

# Aprendizaje por refuerzo

Clase 6: Control libre de modelo





# Para el día de hoy...

- Control en política
  - Monte Carlo
  - Diferencia temporal
- Fuera de política
  - Diferencia temporal



# Evaluación de política libre de modelo

$$v_{n+1}(S_t) = v_n(S_t) + \alpha(G_t - v_n(S_t))$$

- Variantes para una política  $\pi$  dada:
  - MC:  $G_t^{MC} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = R_{t+1} + \gamma G_{t+1}^{MC}$
  - TD(0):  $G_t^{(1)} = R_{t+1} + \gamma v_t(S_{t+1})$
  - TD(n):  $G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^n v_t(S_{t+n}) = R_{t+1} + \gamma G_{t+1}^{(n-1)}$
  - TD( $\lambda$ ):  $G_t^\lambda = R_{t+1} + \gamma[(1 - \lambda)v_t(S_{t+1}) + \lambda G_{t+1}^\lambda]$

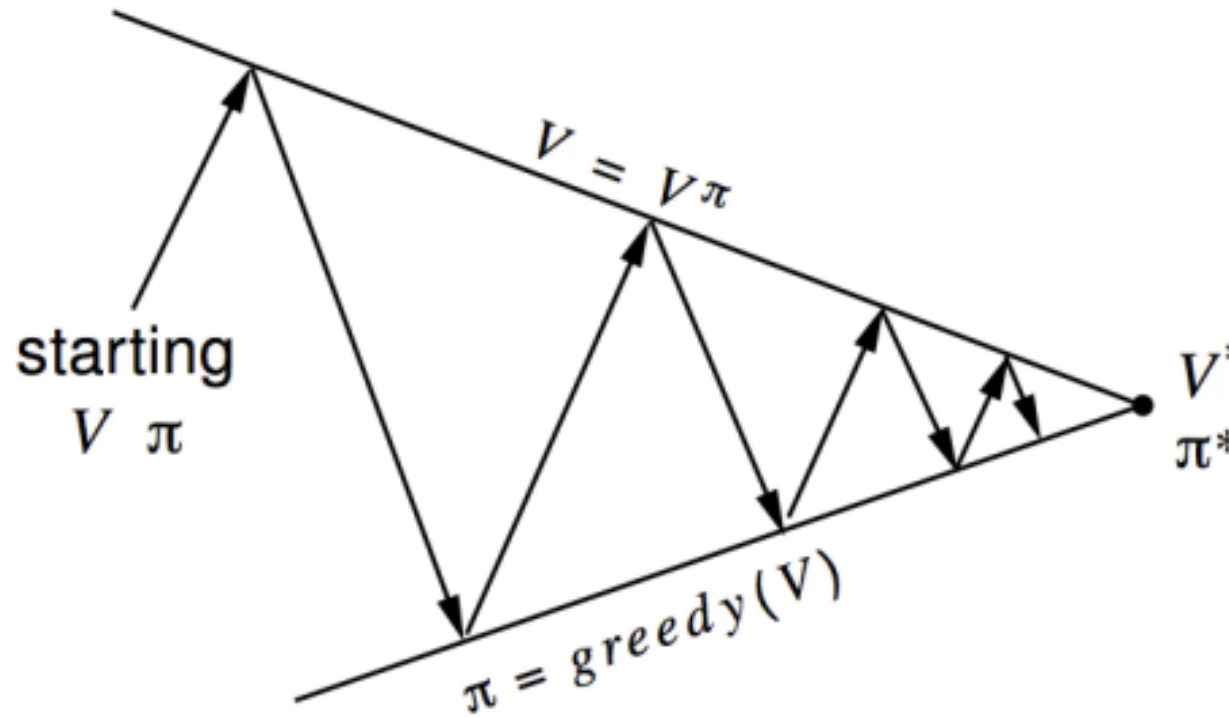


# Control en política y fuera de política

---

- Aprendizaje en política
  - Aprender en el trabajo
  - Aprender de la política  $\pi$  con la experiencia muestreada de  $\pi$
- Aprendizaje fuera de política
  - Supervisar a alguien
  - Aprender de la política  $\pi$  con la experiencia muestreada de  $\mu$

# Iteración de política



- Evaluación de política:
  - Estimar  $v_\pi$
  - Ejemplo evaluación iterativa de política
- Mejora de política:
  - Generar  $\pi' \geq \pi$
  - Mejora voraz de política

## Iteración de política libre de modelo usando la función de acción valor

- La mejora voraz de la política sobre  $v(s)$  requiere el modelo del MDP

$$\pi'(s) = \operatorname{argmax}_a \mathbb{E}[R_{t+1} + \gamma v(S_{t+1} | S_t = s, A_t = a)]$$

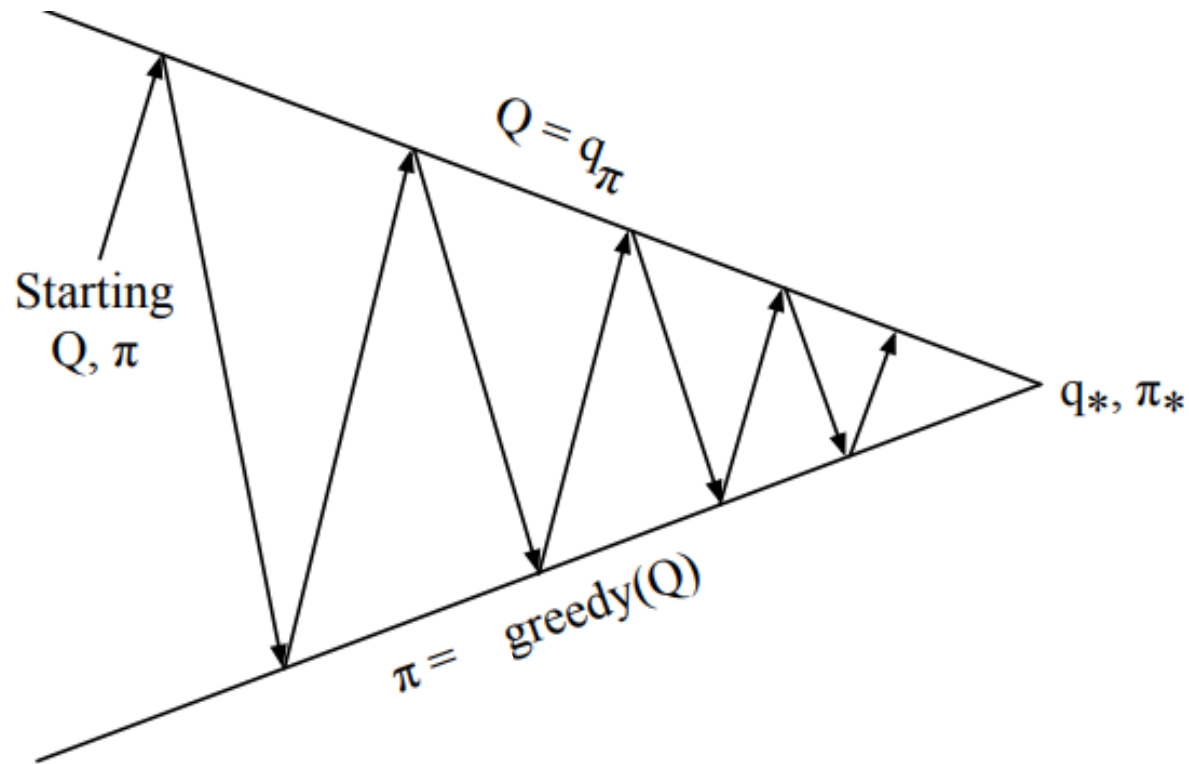
- La mejora voraz de política sobre  $q(s, a)$  es libre de modelo

$$\pi'(s) = \operatorname{argmax}_a q(s, a)$$

- Esto hace a los valores de acción convenientes



# Control con aprendizaje de Monte Carlo (segundo intento)



- Evaluación de política
  - Evaluación de política de Monte Carlo,  $q \approx q_\pi$
- Mejora de política
  - Mejora voraz de política
- ¿Será todo?



"Behind one door is tenure - behind the other is flipping burgers at McDonald's."

## Ejemplo de la selección voraz de una acción

- Existen dos puertas en frente de ustedes
  - Abren la puerta de la izquierda y obtienen 0.  $v(\text{left}) = 0$
  - Abren la puerta de la derecha y obtienen +1.  $v(\text{right}) = 1$
  - Abren la puerta de la derecha y obtienen +3.  $v(\text{right}) = 2$
  - Abren la puerta de la derecha y obtienen +2.  $v(\text{right}) = 2$
  - $\vdots$
  - ¿Eligieron correctamente?



# Exploración $\epsilon$ -voraz

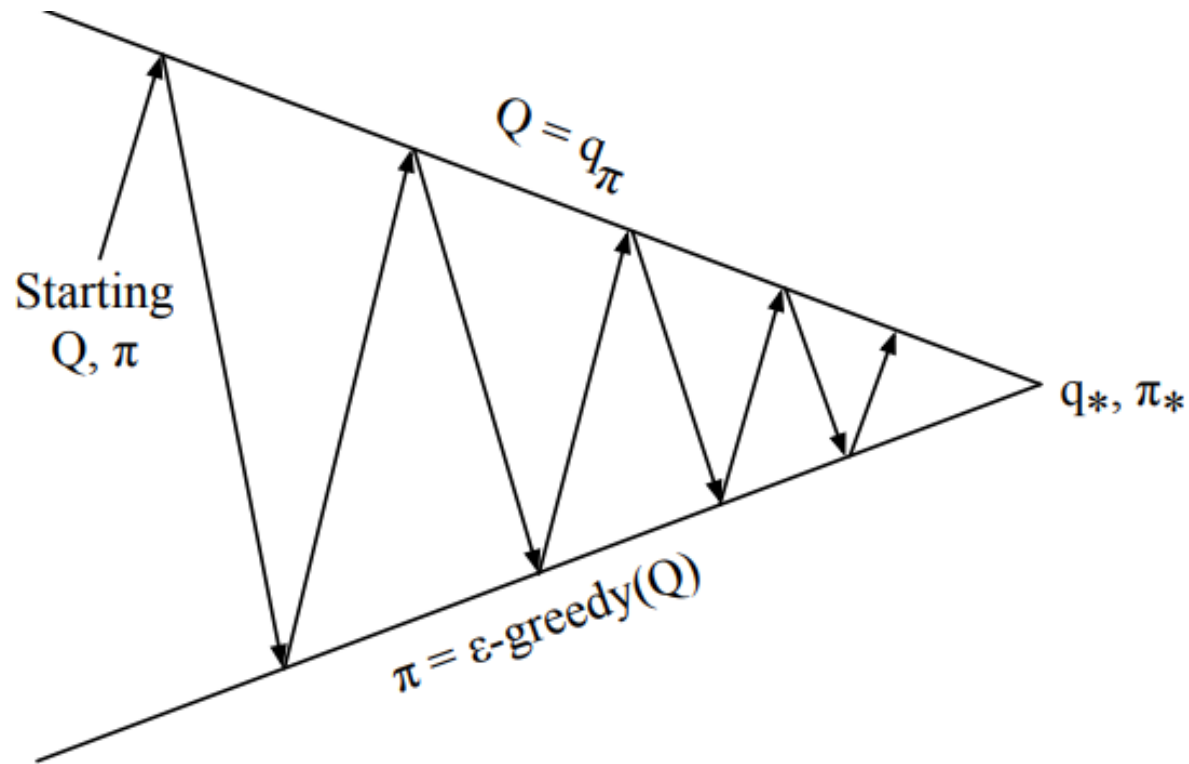
- Es la idea más simple para asegurarse de exploración continua
- Todas las  $m$  acciones tienen probabilidad positiva
- Con probabilidad  $1 - \epsilon$  elegimos la acción voraz
- Con probabilidad  $\epsilon$  elegimos una acción aleatoriamente

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{m} & \text{si } a^* = \operatorname{argmax}_{a \in A} q(s, a) \\ \frac{\epsilon}{m} & \text{de lo contrario} \end{cases}$$

# Mejora de política $\epsilon$ -voraz

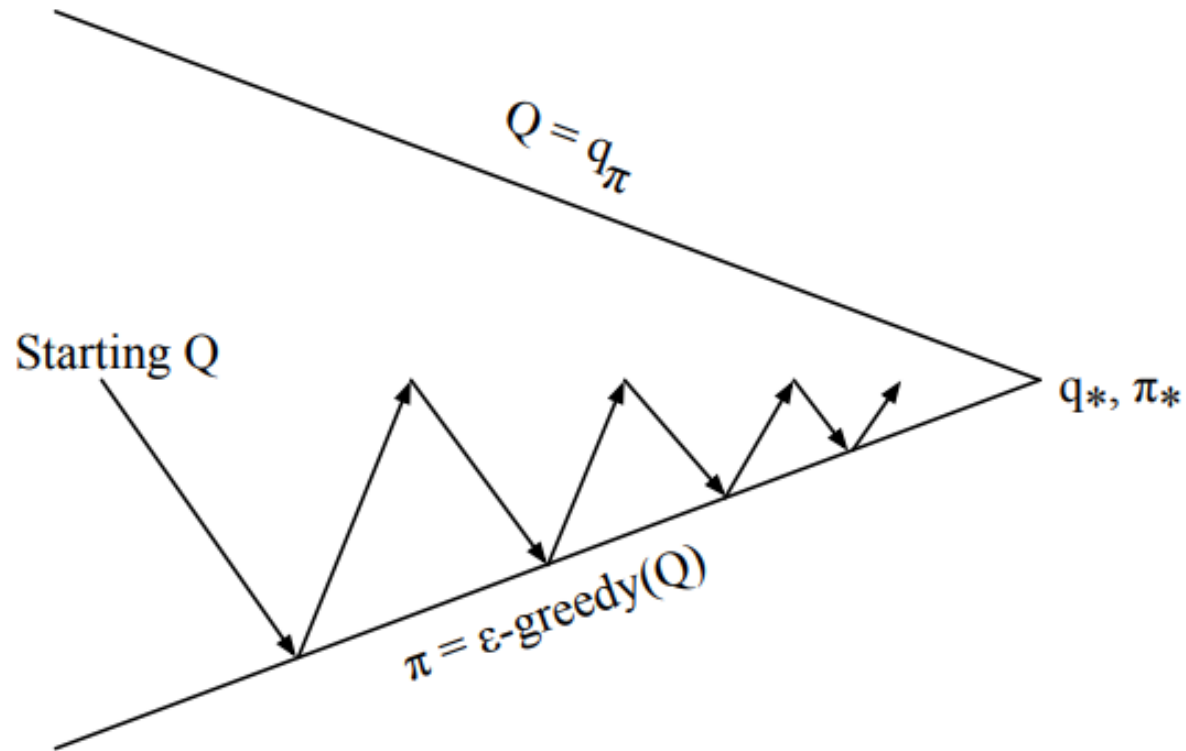
- Teorema: para cualquier política  $\epsilon$ -voraz  $\pi$ , la política  $\epsilon$ -voraz  $\pi'$  con respecto a  $q_\pi$  es una mejora,  $v_{\pi'} \geq v_\pi(s)$

# Control con aprendizaje de Monte Carlo (tercer intento)



- Evaluación de política
  - Evaluación de política de Monte Carlo,  $q \approx q_\pi$
- Mejora de política
  - Mejora  $\epsilon$ -voraz de política
- ¿Por fin?

# Control con aprendizaje de Monte Carlo (n intento)



- Para cada episodio
- Evaluación de política
  - Evaluación de política de Monte Carlo,  $q \approx q_\pi$
- Mejora de política
  - Mejora  $\epsilon$ -voraz de política



# Voraz en el límite con exploración infinita (GLIE)

- Todos los pares de estado acción son explorados un número infinito de veces

$$\lim_{k \rightarrow \infty} N_k(s, a) = \infty$$

- La política converge a una política voraz

$$\lim_{k \rightarrow \infty} \pi_k(a|s) = 1(a = \operatorname{argmax}_{a' \in A} q_k(s, a'))$$

- Por ejemplo,  $\epsilon$ -voraz es GLIE si  $\epsilon$  se reduce a cero en  $\epsilon_k = \frac{1}{k}$

# Control GLIE de Monte Carlo

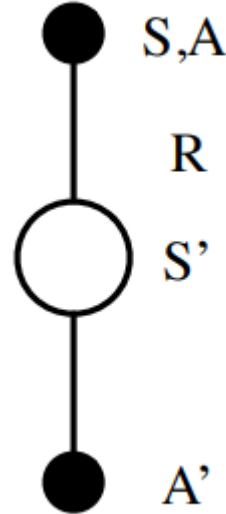
- Muestrar el k-esimo episodio usando  $\pi: \{S_1, A_1, R_2, \dots, S_T\} \sim \pi$
- Para cada estado  $S_t$  y acción  $A_t$  en el episodio
  - $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
  - $q(S_t, A_t) \leftarrow q(S_t, A_t) + \frac{1}{N(S_t, A_t)} (G_t - q(S_t, A_t))$
- Mejorar la política basada en la nueva función acción valor
  - $\epsilon = \frac{1}{k}$
  - $\pi \leftarrow \epsilon\text{-voraz}(q)$

# Control MC vs TD

- El aprendizaje de diferencia temporal (TD) tiene varias ventajas sobre Monte Carlo (MC)
  - Baja varianza
  - En línea
  - Secuencias incompletas
- Idea: usar TD en lugar de MC en nuestro ciclo de control
  - Aplicar TD a  $q(S, A)$
  - Utilizar la mejora de política  $\epsilon$ -voraz
  - Actualizar en cada paso

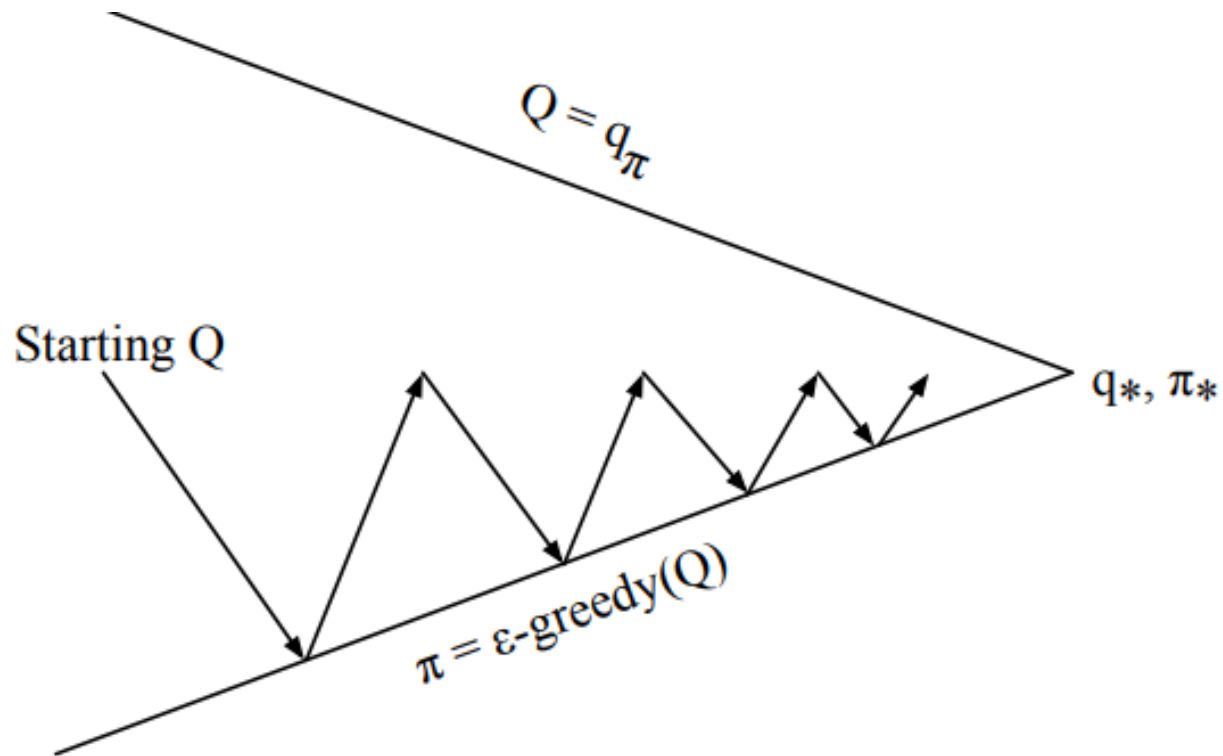
# Actualizar la función de acción valor con SARSA

- $q_{t+1}(S_t, A_t) = q_t(S_t, A_t) + \alpha_t(R_{t+1} + \gamma q_t(S_{t+1}, A_{t+1}) - q_t(S_t, A_t))$





# Control en política con SARSA



- Cada paso
- Evaluar la política SARSA,  $q \approx q_\pi$
- Mejora de política  $\epsilon$ -voraz

# El algoritmo SARSA

- Teorema: SARSA tabular converge a la función optima de acción valor,  $q(s, a) \rightarrow q_*(s, a)$  si la política es GLIE

## Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size  $\alpha \in (0, 1]$ , small  $\varepsilon > 0$

Initialize  $Q(s, a)$ , for all  $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$ , arbitrarily except that  $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

    Initialize  $S$

    Choose  $A$  from  $S$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

    Loop for each step of episode:

        Take action  $A$ , observe  $R, S'$

        Choose  $A'$  from  $S'$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma Q(S', A') - Q(S, A)]$

$S \leftarrow S'; A \leftarrow A';$

    until  $S$  is terminal

# Aprendizaje fuera de política

- Evaluar la política objetivo  $\pi(a|s)$  para calcular  $v_\pi(s)$  o  $q_\pi(s, a)$
- Mientras seguimos la política  $\mu(a|s)$

$$\{S_1, A_1, R_2, \dots, S_T\} \sim \mu$$

- ¿Por qué es importante?
  - Aprender de humanos u otros agentes
  - Reutilizar experiencia generada de antiguas políticas
  - Aprender la política optima mientras exploramos políticas
  - Aprender de múltiples políticas siguiendo una política

# Aprendizaje Q

- La siguiente acción es elegida usando la política

$$A_{t+1} \sim \mu(\cdot, S_t)$$

- Se considera una acción sucesor alternativa  $A' \sim \pi(\cdot, S_t)$

- Actualizamos  $q(S_t, A_t)$  hacia la acción alternativa

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha(R_{t+1} + \gamma q(S_{t+1}, A') - q(S_t, A_t))$$

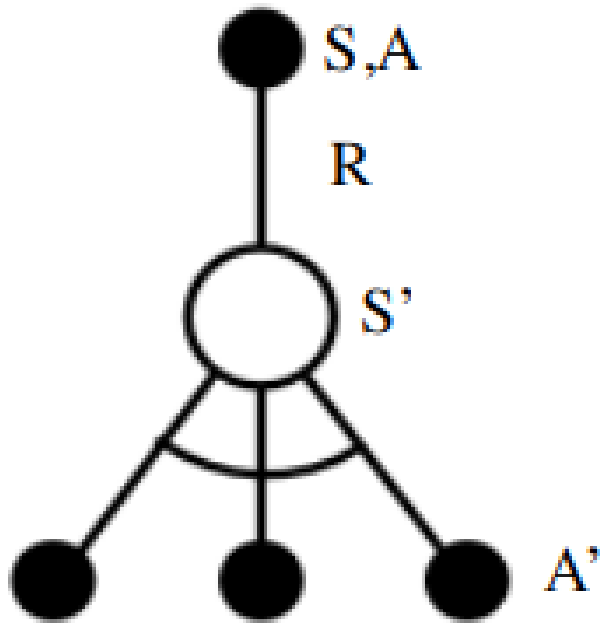


# Control fuera de política con aprendizaje Q

- Ahora permitimos que tanto el comportamiento como la política objetivo mejoren
- La política objetivo  $\pi$  es voraz con respecto a  $q(s, a)$ 
$$\pi(S_{t+1}) = \operatorname{argmax}_{a'} q(S_{t+1}, a')$$
- La política  $\mu$  es  $\epsilon$ -voraz con respecto a  $q(s, a)$ 
  - $R_{t+1} + \gamma q(S_{t+1}, A')$
  - $= R_{t+1} + \gamma q(S_{t+1}, \operatorname{argmax}_{a'} q(S_{t+1}, a'))$
  - $= R_{t+1} + \gamma \operatorname{argmax}_{a'} q(S_{t+1}, a')$

# Algoritmo de control de aprendizaje Q

- $q_{t+1}(S_t, A_t) \leftarrow q_t(S_t, A_t) + \alpha_t(R_{t+1} + \gamma \max_{a'} q_t(S_{t+1}, a') - q_t(S_t, A_t))$
- Teorema
  - Control con aprendizaje Q converge a la función acción valor óptima  $q \rightarrow q^*$  en el límite



## Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size  $\alpha \in (0, 1]$ , small  $\varepsilon > 0$

Initialize  $Q(s, a)$ , for all  $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$ , arbitrarily except that  $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

Initialize  $S$

Loop for each step of episode:

Choose  $A$  from  $S$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

Take action  $A$ , observe  $R, S'$

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_{a'} Q(S', a) - Q(S, A)]$

$S \leftarrow S'$

until  $S$  is terminal

# Los algoritmos de programación dinámica

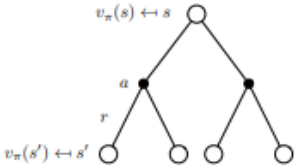
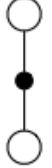
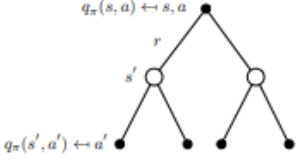
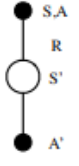
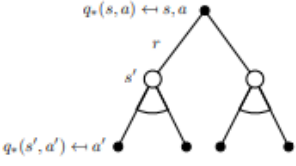
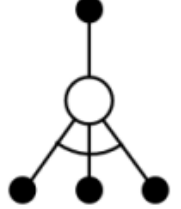
- Evaluación de política
  - $v_{k+1}(s) = \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s, A_t \sim \pi(S_t)]$
- Iteración de valor
  - $v_{k+1}(s) = \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s, A_t = a]$
- Evaluación de política
  - $q_{k+1}(s, a) = \mathbb{E}[R_{t+1} + \gamma q_k(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$
- Iteración de política
  - $q_{k+1}(s, a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_k(S_{t+1}, a') | S_t = s, A_t = a]$

# Los algoritmos de diferencia temporal

- TD ~ evaluación de política con función de valor
  - $v_{t+1}(S_t) = v_t(S_t) + \alpha_t(R_{t+1} + \gamma v_t(S_{t+1}) - v_t(S_t))$
- Iteración de valor
  - ¿?
- SARSA ~ iteración de política con función de acción valor
  - $q_{t+1}(S_t, A_t) = q_t(S_t, A_t) + \alpha_t(R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) - q(S_t, A_t))$
- Aprendizaje Q ~ iteración de valor con función de acción valor
  - $q_{t+1}(S_t, A_t) \leftarrow q_t(S_t, A_t) + \alpha_t(R_{t+1} + \gamma \max_{a'} q_t(S_{t+1}, a') - q_t(S_t, A_t))$



# La relación entre DP y TD

	Full Backup (DP)	Sample Backup (TD)
Bellman Expectation Equation for $v_{\pi}(s)$	 <p>Iterative Policy Evaluation</p>	 <p>TD Learning</p>
Bellman Expectation Equation for $q_{\pi}(s, a)$	 <p>Q-Policy Iteration</p>	 <p>Sarsa</p>
Bellman Optimality Equation for $q_{*}(s, a)$	 <p>Q-Value Iteration</p>	 <p>Q-Learning</p>

# Para la otra vez...

- Funciones de aproximación



iimas

The End.