Aprendizaje Reforzado

Maestría en Ciencia de Datos, DC - UBA

Julián Martínez
Javier Kreiner

Monte Carlo - Predicción libre de modelo

$$v_\pi(s) = \mathcal{R}_s^\pi + \sum_{s'} v_\pi(s')\color{red}{p^\pi_{s,s'}}$$

En general NO CONOCEMOS $\color{red}{p^\pi_{s,s'}}$

¿Podemos hacer evaluación, aprendiendo tan sólo de la experiencia?

Ley de los grandes números

$$\frac{\sum_{i=1}^{n} X_i}{n} \to E[X], \qquad X_i,\ iid$$

Episodio

$$s_0,\ a_0, s_1,\ a_1,\ s_2,\ a_2, {\color{red}s},\ \ldots,\ {\color{red}s},\ a_k,\ s_{k+1},\ a_{k+1}, \ldots,\ s_T,\ a_T$$

Detalle computacional

$$\mu_k := \frac{\sum_{j=1}^{k} x_j}{k} = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$$

$$N(S_t) \leftarrow N(S_t) + 1$$

$$V(S_t) \leftarrow V(S_t) + \frac{(G_t - V(S_t))}{N(S_t)}$$

Diferencia Temporal (TD)

$$v_\pi(s) = E[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

$$v_\pi^{n+1}(S_t) = v_\pi^n(S_t) + \alpha[R_{t+1} + \gamma v_\pi^{n+1}(S_{t+1}) - v_\pi^n(S_{t+1})]$$

Retorno estimado: $R_{t+1} + \gamma v_\pi^{n+1}(S_{t+1})$

Error de TD: $[R_{t+1} + \gamma v_\pi^{n+1}(S_{t+1}) - v_\pi^n(S_{t+1})]$

Comparativa

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = S]$$

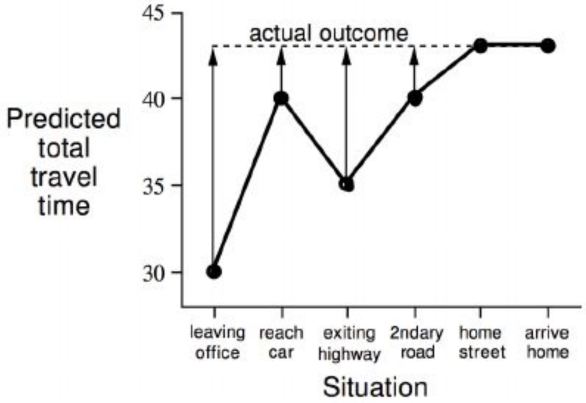
- Programación Dinámica: Actualiza directamente con las esperanzas.
- Monte Carlo: Actualiza usando como target una aproximación de la esperanza que se actualiza *sólo al final del episodio.*
- Diferencia Temporal: Utiliza otra aproximación de la esperanza, pero *se actualiza en cada paso.*

- *Bootstrapping*: El update actualiza una *estimación previa*

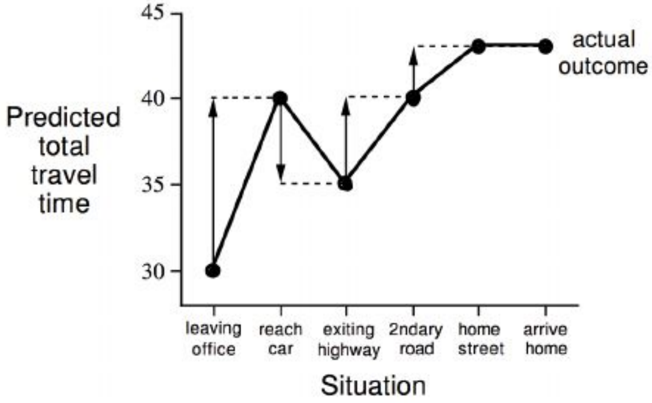
Driving Home (Ejemplo 6.1)

State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

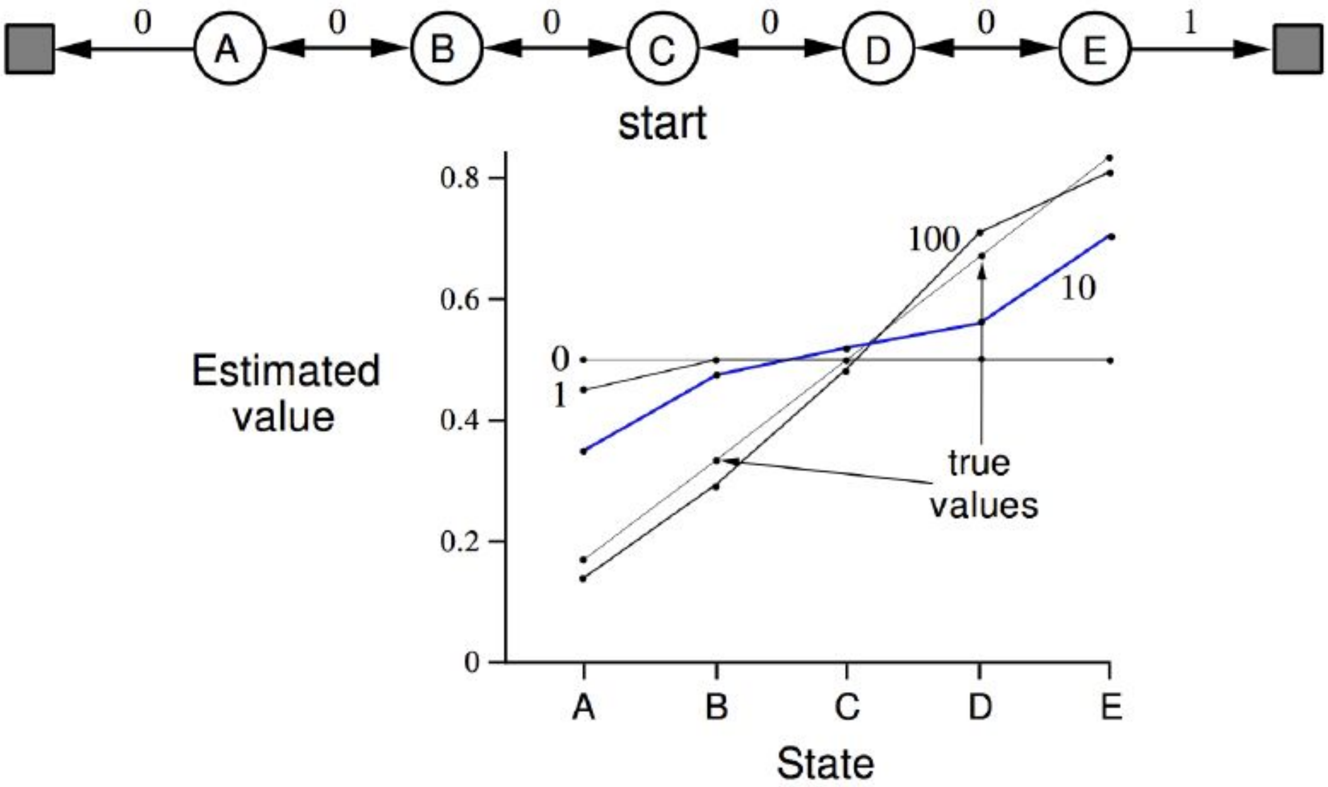
Changes recommended by Monte Carlo methods (α=1)

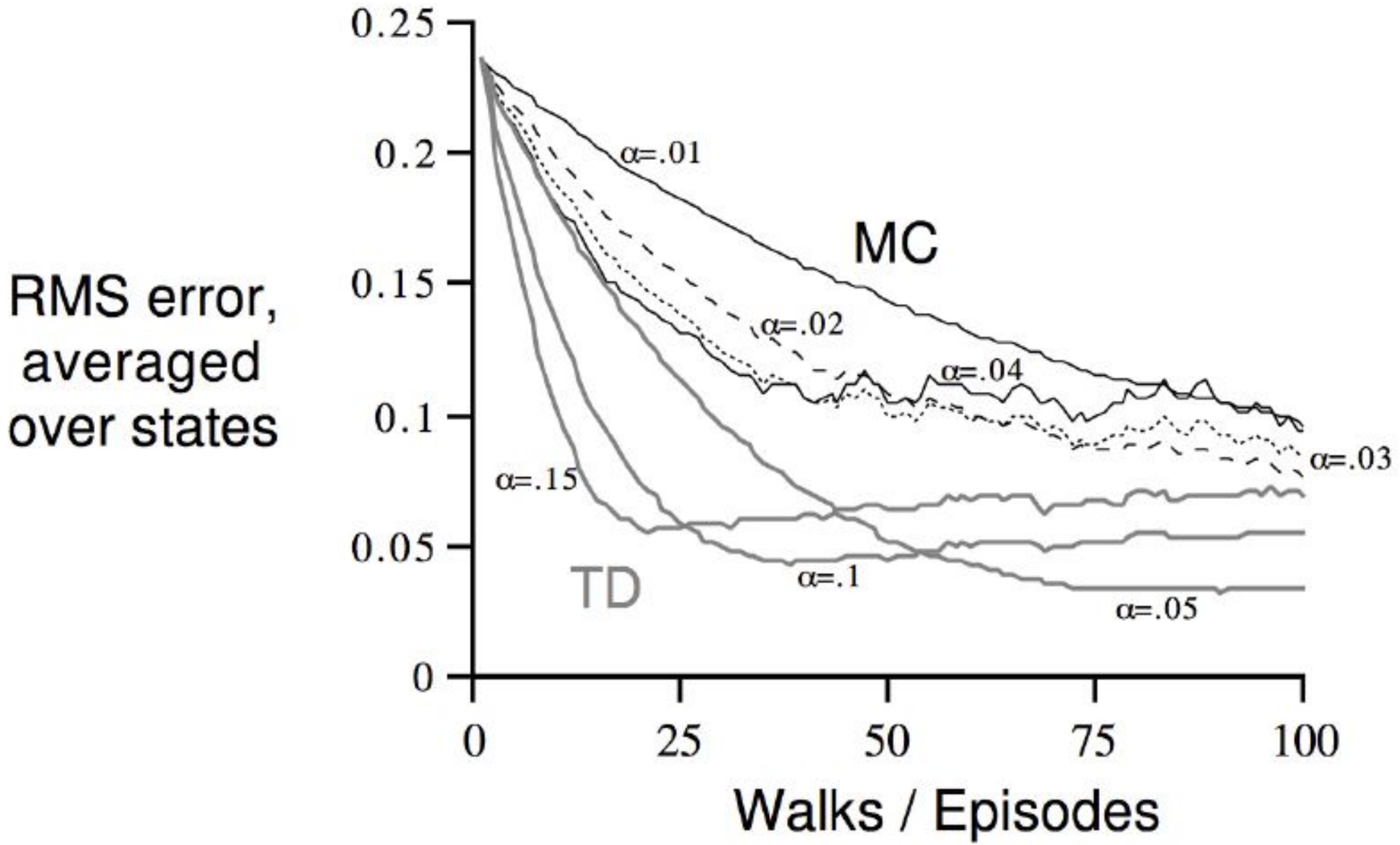


Changes recommended by TD methods (α=1)



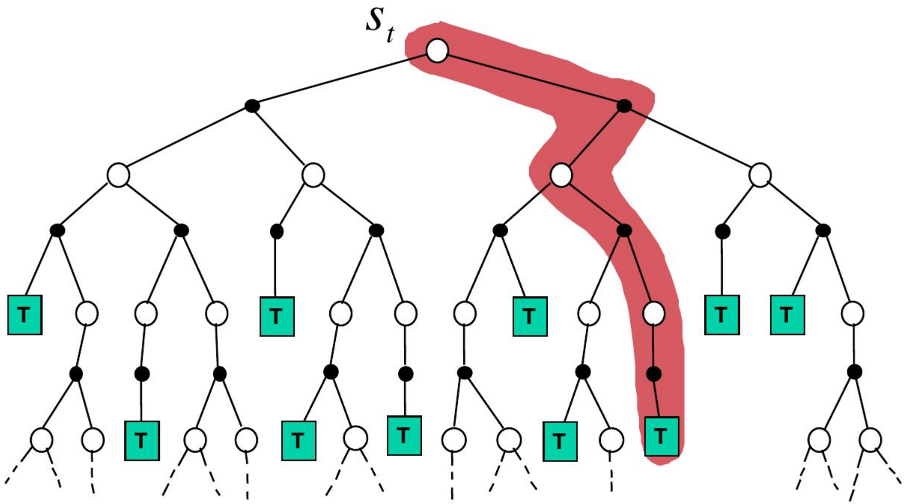
Ejemplo 6.2: Caminata aleatoria



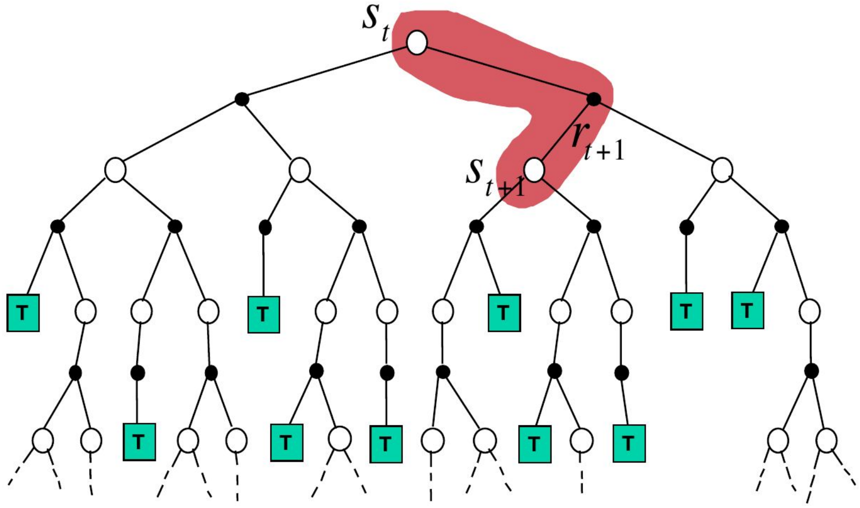


RMS error, averaged over states
0.25
0.2
0.15
0.1
0.05
0
α=.01
MC
α=.02
α=.04
α=.03
α=.15
α=.1
α=.05
TD
0
25
50
75
100
Walks / Episodes

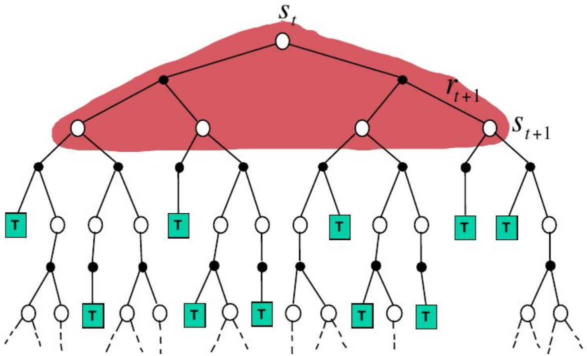
$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t) \right)$$



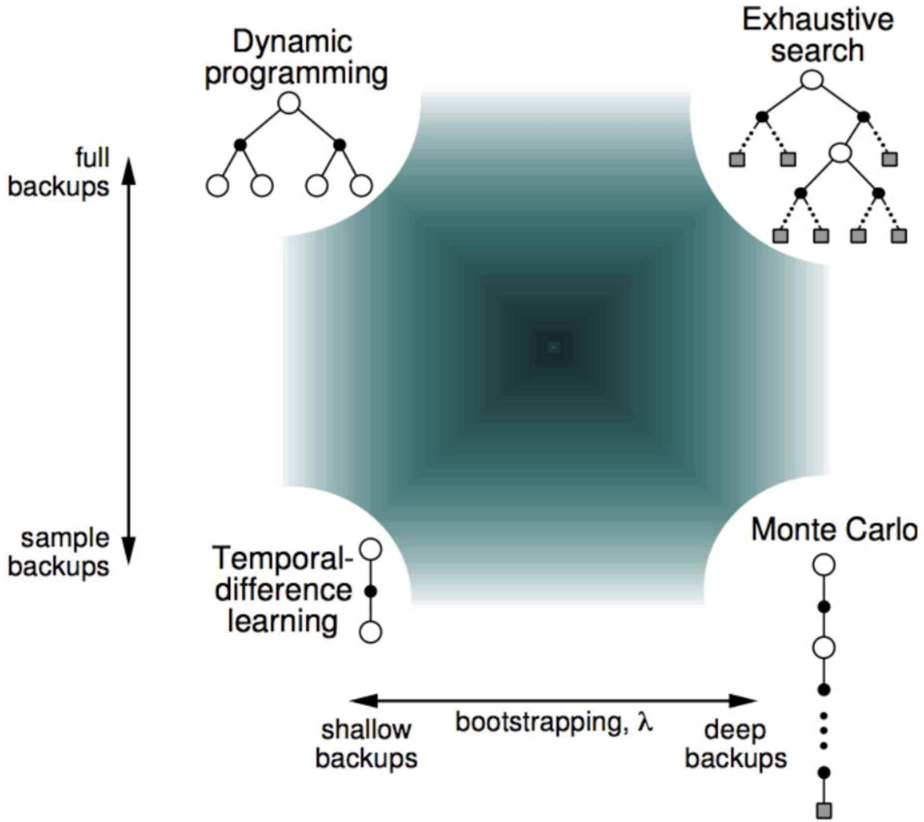
$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



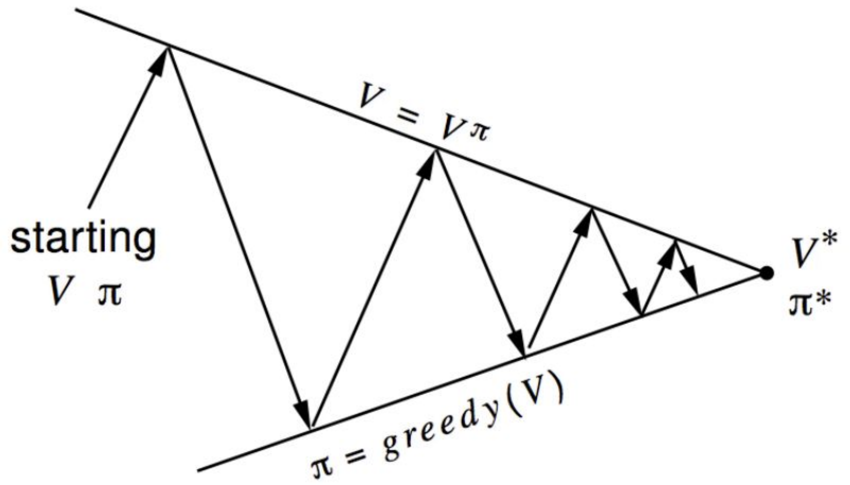
$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1}) \right]$$



Comparativa de los métodos vistos



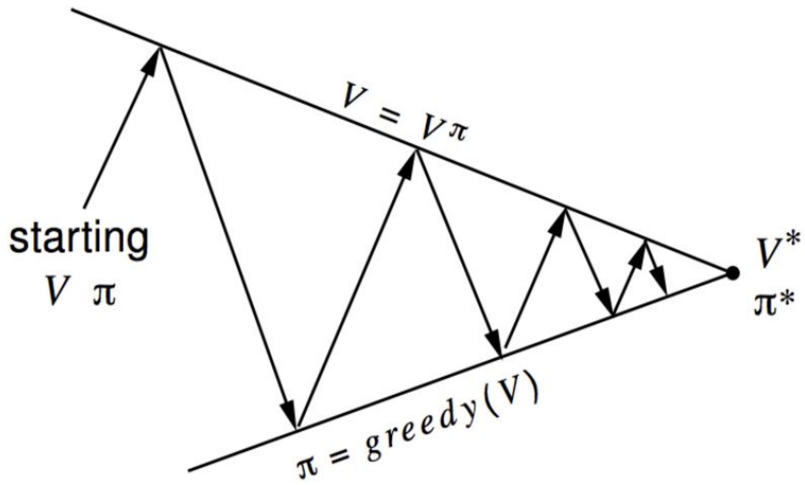
Control (Improvement) - Monte Carlo



$$q_{\pi_k}(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} v_{\pi_k(s')} p_{s,s'}^a$$

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

Control (Improvement) - Monte Carlo

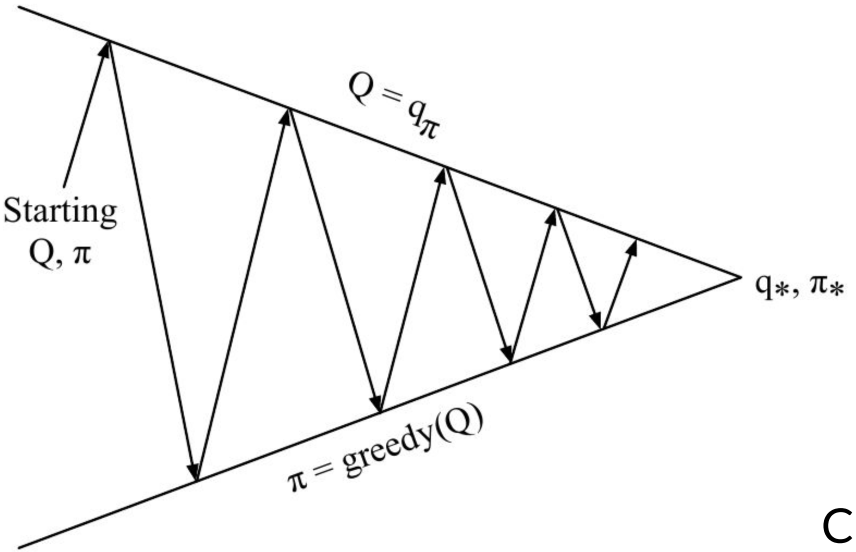


- No tengo como dato la matriz de transición. (*Model Free*)
- No tengo la esperanza *exacta*, donde están *todos* los posibles escenarios. (*Exploration vs. Explotation*)

$$q_{\pi_k}(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} v_{\pi_k(s')} p_{s,s'}^a$$

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

Los parches - Dependiente del modelo (p)



Hacer evaluación MC de

$$q_{\pi_k}(s,a) =: Q_k(s,a)$$

Cambio la esperanza que aproximo.

Probar un poco todo (epsilon - greedy policy)

$$\pi^{\varepsilon}(a|s) = \begin{cases} (1-\varepsilon) + \varepsilon \, \frac{1}{|\mathcal{A}|} & \text{si } a = arg\, \max_a q_{\pi}(s,a) \\ \varepsilon \, \frac{1}{|\mathcal{A}|} & \text{caso contrario} \end{cases}$$

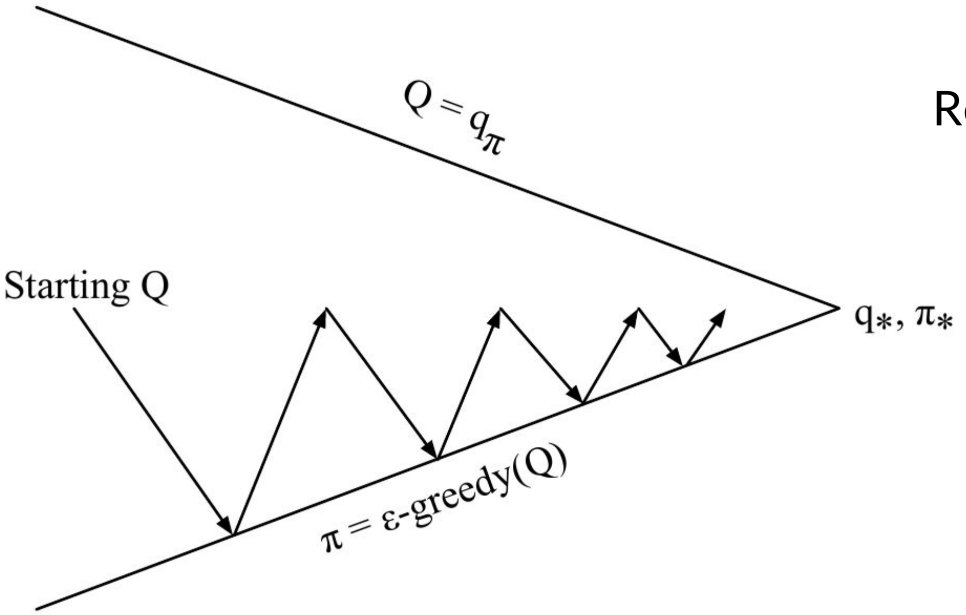
Teorema:

Si π^{ε} una política ε-greedy, $\pi'(s) := arg\, \max_a q_{\pi^{\varepsilon}}(s,a)$.

Entonces:

$$v_{\pi^{\varepsilon}}(s) \leq v_{\pi'^{\varepsilon}}(s)$$

Algunas mejoras



Realizar la actualización en cada episodio.

GLIE Monte Carlo

- Simular el episodio k utilizando la política π_k^ε: $\{S_1, A_1, R_2, \ldots, S_T\}$.
- Para cada par (S_t, A_t) del episodio

$$N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)}(G_t - Q(S_t, A_t))$$

- $$\varepsilon = \frac{1}{k}, \qquad \pi_{k+1}^\varepsilon = \varepsilon - \text{greedy}(Q(s, a))$$

Monte Carlo (programación)

- Ejemplo de Blackjack
- Predicción Monte Carlo
- Predicción TD
- Control Monte Carlo on-policy con políticas epsilon greedy

Ejercicio - Leer:

That any ε-greedy policy with respect to q_π is an improvement over any ε-soft policy π is assured by the policy improvement theorem. Let π' be the ε-greedy policy. The conditions of the policy improvement theorem apply because for any $s \in \mathcal{S}$:

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \max_a q_\pi(s,a) \qquad (5.2) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a)
\end{aligned}
$$

(the sum is a weighted average with nonnegative weights summing to 1, and as such it must be less than or equal to the largest number averaged)

$$
\begin{aligned}
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; - \; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s).
\end{aligned}
$$

Thus, by the policy improvement theorem, $\pi' \geq \pi$ (i.e., $v_{\pi'}(s) \geq v_\pi(s)$, for all $s \in \mathcal{S}$).

Lectura recomendada

- Problemas de reproducibilidad en Deep RL: Deep Reinforcement Learning that Matters, https://arxiv.org/abs/1709.06560