Agentes Inteligentes

Franz Mayr
mayr@ort.edu.uy

Universidad ORT Uruguay
8 de mayo de 2023

9. Sesgo de Maximización y Doble Q-Learning

Sesgo de Maximización

- Los algoritmos TD vistos hasta el momento implican maximizar al armar el target. Esto genera un sesgo de maximización (se sobreestima el máximo).

- ¿Por qué? Se tiende a seleccionar como estimación del valor máximo, el máximo valor estimado. O sea, se "confunde" el máximo de las estimaciones con la estimación del máximo.

- ¿Cómo prevenirlo? Usar estimaciones no sesgadas.

Repaso: SARSA

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

Repetir:

Inicializar S

$A \leftarrow$ acción desde S según política ε-greedy basada en Q

Repetir:

Ejecutar la acción A; observar R, S'

$A' \leftarrow$ acción desde S' según política ε-greedy basada en Q

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

$S \leftarrow S'$

$A \leftarrow A'$

hasta que S sea terminal

La tasa de aprendizaje $\alpha \in (0, 1]$ es un parámetro del algoritmo.

Repaso: Q-Learning

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

Repetir:

  Inicializar S

  Repetir:

    $A \leftarrow$ acción desde S según política ε-greedy basada en Q

    Ejecutar la acción A; observar R, S'

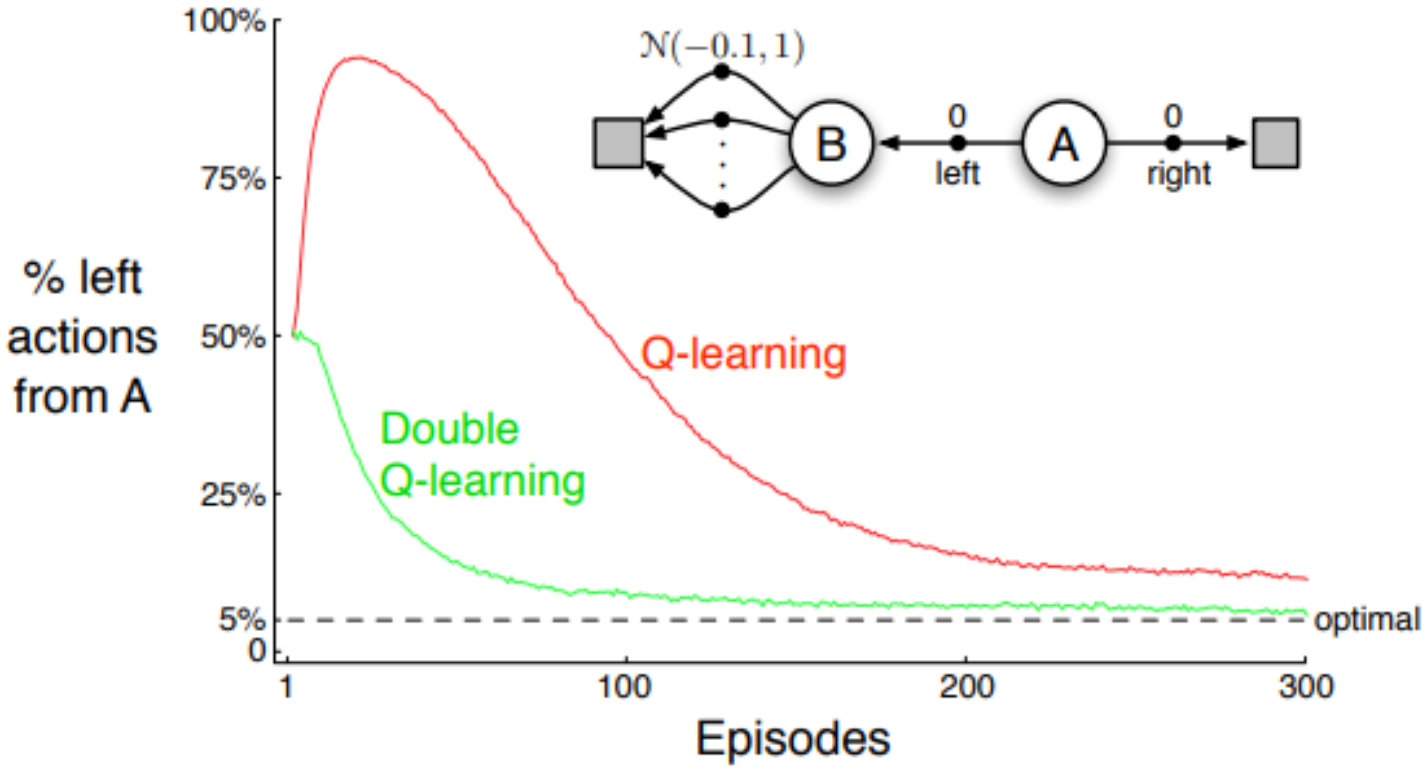
    $Q(S, A) \leftarrow Q(S, A) + \alpha \left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$

    $S \leftarrow S'$

  hasta que S sea terminal

La tasa de aprendizaje $\alpha \in (0, 1]$ es un parámetro del algoritmo.

Ejemplo


100%
75%
50%
25%
5%
0
% left actions from A
$\mathcal{N}(-0.1, 1)$
B
A
0
left
0
right
Q-learning
Double Q-learning
optimal
1
100
200
300
Episodes

Double Q-Learning

Inicializar $Q_1(s,a)$ y $Q_2(s,a)$ arbitrariamente $\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repetir:

Inicializar S

Repetir:

$A \leftarrow$ acción desde S según política ε-greedy basada en $Q_1 + Q_2$

Ejecutar la acción A; observar R, S'

Con probabilidad 0.5:

$Q_1(S,A) \leftarrow Q_1(S,A) + \alpha \left[R + \gamma\, Q_2(S', argmax_a Q_1(S',a)) - Q_1(S,A)\right]$

Caso contrario:

$Q_2(S,A) \leftarrow Q_2(S,A) + \alpha \left[R + \gamma\, Q_1(S', argmax_a Q_2(S',a)) - Q_2(S,A)\right]$

$S \leftarrow S'$

hasta que S sea terminal

La tasa de aprendizaje $\alpha \in (0,1]$ es un parámetro del algoritmo.

Repaso: Q-learning semi-gradiente para estimar q_*

$\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$ es una función parametrizable
con pesos $\mathbf{w} \in \mathbb{R}^d$ (ej.: red neuronal)

Inicializar $\mathbf{w} \in \mathbb{R}^d$ arbitrariamente

Repetir:
  Inicializar S
  Repetir:
    $A' \leftarrow$ acción desde S' según polít. ε-greedy basada en $\hat{q}(S, \cdot, \mathbf{w})$
    Ejecutar la acción A; observar R, S'
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha \left[R + \gamma \max_a \hat{q}(S', a, \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\right] \nabla \hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
  hasta que S sea terminal

Repaso: Q-learning semi-gradiente para estimar q_*

$\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$ es una función parametrizable
con pesos $\mathbf{w} \in \mathbb{R}^d$ (ej.: red neuronal)

Inicializar $\mathbf{w} \in \mathbb{R}^d$ arbitrariamente

Repetir:
  Inicializar S
  Repetir:
    $A' \leftarrow$ acción desde S' según polít. ε-greedy basada en $\hat{q}(S, \cdot, \mathbf{w})$
    Ejecutar la acción A; observar R, S'
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha \left[R + \gamma \operatorname{máx}_a \hat{q}(S', a, \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\right] \nabla \hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
  hasta que S sea terminal

En estos algoritmos no hay garantía de convergencia. Igual se los usa mucho. Observación: Los algoritmos tabulares son casos particulares, y sí poseen buenas propiedades de convergencia.

Doube Deep Q-learning

$\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$ es una función parametrizable
con dos sets de pesos $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$

Inicializar $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$ arbitrariamente
Repetir:
  Inicializar S
  Repetir:
    $A' \leftarrow$ acción desde S' según polít. ε-greedy basada en
    $\hat{q}(S, \cdot, \mathbf{w}_1)$ y $\hat{q}(S, \cdot, \mathbf{w}_2)$
    Ejecutar la acción A; observar R, S'
    Con probabilidad 0.5:
    $\mathbf{w}_1 \leftarrow \mathbf{w}_1 + \alpha \left[R + \gamma\, \hat{q}(S', argmax_a \hat{q}(S', a, \mathbf{w}_1), \mathbf{w}_2) - \hat{q}(S, A, \mathbf{w}_1)\right] \nabla \hat{q}(S, A, \mathbf{w}_1)$
    Caso contrario:
    $\mathbf{w}_2 \leftarrow \mathbf{w}_2 + \alpha \left[R + \gamma\, \hat{q}(S', argmax_a \hat{q}(S', a, \mathbf{w}_2), \mathbf{w}_1) - \hat{q}(S, A, \mathbf{w}_2)\right] \nabla \hat{q}(S, A, \mathbf{w}_2)$
    $S \leftarrow S'$
  hasta que S sea terminal

Bibliografía

- Reinforcement Learning. An Introduction", R.S. Sutton A.G. Barto (2018), MIT Press, 2nd ed.
- Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16). AAAI Press, 2094–2100.