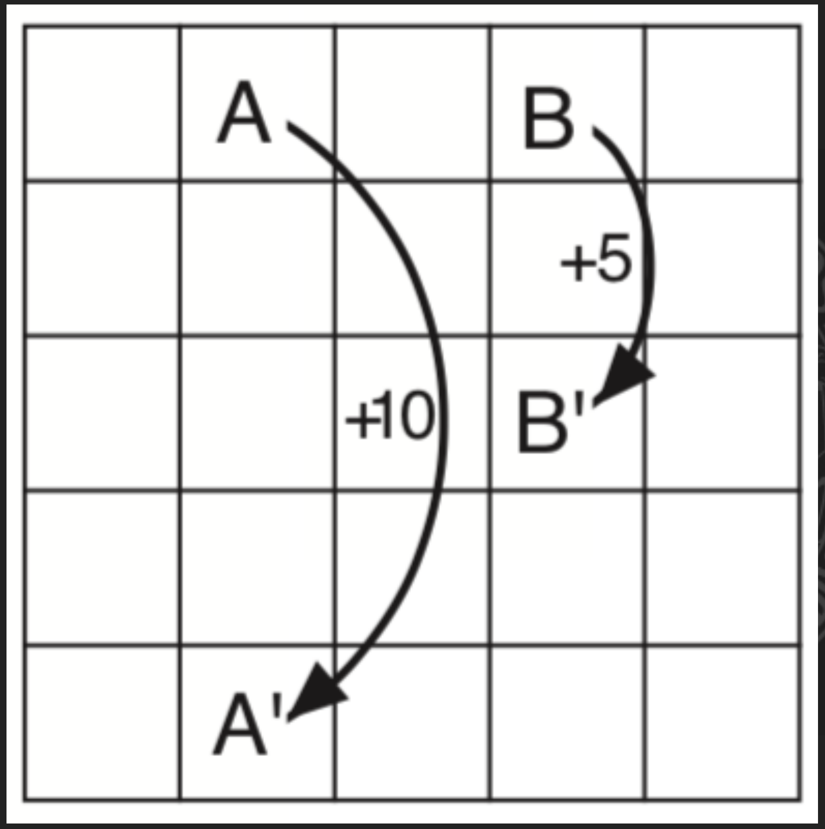
4

FUNDAMENTOS DEL APRENDIZAJE POR REFUERZO (3)

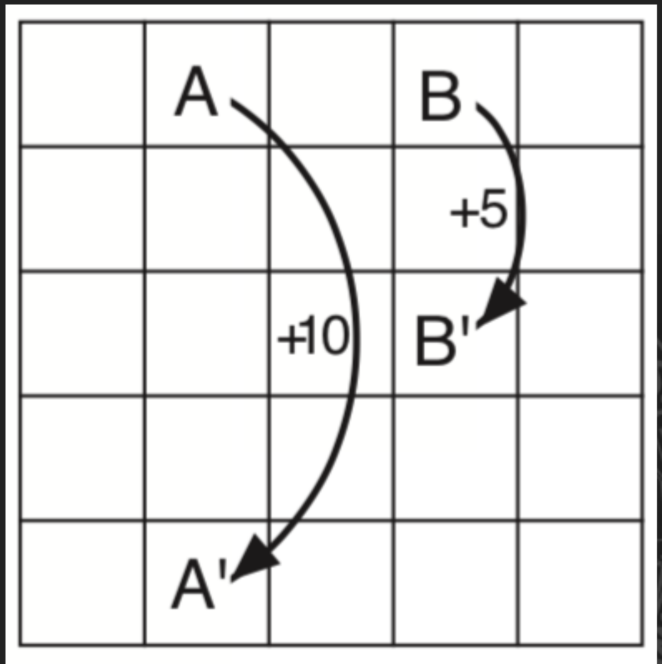
VALOR DE UN ESTADO

- Ejemplo gridworld 5x5
- La figura muestra una representación rectangular de un mundo cuadriculado (Gridworld) de un MDP finito simple.
- Las celdas de la cuadrícula corresponden a los estados del entorno.
- En cada celda, hay cuatro acciones posibles: norte, sur, este y oeste, que determinísticamente hacen que el agente se mueva una celda en la dirección respectiva en la cuadrícula.
- Las acciones que llevarían al agente fuera de los límites de la cuadrícula hacen que permanezca en su ubicación, pero también resultan en una recompensa de −1.
- Otras acciones resultan en una recompensa de 0, excepto aquellas que mueven al agente fuera de los estados especiales A y B.
- Desde el estado A, todas las acciones generan una recompensa de +10 y llevan al agente a A'.
- Desde el estado B, todas las acciones generan una recompensa de +5 y llevan al agente a B'.



VALOR DE UN ESTADO

- Supongamos que el agente selecciona todas las acciones con igual probabilidad en todos los estados.
- La figura inferíor muestra la función de valor de estado $v_\pi(s)$, para esta política, en el caso de recompensa descontada con $\gamma=0.9$.
- Los valores negativos cerca del borde inferior son el resultado de la alta probabilidad de chocar contra el borde de la cuadrícula bajo la política aleatoria.
- El estado A es el mejor estado en el que estar bajo esta política, pero su retorno esperado es menor que 10, su recompensa inmediata, porque desde A el agente es llevado a A', desde donde es probable que choque con el borde de la cuadrícula.
- El estado B tiene un valor mayor a 5, su recompensa inmediata, porque desde B el agente es llevado a B', que tiene un valor positivo.
- Desde B', la penalización (recompensa negativa) por chocar con un borde es más que compensada por la recompensa esperada de posiblemente llegar a A o B.



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

EVALUACIÓN DE POLÍTICA

- Todos los algoritmos anteriores realizan Evaluación de Política.
- Para ello:
  - ✓ Se calcula $v_\pi(s)$ para una política π fija, para todo s.
  - ✓ Se repite el cálculo hasta que $v_\pi(s)$ converge.
- Pseudocódigo del algoritmo de Evaluación de Política Iterativa:

Entrada π (la política a ser evaluada)
Inicializar un umbral $\theta \geq 0$ (determina la precisión de la estimación)
$V(s) \leftarrow 0$

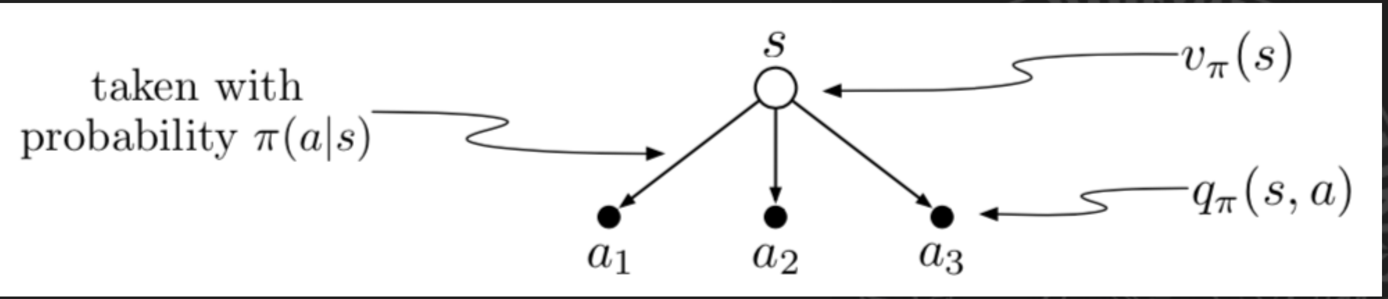
Repetir:
  $\Delta \leftarrow 0$
  Para cada $s \in S$
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
hasta que $\Delta < \theta$

FUNCIONES DE VALOR

- V(s) mide el valor de estar en un estado.

- El valor de un estado depende de los valores de las acciones posibles en ese estado y de la probabilidad de que se tome cada acción bajo la política actual.

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

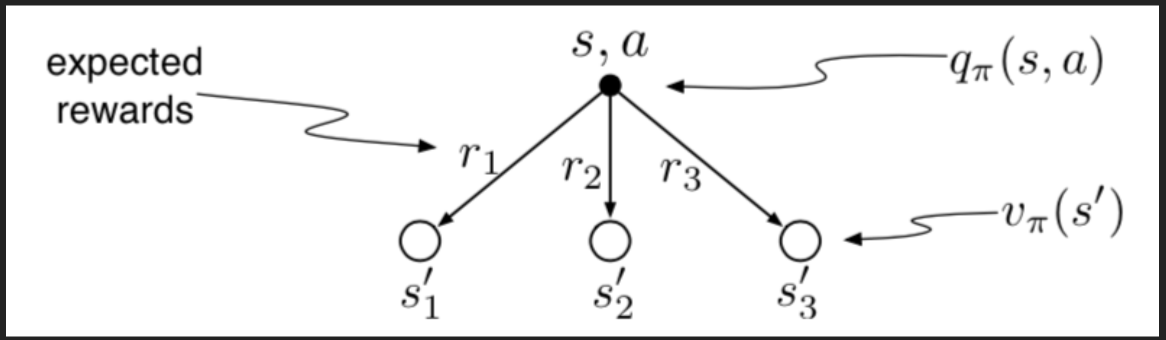


- Q(s, a) mide el valor de tomar una acción específica en un estado determinado y bajo una política.

- El valor de una acción depende de la próxima recompensa esperada y la suma esperada de las recompensas restantes.

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

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



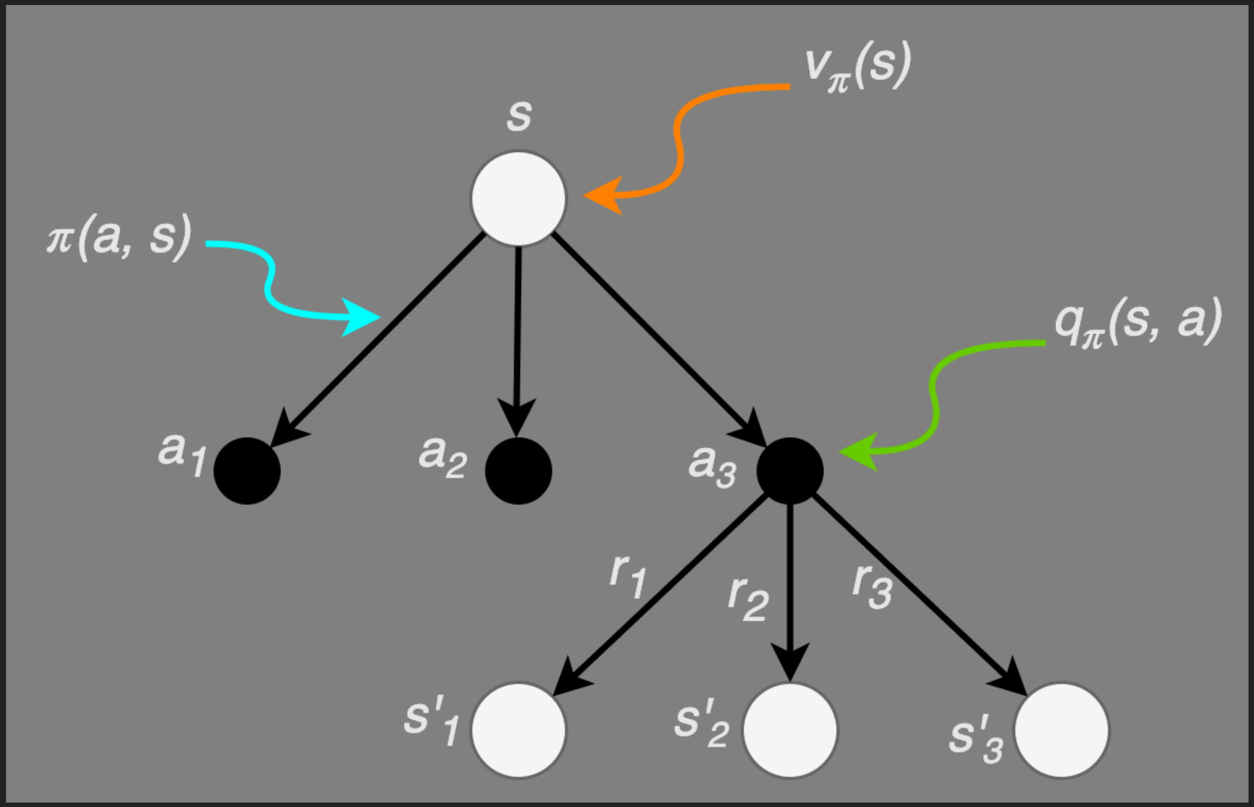
FUNCIONES DE VALOR

- Tanto V(s) como Q(s, a) se conocen como Funciones de Valor.

Funciones de Valor

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

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



ECUACIÓN DE OPTIMALIDAD DE BELLMAN

- Resolver una tarea de aprendizaje por refuerzo significa, a grandes rasgos, encontrar una política que genere una gran recompensa a largo plazo.
- Una política π se define como mejor o igual que una política π' si su rendimiento esperado es mayor o igual que el de π' para todos los estados.

$\pi >= \pi'$ si y solo si $v_\pi(s) >= v_{\pi'}(s)$

- Dado que v* para que sea optima necesita encontrar esa política que obtenga la máxima recompensa entonces π es la incógnita y por lo tanto v* puede expresarse sin referencia a ninguna política específica.

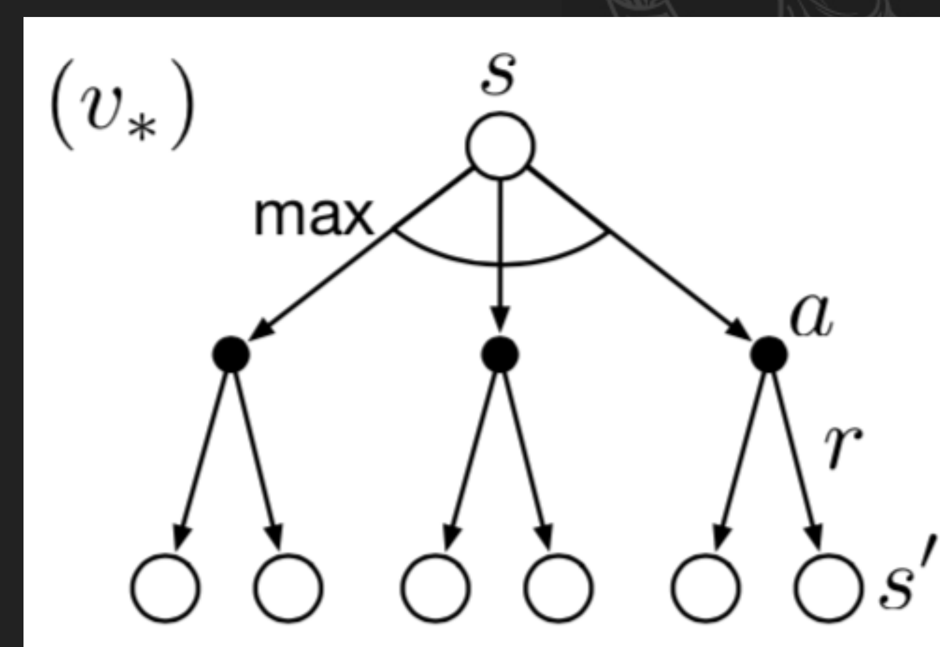
$$v_*(s) \doteq \max_\pi v_\pi(s)$$

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

ECUACIÓN DE OPTIMALIDAD DE BELLMAN

- La ecuación de optimalidad de Bellman expresa que el valor de un estado bajo una política óptima debe ser igual al rendimiento esperado de la mejor acción para ese estado.

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

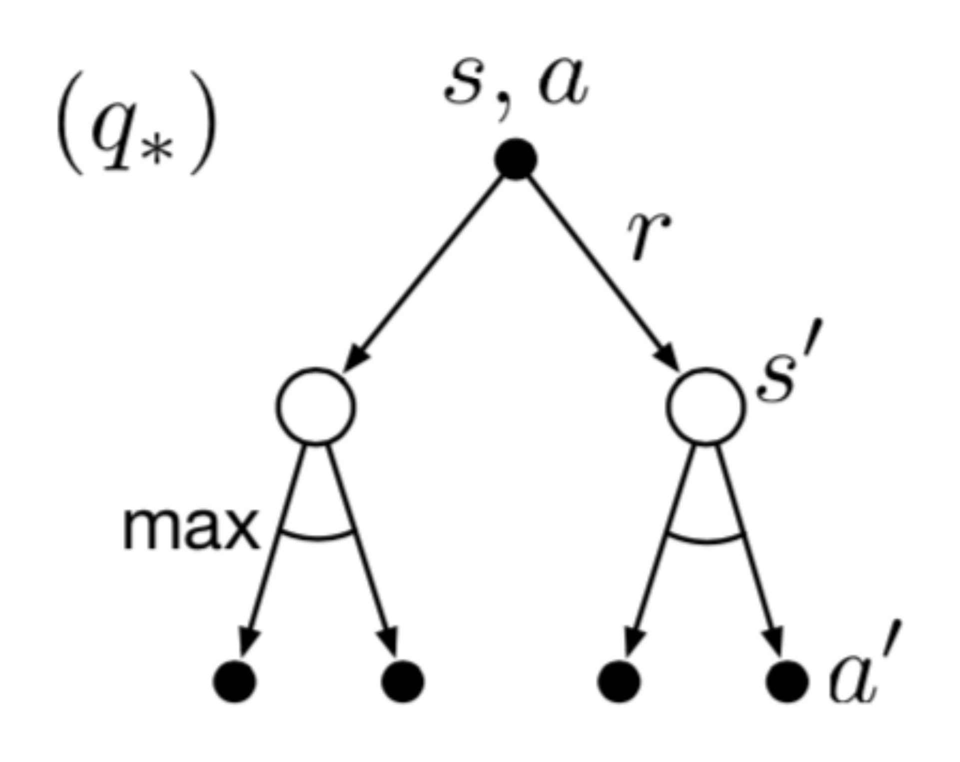


ECUACIÓN DE OPTIMALIDAD DE BELLMAN

- La ecuación de optimalidad de Bellman para q* es:

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

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



MEJORA DE LA POLÍTICA

- Vamos a suponer nuevamente el caso del robot repositor:
- Estados: alto (estante lleno), medio (estante parcialmente vacío), bajo (estante casi vacío)
- Acciones: reponer, no_reponer
- Política:
  - ✓ Si el estante está alto, el robot no repondrá (π(no_reponer | alto) = 1).
  - ✓ Si el estante está medio, el robot repondrá (π(reponer | medio) = 1).
  - ✓ Si el estante está bajo, el robot repondrá (π(reponer | bajo) = 1).
- después de evaluar esta política obtenemos:
  - ✓ v_π(alto) = 6
  - ✓ v_π(medio) = 4
  - ✓ v_π(bajo) = 2

- Estos valores nos dicen qué tan "buenos" son cada uno de estos estados bajo la política actual (π).

MEJORA DE LA POLÍTICA

- ¿Se puede mejorar la política? Se puede analizar el valor de las acciones en cada estado q(s, a) y elegir acciones diferentes a la que dicta la política actual.

MEJORA DE LA POLÍTICA

- Analicemos el estado “alto”: Actualmente π(no_reponer | alto) = 1, cambiaremos a π(reponer | alto) = 1.
- Supongamos que estando en estado “alto” elegimos reponer:
    - ✓ Hay un 80% de prob. de que el estante siga "alto" (con recompensa 1)
    - ✓ Hay un 20% de que baje a "medio" (con recompensa 0).
    - ✓ Factor de descuento γ es 0.9.
- q_π(alto, reponer) = 0.8 * (1 + 0.9 * 6) + 0.2 * (0 + 0.9 * 4) = 5.76 + 0.72 = 6.48
- Dado que q_π(alto, reponer) = 6.48 > v_π(alto) = 6, => hay una mejora. Es mejor reponer en el estado alto que seguir la política actual de no_reponer.

MEJORA DE LA POLÍTICA

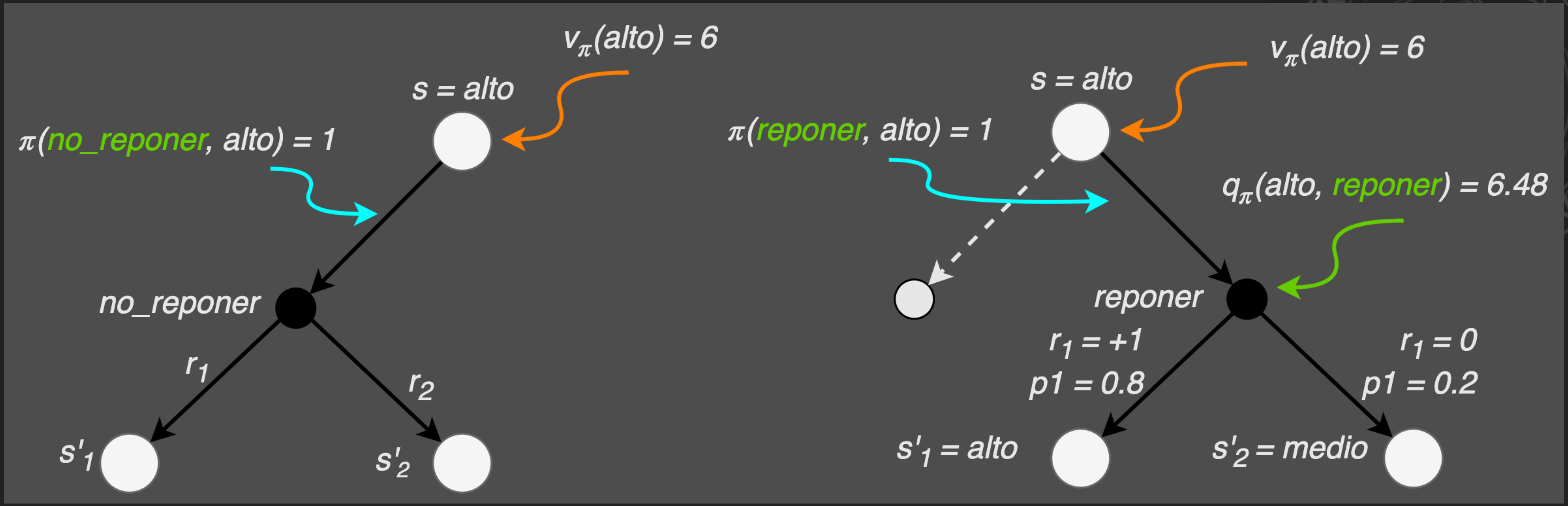
- Analicemos el estado "medio": Actualmente π(reponer | medio) = 1, cambiaremos a π(no_reponer | medio) = 1.
- Supongamos que estando en estado "medio" elegimos no_reponer:
    - ✓ Hay un 70% de prob. de que el estante pase a "bajo" (con recompensa 0)
    - ✓ Hay un 30% de que permanezca en "medio" (con recompensa 1).
    - ✓ Factor de descuento γ es 0.9.
- q_π(medio, no_reponer) = 0.7 * (0 + 0.9 * 2) + 0.3 * (1 + 0.9 * 4) = 1.26 + 1.38 = 2.64
- Dado que q_π(medio, no_reponer) = 2.64 < v_π(medio) = 4, => seguir con la política actual (reponer) es mejor.

MEJORA DE LA POLÍTICA

- Nuestra nueva política (π') es:
  - ✓ Si el estante está alto, el robot no repondrá (π'(reponer | alto) = 1).
  - ✓ Si el estante está medio, el robot repondrá (π'(reponer | medio) = 1).
  - ✓ Si el estante está bajo, el robot repondrá (π'(reponer | bajo) = 1).
- Habría que iterar y volver a evaluar esta nueva política π' (calculando $v_{\pi'}(s)$) y luego intentar mejorarla nuevamente.
- Este proceso de evaluación y mejora puede repetirse hasta que no podamos encontrar más mejoras, en cuyo caso habremos encontrado la política óptima.

MEJORA DE LA POLÍTICA

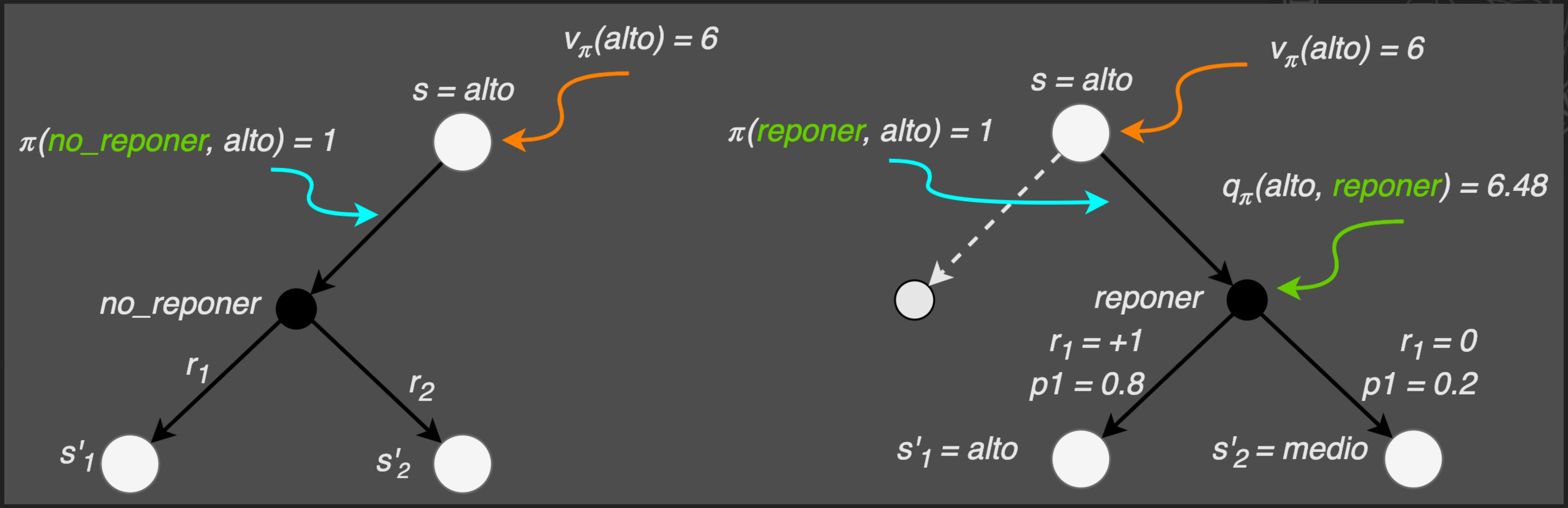
- Formalmente el objetivo de calcular la función de valor $v(s)$ o $q(s, a)$ de una política es tratar de encontrar mejores políticas.
- ¿Sería mejor o peor cambiar a la nueva política? Una forma de saberlo es considerar seleccionar a en s y, a partir de entonces, seguir la política existente.



- Si $q_\pi(s, a) > v_\pi(s)$, entonces tomar la acción a en el estado s y luego seguir la política π es mejor que simplemente seguir la política π desde el estado s.

MEJORA DE LA POLÍTICA

- Podemos traducir el usar $v_\pi(s)$ como: *"Si sigo haciendo lo que mi política actual me dice que haga, esto es lo que puedo esperar obtener a largo plazo $v_\pi(s) = 6$".*
- Podemos traducir el usar $q_\pi(s, a)$ como: *"Si hago algo diferente a lo que mi política actual me dice que haga **solo una vez**, y luego sigo haciendo lo que mi política actual me dice que haga, esto es lo que puedo esperar obtener a largo plazo $q_\pi(s, a) = 6.48$".*



TEOREMA DE MEJORA DE POLÍTICAS

- Si $q_\pi(s, a) > v_\pi(s)$ entonces cabría esperar que fuera aún mejor seleccionar a (reponer) cada vez que se encuentra s (alto), y que la nueva política fuera mejor en general.
- Que esto sea cierto es un caso especial de un resultado general llamado Teorema de mejora de políticas.
- *Sea π y π' dos políticas deterministas. Si $q\pi(s, \pi'(s)) >= v\pi(s)$ para todos los estados s, entonces la política π' es al menos tan buena como la política π (es decir, $v\,\pi'(s) >= v\pi(s)$ para todos los estados s).*
- Si existe una desigualdad estricta (>) para al menos un estado, entonces la política π' es estrictamente mejor que la política π.
- En otras palabras: Si para todos los estados s, el valor de tomar la acción que π' recomienda (y luego seguir π) es al menos tan bueno como seguir π directamente, entonces π' es una mejora sobre π.

TEOREMA DE MEJORA DE POLÍTICAS

- Una vez que encontré una política π' que es mejor que la anterior política π , el siguiente paso es considerar cambiar la política en todos los estados y para todas las acciones. Esto lleva a la idea de una "greedy policy" π':

$$\pi'(s) = \mathrm{argmax}_a \; q_\pi(s, a)$$

$$= \mathrm{argmax}_a \; E[\, Rt+1 + \gamma * v_\pi(St+1) \mid St = s, At = a \,]$$

$$= \mathrm{argmax}_a \; \Sigma_{s,\mathbf{r}} \; p(s', r \mid s, a)\,[r + \gamma * v_\pi(s')]$$

- La "greedy policy" elige, en cada estado, la acción que parece mejor a corto plazo (después de un paso de "lookahead") según la función de valor de la política original π.
- Si la "greedy policy" π' es igual a la política original π, entonces la política original π es óptima. En ese caso la política actual es la mejor.
- Este proceso de evaluar una política (calculando $v\pi$) y luego mejorarla (creando una "greedy policy" π') se puede repetir iterativamente para encontrar la política óptima.

REFERENCIAS BIBLIOGRÁFICAS Y WEB (I)

- R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.