Inteligencia Artificial

Procesos de Decisión de Markov

Edgar Andrade, Ph.D.

Matemáticas Aplicadas y Ciencias de la computación

Última revisión: Noviembre de 2023

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

Policy iteration

Value iteration

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

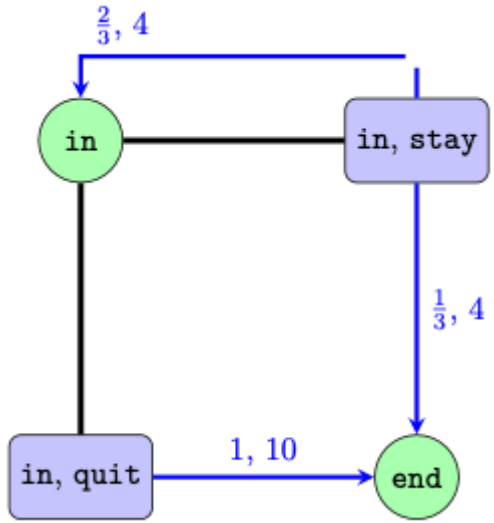
Policy iteration

Value iteration

Procesos de decisión markovianos

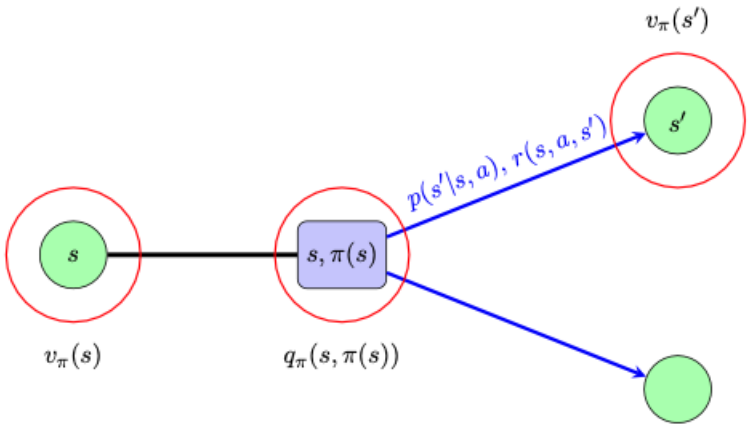
Para cada ronda $k = 1, 2, \ldots$

- Estados: `in` o `end`
- Acciones: `stay` o `quit`
- Si acción $==$ `quit`:
  recompensa $= \$10$
  utilidad $+=$ recompensa y
  `end`
- Si acción $==$ `stay`:
  recompensa $= \$4$
  utilidad $+=$ recompenza y
  lanza el dado:
  - Si `dado` en $\{1, 2\}$: `end`
  - Si no: siguiente ronda



Ecuación de Bellman

$$v_\pi(s) = \begin{cases} 0, & \text{si } s \text{ es terminal} \\ q_\pi(s, \pi(s)), & \text{en otro caso} \end{cases} \qquad q_\pi(s, \pi(s)) = \sum_{s'} \left(p(s'|s, \pi(s)) \left[r + \gamma v_\pi(s') \right] \right)$$



$$v_\pi(s) = \begin{cases} 0, & \text{si } s \text{ es terminal} \\ \sum_{s'} \left(p(s'|s, \pi(s)) \left[r + \gamma v_\pi(s') \right] \right), & \text{en otro caso} \end{cases}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Policy evaluation

☞ Usar la ecuación de Bellman como una regla iterativa:

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

Policy evaluation

Algorithm 1: Iterative policy evaluation

Data: una política π
Result: valor $V_\pi(s)$ para cada s
$V(s) \leftarrow 0$ para cada estado s;
repeat
  $\Delta \leftarrow 0$;
  for *cada estado* s **do**
    $v \leftarrow V(s)$;
    $V(s) \leftarrow \sum_{s'} \Big(p(s'|s, \pi(s))\Big[r + \gamma V(s')\Big]\Big)$;
    $\Delta \leftarrow \text{máx}\Big(\Delta, |v - V(s)|\Big)$;
  end
until $\Delta < \theta$;

Ejercicio

Suponga el siguiente entorno con $\gamma = 0{,}8$:

A	B	C

- El agente parte de la casilla A y puede moverse a izquierda o a derecha.
- Si el agente llega a C, el juego termina.
- Moverse a una casilla distinta a C tiene como recompensa -1. La recompensa de moverse a C es 10.
- Moverse a la izquierda resulta en la casilla de la izquierda con probabilidad 1. Moverse a la izquierda en la casilla A deja al agente en A.
- Moverse a la derecha resulta en la casilla de la derecha con probabilidad 0.9 y en la misma casilla con probabilidad 0.1.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejercicio

Sea $\pi(s)$ la política de moverse a la derecha para todo s.

Realice dos iteraciones del algoritmo iterativo de evaluación de política para la política π.

Solución

$$V(A) = 0 \qquad V(B) = 0 \qquad V(C) = 0$$

Solución

$$V(A) = 0 \qquad V(B) = 0 \qquad V(C) = 0$$

$$V(A) = -1 \qquad V(B) = 8{,}9 \qquad V(C) = 0$$

Solución

$$V(A) = 0 \quad V(B) = 0 \quad V(C) = 0$$

$$V(A) = -1 \quad V(B) = 8{,}9 \quad V(C) = 0$$

$$V(A) = 5{,}328 \quad V(B) = 9{,}612 \quad V(C) = 0$$

Solución

$$V(A) = 0 \qquad V(B) = 0 \qquad V(C) = 0$$

$$V(A) = -1 \qquad V(B) = 8{,}9 \qquad V(C) = 0$$

$$V(A) = 5{,}328 \qquad V(B) = 9{,}612 \qquad V(C) = 0$$

$$V(A) = 6{,}347 \qquad V(B) = 9{,}669 \qquad V(C) = 0$$

Solución

$$V(A) = 0 \qquad V(B) = 0 \qquad V(C) = 0$$

$$V(A) = -1 \qquad V(B) = 8{,}9 \qquad V(C) = 0$$

$$V(A) = 5{,}328 \qquad V(B) = 9{,}612 \qquad V(C) = 0$$

$$V(A) = 6{,}347 \qquad V(B) = 9{,}669 \qquad V(C) = 0$$

$$V(A) = 6{,}469 \qquad V(B) = 9{,}674 \qquad V(C) = 0$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

Policy iteration

Value iteration

Política óptima

$$\pi' \geq \pi \qquad \text{sii} \qquad v_{\pi'}(s) \geq v_{\pi}(s) \text{ para todo } s$$

Política óptima

$$\pi' \geq \pi \qquad \text{sii} \qquad v_{\pi'}(s) \geq v_{\pi}(s) \text{ para todo } s$$

$$\pi^* \text{ es óptima} \qquad \text{sii} \qquad \pi^* \geq \pi \text{ para toda } \pi$$

Política óptima

$$\pi' \geq \pi \qquad \text{sii} \qquad v_{\pi'}(s) \geq v_{\pi}(s) \text{ para todo } s$$

$$\pi^* \text{ es óptima} \qquad \text{sii} \qquad \pi^* \geq \pi \text{ para toda } \pi$$

$$v_*(s) = \max_{\pi} v_{\pi}(s)$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Política óptima

$$\pi' \geq \pi \qquad \text{sii} \qquad v_{\pi'}(s) \geq v_{\pi}(s) \text{ para todo } s$$

$$\pi^* \text{ es óptima} \qquad \text{sii} \qquad \pi^* \geq \pi \text{ para toda } \pi$$

$$v_*(s) = \max_{\pi} v_{\pi}(s)$$

$$q_*(s,a) = \max_{\pi} q_{\pi}(s,a)$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Backup diagrams

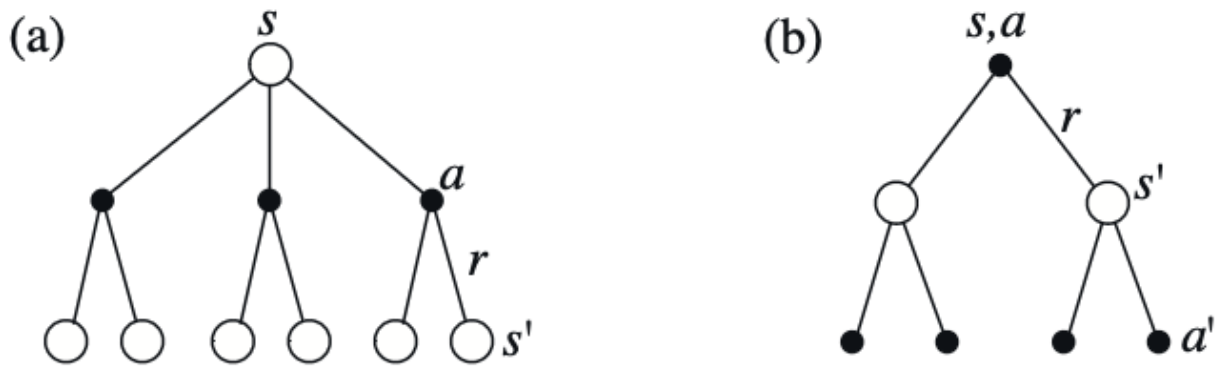


Figure 3.4: Backup diagrams for (a) v_π and (b) q_π.

Backup diagrams

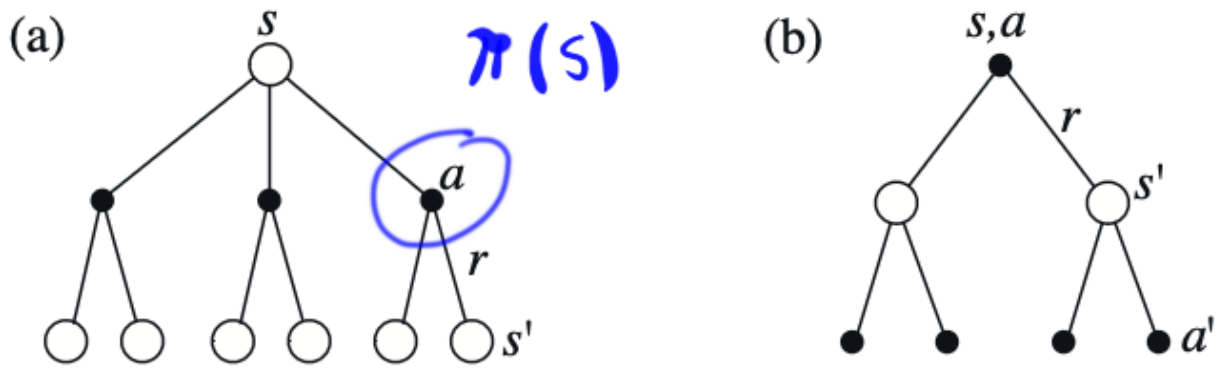


Figure 3.4: Backup diagrams for (a) v_π and (b) q_π.

Backup diagrams

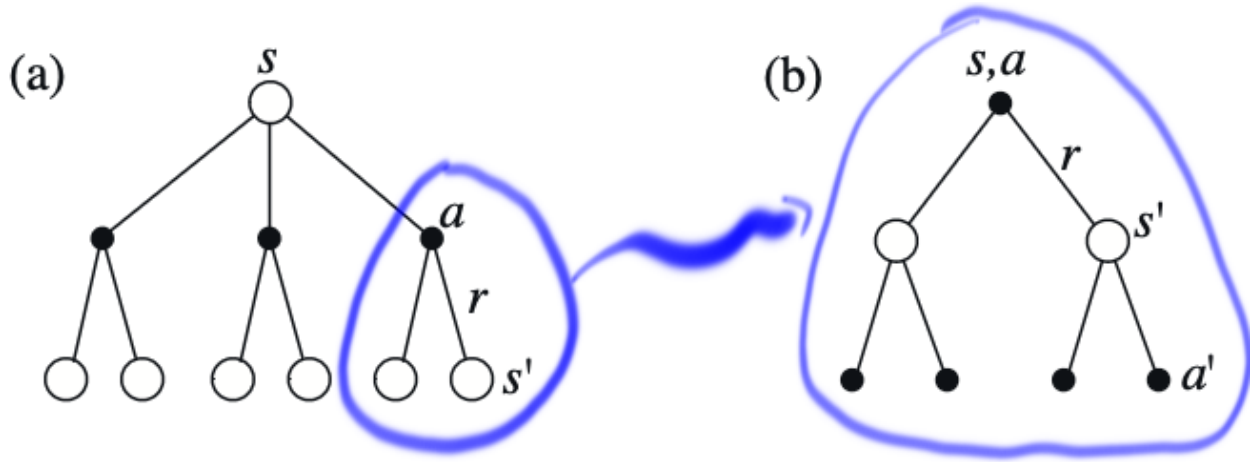
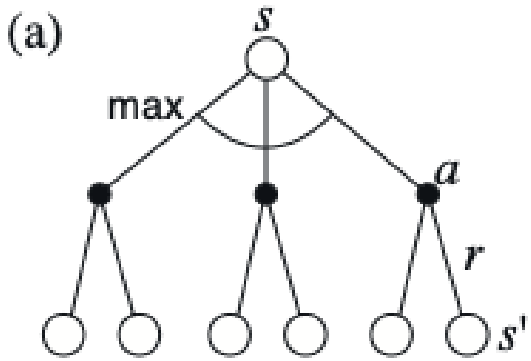


Figure 3.4: Backup diagrams for (a) v_π and (b) q_π.

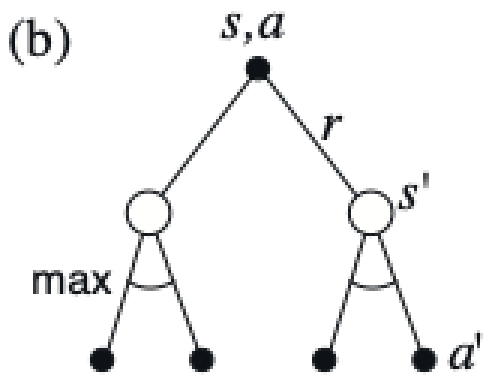
Ecuación de Bellman (2/2) v_*



Backup diagram for (a) v_*

$$v_*(s) = \max_a q_*(s, a)$$

Ecuación de Bellman (2/2) q_*



Backup diagram for (b) q_*

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

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

Policy iteration

Value iteration

Mejoramiento de una política

- Mejorar π cambiando la acción para un solo estado s.

$$\pi'(s') = \begin{cases} a, & \text{si } s' = s \\ \pi(s'), & \text{en otro caso} \end{cases}$$

Mejoramiento de una política

- Mejorar π cambiando la acción para un solo estado s.

$$\pi'(s') = \begin{cases} a, & \text{si } s' = s \\ \pi(s'), & \text{en otro caso} \end{cases}$$

- Si $q_{\pi'}(s,a) \geq q_{\pi}(s,a)$, entonces $v_{\pi'}(s) \geq v_{\pi}(s)$ para todo s.

Mejoramiento de una política

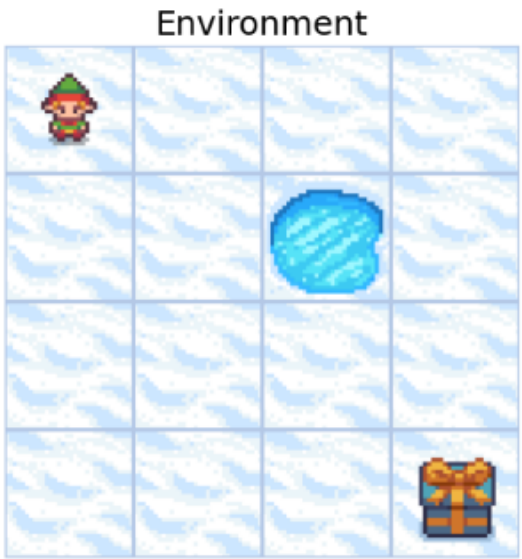
- Mejorar π cambiando la acción para un solo estado s.

$$\pi'(s') = \begin{cases} a, & \text{si } s' = s \\ \pi(s'), & \text{en otro caso} \end{cases}$$

- Si $q_{\pi'}(s,a) \geq q_{\pi}(s,a)$, entonces $v_{\pi'}(s) \geq v_{\pi}(s)$ para todo s.
- Mejoramiento avaro dada π:

$$\begin{aligned} \pi'(s) &= \arg\max_a q_\pi(s,a) \\ &= \arg\max_a \sum_{s'} \left(p(s'|s,a) \left[r + \gamma v_\pi(s') \right] \right) \end{aligned}$$

Ejemplo ilustrativo (2/2)



Lago congelado

Adelante: $\frac{1}{3}$, Izquierda: $\frac{1}{3}$, Derecha: $\frac{1}{3}$,

Recompensa: 1 el regalo.

Descuento: 0.9

MACC
Matemáticas Aplicadas y Ciencias de la Computación

El valor aumenta

$$v_{k+1}(s) = q_{\pi_{k+1}}(s, \pi_{k+1}(s))$$

☞ Por definición de $v_{k+1}(s)$

El valor aumenta

$$\begin{aligned} v_{k+1}(s) &= q_{\pi_{k+1}}(s, \pi_{k+1}(s)) \\ &= \max_a q_{\pi_k}(s, a) \end{aligned}$$

☞ Por definición de $\pi_{k+1}(s) = \arg\max_a q_{\pi_k}(s, a)$

El valor aumenta

$$\begin{aligned} v_{k+1}(s) &= q_{\pi_{k+1}}(s, \pi_{k+1}(s)) \\ &= \max_a q_{\pi_k}(s, a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \end{aligned}$$

☞ Por definición de máx

MACC
Matemáticas Aplicadas y Ciencias de la Computación

El valor aumenta

$$\begin{aligned} v_{k+1}(s) &= q_{\pi_{k+1}}(s, \pi_{k+1}(s)) \\ &= \max_a q_{\pi_k}(s, a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &= v_{\pi_k}(s) \end{aligned}$$

☞ Por definición de $v_k(s)$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

Policy iteration

Value iteration

Policy iteration

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \ldots \pi^* \xrightarrow{E} v_* \xrightarrow{I} \pi^*$$

$\pi_k \xrightarrow{E} v_{\pi_k}$

Encontrar $v_{\pi_k}(s)$ para todo s

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Policy iteration

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \ldots \pi^* \xrightarrow{E} v_* \xrightarrow{I} \pi^*$$

$\pi_k \xrightarrow{E} v_{\pi_k}$

Encontrar $v_{\pi_k}(s)$ para todo s

$v_{\pi_k} \xrightarrow{I} \pi_k$

Mejorar π_k con base en $v_{\pi_k}(s)$ para todo s

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Policy iteration

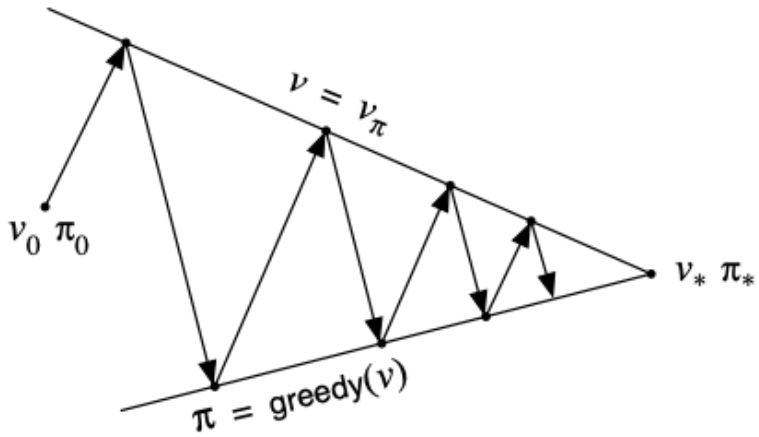
$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \ldots \pi^* \xrightarrow{E} v_* \xrightarrow{I} \pi^*$$

$\pi_k \xrightarrow{E} v_{\pi_k}$

Encontrar $v_{\pi_k}(s)$ para todo s

$v_{\pi_k} \xrightarrow{I} \pi_k$

Mejorar π_k con base en $v_{\pi_k}(s)$ para todo s



Policy iteration — pseudocódigo —

Algorithm 2: Policy iteration

Data: una política π
Result: una política óptima π^*
`// 1. Inicialización:`
$V(s) \leftarrow 0$ para cada estado s ;
policy_stable $\leftarrow$ `False`;
repeat
  `// 2. Evaluación de política:`
  $V \leftarrow policy\_evaluation(\pi)$;
  `// 3. Mejoramiento de política:`
  policy_stable $\leftarrow$ `True`;
  for *cada estado* s **do**
    $a \leftarrow \pi(s)$;
    $\pi(s) \leftarrow \arg\max_a \sum_{s'} \left(p(s'|s,a)\left[r + \gamma V(s')\right]\right)$;
    if $a \neq \pi(s)$ **then**
      policy_stable $\leftarrow$ `False`;
    end
  end
until *policy_stable*;

Contenido

Evaluación de políticas

Políticas óptimas

Mejoramiento

Policy iteration

Value iteration

Value iteration (1/3)

- Truncar la evaluación de la política después de una iteración para cada estado.

Value iteration (1/3)

- Truncar la evaluación de la política después de una iteración para cada estado.
- Se combina el mejoramiento de la política con la evaluación truncada de la política:

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

Value iteration (1/3)

- Truncar la evaluación de la política después de una iteración para cada estado.
- Se combina el mejoramiento de la política con la evaluación truncada de la política:

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

- Se puede demostrar que la sucesión $\{v_k\}$ converge a v_*.

Value iteration (1/3)

- Truncar la evaluación de la política después de una iteración para cada estado.
- Se combina el mejoramiento de la política con la evaluación truncada de la política:

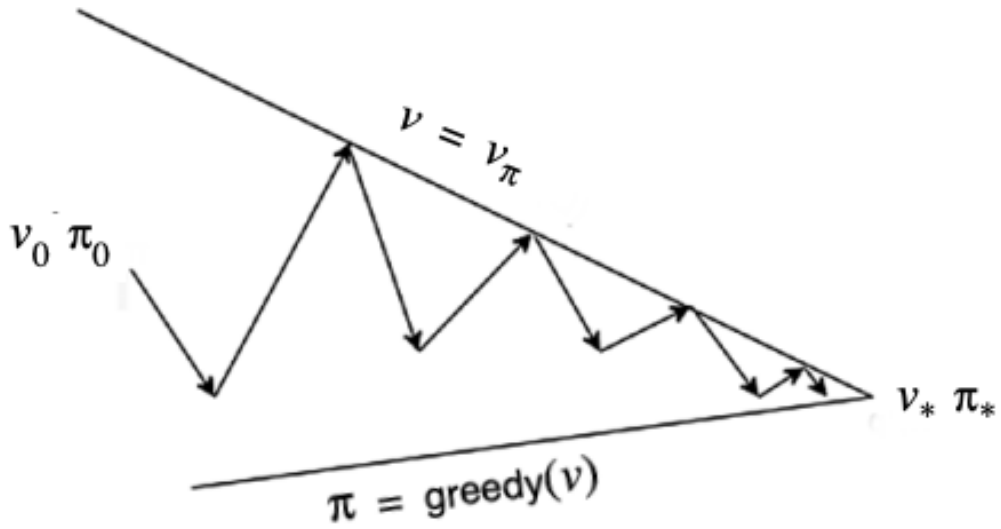
$$v_{k+1}(s) = \max_a \sum_{s'} \left(p(s'|s,a) \left[r + \gamma v_k(s') \right] \right)$$

- Se puede demostrar que la sucesión $\{v_k\}$ converge a v_*.
- Finalmente, se obtiene π_*:

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

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Value iteration (2/3)



Value iteration (3/3)

Algorithm 3: Value iteration

Result: una política óptima π^*

$V(s) \leftarrow 0$ para cada estado s;

repeat

  $\Delta \leftarrow 0$;

  for *cada estado* s **do**

    $v \leftarrow V(s)$;

    $V(s) \leftarrow \max_a \sum_{s'} \Big(p(s'|s,a)\Big[r + \gamma V(s')\Big]\Big)$;

    $\Delta \leftarrow \max\Big(\Delta, |v - V(s)|\Big)$;

  end

until $\Delta < \theta$;

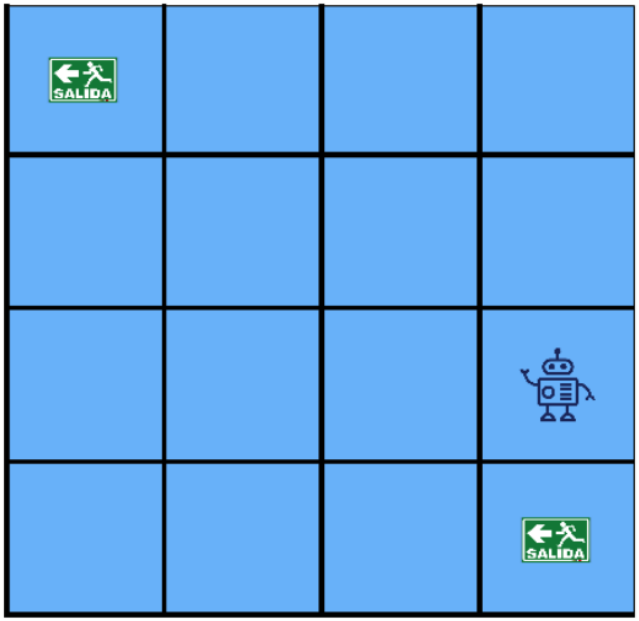
for *cada estado* s **do**

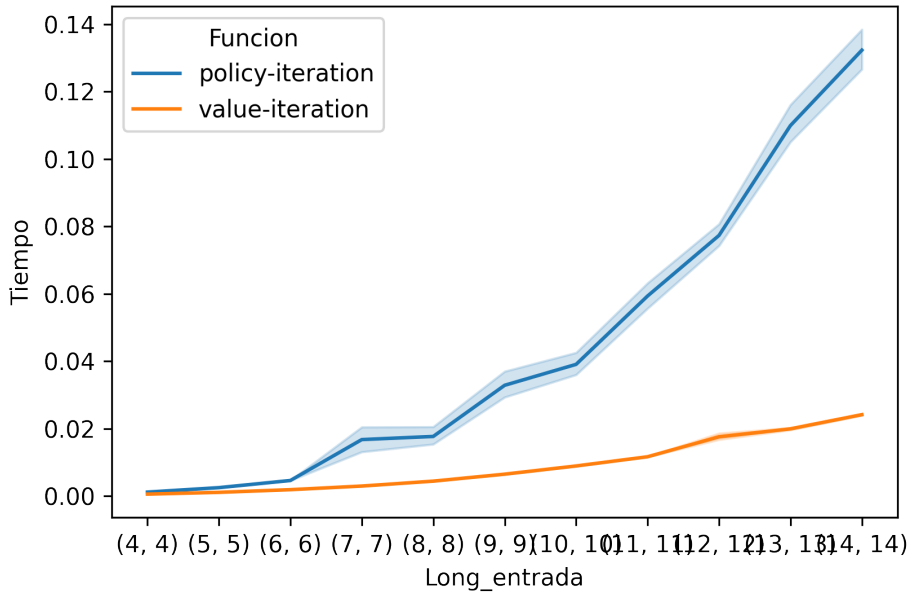
  $\pi^*(s) \leftarrow \arg\max_a \sum_{s'} \Big(p(s'|s,a)\Big[r + \gamma V(s')\Big]\Big)$;

end

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Comparación de tiempos





MACC
Matemáticas Aplicadas y Ciencias de la Computación

Take away

En esta sesión usted aprendió:

- Usar programación dinámica para evaluar una política.
- Definir el concepto de política óptima.
- Definir un proceso de mejora de una política.
- Usar los algoritmos de Policy iteration y Value iteration para buscar una política óptima.