Agentes Inteligentes

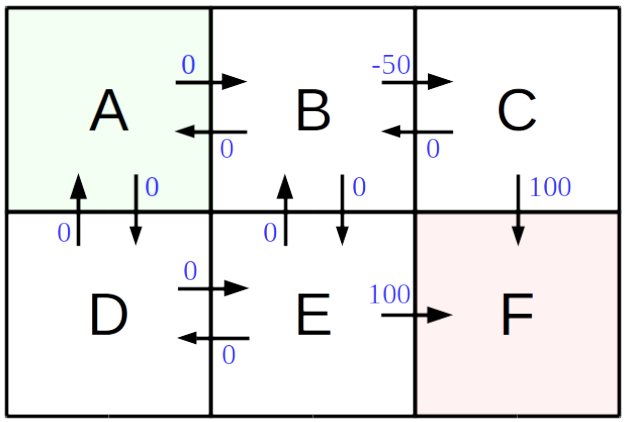
Franz Mayr
mayr@ort.edu.uy

Universidad ORT Uruguay
27 de marzo de 2023

5. Métodos Monte Carlo

Repaso: MDPs y Value Iteration

Supongamos que tenemos conocimiento completo del ambiente:



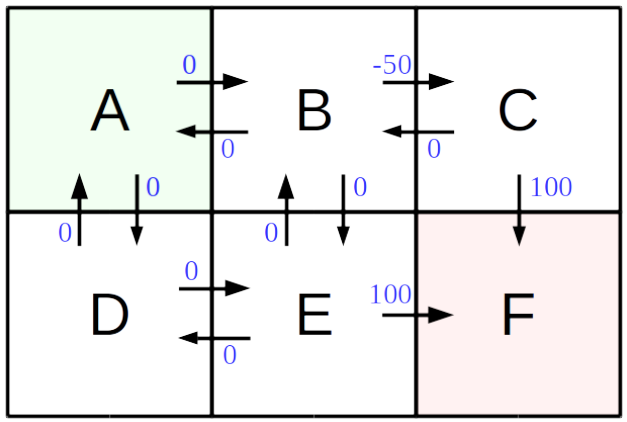
$$p(\mathsf{B}, 0 \mid \mathsf{A}, \rightarrow) = 1$$
$$p(s, r \mid \mathsf{A}, \rightarrow) = 0 \;\; \forall (s, r) \neq (\mathsf{B}, 0)$$

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

etc. (y descuento $\gamma = 0{,}9$)

Repaso: MDPs y Value Iteration

Supongamos que tenemos conocimiento completo del ambiente:



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

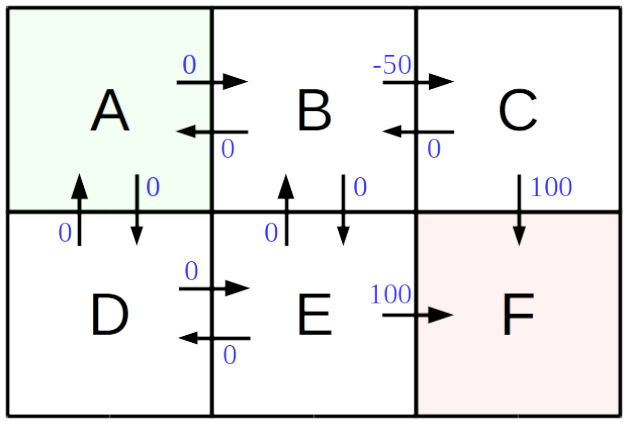
$$p(\mathsf{F}, 100 \mid \mathsf{C}, \downarrow) = 1$$
$$p(s, r \mid \mathsf{C}, \downarrow) = 0 \quad \forall (s, r) \neq (\mathsf{F}, 100)$$

etc. (y descuento $\gamma = 0{,}9$)

¿Cómo hacemos para encontrar una política óptima π_*?

Repaso: MDPs y Value Iteration

Supongamos que tenemos conocimiento completo del ambiente:



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

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

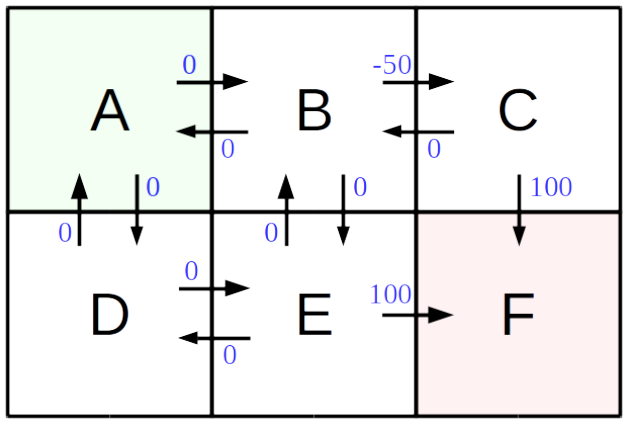
etc. (y descuento $\gamma = 0{,}9$)

¿Cómo hacemos para encontrar una política óptima π_*?

¡El ejemplo de arriba es trivial! Imaginemos un ambiente con miles/millones de estados, ya no dispuestos prolijamente en una grilla, con decenas/cientos de acciones posibles, etc.

Repaso: MDPs y Value Iteration

Supongamos que tenemos conocimiento completo del ambiente:



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

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

etc. (y descuento $\gamma = 0{,}9$)

¿Cómo hacemos para encontrar una política óptima π_*?

¡El ejemplo de arriba es trivial! Imaginemos un ambiente con miles/millones de estados, ya no dispuestos prolijamente en una grilla, con decenas/cientos de acciones posibles, etc.

Incluso con conocimiento completo de $p(s', r \mid s, a)$, ¿cómo hacemos para encontrar una política óptima π_*?

¿Sirven para algo las técnicas que aprendimos en materias de programación? ¿Fuerza bruta, backtracking?

Repaso: MDPs y Value Iteration

Con conocimiento completo del ambiente, el método Value Iteration estima (mediante programación dinámica) una política óptima π_*:

Inicializar $V(s)$ arbitrariamente $\forall s \in \mathcal{S}$, excepto $V(\text{terminal}) = 0$.
Repetir:
$\Delta \leftarrow 0$
Para cada $s \in \mathcal{S}$:
$v \leftarrow V(s)$
$V(s) \leftarrow \text{máx}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma\, V(s')\big]$
$\Delta \leftarrow \text{máx}\left(\Delta, |v - V(s)|\right)$
Hasta que $\Delta < \theta$ para algún umbral θ pequeño.
Devolver una política determinística $\pi \approx \pi_*$ tal que:
$\pi(s) = \arg\text{máx}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma\, V(s')\big]$

Repaso: MDPs y Value Iteration

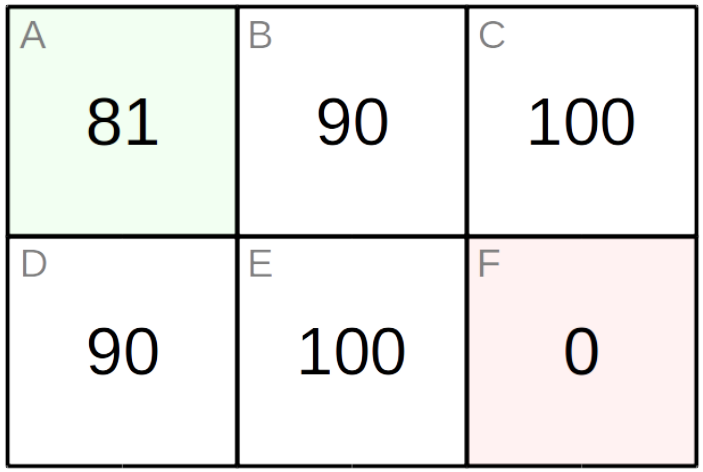
Con conocimiento completo del ambiente, el método Value Iteration estima (mediante programación dinámica) una política óptima π_*:

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

Pero este método muchas veces es inviable, porque:

1. hay que precomputar todas las $p(s', r \mid s, a)$ antes de empezar; y
2. es carísimo computacionalmente: "Para cada $s \in \mathcal{S}$..." en cada iteración.

Supongamos que obtenemos una estimación de v_* mediante algún método (por ejemplo, Value Iteration):

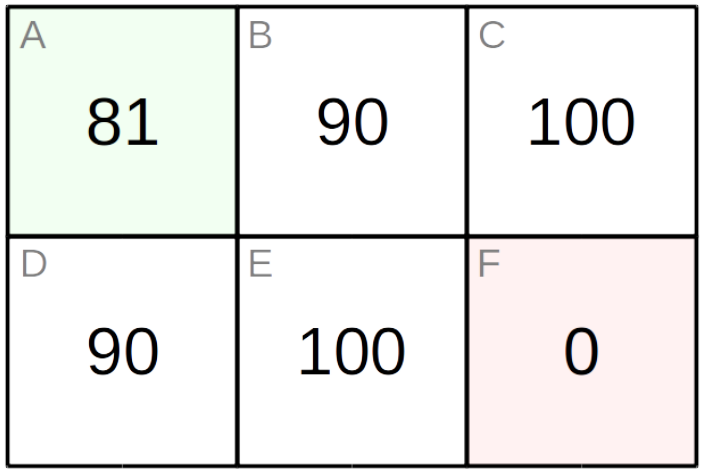


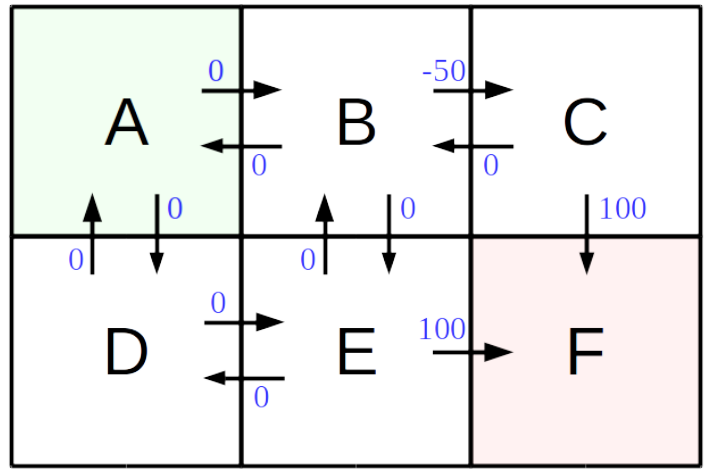
Supongamos que obtenemos una estimación de v_* mediante algún método (por ejemplo, Value Iteration):

A 81	B 90	C 100
D 90	E 100	F 0

¿Alcanza esto para definir π_*? Ej.: ¿cuál debería ser $\pi_*(B)$?

Supongamos que obtenemos una estimación de v_* mediante algún método (por ejemplo, Value Iteration):





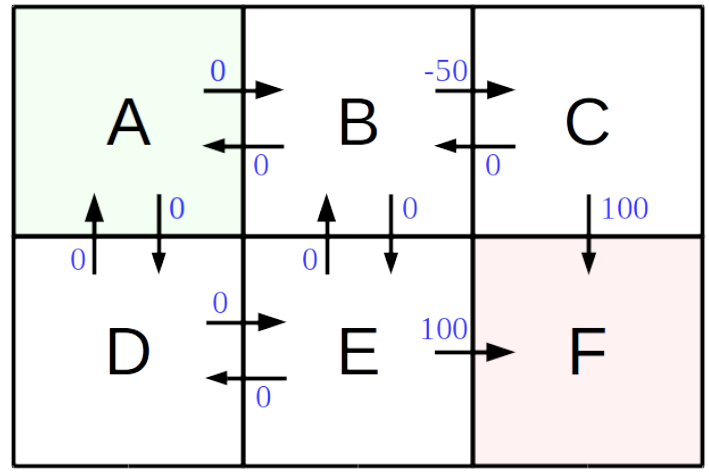
¿Alcanza esto para definir π_*? Ej.: ¿cuál debería ser $\pi_*(B)$?

Para que v_* sea útil, necesitamos conocer $p(s', r \mid s, a)$.

$$\pi_*(B) = \underset{a \in \{\leftarrow, \downarrow, \rightarrow\}}{\arg\max} \; r_a + \gamma \, v_*(s'_a)$$

Supongamos que obtenemos una estimación de v_* mediante algún método (por ejemplo, Value Iteration):

A	B	C
81	90	100
D	**E**	**F**
90	100	0

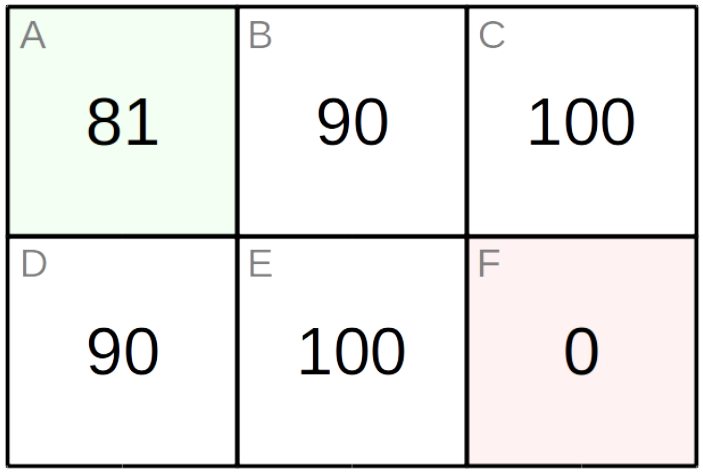


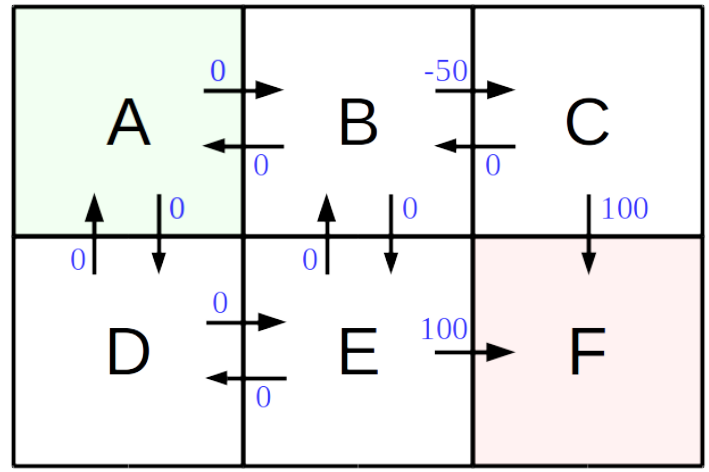
¿Alcanza esto para definir π_*? Ej.: ¿cuál debería ser $\pi_*(B)$?

Para que v_* sea útil, necesitamos conocer $p(s', r \mid s, a)$.

$$\begin{aligned}
\pi_*(B) &= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; r_a + \gamma\, v_*(s'_a) \\
&= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; \{0 + \gamma\, v_*(A),\; 0 + \gamma\, v_*(E),\; -50 + \gamma\, v_*(C)\} \\
&= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; \{72{,}9,\; 90,\; 40\} = \downarrow
\end{aligned}$$

Supongamos que obtenemos una estimación de v_* mediante algún método (por ejemplo, Value Iteration):





¿Alcanza esto para definir π_*? Ej.: ¿cuál debería ser $\pi_*(B)$?

Para que v_* sea útil, necesitamos conocer $p(s', r \mid s, a)$.

$$\begin{aligned}
\pi_*(B) &= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; r_a + \gamma\, v_*(s'_a) \\
&= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; \{0 + \gamma\, v_*(A),\; 0 + \gamma\, v_*(E),\; -50 + \gamma\, v_*(C)\} \\
&= \operatorname*{arg\,máx}_{a \in \{\leftarrow, \downarrow, \rightarrow\}} \; \{72{,}9,\; 90,\; 40\} = \downarrow
\end{aligned}$$

Observación: $q_*(B, \leftarrow) = 72{,}9$, $q_*(B, \downarrow) = 90$, $q_*(B, \rightarrow) = 40$

Repaso: ¿Cómo podemos estimar π_*?

¿Conocemos el ambiente $p(s', r \mid s, a)$?

- **Sí** $\rightarrow$ con una estimación de $v_*(s)$ podemos estimar π_*.
- **No** $\rightarrow$ necesitamos estimar $q_*(s, a)$ para estimar π_*.

Repaso: ¿Cómo podemos estimar π_*?

¿Conocemos el ambiente $p(s', r \mid s, a)$?

- **Sí** $\rightarrow$ con una estimación de $v_*(s)$ podemos estimar π_*.
- **No** $\rightarrow$ necesitamos estimar $q_*(s, a)$ para estimar π_*.

Vamos a ver dos métodos, Monte Carlo (hoy) y Diferencias Temporales (la clase próxima)

Repaso: ¿Cómo podemos estimar π_*?

¿Conocemos el ambiente $p(s', r \mid s, a)$?

- **Sí** $\rightarrow$ con una estimación de $v_*(s)$ podemos estimar π_*.
- **No** $\rightarrow$ necesitamos estimar $q_*(s, a)$ para estimar π_*.

Vamos a ver dos métodos, Monte Carlo (hoy) y Diferencias Temporales (la clase próxima), que:

- por un lado, atacan el problema de la inviabilidad práctica de Value Iteration para problemas complejos;

Repaso: ¿Cómo podemos estimar π_*?

¿Conocemos el ambiente $p(s', r \mid s, a)$?

- **Sí** $\rightarrow$ con una estimación de $v_*(s)$ podemos estimar π_*.
- **No** $\rightarrow$ necesitamos estimar $q_*(s, a)$ para estimar π_*.

Vamos a ver dos métodos, Monte Carlo (hoy) y Diferencias Temporales (la clase próxima), que:

- por un lado, atacan el problema de la inviabilidad práctica de Value Iteration para problemas complejos;
- por otro lado, contemplan por separado el caso en que conocemos el ambiente (estiman $v_*(a)$) y el caso más general en que no conocemos el ambiente (estiman $q_*(s, a)$).

Métodos Monte Carlo

Son una familia de métodos numéricos que permiten obtener soluciones de problemas matemáticos por medio de pruebas aleatorias repetidas.

Ejemplo: estimación de π

Métodos Monte Carlo

Son una familia de métodos numéricos que permiten obtener soluciones de problemas matemáticos por medio de pruebas aleatorias repetidas.

Ejemplo: estimación de π

En **Aprendizaje Reforzado**, los métodos Monte Carlo se basan en generar episodios, observar y aprender de lo ocurrido.

Métodos Monte Carlo

Son una familia de métodos numéricos que permiten obtener soluciones de problemas matemáticos por medio de pruebas aleatorias repetidas.

Ejemplo: estimación de π

En **Aprendizaje Reforzado**, los métodos Monte Carlo se basan en generar episodios, observar y aprender de lo ocurrido.

- Si conocemos $p(s', r \mid s, a)$, los episodios pueden generarse offline: son simulaciones de interacciones con el ambiente.
- Si no conocemos $p(s', r \mid s, a)$, los episodios son interacciones reales con el ambiente.

Estimación Monte Carlo de $v_\pi(s)$

Inicializar:

$V(s) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}$

$Retornos(s) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}$

Estimación Monte Carlo de $v_\pi(s)$

Inicializar:

$V(s) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}$

$Retornos(s) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots, S_{T-1}, A_{T-1}, R_T$.

Estimación Monte Carlo de $v_\pi(s)$

Inicializar:

$V(s) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}$

$Retornos(s) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Estimación Monte Carlo de $v_\pi(s)$

Inicializar:

$V(s) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}$

$Retornos(s) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Si S_t no aparece en $S_0, S_1, \ldots, S_{t-1}$:

Agregar G a $Retornos(S_t)$

$V(S_t) \leftarrow$ promedio$(Retornos(S_t))$

Estimación Monte Carlo de $v_\pi(s)$

Inicializar:

$V(s) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}$

$Retornos(s) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Si S_t no aparece en $S_0, S_1, \ldots, S_{t-1}$:

Agregar G a $Retornos(S_t)$

$V(S_t) \leftarrow$ promedio$(Retornos(S_t))$

$V(s)$ converge a $v_\pi(s)$ (pero sólo para los estados visitados).

Estimación Monte Carlo de $q_\pi(s, a)$

Inicializar:

$Q(s, a) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Retornos(s, a) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Si (S_t, A_t) no aparece en $(S_0, A_0), (S_1, A_1), \ldots, (S_{t-1}, A_{t-1})$:

Agregar G a $Retornos(S_t, A_t)$

$Q(S_t, A_t) \leftarrow$ promedio$(Retornos(S_t, A_t))$

$Q(s, a)$ converge a $q_\pi(s, a)$ (pero sólo para los estados visitados).

Estimación Monte Carlo de $q_\pi(s, a)$

Problema:

- $Q(s, a)$ converge a $q_\pi(s, a)$ a medida que la cantidad de visitas a (s, a) tiende a infinito.
- Pero muchos pares (s, a) serán visitados muy pocas veces (o nunca), y para ellos no podremos estimar $q_\pi(s, a)$.
- Por eso, debemos intentar visitar **todas** las acciones de **todos** los estados.

Estimación Monte Carlo de $q_\pi(s, a)$

Problema:

- $Q(s, a)$ converge a $q_\pi(s, a)$ a medida que la cantidad de visitas a (s, a) tiende a infinito.
- Pero muchos pares (s, a) serán visitados muy pocas veces (o nunca), y para ellos no podremos estimar $q_\pi(s, a)$.
- Por eso, debemos intentar visitar **todas** las acciones de **todos** los estados.

Idea:

- Que los episodios comiencen en un par (S_0, A_0) elegido al azar, de modo que todos los pares tengan probabilidad no nula de ser iniciales.
- Esto se conoce como exploración inicial.

Estimación MC de $q_\pi(s, a)$ con exploración inicial

Inicializar:

$Q(s, a) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Retornos(s, a) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repetir:

Elegir $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$ al azar†

Generar un episodio según π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Si (S_t, A_t) no aparece en $(S_0, A_0), (S_1, A_1), \ldots, (S_{t-1}, A_{t-1})$:

Agregar G a $Retornos(S_t, A_t)$

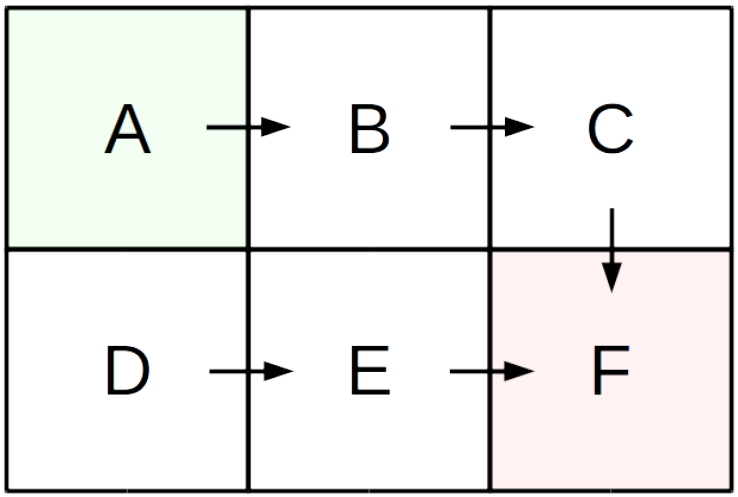
$Q(S_t, A_t) \leftarrow$ promedio$(Retornos(S_t, A_t))$

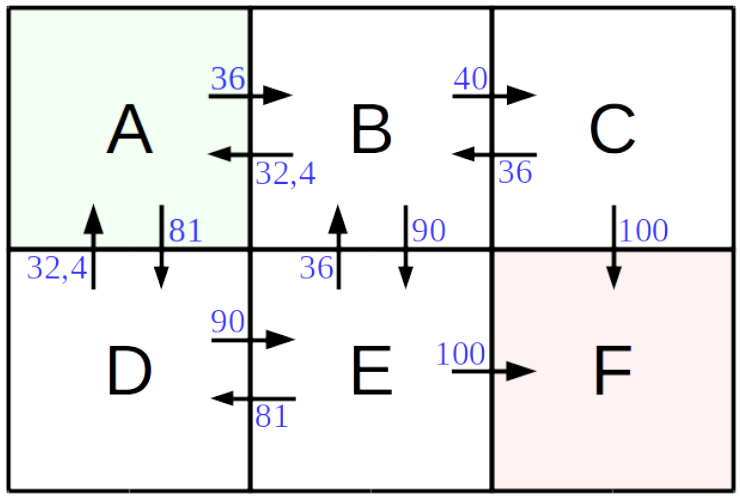
$Q(s, a)$ converge a $q_\pi(s, a)$.

† Todos los pares (s, a) tienen probabilidad > 0 de ser elegidos.

Repaso: Mejora de una política π

Suponiendo que conocemos q_π (derecha), ¿cómo podemos mejorar la política π (izquierda)?





Una política π' es una mejora greedy de π si:

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

Control MC con exploración inicial

Al igual que en Policy Iteration, podríamos avanzar en forma iterativa hacia una estimación de una política óptima π_*:

evaluar π; mejorar π; repetir.

Control MC con exploración inicial

Al igual que en Policy Iteration, podríamos avanzar en forma iterativa hacia una estimación de una política óptima π_*:

evaluar π; mejorar π; repetir.

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

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

Control MC con exploración inicial

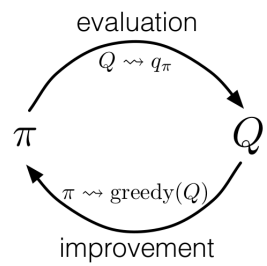
Al igual que en Policy Iteration, podríamos avanzar en forma iterativa hacia una estimación de una política óptima π_*:

evaluar π; mejorar π; repetir.

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

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

Observación: Esta técnica general, que evalúa y mejora políticas en forma iterativa, se llama Generalized Policy Iteration (GPI).



Control MC con exploración inicial

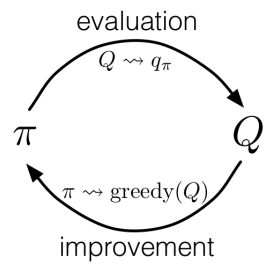
Al igual que en Policy Iteration, podríamos avanzar en forma iterativa hacia una estimación de una política óptima π_*:

evaluar π; mejorar π; repetir.

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

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

Observación: Esta técnica general, que evalúa y mejora políticas en forma iterativa, se llama Generalized Policy Iteration (GPI).



Pero estimar q_{π_k} en cada paso demora mucho. Mejor juntar evaluación y mejora en cada iteración, como en Value Iteration.

Control MC con exploración inicial

Inicializar:

$\pi(s) \in \mathcal{A}(s)$ arbitrariamente, $\forall s \in \mathcal{S}$

$Q(s,a) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Retornos(s,a) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repetir:

Elegir $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$ al azar

Generar un episodio según π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma\, G + R_{t+1}$

Si (S_t, A_t) no aparece en $(S_0, A_0), (S_1, A_1), \ldots, (S_{t-1}, A_{t-1})$:

Agregar G a $Retornos(S_t, A_t)$

$Q(S_t, A_t) \leftarrow$ promedio$(Retornos(S_t, A_t))$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Devuelve una política determinística $\pi(s) \approx \pi_*$.

Control MC sin exploración inicial

La idea de comenzar los episodios en un par (S_0, A_0) elegido al azar es un tanto artificial, y muchas veces no tiene sentido (por ejemplo, en el ajedrez).

Control MC sin exploración inicial

La idea de comenzar los episodios en un par (S_0, A_0) elegido al azar es un tanto artificial, y muchas veces no tiene sentido (por ejemplo, en el ajedrez).

Recordemos de k-bandidos que una forma de **explorar** el espacio de acciones es mediante una política ε-greedy:

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{si } a = A^* \\ \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{si } a \neq A^* \end{cases} \tag{1}$$

donde A^* es la acción greedy según la política actual, desempatando al azar si es necesario.

Control MC sin exploración inicial

La idea de comenzar los episodios en un par (S_0, A_0) elegido al azar es un tanto artificial, y muchas veces no tiene sentido (por ejemplo, en el ajedrez).

Recordemos de k-bandidos que una forma de **explorar** el espacio de acciones es mediante una política ε-greedy:

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{si } a = A^* \\ \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{si } a \neq A^* \end{cases} \tag{1}$$

donde A^* es la acción greedy según la política actual, desempatando al azar si es necesario.

En palabras:

- con probabilidad ε se elige una acción al azar;
- con probabilidad $1 - \varepsilon$ se elige la acción greedy (A^*).

Control MC sin exploración inicial

Inicializar:

$\pi \leftarrow$ política ε-soft al azar (ε-soft: $\pi(a \mid s) > 0 \ \forall s, a$)

$Q(s, a) \in \mathbb{R}$ arbitrariamente, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Retornos(s, a) \leftarrow$ lista vacía, $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repetir:

Generar un episodio según π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$.

$G \leftarrow 0$

Para cada paso del episodio, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Si (S_t, A_t) no aparece en $(S_0, A_0), (S_1, A_1), \ldots, (S_{t-1}, A_{t-1})$:

Agregar G a $Retornos(S_t, A_t)$

$Q(S_t, A_t) \leftarrow$ promedio$(Retornos(S_t, A_t))$

$A^* \leftarrow \arg\max_a Q(S_t, a)$ (desempatando al azar)

$$\pi(a \mid S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{si } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{si } a \neq A^* \end{cases}$$

Devuelve una política estocástica $\pi(a \mid s) \approx \pi_*$.

Métodos on-policy vs. off-policy

Estos métodos de control MC son on-policy: evalúan y mejoran la misma política que usan para generar comportamiento.

Métodos on-policy vs. off-policy

Estos métodos de control MC son on-policy: evalúan y mejoran la misma política que usan para generar comportamiento.

También hay métodos off-policy, que tienen una política a evaluar y mejorar (la política objetivo, o π), y otra política, posiblemente distinta, para decidir qué acciones tomar (la política de comportamiento, o b).

Los métodos off-policy suelen tardar más en converger, pero también suelen ser más potentes.

Opcional: En las secciones 5.5 y siguientes del libro de S&B pueden ver un ejemplo de método MC off-policy: importance sampling.

Resumen - Métodos Monte Carlo

Monte Carlo en Aprendizaje Reforzado: generar episodios, observar y aprender de lo ocurrido.

- Estimación MC de $v_\pi(s)$ y de $q_\pi(s, a)$.
- Control MC con exploración inicial.
- Control MC con política ε-greedy.

Próxima clase: Métodos de Diferencias Temporales: Sarsa, Q-learning.