Métodos tabulares para aprendizaje por refuerzo
Segunda parte

Lorenzo Mandow
lawrence@lcc.uma.es
Universidad de Málaga

Fuentes principales

- “Reinforcement Learning. An introduction” R.S. Sutton & A. G. Barto. The MIT Press (1998). Capítulos 4 y 6.
- “Artificial Intelligence: a modern approach” S. Russell & P. Norvig. Pearson (3ª ed.) (2010). Capítulos 17 y 21.

Breve Índice

- Introducción
- Programación dinámica
  - Evaluación de políticas
  - Mejora de políticas
- Aprendizaje por diferencias temporales
  - Algoritmo TD(0)
  - Algoritmo Q-learning

Diferencias temporales

- El aprendizaje por diferencias temporales permite resolver un problema de decisión markoviano,:

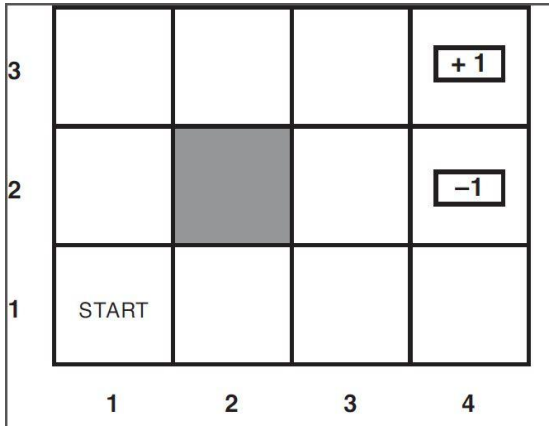
a) A partir de **experiencia** en el entorno.

b) Cuando no disponemos de un **modelo** de la dinámica del entorno - p(s'/s,a)

Diferencias temporales

- Supongamos que partimos de una política π arbitraria y nuestro agente se desenvuelve en el entorno.
- Llamaremos episodio a una secuencia alterna de estados y acciones que acaba e un estado final.

Diferencias temporales



Algunos episodios podrían ser:

(1,1) ↑ (1,2) ↑ (1,2) ↑ (1,3) →(2,3) →(3,3) →(3,2) ↑ (3,3) → **(4,3)**

(1,1) ↑ (2,1) →(3,1) ↑ (3,2) ↑ (3,3) → **(4,3)**

…

Imagen: Russell & Norvig. AIMA

Diferencias temporales

- Al igual que en el caso de la programación dinámica, estudiaremos cómo evaluar una política y cómo mejorarla, aunque esta vez sin utilizar un modelo, y únicamente a partir de la experiencia disponible.
- Iteración de políticas:

$$\pi_0 \rightarrow v_0 \rightarrow \pi_1 \rightarrow v_1 \rightarrow \ldots \pi_* \rightarrow v_*$$

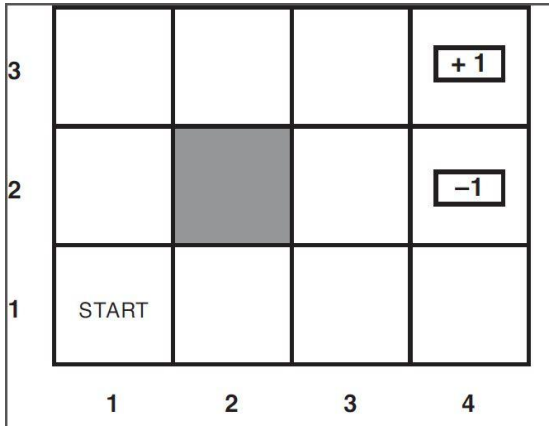
Algoritmo TD(0)

- El método TD(0) permite aproximar el valor de un estado cada vez que un episodio pasa por el. Sea S_t el estado visitado en el paso t de un episodio, y S_{t+1} el siguiente estado visitado.

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

Algoritmo TD(0)

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



Supongamos que en un momento dado

V(1,1) = 0.400
V(1,2) = 0.550

Al realizar la transición de (1,1) a (1,2):

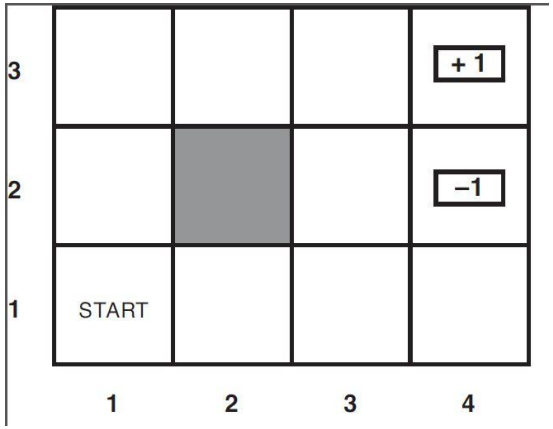
V(1,1) ← 0.400 + 0.1[-0.04 + 0.550 – 0.400] = 0.411

(1,1) ↑ (1,2) ↑ (1,2) ↑ (1,3) →(2,3) →(3,3) →(3,2) ↑ (3,3) → **(4,3)**

Imagen: Russell & Norvig. AIMA

Algoritmo TD(0)

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



Supongamos que al inicial el segundo episodio tenemos:

V(1,1) = 0.411
V(2,1) = 0.300
....

V(1,1) ← 0.411 + 0.1[-0.04 + 0.300 – 0.411] = 0.3959

(1,1) ↑ (2,1) →(3,1) ↑ (3,2) ↑ (3,3) → **(4,3)**

Imagen: Russell & Norvig. AIMA

Algoritmo TD(0)

- Esto nos permite evaluar una política, pero para mejorarla seguiríamos necesitando un modelo de la dinámica del entorno.

$$\pi'(s) = \underset{a \in A(s)}{\text{argmáx}} \sum_{s'} p(s'/s,a) v_\pi(s')$$

- Una posibilidad es aprender también el modelo – *p(s'/s,a)*

Breve Índice

- Introducción
- Programación dinámica
  - Evaluación de políticas
  - Mejora de políticas
- Aprendizaje por diferencias temporales
  - Algoritmo TD(0)
  - Algoritmo Q-learning

Algoritmo Q-learning

- Una solución alternativa es aprender el valor de las acciones, en lugar de los estados.
- Denominaremos *Q(s,a)* al valor esperado de realizar la acción a en el estado *s*.

$$V_\pi(s) = \max_{a \in A(s)} Q_\pi(s,a)$$

Algoritmo Q-learning

- La ecuación para aproximar los valores de Q es análoga a la del algoritmo TD(0):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

- Para seleccionar la acción a realizar en el estado *s*, sólo hay que consultar los valores de *Q(s,a)*.

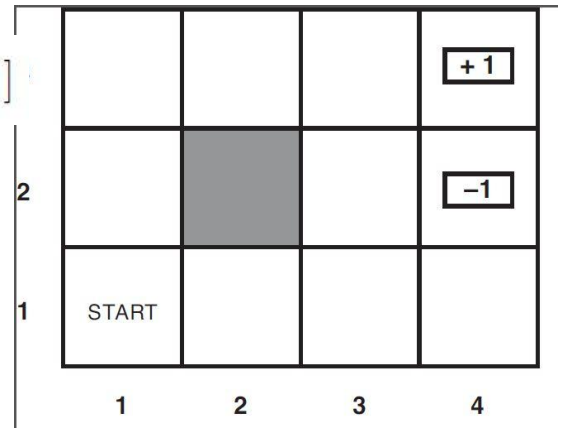
Algoritmo Q-learning

Para asegurar una correcta convergencia, es necesario combinar de alguna manera la explotación del conocimiento adquirido con una exploración de posibilidades.

Dicho de otra manera, en la interacción con el entorno no se debe seguir siempre el dictado de la política, sino que con al menos una pequeña probabilidad, se deben explorar acciones que de otro modo no serían elegidas.

Algoritmo Q-learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$



Q(1,1, ↑) = 0.4
Q(1,1, →) = 0.3
Q(1,1, ↓) = 0.1
Q(1,1, ←) = 0.2

Q(2,1, ↑) = 0.6
Q(2,1, →) = 0.3
Q(2,1, ↓) = 0.2
Q(2,1, ←) = 0.3

Ejercicio propuesto: ¿Cuál será el valor de Q(1,1, ↑) tras la primera transición del episodio mostrado abajo?
Suponer α = 0,1 y γ= 1

(1,1) ↑ (2,1) →(3,1) ↑ (3,2) ↑ (3,3) → **(4,3)**

Imagen: Russell & Norvig. AIMA

Breve Índice

- Introducción
- Programación dinámica
  - Evaluación de políticas
  - Mejora de políticas
- Aprendizaje por diferencias temporales
  - Algoritmo TD(0)
  - Algoritmo Q-learning