

Aprendizaje Automático
Segundo Cuatrimestre de 2015

Aprendizaje por Refuerzos

Gracias a Carlos “Greg” Diuk por los
materiales para las transparencias.

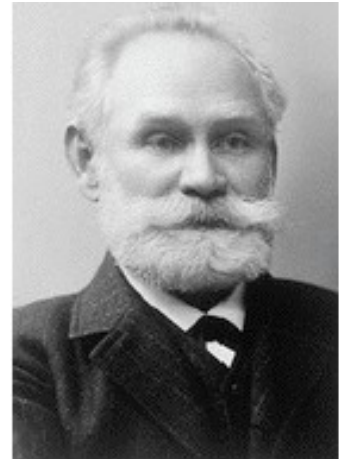


DEPARTAMENTO
DE COMPUTACION

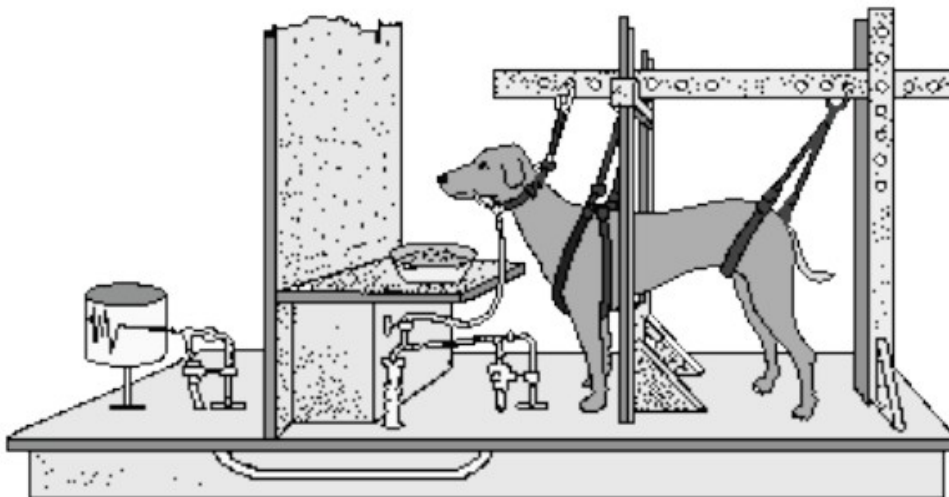
Facultad de Ciencias Exactas y Naturales - UBA

Aprendizaje por Condicionamiento

- Primeras teorías:
 - Condicionamiento clásico o Pavloviano.

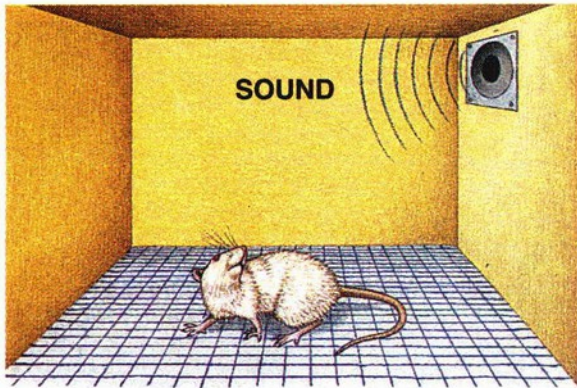


Ivan Pavlov
(1849-1936)

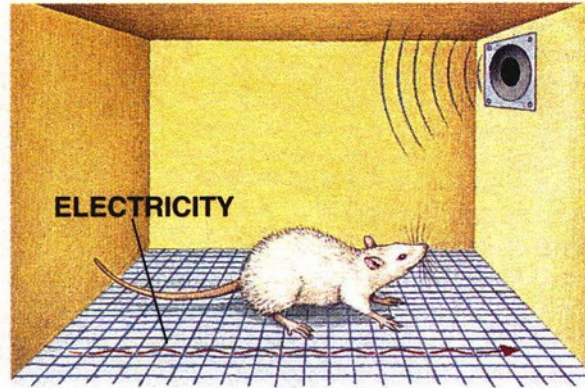


Aprendizaje por Condicionamiento

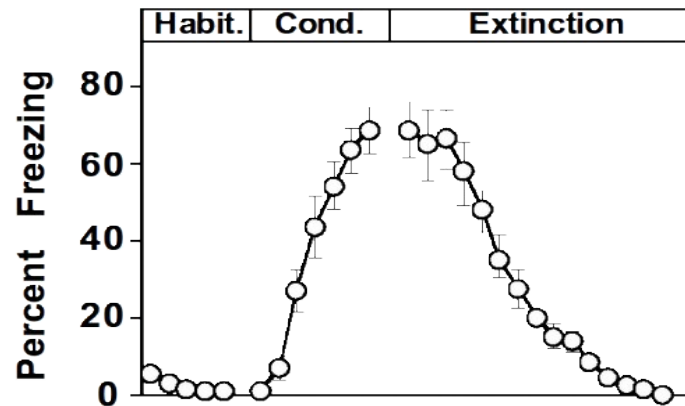
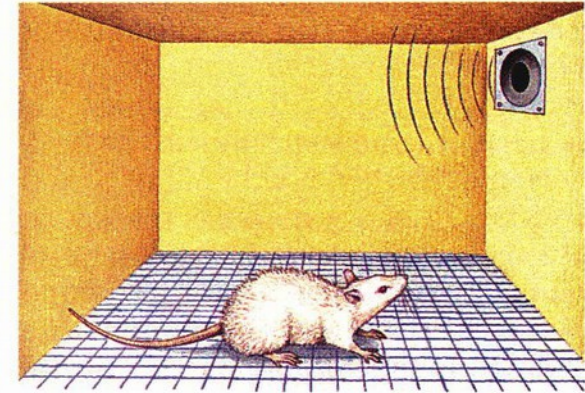
Habitación (tono)



Condicionamiento (tono+shock)



Extinción (tono)



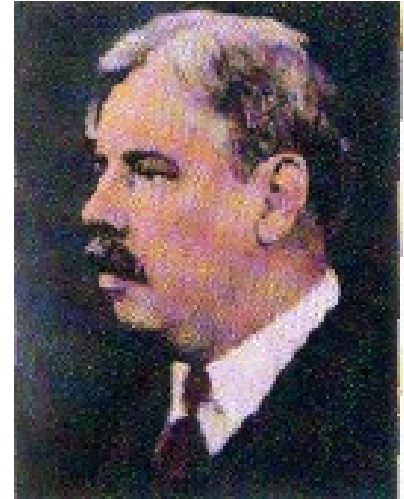
Crédito: Quirk lab, U de Puerto Rico

- Visión ultra-limitada del aprendizaje, pero funciona!
Demuestra que los animales pueden aprender relaciones arbitrarias entre estímulo → respuesta.

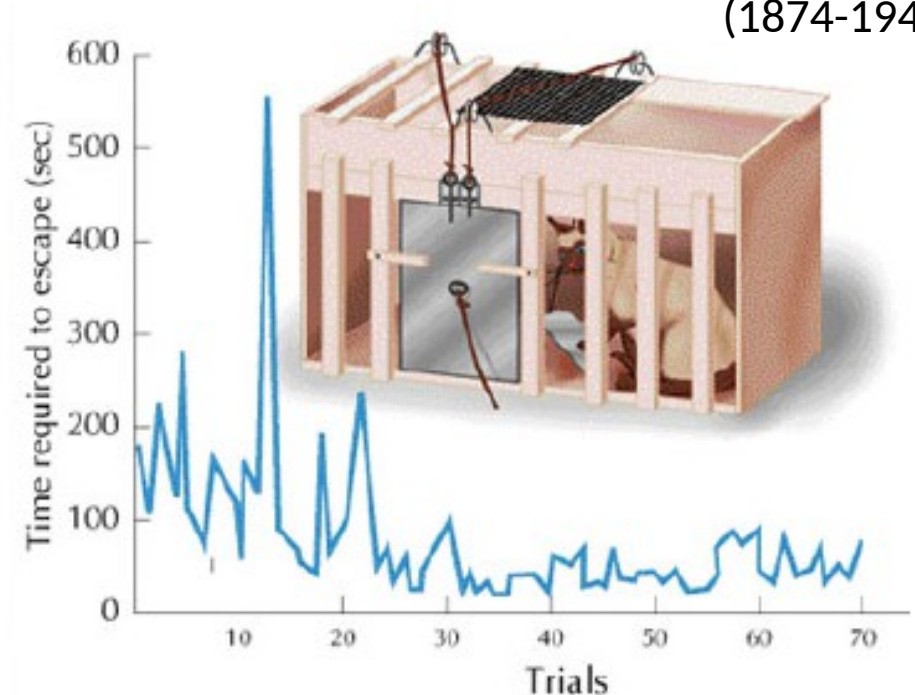
- Pero el perro de Pavlov **no tomó decisiones**.
- Hasta ahora no tuvo que hacer nada, no tomó decisiones. Sólo salivó porque la campana le recordó el churrasco.
- Agreguemos “**control**”: no solamente estímulo-respuesta.
- Las acciones que tomamos tienen consecuencias.

Condicionamiento Instrumental/Operacional

- Thorndike experimentó con gatos hambrientos tratando de escapar de una jaula.
- Midió el “tiempo hasta escapar” como métrica de aprendizaje.
- “Curva de aprendizaje”



Edward Thorndike
(1874-1949)



Condicionamiento Instrumental/Operacional

- La lección importante a extraer:

Los animales no sólo pueden aprender relaciones estímulo-respuesta arbitrarias, sino también comportamientos arbitrarios en base a dichos estímulos.



Crédito: Björn Brembs, FU Berlin

Rescorla & Wagner (1972)



- Aprendizaje guiado por errores: el cambio en el *valor* de una asociación es proporcional a la diferencia entre nuestra predicción y lo observado:

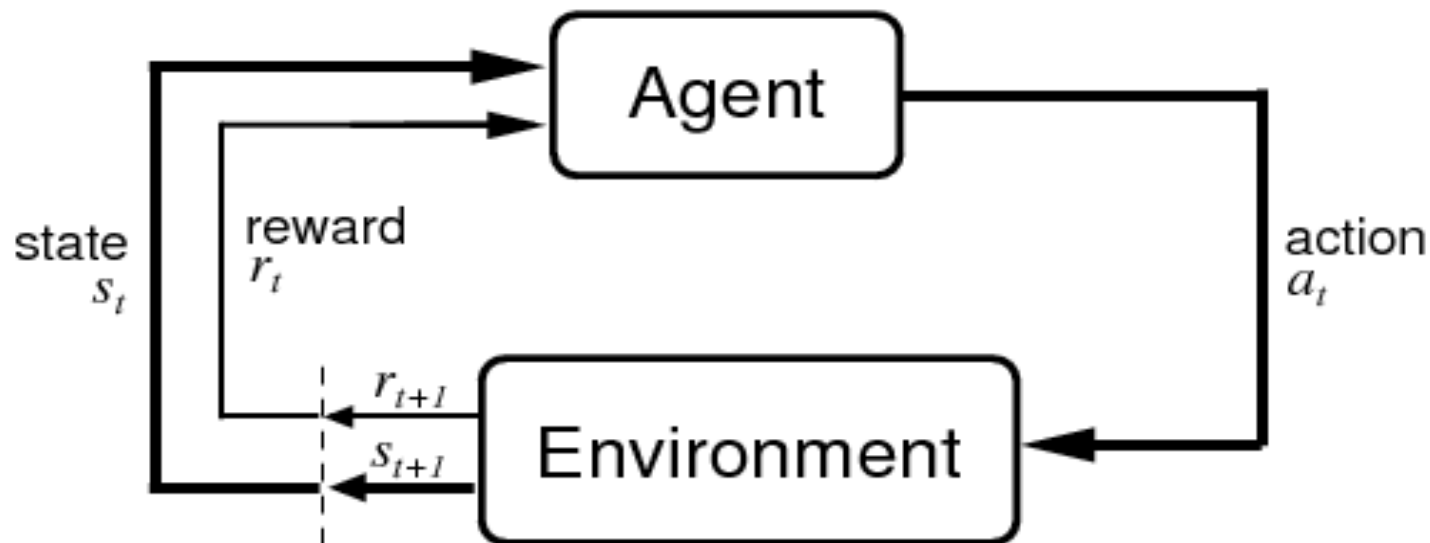
$$V_{\text{nuevo}} = V_{\text{viejo}} + \alpha \underbrace{(R - V_{\text{viejo}})}_{\text{error de predicción}}$$

tasa de aprendizaje

Aprendizaje por Refuerzos



- **Agente** (ej, un robot):
 - Tiene sensores para observar el **estado** de su entorno.
 - Puede realizar **acciones** para alterar ese estado.
 - Cada acción conlleva una **recompensa** numérica inmediata.



Crédito: Sutton y Barto

Aprendizaje por Refuerzos



- Tarea del agente: **aprender** una **estrategia o política de control** para elegir las acciones que maximicen las recompensas.

$$\pi : S \rightarrow A$$

- **Diferencias** con aprendizaje **supervisado**:
 - Las recompensas vienen con **demora** (ej: ganar un juego te premia al final). Un agente debe aprender a discernir cuáles acciones de una secuencia fueron las meritorias.
 - **Exploración**: la distribución de los datos de entrenamiento está determinada por las acciones elegidas por el agente.

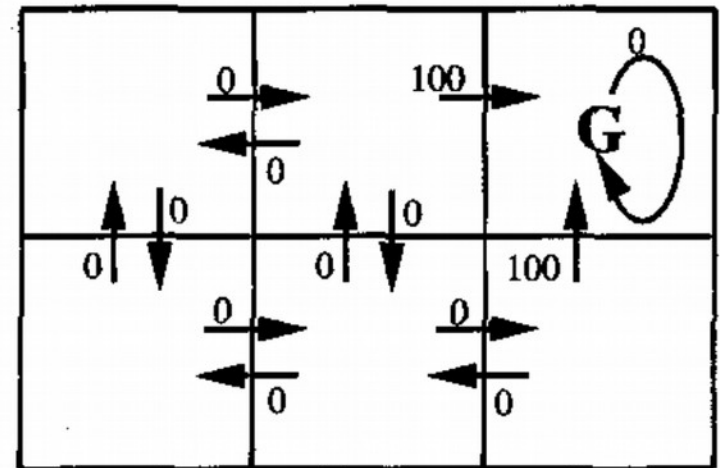
Formalismo estándar

Proceso de Decisión de Markov (MDP)

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

- Ejemplo: *Grid World*

6 estados, 4 acciones ($\leftarrow \uparrow \rightarrow \downarrow$)



Proceso de Decisión de Markov (MDP): $\langle S, A, T(s, a), R(s, a) \rangle$

Definimos:

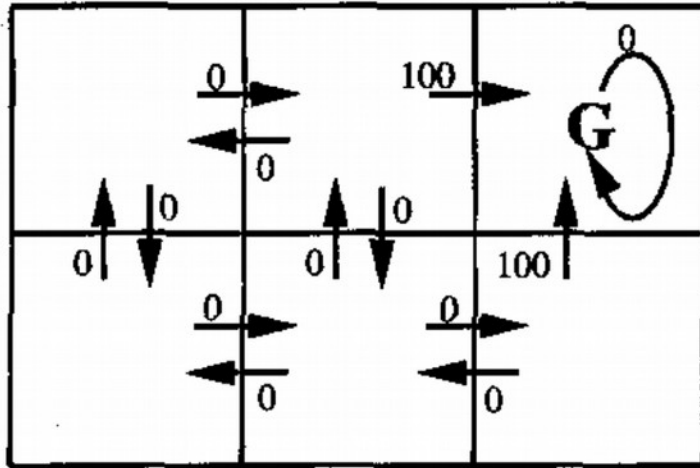
- **Política** $\pi : S \rightarrow A$

- Función de **valor**:

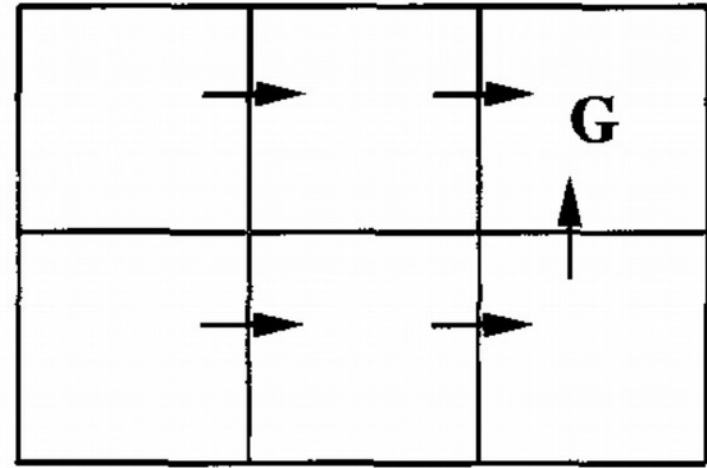
$$V^\pi(s_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots = \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}$$

- $V^\pi(s_t)$ es el valor acumulado que se consigue al seguir la política π a partir del estado s_t .
- γ se conoce como **factor de descuento** ($0 \leq \gamma < 1$).

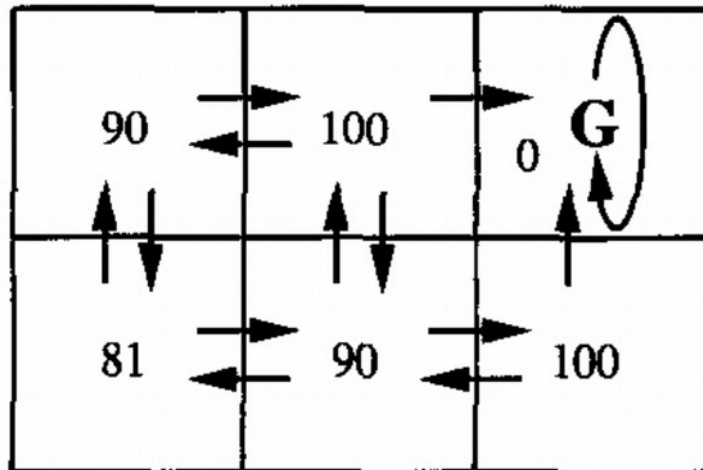
Función de recompensa R :



Política π :



Función de valor V^π :



- Función de **valor**:

$$V^\pi(s_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots = \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}$$

- Definición **recursiva** (ecuación de Bellman):

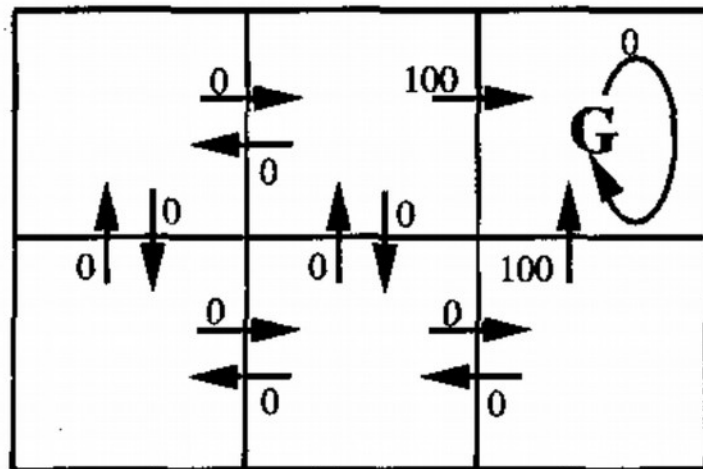
$$V^\pi(s) = \underbrace{R(s, \pi(s))}_{\text{recompensa inmediata}} + \gamma \underbrace{V^\pi(T(s, \pi(s)))}_{\text{valor del siguiente estado}}$$

- Objetivo del agente: aprender la política óptima π^*

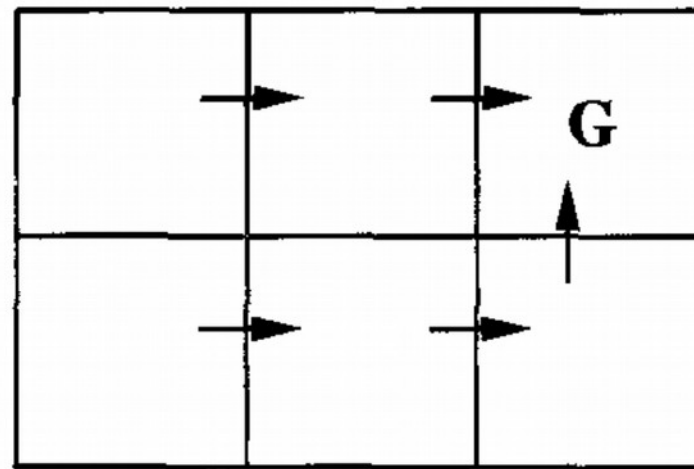
$$\pi^*(s) = \operatorname{argmax}_a \left[\underbrace{R(s, a) + \gamma V^*(T(s, a))}_{Q(s, a)} \right]$$

- $Q(s, a)$ es la máxima ganancia esperada a partir de s , ejecutando la acción a .

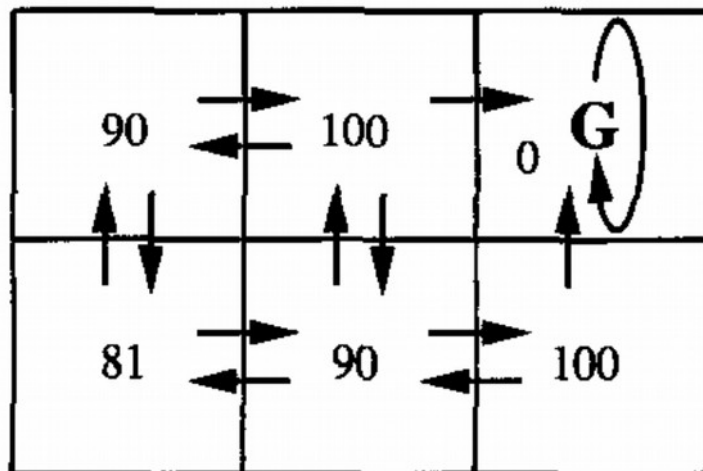
Función de recompensa R :



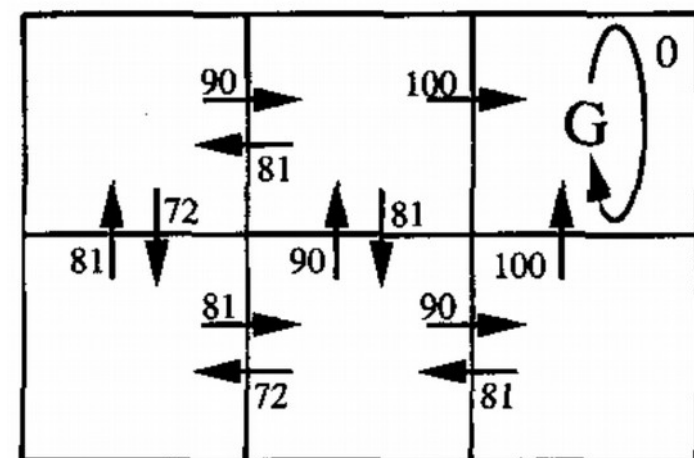
Política óptima π^* :



Función de valor V^* :



$Q(s, a)$:



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

$$Q(s, a) = R(s, a) + \gamma V^*(T(s, a))$$

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

Por lo tanto, sustituyendo:

$$V^*(s) = \max_a [R(s, a) + \gamma V^*(T(s, a))]$$

Idea para un **algoritmo iterativo de aprendizaje**:

$$V(s) \leftarrow \max_a [R(s, a) + \gamma V(T(s, a))]$$

Algoritmo 1: *Value Iteration*

$V(s) \leftarrow 0 \quad \forall s \in S$

$\Delta \leftarrow 0$

Repetir:

Para cada $s \in S$:

$v \leftarrow V(s)$

$V(s) \leftarrow \max_a [R(s, a) + \gamma V(T(s, a))]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

Fin para

Hasta que $\Delta < \text{threshold}$

Definimos $\pi(s) = \operatorname{argmax}_a [R(s, a) + \gamma V(T(s, a))]$

- VI en acción: <http://www.cs.ubc.ca/~poole/demos/mdp/vi.html>
- VI requiere **conocer el modelo**: S , A , R y T .

$$Q(s, a) = R(s, a) + \gamma V^*(T(s, a))$$

$$V^*(s) = \max_{a'} Q(s, a')$$

Por lo tanto, sustituyendo:

$$Q(s, a) = R(s, a) + \gamma \max_{a'} Q(T(s, a), a')$$

Idea para otro **algoritmo iterativo de aprendizaje**:

$$Q(s, a) \leftarrow R(s, a) + \gamma \max_{a'} Q(T(s, a), a')$$

Si incorporamos una **tasa de aprendizaje** α ($0 < \alpha \leq 1$):

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

Algoritmo 2: *Q-Learning*

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'

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

$$s \leftarrow s'$$

Hasta que s sea terminal

Definimos $\pi(s) = \operatorname{argmax}_a [Q(s, a)]$

- QL en acción: <http://www.cs.ubc.ca/~poole/demos/rl/q.html>
- QL **no** requiere **conocer el modelo**.
- Falta decidir cómo elegir la acción a en el estado s .

K-armed bandits



Cada máquina tiene su propia probabilidad de pagar.
¿Cómo hacemos para maximizar nuestra ganancia?

Dilema exploración-explotación: cuándo **explotar** lo que ya sabemos, cuándo seguir **explorando** mejores opciones.

Bandidos: Eligiendo un brazo

- Estrategia *ϵ -greedy*:
 - Con prob. ϵ : elegir al azar (distribución uniforme).
 - Con prob. $(1-\epsilon)$: elegir el mejor brazo conocido.
- Estrategia *ϵ -first* (primero exploración, después explotación).
 - Primeros $(1-\epsilon)\%$ trials: elegimos al azar (distrib. uniforme).
 - Restantes $\epsilon\%$ trials: elegimos el mejor brazo conocido.
- Estrategia *softmax*:
 - La probabilidad de elegir a es proporcional al valor $Q(s,a)$:
$$\Pr(a) = \frac{e^{Q(s,a)/\tau}}{\sum_a e^{Q(s,a)/\tau}}$$
 $\tau > 0$ es la **temperatura**.
 τ alta: elección al azar (distr. uniforme)
 τ se reduce gradualmente.

Resumen

- Proceso de Decisión de Markov (MDP):

$$\langle S, A, T(s, a), R(s, a) \rangle$$

- Política π , función de valor V , función Q .
- Algoritmos *Value Iteration* y *Q-Learning*.
- *K-armed bandits*. Estrategias ϵ -greedy, ϵ -first, softmax.

- Taller opcional:

<https://inst.eecs.berkeley.edu/~cs188/sp12/projects/reinforcement/reinforcement.html>