

Sistemas Recomendadores

IIC-3633

Aprendizaje Reforzado en Sistemas Recomendadores

Esta clase

1. Recomendación basada en aprendizaje reforzado

Tipos de aprendizaje en recomendación

Aprendizaje Supervisado

Aprende de feedback dado por el usuario:

- Ratings
- Interacciones
- Clasificación

Aprendizaje No Supervisado

Caracteriza a usuarios e ítems sin etiquetas.

Ejemplo: clustering.

Aprendizaje Reforzado

Aprende de interacción en tiempo real del usuario con la plataforma/entorno.

Quiero **optimizar algo que no se puede expresar en una función de pérdida.**

RESTAURANTE NUEVO VIETNAM



Mayor grado de exploración

RESTAURANTE NUEVO CHINA



Grado de exploración intermedio

RESTAURANTE DE SIEMPRE PERÚ



Baja exploración

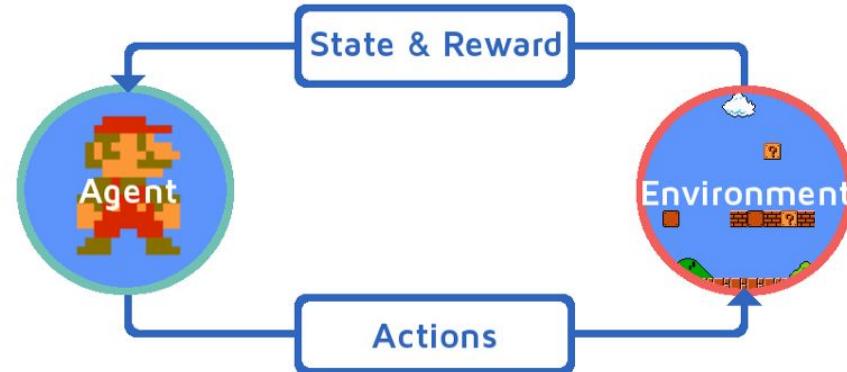
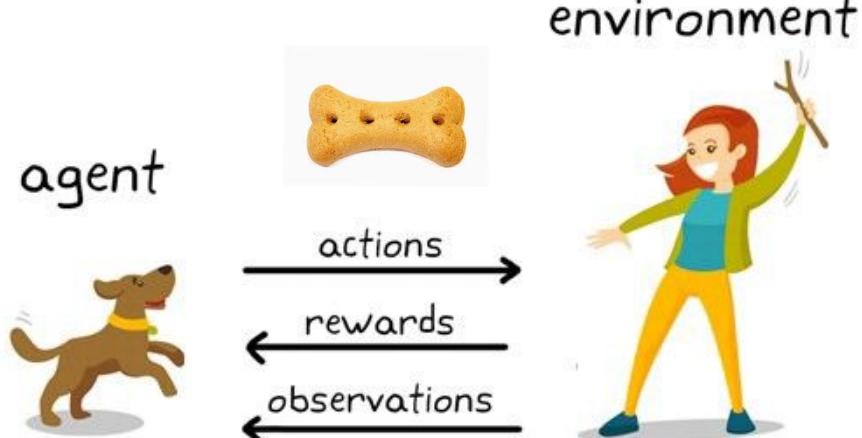


Dilema: Exploración vs Explotación

APRENDIZAJE REFORZADO

APRENDIZAJE POR PRUEBA Y ERROR.

EXPLORAR ACCIONES NUEVAS APOSTANDO POR UNA RECOMPENSA A COSTA DE PERDER TIEMPO Y RECURSOS.



Aprendizaje reforzado en sistemas recomendadores

Manejo de la incertidumbre.

COLD START (USUARIOS E ÍTEMS)

CAMBIO DINÁMICO EN LAS PREFERENCIAS DE LOS USUARIOS

CAMBIO DINÁMICO EN EL CATÁLOGO DE PRODUCTOS



CAMBIOS EN EL CONTEXTO DE USO DE LA PLATAFORMA.
(WEB, TV, MÓVIL)

INFORMACIÓN CAMBIA CONSTANTEMENTE (ej.
Noticias).

Trade-off: Exploración y Explotación en recomendación

EXPLORACIÓN:

¿SEGUIMOS
RECOMENDANDO BASADO
EN INFORMACIÓN
HISTÓRICA DEL USUARIO EN
LA PLATAFORMA?

RIESGO: ABURRIMIENTO DE
RECIBIR SIEMPRE LAS
MISMAS
RECOMENDACIONES.



EXPLORACIÓN:

¿RECOMENDAMOS
CONTENIDO DISTINTO A SUS
PREFERENCIAS HISTÓRICAS?

RIESGO: RECOMENDAR UN
CONTENIDO MUY ALEJADO
DE SUS PREFERENCIAS QUE
PUEDE HACERLO SALIR..

Trade-off: Exploración y Explotación en recomendación

Mayor grado de exploración



**EXPLORAR NUEVOS
LOCALES DE COMIDA**

POSIBLE RECOMPENSA
A LARGO PLAZO.
DIVERSIFICACIÓN.

Grado de exploración intermedio



Bajo grado de exploración



**EXPLOTAR LO QUE
SIEMPRE HE VISTO**



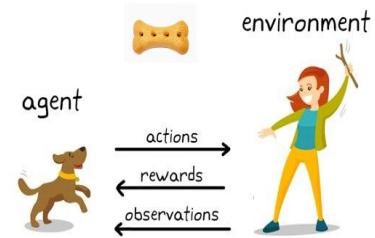
RECOMPENSA DE
CORTO PLAZO
QUE DECAE CON EL
TIEMPO

Balancear exploración y explotación es importante para aprender una forma de tomar mejores decisiones en caso de incertezas.



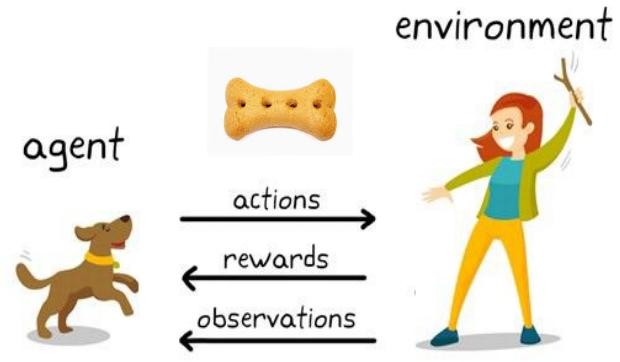
Conceptos claves en Aprendizaje Reforzado (1 / 2)

- **Environment:** es el mundo que reacciona a acciones hechas por agente. Ej Plataforma.
- **State:** parámetros que describen el estado actual. Ej. screen actual de un juego, screen de una página web.
- **Agent:** es el jugador que ejecuta acciones y aprende de ellas al interactuar con el entorno. Ej.
- **Action:** un agente puede seleccionar una acción de un set de acciones.
- **Step:** una vez que el agente hace una acción sobre el entorno el estado cambia a otro nuevo. Este cambio se llama step.



Conceptos claves en aprendizaje reforzado (2 / 2)

- **Reward:** dependiendo de la acción que toma el agente, el entorno le da un premio o un castigo.
- **Target:** lo que el agente busca maximizar al terminar un episodio.
- **Policy:** lo que aprende el modelo donde dado un estado nuevo me predice que acción tomar.
- **Episode:** conjunto de pasos que se define dependiendo del problema, en videojuegos puede ser una partida.



MULTI-ARMED BANDITS.

Acción (at):
¿Qué máquina juego?



1



2



3

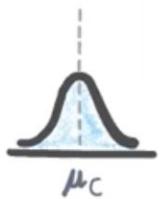
Recompensa (rt):
Dinero \$

Reglas:
Tengo un número limitado de intentos (fichas).

Objetivo:
Maximizar ganancia futura.

MULTI-ARMED BANDITS.

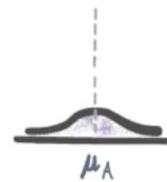
Tenemos cada tragamonedas tiene una distribución de rewards por detrás que **no conozco**.
Tengo que estimar la posible reward de cada máquina con el mínimo de juegos.



1



2



3

Jugar a máquina 1 , 2 o 3 son las K posibles acciones

Reglas

- Número limitado de intentos
- Acción: Puedo jugar a la máquina 1, 2 o 3.
- Feedback se observa solo si juego en la máquina.

Objetivo:

Maximizar mi ganancia

¿Cómo medir el éxito?

Objetivo	Definición
Maximizar suma de reward	Sumar el máximo \$ posible
Maximizar reward promedio	Sumar máximo \$ promedio
Maximizar % de acciones correctas	Escoger la acción que me da mayor ganancia
Disminuir disminuyen arrepentimiento	Regret de tomar malas acciones que mi ganancia.

Veamos un caso práctico



Juguemos al tragamonedas y veamos cómo nos va:

http://apbarraza.com/bandits_activity/

Discusión

¿Qué estrategia tomaron?

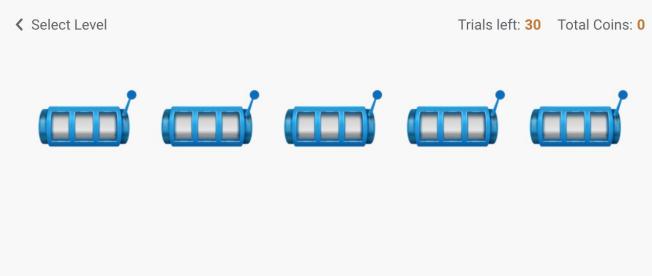
¿Cómo les fué?

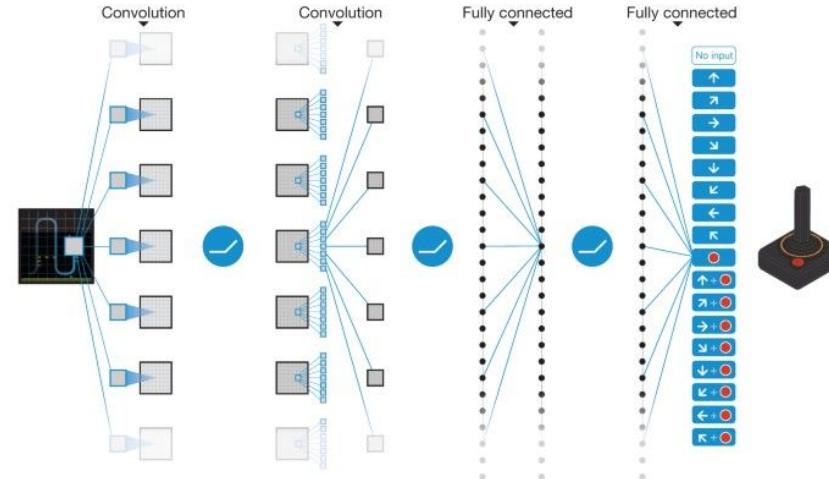
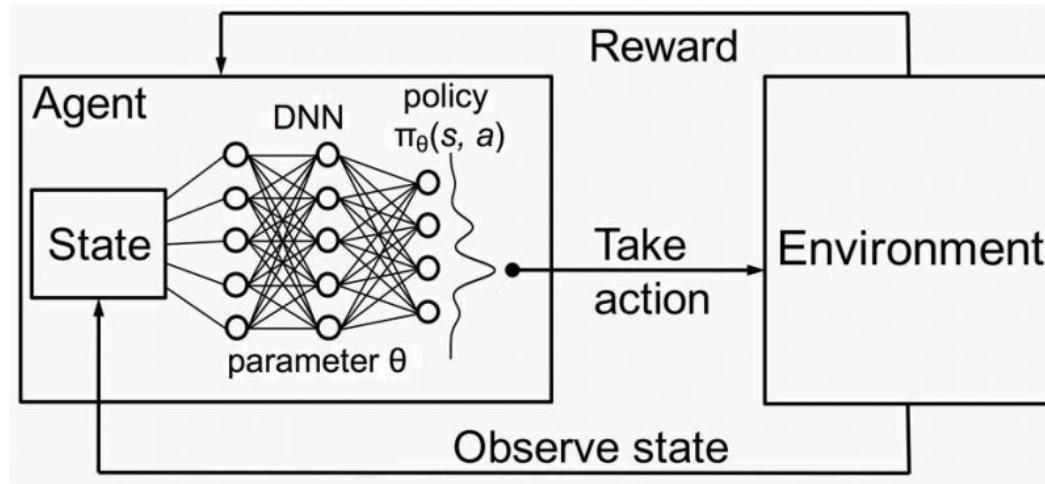
¿Qué hace que el nivel hard hace que sea más difícil que el resto?

Multi-Armed Bandits Activity

Try to get the most coins that you can!

< Select Level Trials left: 30 Total Coins: 0





Redes Neuronales en Aprendizaje Reforzado

Funciones de Valor: las redes neuronales aproximan $V(s)$ y $Q(s,a)$ entregando una estimación del valor acumulado basada en un estado (s) y/o acción.

Funciones de policy: las redes predicen una distribución de acciones basada en un estado.

Modelos de entorno: las redes aprenden cómo el entorno responde a acciones, modelando su dinámica.

Juguemos



MÁQUINA 1

1

\$5.000

2

3

4

5

6

7

8

Juguemos



MÁQUINA 1

1

\$5.000

2

\$ 0

3

4

5

6

7

8

Juguemos



MÁQUINA 1



MÁQUINA 2

1

\$5.000

2

\$ 0

3

\$5.000

4

5

6

7

8

Juguemos



MÁQUINA 1

1 \$5.000

MÁQUINA 2

\$ 0

2 \$5.000

3 \$5.000

4

5

6

7

8

Juguemos



MÁQUINA 1

1 \$5.000

2 \$5.000

3 \$5.000

4 \$0

5 \$0

6 \$0

7 \$0

8 \$0

MÁQUINA 2

\$0

\$0

\$0

Juguemos



MÁQUINA 1

MÁQUINA 2

JUGUEMOS SOLO A LA MÁQUINA 1	1	\$5.000	\$ 0
....	2	\$5.000	\$ 0
	3	\$5.000	\$ 0
	4	\$5.000	\$ 0
	5	\$5.000	\$ 0
	6	\$5.000	
	7	\$5.000	
	8	\$5.000	\$ ganancia total \$ 40.000

Veamos el resultado...



**MÁQUINA 1 PAGA
\$5.000 SIEMPRE**

1

MÁQUINA 1

MÁQUINA 2

\$5.000

\$60.000

**MÁQUINA 2 PAGA
\$60.000 EL 50% DE LAS
VECES
~ \$30.000 POR JUEGO
MAQUINA ÓPTIMO.**

2

\$5.000

\$0

3

\$5.000

\$60.000

4

\$5.000

\$0

5

\$5.000

\$60.000

**ENSEÑANZA:
NOS FALTÓ MÁS
EXPLORACIÓN ANTES DE
QUEDARNOS CON LA
MÁQUINA 1....**

6

\$5.000

\$0

7

\$5.000

\$60.000

8

\$5.000

ganancia potencial: \$40.000

ganancia potencial: \$240.000

si exploro mucho, a corto plazo voy a obtener menos ganancias.

TRADE-OFF:

EXPLORACIÓN / EXPLOTACIÓN

CORTO PLAZO / LARGO PLAZO

explotar: me puede hacer perder la máquina más óptima si yo no tengo la información suficiente para tomar la mejor decisión.

En el [largo plazo](#) puedo perder porque no obtuve la suficiente información.



¿QUÉ ESTRATEGIAS NOS PUEDEN AYUDAR A ESCOGER LA MEJOR ACCIÓN?

- Greedy exploration for n steps
- Epsilon greedy
- Decreasing epsilon greedy
- Upper confidence bound (UCB)

Explorar y luego explotar (greedy exploration)

EXPLORAR

ACCIÓN ALEATORIA

N PASOS DE EXPLORACIÓN

EXPLOTAR

MEJOR ACCIÓN CONOCIDA

[GREEDY ACTION]
ESCOGER LA ACCIÓN CON
VALOR DE RECOMPENSA
ESPERADA MÁS ALTA.

$Q_t(a)$: función a la que nosotros le damos la acción y retorna cual es el valor que creemos que esa acción tendrá en ese momento t.

Forma sencilla de estimarlo (promedio de las rewards observadas para cada acción $Q(0)$, $Q(1)$, $Q(2)$ etc...)

$Q_t(a) = \text{sum(rewards)} / N \text{ pasos exploración}$

$Q_t(a) = \text{reward promedio exploración para acción a}$

Explorar y luego explotar: estrategias

EXPLORAR

ACCIÓN ALEATORIA

N PASOS DE EXPLORACIÓN

Explora cada
acción una vez

Explorar K pasos
 $K = \text{número de acciones}$

Explora acciones
aleatoriamente
durante una
proporción (e) de T

EXPLOTAR

MEJOR ACCIÓN CONOCIDA

[GREEDY ACTION]
ESCOGER LA ACCIÓN QUE
LE VA MEJOR EN ESE
MOMENTO

EXPLORAR
 $T - K$
Pasos

EXPLOTAR
 $T - K * e$
Pasos

T
total
pasos

1

2

Sistemas recomendadores basados en bandits



Pero

¿Qué pasa si cuando exploramos las acciones justo nos fue mal?

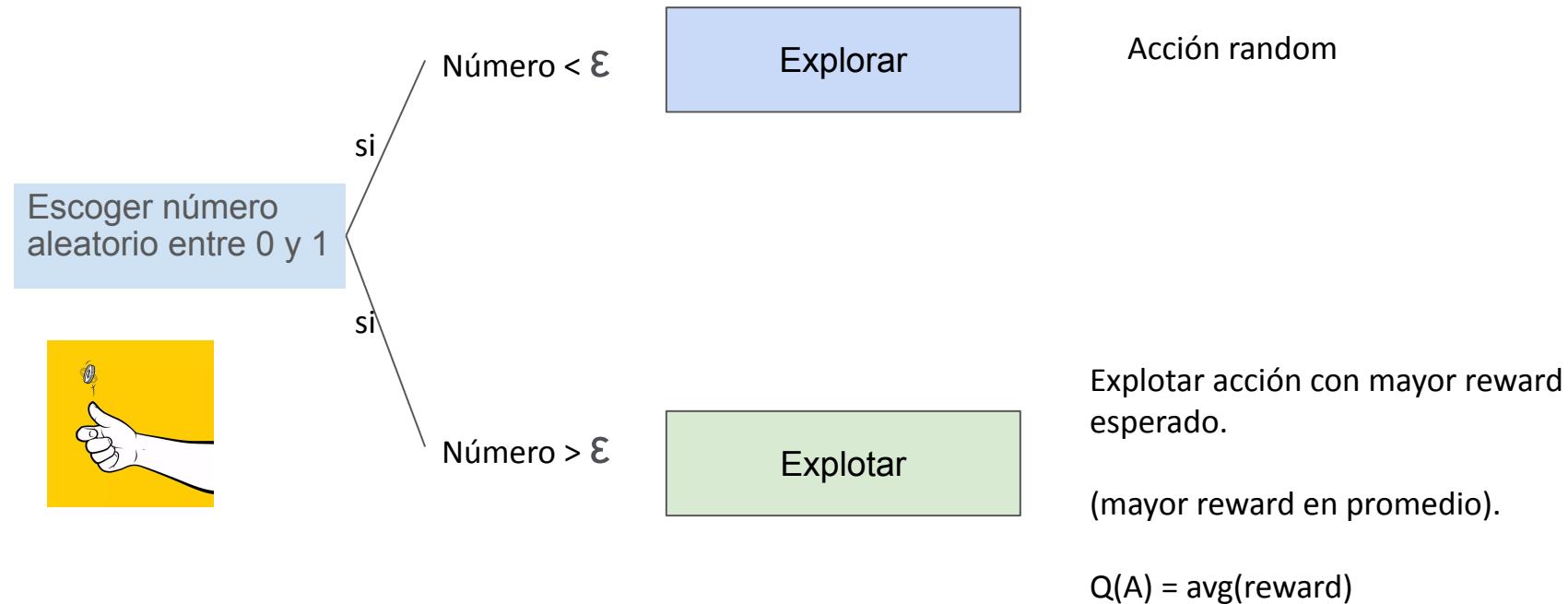
En este caso explotaremos eternamente una acción que no es óptima.

Epsilon Greedy

Un epsilon

- **más grande** implica mayor exploración.
- **más chico** implica mayor explotación.

Tenemos que escoger epsilon (ϵ) entre 0 y 1



Curva de aprendizaje de epsilon greedy

10 posibles acciones

Azul: Epsilon greedy (ϵ =0.1)

Rojo: Epsilon greedy (ϵ =0.01)

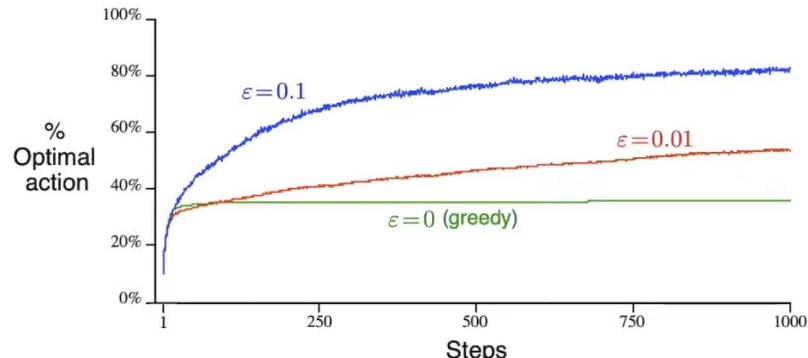
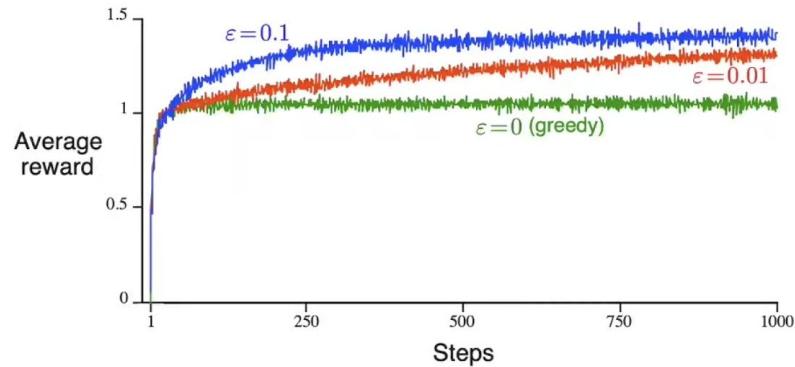
Verde: Epsilon greedy (ϵ =0)

Epsilon mayor (0.1) tiene más chances de explorar más y por ende tiene una recompensa a largo plazo.

Epsilon de 0.01 mejora más lento porque explota mucho y explora pocas veces.

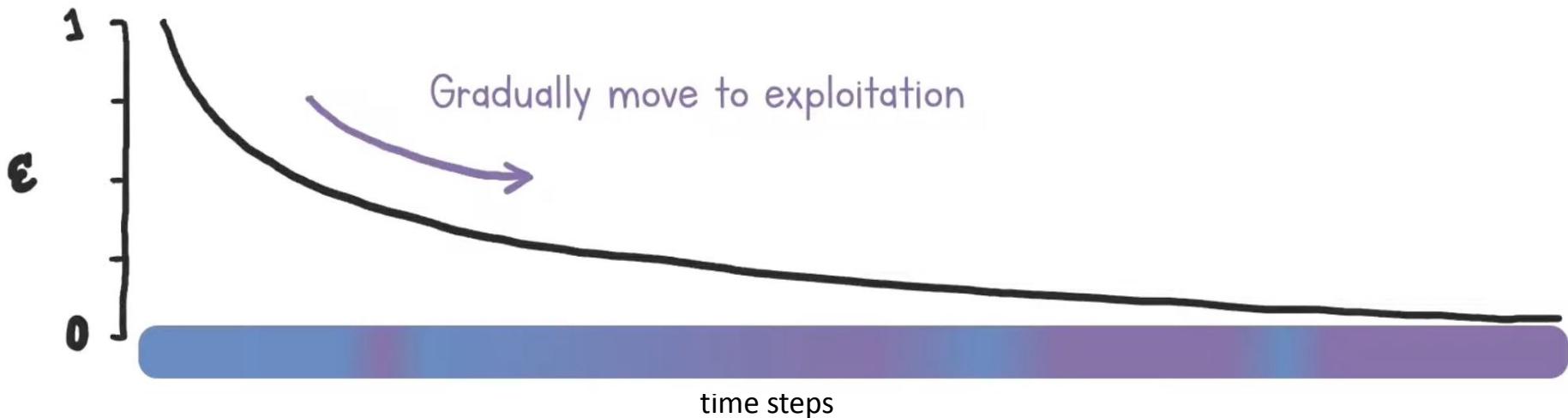
Pero en algún momento el modelo con epsilon de 0.01 va a superar al de 0.1.

Epsilon 0 explota siempre, por lo tanto nunca aprende. Reward queda casi constante.



Estrategia de Epsilon decreciente

valor de epsilon

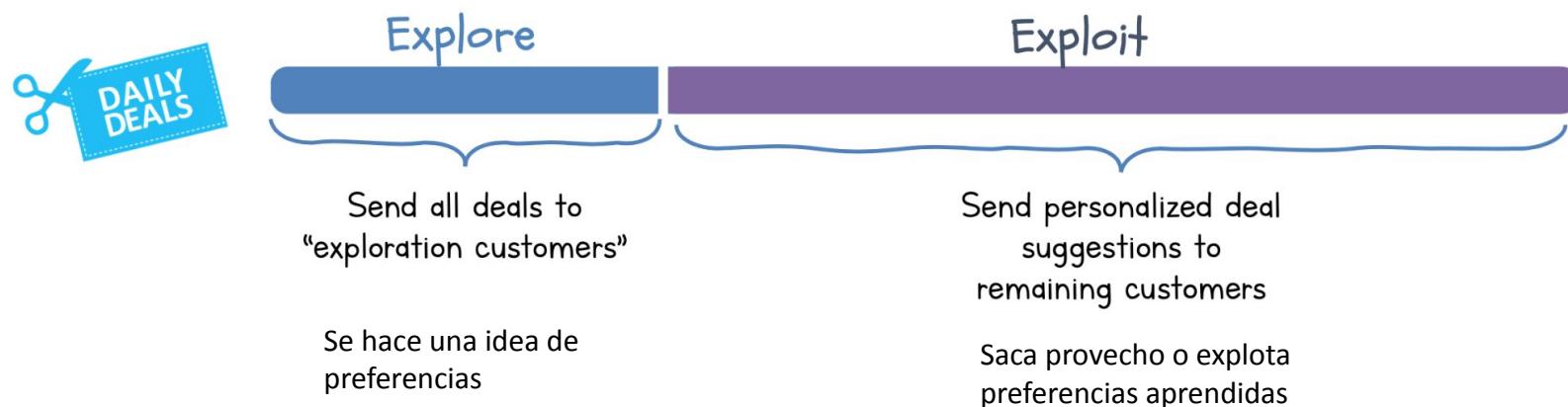


Podemos tomar una estrategia donde:

- Comenzamos con un epsilon grande que explora más.
- Una vez que exploramos lo suficiente lo disminuimos en el tiempo para aumentar la explotación.

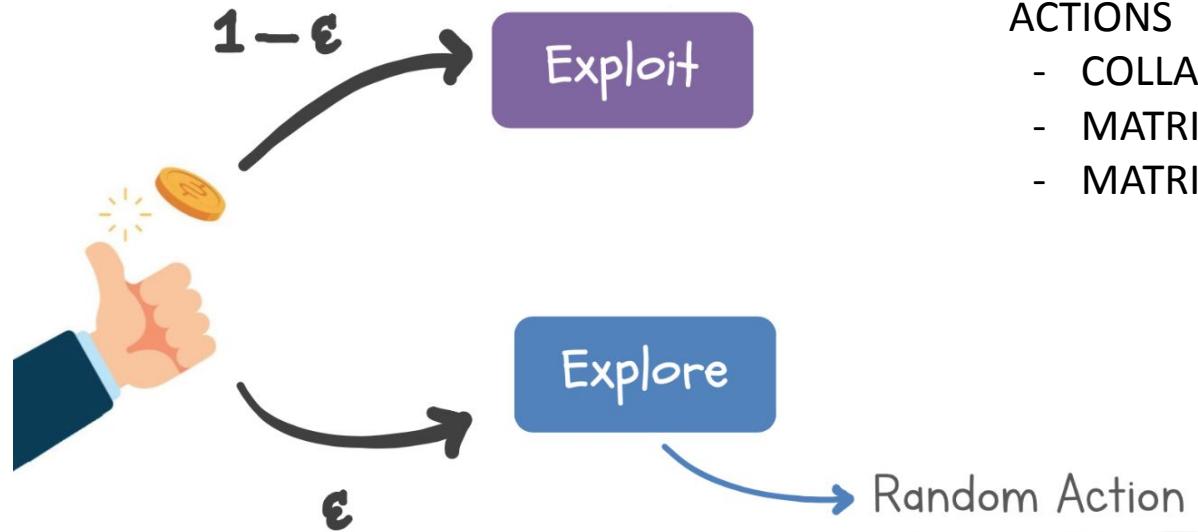
Aplicación 1: Ofertas diarias

Explore then Exploit
meets Recommender Systems (RecSys)



[1] A. Lacerda, R. Santos, et al. 2015. Improving daily deals recommendation using explore-then-exploit strategies.

Aplicación 2: Epsilon Greedy + RecSys



ACTIONS

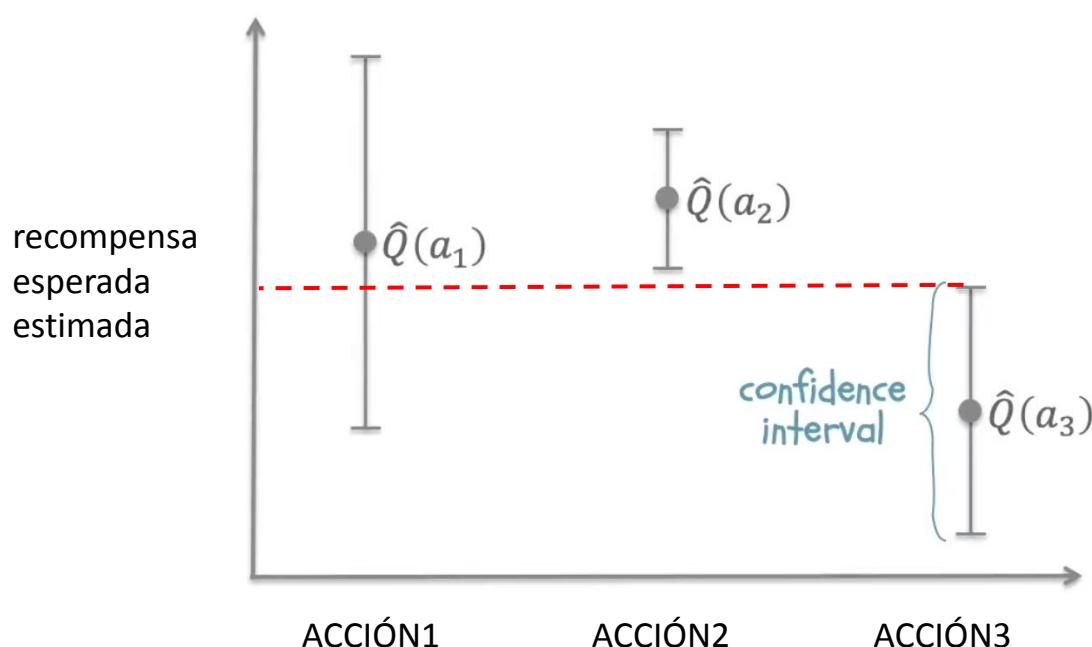
- COLLABORATIVE FILTERING
- MATRIX FACTORIZATION ALS
- MATRIX FACTORIZATION BPR

Upper confidence bound

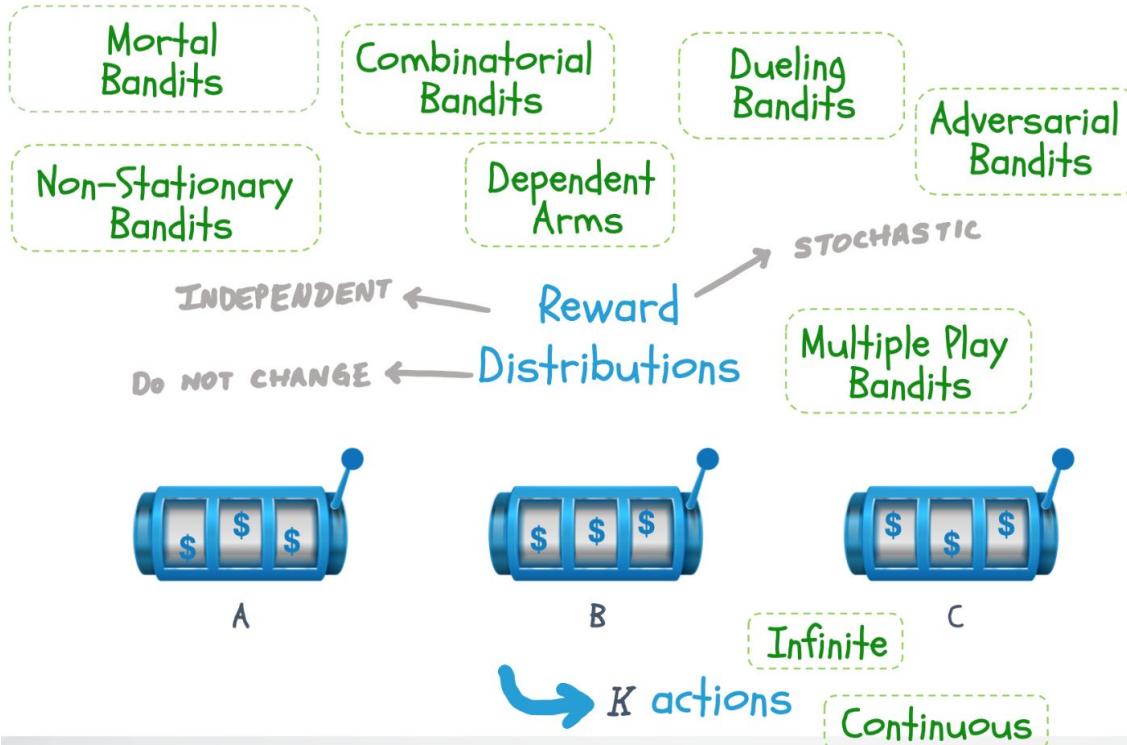
3 posibles acciones (a_1, a_2, a_3)

Acción 3 ya es subóptima porque su intervalo hacia arriba es peor que el máximo intervalo de Acción 1 y 2.

En este caso escogemos la acción 1, que tiene la mayor recompensa esperada en el intervalo de arriba.



Variantes



Variants !

- Maximize total payoff.
- Play for T trials.
- Feedback is observed only for played arm.

Infinite

Full Feedback

and many more...



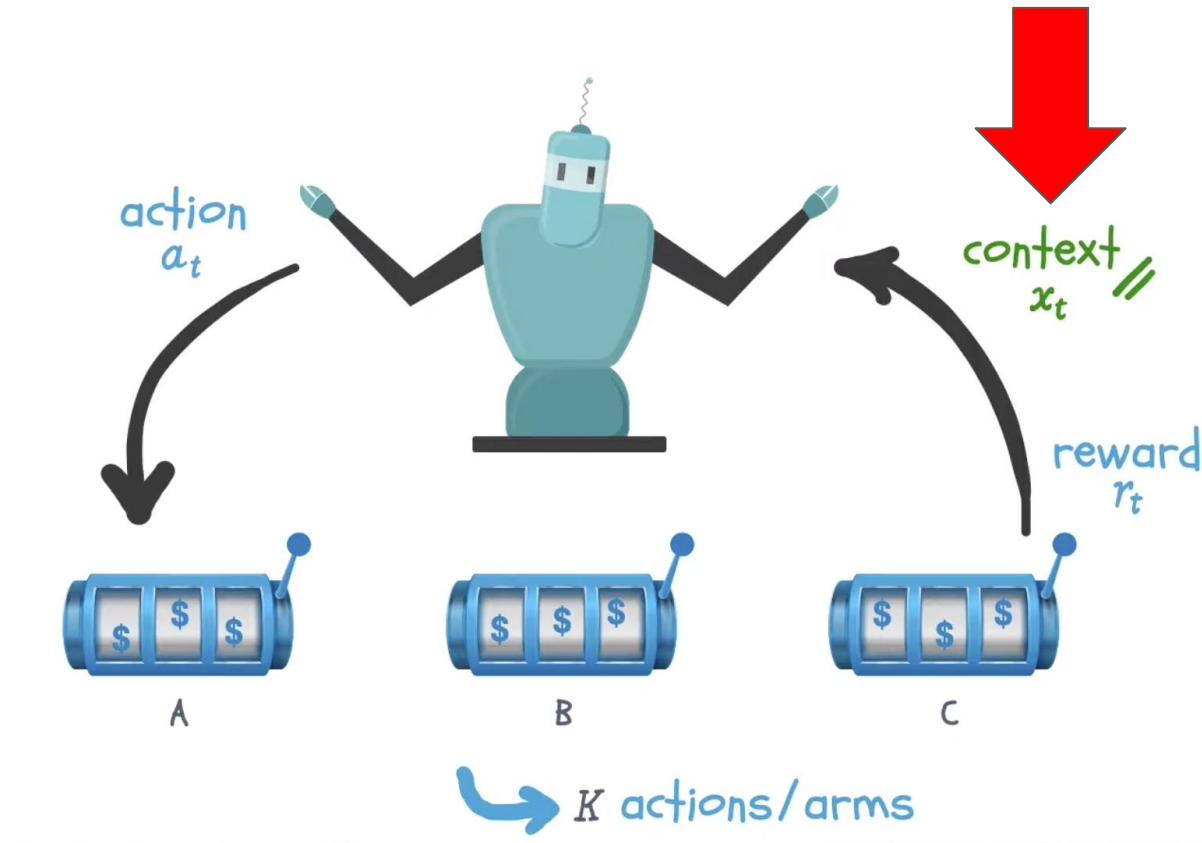
Contextual bandits

En este caso la recompensa depende de:

- la acción
- el contexto

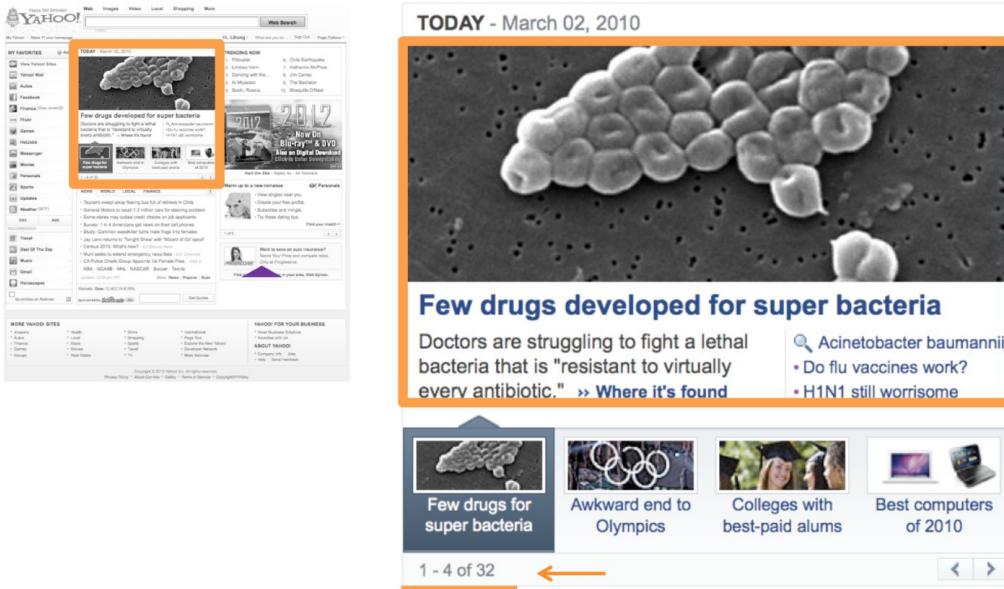
contexto puede ser:

- dia de la semana
- hora
- temporada
- etc...



LinUCB: Caso de Yahoo! Web Page

Goal: Maximize user click feedback



LinUCB
for
Online
News
Recommendation

[1] L. Li, W. Chu, J. Langford, and R. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation.

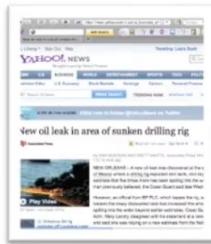
Contextual bandits [LinUCB]

LinUCB: incorpora combinaciones lineales de features de usuarios a las rewards esperadas de los bandits.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3

Feature contextual (edad)



$$x = [1, 0]$$



$$x = [0, 1]$$

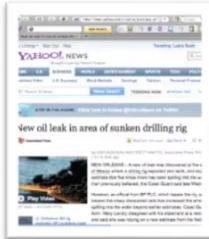
Contextual bandits [LinUCB]

LinUCB: incorpora combinaciones lineales de features de usuarios a las rewards esperadas de los bandits.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3



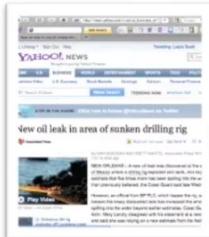
Contextual bandits [LinUCB]

LinUCB: incorpora combinaciones lineales de features de usuarios a las rewards esperadas de los bandits.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3



$$x = [1, 0]$$

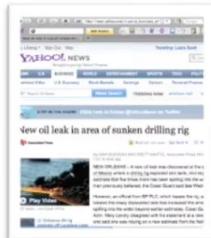
Contextual bandits [LinUCB]

LinUCB: incorpora combinaciones lineales de features de usuarios a las rewards esperadas de los bandits.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3



$x = [0,1]$

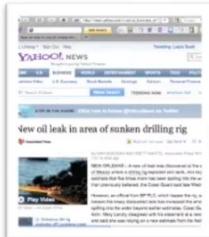
Contextual bandits [LinUCB]

LinUCB: incorpora combinaciones lineales de features de usuarios a las rewards esperadas de los bandits.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3



$$x = [0,1]$$

Contextual bandits [LinUCB]

$$\hat{Q}_t(x_t, a) = x_t^T \cdot \hat{\theta}^a$$

Recompensa esperada (**Q**) depende del **contexto** y de la **acción**.
theta a hat = recompensa esperada para ese segmento de usuarios.
XtT = vector de features
Ahora la recompensa es la combinación lineal de ambos.

Posibles acciones (a_1, a_2, a_3)



a_1



a_2



a_3

Feature contextual (edad)



$x = [1, 0]$



$x = [0, 1]$