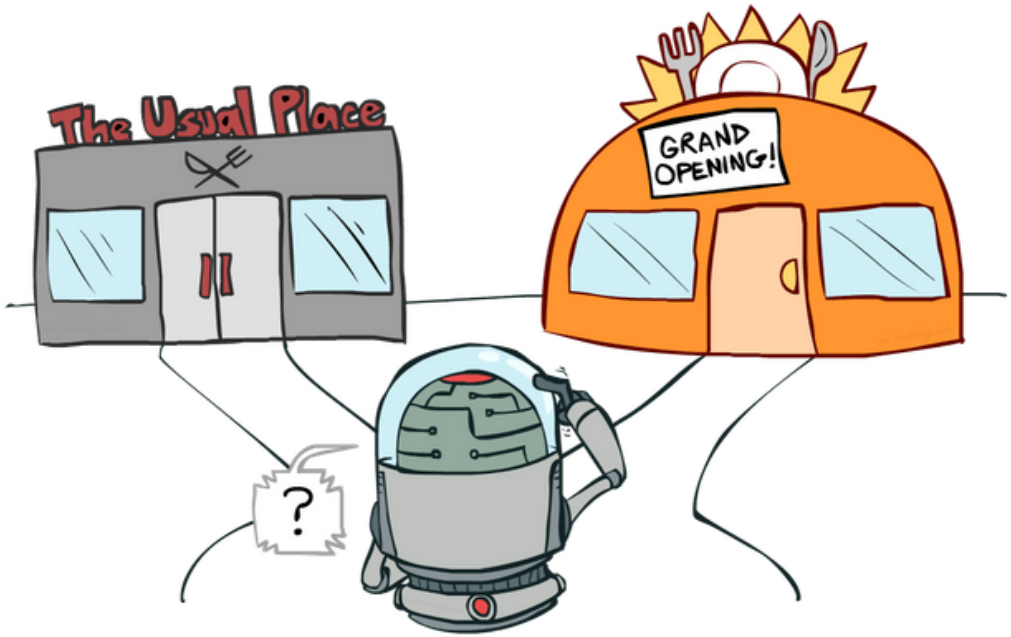
Introducción a los procesos de los procesos de decisión de Markov (PDM)

Cristian Felipe Correa

Departamento de Matemáticas
Cinvestav
Ciudad de México

Introducción

Esta clase de problemas se pueden modelar, desde una perspectiva de teoría de toma de decisiones, como **Procesos de Decisión de Markov**.



Preliminares

- **Variable aleatoria:** Es una función real definida a un espacio de probabilidad que asignar un valor, usualmente numérico, al resultado de un experimento aleatorio.
- **Proceso estocástico:** Es una familia de variables aleatorias que depende de un parámetro.

Preliminares

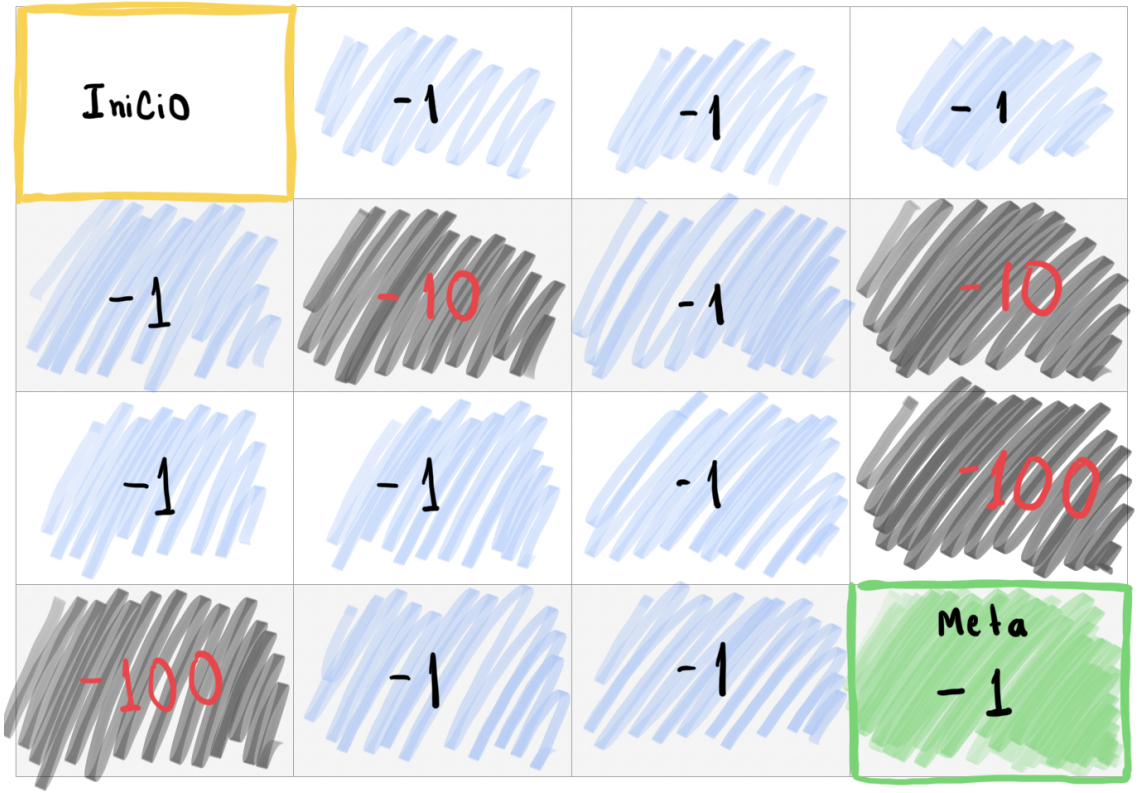
- **Distribución de Probabilidad:** Es una función que asigna a cada suceso definido sobre la variable aleatoria, la probabilidad que dicho suceso ocurra.
- **Propiedad de Markov**: Es una propiedad de ciertos procesos estocásticos por la cual la distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor presente.
- El proceso estocástico $\{s_t, t = 0, 1, 2, 3...\}$ cumple la propiedad de Markov si

$$P(s_{t+1}|s_0, s_1, ..s_t) = P(s_{t+1}|s_t)$$

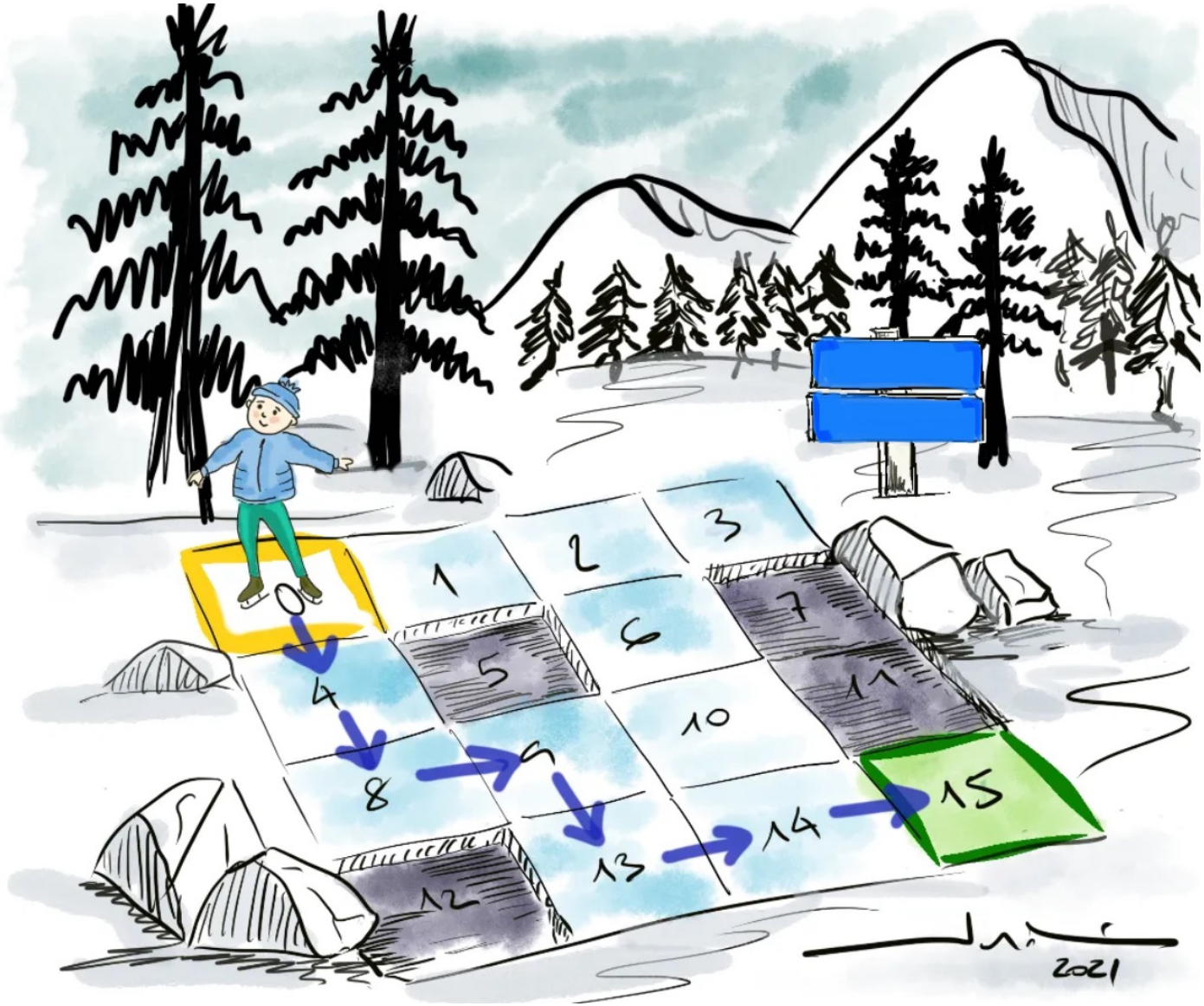
Ejemplo

Ejemplo

- $S = \{0, 1, 2, 3, 4, 5, 6..., 15\}$
- $A = \{\leftarrow, \uparrow, \rightarrow, \downarrow\}$
  - $A(0) = \{\downarrow, \rightarrow\}$
  - $A(12) = \{\rightarrow, \uparrow\}$
- Función de penalización, recompensa o costo.



Ejemplo



Procesos de control de Markov

Definición : Un Modelo de control de Markov, estacionario, a tiempo discreto, consiste en una quíntupla

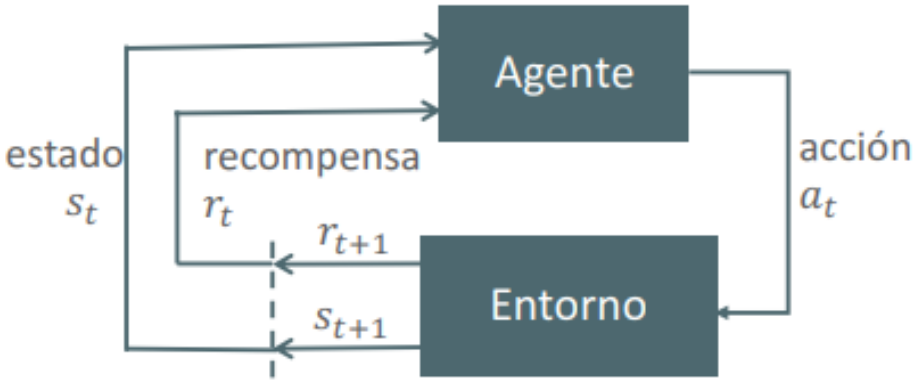
$$\{S, A, \{A(s) : s \in S\}, \mathbf{Q}, c\} \tag{1}$$

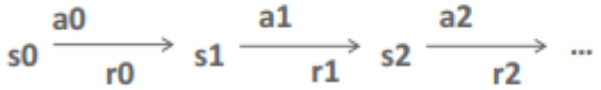
- S conjunto no vacío, llamado espacio de estados.
- A conjunto no vacío, llamado espacio de acciones.
- $\{A(s) : s \in S\}$ es un familia de subconjuntos medibles, no vacíos de A.
- El conjunto de pares estado-acción admisible es,

$$\mathbb{K} = \{(s, a) : s \in S, a \in A(s)\}, \tag{2}$$

  transición de los estados del sistema.
- $c : \mathbb{K} \to \mathbb{R}$ es una función que corresponde al costo por etapa.

Dinámica del sistema





Dinámica del sistema

Un sistema dinámico de toma de decisiones, que depende del tiempo con condición inicial $s_0 = s$ puede ser expresado de la siguientes formas:

- $s_{t+1} = F(s_t, a_t)$
- $s_{t+1} = F(s_t, a_t, \xi_t)$
- $\mathbf{Q}(B|s_t, a_t) = Pr(s_{t+1} \in B|s_t, a_t)$

donde $\{\xi_t, t = 0, 1, 2..\}$ son una sucesión de variables conocidas como **perturbaciones**

Políticas

- **Definición:** Una política es una sucesión de acciones o decisiones admisibles.

$$\{\pi_t, t = 0, 1, 2, ...\}$$

  - **Caso determinista:** La política es una sucesión de funciones medibles, tal que, $\pi_t : S \rightarrow A$, es decir, en cada instante de tiempo t, $\pi_t(s_t) = a_t$.
  - **Caso Aleatorio:** La política es una sucesión de distribuciones de probabilidad condicionadas, tal que, $a_t \sim \pi_t(\cdot|s_t)$, es decir, $\pi_t(B|s_t) = Pr(a_t \in B)$ para $B \in A(s_t)$

Criterio de Rendimiento

Consideremos un modelo de control de Markov fijo y un conjunto de políticas Π. Definimos para cada $s \in S$ y $\pi \in \Pi$

$$V(\pi, s) := E_s^{\pi} \left[\sum_{t=0}^{N-1} c(s_t, a_t) + c_N(x_N) \right] \tag{3}$$

$$V(\pi, s) := E_s^{\pi} \left[\sum_{t=0}^{\infty} \alpha^t c(s_t, a_t) \right] \tag{4}$$

con $\alpha \in (0, 1)$.

Criterio de Rendimiento

Definición: Para cada $s \in S$ definimos

$$V^*(s) = \inf_{\pi \in \Pi} V^\pi(s) \tag{5}$$

V^* se llama la función de valor óptimo.

$$V(\pi^*, s) = V^*(s) = \inf_{\pi \in \Pi} V^\pi(s) \tag{6}$$

π^* se llama la política óptima.

Ecuaciones de Programación dinámica

Teorema: Sean $V_0, V_1, ..., V_N$ funciones sobre S definidas por

$$V_N(s) = C_N(s) \tag{7}$$

y para cada $t = 0, 1, ..., N-1$

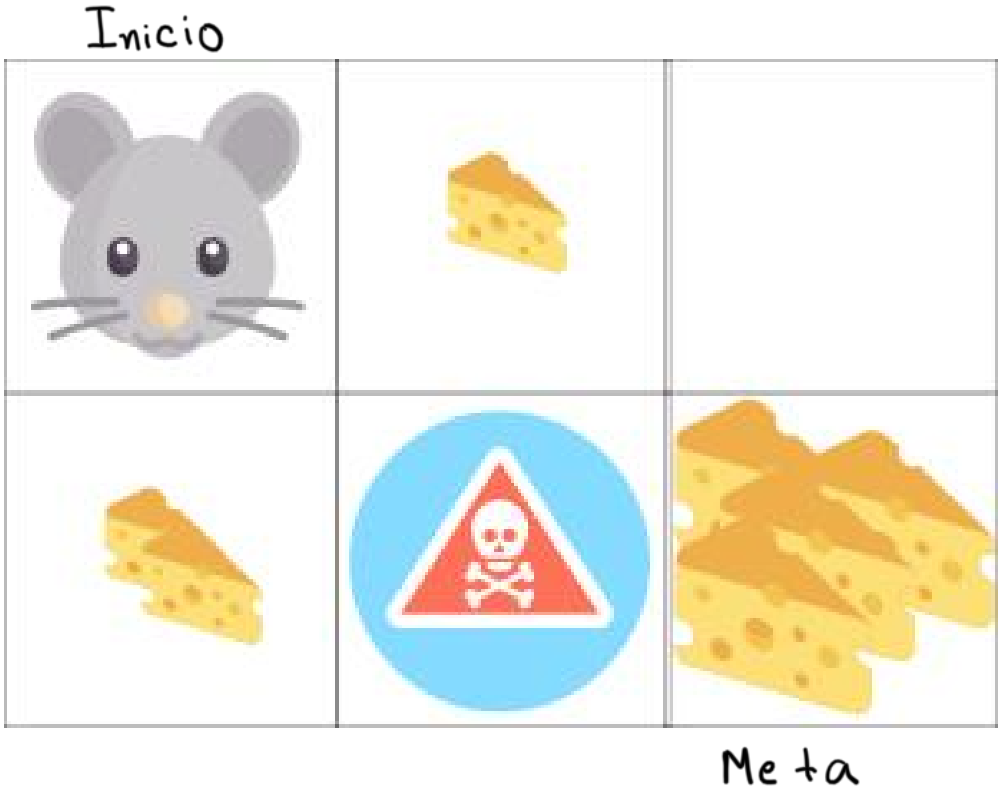
$$V_t(s) = \min_{a \in A(s)} \left[c(s,a) + \int_S V_{t+1}(y) \mathbf{Q}(dy \mid s, a) \right] \tag{8}$$

Supongamos que estas funciones son medibles y que para cada $t = 0, 1, ..., N-1$ existe una función π_t con $\pi_t(s) \in A(s)$, tal que

$$V_t(s) = \left[c(x, \pi_t(s)) + \int_S V_{t+1}(y) \mathbf{Q}(dy \mid s, \pi_t(s)) \right] \tag{9}$$

Entonces, la política determinista $\pi^* = (\pi_0, \pi_1, ..., \pi_{N-1})$ es óptima y la función de valor óptimo V^* es V_0.

Aprendizaje Reforzado



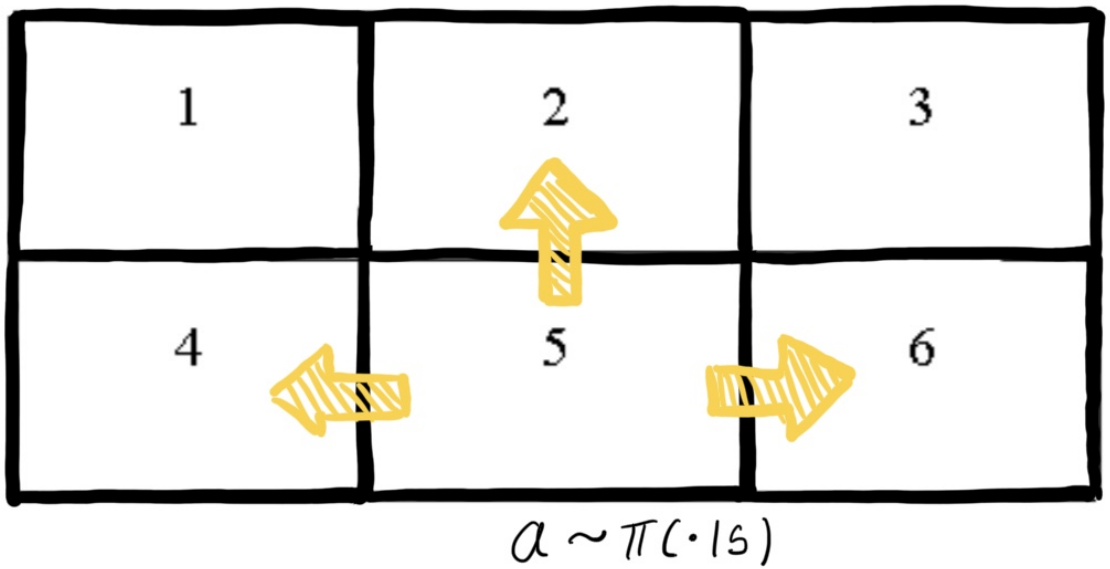
Aprendizaje Reforzado

Aprendizaje reforzado basado en modelos:

- Asumimos que todos los elementos del PDM

$$\{S, A, \{A(s) : s \in S\}, \mathbf{Q}, c\} \tag{10}$$

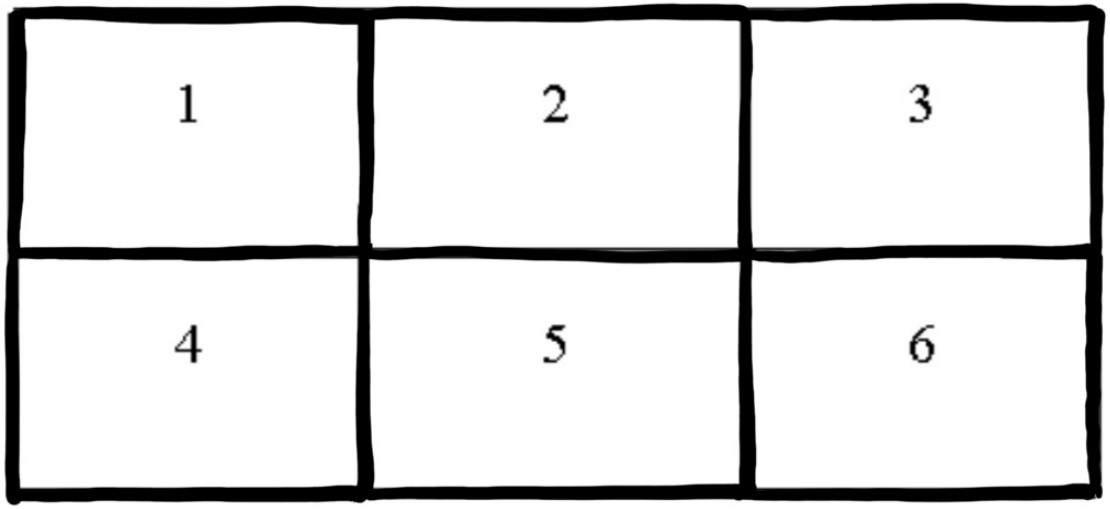
son **conocidos**.



Tipos de Aprendizaje Reforzado

Aprendizaje reforzado libre de modelos

- Los elementos del PDM son parcialmente conocidos, esto quiere decir que $P(\cdot \mid s, a)$ y $r(s, a)$ no son conocidos.
- En este caso el agente aprende por medio de un proceso que se conoce como la **prueba y el error.**



Criterios de Rendimiento

Criterios de rendimiento:, Definimos para cada $s \in S$, $a \in A(s)$ y $\pi \in \Pi$

$$Q^{\pi}(s,a) = E^{\pi}(\sum_{t=0}^{N} \alpha^{t} r(s_t, a_t) | s_0 = s, a_0 = a) \tag{11}$$

con $\alpha \in (0,1)$. Si $a_0 \sim \pi(\cdot|s)$, entonces

$$V(\pi, s) = \sum_{a} \pi(a|s) Q^{\pi}(s,a) \tag{12}$$

Criterio de Rendimiento

Definición: Para cada $s \in S$ y $a \in A(s)$ definimos

$$Q^*(s,a) = \max_{\pi \in \Pi} Q^\pi(s,a) \tag{13}$$

Q^* se llama la función de acción-estado óptimo.

$$Q(\pi^*, s, a) = Q^*(s,a) = \max_{\pi \in \Pi} Q^\pi(s,a) \tag{14}$$

π^* se llama la política óptima.

Ecuación de Bellman

Para $\pi \in \Pi$ y para todo $s \in S$ tendremos en este caso,

$$Q^*(s,a) = r(s,a) + \alpha \sum_{s' \in S} \mathbf{Q}(s' \mid s,a) \max_{a'} Q^*(s',a') \tag{15}$$

Considerando que: $V^*(s) = \max_a Q^*(s,a)$

Tipos de algoritmos

- Tipos de algoritmos de aprendizaje por refuerzo:
  - **Basado en la función de valor**(value-based): El agente aprende la función óptima de acción-valor:

$$Q^*(s,a) = r(s,a) + \alpha \sum_{s' \in S} \mathbf{Q}(s' \mid s,a) \max_{a'} Q^*(s',a') \tag{16}$$

  -

$$Q^*(s,a) = E^{\pi}(r(s,a) + \alpha \max_{a'} Q^*(s',a')|s,a) \tag{17}$$

    sin necesidad de calcular o buscar directamente la política óptima.

Algoritmo

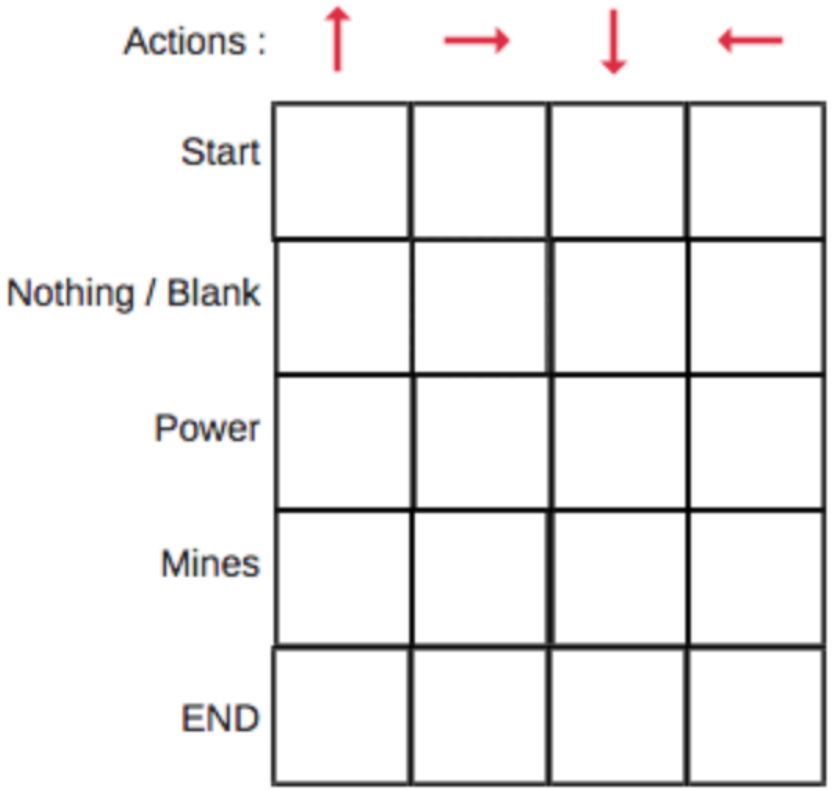
- El algoritmo de Q-learning, consiste en hacer una aproximación de la función Q^* para cada par estado-acción, por medio de la siguiente ecuación.

$$Q(s,a) \leftarrow Q(s,a) + \gamma(r(s,a) + \alpha \max_{a' \in A} Q(s',a') - Q(s,a)) \tag{18}$$

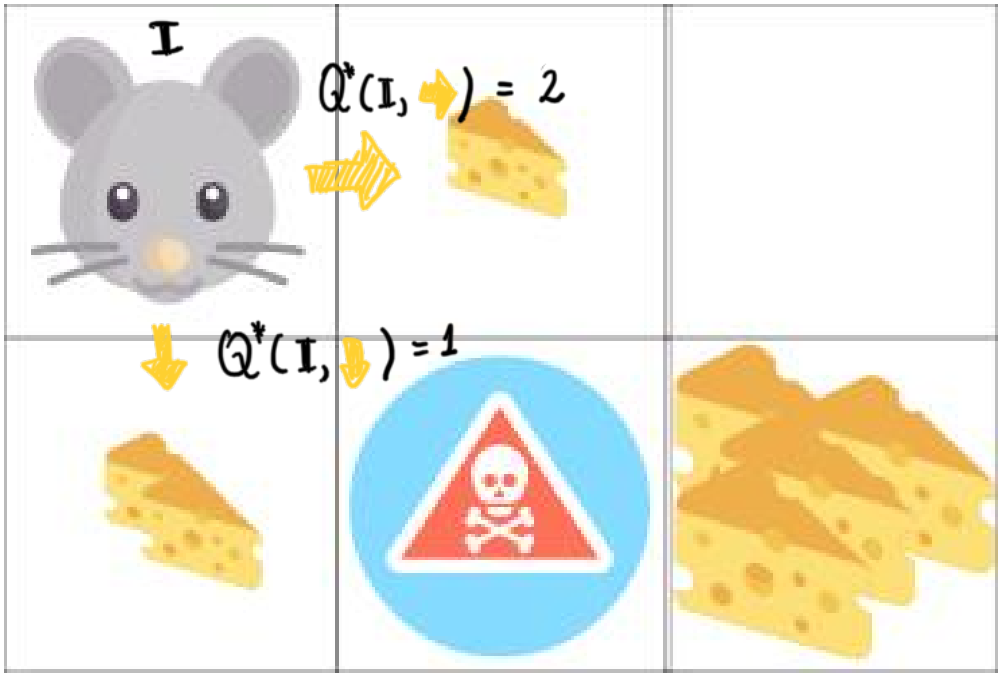
- Donde $\gamma \in (0,1)$ corresponde al ratio de aprendizaje y $\alpha \in (0,1)$ el factor de descuento.

Q- Tabla

- Estas aproximaciones de las funciones Q^* serán almacenadas en una tabla (matriz) llamada Q-tabla.



Q-Learning



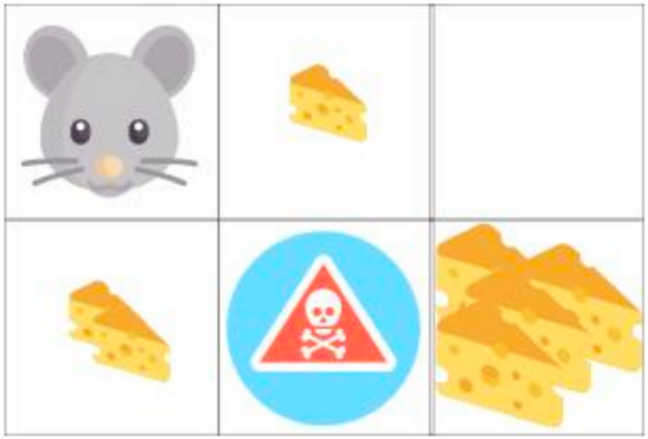
Objetivo

- El objetivo principal es que el agente aprenda a tomar para cada par estado-acción el mayor valor de las funciones Q^*
- Para realizar lo anterior el agente debe aprender a decidir si explorar o explotar las acciones. Esto es:
  - Seleccionar una acción con el valor Q^* más alto para ese estado (explotación).
  - Seleccionar una acción al azar (exploración)
- Esta selección el agente la hace con un algoritmo llamado ϵ-greedy.

ϵ-greedy

$$\pi^{\epsilon}(s) = \begin{cases} argmax_a Q(s,a) & probabilidad \quad 1-\epsilon \\ \\ Uniform(A(s)) & probabilidad \quad \epsilon \end{cases}$$

Ejemplo



$Q - Tabla$	$\leftarrow$	$\rightarrow$	$\uparrow$	$\downarrow$
Inicio	0	0	0	0
1-queso	0	0	0	0
Nada	0	0	0	0
2-quesos	0	0	0	0
Muerte	0	0	0	0
Muchos quesos	0	0	0	0

$Q - Tabla$	$\leftarrow$	$\rightarrow$	$\uparrow$	$\downarrow$
Inicio	0	0	0	0
1-queso	0	0	0	0
Nada	0	0	0	0
2-quesos	0	0	0	0
Muerte	0	0	0	0
Muchos quesos	0	0	0	0

Ejemplo

$$newQ(s,a) = OldQ(s,a) + \gamma(R(s,a) + \alpha \max_{a' \in A(s')} Q(s',a') - Q(s,a)) \tag{19}$$

$$\begin{aligned} newQ(I,D) &= Q(I,D) + \gamma((R(I,D) + \alpha \max_{a'} Q(1q,a') - Q(I,D))) \\ &= Q(I,D) + \gamma(1 + 0.9 \max(Q(1q,Iz), Q(1q,D), Q(1q,Ab)) - Q(I,D)) \\ &= 0 + 0.1(1 + 0.9 * 0 - 0) \\ &= 0.1 \end{aligned} \tag{20}$$

Ejemplo

$Q - Tabla$	$\leftarrow$	$\rightarrow$	$\uparrow$	$\downarrow$
Inicio	0	0,1	0	0
1-queso	0	0	0	0
Nada	0	0	0	0
2-quesos	0	0	0	0
Muerte	0	0	0	0
Muchos quesos	0	0	0	0

$$newQ(s,a) = OldQ(s,a) + \gamma(R(s,a) + \alpha \max_{a' \in A(s')} Q(s',a') - Q(s,a))$$

Algoritmo Q- Learning

Q-Learning(π)

1 $Q \leftarrow Q_0$ $\triangleright$ initialization, e.g., $Q_0 = 0$.
2 **for** $t \leftarrow 0$ **to** T **do**
3 $s \leftarrow$ SelectState()
4 **for** each step of epoch t **do**
5 $a \leftarrow$ SelectAction(π, s) $\triangleright$ policy π derived from Q, e.g., ϵ-greedy.
6 $r' \leftarrow$ Reward(s, a)
7 $s' \leftarrow$ NextState(s, a)
8 $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r' + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
9 $s \leftarrow s'$
10 **return** Q

Bibliografía

- Hernández-Lerma, O., Lasserre, J.B (1996) *Discrete-time Markov control processes. Basic optimality criteria.* Springer-Velrag, New York.
- Bertsekas, D., Shreve, S.E. (1978) *Optimal control : the discrete-time case.* Academic Press, New York.
- Haurie, A. ,Brawczyk, J.B., Zaccour, G (2012) *Games and dynamics games.* World Scientific, Singapore.
- Hernández-Lerma, O.(2005) *Control óptimo y juegos estocásticos.* EMALCA, CIMAT, Guanajuato, México.
- Mohri, M., Rostamizadeh, A., Talwalkar, A. (2012) *Foundations of Machine Learning.* MIT Press, Cambridge, Massachusetts.
- Sutton, R.S., Barto, A.G. (2018) *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, Massachusetts.