Agentes Inteligentes

Franz Mayr
mayr@ort.edu.uy

Universidad ORT Uruguay
19 de marzo de 2023

3. Procesos de Decisión de Markov

Procesos de Decisión de Markov

En los bandidos de k brazos, estimamos $q_*(a)$ (el valor de cada acción) para decidir cuál acción ejecutar. Cada elección es independiente de las anteriores.

Procesos de Decisión de Markov

En los bandidos de k brazos, estimamos $q_*(a)$ (el valor de cada acción) para decidir cuál acción ejecutar. Cada elección es independiente de las anteriores.

Ahora, incorporamos el concepto de estado $s \in \mathcal{S}$, y pasamos a estimar $q_*(s, a)$ (el valor de la acción a en el estado s) y $v_*(s)$ (el valor del estado s).

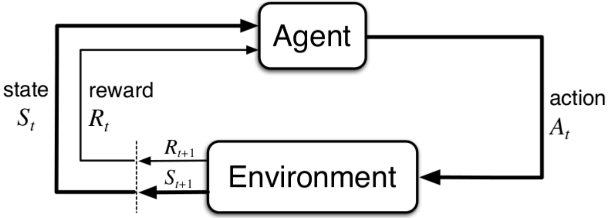
Procesos de Decisión de Markov

En los bandidos de k brazos, estimamos $q_*(a)$ (el valor de cada acción) para decidir cuál acción ejecutar. Cada elección es independiente de las anteriores.

Ahora, incorporamos el concepto de estado $s \in \mathcal{S}$, y pasamos a estimar $q_*(s, a)$ (el valor de la acción a en el estado s) y $v_*(s)$ (el valor del estado s).

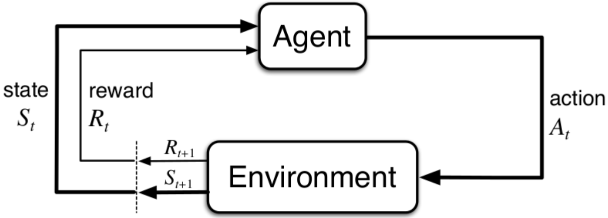
Los Procesos de Decisión de Markov (MDP) son un formalismo útil para modelar el proceso de toma de decisiones secuenciales, en los que cada decisión afecta a las siguientes.

Procesos de Decisión de Markov



$$S_t \in \mathcal{S};\ A_t \in \mathcal{A}(S_t);\ R_t \in \mathcal{R} \subset \mathbb{R}$$

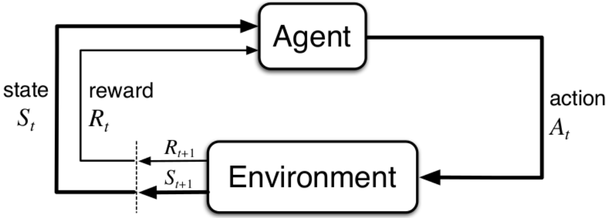
Procesos de Decisión de Markov



$$S_t \in \mathcal{S};\ A_t \in \mathcal{A}(S_t);\ R_t \in \mathcal{R} \subset \mathbb{R}$$

El tiempo transcurre de manera discreta: $t = 0, 1, 2, 3, \ldots.$

Procesos de Decisión de Markov



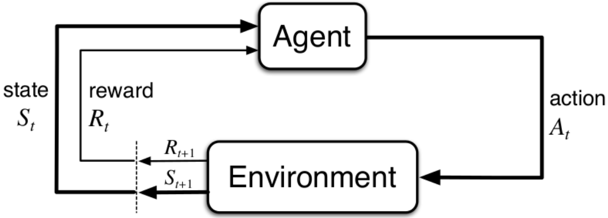
$$S_t \in \mathcal{S};\ A_t \in \mathcal{A}(S_t);\ R_t \in \mathcal{R} \subset \mathbb{R}$$

El tiempo transcurre de manera discreta: $t = 0, 1, 2, 3, \ldots$.

En un MDP finito, los conjuntos $\mathcal{S}$, $\mathcal{A}$ y $\mathcal{R}$ son finitos.

En particular, $\mathcal{R}$ es un conjunto finito de números reales.

Procesos de Decisión de Markov



$$S_t \in \mathcal{S};\ A_t \in \mathcal{A}(S_t);\ R_t \in \mathcal{R} \subset \mathbb{R}$$

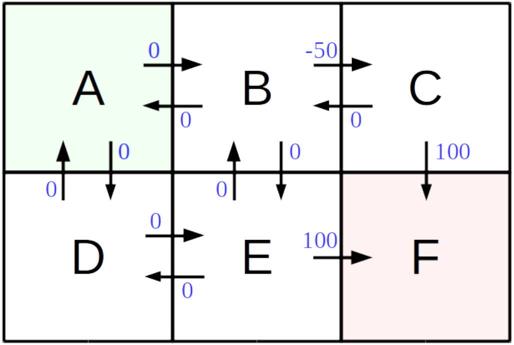
El tiempo transcurre de manera discreta: $t = 0, 1, 2, 3, \ldots$.

En un MDP finito, los conjuntos $\mathcal{S}$, $\mathcal{A}$ y $\mathcal{R}$ son finitos.

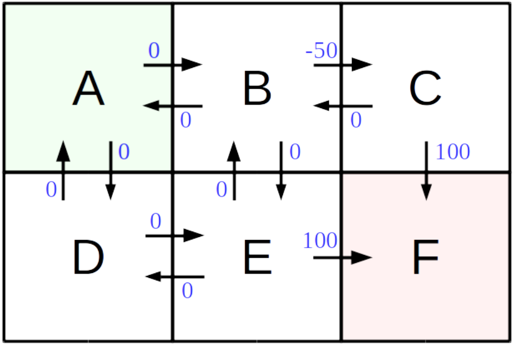
En particular, $\mathcal{R}$ es un conjunto finito de números reales.

Una trayectoria se define como una secuencia $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$. Por ejemplo, R_2 y S_2 son el efecto de haber ejecutado la acción A_1 en el estado S_1.

Ejemplo: Grid world

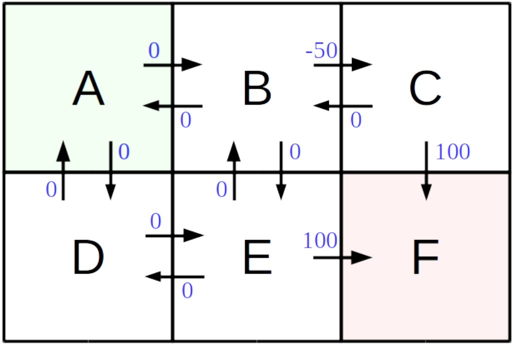


Ejemplo: Grid world



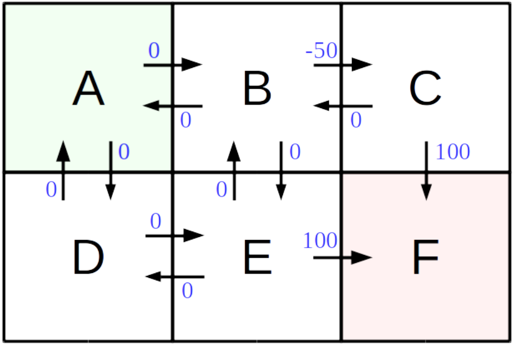
$\mathcal{S} = \{\mathsf{A, B, C, D, E, F}\}$

Ejemplo: Grid world



$\mathcal{S} = \{\text{A, B, C, D, E, F}\}$ $\mathcal{R} = \{0, -50, 100\}$

Ejemplo: Grid world



$\mathcal{S} = \{\mathsf{A, B, C, D, E, F}\}$

$\mathcal{R} = \{0, -50, 100\}$

$\mathcal{A}(\mathsf{A}) = \{\downarrow, \rightarrow\}$
$\mathcal{A}(\mathsf{B}) = \{\leftarrow, \downarrow, \rightarrow\}$
$\mathcal{A}(\mathsf{C}) = \{\leftarrow, \downarrow\}$

$\mathcal{A}(\mathsf{D}) = \{\uparrow, \rightarrow\}$
$\mathcal{A}(\mathsf{E}) = \{\leftarrow, \uparrow, \rightarrow\}$
$\mathcal{A}(\mathsf{F}) = \varnothing$ (estado terminal)

Procesos de Decisión de Markov

<u>DEFINICIÓN</u>: Probabilidad de llegar al estado s' con recompensa r, luego de ejecutar la acción a en el estado s:

$$\boxed{p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

Procesos de Decisión de Markov

<u>Definición</u>: Probabilidad de llegar al estado s' con recompensa r, luego de ejecutar la acción a en el estado s:

$$\boxed{p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

La función $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ especifica la distribución de probabilidad de los efectos de cada par (s, a):

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1 \quad \text{para cada } s \in \mathcal{S}, a \in \mathcal{A}(s)$$

Procesos de Decisión de Markov

DEFINICIÓN: Probabilidad de llegar al estado s' con recompensa r, luego de ejecutar la acción a en el estado s:

$$\boxed{p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

La función $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ especifica la distribución de probabilidad de los efectos de cada par (s, a):

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1 \quad \text{para cada } s \in \mathcal{S}, a \in \mathcal{A}(s)$$

La función p caracteriza por completo la dinámica del ambiente. La probabilidad de cada posible valor de (S_t, R_t) depende sólo de (S_{t-1}, A_{t-1}), y no de estados o acciones anteriores.

Esto se conoce como la propiedad de Markov. Restringe cómo definimos nuestros estados, que deberían incluir toda la información del pasado que será relevante para el futuro.

Ejemplo: Grid world resbaladizo (estocástico)

Supongamos que el suelo está resbaladizo. Al ejecutar cualquier acción, con cierta probabilidad, podemos terminar en un estado distinto del deseado y/o recibir una penalidad.

Ejemplo: Grid world resbaladizo (estocástico)

Supongamos que el suelo está resbaladizo. Al ejecutar cualquier acción, con cierta probabilidad, podemos terminar en un estado distinto del deseado y/o recibir una penalidad.

Formalmente, para el estado A y la acción $\rightarrow$ podríamos tener, por ejemplo:

$$
\begin{aligned}
p(\mathsf{B}, 0 \,|\, \mathsf{A}, \rightarrow) &= 0{,}9 \\
p(\mathsf{B}, -1 \,|\, \mathsf{A}, \rightarrow) &= 0{,}04 \\
p(\mathsf{D}, 0 \,|\, \mathsf{A}, \rightarrow) &= 0{,}04 \\
p(\mathsf{D}, -1 \,|\, \mathsf{A}, \rightarrow) &= 0{,}02
\end{aligned}
$$

Ejemplo: Grid world resbaladizo (estocástico)

Supongamos que el suelo está resbaladizo. Al ejecutar cualquier acción, con cierta probabilidad, podemos terminar en un estado distinto del deseado y/o recibir una penalidad.

Formalmente, para el estado A y la acción $\rightarrow$ podríamos tener, por ejemplo:

$$
\begin{aligned}
p(\mathsf{B}, 0 \,|\, \mathsf{A}, \rightarrow) &= 0{,}9 \\
p(\mathsf{B}, -1 \,|\, \mathsf{A}, \rightarrow) &= 0{,}04 \\
p(\mathsf{D}, 0 \,|\, \mathsf{A}, \rightarrow) &= 0{,}04 \\
p(\mathsf{D}, -1 \,|\, \mathsf{A}, \rightarrow) &= 0{,}02
\end{aligned}
$$

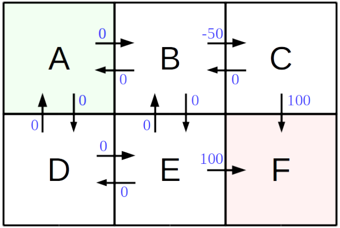
Y para el estado C y la acción $\downarrow$, podría ser:

$$
\begin{aligned}
p(\mathsf{F}, 100 \,|\, \mathsf{C}, \downarrow) &= 0{,}98 \\
p(\mathsf{B}, -1 \,|\, \mathsf{C}, \downarrow) &= 0{,}02
\end{aligned}
$$

etc.

Ejemplo: Grid world determinístico

Es más sencillo usar ejemplos con ambientes determinísticos; son más intuitivos, fáciles de entender y de visualizar:

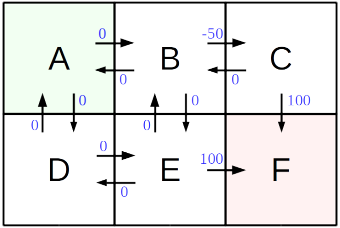


$$p(\mathsf{B}, 0 \mid \mathsf{A}, \rightarrow) = 1$$
$$p(s, r \mid \mathsf{A}, \rightarrow) = 0 \;\; \forall (s, r) \neq (\mathsf{B}, 0)$$

$$p(\mathsf{F}, 100 \mid \mathsf{C}, \downarrow) = 1$$
$$p(s, r \mid \mathsf{C}, \downarrow) = 0 \;\; \forall (s, r) \neq (\mathsf{F}, 100)$$

Ejemplo: Grid world determinístico

Es más sencillo usar ejemplos con ambientes determinísticos; son más intuitivos, fáciles de entender y de visualizar:



$$p(\mathsf{B}, 0 \mid \mathsf{A}, \rightarrow) = 1$$
$$p(s, r \mid \mathsf{A}, \rightarrow) = 0 \ \ \forall (s, r) \neq (\mathsf{B}, 0)$$

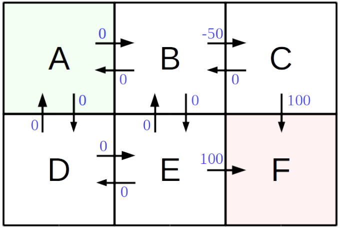
$$p(\mathsf{F}, 100 \mid \mathsf{C}, \downarrow) = 1$$
$$p(s, r \mid \mathsf{C}, \downarrow) = 0 \ \ \forall (s, r) \neq (\mathsf{F}, 100)$$

Por eso, vamos a usar casi siempre ejemplos determinísticos.

Pero no debemos perder de vista el caso general estocástico, con la caracterización completa del ambiente dada por $p(s', r \mid s, a)$.

Ejemplo: Grid world determinístico

Es más sencillo usar ejemplos con ambientes determinísticos; son más intuitivos, fáciles de entender y de visualizar:



$$p(\mathsf{B}, 0 \mid \mathsf{A}, \rightarrow) = 1$$
$$p(s, r \mid \mathsf{A}, \rightarrow) = 0 \;\; \forall (s, r) \neq (\mathsf{B}, 0)$$

$$p(\mathsf{F}, 100 \mid \mathsf{C}, \downarrow) = 1$$
$$p(s, r \mid \mathsf{C}, \downarrow) = 0 \;\; \forall (s, r) \neq (\mathsf{F}, 100)$$

Por eso, vamos a usar casi siempre ejemplos determinísticos.

Pero no debemos perder de vista el caso general estocástico, con la caracterización completa del ambiente dada por $p(s', r \mid s, a)$.

Observación: Varios trabajos y libros definen funciones de transición $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ y de recompensa $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. Notar que esas funciones son determinísticas; por lo tanto, usar $p(s', r \mid s, a)$ es más general.

Procesos de Decisión de Markov

$$\boxed{p(s', r \mid s, a) \ \doteq \ \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

A partir de p, se puede computar cualquier cosa que querramos saber del ambiente. Por ejemplo:

Procesos de Decisión de Markov

$$\boxed{p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

A partir de p, se puede computar cualquier cosa que querramos saber del ambiente. Por ejemplo:

Probabilidades de transición entre estados:

$$p(s' \mid s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r \mid s, a)$$

¿Qué pasa si las transiciones son determinísticas?

Procesos de Decisión de Markov

$$\boxed{p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}}$$

A partir de p, se puede computar cualquier cosa que querramos saber del ambiente. Por ejemplo:

Probabilidades de transición entre estados:

$$p(s' \mid s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r \mid s, a)$$

¿Qué pasa si las transiciones son determinísticas?

Recompensa esperada para un par estado-acción:

$$r(s, a) \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a)$$

¿Qué pasa si las recompensas son determinísticas?

Retornos y episodios

<u>Definición</u>: El retorno, o ganancia acumulada esperada, a partir del instante de tiempo t:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$$

donde T es el último instante de tiempo del episodio.

Retornos y episodios

<u>Definición</u>: El retorno, o ganancia acumulada esperada, a partir del instante de tiempo t:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$$

donde T es el último instante de tiempo del episodio.

Ejemplos de episodios: partida de ajedrez; (intento de) escape de un laberinto; una vida en un juego de Atari.

Cada episodio comienza de manera independiente del anterior, y concluye en un estado terminal.

Retornos y episodios

<u>Definición</u>: El retorno, o ganancia acumulada esperada, a partir del instante de tiempo t:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$$

donde T es el último instante de tiempo del episodio.

Ejemplos de episodios: partida de ajedrez; (intento de) escape de un laberinto; una vida en un juego de Atari.

Cada episodio comienza de manera independiente del anterior, y concluye en un estado terminal.

Las tareas pueden ser episódicas cuando tiene sentido hablar de episodios, o bien continuas cuando la tarea prosigue, sin división en episodios.

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Si $\gamma < 1$, la suma infinita tiene un valor finito, siempre y cuando la secuencia de recompensas esté acotada.

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Si $\gamma < 1$, la suma infinita tiene un valor finito, siempre y cuando la secuencia de recompensas esté acotada.
- Si $\gamma = 0$, el agente es miope: sólo le importa maximizar las recompensas inmediatas.

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Si $\gamma < 1$, la suma infinita tiene un valor finito, siempre y cuando la secuencia de recompensas esté acotada.
- Si $\gamma = 0$, el agente es miope: sólo le importa maximizar las recompensas inmediatas.
- G_t puede plantearse de manera recursiva:
  $G_t \doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots$

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Si $\gamma < 1$, la suma infinita tiene un valor finito, siempre y cuando la secuencia de recompensas esté acotada.
- Si $\gamma = 0$, el agente es miope: sólo le importa maximizar las recompensas inmediatas.
- G_t puede plantearse de manera recursiva:

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots \\ &= R_{t+1} + \gamma \big(R_{t+2} + \gamma\, R_{t+3} + \gamma^2 R_{t+4} + \ldots \big) \end{aligned}$$

Retornos y episodios

En tareas continuas, la definición de G_t es problemática, porque $T = \infty$. Introducimos entonces el concepto de descuento γ.

$$G_t \doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Si $\gamma < 1$, la suma infinita tiene un valor finito, siempre y cuando la secuencia de recompensas esté acotada.
- Si $\gamma = 0$, el agente es miope: sólo le importa maximizar las recompensas inmediatas.
- G_t puede plantearse de manera recursiva:

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots \\ &= R_{t+1} + \gamma\big(R_{t+2} + \gamma\, R_{t+3} + \gamma^2 R_{t+4} + \ldots\big) \\ &= R_{t+1} + \gamma\, G_{t+1} \qquad\qquad (t < T) \end{aligned}$$

Políticas y funciones de valor

<u>Definición</u>: Una política $\pi(a|s)$ es la probabilidad de que el agente ejecute la acción $a \in \mathcal{A}(s)$ en el estado $s \in \mathcal{S}$.

En general, consideramos políticas estocásticas. Si π determina una única acción a en cada estado s, π es determinística.

Políticas y funciones de valor

DEFINICIÓN: Una política $\pi(a|s)$ es la probabilidad de que el agente ejecute la acción $a \in \mathcal{A}(s)$ en el estado $s \in \mathcal{S}$.

En general, consideramos políticas estocásticas. Si π determina una única acción a en cada estado s, π es determinística.

DEFINICIÓN: Valor del estado s, según π:

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

Es el retorno esperado de seguir la política π a partir de $s \in \mathcal{S}$.

Políticas y funciones de valor

<u>Definición</u>: Una política $\pi(a|s)$ es la probabilidad de que el agente ejecute la acción $a \in \mathcal{A}(s)$ en el estado $s \in \mathcal{S}$.

En general, consideramos políticas estocásticas. Si π determina una única acción a en cada estado s, π es determinística.

<u>Definición</u>: Valor del estado s, según π:

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

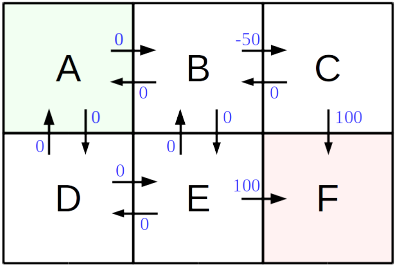
Es el retorno esperado de seguir la política π a partir de $s \in \mathcal{S}$.

<u>Definición</u>: Valor de la acción a en el estado s, según π:

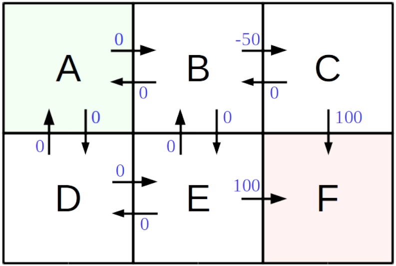
$$q_\pi(s, a) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

Es el retorno esperado de ejecutar la acción $a \in \mathcal{A}(s)$ en $s \in \mathcal{S}$, y después seguir la política π.

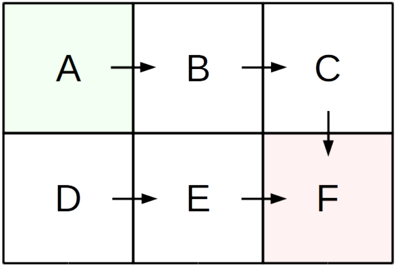
Ambiente (descuento $\gamma = 0{,}9$)



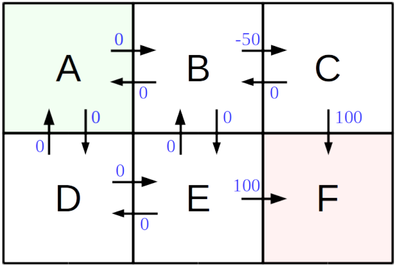
Ambiente (descuento $\gamma = 0{,}9$)



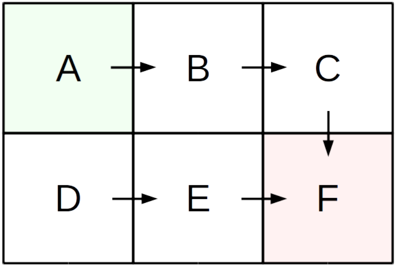
Política π (determinística)



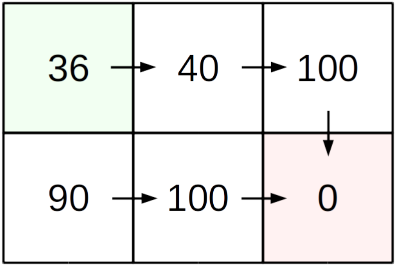
Ambiente (descuento $\gamma = 0{,}9$)



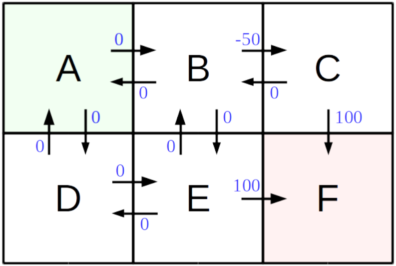
Política π (determinística)



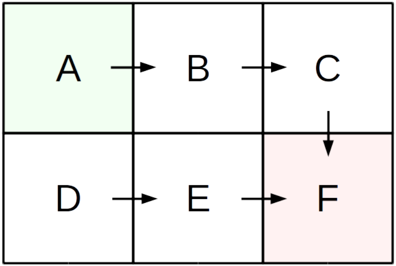
v_π: Valor de los estados según π



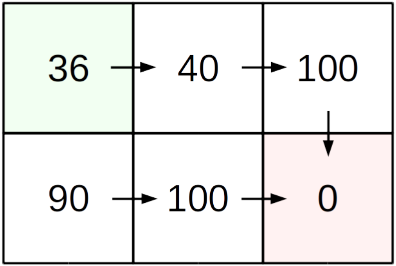
Ambiente (descuento $\gamma = 0{,}9$)



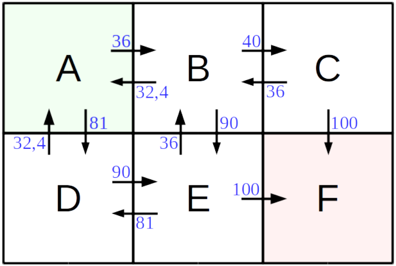
Política π (determinística)



v_π: Valor de los estados según π



q_π: Valor de las acciones según π



Políticas y funciones de valor

Propiedad interesante y útil:

$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$

Políticas y funciones de valor

Propiedad interesante y útil:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma\, G_{t+1} \mid S_t = s]
\end{aligned}
$$

Políticas y funciones de valor

Propiedad interesante y útil:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma\, G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a)\Big[r + \gamma\, \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']\Big]
\end{aligned}
$$

Políticas y funciones de valor

Propiedad interesante y útil:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma\, G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a) \Big[r + \gamma\, \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r \mid s, a) \Big[r + \gamma\, v_\pi(s') \Big] \qquad \forall s \in \mathcal{S}
\end{aligned}
$$

Esto se conoce como la ecuación de Bellman para v_π.

Ecuación de Bellman para v_π

Describe una relación recursiva entre el valor de un estado y el valor de los posibles estados sucesores.

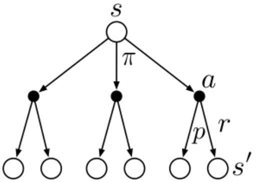
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma\, v_\pi(s')\Big]$$

Ecuación de Bellman para v_π

Describe una relación recursiva entre el valor de un estado y el valor de los posibles estados sucesores.

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

El valor de s se computa como la suma de los valores descontados de cada estado sucesor s', más las recompensas esperadas de cada acción a.

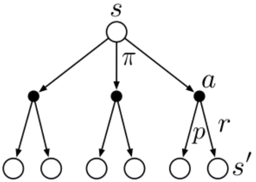


Ecuación de Bellman para v_π

Describe una relación recursiva entre el valor de un estado y el valor de los posibles estados sucesores.

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

El valor de s se computa como la suma de los valores descontados de cada estado sucesor s', más las recompensas esperadas de cada acción a.



Ejercicio: ¿Cómo sería la ecuación de Bellman para q_π?

Políticas óptimas

<u>Definición</u>: Dadas dos políticas π, π', decimos que $\pi \geq \pi'$ sii $v_\pi(s) \geq v_{\pi'}(s) \ \ \forall s \in \mathcal{S}$.

Políticas óptimas

Definición: Dadas dos políticas π, π', decimos que $\pi \geq \pi'$ sii $v_\pi(s) \geq v_{\pi'}(s) \;\; \forall s \in \mathcal{S}$.

Definición: Una política π_* es óptima si $\pi_* \geq \pi'$ para toda política π'.

Políticas óptimas

<u>DEFINICIÓN</u>: Dadas dos políticas π, π', decimos que $\pi \geq \pi'$ sii $v_\pi(s) \geq v_{\pi'}(s) \ \ \forall s \in \mathcal{S}$.

<u>DEFINICIÓN</u>: Una política π_* es óptima si $\pi_* \geq \pi'$ para toda política π'.

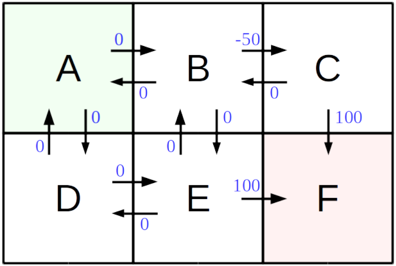
Luego, podemos definir las funciones de valor óptimas:

$$v_*(s) \doteq \max_\pi v_\pi(s) \quad \forall s \in \mathcal{S}$$

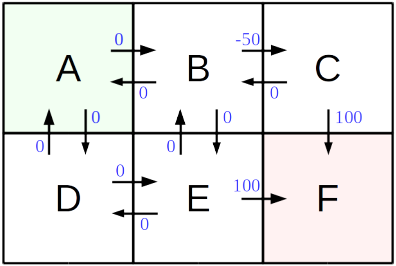
$$q_*(s,a) \doteq \max_\pi q_\pi(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$$

$$q_*(s,a) = \mathbb{E}\big[R_{t+1} + \gamma\, v_*(S_{t+1}) \,\big|\, S_t = s, A_t = a\big]$$

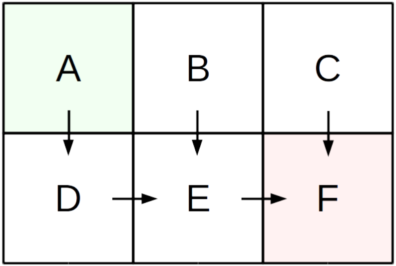
Ambiente (descuento $\gamma = 0{,}9$)



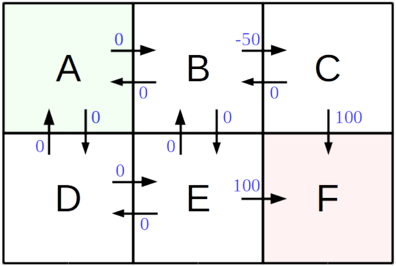
Ambiente (descuento $\gamma = 0{,}9$)



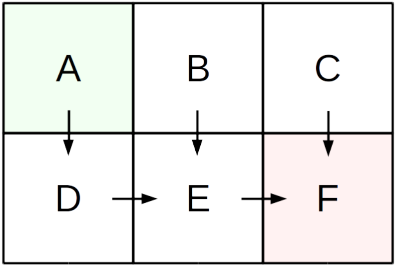
Política óptima π_*



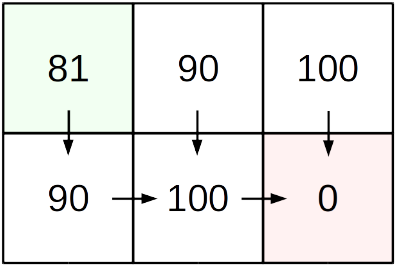
Ambiente (descuento $\gamma = 0{,}9$)



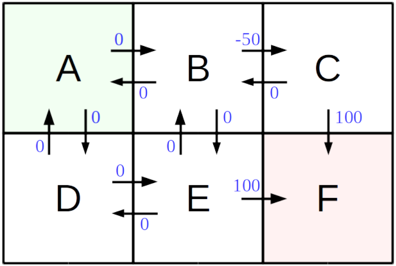
Política óptima π_*



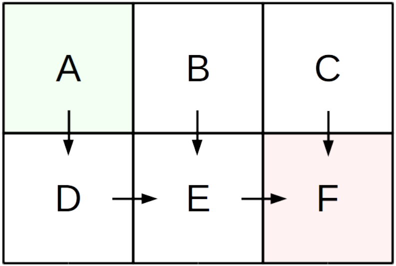
v_*: Valor de los estados según π_*



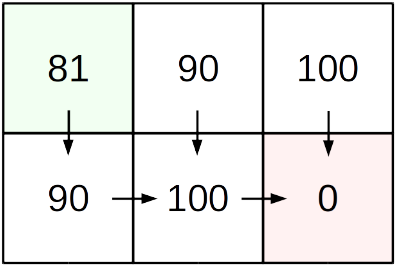
Ambiente (descuento $\gamma = 0{,}9$)



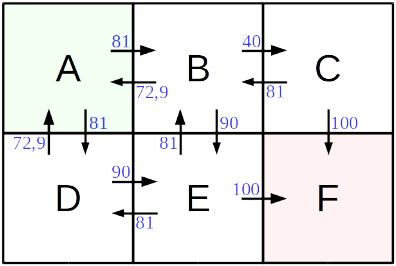
Política óptima π_*



v_*: Valor de los estados según π_*



q_*: Valor de las acciones según π_*



Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a) \\
&= \max_a \mathbb{E}_{\pi_*}\left[G_t \,\middle|\, S_t = s, A_t = a\right]
\end{aligned}
$$

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}\left[G_t \,\middle|\, S_t = s, A_t = a\right] \\ &= \max_a \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma\, G_{t+1} \,\middle|\, S_t = s, A_t = a\right] \end{aligned}$$

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a) \\
&= \max_a \mathbb{E}_{\pi_*}\big[G_t \,\big|\, S_t = s, A_t = a\big] \\
&= \max_a \mathbb{E}_{\pi_*}\big[R_{t+1} + \gamma\, G_{t+1} \,\big|\, S_t = s, A_t = a\big] \\
&= \max_a \mathbb{E}\big[R_{t+1} + \gamma\, v_*(S_{t+1}) \,\big|\, S_t = s, A_t = a\big]
\end{aligned}$$

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}\left[G_t \,\middle|\, S_t = s, A_t = a\right] \\
&= \max_a \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma\, G_{t+1} \,\middle|\, S_t = s, A_t = a\right] \\
&= \max_a \mathbb{E}\left[R_{t+1} + \gamma\, v_*(S_{t+1}) \,\middle|\, S_t = s, A_t = a\right] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma\, v_*(s')\right]
\end{aligned}$$

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$\begin{aligned} v_*(s) &= \max_{a\in\mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}\big[G_t \,\big|\, S_t = s, A_t = a\big] \\ &= \max_a \mathbb{E}_{\pi_*}\big[R_{t+1} + \gamma\, G_{t+1} \,\big|\, S_t = s, A_t = a\big] \\ &= \max_a \mathbb{E}\big[R_{t+1} + \gamma\, v_*(S_{t+1}) \,\big|\, S_t = s, A_t = a\big] \\ &= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma\, v_*(s')\big] \end{aligned}$$

Ecuación de optimalidad de Bellman para q_*:

Ecuaciones de optimalidad de Bellman

Ecuación de optimalidad de Bellman para v_*:

$$\begin{aligned}
v_*(s) &= \max_{a\in\mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}\big[G_t \,\big|\, S_t = s, A_t = a\big] \\
&= \max_a \mathbb{E}_{\pi_*}\big[R_{t+1} + \gamma\, G_{t+1} \,\big|\, S_t = s, A_t = a\big] \\
&= \max_a \mathbb{E}\big[R_{t+1} + \gamma\, v_*(S_{t+1}) \,\big|\, S_t = s, A_t = a\big] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma\, v_*(s')\big]
\end{aligned}$$

Ecuación de optimalidad de Bellman para q_*:

$$\begin{aligned}
q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma\, \max_{a'} q_*(S_{t+1}, a') \,\Big|\, S_t = s, A_t = a\Big] \\
&= \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma\, \max_{a'} q_*(s', a')\Big]
\end{aligned}$$

Ecuaciones de optimalidad de Bellman

Para un problema específico, resolver explícitamente las ecuaciones de optimalidad de Bellman nos permitiría encontrar una política óptima.

Ecuaciones de optimalidad de Bellman

Para un problema específico, resolver explícitamente las ecuaciones de optimalidad de Bellman nos permitiría encontrar una política óptima.

Sin embargo, esta estrategia rara vez es factible, porque se apoya en tres suposiciones improbables en la práctica:

(1) conocemos con precisión las dinámicas del ambiente;

(2) tenemos suficientes recursos computacionales;

(3) el problema cumple con la propiedad de Markov.

Ecuaciones de optimalidad de Bellman

Para un problema específico, resolver explícitamente las ecuaciones de optimalidad de Bellman nos permitiría encontrar una política óptima.

Sin embargo, esta estrategia rara vez es factible, porque se apoya en tres suposiciones improbables en la práctica:

(1) conocemos con precisión las dinámicas del ambiente;

(2) tenemos suficientes recursos computacionales;

(3) el problema cumple con la propiedad de Markov.

Por ejemplo, el backgammon cumple (1) y (3), pero no (2). Entonces, tardaríamos miles de años en computar v_* y q_*.

Ecuaciones de optimalidad de Bellman

Para un problema específico, resolver explícitamente las ecuaciones de optimalidad de Bellman nos permitiría encontrar una política óptima.

Sin embargo, esta estrategia rara vez es factible, porque se apoya en tres suposiciones improbables en la práctica:

(1) conocemos con precisión las dinámicas del ambiente;

(2) tenemos suficientes recursos computacionales;

(3) el problema cumple con la propiedad de Markov.

Por ejemplo, el backgammon cumple (1) y (3), pero no (2). Entonces, tardaríamos miles de años en computar v_* y q_*.

En consecuencia, necesitamos resolver en forma aproximada estas ecuaciones, para estimar v_* y q_*.

Búsqueda de políticas óptimas

Las funciones de valor óptimas v_* y q_* son desconocidas para el agente.

Búsqueda de políticas óptimas

Las funciones de valor óptimas v_* y q_* son desconocidas para el agente.

El Aprendizaje Reforzado tiene distintos métodos para estimar v_* y q_*, de modo de usar esas estimaciones para encontrar buenas políticas.

Búsqueda de políticas óptimas

Las funciones de valor óptimas v_* y q_* son desconocidas para el agente.

El Aprendizaje Reforzado tiene distintos métodos para estimar v_* y q_*, de modo de usar esas estimaciones para encontrar buenas políticas.

Si conocemos la función $p(s', r \mid s, a)$, podremos usar métodos basados en un modelo *(model-based)*; ej.: programación dinámica.

Si no conocemos $p(s', r \mid s, a)$, deberemos usar métodos sin modelo *(model-free)*; ej.: Monte Carlo y diferencias temporales.

Resumen - Procesos de Decisión de Markov

- Estados, acciones y recompensas.
- Probabilidad $p(s', r \mid s, a)$.
- Trayectoria, retorno, episodio, estado terminal (en tareas episódicas), descuento γ (en tareas continuas).
- Política π; funciones de valor v_π y q_π.
- Ecuación de Bellman para v_π.
- Política óptima π_* y funciones de valor óptimas v_* y q_*.

Agentes Inteligentes

Franz Mayr
mayr@ort.edu.uy

Universidad ORT Uruguay
19 de marzo de 2023

4. Métodos de Programación Dinámica

Evaluación de una política π

Empecemos viendo cómo calcular la función de valor v_π para una política particular π. A esto se lo llama evaluar a π.

Evaluación de una política π

Empecemos viendo cómo calcular la función de valor v_π para una política particular π. A esto se lo llama evaluar a π.

Vimos la ecuación de Bellman para v_π:

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

Normalmente, la resolución directa de esta ecuación es impracticable. Entonces, veamos un método iterativo que vaya aproximando gradualmente v_π.

Evaluación de una política π

Empecemos viendo cómo calcular la función de valor v_π para una política particular π. A esto se lo llama evaluar a π.

Vimos la ecuación de Bellman para v_π:

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

Normalmente, la resolución directa de esta ecuación es impracticable. Entonces, veamos un método iterativo que vaya aproximando gradualmente v_π.

<u>Idea</u>: Usar un arreglo V con $|S|$ posiciones para almacenar la aproximación de $v_\pi(s)$ en la k-ésima iteración del algoritmo, y transformar la ecuación de arriba en una asignación.

Algoritmo iterativo para estimar v_π

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

Algoritmo iterativo para estimar v_π

Repetir:

$\quad \Delta \leftarrow 0$

$\quad$ Para cada $s \in \mathcal{S}$:

$\quad\quad v \leftarrow V(s)$

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

$\quad\quad \Delta \leftarrow \max\big(\Delta, |v - V(s)|\big)$

Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

Algoritmo iterativo para estimar v_π

Inicializar $V(s)$ arbitrariamente, excepto $V(\text{terminal}) = 0$.
Repetir:

$\Delta \leftarrow 0$

Para cada $s \in \mathcal{S}$:

$v \leftarrow V(s)$

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

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

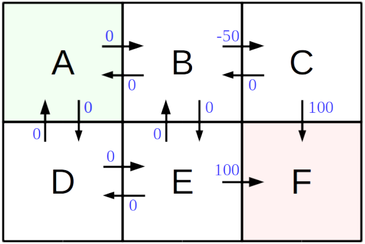
Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

Devolver $V \approx v_\pi$.

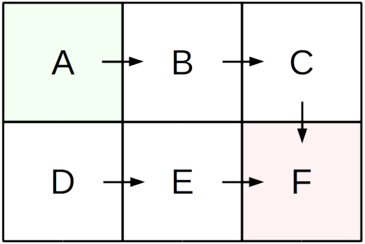
Algoritmo iterativo para estimar v_π

EJERCICIO: Ejecutar a mano en el Grid World de ejemplo el algoritmo iterativo para estimar v_π para la política indicada.

Ambiente ($\gamma = 0{,}9$)



Política π



Mejora de una política π

Una vez conseguida una estimación de v_π, lo siguiente que podemos hacer es mejorar una política π.

Supongamos que, en un estado s, hay una acción a distinta a la acción que elige π. Es decir: $a \neq \pi(s)$.

Si vale $q_\pi(s, a) > v_\pi(s)$, entonces podemos mejorar la política π si elegimos en s la acción a en lugar de $\pi(s)$.

Mejora de una política π

Una vez conseguida una estimación de v_π, lo siguiente que podemos hacer es mejorar una política π.

Supongamos que, en un estado s, hay una acción a distinta a la acción que elige π. Es decir: $a \neq \pi(s)$.

Si vale $q_\pi(s,a) > v_\pi(s)$, entonces podemos mejorar la política π si elegimos en s la acción a en lugar de $\pi(s)$.

En general, podemos definir una política π' de esta forma:

$$\pi'(s) \doteq \arg\max_a q_\pi(s,a)$$

Mejora de una política π

Una vez conseguida una estimación de v_π, lo siguiente que podemos hacer es mejorar una política π.

Supongamos que, en un estado s, hay una acción a distinta a la acción que elige π. Es decir: $a \neq \pi(s)$.

Si vale $q_\pi(s, a) > v_\pi(s)$, entonces podemos mejorar la política π si elegimos en s la acción a en lugar de $\pi(s)$.

En general, podemos definir una política π' de esta forma:

$$\begin{aligned}
\pi'(s) &\doteq \arg\max_a q_\pi(s, a) \\
&= \arg\max_a \mathbb{E}\big[R_{t+1} + \gamma\, v_\pi(S_{t+1}) \,\big|\, S_t = s, A_t = a\big]
\end{aligned}$$

Mejora de una política π

Una vez conseguida una estimación de v_π, lo siguiente que podemos hacer es mejorar una política π.

Supongamos que, en un estado s, hay una acción a distinta a la acción que elige π. Es decir: $a \neq \pi(s)$.

Si vale $q_\pi(s,a) > v_\pi(s)$, entonces podemos mejorar la política π si elegimos en s la acción a en lugar de $\pi(s)$.

En general, podemos definir una política π' de esta forma:

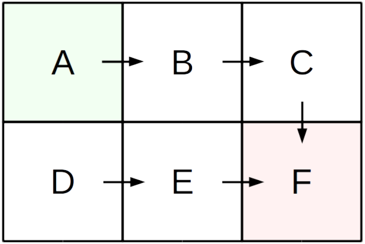
$$\begin{aligned}
\pi'(s) &\doteq \arg\max_a q_\pi(s,a) \\
&= \arg\max_a \mathbb{E}\big[R_{t+1} + \gamma\, v_\pi(S_{t+1}) \mid S_t = s, A_t = a\big] \\
&= \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma\, v_\pi(s')\big]
\end{aligned}$$

y decimos que π' es una mejora greedy de π.

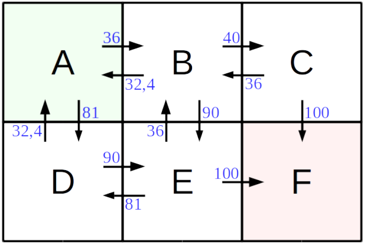
Mejora de una política π

EJERCICIO: Suponiendo que conocemos q_π, ¿cómo podemos mejorar la política π?

Política π



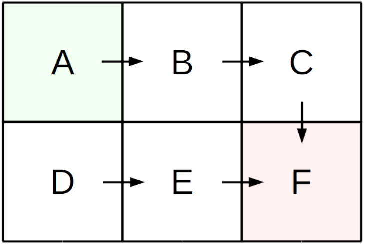
q_π



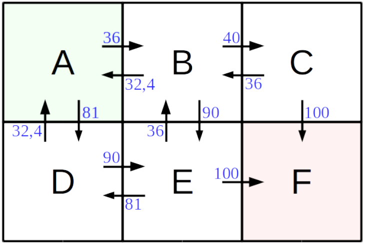
Mejora de una política π

Ejercicio: Suponiendo que conocemos q_π, ¿cómo podemos mejorar la política π?

Política π



q_π



Repetir el procedimiento, pero ahora suponiendo que lo que conocemos es $p(s', r \mid s, a)$ y v_π.

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

$$\text{evaluar } \pi\text{; mejorar } \pi\text{; repetir.}$$

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

evaluar π; mejorar π; repetir.

Dicho de otra forma:

π_0

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

$$\text{evaluar } \pi\text{; mejorar } \pi\text{; repetir.}$$

Dicho de otra forma:

$$\pi_0 \xrightarrow{\text{eval}} v_{\pi_0}$$

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

$$\text{evaluar } \pi; \text{ mejorar } \pi; \text{ repetir.}$$

Dicho de otra forma:

$$\pi_0 \xrightarrow{\text{eval}} v_{\pi_0} \xrightarrow{\text{mej}} \pi_1$$

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

evaluar π; mejorar π; repetir.

Dicho de otra forma:

$$\pi_0 \xrightarrow{\text{eval}} v_{\pi_0} \xrightarrow{\text{mej}} \pi_1 \xrightarrow{\text{eval}} v_{\pi_1} \xrightarrow{\text{mej}} \pi_2$$

Policy Iteration

Ya tenemos las dos operaciones necesarias para avanzar en forma iterativa hacia una política óptima:

$$\text{evaluar } \pi;\ \text{mejorar } \pi;\ \text{repetir.}$$

Dicho de otra forma:

$$\pi_0 \xrightarrow{\text{eval}} v_{\pi_0} \xrightarrow{\text{mej}} \pi_1 \xrightarrow{\text{eval}} v_{\pi_1} \xrightarrow{\text{mej}} \pi_2 \xrightarrow{\text{eval}} \ldots \xrightarrow{\text{mej}} \pi_* \xrightarrow{\text{eval}} v_*$$

donde $\xrightarrow{\text{eval}}$ y $\xrightarrow{\text{mej}}$ son una evaluación y una mejora de una política, respectivamente.

Algoritmo Policy Iteration para estimar π_*

1. **Inicialización**
   Inicializar $V(s) \in \mathbb{R}, \pi(s) \in \mathcal{A}(s)$ arbitrariamente, para todo $s \in \mathcal{S}$

Algoritmo Policy Iteration para estimar π_*

1. **Inicialización**
   Inicializar $V(s) \in \mathbb{R}, \pi(s) \in \mathcal{A}(s)$ arbitrariamente, para todo $s \in \mathcal{S}$

2. **Evaluación de la política actual**
   Repetir:
   - $\Delta \leftarrow 0$
   - Para cada $s \in \mathcal{S}$:
     - $v \leftarrow V(s)$
     - $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
     - $\Delta \leftarrow \max\big(\Delta, |v - V(s)|\big)$

   Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

Algoritmo Policy Iteration para estimar π_*

1. Inicialización
Inicializar $V(s) \in \mathbb{R}, \pi(s) \in \mathcal{A}(s)$ arbitrariamente, para todo $s \in \mathcal{S}$

2. Evaluación de la política actual
Repetir:
- $\Delta \leftarrow 0$
- Para cada $s \in \mathcal{S}$:
  - $v \leftarrow V(s)$
  - $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
  - $\Delta \leftarrow \max\big(\Delta, |v - V(s)|\big)$

Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

3. Mejora de la política actual
política-estable $\leftarrow$ True
Para cada $s \in \mathcal{S}$:
- *acción-vieja* $\leftarrow \pi(s)$
- $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
- Si *acción-vieja* $\neq \pi(s)$: *política-estable* $\leftarrow$ False

Si *política-estable*: Devolver $V \approx v_*$ y $\pi \approx \pi_*$. Si no, volver al paso 2.

De Policy Iteration a Value Iteration

Una seria desventaja de Policy Iteration es que hacer una evaluación completa de π_k en cada paso demora mucho tiempo.

De Policy Iteration a Value Iteration

Una seria desventaja de Policy Iteration es que hacer una evaluación completa de π_k en cada paso demora mucho tiempo.

Si transformamos la ecuación de optimalidad de Bellman para v_* en una asignación, podemos juntar cada evaluación y mejora en un solo paso:

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

De Policy Iteration a Value Iteration

Una seria desventaja de Policy Iteration es que hacer una evaluación completa de π_k en cada paso demora mucho tiempo.

Si transformamos la ecuación de optimalidad de Bellman para v_* en una asignación, podemos juntar cada evaluación y mejora en un solo paso:

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

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

Algoritmo Value Iteration para estimar π_*

$$V(s) \leftarrow \text{máx}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$$

Algoritmo Value Iteration para estimar π_*

Repetir:

$\Delta \leftarrow 0$

Para cada $s \in \mathcal{S}$:

$v \leftarrow V(s)$

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

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

Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

Algoritmo Value Iteration para estimar π_*

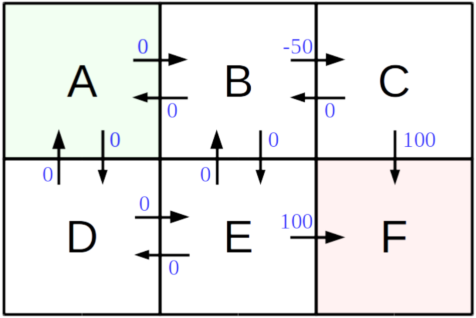
Inicializar $V(s)$ arbitrariamente $\forall s \in \mathcal{S}$, excepto $V(\text{terminal}) = 0$.
Repetir:
    $\Delta \leftarrow 0$
    Para cada $s \in \mathcal{S}$:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \text{máx}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
        $\Delta \leftarrow \text{máx}\big(\Delta, |v - V(s)|\big)$
Hasta que $\Delta < \theta$ para algún umbral θ pequeño, que determina la precisión de la aproximación.

Devolver una política determinística $\pi \approx \pi_*$ tal que:
    $\pi(s) = \arg\text{máx}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

Algoritmo Value Iteration para estimar π_*

EJERCICIO: Ejecutar a mano en el Grid World de ejemplo el algoritmo Value Iteration, para estimar π_*.

Ambiente ($\gamma = 0{,}9$)



Resumen - Métodos de Programación Dinámica

Algoritmos de programación dinámica:

- para estimar el valor de los estados según la política π (es decir, v_π)
- para mejorar una política π a partir de $p(s', r \mid s, a)$ y v_π
- Policy Iteration, para estimar π_* (muy lento)
- Value Iteration, para estimar π_*