Breve introducción a Reinforcement Learning

A. Atutxa

LSI BIlbao

Noviembre 2023

[1] Basado en el libro de Sutton y Barto, y lecciones de Adam y Martha White (U. Alberta), D. Silver (UCL, DeepMind) y Thomas Simonini (Dataiku)

Overview

Origen

El origen se remonta a las teorías conductivistas de Skinner (1904, Pensilvania EE.UU).

ChatGPT

El "Experimento de la caja de Skinner" se refiere a las investigaciones realizadas por el psicólogo Burrhus Frederic Skinner en el campo de la psicología del comportamiento. Skinner desarrolló una cámara o caja experimental que se conoce como la "caja de Skinner" o la "cámara de condicionamiento operante". Este dispositivo fue diseñado para estudiar el condicionamiento operante, un tipo de aprendizaje en el que el comportamiento de un organismo es modificado por las consecuencias que siguen a dicho comportamiento.

La caja de Skinner era un entorno controlado donde se colocaban animales, como ratas o palomas, y se les proporcionaba un mecanismo, como una palanca o un disco, que podían manipular para obtener recompensas, como comida o agua, o evitar castigos. El experimento se centraba en observar cómo los animales aprendían a asociar sus acciones con las consecuencias que seguían, lo que llevaba a la formación y modificación de comportamientos.

El condicionamiento operante se basa en la premisa de que los comportamientos seguidos por recompensas tienden a ser repetidos, mientras que aquellos seguidos por castigos tienden a disminuir. Skinner identificó varios principios clave, como el refuerzo positivo (añadir algo positivo para aumentar la probabilidad de que ocurra un comportamiento) y el castigo (añadir algo negativo para reducir la probabilidad de que ocurra un comportamiento).

El trabajo de Skinner en el condicionamiento operante y la caja de Skinner tuvieron un impacto significativo en la psicología y la comprensión del aprendizaje. Además, sus ideas y técnicas influyeron en diversas áreas, como la educación, la terapia del comportamiento y la modificación de conducta.

La caja de Skinner

Origen

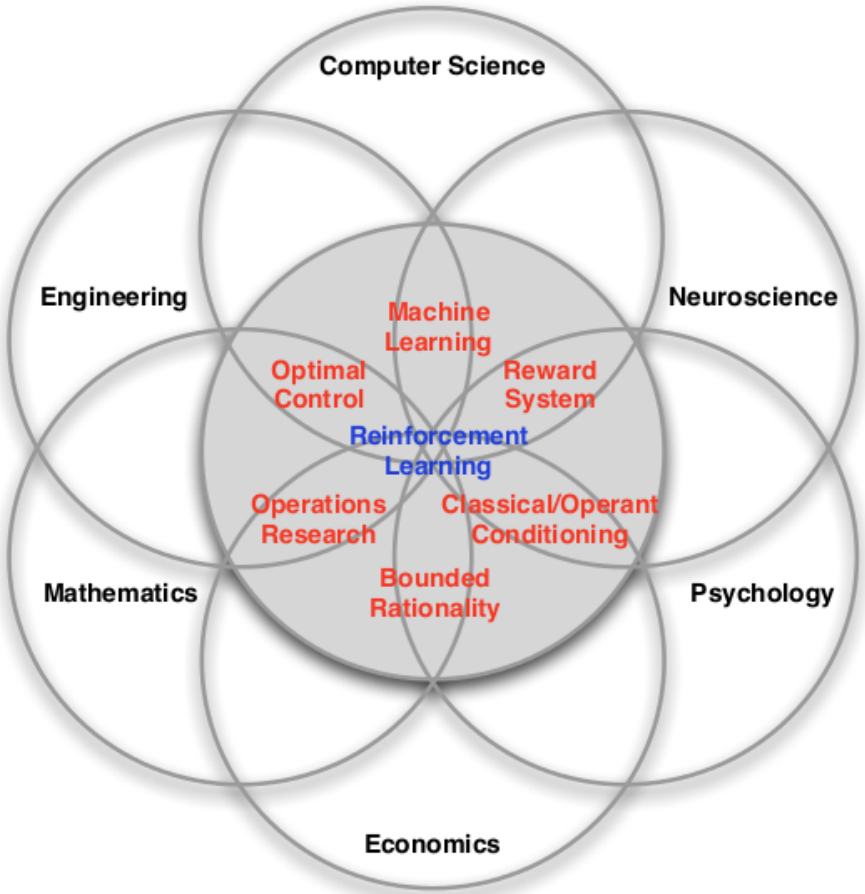
- B. Alexander: Exp. Parque de ratas “No es la persona. Es su jaula”
- `http://www.stuartmcmillen.com/es/comic/el-parque-de-las-ratas/#page-1`

Intuición

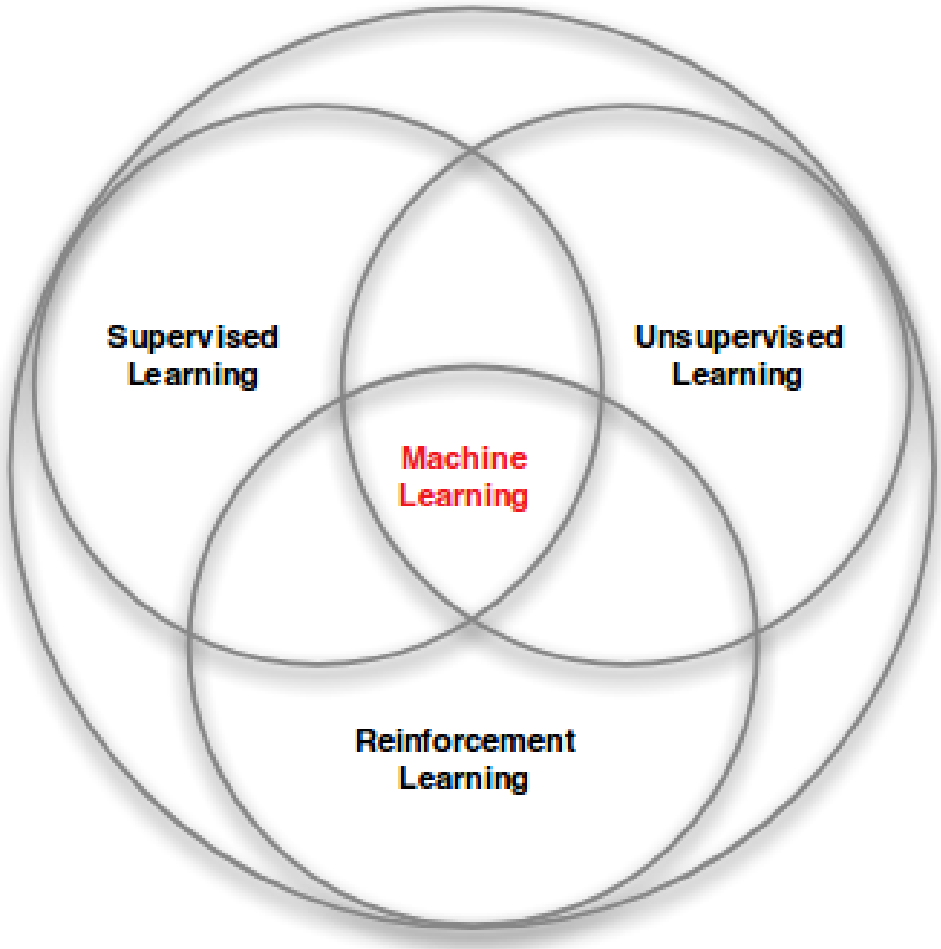
- Aproximación que estudia la toma de decisiones llevadas a cabo por un agente a base de prueba y error a través de su interacción con el entorno y guiado por una función de recompensa
- El agente no conoce cuál es la mejor acción que debe tomar en un momento concreto, pero una vez ejecutada una acción si puede valorar si ha sido positiva o negativa mediante las recompensas, aunque estás no sean inmediatas.

Otras ciencias

El Aprendizaje por Refuerzo (AR-RL) se encuentra en muchas disciplinas



Otras formas de ML



Aspectos diferenciadores

- En el aprendizaje por refuerzo (Reinforcement Learning), el agente genera sus propios datos de entrenamiento al interactuar con el mundo.
- El agente debe conocer las consecuencias de sus propias acciones a través de prueba y error, en lugar de que le digan cúal es la acción correcta.
- En aprendizaje supervisado: el premio es instantáneo, en RL puede ser tardío
- En RL el tiempo y la secuencialidad son importantes

[1] Transparencia basada en David Silver y Adam White

Ejemplos de RL

- Ganar contra un humano al BackGammon
- Un robot que aprende a realizar acciones (caminar, abrir puertas, etc)
- Gestionar una cartera de inversiones
- Jugar juegos

El Premio

- Un premio es un feedback escalar
- Indica una medida de lo bien/mal de la acción del agente en el paso t
- El objetivo del agente es obtener el mayor cúmulo de premios

RL se basa en la siguiente hipótesis (premisa):

Definición (La hipotesis del premio)

Todo *Objetivo-Goal* puede ser descrito como una maximización del cúmulo de premios esperado

Premios en ejemplos de RL

- Ganar contra un humano al BackGammon: puede que solo haya un premio final que es ganar (no habrá premios intermedios)
- Un robot que aprende a realizar acciones (caminar: premio por cada metro que avance, premio negativo si se cae)
- Gestionar una cartera de inversiones: premio por cada euro que ganes

Toma de decisiones secuencial

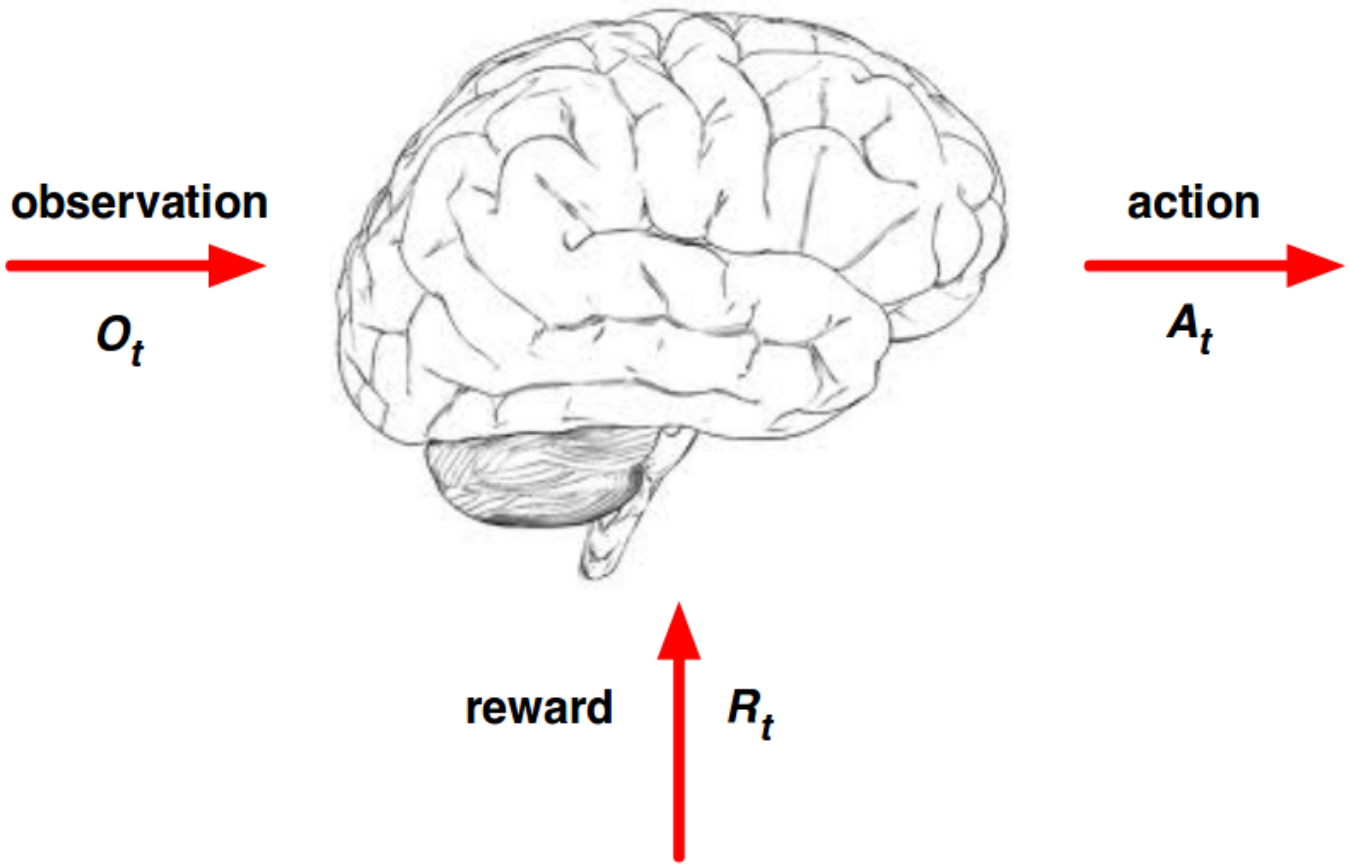
- **Objetivo de RL:** Seleccionar las acciones que maximicen el futuro premio acumulado
  - Cada acción puede tener consecuencias a largo plazo.
  - El premio no tiene porque ser inmediato
  - El mejor premio a corto plazo no tiene por qué ser el mejor a largo plazo. Actuar de forma *cortoplacista o Greedy* no siempre es la mejor estrategia (p.e. las inversiones)

Recordad que RL se basa en la siguiente hipótesis (premisa):

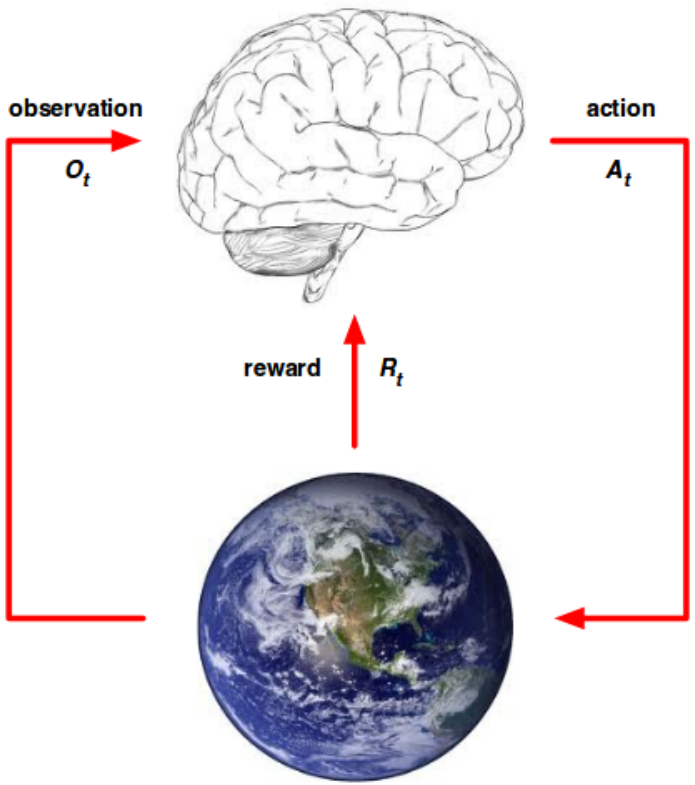
Definición (La hipotesis del premio)

Todo Goal puede ser descrito como una maximización del cúmulo de premios esperado

Contexto: Agente



Contexto: Agente y Entorno



- En cada paso t el agente:
  - Recibe/Percibe una observación
  - Ejecuta una acción
  - Recibe un premio
- En cada paso t el entorno:
  - Emite una observación
  - Recibe una acción
  - Emite un premio

Concepto clave

- **El entorno es incierto:** Cada acción que tome el agente genera una nueva observación (por ejemplo le lleva a un nuevo estado) pero no es un entorno determinista, es incierto, estocástico, es decir, el entorno regido por ciertas probabilidades subyacentes que hacen que desconozcamos lo que va a suceder.

Distintos tipos de entornos

- El entorno incierto más sencillo de modelar: **k-armed-bandits (máquinas tragaperras)**.
  - Realizar una u otra acción no tiene repercusión sobre los premios futuros.

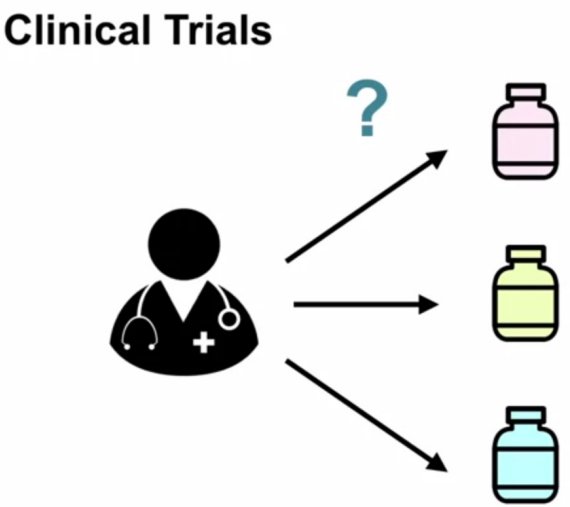
El problema de los k-armed bandits (las K máquinas tragaperras)

- Ejemplo básico de la *biblia* de RL (libro de Richard S. Sutton y Andrew G. Barto[1])
- Nos va a permitir:
  - Formalizar **la toma de decisiones** bajo **incertidunbre**
  - Entender: **acción, premio, valor de una acción**
- Ejecutar :
  `https://mdp.ai/coursera/c01-k-armed-bandit/`

[1] `https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf`

El problema de los k-armed bandits (Decisión médica)

El médico se enfrenta al problema de decidir qué tratamiento debe prescribir



- El agente es el médico
- El entorno proporciona:
  - Una observación, en este caso, las acciones o prescripciones posibles
  - El premio asociado a cada prescripción
- El agente deberá elegir entre las acciones o prescipciones de distintos tratamientos.

El valor de una acción

Concepto: **valor de una acción**

- Para Seleccionar una acción:
  - necesitamos asociar un **valor a cada acción** (action-value)

Definición del valor de una acción

Esperanza de premio para esa acción: El valor del premio esperado si realizamos esa acción (bajo el supuesto de ser conocedores de la distribución que siguen los premios-rewards)

$$q*(a) \doteq \mathbb{E}[R_t|A_t = a] \, \forall a\{1, 2, ...k\}$$

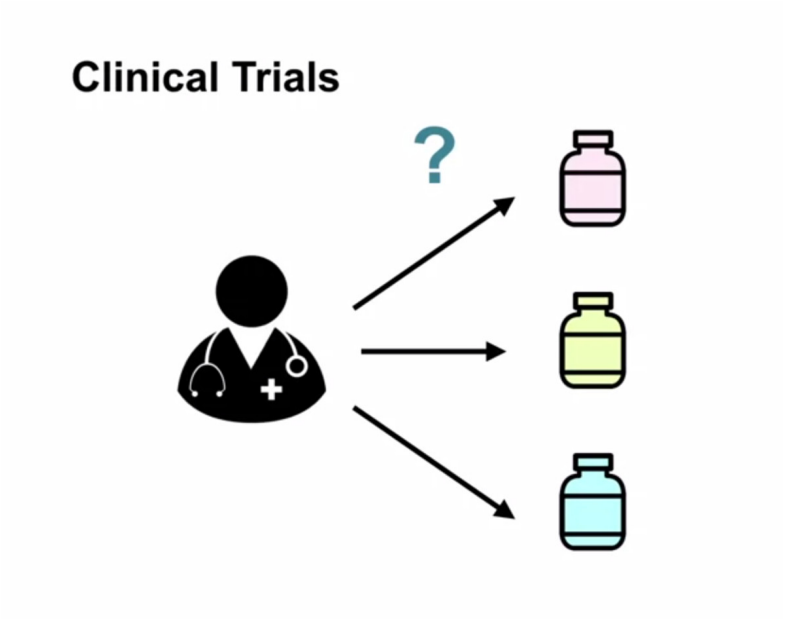
$$\mathbb{E}[R_t|A_t = a] = \sum_r p(r|a) * r$$

Definición (La hipotesis del premio)

Todo *Objetivo* puede ser descrito como una maximización del cúmulo total de premios esperado

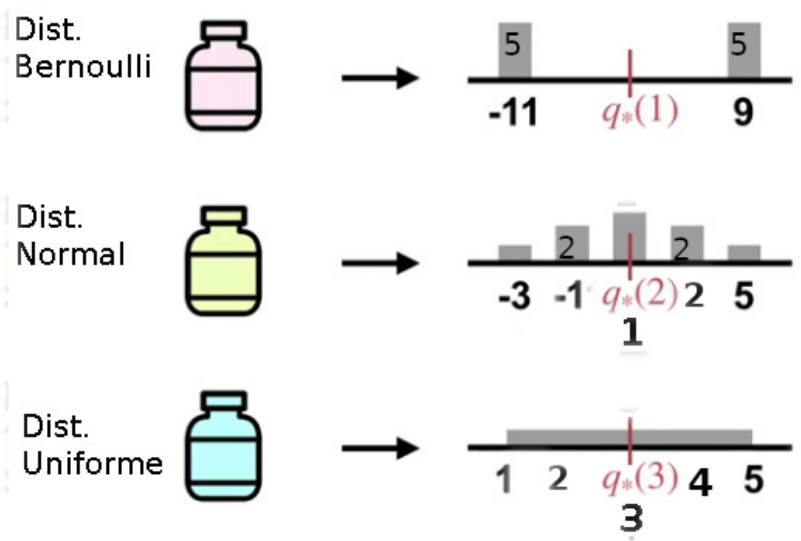
El problema de los k-armed bandits (Decisión médica)

1. **Observación:**
El agente dispone de 3 posibles tratamientos



El problema de los k-armed bandits (Decisión médica)

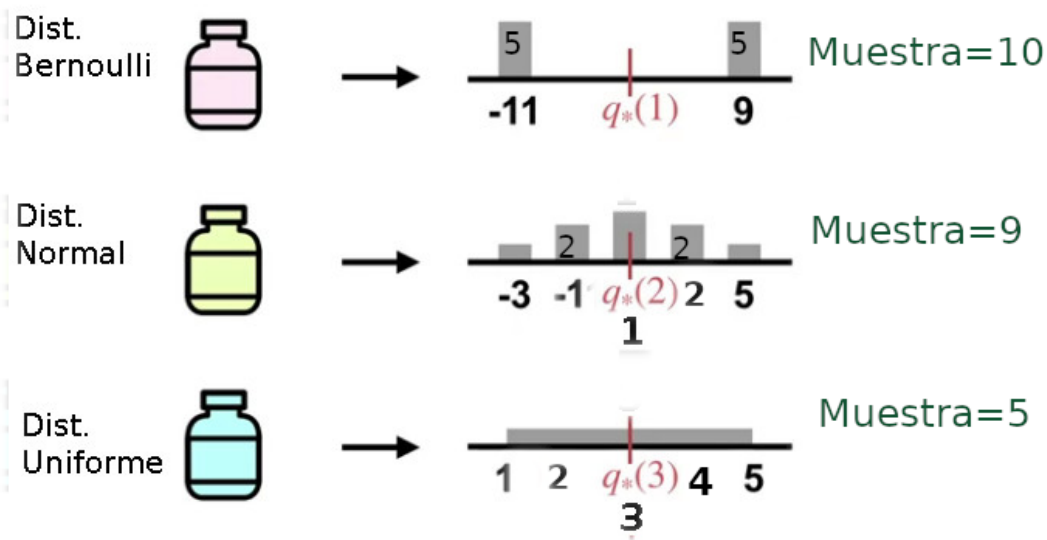
- Sabemos que las medicaciones alteran la capacidad de bombeo del corazón decrementándola o incrementándola.
- Elegiremos aquella cuya esperanza de aumentar la capacidad de bombeo es la mayor.
- Supongamos que conocemos la distribución de premios subyacente en cada opción.



El problema de los k-armed bandits (Decisión médica)

- Supongamos que conocemos la distribución de premios subyacente en cada opción.

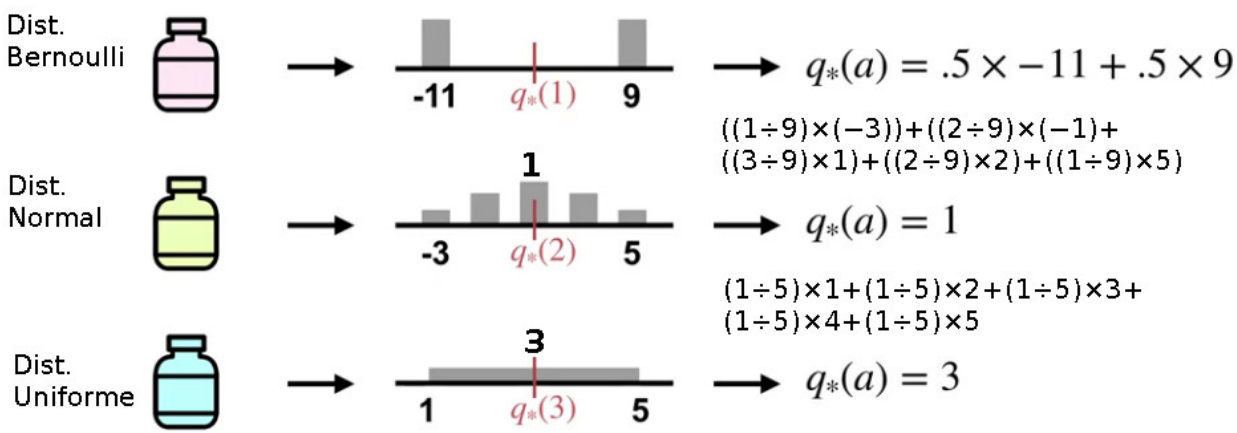
$$\mathbb{E}[R_t|A_t = a] = \sum_r p(r|a) * r$$



El problema de los k-armed bandits (Decisión médica)

- Supongamos que conocemos la distribución de premios subyacente en cada opción.

$$\mathbb{E}[R_t|A_t = a] = \sum_r p(r|a) * r$$

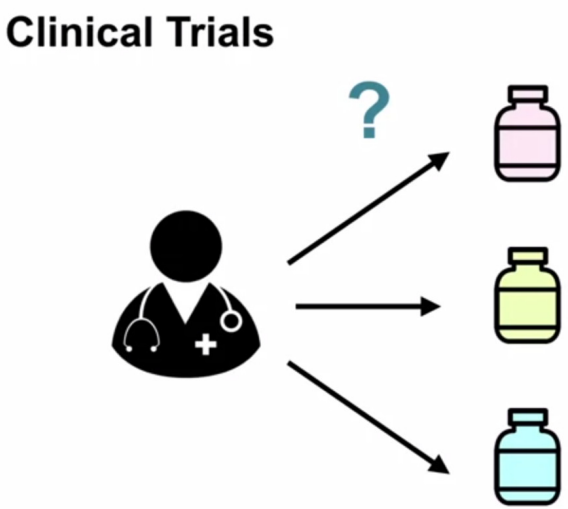


El problema de los k-armed bandits (Decisión médica)

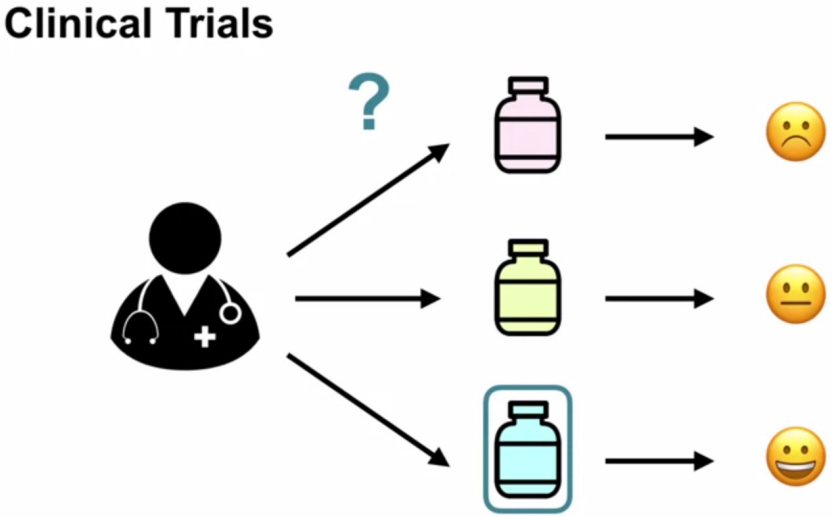
- La realidad es que el médico no sabe cuál es la distribución tras cada tratamiento
- Tiene que realizar una a la estimación a la distribución real....

El problema de los k-armed bandits (Decisión médica)

1. Observación:
tenemos 3 posibles tratamientos



2. Probamos aleatoriamente:
tras n intentos dispondremos de una estimación



Estimando el valor de una acción: valor medio de la muestra

Definición del valor medio de la muestra de una acción

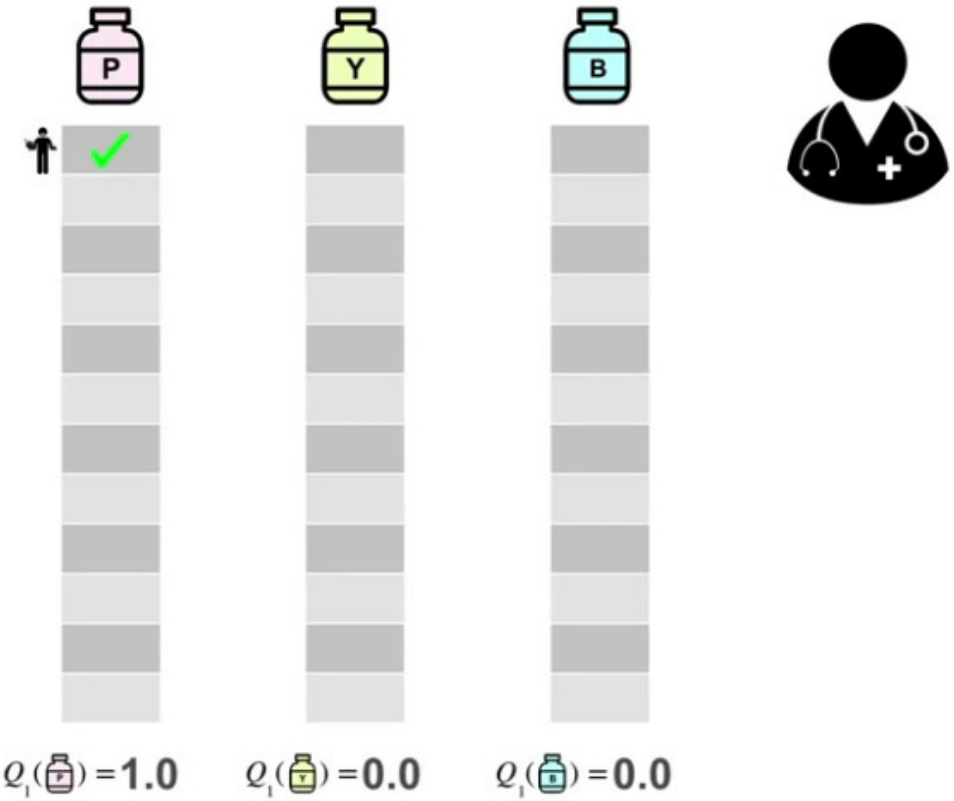
$$Q_t(a) \doteq \frac{\textit{suma de los premios al tomar la accion a previamente a } t}{\textit{num veces que se ha elegido a previamente a } t}$$

$$\frac{\sum_{i=1}^{t-1} R_i^a}{\#a_{(1..t-1)}}$$

Estimando el valor de una acción en base a la estimación anterior

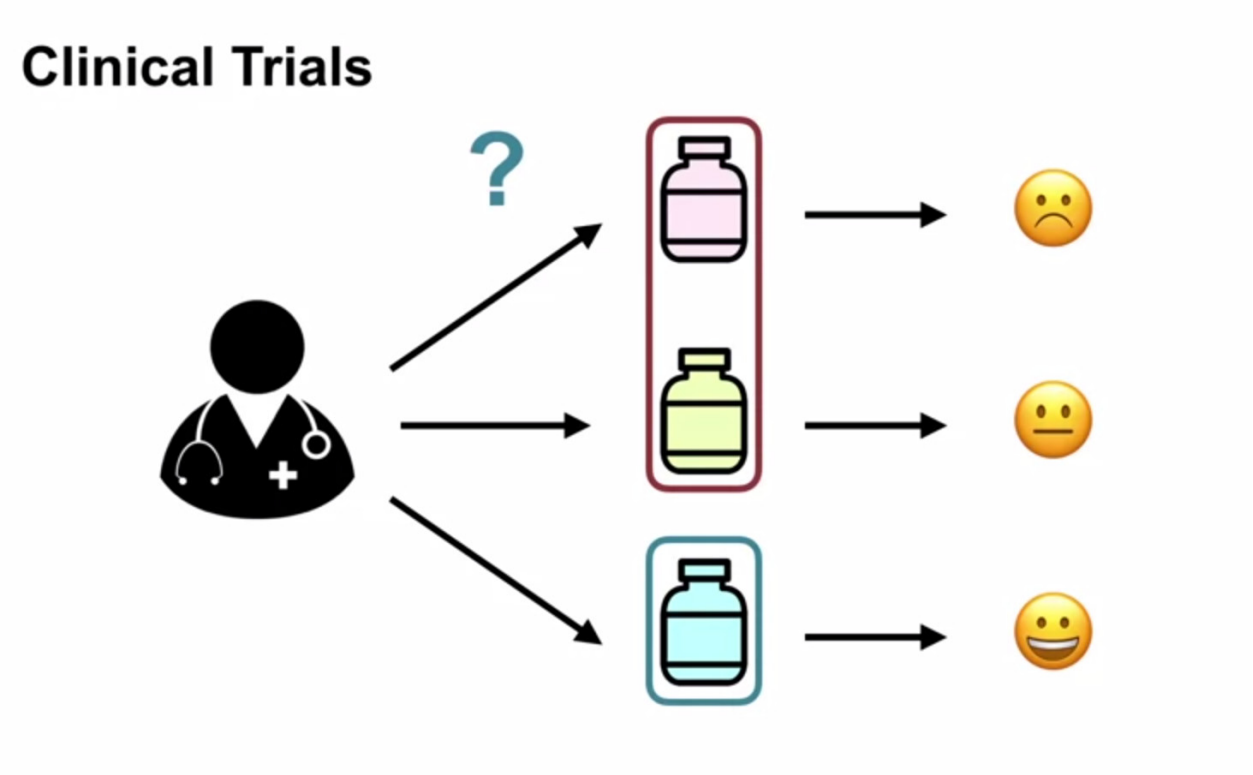
El premio es 1 si el tratamiento tiene exito y 0 en caso contrario

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



El problema de los k-armed bandits (Decisión médica)

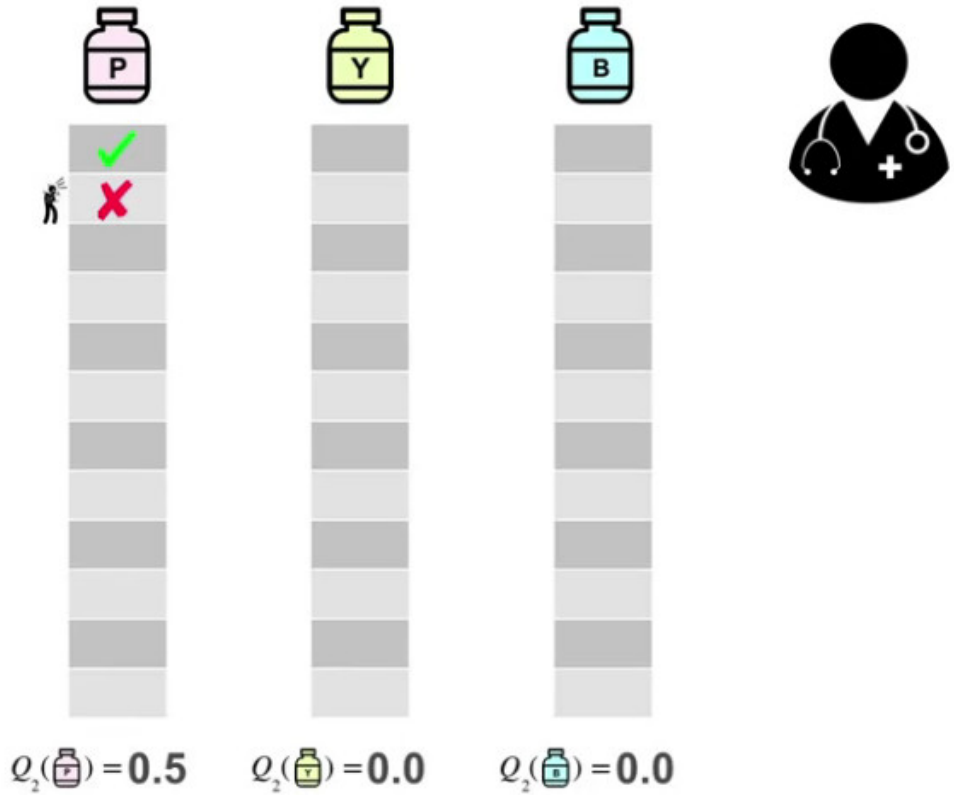
3. Seleccionamos una acción
lo cual implica desechar
las otras alternativas



Estimando el valor de una acción en base a la estimación anterior

El premio es 1 si el tratamiento tiene exito y 0 en caso contrario

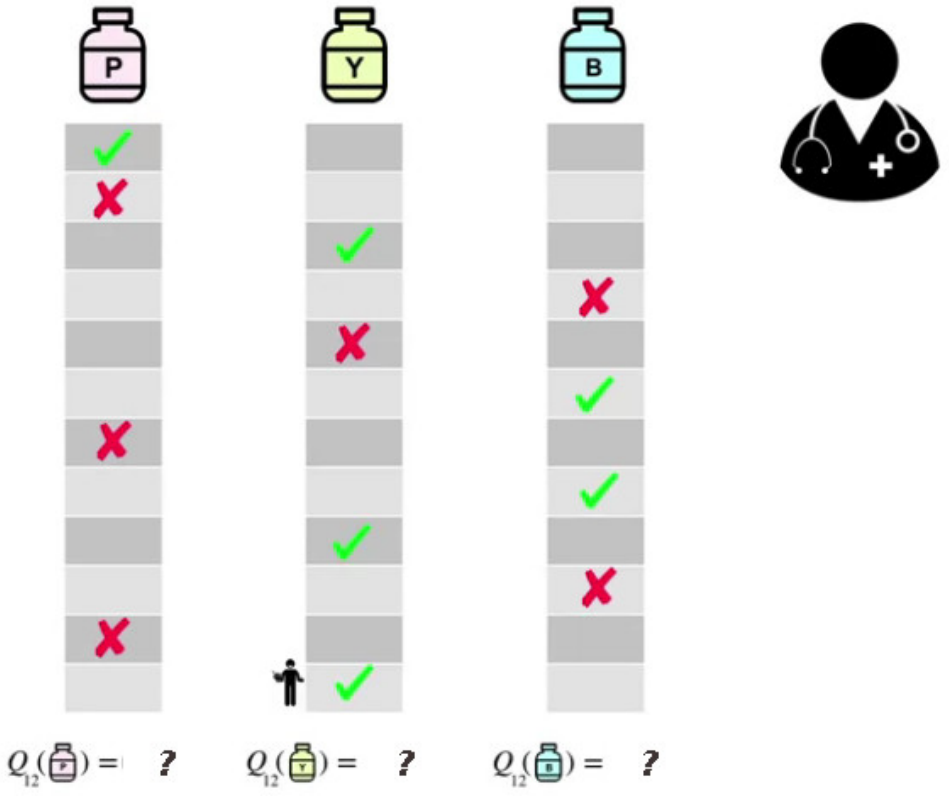
$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



Estimando el valor de una acción en base a la estimación anterior

El premio es 1 si el tratamiento tiene exito y 0 en caso contrario

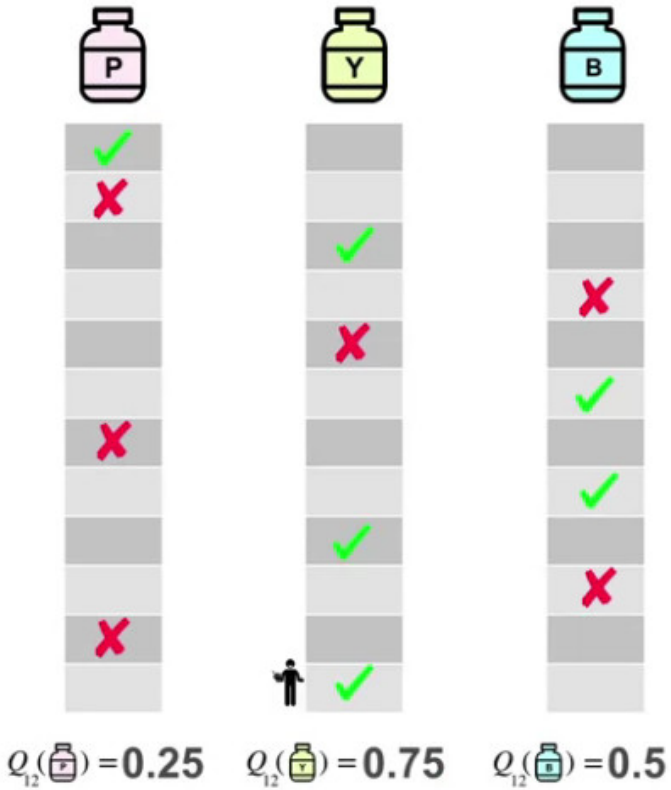
$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



Estimando el valor de una acción en base a la estimación anterior

El premio es 1 si el tratamiento tiene exito y 0 en caso contrario

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



Estimando el valor de una acción: valor medio de la muestra

Así no hay que almacenar ni recalcular todo otra vez.

$$Q_{t+1}(a) = \frac{1}{n}\sum_{i=1}^{t} R_i^a$$

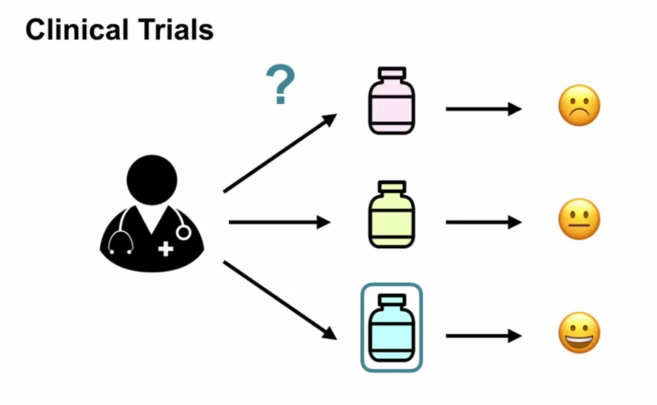
$$\frac{1}{n}\left(R_t^a + \sum_{i=1}^{t-1} R_i^a\right) = \frac{1}{n}\left(R_t^a + \frac{n-1}{n-1}\sum_{i=1}^{t-1} R_i^a\right)$$

$$\frac{1}{n}\left(R_t^a + (n-1)\frac{1}{n-1}\sum_{i=1}^{t-1} R_i^a\right) = \frac{1}{n}\left(R_t^a + (n-1)\,Q_t(a)\right)$$

$$\frac{1}{n}\left(R_t^a + nQ_t(a) - Q_t(a)\right) = Q_t(a) + \frac{1}{n}\left(R_t^a - Q_t(a)\right)$$

K-bandits vs. Markov Decision Processes (MDP)

Consecuencias de cada accion en el entorno: No influencia sobre posteriores premios



Consecuencias de cada accion en el entorno: influencia sobre posteriores premios

Formalización

Formalización como proceso de decisión de Markov

- $M =< S, >, T, R >$
- S: Conjunto finito de estados, $S_t \in S$
- A: Conjunto finito de acciones disponibles. $A_t \in A(S_t)$. A_t es la acción en el instante t que pertenece a las acciones disponibles en el estado S_t.
- T: Función de transición. Cuando se trata de un entorno estocástico $T : S \times A \times S \rightarrow P(S)$.

$$T(s'|s,a) = Pr(S_{t+1} = s'|S_t = s, A_t = a)$$

$$\sum_{s' \in S} T(s'|s,a) = 1$$

- $R : S \times A \times S \wedge R \in \mathbb{R}$

Formalización

Formalización como proceso de decisión de Markov

- Los MDP se caracterizar por estar modelados de forma que cumplen la propiedad de Markov: Un estado S_t es un estado Markoviano si y solo si

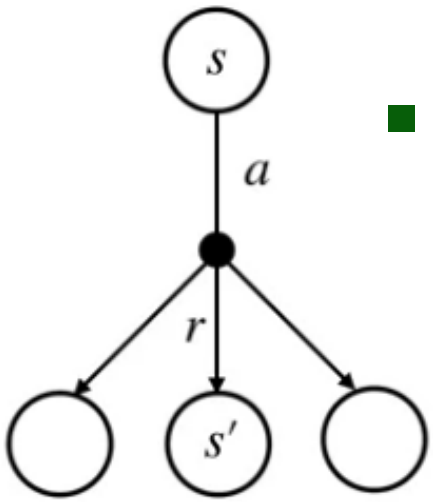
$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2, ...S_t]$$

Más adelante se hablará de los estados

Contexto: Agente y Entorno en MDPs

- Supongamos que conocemos la distribución de probabilidades asociada a cada acción.

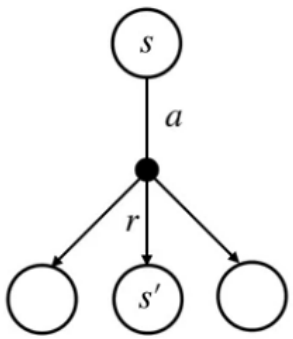
Contexto: Agente y Entorno en MDPs



- En s realizamos a
- Recibiremos un premio (r) y alcanzaremos el estado s´, dependiendo de la distribución de probabilidad oculta
- En el ejemplo del pacman si nos decidimos a ir a la izquierda:
  - con prob(X): premio $+10$ y $s´ =$ la cereza no está y el fantasma se ha movido hacia la izda
  - con prob(1-X): premio -100 y $s´ =$ la cereza y el pacman no están!!, porque el fantasma se ha movido hacia la derecha y nos ha comido!!

Contexto: Agente y Entorno en MDPs

Probabillidad de transición



$$\sum_{s}\sum_{a} Prob(s',r|s,a) = 1, \forall s \in S \forall a \in A(s)$$

Contexto: Agente y Entorno en MDPs

Bajo el supuesto de que la distribución subyacente es conocida, existe una fórmula para calcular la bonanza de un estado como veremos más adelante (Value Iteration[2]):

$$\forall s \in S, V_{k+1} \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

Pero antes aprendamos a formalizar un MDP

[2] gamma: es el factor de descuento que representa la preferencia por premios a corto plazo versus premios a futuro

Modelar un MDPs

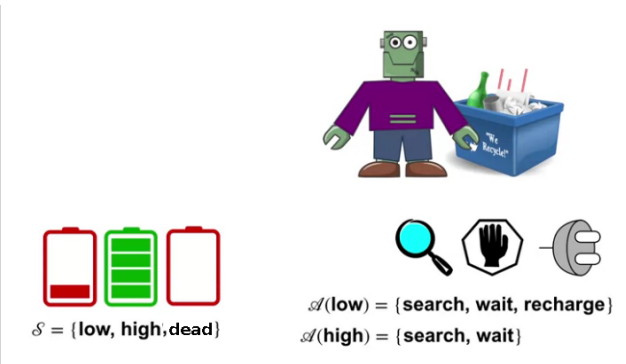
Supongamos un robot reciclador que busca y recoge latas[3]

- **Objetivo del robot:** Recoger el máximo de latas hasta gastarse la batería

[3] Ejemplo adaptado de Adam y Martha White

Modelar un MDPs

Supongamos un robot reciclador que busca latas[4]



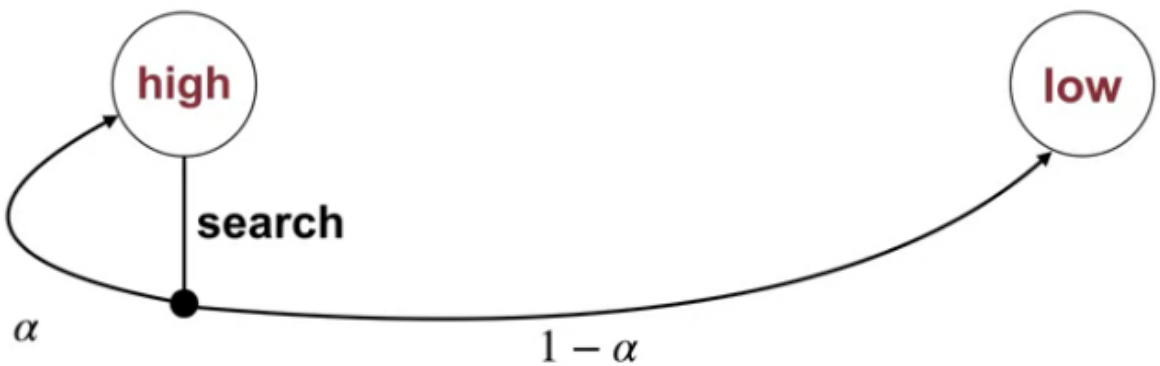
- **Estados:** {batería OK, batería baja, muerto}
- **Acciones:**{Buscar, Esperar, VolverAlDock}

[4] Ejemplo adaptado de Adam y Martha White

Función de transición:

$T(high, search, low) = 1 - \alpha$

$T(high, search, high) = \alpha$

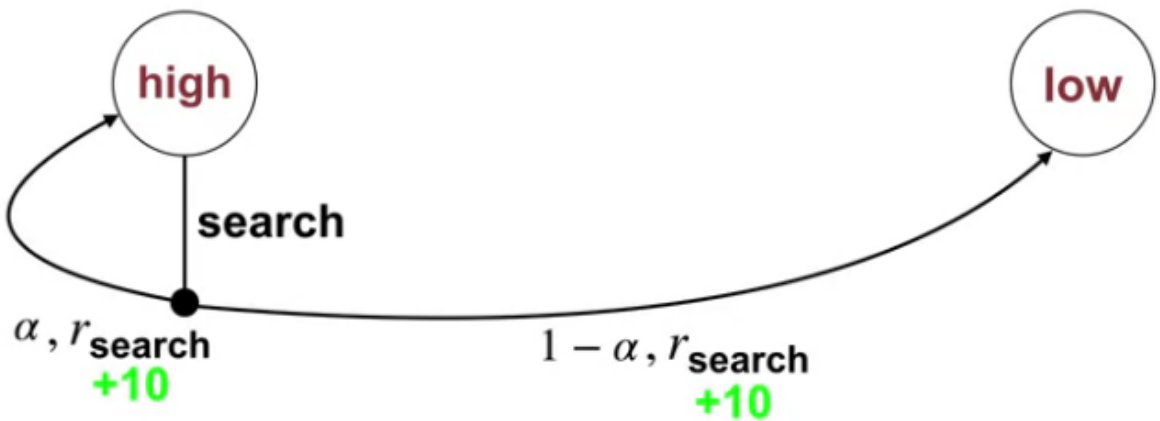


Modelar un MDPs

Función de Reward (premio):

$R(high, search, low) = +10$

$R(high, search, high) = +10$



Modelar un MDPs

Función de transición:

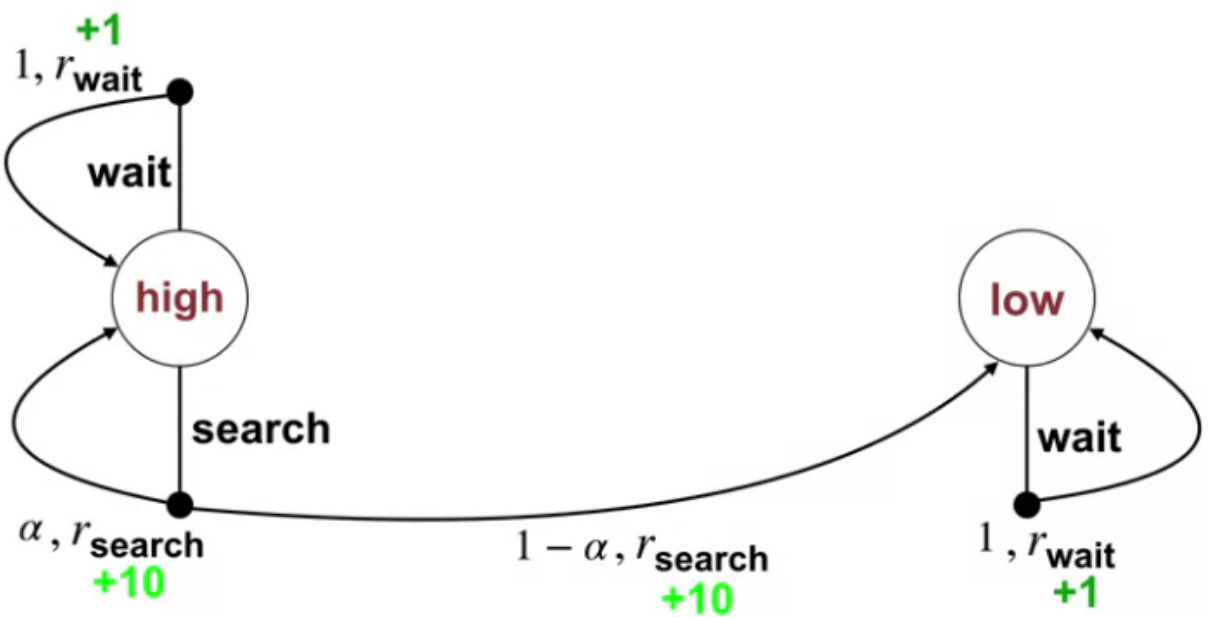
$T(high, wait, high) = 1$

$T(low, wait, low) = 1$

Función de Reward (premio):

$R(high, wait, high) = +1$

$R(low, wait, low) = +1$



Modelar un MDPs

Función de transición:

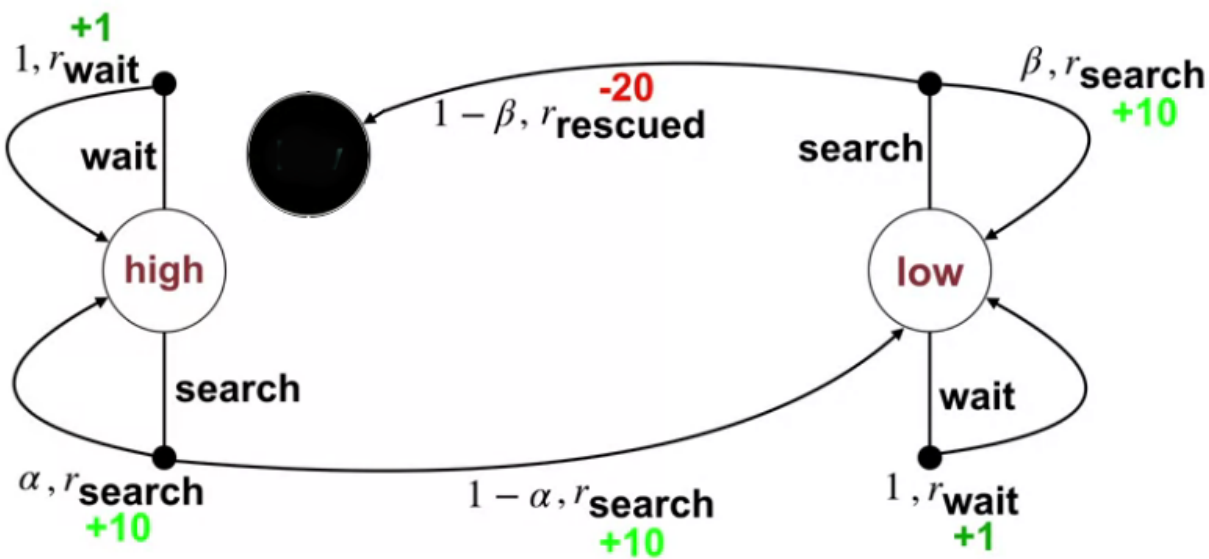
$T(low, search, low) = \beta$

$T(low, search, dead) = 1 - \beta$

Función de Reward (premio):

$R(low, search, low) = +10$

$R(low, search, dead) = -20$



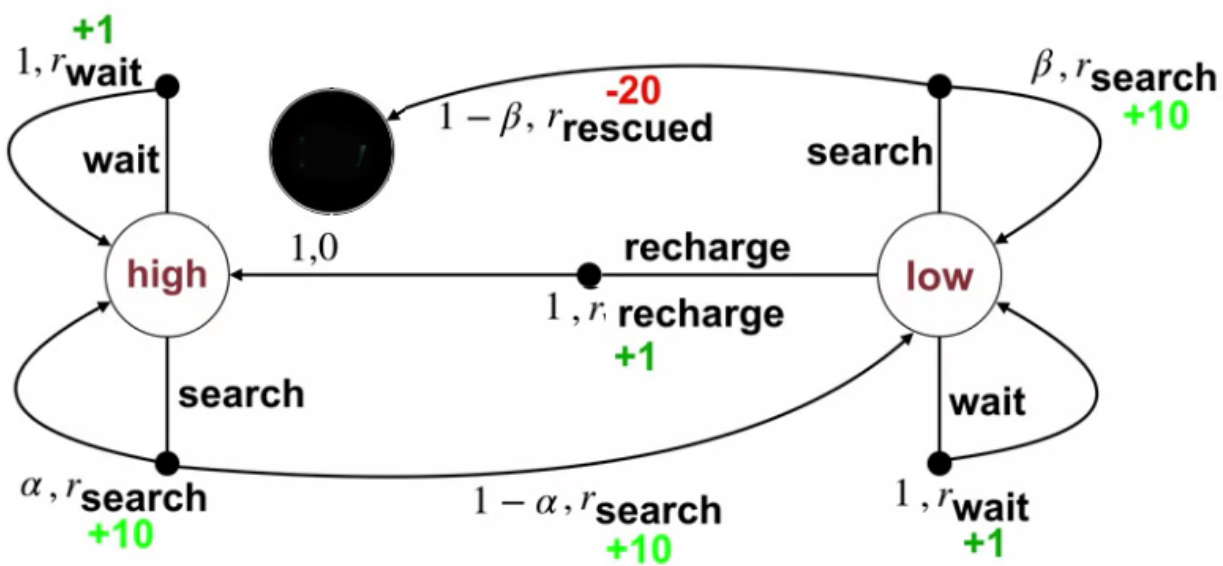
Modelar un MDPs

Función de transición:

$T(low, recharge, high) = 1$

Función de Reward (premio):

$R(low, recharge, high) = +1$



Modelar un MDPs

Función de transición:

$T(high, search, low) = 1 - \alpha$
$T(high, search, high) = \alpha$
$T(low, search, low) = \beta$
$T(low, search, dead) = 1 - \beta$
$T(high, wait, high) = 1$
$T(low, wait, low) = 1$
$T(low, recharge, high) = 1$

Función de premio:

$R(high, search, low) = +10$
$R(high, search, high) = +10$
$R(low, search, low) = +10$
$R(low, search, dead) = -20$
$R(high, wait, high) = +1$
$R(low, wait, low) = +1$
$R(low, recharge, high) = +1$

Modelar un MDP

Formalizar el siguiente ejemplo[5]:
Deseamos recorrer la mayor distancia posible con un coche viejo que corre el riesgo de estropearse si lo forzamos demasiado. Así podemos decidir acelerar o no, si aceleramos recorremos más distancia y si vamos despacio recorremos menos pero existe menos riesgo de que el coche se recaliente y se termine estropeando.

Así, supondremos que los estados son:

Las acciones son: acelerar, no acelerar

[5] Ejemplo adaptado de Dan Klein

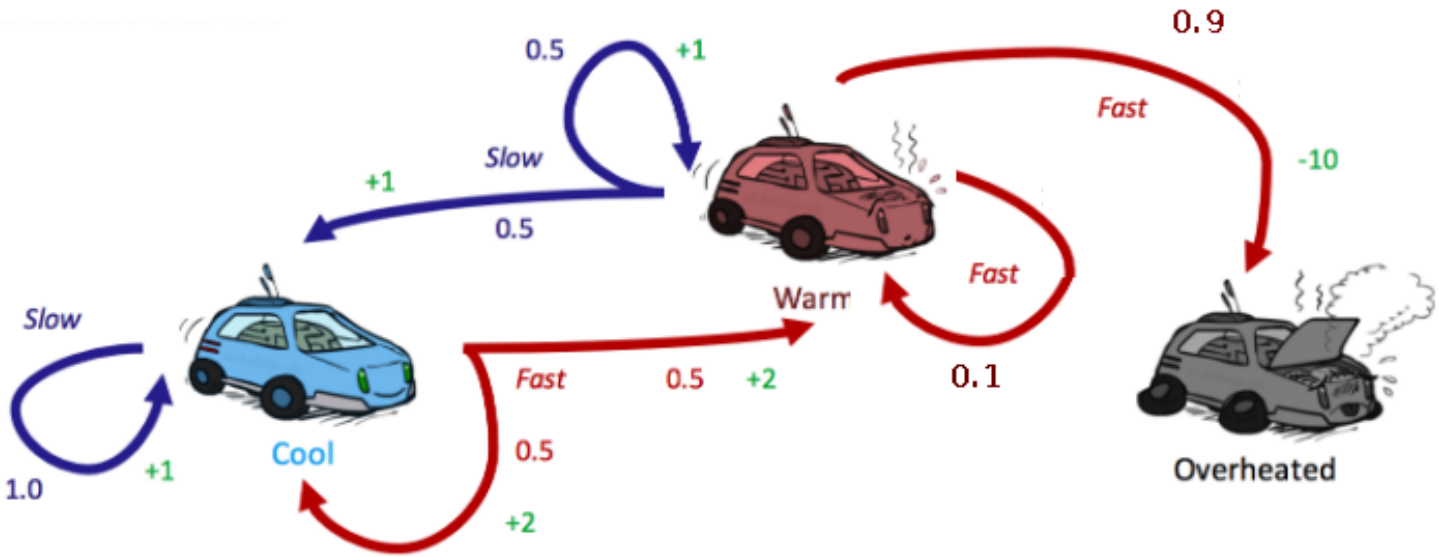
Modelar un MDP

Formalizar el siguiente ejemplo:
Cuando el coche está normal (frío), la probabilidad de que al acelerar se recaliente es de 0.5. Cuando el coche está recalentado, la probabilidad de que al no acelerar se enfrie y vuelva a estar normal es de 0.5, mientras que si estando recalentado aceleramos la probabilidad de que se estropee es de 0.9. Dibuja el MDP. ¿Cúal sería la función de transición asociada? Asigna premios +1, +2 y -20, según creas conveniente recordando que el objetivo es recorrer la mayor distancia posible.

Así supondremos que los estados son:

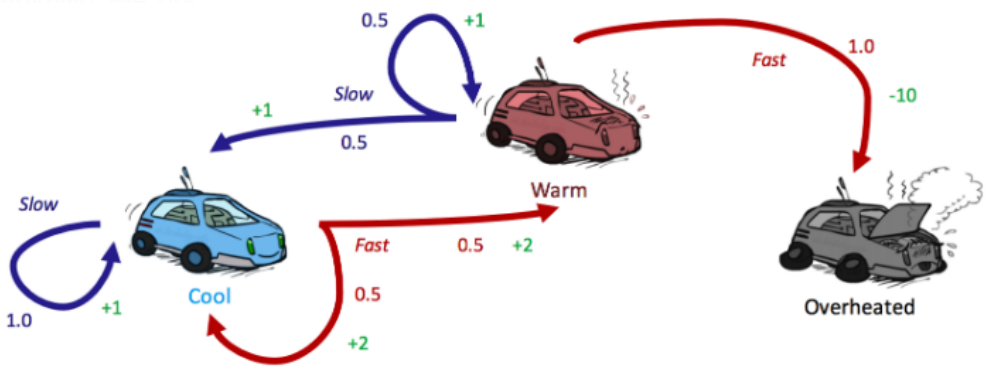
Las acciones son: acelerar, no acelerar

Modelar un MDP



Value Iteration

Suponiendo un MDP similar al anterior:



Existe una fórmula para calcular la bonanza de un estado (Value Iteration[6]):

$$\forall s \in S,\ V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

[6] γ: factor de descuento representa preferencia por premios a corto plazo versus premios a futuro. Si $\gamma = 0$, el Agente preferencia por la recompensa inmediata y descarta la rentabilidad a largo plazo. Si $\gamma = 1$, considerará todas las recompensas futuras iguales a la recompensa inmediata.

Value Iteration

Aplicando la fórmula de la iteración del valor para calcular $V_k + 1(\gamma = 0.5)$:

$$\forall s \in S,\ V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

Función de transición:

- $T(nor, no\text{-}acel, nor) = 1$
- $T(nor, acel, nor) = 0.5$
- $T(nor, acel, calien) = 0.5$
- $T(cal, no\text{-}acel, nor) = 0.5$
- $T(cal, no\text{-}acel, cal) = 0.5$
- $T(cal, acel, muerto) = 1$

	cool	warm	overheated
V_0	0	0	0

$$\begin{aligned}
V_1(cool) &= \max\{1\cdot[1+0.5\cdot 0],\ 0.5\cdot[2+0.5\cdot 0]+0.5\cdot[2+0.5\cdot 0]\} \\
&= \max\{1,2\} \\
&= \boxed{2} \\
V_1(warm) &= \max\{0.5\cdot[1+0.5\cdot 0]+0.5\cdot[1+0.5\cdot 0],\ 1\cdot[-10+0.5\cdot 0]\} \\
&= \max\{1,-10\} \\
&= \boxed{1} \\
V_1(overheated) &= \max\{\} \\
&= \boxed{0}
\end{aligned}$$

	cool	warm	overheated
V_0	0	0	0
V_1	2	1	0

Value Iteration

Aplicando la fórmula de la iteración del valor para calcular $V_k + 1(\gamma = 0.5)$:

$$\forall s \in S,\ V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

Función de transición:

- $T(nor, no\text{-}acel, nor) = 1$
- $T(nor, acel, nor) = 0.5$
- $T(nor, acel, calien) = 0.5$
- $T(cal, no\text{-}acel, nor) = 0.5$
- $T(cal, no\text{-}acel, cal) = 0.5$
- $T(cal, acel, muerto) = 1$

$$\begin{aligned}
V_2(cool) &= \max\{1\cdot[1+0.5\cdot 2],\ 0.5\cdot[2+0.5\cdot 2]+0.5\cdot[2+0.5\cdot 1]\} \\
&= \max\{2, 2.75\} \\
&= \boxed{2.75} \\
V_2(warm) &= \max\{0.5\cdot[1+0.5\cdot 2]+0.5\cdot[1+0.5\cdot 1],\ 1\cdot[-10+0.5\cdot 0]\} \\
&= \max\{1.75, -10\} \\
&= \boxed{1.75} \\
V_2(overheated) &= \max\{\} \\
&= \boxed{0}
\end{aligned}$$

	cool	warm	overheated
V_0	0	0	0
V_1	2	1	0
V_2	2.75	1.75	0

Value Iteration

Algorithm 1 pseudocodigo Value Iteration

1: **while** ¬convergencia **do**
2: k=0
3: **while** k¡=num estados **do**
4:

$$V_{k+1} \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

5: **end while**
6: **end while**

- versión síncrona o asíncrona: Asíncrona emplea en cada vuelta los valores actualizados en esa vuelta. Síncrona, emplea los valores actuales y actualiza al final de cada vuelta (laboratorio).
- convergencia: número n de iteraciones y/o$\forall s | V_k[s] - V_{k-1}[s]| < \Theta$

Política

Concepto: **Política**

- La política: mapeo entre los estados del entorno percibidos por el agente y las acciones que el agente realizará cuando alcance cada uno de esos estados.
- Se suele representar con la letra griega π y habrá tantas como combinaciones de acciones y estados haya
- El aprendizaje consiste en encontrar las política optima π^* de entre todas las posibles
- Value Iteration Permite encontrar la política óptima si conocemos la distribución subyacente del entorno .
- Al finalizar el Value Iteration sabemos cuales son los valores optimos V^* de cada estado:

$$\pi^* = \arg\max_{a \in A} \sum_{s' \in S} T(s, a, s')(R(s, a, s')) + \gamma V^*(s')$$

Política

Concepto: **Política**

3 pasos:

- Inicialización: Seleccionar de forma aleatoria una política, es decir, dado un estado fijar una acción.
- Evaluación: Obtener un valor para esa política.
- Actualización: En base a los valores de cada estado s', seleccionando una mejor política. Policy improvement:

$$\pi^* = \arg\max_{a \in A} \sum_{s',r} T(s,a,s')(R(s,a,s')) + \gamma V^(s')$$

Contexto: Historia vs Estado

- Historia: secuencia de observaciones, acciones y premios

$$H_t = A_1, O_1, P_1, A_2, O_2, P_2, ..., A_t, O_t, P_t$$

- El futuro depende del pasado (la historia).
  - El agente seleccionará la acción en base a la historia. Nuestro objetivo es crear un mapping entre la historia en t y una acción
  - El entorno proveerá de una observación también dependiendo de la historia
- Estado: la historia es demasiado compleja de computar. Se emplea el estado que de alguna forma debe representar/resumir esa historia.

$$S_t = f(H_t)$$

Formalización del estado: Asumiendo Markov

Un estado contiene información útil sobre la historia:

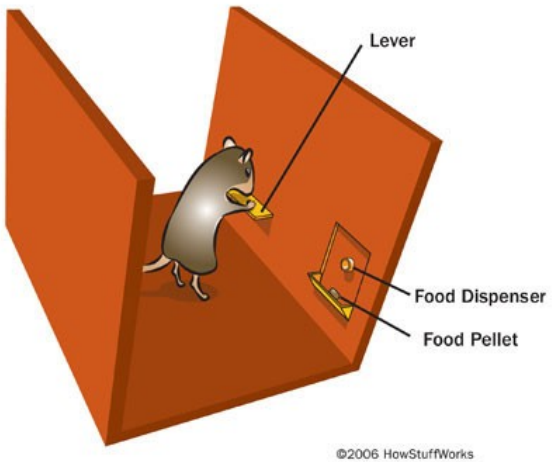
Definición

Un estado S_t es un estado Markoviano si y solo si

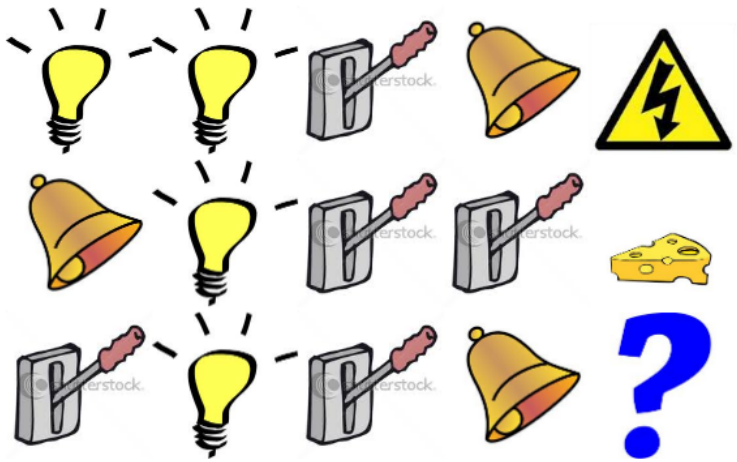
$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2, ...S_t]$$

El futuro es independiente del pasado dado el presente (el estado actual)

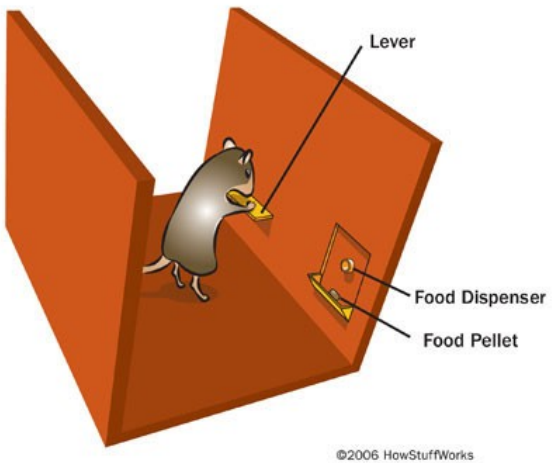
El Estado



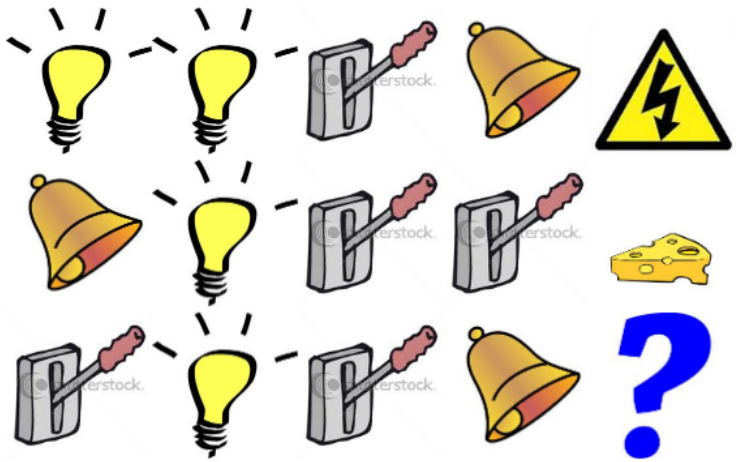
Skinner Box



El Estado

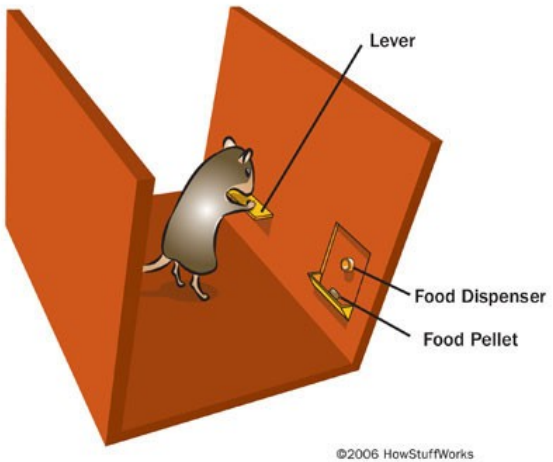


Skinner Box

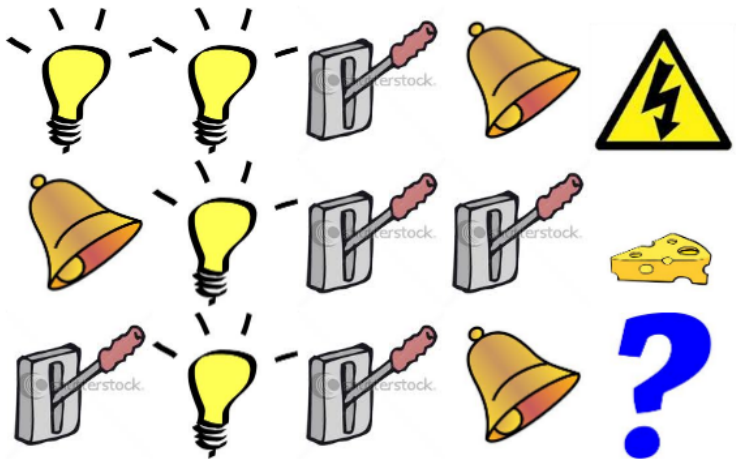


- ¿Si el estado = los 3 últimos elementos?

El Estado

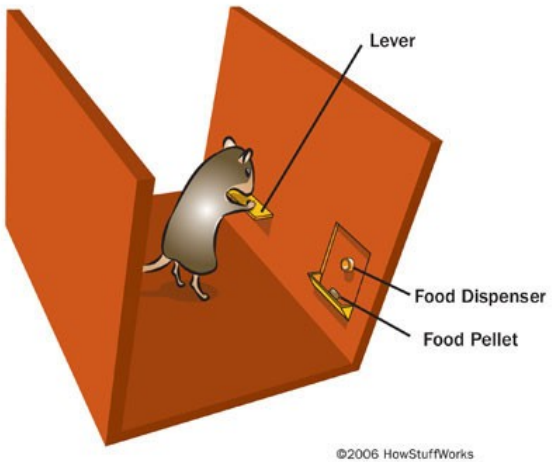


Skinner Box

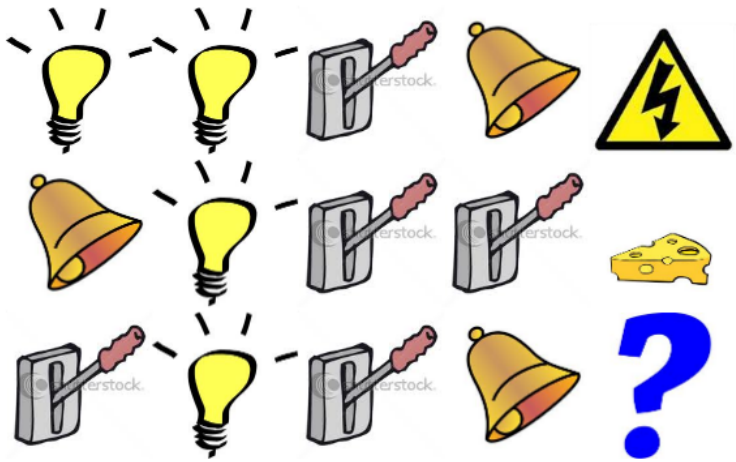


- ¿Si el estado = los 3 últimos elementos?
- ¿Si el estado = contadores de luces, campanas y palancas?

El Estado



Skinner Box



- ¿Si el estado = los 3 últimos elementos?
- ¿Si el estado = contadores de luces, campanas y palancas?
- ¿Si el estado = la secuencia completa?

Value Iteration (Iteración del Valor)

- La iteración del valor se utiliza cuando se conocen las probabilidades de transición. Por ejemplo, $T(high, search, low) = 0.8$ $T(high, search, high) = 0.2$
- En la mayoria de los casos no se conoce la probabilidad de transición. No dispone de la distribución oculta del modelo (entorno). Entonces se emplea el Q learning.

Bibliografía

- Reinforcement Learning, An Introduction (Second Edition). By Richard S. Sutton and Andrew G. Barto
  `https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf`
- DLRL2019 (Adam White):
  `https://www.youtube.com/watch?v=RancMV1wECg`
- experimentos de Skinner
  `https://www.stuartmcmillen.com/es/comic/el-parque-de-las-ratas/#page-14`
- Ejemplo del código del TIC-TAC-TOE:
  https://towardsdatascience.com/reinforcement-learning-implement-tictactoe-189582bea542