Aprendizaje Automático

# Aprendizaje por refuerzo y control

# Índice


Esquema 3

Ideas clave 4

12.1. ¿Cómo estudiar este tema? 4

12.2. Introducción al aprendizaje por refuerzo 4

12.3. Algoritmos de aprendizaje por refuerzo 7

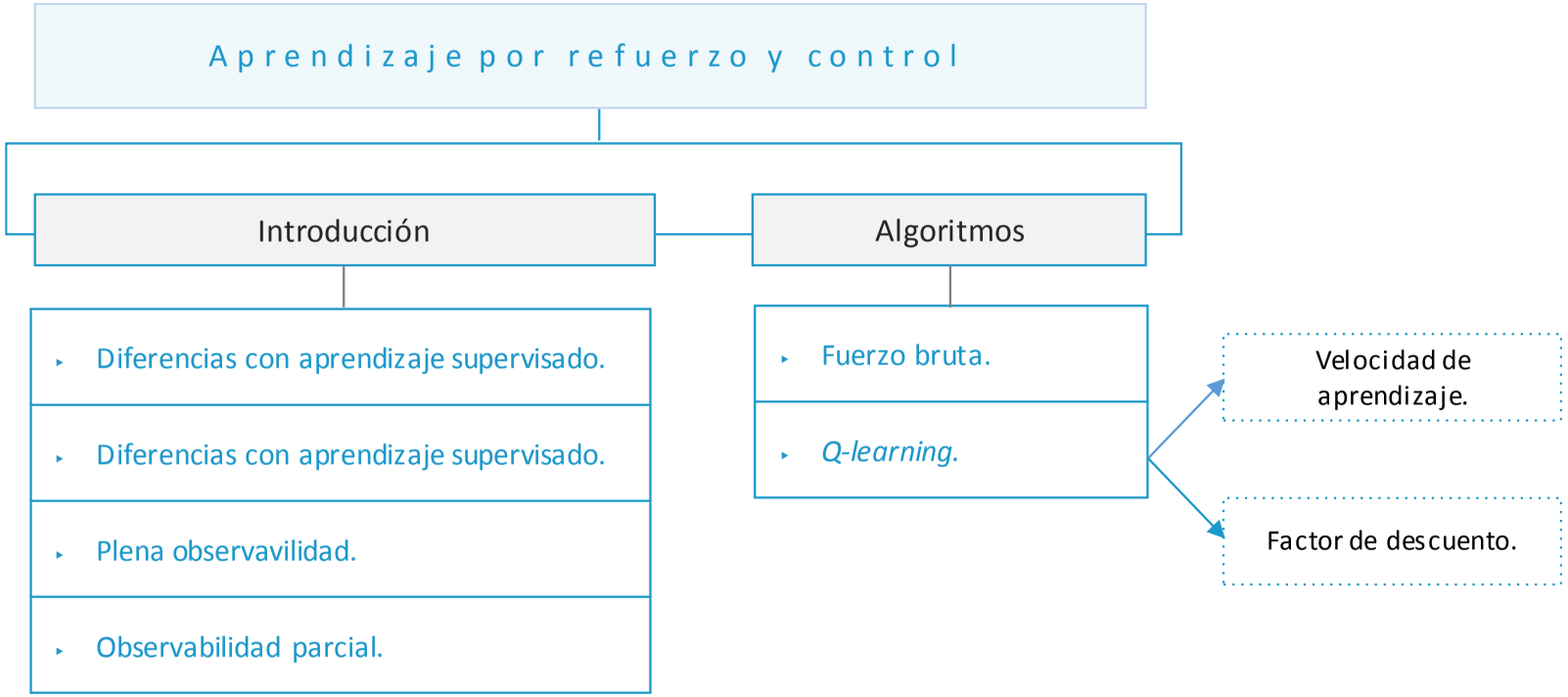
12.4. Referencias bibliográficas 11

Lo + recomendado 12

+ Información 15

Test 16

# Esquema



# Ideas clave

## 12.1. ¿Cómo estudiar este tema?

Estudia este tema a través de las **Ideas clave** disponibles a continuación.

En este tema se presentan los **algoritmos de aprendizaje por refuerzo.** En primer lugar, se realiza una introducción a los mismos y se presentan las diferencias de estos algoritmos con las técnicas de aprendizaje supervisado y aprendizaje no supervisado.

A continuación, se describe formalmente un proceso de decisión de Markov y los mecanismos que se realizan para aprender del entorno.

Posteriormente, se describen los algoritmos de aprendizaje por refuerzo basados en fuerza bruta y el algoritmo *Q-learning.*

## 12.2. Introducción al aprendizaje por refuerzo

Los seres vivos, y en particular los seres humanos, aprendemos las acciones a realizar en función del *feedback* o resultado que hemos observado por estas acciones previamente. Este aprendizaje se basa en las **técnicas de aprendizaje por refuerzo,** las cuales están inspiradas en la psicología conductista.

Uno de los ejemplos más populares y conocidos del aprendizaje por refuerzo es el **estudio del perro de Iván Pávlov** (1849-1936), donde se condicionaba a la mascota en función de un premio o penalización por sus acciones.

Curiosamente, la aproximación del aprendizaje supervisado existe con una mayor frecuencia en la naturaleza que los algoritmos de aprendizaje supervisado estudiados previamente. El campo del aprendizaje por refuerzo estudia los **algoritmos que son capaces de aprender de su entorno**.

## Diferencias con aprendizaje supervisado

La **principal diferencia** de los algoritmos de aprendizaje por refuerzo respecto de los algoritmos supervisados y no supervisados, es que **reciben información del entorno acerca de lo que es apropiado.** El aprendizaje por refuerzo se estudia en diversas disciplinas como la teoría de juegos, la teoría de control o la simulación. En estos algoritmos las recompensas vienen con retraso (ganar un juego se premia al final), mientras que en el aprendizaje supervisado se optimiza una acción-efecto concreta, es decir, no se tienen en cuenta la serie de acciones futuras. En el aprendizaje por refuerzo el número de combinaciones que un agente puede llevar a cabo para conseguir el objetivo es muy grande.

## Diferencias con aprendizaje no supervisado

En aprendizaje por refuerzo existe una **relación entre la entrada y salida** que no está presente en el aprendizaje no supervisado. En el aprendizaje no supervisado el objetivo es encontrar los patrones ocultos en lugar del mapeo acción-resultado. Por ejemplo, si el caso de uso es sugerir nuevas noticias a una persona: un modelo no supervisado tendrá en cuenta artículos similares a los que ha visto la persona y le serán sugeridos, mientras que un modelo de aprendizaje por refuerzo sugiere continuamente nuevos artículos para construir un «grafo de conocimiento» de los artículos que le gustan a una persona.

Para simular el aprendizaje automático en algoritmos, es necesario realizar algunas suposiciones, las cuales permiten tener un sistema más flexible capaz de una mayor generalización. En general, lo habitual es suponer que los agentes que aprenden del

entorno siguen un **proceso de decisión de Markov** (en inglés, Markov Decision Process, MDP). Básicamente, esta situación se pude definir de la siguiente forma:

- El agente puede percibir un conjunto finito *(S)* de estados diferentes en su entorno y dispone de un conjunto finito *(A)* de acciones para interactuar con el entorno.
- El tiempo avanza de forma discreta en cada instancia de tiempo *t,* el agente percibe un estado concreto *St*, selecciona una posible acción, *at* y la ejecuta, lo cual da lugar a un nuevo estado que se define como: $St+1 = at\ (St)$.
- El entorno responde a cada de una las acciones del agente por medio de un castigo o recompensa, que se puede denotar por $r(St, at)$, y que por medio del uso de un número que cuanto mayor es, indica que mayor será el beneficio.

Una cuestión importante de este algoritmo es que cumple la **propiedad de Markov.** Esto quiere decir que tanto la recompensa como el estado siguiente dependen únicamente del estado actual y de la acción tomada.

La finalidad de estos algoritmos es que el agente se adelante a las consecuencias de las acciones tomadas y sea capaz de identificar los estados que le llevan a conseguir una mayor eficacia y mayores recompensas.

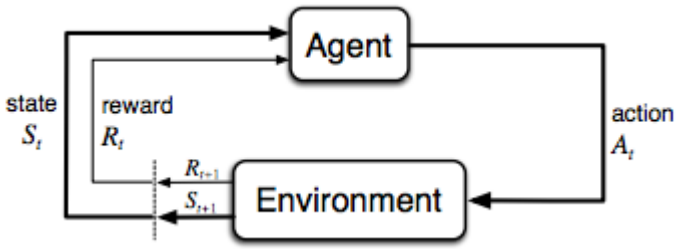



Figura 1. Esquema de funcionamiento de interacción de un agente con su entorno en un proceso de aprendizaje por refuerzo. Fuente: Sutton y Barto, 1998.

El **objetivo del aprendizaje por refuerzo** es establecer aquellas acciones que deben ser elegidas en los diferentes estados con el objetivo de maximizar la recompensa. Es decir, se busca que el agente aprenda una política que consiste en la mejor decisión a llevar a cabo en cada uno de los estados.

Hay situaciones en las cuales el agente puede observar el entorno por completo y son definidos como **plena observabilidad** y en otras se trata de **observabilidad parcial**. También situaciones con restricciones sobre las acciones que puede llevar a cabo el agente.

El aprendizaje automático es un área que ha sido aplicada con éxito a problemas de control de robots, aprendizaje de juego como el *backgammon* y las damas.

El **aprendizaje por refuerzo** estudia los algoritmos que son capaces de aprender de su entorno. En estas situaciones el agente que interactúa con el entorno puede tener plena observabilidad o bien observabilidad parcial.

## 12.3. Algoritmos de aprendizaje por refuerzo

Existen varios algoritmos o formas de implementar los conceptos de aprendizaje por refuerzo. Antes de entrar en detalle en los algoritmos vamos a hacer una definición formal de un proceso de decisión de Markov, donde tenemos los siguientes elementos:

- Conjunto de **estados:** *S.*
- Conjunto de **acciones:** *A.*
- Función de **transición:** $T: S\ x\ A\ \rightarrow S$.
- Función de **recompensa:** $R: S\ x\ A\ \rightarrow \mathbb{R}$

Un proceso de decisión de Markov (**MDP**) se define: $\langle S, A, T(s,a), R(s,a)\rangle$

Donde tenemos una **política**: $\pi: S\ \rightarrow A$ una función de **valor**:

$$V^{\pi}(s_t) = r_{t+1} + \gamma r_{t+2} + \gamma r_{t+3} + \cdots += \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+1}$$

Donde $V^{\pi}(s_t)$ es el valor acumulado que se consigue al seguir la política $\pi$ a partir del estado $S_t$; $\gamma$ es un **factor de descuento** ($0 \leq \gamma \leq 1$)

La función de valor se puede definir de forma recursiva utilizando la **ecuación de Bellman**:

$$V^{\pi}(S) = R\big(s, \pi(s)\big) + \gamma V^{\pi}(T(s, \pi(s)))$$

$R\big(s, \pi(s)\big)$ es la recompensa inmediata y $\gamma V^{\pi}(T(s, \pi(s)))$ el valor del siguiente estado.

El objetivo del agente es aprender la política óptima $\pi^*$:

$$\pi^*(s) = \text{argmax}\,[R(s, a) + \gamma V^*(T(s, a))]$$

Donde se busca la máxima ganancia esperada a partir de *s,* ejecutando la acción *a.*

## Fuerza bruta

Se trata de los algoritmos conceptualmente más sencillos de implementar. El tipo de algoritmos basados en fuerza bruta conlleva las siguientes fases:

1. Para cada acción posible, muestrear los resultados.
2. Elegir la acción con el mayor retorno esperado.

El **problema de este método** es que el número de políticas suele ser extremadamente grande, o incluso infinito. Además, la varianza de los rendimientos puede ser muy grande, lo cual hace necesario un gran número de muestras para estimar con más precisión el retorno de las acciones.

## *Q-Learning*

Se trata de un algoritmo de aprendizaje por refuerzo clásico inventado hace más de 25 años, en el que **el agente aprende a asignar valores de bondad a los pares**

(estado, acción). Es uno de los métodos más populares por su efectividad y por las posibilidades que ofrece para combinarlo con otras técnicas, como redes de neuronas o *deep learning.*

Si un agente está en un determinado estado y toma una acción, estamos interesados en conocer el resultado de esa acción, pero también en las recompensas futuras que se pueden obtener por pasar a otros estados. Es decir, deberemos de ser capaces de evaluar no solamente la recompensa actual sino también la recompensa futura de las posibles acciones posteriores.

En el algoritmo *Q-learning* el valor *Q* contiene la suma de todas las posibles recompensas futuras. El problema es que este valor puede ser infinito en el caso de que no haya un estado terminal que alcanzar. Además, es necesario establecer diferentes ponderaciones a las recompensas más recientes frente a las más lejanas. Para este último propósito se utiliza lo que se conoce como refuerzo acumulado con descuento, donde las recompensas futuras están ponderadas por un valor entre 0 y 1.

El **reto es en las primeras interacciones del agente con el entorno,** momento en el cual no se tiene la información necesaria para calcular el valor *Q*. Por tanto, se utiliza:

- Si una acción en un estado determinado es la causante de un resultado no deseado, se utiliza esta situación para no utilizar esta acción en ese estado en el futuro. De forma contraria, si una acción causa un resultado deseado, hay que aprender a aplicar esa acción en ese estado.

- Si todas las acciones que se pueden realizar desde un determinado estado dan un resultado negativo, se aprende este patrón para no tomar acciones desde otros estados que lleven a este. Por otro lado, si cualquier acción en un estado determinado proporciona un resultado positivo, es necesario aprender que se debe buscar ese estado. De esta forma se propaga la recompensa de un par (estado, acción) a los pares de los estados adyacentes.

## Algoritmo

Inicializar $Q(s,a)$ al azar.

**Repetir** (para cada episodio)

- Inicializar *s.*
- Repetir (para cada paso del episodio):
  - Elegir *a* en *s* según una política basada en *Q.*
  - Ejecutar la acción *a,* observar *r, s'.*
  - $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
  - $s \longleftarrow s'$

**Hasta que** *s* sea terminal.

Definimos $\pi(s) = argmax_a[Q(s,a)]$.

Los **episodios** incluyen un estado, la acción realizada y la recompensa recibida. El algoritmo *Q-learning* va iterando para ir rellenado todos los efectos posibles de las acciones en el concepto de experiencia.

Todo lo que necesita este algoritmo para poder ser entrenado en memoria es una **tabla para almacenar las recompensas para los estados y las acciones.** La tabla contiene la mejor estimación de cada recompensa, al principio será una estimación muy mala, pero a medida que el algoritmo aprende se irá volviendo más y más precisa.

El algoritmo necesita dos parámetros que debemos ajustar en función del problema que estamos resolviendo:

- **Velocidad de aprendizaje (*learning rate):*** es un valor entre 0 y 1 que indica cuánto se puede aprender en cada episodio. En el caso de cero indica que no se aprende nada de ese episodio y en el caso de uno establece que se borra lo que se sabía y se confía en el nuevo episodio.
- **Factor de descuento *(discount rate):*** también es un valor entre 0 y 1 que indica cómo de importante es el largo plazo respecto del corto. Un valor de 0 significa que solo son importantes los refuerzos inmediatos, mientras que un valor de 1 implica que solo son importantes los refuerzos a largo plazo.

En ambos parámetros es interesante **moverse fuera de los extremos,** pues en este caso proporcionan poca utilidad. La velocidad de aprendizaje se puede ir ajustando en función de la incertidumbre respecto de los estados siguientes. Por otro lado, el factor de descuento establece el balance entre el refuerzo inmediato y a largo plazo.

## 12.4. Referencias bibliográficas

Sutton, R. S. y Barto, A. (1998). *Reinforcement Learning: An Introduction.* MIT Press.

# Lo + recomendado

## No dejes de leer

**Introducción al aprendizaje con refuerzo y *OpenA*I**

Francis, J. (13 de julio de 2017). Introduction to reinforcement learning and OpenAI Gym.

Blog post introductorio del aprendizaje por refuerzo y ejemplos de uso de la librería Gym de OpenAI para desarrollar y probar algoritmos de aprendizaje por refuerzo.

Accede al post a través del aula virtual o desde la siguiente dirección web:
https://www.oreilly.com/learning/introduction-to-reinforcement-learning-and-openai-gym

**Taller de aprendizaje por refuerzo**

University of California, Berkeley. (s. f.). Reinforcement Learning.

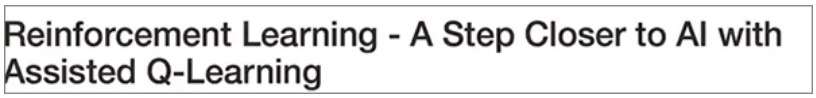
Ejemplo de un modelo de aprendizaje por refuerzo para implementar la lógica del juego de pacman utilizando Python y estrategias de *Q-learning*.

Accede a la página a través del aula virtual o desde la siguiente dirección web:
https://inst.eecs.berkeley.edu/~cs188/sp12/projects/reinforcement/reinforcement.html

# No dejes de ver

**Aprendizaje por refuerzo**

Vídeo demostrativo del uso del aprendizaje por refuerzo *(Q-learning)* con Python para encontrar el camino más corto entre dos puntos.




Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

https://www.youtube.com/watch?v=nSxaG_Kjw_w

**Tutorial de *Reinforcement Learning***

Vídeo tutorial de Microsoft Research sobre *deep learning*. Proporciona una descripción de los procesos de decisión de Markov (MDP) incluyendo los métodos de programación dinámica de Monte Carlo. Se centra en la combinación de estos métodos con aproximaciones paramétricas para buscar buenas soluciones a los problemas que de otra forma serían muy largos de ser llevados a cabo.


Accede al vídeo al través del aula virtual o desde la siguiente dirección web:

https://youtu.be/ggqnxyjaKe4

**Introduction to Reinforcement Learning.**

Charla de David Silver sobre aprendizaje por refuerzo con bastantes ejemplos intuitivos y la aplicación en los juegos.


Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

https://www.youtube.com/watch?v=2pWv7GOvufo

# + Información

## A fondo

***Practical Reinforcement Learning***

Farrukh, S. M. (2017). *Practical Reinforcement Learning.* Packt.

Este libro te va a ayudar a dominar diferentes técnicas de aprendizaje de refuerzo y su implementación práctica usando OpenAI Gym, Python y Java.

Accede al libro a través del aula virtual o desde la siguiente dirección web:
https://www.amazon.es/Practical-Reinforcement-Learning-Farrukh-Akhtar/dp/1787128725/ref=sr_1_3?ie=UTF8&qid=1519947114&sr=8-3

## Bibliografía

Sutton, R. S. y Barto, A. (1998). *Reinforcement Learning: An Introduction.* MIT Press.

# Test

**1.** El aprendizaje por refuerzo:

A. Es un tipo de aprendizaje supervisado.

B. Es un tipo de aprendizaje no supervisado.

C. Ninguna de las anteriores es correcta.

**2.** El aprendizaje por refuerzo:

A. Va aprendiendo del *feedback* obtenido por cada acción.

B. Se utiliza en las situaciones en las que un agente puede observar el entorno.

C. Comprende los algoritmos que son capaces de aprender del entorno.

**3.** En un proceso de decisión de Markov:

A. Solo se tienen en cuenta los estados posteriores.

B. Solo se tienen en cuenta el estado previo.

C. Se tienen en cuenta el estado previo y los siguientes.

**4.** En el algoritmo *Q-learning:*

A. Si una acción en un estado es la causante de un resultado no deseado, esta acción no se usará en el futuro.

B. Si una acción en un estado es la causante de un resultado deseado, se aplicará esa acción es ese estado.

C. La mejora del algoritmo *Q-learning* es porque no es necesario utilizar el estado.

**5.** Los parámetros de *learning rate* y *discount rate* del algoritmo *Q-learning*:

A. Es mejor que estén cercanos a 1.

B. Es mejor que estén cercanos a 0.

C. Idealmente deberían estar alejados de los extremos.

**6.** La ecuación de Bellman:

A. Actualmente está en desuso.

B. Se utiliza como punto inicial del aprendizaje.

C. Permite definir el valor de forma recursiva.

**7.** El algoritmo de aprendizaje por refuerzo de fuerza bruta:

A. Es una forma óptima de solucionar el problema.

B. Explora todas las posibles combinaciones.

C. Es un método costoso.

**8.** El algoritmo *Q-learning:*

A. Únicamente tiene en cuenta las recompensas a largo plazo.

B. El valor *Q* contiene la suma de todas las posibles recompensas futuras.

C. Tiene en cuenta tanto las recompensas a largo plazo como a corto.

**9.** La velocidad de aprendizaje del algoritmo *Q-learning:*

A. Es un valor entre 0 y 1 que indica cuanto se puede aprender en cada episodio.

B. En el caso de 0 no se aprende nada.

C. En el caso de 1 se borra lo anterior y se aprende de nuevo.

**10.** El factor de descuento del algoritmo *Q-learning:*

A. Es un valor entre 0 y 100 que indica la importancia del largo plazo respecto del corto plazo.

B. Es un valor entre 0 y 1 que indica la importancia del largo plazo respecto del corto.

C. Es un valor entre 0 y 1 que indica la importancia de las instancias.