, observamos que uno de los cálculos es idéntico, y que se puede hacer sólo uno, teniendo en cuenta que $q_\pi(s,a)$, toma el máximo

Unificando ambas ecuaciones…

$$v_{k+1} = max_a \sum_{s',r} p(s',r|s,a)\,[r + \gamma v_\pi(s')]$$

- Se toma el máximo de todas las acciones
- La política no es tenida en cuenta, debido a que están en un misma iteración
- Pareciera ser sólo una función estado-valor, pero con max_a , se incluyen las acciones.

Iteración de valores – Pseudocódigo

Se introduce una función estado-valor arbitraria, **ENVÉS** de política arbitraria. Puede ser inicializada en 0 o en cualquier otro valor

- Entradas: una función estado-valor arbitraria ($v_\pi(s)$); todos los elementos del Proceso de Decisión de Markov (incluyendo γ); el parámetro $\boldsymbol{\theta}$ que controla la convergencia del algoritmo
- Iterar indefinidamente y en cada iteración:
  - $v_{k+1}(s) \leftarrow max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_k(s')]$
  - Calcular $\boldsymbol{\Delta}$
  - Detener si $\boldsymbol{\Delta} < \boldsymbol{\theta}$
- Obtener la política óptima con base en la función estado-valor óptima obtenida en el paso anterior:

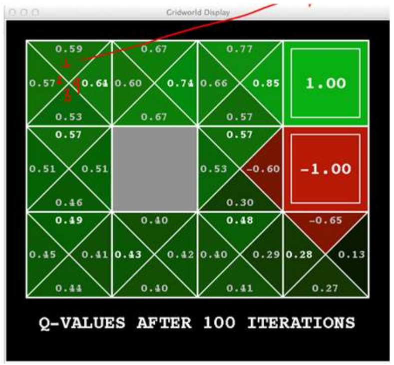
$$\pi(s) = argmax_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_k(s')]$$

Se actualiza la función estado-valor, y de manera simultánea, la acción que maximice el valor para ese estado

Se actualiza Δ, sabiendo que es la variación entre el valor del estado actual y anterior.

Sin cambios significativos

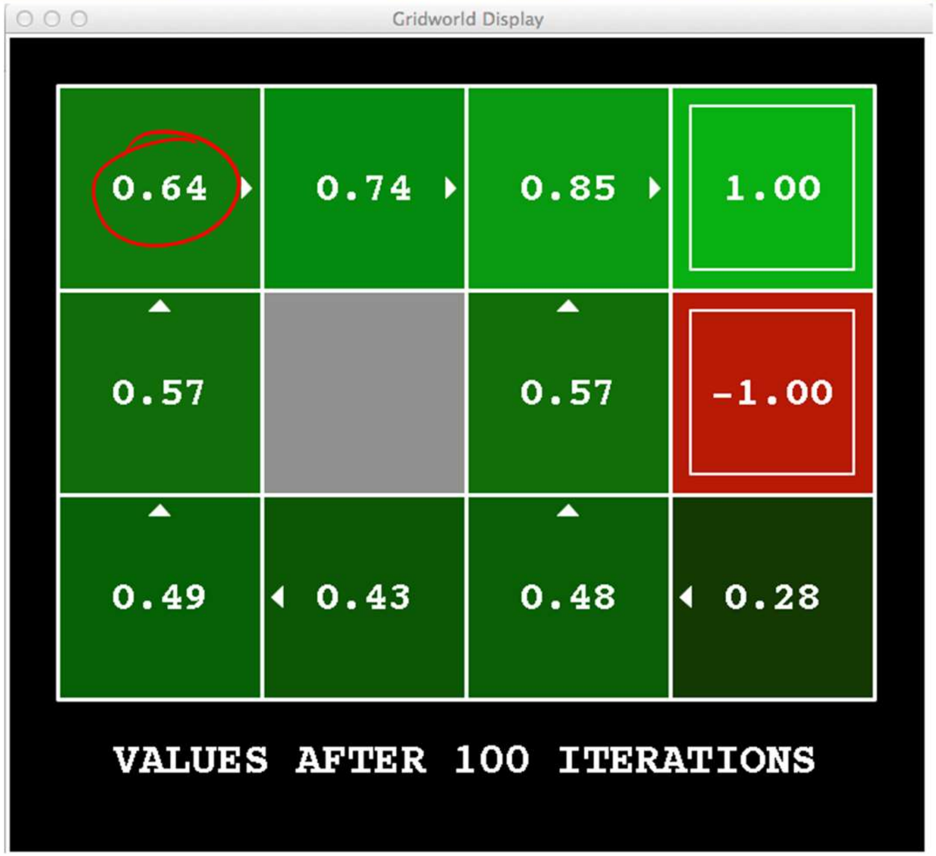
Al terminar las iteraciones, ya se tiene la función estado-valor ÓPTIMA, y con esto se puede obtener la política ÓPTIMA, analizando los valores mas grandes de la tabla de la función q

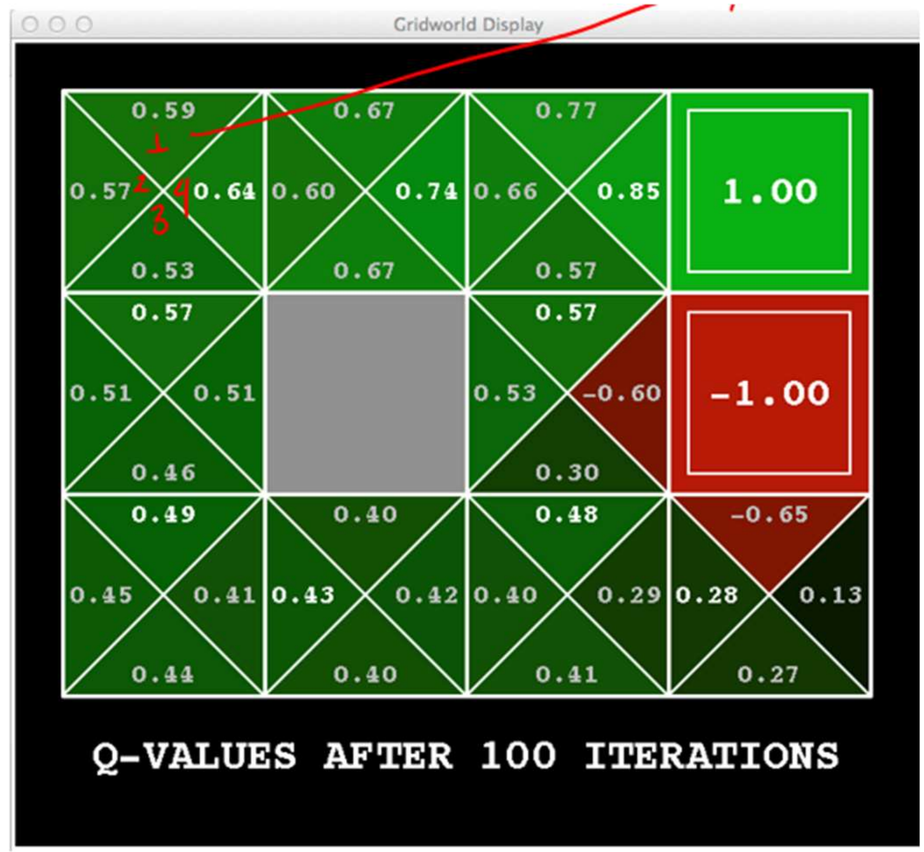


NOTA: Finalmente se obtiene una política óptima, gracias a la previa búsqueda de una función estado-valor óptima, donde cada acción que se selecciona es la mejor posible para el agente

1. Iteración por Valor (Value Iteration):

- **Propósito:** Calcula el valor óptimo de cada estado.
- **Proceso:**
    1. Se inicializa un valor arbitrario para todos los estados. Ej: $\emptyset$
    2. En cada iteración, se actualiza el valor de cada estado utilizando la ecuación de Bellman, que combina la recompensa inmediata y el valor esperado de los estados futuros. Mirar al rededor
    3. Se continúa iterando hasta que los valores convergen (es decir, los cambios en los valores sean menores a un umbral).
    4. Una vez que los valores han convergido, la política óptima se deriva eligiendo la acción que maximiza el valor esperado para cada estado.
- **Resultado:** Una función de valor para cada estado y una política óptima derivada de estos valores.





Esta técnica implica obtener la política óptima, que se refiere a la mejor acción para un estado determinado, seleccionando la acción que maximiza la función de valor de estado óptima para ese estado. Esta función de valor de estado óptima se calcula mediante un proceso iterativo. El algoritmo se denomina iteración de valor debido a este enfoque.

El método inicializa la función de valor de estado (V) con valores aleatorios y luego mejora iterativamente su estimación hasta la convergencia. Durante cada iteración, se actualizan tanto los valores Q(s,a) como V(s). La iteración de valores garantiza los mejores resultados posibles al optimizar la función de valor de estado hasta que converge a una solución óptima.

Iteración de valor

A diferencia de la evaluación de políticas, que tiene ecuaciones lineales que se pueden resolver directamente, en la iteración de valores, debido a la operación *máxima* , las ecuaciones ya no son lineales. Como resultado, tenemos que usar un procedimiento iterativo para resolverlas.

De manera similar a lo que hicimos en la iteración de políticas, comenzamos inicializando la utilidad de cada estado como cero y fijamos γ en 0,5. Lo que tenemos que hacer es recorrer los estados utilizando la ecuación de Bellman.

$$v(s) = r(s) + \gamma \max_a \left[\sum_{s'} p(s'|s,a) v(s') \right]$$

Empecemos desde $s = 0$,

$$v(0) = r(0) + \gamma \max_a \left[\sum_{s'} p(s'|s=0,a) v(s') \right]$$

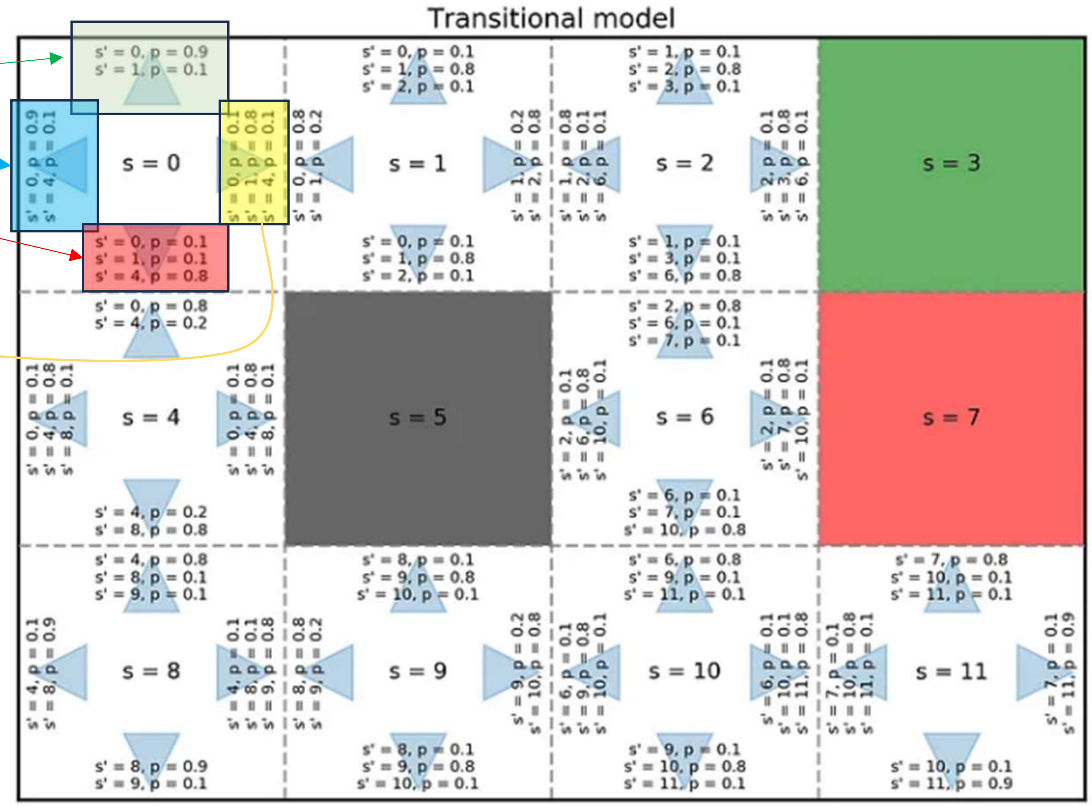
$$= r(0) + \gamma \max \begin{bmatrix} \sum_{s'} p(s'|s=0,a=\mathrm{UP}) v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{LEFT}) v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{DOWN}) v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{RIGHT}) v(s') \end{bmatrix}$$

$$= r(0) + \gamma \max \begin{bmatrix} 0.9\, v(0) + 0.1 v(1) \\ 0.9\, v(0) + 0.1 v(4) \\ 0.8 v(4) + 0.1 v(1) + 0.1 v(0) \\ 0.8 v(1) + 0.1 v(0) + 0.1 v(4) \end{bmatrix}$$

$$= -0.04 + 0.5 \times \max \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = -0.04$$

NOTA: La recompensa para este ejercicio por cada estado no final es de -0.04

1. Al inicio los valores $v(s')$ están en 0
2. Mirar cada una de las acciones en la sumatoria



Modelo de transición del ejemplo del mundo de la red

Empecemos desde $s = 0$,

$$v(0) = r(0) + \gamma \max_a \left[\sum_{s'} p(s'|s=0,a)v(s') \right]$$

$$= r(0) + \gamma \max \begin{bmatrix} \sum_{s'} p(s'|s=0,a=\mathrm{UP})v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{LEFT})v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{DOWN})v(s') \\ \sum_{s'} p(s'|s=0,a=\mathrm{RIGHT})v(s') \end{bmatrix}$$

$$= r(0) + \gamma \max \begin{bmatrix} 0.8v(0) + 0.1v(0) + 0.1v(1) \\ 0.8v(0) + 0.1v(4) + 0.1v(0) \\ 0.8v(4) + 0.1v(1) + 0.1v(0) \\ 0.8v(1) + 0.1v(0) + 0.1v(4) \end{bmatrix}$$

$$= -0.04 + 0.5 \times \max \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \boxed{-0.04}$$

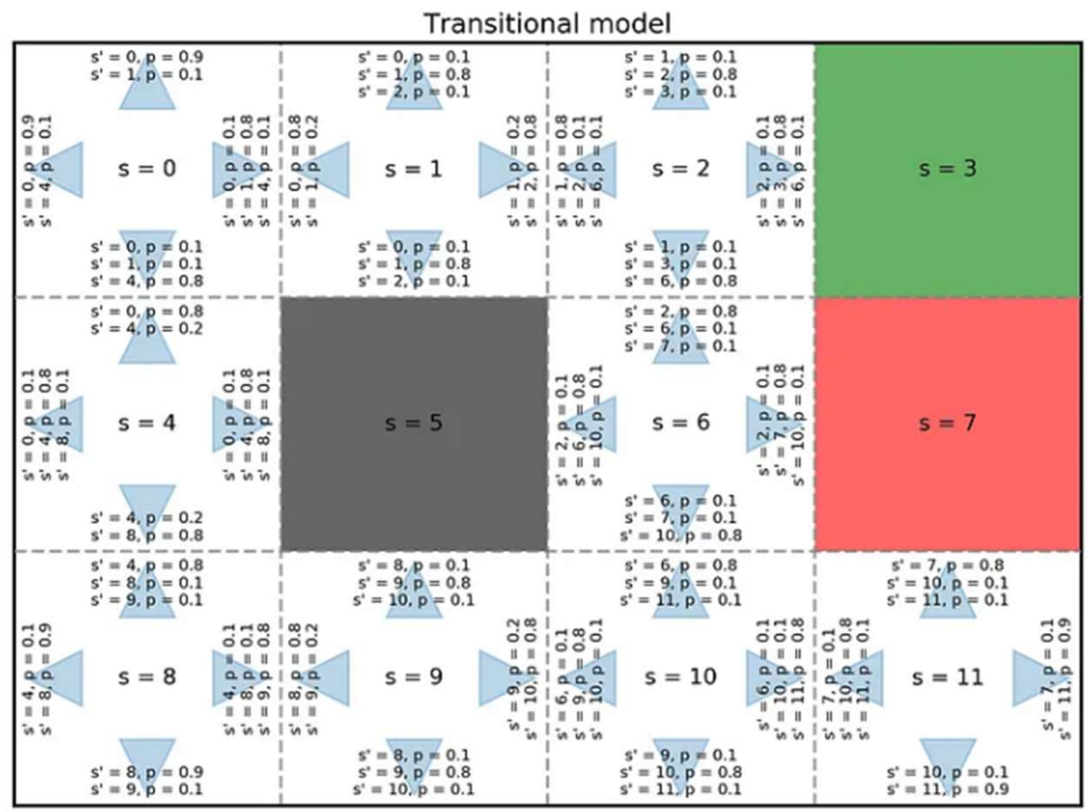
Nuevamente estamos usando un procedimiento en el lugar, lo que significa que de ahora en adelante, siempre que veamos $v(0)$, será $-0{,}04$ en lugar de 0.

Pasemos a $s = 1$, tenemos

$$v(1) = r(1) + \gamma \max \begin{bmatrix} 0.8v(1) + 0.1v(0) + 0.1v(2) \\ 0.8v(0) + 0.1v(1) + 0.1v(1) \\ 0.8v(1) + 0.1v(2) + 0.1v(0) \\ 0.8v(2) + 0.1v(1) + 0.1v(1) \end{bmatrix}$$

$$= -0.04 + 0.5 \times \max \begin{bmatrix} 0.1 \times (-0.04) \\ 0.8 \times (-0.04) \\ 0.1 \times (-0.04) \\ 0 \end{bmatrix} = -0.04$$

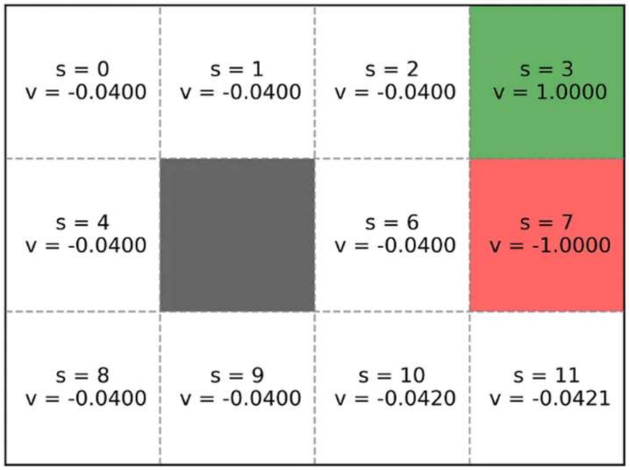
1. Al inicio los valores $v(s')$ están en 0



Modelo de transición del ejemplo del mundo de la red

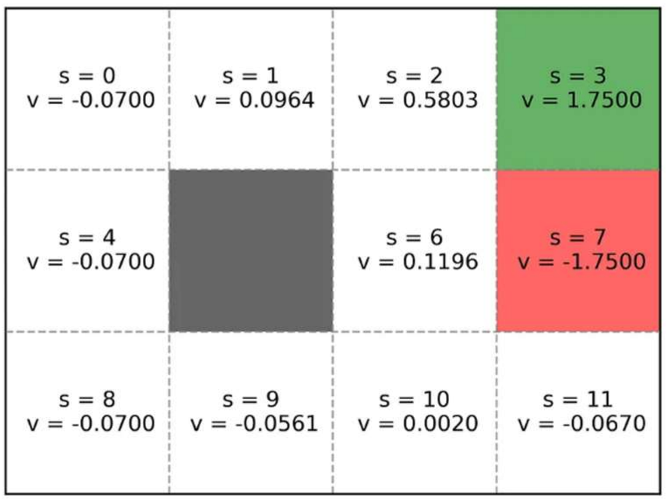
Después de repetir esto para los estados 2, 3, ... hasta el 11, obtenemos esta utilidad.

1



Y vuelve a iterar,

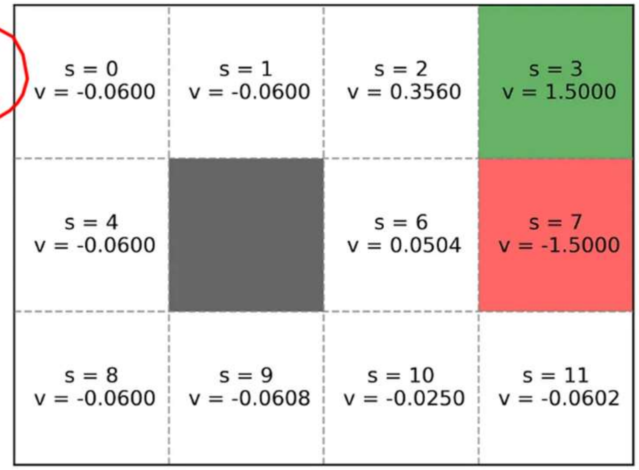
3



Utilidad después de la tercera iteración

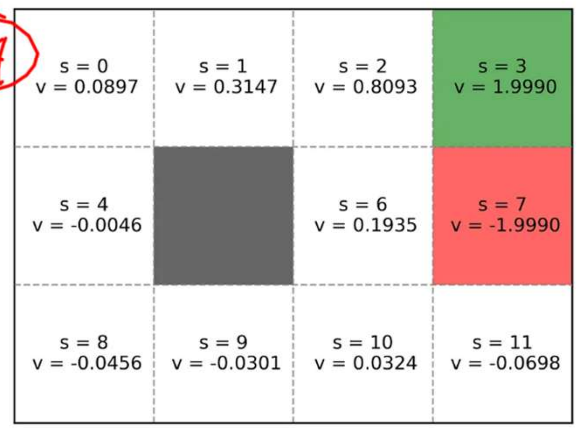
Ahora es el momento de iterar nuevamente, comenzando desde $s = 0$ hasta $s = 11$.

2



Repetimos la iteración hasta que el cambio de utilidad entre dos iteraciones consecutivas sea marginal. Después de 11 iteraciones, el cambio del valor de utilidad de cualquier estado es menor que 0,001. Nos detenemos aquí y la utilidad que obtenemos es la utilidad asociada con la política óptima.

4



Utilidad después de 11 iteraciones

En comparación con la iteración de políticas, la iteración de valores también funciona porque incorpora la operación *máxima* duran te las iteraciones de valores. Como elegimos la utilidad máxima en cada iteración, realizamos implícitamente la operación *argmax* para excluir las acciones subóptimas y converger a la acción óptima.

Obtenga una política óptima

Mediante la iteración de valores, determinamos la utilidad de la política óptima. De manera similar a la iteración de políticas, podemos obtener la política óptima aplicando la siguiente ecuación para cada estado.

$$\pi(s) \leftarrow \mathrm{argmax}_{a} \left[\sum_{s'} p(s'|s,a)v(s') \right]$$

acción que maximiza la sumatoria, osea el valor más grande de todo el análisis.

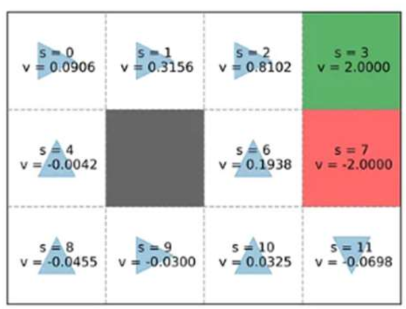
Juntos: **arg max de una sumatoria**

Cuando ves algo como:

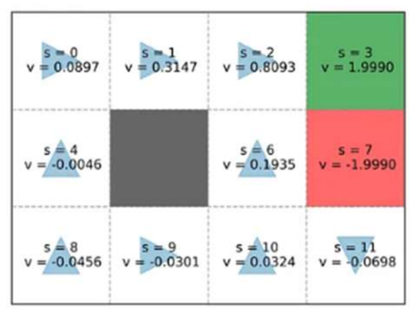
$$\arg\max_x \sum_{i=1}^{n} h(x,i)$$

Esto significa que deseas encontrar el valor de x que maximiza la sumatoria de $h(x,i)$ para todos los valores de i de 1 a n.

Si comparamos las utilidades obtenidas mediante la iteración de valor con las obtenidas mediante la iteración de política, podemos encontrar que las utilidades son muy similares. Como hemos comentado antes, estas utilidades son soluciones de las ecuaciones de Bellman. La iteración de política y la iteración de valor son sólo dos métodos alternativos para resolver las ecuaciones de Bellman. Por lo tanto, para el mismo MDP con las mismas ecuaciones de Bellman, independientemente del método, deberíamos obtener los mismos resultados. En la práctica, debido a las diferencias, como el criterio de parada en los algoritmos de iteración de política e iteración de valor, obtenemos resultados ligeramente diferentes.



Policy iteration



Value iteration