UA

cursos d'estiu | cursos de verano
rafael altamira

Introducción al *Machine Learning*

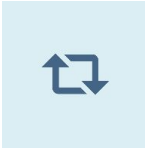
Aprendizaje por refuerzo

Profesor: Jorge Calvo Zaragoza

Tipos de aprendizaje

- **Aprendizaje supervisado**

- **Aprendizaje no supervisado**

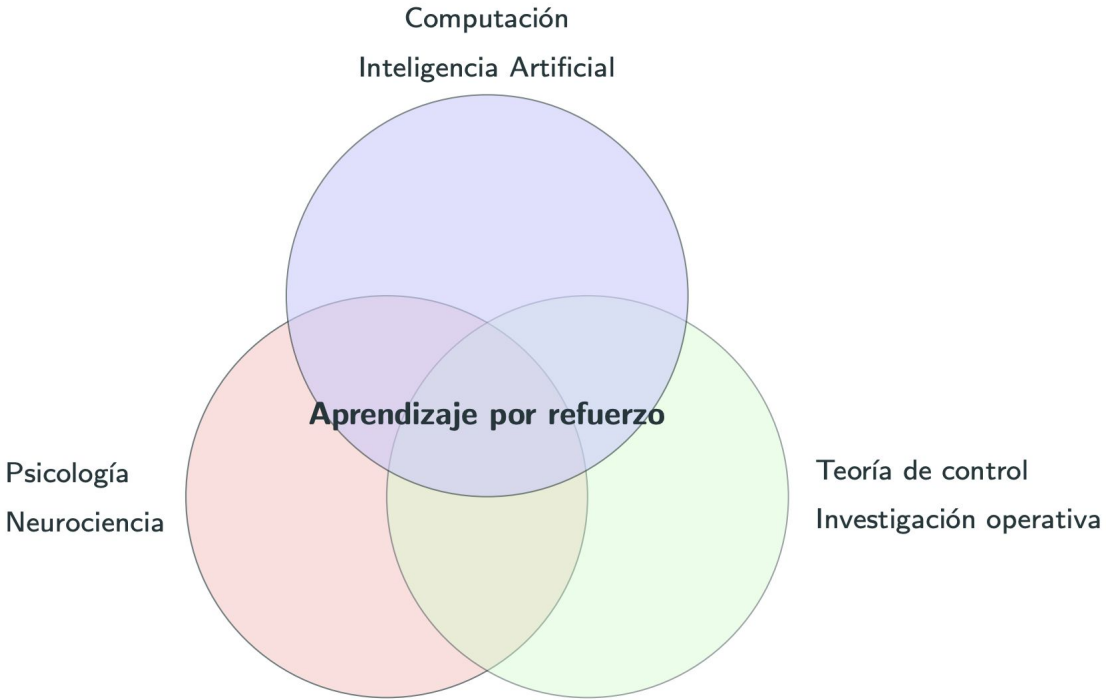


- **Aprendizaje por refuerzo**

Contenidos

- El aprendizaje por refuerzo (*Reinforcement Learning*, **RL**) se basa en aprender de **la experiencia** a través de **recompensas**.
- Esto es en contraste con aprender a partir de **ejemplos**.

Dominio


Computación
Inteligencia Artificial
Aprendizaje por refuerzo
Psicología
Neurociencia
Teoría de control
Investigación operativa

Ejemplos de RL

- Un robot móvil decide si entrar en una nueva habitación en busca de más basura para recoger o comenzar a buscar el camino de regreso a su estación de recarga.
- Un jugador de ajedrez decide su próximo movimiento.
- Una cría de gacela apenas se mantiene en pie justo después de nacer. Media hora más tarde, corre a 30 km/h.

Usos clásicos del RL

Los usos más típicos para mostrar el poder de RL son los videojuegos (bots) y la robótica:

- Videojuegos: aprender a jugar a la Atari
- Robótica: resolviendo el cubo de Rubik

Investigación en RL

- Aprender a partir de la experiencia se considera un problema **abierto**.
- El aprendizaje por refuerzo se ha convertido en una de las áreas de investigación **más activas**.
  - Drástica popularidad desde que un modelo de Google DeepMind derrotó al campeón del mundo del *Go* con aprendizaje por refuerzo.

Usos recientes del RL

- **AlphaFold**: predicción de estructura de proteína a partir de su secuencia de aminoácidos.
- Descubrimiento de **nuevos algoritmos**:
  - Ordenación
  - Multiplicación de matrices
- ***Humanizar*** las respuestas de modelos de conversación (ChatGPT).

Contenidos

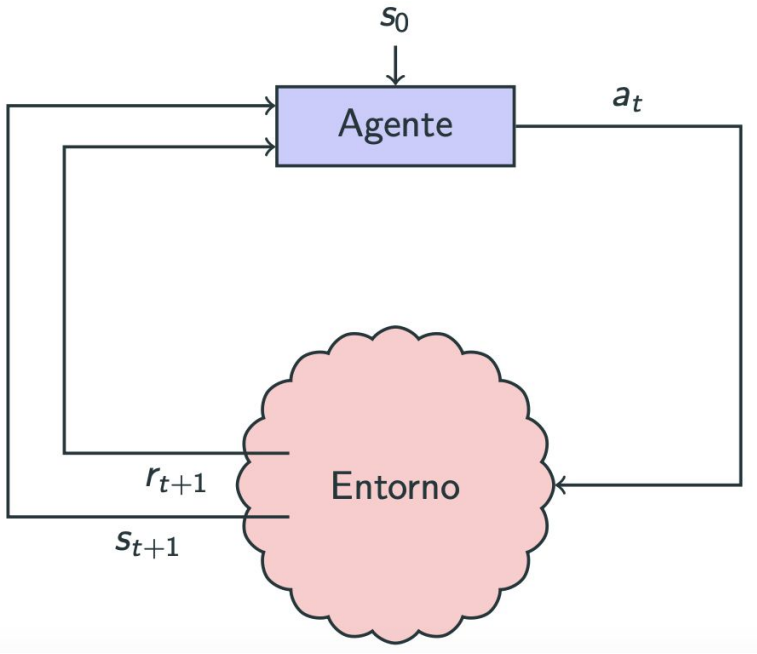
- Fundamentos del aprendizaje por refuerzo
- Q-Learning
- Caso práctico

Fundamentos

Formulación

- El aprendizaje por refuerzo proporciona enfoques al problema de toma de decisiones secuencial.
  - Decidir qué **acción** tomar en cada **estado**.
  - La **acción** provoca un cambio de **estado** y una **recompensa**.
  - Se busca maximizar la **recompensa acumulada** a lo largo de todo el proceso.

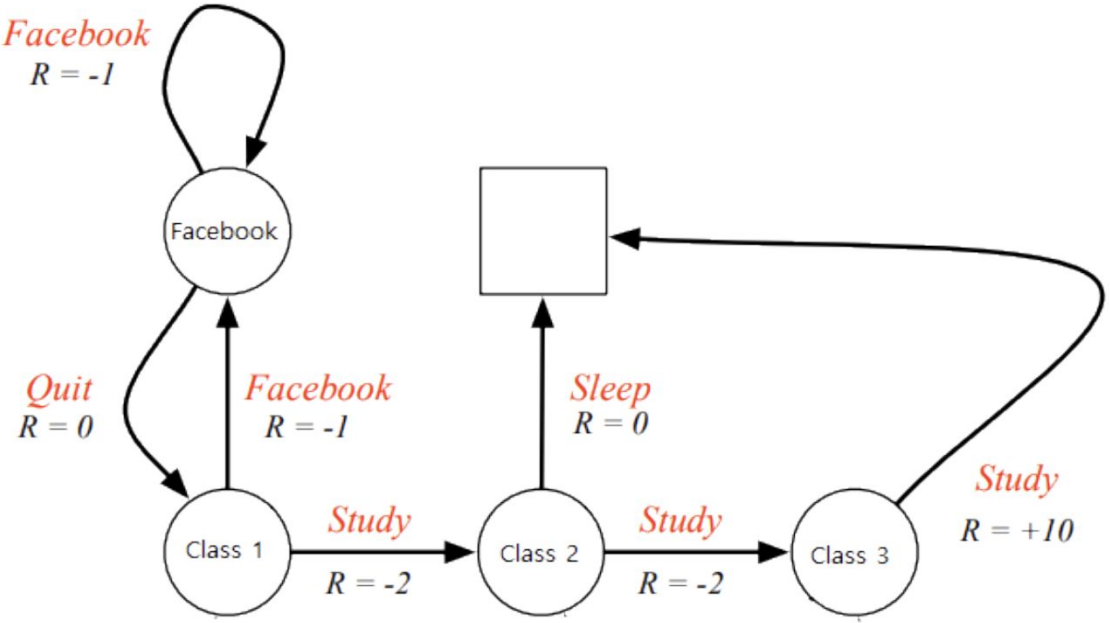
Formulación



Objetivo

maximizar $\sum_i r_i$

Formulación



Créditos imagen: "*Introduction to reinforcement learning.*" (DeepMind)

¿Qué es una recompensa?

- En aprendizaje por refuerzo, la recompensa es un valor escalar.
- ¿Suficiente para definir objetivos? Hipótesis de la recompensa.
- La elección de la recompensa juega un papel **fundamental**.
- No cualquier recompensa correlaciona con el objetivo perseguido: Ejemplo CoastRunners.

¿Cuál sería una buena recompensa?

1. Jugando al ajedrez.
2. Tirando un penalty.
3. Humanizando las respuestas de un *chatbot*.

Política

- El objetivo último es aprender una **estrategia para maximizar la recompensa acumulada**.
- En RL, la estrategia se conoce como política (*policy*).
- Puede ser **determinista** o **estocástica**.

¿Por qué es complicado el RL?

- Estimación del valor de una acción
- Dilema exploración-explotación

¿Por qué es complicado el RL?

- Estimación del valor de una acción
  - El valor se mide como el retorno que podemos esperar de una acción desde un estado.
  - Todos los algoritmos de RL tratan de estimar esto (explícita o implícitamente).
  - El problema fundamental es correlacionar a**cciones inmediatas con retornos futuros**.

¿Por qué es complicado el RL?

- Dilema exploración-explotación
  - No es posible saber si una política es óptima.
  - El agente debe **explorar** para obtener nueva información del entorno, a la vez que **explota** el conocimiento adquirido para obtener el mayor retorno.
  - Dilema:
    - ¿Qué plato pido en un restaurante?
    - ¿Qué ruta hago de camino a casa?

¿Por qué es complicado el RL?

Ejemplo

Algoritmos

- Existen multitud de enfoques para RL, con muchos algoritmos para cada uno de ellos (algoritmos de búsqueda, algoritmos evolutivos, algoritmos de aprendizaje...).
- Nos centraremos en un algoritmo muy conocido de aprendizaje para pares estado-acción: Q-Learning.

Q-Learning

Q-Learning

- Uno de los algoritmos clave en aprendizaje por refuerzo se conoce como **Q-Learning**.
  - Se basa en estimar una función de valor para cada par estado-acción.
  - Se implementa mediante una **tabla** (Q) con dos índices.
  - La tabla se actualiza tras cada acción, acercando la estimación a la que viene dada por la nueva experiencia.

Q-Learning: pasos

- **Inicialización**: se inicializa la tabla con valores escogidos o aleatorios.
- **Actualización**: asumiendo que
  - Se escoge una acción *at* desde el estado *st*
  - Se pasa al estado s_{t+1} y se recibe la recompensa r_{t+1}

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \right)$$

  - α es la tasa de aprendizaje y γ es el factor de descuento

Q-Learning: algoritmo

$Q \leftarrow \text{initialize}(S, A)$;
while *number of episodes is not reached* **do**
  $s_t = s_0$;
  while *episode is active* **do**
    $a_t \leftarrow \pi_Q(s_t)$;
    $s_{t+1}, r_{t+1} \leftarrow \text{perform}(a_t)$;
    $Q[s_t, a_t] += \alpha \left(r_{t+1} + \gamma \max_a Q[s_{t+1}, a] - Q[s_t, a_t]\right)$;
    $s_t = s_{t+1}$;
  end
end

Q-Learning: politica

- Q-Learning sirve para estimar el valor de realizar una determinada acción en un determinado estado.
- Otra cuestión diferente es determinar qué acción tomar cuando el agente se encuentra en un determinado estado (política).

¿Por qué esta diferencia?

Q-Learning: políticas

Estando en un estado concreto.

- **Aleatoria**: escogemos una acción al azar.
- **Voraz**: escogemos la acción que maximiza el valor para ese estado según la tabla Q.
- **ϵ-voraz**: con probabilidad $\epsilon \in [0, 1]$ escogemos una acción al azar; si no, escogemos la voraz.

Lo habitual es utilizar una política ϵ-voraz.

Q-Learning: ϵ-voraz

Balance exploración-explotación

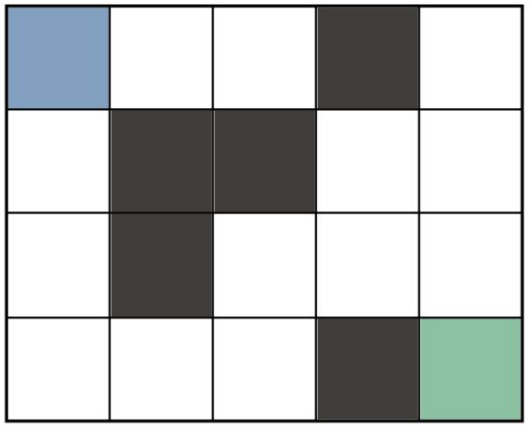
- En las primeras iteraciones, la estimación es pobre: interesa explorar para conocer mejor el entorno.
- A medida que la estimación mejora, nos interesa explotar.

¿Cómo podemos adaptar este balance de forma gradual?

- Decaimiento de ϵ:
  - Tras cada episodio $\epsilon = \epsilon \cdot \delta$
    - donde $\delta \in [0, 1]$ regula la velocidad de decaimiento.

Ejemplo «escapar del laberinto»

Un robot se mueve por un laberinto discreto. Debe llegar desde el principio (celda azul) hasta la salida (celda verde). Cada estado es un par (fila, columna).



- Las acciones son {abajo, izquierda, arriba, derecha}.
- Las recompensas están condicionadas a la celda a la que llega:
  - Gris: -10
  - Blanca/Azul: -1 (¿0?)
  - Verde: +10
- Moverse hacia fuera del laberinto no provoca cambios en el estado.

Ejemplo «escapar del laberinto»

Inicialización

Aleatoria uniforme [-4,4]

Q[(i,j),'abajo']

2.0	-3.0	-2.0	0.0	0.0
4.0	4.0	3.0	1.0	-4.0
4.0	-4.0	-3.0	2.0	0.0
-4.0	-2.0	2.0	2.0	0.0

Q[(i,j),'izquierda']

-2.0	4.0	-1.0	-3.0	2.0
-5.0	-5.0	-2.0	-3.0	1.0
-4.0	-1.0	2.0	-4.0	0.0
0.0	4.0	3.0	-1.0	0.0

Q[(i,j),'arriba']

-1.0	1.0	2.0	-1.0	-4.0
0.0	-3.0	-3.0	-1.0	-2.0
3.0	-2.0	-5.0	-2.0	4.0
4.0	2.0	-1.0	4.0	0.0

Q[(i,j),'derecha']

1.0	1.0	2.0	2.0	-4.0
-1.0	3.0	1.0	-1.0	3.0
3.0	4.0	1.0	-2.0	-4.0
-2.0	-4.0	1.0	-4.0	0.0

Ejemplo «escapar del laberinto»

Aprendizaje

- 200 episodios
- δ = 0.9
- α = 0.2
- γ = 0.99

Q[(i,j),'abajo']

1.9	-8.0	-6.0	0.4	-0.9
3.1	-3.9	3.2	3.3	-2.0
4.3	-4.0	-3.0	-2.5	11.8
-1.5	-1.2	4.6	1.1	0.0

Q[(i,j),'izquierda']

-3.6	-4.4	-5.0	-3.0	-2.0
-2.8	-5.0	-2.0	-3.0	-0.1
-2.8	-1.6	-1.5	-4.0	0.7
-2.0	-2.5	0.8	-1.0	0.0

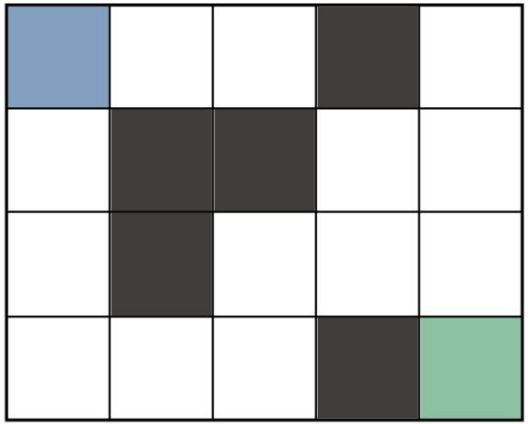
Q[(i,j),'arriba']

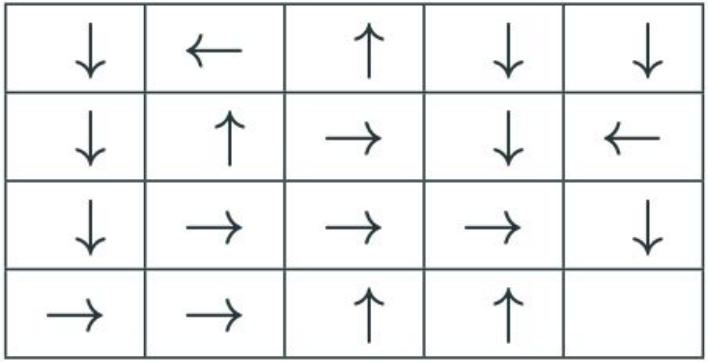
-1.6	-5.4	-4.9	-0.9	-4.0
-5.3	-3.0	-3.0	-2.6	-2.0
-3.6	-3.4	-5.0	-2.0	1.2
-2.8	-3.4	8.0	3.5	0.0

Q[(i,j),'derecha']

-5.8	-5.4	-7.1	-0.2	-4.0
-7.4	-3.8	1.0	-0.8	-1.2
-5.9	3.1	9.2	9.9	-4.0
5.5	6.7	-3.8	-4.0	0.0

Ejemplo «escapar del laberinto»





Decisión voraz por estado

¿Por qué no se ha estimado la acción óptima de cada estado?

Q-Learning

- Algoritmo muy popular hasta la llegada del Deep Learning.
- Ejemplo Robocup
- Veamos ahora un ejemplo propio sencillo: CartPole.

Cierre

Aunque el aprendizaje por refuerzo ofrece una formulación robusta en la teoría, llevarlo a la práctica (real) puede ser complejo.

- **Coste del aprendizaje**: no siempre se puede permitir un proceso de aprendizaje (tiempo, coste, factibilidad).
- **Diferencia entre entorno simulado y entorno real**: los entornos simulados —aunque cada vez más realistas— no siempre pueden modelar la casuística del mundo real al completo.
- **Diseño de recompensas apropiadas**: no siempre una recompensa que modela bien el objetivo es adecuada para aprender.

Cómo continuar (I)

1. Aprendizaje automático (Machine Learning)
2. Aprendizaje profundo (Deep Learning)
3. Aprendizaje profundo por refuerzo (Deep Reinforcement Learning)

Cómo continuar (II)

- Libro Sutton y Barto
- Curso DeepMind (Youtube)
- OpenAI Spinning up

UA

cursos d'estiu | cursos de verano
rafael altamira

Introducción al *Machine Learning*

Aprendizaje por refuerzo

Profesor: Jorge Calvo Zaragoza