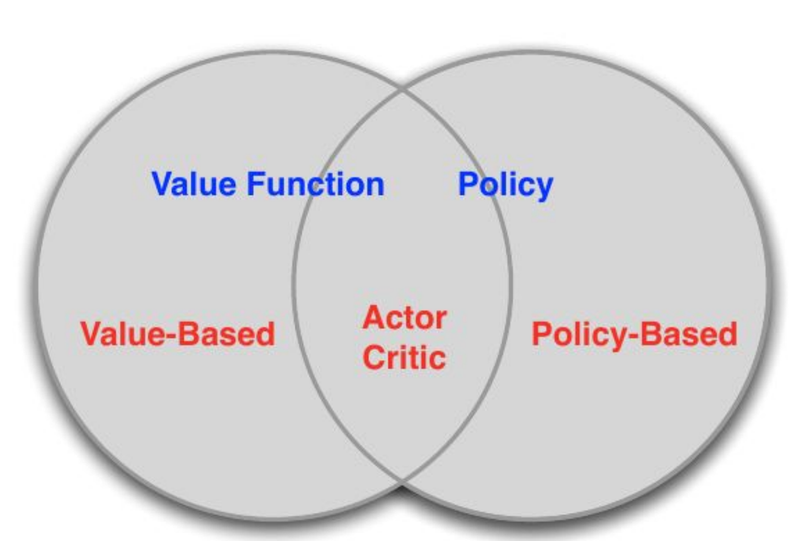
Aprendizaje Reforzado

Maestría en Ciencia de Datos, DC - UBA

Julián Martínez
Javier Kreiner

Policy Gradient



Optimizamos *directamente* sobre la política.

Modelizamos $\pi(a|s, \theta)$ *como una distribución* parametrizada por θ.

Recordar: Regresión Logística

- Puede aproximar más rápidamente una política determinística.
- En algunos contextos, la política óptima *puede no ser determinística*.

Diferentes posibilidades

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_b e^{h(s,b,\boldsymbol{\theta})}}, \qquad h(s,a,\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}(s,a),$$

Softmax

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\right),$$

Gaussianas (acciones continuas)

Funciones objetivo

- Tareas episódicas:

$$J_0(\theta) = v_{\pi_\theta}(s_0),$$

- Tareas continuas:

$$J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s) v_{\pi_\theta}(s),$$

- Recompensa promedio en un paso:

$$J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a)\mathcal{R}_s^a,$$

con d^{π_θ} la distribución invariante de $S_t | A_0, \ldots, A_{t-1}$.

Policy Gradient Theorem - Caso simple

$$J(\theta) = \sum_s d(s) E_{\pi_\theta}[R_1 | S_0 = s] = \sum_s d(s) \sum_a \pi_\theta(s,a) \mathcal{R}_s^a,$$

$$\nabla_\theta \pi_\theta(s,a) = \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)}$$
$$= \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a)$$

$$\nabla J(\theta) = \sum_s d(s) \sum_a \nabla \pi_\theta(s,a) \mathcal{R}_s^a,$$
$$= \sum_s d(s) \sum_a \pi_\theta(s,a) \nabla \log \pi_\theta(s,a) \mathcal{R}_s^a,$$
$$= \sum_s d(s) E_{\pi_\theta}[\nabla \log \pi_\theta(S_0, A_0)\ R_1 | S_0 = s].$$

Policy Gradient Theorem

$$J(\theta) = E_d[E_{\pi_\theta}[q_{\pi_\theta}(S_0, A_0)|S_0]],$$

entonces

$$\nabla_\theta J(\theta) = E_d[E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(S_0, A_0) q_{\pi_\theta}(S_0, A_0)|S_0]],$$

Me da una expresión explícita para calcular el gradiente *en términos de los ingredientes que sí puedo aproximar!*

Ejemplos de función score

$$\pi_\theta(s, a) \propto e^{\phi(s,a)^\top \theta}$$

$$\nabla_\theta \log \pi_\theta(s, a) = \phi(s, a) - \mathbb{E}_{\pi_\theta}\left[\phi(s, \cdot)\right]$$

$$\pi(a|s, \boldsymbol{\theta}) \doteq \frac{1}{\sigma(s, \boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{(a - \mu(s, \boldsymbol{\theta}))^2}{2\sigma(s, \boldsymbol{\theta})^2}\right),$$

$$\nabla_\theta \log \pi_\theta(s, a) = \frac{(a - \mu(s))\phi(s)}{\sigma^2}$$

REINFORCE - MC Policy Gradient

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ $\qquad (G_t)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Algoritmo Actor-Crítico

MC Policy Gradient tiene mucha varianza. ¿Cómo reducirla?

- crítico: Evalua la política.

$$\hat{q}(s, a; w) \approx q_{\pi_0}(s, a)$$

- actor: Actualiza los parámetros de la política siguiendo las sugerencias del crítico.

$$\Delta\theta^{k+1} = \alpha \nabla_{\theta^k} \log \pi_{\theta^k}(s, a) \hat{q}(s, a; w)$$

- Simple actor-critic algorithm based on action-value critic
- Using linear value fn approx. $Q_w(s,a) = \phi(s,a)^\top w$

  Critic Updates w by linear TD(0)

  Actor Updates θ by policy gradient

function QAC
    Initialise s, θ
    Sample $a \sim \pi_\theta$
    for each step **do**
        Sample reward $r = \mathcal{R}_s^a$; sample transition $s' \sim \mathcal{P}_{s,\cdot}^a$
        Sample action $a' \sim \pi_\theta(s', a')$
        $\delta = r + \gamma Q_w(s', a') - Q_w(s, a)$
        $\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s, a) Q_w(s, a)$
        $w \leftarrow w + \beta \delta \phi(s, a)$
        $a \leftarrow a', s \leftarrow s'$
    end for
end function

Sesgo en Actor-Crítico

$$\hat{q}(s, a; w) \approx q_{\pi_0}(s, a)$$

Introduce un sesgo en el gradiente.
¿Cómo controlar cuánto afecta en la actualización de la política?

$$\nabla_w Q_w(s, a) = \nabla_\theta \log \pi_\theta(s, a)$$

Compatible

Minimiza el EM

$$\mathbb{E}_{\pi_\theta}\left[\left(Q^{\pi_\theta}(s, a) - Q_w(s, a)\right)^2\right]$$

El gradiente es exacto

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s, a)\ Q_w(s, a)\right]$$

Reduciendo la varianza de MC Policy Gradient

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) B(s)\right] &= \sum_{s \in \mathcal{S}} d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(s,a) B(s) \\ &= \sum_{s \in \mathcal{S}} d^{\pi_\theta} B(s) \nabla_\theta \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \\ &= 0\end{aligned}$$

$$B(s) = V^{\pi_\theta}(s)$$

Restarle una *Baseline* puede reducir la varianza **sin modificar el gradiente.**

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\; A^{\pi_\theta}(s,a)\right]$$

Más parámetros...

Actualizar las dos funciones de valor usando TD

$$V_v(s) \approx V^{\pi_\theta}(s)$$

$$Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$$

$$A(s,a) = Q_w(s,a) - V_v(s)$$

Función de Ventaja

Otra posibilidad

$$\begin{aligned} A^{\pi_\theta}(s,a) &\approx r^+ + \gamma V^{\pi_\theta}(s^+) - V^{\pi_\theta}(s) \\ &\approx r^+ + \gamma V_v(s^+) - V_v(s) \end{aligned}$$

- For the true value function $V^{\pi_\theta}(s)$, the TD error δ^{π_θ}

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

- is an unbiased estimate of the advantage function

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[\delta^{\pi_\theta}|s,a\right] &= \mathbb{E}_{\pi_\theta}\left[r + \gamma V^{\pi_\theta}(s')|s,a\right] - V^{\pi_\theta}(s) \\ &= Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \\ &= A^{\pi_\theta}(s,a)\end{aligned}$$

- So we can use the TD error to compute the policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ \delta^{\pi_\theta}\right]$$

- In practice we can use an approximate TD error

$$\delta_v = r + \gamma V_v(s') - V_v(s)$$

- This approach only requires one set of critic parameters v

Exploración vs Explotación

- **Explotación**: Tomar la acción que es **más conveniente** *en el momento*.
- **Exploración**: Tomar **decisiones sub-óptimas** con el propósito de *obtener más información*.

Ejemplos:

¿Qué publicidad muestro? ¿Dónde realizó pozos de petróleo?

Bandidos Multi-brazo

Sólo hay acciones y recompensas.

$$A_t \longrightarrow R_t$$

Función de valor

$$q_*(a) = E[R_t \mid A_t = a]$$

$$a = 1, \ldots, k$$

En general, *no dispongo de toda la información sobre la recompensa* de cada acción!

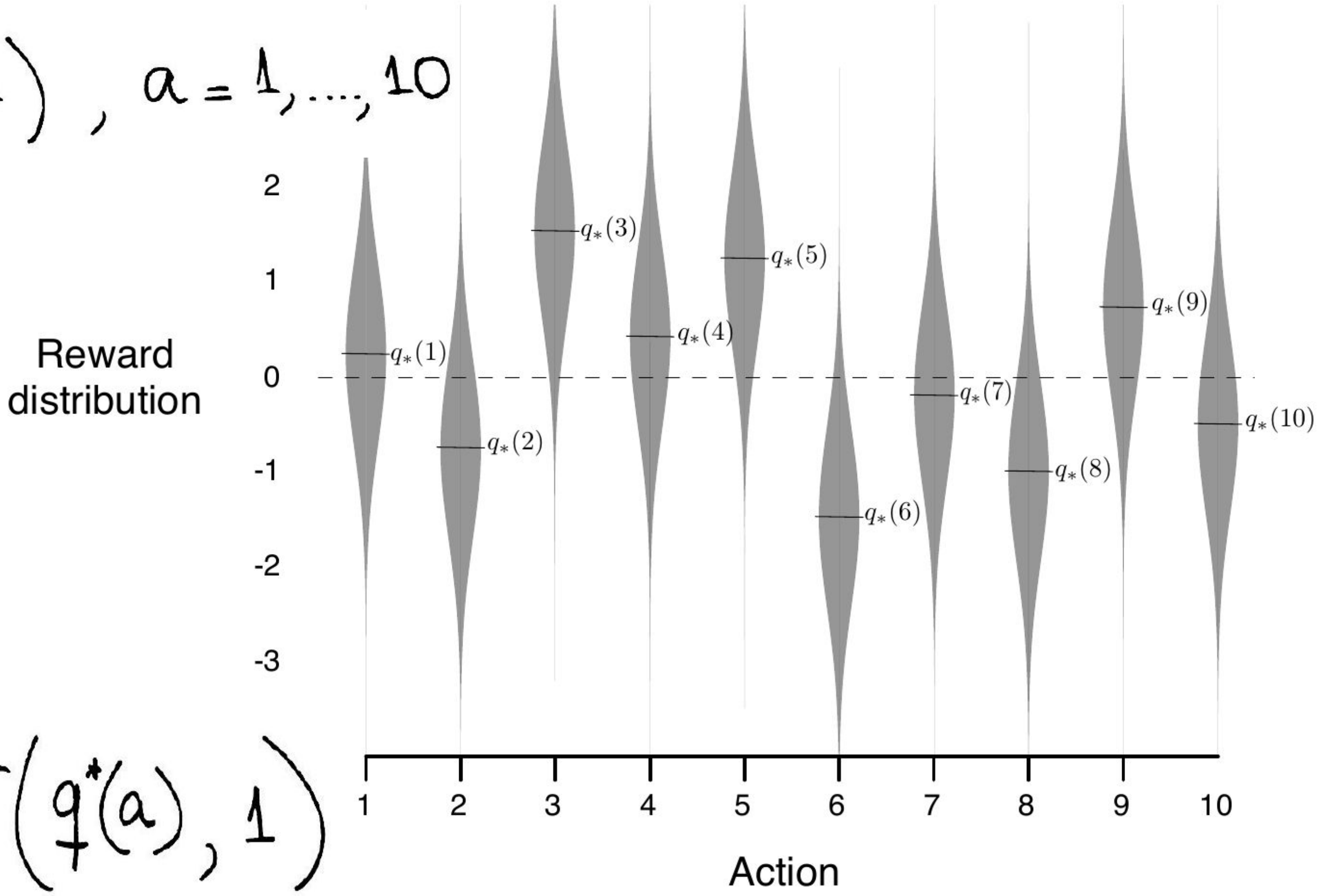
Métodos basados en la función de valor

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \, \mathbb{1}_{A_i = a}}{N_t(a)}$$

$$A_t = \underset{a}{\operatorname{argmax}} \; Q_t(a)$$

10-armed Testbed

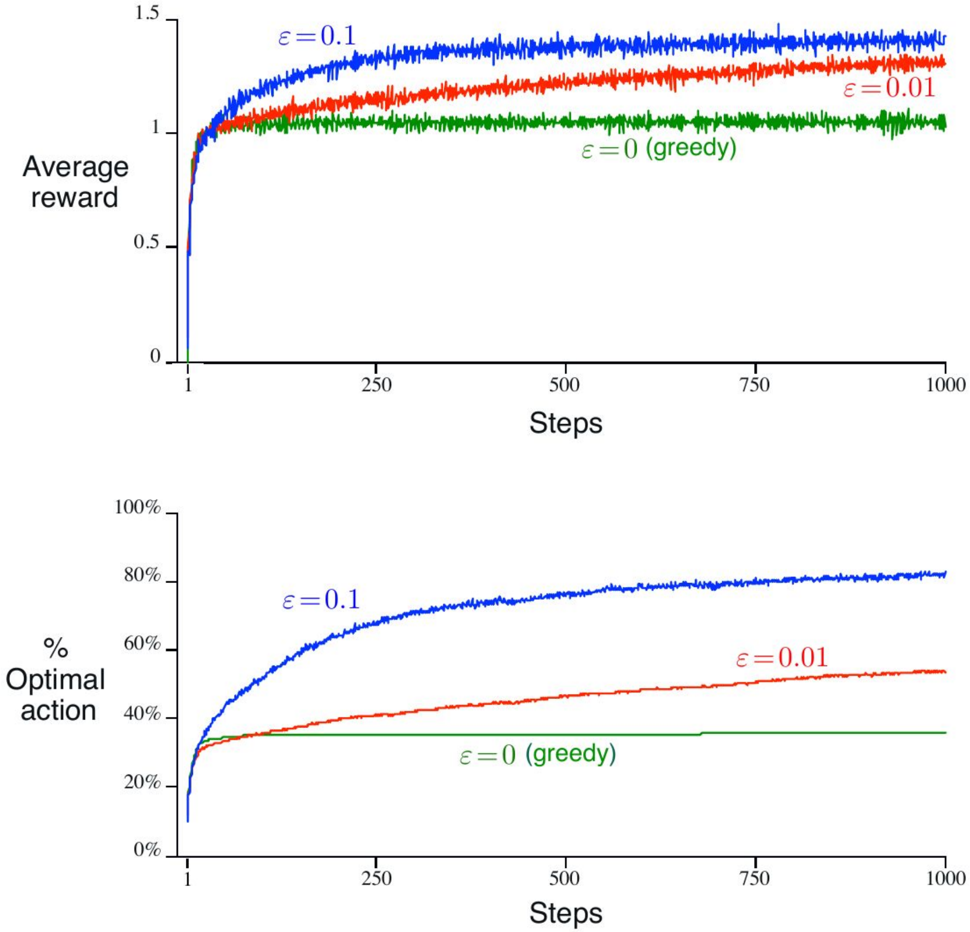
$$q^*(a) \sim \mathcal{N}(0, 1), \quad a = 1, \ldots, 10$$



$$R_t \mid A_t = a \sim \mathcal{N}(q^*(a), 1)$$

10-armed Testbed - e-greedy

- Dependiendo del ruído de la recompensa, se modifica el rendimiento del ε greedy.
- Inclusive, en casos donde la recompensa es determinística (en función de la acción) puede convenir ε-greedy (caso no estacionario).



Algoritmo y caso no estacionario

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$$Q(a) \leftarrow 0$$
$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$
$$R \leftarrow bandit(A)$$
$$N(A) \leftarrow N(A) + 1$$
$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\left[R - Q(A)\right]$$

Caso no estacionario

$$Q_{n+1} \doteq Q_n + \alpha \Big[R_n - Q_n \Big] = \quad (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1-\alpha)^{n-i} R_i .$$

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty$$

Controla la fluctuaciones del comienzo

$$\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty .$$

Garantiza la convergencia

Método "Optimístico"

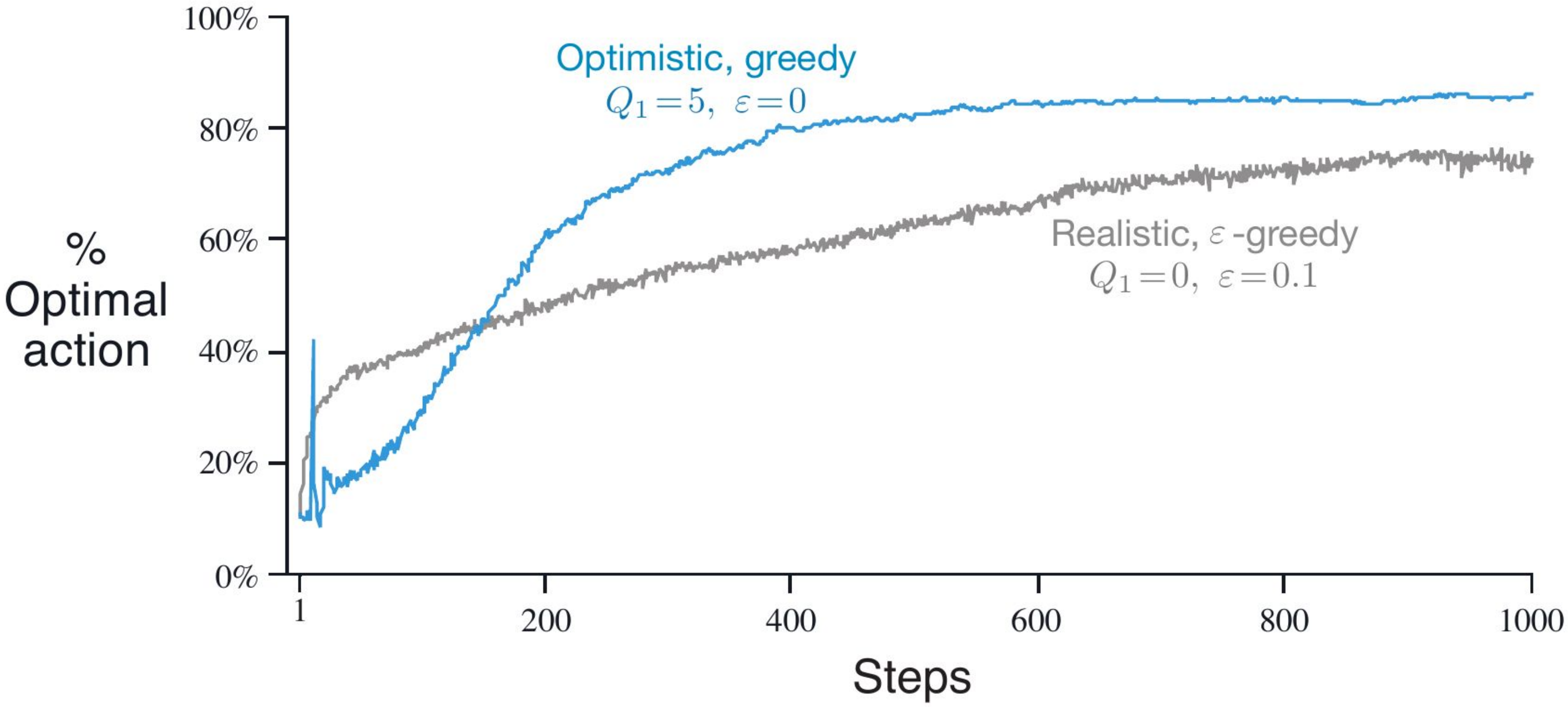
El método anterior es muy dependiente de las *condiciones iniciales*,

$$Q_1(a)$$

$$Q_2(a) = Q_1(a) + \frac{1}{N_1(a)}\left[R_1 - Q_1(a)\right]$$

Si defino todas las condiciones iniciales de manera ***optimista***, la recompensa no va a cumplir con las expectativas por lo que *muchas veces voy a cambiar de acción*.

Es decir, **voy a explorar**!


100%
80%
60%
40%
20%
0%
% Optimal action
Optimistic, greedy
$Q_1 = 5, \ \varepsilon = 0$
Realistic, ε-greedy
$Q_1 = 0, \ \varepsilon = 0.1$
1
200
400
600
800
1000
Steps

Selección de acción basada en Cota Superior de Confianza (UCB)

e-greedy selecciona las acciones no óptimas *sin utilizar **nada** de información* sobre las mismas.

Cota de la varianza del Q

$$A_t = \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$