Práctica 1: Problema del Bandido de k-brazos.

Fernández Campillo, Pedro José Pujante Sáez, Jaime Sendra Lázaro, Ricardo Javier

8 de marzo de 2025

Resumen

Este documento presenta un estudio exhaustivo del problema del bandido de k-brazos, un modelo fundamental para la toma de decisiones secuencial bajo incertidumbre. Este problema, relevante en áreas como la optimización de publicidad online y la selección de tratamientos médicos, explora el dilema entre **exploración de diferentes opciones y explotación de las que han demostrado ser más rentables**. El objetivo principal del estudio es implementar y comparar el rendimiento de diversos algoritmos de selección de brazos. Para ello, se analizan en detalle las familias de algoritmos ε-greedy, Upper Confidence Bound (UCB) y Métodos de Ascenso del Gradiente, incluyendo sus variantes como UCB1, UCB2 y Softmax, así como el algoritmo de Gradiente de Preferencias. El documento también describe formalmente el problema, los modelos de recompensa utilizados (Bernoulli, Binomial y Normal), y la estructura del informe que incluye la introducción, el desarrollo teórico, la descripción de los algoritmos, la evaluación experimental, las conclusiones y la bibliografía.

La evaluación experimental de los algoritmos se realizó en un entorno simulado con diferentes distribuciones de recompensa y parámetros específicos para cada algoritmo, utilizando el **rechazo acumulado y el porcentaje de selección del brazo óptimo** como métricas principales. Los resultados mostraron que la elección adecuada de los parámetros, como ε en ε-greedy, α en UCB2 y Gradiente de Preferencias, y τ en Softmax, es crucial para equilibrar exploración y explotación. En general, se observó que un valor intermedio de ε (0.1) ofrecía el mejor rendimiento para ε-greedy, mientras que UCB1 mostraba estabilidad, aunque con convergencia más lenta en distribuciones inciertas. UCB2 requería una calibración fina del parámetro α. Finalmente, **el Gradiente de Preferencias con un α de 0.5 demostró ser la estrategia más efectiva** entre los algoritmos basados en gradiente. El estudio concluye resaltando la importancia de seleccionar el algoritmo más apropiado según la naturaleza del problema y las restricciones del entorno para optimizar el rendimiento.

1. Introducción

En esta práctica, analizamos el problema del bandido de k brazos, que modela una situación secuencial de toma de decisiones bajo incertidumbre. En este contexto, un agente debe seleccionar entre k opciones distintas (brazos), cada una con una recompensa desconocida, con el objetivo de maximizar la recompensa acumulada a lo largo del tiempo. Este problema es un caso representativo del dilema exploración-explotación, siendo relevante en aplicaciones como la optimización de la publicidad online, la selección de tratamientos médicos y la gestión de inventarios. En este sentido, el estudio de los algoritmos k-arm bandit es fundamental en el aprendizaje por refuerzo y la toma de decisiones automatizada. La implementación y comparación de distintas estrategias de selección de brazos nos permite comprender mejor el equilibrio entre exploración y explotación, así como evaluar qué enfoques son más eficaces en distintos entornos. Además, la capacidad de optimizar decisiones secuenciales tiene aplicaciones prácticas en múltiples ámbitos, desde la economía a la inteligencia artificial. Para ello, nuestro principal objetivo es implementar y comparar el rendimiento de distintos algoritmos de selección de brazos. Para ello, realizamos un análisis exhaustivo de la familia ε-greedy y exploramos otras estrategias avanzadas, como los métodos UCB y el gradient ascent. Evaluamos estas técnicas estudiando el rechazo acumulado(*regret*) y el porcentaje de elecciones óptimas de brazos, identificando los algoritmos más eficientes para cada tipo de distribución de recompensas. Por ultimo cabe destacar la estructura de este informe sigue un esquema claro para presentar nuestro estudio de manera comprensible y detallada:

1. **Introducción:** Se presenta una visión general del problema, su relevancia, la motivación del estudio y los objetivos del trabajo.
2. **Desarrollo:** Se expone la definición formal del problema, se describen enfoques existentes en la literatura, y se presentan los antecedentes necesarios. Además, se explican los métodos utilizados para abordar el problema y se justifica cómo nuestra implementación mejora o complementa los enfoques previos.
3. **Algoritmos:** Se detallan los pasos de los algoritmos empleados, incluyendo pseudocódigo y diagramas de flujo cuando sea necesario. También se justifica la elección de los algoritmos y su aplicabilidad al problema estudiado.
4. **Evaluación y Experimentos:** Se describe la configuración experimental, detallando herramientas utilizadas, entornos de prueba y datasets empleados. Presentamos los resultados obtenidos en tablas y gráficos relevantes, y realizamos un análisis crítico comparativo con otros enfoques.
5. **Conclusiones:** Se resumen los principales hallazgos del estudio, se discuten sus limitaciones y se proponen mejoras futuras. Asimismo, se reflexiona sobre la importancia del trabajo y su impacto en el campo del aprendizaje por refuerzo.
6. **Bibliografía:** Se listan todas las referencias utilizadas en el trabajo, utilizando BibTeX como gestor de citas para una mejor organización.

2. Desarrollo

2.1. Definición formal del problema

El problema del bandido de k-brazos es un modelo fundamental en la toma de decisiones secuencial bajo incertidumbre. Se enmarca dentro del aprendizaje por refuerzo y permite estudiar el dilema de exploración-explotación en su forma más simple [5].

Formalmente, el problema se define como sigue:

- En cada paso de tiempo $t = 1, 2, \ldots, T$, un agente debe seleccionar una acción (o brazo) $a_t \in \mathcal{A}$, donde $\mathcal{A} = \{1, 2, \ldots, k\}$ representa el conjunto de brazos disponibles.
- Cada brazo i está asociado a una distribución de recompensa desconocida $\mathcal{P}_i$ con media $\mu_i = \mathbb{E}[r_t | a_t = i]$.
- Tras elegir un brazo a_t, el agente recibe una recompensa r_t, obtenida como una realización de la distribución $\mathcal{P}_{a_t}$.
- El objetivo del agente es maximizar la recompensa acumulada definida en la Ecuación (1).

$$R(T) = \sum_{t=1}^{T} r_t. \tag{1}$$

Dado que las distribuciones de recompensa son desconocidas, el agente debe balancear dos estrategias opuestas:

- **Exploración:** Probar distintos brazos para estimar sus recompensas esperadas.
- **Explotación:** Seleccionar los brazos que hasta el momento han mostrado mejor rendimiento.

El rendimiento de un algoritmo de selección de brazos suele evaluarse mediante el **rechazo** (*regret*), que mide la pérdida incurrida por no haber elegido siempre el mejor brazo. Se define en la Ecuación (2) como la diferencia entre la recompensa óptima y la recompensa acumulada por la estrategia del agente:

$$R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right], \tag{2}$$

donde $\mu^* = \max_i \mu_i$ es la recompensa esperada del mejor brazo. Un buen algoritmo minimizará el rechazo a medida que crezca el horizonte temporal T.

El problema del bandido de k brazos se considera el caso más sencillo del aprendizaje por refuerzo, ya que carece de estados y transiciones dinámicas. Sin embargo, constituye la base de algoritmos más complejos utilizados en problemas de planificación secuencial y toma de decisiones [5].

2.1.1. Modelado de las recompensas: Distribuciones de probabilidad

En nuestro estudio, consideramos que las recompensas de cada brazo siguen una de tres distribuciones estadísticas: **Bernoulli, Binomial y Normal**. Estas distribuciones representan distintos tipos de problemas y permiten evaluar el desempeño de los algoritmos bajo diferentes condiciones.

Distribución de Bernoulli

La distribución de Bernoulli es el caso más simple de modelado de recompensas. Cada brazo i tiene una probabilidad de éxito $p_i \in [0, 1]$ y genera recompensas binarias:

$$r_t \sim \text{Bernoulli}(p_i), \tag{3}$$

donde la recompensa toma los valores $r_t \in \{0, 1\}$ con probabilidad p_i y $1 - p_i$, respectivamente. Este modelo es útil en problemas donde cada acción solo puede tener éxito o fracaso, siguiendo el ejemplo del enunciado podría ser la tasa de clics en publicidad digital [2].

Distribución Binomial

La distribución binomial generaliza la de Bernoulli considerando múltiples intentos n. Un brazo i con probabilidad de éxito p_i genera recompensas siguiendo:

$$r_t \sim \text{Binomial}(n, p_i), \tag{4}$$

donde la recompensa r_t representa el número de éxitos obtenidos en n ensayos independientes con probabilidad p_i. Este modelo se aplica en contextos que como siguiendo el ejemplo del enunciado podría ser el numero de veces que se accede a una pagina web en funcion de distintas plataformas de anuncios de esta[3].

Distribución Normal

Para modelar recompensas continuas, utilizamos la distribución Normal:

$$r_t \sim \mathcal{N}(\mu_i, \sigma_i^2), \tag{5}$$

donde cada brazo i tiene una media μ_i y una varianza σ_i^2. Esta distribución es relevante en problemas donde las recompensas fluctúan alrededor de un valor esperado, como por ejemplo el tiempo de respuesta de un sistema [4].

2.2. Soluciones al problema

Para abordar el problema del bandido de k-brazos, implementamos, entre otras muchas, tres familias de algoritmos: ε-greedy, Upper Confidence Bound (UCB) y Métodos de Ascenso del Gradiente.

2.2.1. ε-greedy

El método ε-greedy equilibra exploración y explotación mediante una estrategia probabilística:

- Con probabilidad ε, elige un brazo al azar (exploración).
- Con probabilidad $1 - \varepsilon$, selecciona el brazo con la mayor recompensa promedio observada (explotación).

El valor de ε determina el grado de exploración y típicamente se ajusta dinámicamente para reducir la exploración a medida que se obtiene más información [5].

2.2.2. Upper Confidence Bound (UCB)

UCB se basa en el principio de optimismo en el manejo de la incertidumbre. Se calcula un valor de confianza superior para cada brazo, combinando la recompensa promedio con un término de incertidumbre:

$$Q_i + c\sqrt{\frac{\ln t}{N_i}}, \tag{6}$$

donde Q_i es la recompensa media del brazo i, N_i es el número de veces que ha sido seleccionado, y c es un parámetro de ajuste [5].

2.2.3. Métodos de Ascenso del Gradiente

Los métodos de ascenso del gradiente modelan las probabilidades de selección de cada brazo mediante una función de preferencia:

$$H_i \leftarrow H_i + \alpha(r_t - \bar{R})(1 - P_i), \tag{7}$$

donde H_i es la preferencia del brazo i, α es la tasa de aprendizaje, $\bar{R}$ es la recompensa media, y P_i es la probabilidad de seleccionar el brazo i[5].

3. Algoritmos

En esta sección, describimos en detalle los algoritmos implementados para abordar el problema del bandido de k-brazos. Presentamos el pseudocódigo correspondiente y justificamos la elección de cada estrategia en términos de su aplicabilidad y rendimiento.

3.1. ε-greedy

El algoritmo ε-greedy selecciona un brazo de manera aleatoria con probabilidad ε (exploración) y elige el brazo con la mayor recompensa promedio observada con probabilidad $1 - \varepsilon$ (explotación). Su implementación se detalla en el *Algoritmo 1*.

Algorithm 1 ε-greedy Algorithm

Require: Number of arms k, exploration probability ε
Initialize $Q_i = 0$ and $N_i = 0$ for all arms $i \in \{1, ..., k\}$
for each step t **do**
  Generate a random number $p \sim U(0, 1)$
  if $p < \varepsilon$ **then** ▷ Exploration
    Select a random arm a_t
  else ▷ Exploitation
    Select arm $a_t = \arg\max_i Q_i$
  end if
  Observe reward r_t
  Update counts: $N_{a_t} = N_{a_t} + 1$
  Update value estimate:

$$Q_{a_t} = Q_{a_t} + \frac{1}{N_{a_t}}(r_t - Q_{a_t}) \tag{8}$$

end for

3.2. Upper Confidence Bound (UCB1)

El algoritmo **UCB1** favorece la exploración temprana seleccionando brazos con mayor incertidumbre en sus estimaciones de recompensa. La ecuación (9) define el criterio de selección de cada brazo, considerando la media de recompensas observadas y un término de incertidumbre.

Algorithm 2 UCB1 Algorithm

Require: Number of arms k
Initialize $Q_i = 0$ and $N_i = 0$ for all arms i
Set total selection count $t = 0$
for each step t **do**
  if $t < k$ **then** ▷ Primero selecciona cada brazo al menos una vez
    Select arm $a_t = t$
  else
    for each arm i **do**
      Compute UCB value:

$$UCB_i = Q_i + \sqrt{\frac{2\ln t}{N_i}} \tag{9}$$

    end for
    Select arm $a_t = \arg\max_i UCB_i$
  end if
  Observe reward r_t
  Update the selected arm using:

$$Q_{a_t} = Q_{a_t} + \frac{1}{N_{a_t}}(r_t - Q_{a_t}) \tag{10}$$

  Increment selection count: $t = t + 1$
end for

3.3. UCB2: Mejora sobre UCB1

El algoritmo **UCB2** es una variante de **UCB1** que introduce un parámetro α para mejorar la exploración a largo plazo. En lugar de seleccionar brazos en cada iteración, **UCB2** divide la exploración en fases más largas para estabilizar la selección de brazos. La ecuación (11) define el criterio de selección en **UCB2**.

El algoritmo **UCB2** mejora la eficiencia exploratoria al controlar cuándo se vuelven a evaluar los brazos, evitando cambios demasiado frecuentes entre exploración y explotación [5].

3.4. Softmax

El algoritmo **Softmax** es una estrategia basada en asignar probabilidades de selección a cada brazo según una función exponencial de sus estimaciones de recompensa. Esto permite una exploración más controlada en comparación con métodos como ε-greedy, donde la exploración es completamente aleatoria.

La probabilidad de seleccionar un brazo i se define mediante la ecuación (14):

$$P_i = \frac{e^{Q_i/\tau}}{\sum_{j=1}^{k} e^{Q_j/\tau}} \tag{14}$$

Algorithm 3 UCB2 Algorithm

Require: Number of arms k, exploration parameter α

Initialize $Q_i = 0$ and $N_i = 0$ for all arms i
Set $t = 0$
Initialize exploration thresholds $r_i = 0$ for all arms i
for each step t **do**
  for each arm i **do**
    if $N_i < r_i$ **then** ▷ Asegura que cada brazo se pruebe según su fase de exploración
      Select arm $a_t = i$
      break
    end if
  end for
  if no arm was selected **then**
    for each arm i **do**
      Compute UCB2 value:

$$UCB2_i = Q_i + \sqrt{\frac{(1+\alpha)\ln(e \cdot N_i)}{2N_i}} \tag{11}$$

    end for
    Select arm $a_t = \arg\max_i UCB2_i$
  end if
  Observe reward r_t
  Update selected arm using:

$$Q_{a_t} = Q_{a_t} + \frac{1}{N_{a_t}}(r_t - Q_{a_t}) \tag{12}$$

  Update exploration threshold:

$$r_{a_t} = N_{a_t}(1+\alpha) \tag{13}$$

  Increment t
end for

donde τ es un parámetro de temperatura que controla el nivel de aleatoriedad. Valores altos de τ hacen que la selección de brazos sea más uniforme, mientras que valores bajos favorecen la explotación.

Algorithm 4 Softmax Algorithm

Require: Number of arms k, temperature parameter τ

Initialize $Q_i = 0$ and $N_i = 0$ for all arms i
for each step t **do**
  Compute selection probabilities:

$$P_i = \frac{e^{Q_i/\tau}}{\sum_{j=1}^{k} e^{Q_j/\tau}} \tag{15}$$

  Select arm a_t using probability distribution P
  Observe reward r_t
  Update selected arm using:

$$Q_{a_t} = Q_{a_t} + \frac{1}{N_{a_t}}(r_t - Q_{a_t}) \tag{16}$$

  Increment t
end for

El método Softmax es útil en entornos donde la recompensa de los brazos puede variar gradualmente, ya que evita cambios abruptos en la selección de acciones.

3.5. Gradiente de Preferencias

El algoritmo **Gradiente de Preferencias** es un método basado en aprendizaje de preferencias. En lugar de actualizar estimaciones de recompensa, ajusta directamente las preferencias de cada brazo usando la diferencia entre la recompensa recibida y la recompensa media observada.

La actualización de preferencias se define mediante la ecuación (17):

$$H_i \leftarrow H_i + \alpha(r_t - \bar{R})(1 - P_i), \tag{17}$$

donde H_i es la preferencia del brazo i, α es la tasa de aprendizaje, $\bar{R}$ es la recompensa promedio, y P_i es la probabilidad de selección del brazo.

Este método es particularmente útil en entornos no estacionarios donde las recompensas cambian con el tiempo, ya que ajusta dinámicamente las probabilidades de selección sin basarse en recompensas fijas [5].

4. Evaluación y Experimentos

En esta sección, describimos la configuración experimental utilizada para evaluar los algoritmos implementados. Presentamos las métricas de evaluación, los resultados obtenidos y un análisis crítico de los mismos.

4.1. Configuración experimental

Para evaluar el desempeño de los algoritmos de selección de brazos, realizamos experimentos en un entorno simulado. Implementamos tres variantes del problema del bandido de k-

Algorithm 5 Gradient Bandit Algorithm

Require: Number of arms k, learning rate α
Initialize preferences $H_i = 0$ for all arms i
Set average reward $\bar{R} = 0$
for each step t **do**
    Compute action probabilities using softmax:

$$P_i = \frac{e^{H_i}}{\sum_{j=1}^{k} e^{H_j}} \tag{18}$$

    Select arm a_t using probability distribution P
    Observe reward r_t
    Update baseline reward:

$$\bar{R} = \bar{R} + \frac{1}{t}(r_t - \bar{R}) \tag{19}$$

    Update preferences:
    for each arm i **do**
        if $i = a_t$ **then**
            $H_i \leftarrow H_i + \alpha(r_t - \bar{R})(1 - P_i)$
        else
            $H_i \leftarrow H_i - \alpha(r_t - \bar{R})P_i$
        end if
    end for
end for

brazos, cada una con una distribución de recompensas diferentes sigiendo lo mencionado en la Sección 2.1.1:

- **Bandido con distribución Normal**
- **Bandido con distribución Binomial**
- **Bandido con distribución Bernoulli**

Cada configuración de bandido se evaluó con diferentes algoritmos:

- **ε-greedy**: Se probaron tres valores de ε: 0,5, 0,1 y 0,01.
- **UCB1**: No requiere parámetros adicionales.
- **UCB2**: Se probó con valores de $\alpha = 0{,}05$ y $\alpha = 0{,}5$.
- **SoftMax**: Se probó valores de $\tau = 1$ y $\tau = 0{,}5$
- **Gradiente de Preferencias**: Se probó con valores de $\alpha = 0{,}01$ y $\alpha = 0{,}5$.

Los experimentos se realizaron con los siguientes parámetros:

- Número de brazos: $k = 10$.
- Número de pasos por ejecución: 1000.
- Número de ejecuciones por experimento: 500.

4.2. Métricas de evaluación

Para analizar el rendimiento de los algoritmos evaluados, hemos seleccionado dos métricas principales: el *rechazo acumulado* y *el porcentaje de selección del brazo óptimo*. Estas métricas nos permiten evaluar eficazmente la eficiencia de cada estrategia en términos de exploración y explotación.

El **rechazo acumulativo** (*regret*) es la métrica fundamental en problemas de toma de decisiones secuenciales, ya que cuantifica la pérdida de recompensa en comparación con una estrategia óptima. Su análisis permite identificar qué algoritmos tardan más en encontrar el mejor brazo y, por tanto, desperdician más recursos en el proceso. Además, su interpretación es directa: una curva ascendente más lenta indica un algoritmo más eficiente en la explotación del mejor brazo [5].

Por otro lado, el **porcentaje de selección del brazo óptimo** complementa el análisis mostrando la frecuencia con la que el algoritmo elige la mejor opción a lo largo del tiempo. Esto nos ayuda a comprender la rapidez con la que un método converge a una estrategia óptima y lo coherente que es en su selección. Si un algoritmo tiene un alto índice de selección del brazo óptimo, significa que su exploración inicial fue eficiente y consiguió estabilizarse en la mejor decisión.

Se consideraron otras métricas, como la **recompensa media** y la **distribución de selecciones**, pero se descartaron en este análisis. La recompensa media es menos útil para evaluar las oportunidades perdidas, ya que esta información puede deducirse en gran medida del gráfico de rechazo acumulativo. Además, la distribución de selecciones no nos permite diagnosticar si el algoritmo mejora o empeora su rendimiento con el tiempo. Por estas razones, nos centraremos exclusivamente en el rechazo acumulado y en la selección óptima de brazos para evaluar el rendimiento del algoritmo. Las metricas escogidas pueden verse como:

- **Rechazo acumulado** (*Regret*): Cuantifica la diferencia entre la recompensa acumulada del mejor brazo y la obtenida por el algoritmo:

$$R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right], \tag{20}$$

  donde $\mu^* = \max_i \mu_i$ representa la media de la mejor recompensa posible.
- **Frecuencia de selección del brazo óptimo**: Porcentaje de veces que el algoritmo seleccionó el brazo con mayor recompensa esperada en cada paso de tiempo.

Todas las métricas anteriores se evaluaron de forma gráfica en función de cada paso de tiempo. Por lo tanto se obtuvieron 6 gráficas por cada uno de los distintos algoritmos (2 graficas por problema)

4.3. Resultados obtenidos

4.3.1. Resultados del ε-greedy

Para evaluar el desempeño del algoritmo ε-greedy, realizamos experimentos con tres valores de ε: 0,5, 0,1 y 0,01. A continuación, se presentan los resultados obtenidos para cada tipo de bandido.

Rechazo acumulado

En las siguientes figuras se muestra la evolución del rechazo acumulado (*regret*) a lo largo del tiempo para cada distribución de recompensa Figuras 1,2,3.

Porcentaje de selección del brazo óptimo

En las siguientes figuras se muestra el porcentaje de veces que el algoritmo seleccionó el brazo óptimo en función del tiempo Figuras 4,5,6.

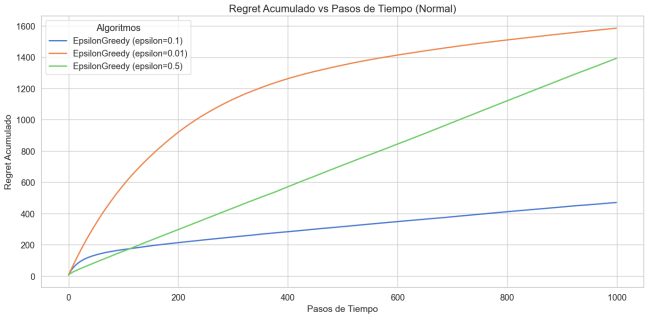

Figura 1: Evolución del rechazo acumulado con distribución Normal.

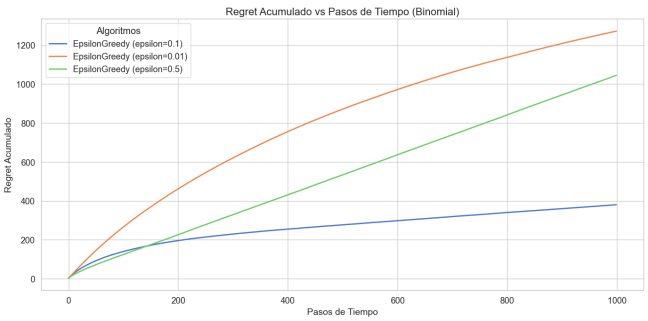

Figura 2: Evolución del rechazo acumulado con distribución Binomial.

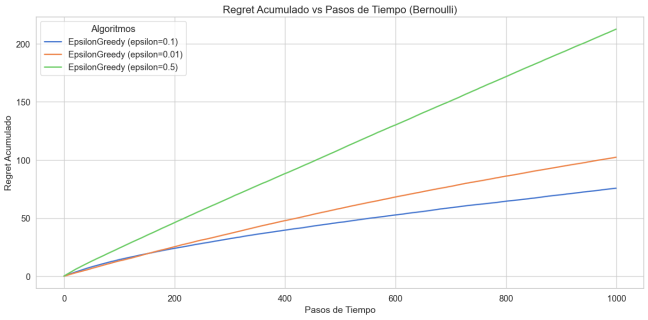

Figura 3: Evolución del rechazo acumulado con distribución Bernoulli.

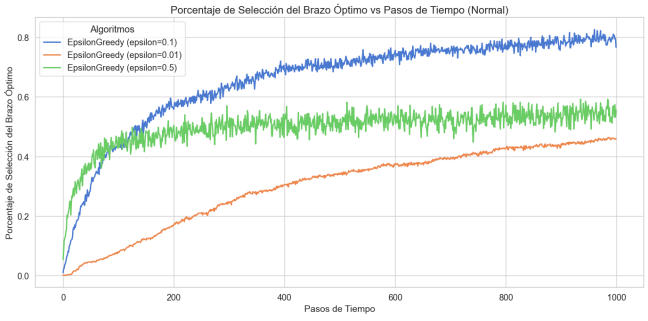

Figura 4: Frecuencia de selección del brazo óptimo con distribución Normal.

4.3.2. UCB1 y UCB2

En esta sección, analizamos el desempeño de los algoritmos **Upper Confidence Bound** (UCB1 y UCB2). Mientras que UCB1 no requiere parámetros adicionales, UCB2 introduce el parámetro α, el cual controla la frecuencia con la que un brazo

Figura 5: Frecuencia de selección del brazo óptimo con distribución Binomial.

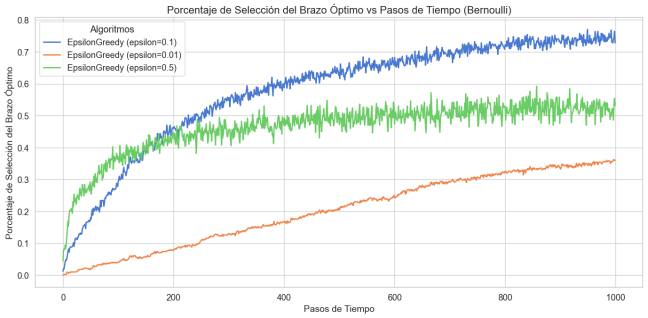

Figura 6: Frecuencia de selección del brazo óptimo con distribución Bernoulli.

es reevaluado. Evaluamos UCB2 con dos valores distintos de α: 0,05 y 0,5.

Rechazo acumulado

En las siguientes figuras se muestra la evolución del rechazo acumulado (*regret*) a lo largo del tiempo para cada distribución de recompensa Figuras 7,8,9.

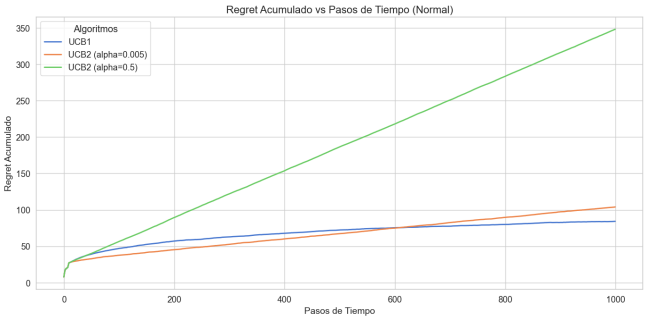

Figura 7: Evolución del rechazo acumulado con UCB en la distribución Normal.

Porcentaje de selección del brazo óptimo

En las siguientes figuras se muestra el porcentaje de veces que el algoritmo seleccionó el brazo óptimo en función del tiempo Figuras 10,11,12.

4.3.3. Resultados de Softmax y Gradiente de Preferencias

En esta sección analizamos el desempeño de los algoritmos basados en métodos de gradiente: *Softmax* y *Gradiente de Preferencias*. A diferencia de ε-greedy y UCB, estos algoritmos ajustan las probabilidades de selección de cada brazo de manera adaptativa, favoreciendo decisiones más flexibles a lo

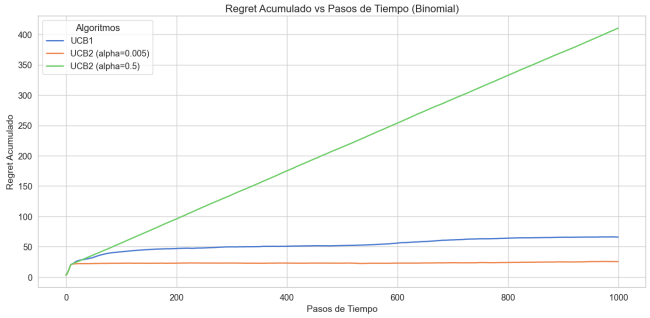

Figura 8: Evolución del rechazo acumulado con UCB en la distribución Binomial.

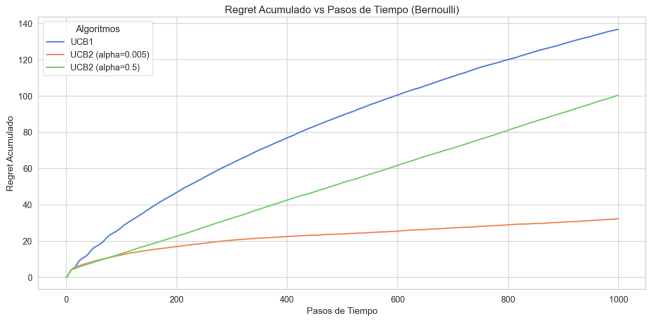

Figura 9: Evolución del rechazo acumulado con UCB en la distribución Bernoulli.

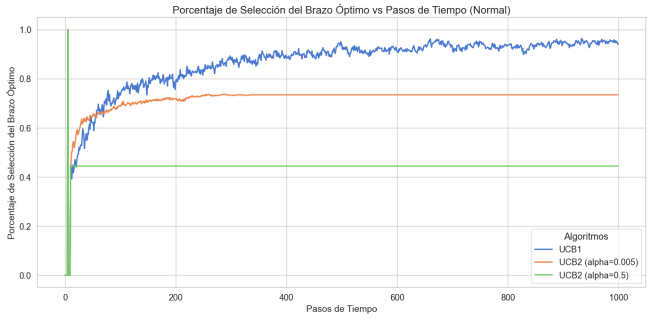

Figura 10: Frecuencia de selección del brazo óptimo con UCB en la distribución Normal.

Figura 11: Frecuencia de selección del brazo óptimo con UCB en la distribución Binomial.

largo del tiempo.

Rechazo acumulado

En las siguientes figuras se muestra la evolución del rechazo acumulado (*regret*) a lo largo del tiempo para cada distribución de recompensa Figuras 13,14,15.

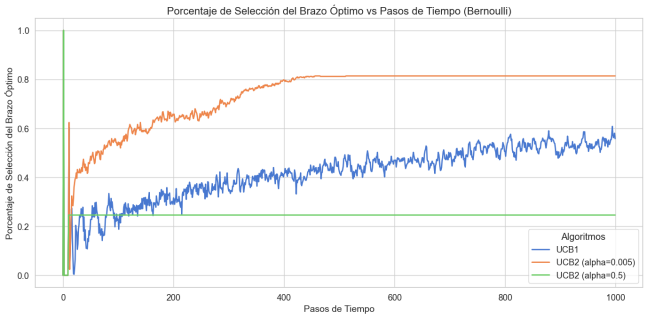

Figura 12: Frecuencia de selección del brazo óptimo con UCB en la distribución Bernoulli.

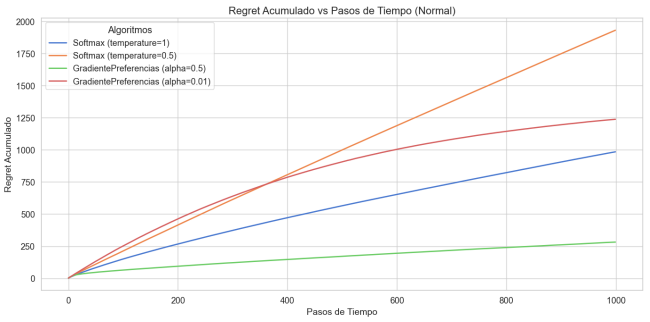

Figura 13: Evolución del rechazo acumulado con Softmax y Gradiente de Preferencias en la distribución Normal.

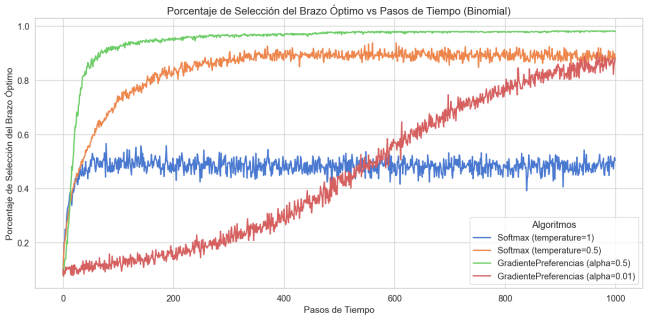

Figura 14: Evolución del rechazo acumulado con Softmax y Gradiente de Preferencias en la distribución Binomial.

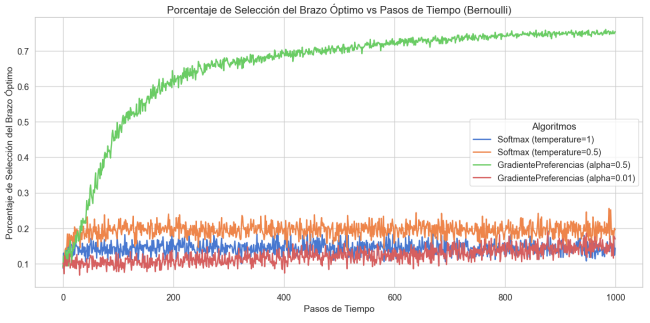

Figura 15: Evolución del rechazo acumulado con Softmax y Gradiente de Preferencias en la distribución Bernoulli.

4.3.4. Porcentaje de selección del brazo óptimo

A continuación, se muestra el porcentaje de veces que Softmax y Gradiente de Preferencias seleccionaron el brazo óptimo en función del tiempo Figuras 16,17,18.

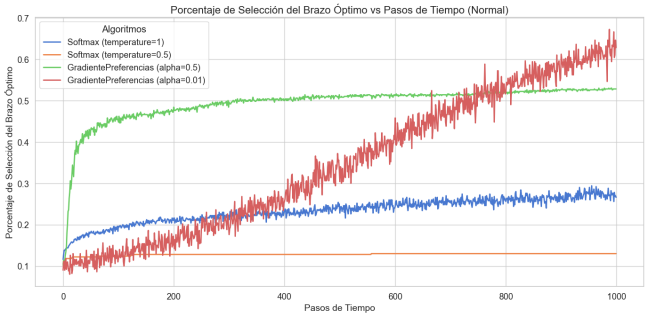

Figura 16: Frecuencia de selección del brazo óptimo con Softmax y Gradiente de Preferencias en la distribución Normal.

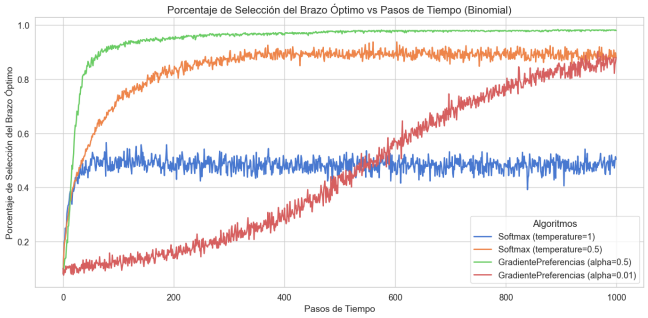

Figura 17: Frecuencia de selección del brazo óptimo con Softmax y Gradiente de Preferencias en la distribución Binomial.

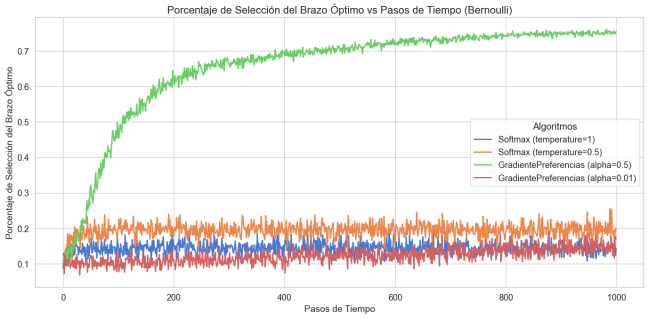

Figura 18: Frecuencia de selección del brazo óptimo con Softmax y Gradiente de Preferencias en la distribución Bernoulli.

4.4. Análisis de los resultados

4.4.1. ε-greedy

Los resultados obtenidos para ε-greedy muestran diferencias significativas en el desempeño de cada variante de ε, lo que confirma la importancia de elegir un valor adecuado de exploración.

Los valores altos de ε (como 0,5) resultan en un rechazo acumulado que crece de manera estable a lo largo del tiempo. Esto indica que la estrategia de exploración excesiva impide al algoritmo consolidarse en el mejor brazo, ya que sigue explorando incluso cuando ha identificado una opción claramente superior. Como consecuencia, el algoritmo no logra explotar de manera efectiva las mejores decisiones, lo que genera un desempeño subóptimo en todos los entornos evaluados.

Por otro lado, los valores bajos de ε (como 0,01) muestran un comportamiento diferente. En este caso, el rechazo acumulado se dispara rápidamente en las primeras etapas del experimento, lo que indica que el algoritmo sobreexplota brazos subóptimos desde el inicio. Esto sucede porque con una exploración tan reducida, el algoritmo puede converger prematuramente a un brazo que parece óptimo en una fase temprana, sin haber explorado adecuadamente otras opciones. No obstante, tras esta fase inicial, el rechazo acumulado tiende a estabilizarse, ya que el algoritmo conforme pasa el tiempo va explorando lentamente encontrando finalmente las opciones optimas.

El mejor desempeño se observa con $\varepsilon = 0{,}1$, que representa un punto intermedio entre exploración y explotación. En este caso, el rechazo acumulado crece rápidamente al inicio debido a la fase de exploración, pero se estabiliza rápidamente a medida que el algoritmo converge hacia el brazo óptimo. Además, la frecuencia de selección del brazo óptimo alcanza valores significativamente más altos en comparación con los otros dos valores de ε, lo que confirma que esta configuración permite al algoritmo aprender de manera más efectiva.

Comparando entre las distintas distribuciones de recompensa, los efectos de cada valor de ε son consistentes en los tres tipos de bandidos. Sin embargo, en la distribución Normal, donde la variabilidad de las recompensas es mayor, los efectos negativos de una mala elección de ε son más pronunciados. En la distribución Bernoulli, donde las recompensas son binarias, la diferencia de desempeño entre los valores de ε es menos extrema, aunque sigue siendo evidente la superioridad del valor intermedio.

En conclusión, estos resultados confirman que $\varepsilon = 0{,}1$ es la mejor opción en términos de equilibrio entre exploración y explotación. Valores demasiado altos impiden una explotación eficiente, mientras que valores demasiado bajos llevan a una convergencia prematura en opciones subóptimas.

4.4.2. UCB

Los resultados obtenidos para **Upper Confidence Bound** (UCB1 y UCB2) reflejan diferencias en su desempeño según el valor del parámetro α en UCB2 y el tipo de distribución de recompensa.

En primer lugar, UCB1 muestra un rendimiento estable, especialmente en la distribución Normal, donde la tasa de selección del brazo óptimo converge rápidamente a valores cercanos al 100 %. Sin embargo, en distribuciones más inciertas como la Binomial y Bernoulli, la convergencia es más lenta, lo que sugiere que UCB1 enfrenta dificultades en entornos con mayor variabilidad en las recompensas. A pesar de esto, su rechazo acumulado se mantiene en niveles moderados, evitando exploraciones excesivas o decisiones precipitadas.

Por otro lado, UCB2 introduce un mecanismo de exploración más flexible mediante el parámetro α. Un valor bajo de α implica una mayor exploración, permitiendo que el algoritmo pruebe con más frecuencia diferentes brazos antes de comprometerse con una opción. Esto puede ser beneficioso en distribuciones con alta incertidumbre, como la Bernoulli, donde explorar más puede ayudar a evitar convergencias prematuras en soluciones subóptimas. No obstante, un α demasiado bajo puede generar indecisión, retrasando la explotación del mejor brazo y aumentando el rechazo acumulado en el corto

plazo.

En contraste, valores altos de α reducen la exploración, haciendo que el algoritmo dedique más iteraciones a un mismo brazo antes de cambiar a otro. Esto puede llevar a decisiones precipitadas, ya que el algoritmo puede fijarse en una opción subóptima demasiado rápido sin haber explorado suficientes alternativas. En las gráficas observamos que, en la distribución Binomial y Bernoulli, un α alto produce una selección del brazo óptimo baja y un rechazo acumulado creciente, lo que indica que el algoritmo no reconsidera su decisión inicial de manera efectiva.

En general, UCB2 con un α bien calibrado puede superar a UCB1 al encontrar soluciones óptimas más rápidamente y con menos rechazo acumulado. Sin embargo, su rendimiento depende en gran medida de la correcta elección del parámetro. Si α es demasiado alto, la exploración es insuficiente y el algoritmo se fija en decisiones incorrectas; si es demasiado bajo, se desperdician oportunidades de explotación. Estos hallazgos sugieren que UCB2 puede ser una alternativa eficiente, pero requiere ajuste fino para cada tipo de problema, mientras que UCB1 ofrece una solución más estable sin necesidad de calibración.

4.4.3. Softmax y Gradiente de Preferencias

El algoritmo de Softmax selecciona las acciones basándose en una distribución de probabilidad determinada por sus valores de acción estimados, controlados por un parámetro τ. Este parámetro regula el balance entre exploración y explotación: valores altos de τ inducen una exploración más uniforme entre las acciones, mientras que valores bajos favorecen la explotación de las mejores acciones encontradas hasta el momento. Por otro lado, el método de Gradiente de Preferencias en Bandits sigue una estrategia basada en la optimización de la probabilidad de selección de cada acción a través de ascenso de gradiente estocástico.

En las gráficas de porcentaje de selección del brazo óptimo, observamos que el algoritmo de Gradiente de Preferencias con un valor de $\alpha = 0{,}5$ muestra una convergencia más rápida hacia la acción óptima en comparación con las variantes de Softmax. Sin embargo, valores más bajos de α, como 0,01, exhiben un aprendizaje más lento pero constante, similar al comportamiento de Softmax con $\tau = 1$. Esto se debe a que un α mayor produce cambios más agresivos en las preferencias de acción, lo que facilita encontrar una buena solución rápidamente, aunque con riesgo de converger prematuramente a una solución subóptima.

En contraste, el algoritmo de Softmax muestra una diferencia marcada entre $\tau = 0{,}5$ y $\tau = 1$. Con $\tau = 0{,}5$, se observa una mejor selección del brazo óptimo en comparación con $\tau = 1$, ya que este último favorece demasiado la exploración, evitando una explotación efectiva de las mejores opciones identificadas. Sin embargo, ninguno de los valores de τ logra superar al Gradiente de Preferencias en términos de aprendizaje del brazo óptimo.

Las gráficas de rechazo acumulado confirman estas observaciones. El Gradiente de Preferencias con $\alpha = 0{,}5$ obtiene el menor rechazo acumulado, indicando que encuentra soluciones óptimas de manera más eficiente. En cambio, $\alpha = 0{,}01$ sufre un rechazo mayor debido a su carácter más exploratorio. Por su parte, Softmax con $\tau = 1$ muestra un rechazo más pronunciado en comparación con $\tau = 0{,}5$, lo que evidencia que un exceso de exploración impide la optimización del rendimiento a largo plazo.

En conclusión, el Gradiente de Preferencias con $\alpha = 0{,}5$ demuestra ser la estrategia más efectiva dentro de este conjunto de algoritmos, logrando la mejor combinación entre exploración y explotación. Softmax, aunque útil, depende críticamente de la elección del parámetro τ, siendo menos eficiente en comparación con el método basado en gradientes. Estas observaciones respaldan el uso de técnicas de optimización basadas en gradiente para problemas de selección secuencial de acciones.

5. Conclusiones

En este trabajo se ha llevado a cabo un estudio detallado de las técnicas y algoritmos clásicos de aprendizaje por refuerzo en el contexto del problema del bandido de k-brazos. Se han analizado distintos enfoques, incluyendo ε-Greedy, Upper Confidence Bound (UCB), Softmax y Gradiente de Preferencias, evaluando su desempeño en términos de selección del brazo óptimo y rechazo acumulado.

Además, se ha examinado el impacto de los parámetros específicos de cada algoritmo en su rendimiento. Se ha observado que la elección adecuada de estos parámetros es crucial para lograr un equilibrio entre exploración y explotación. Por ejemplo, en el caso de ε-Greedy, valores intermedios de ε ofrecieron un mejor desempeño al evitar tanto una exploración excesiva como una explotación prematura de soluciones subóptimas. De manera similar, en los algoritmos basados en UCB, el parámetro α influye significativamente en la rapidez y estabilidad con la que se identifica el brazo óptimo.

El análisis también ha puesto de manifiesto que diferentes entornos requieren estrategias distintas. Algoritmos como Gradiente de Preferencias pueden ser más efectivos en escenarios con cambios graduales en la distribución de recompensas, mientras que enfoques como UCB son adecuados cuando se busca minimizar el rechazo acumulado a lo largo del tiempo. Esta variabilidad en los resultados resalta la importancia de seleccionar el algoritmo más apropiado según la naturaleza del problema y las restricciones del entorno.

En conclusión, este estudio permite comprender cómo los métodos clásicos de aprendizaje por refuerzo pueden ser ajustados y aplicados en distintos dominios, optimizando su rendimiento en función de las características específicas del problema abordado.

Referencias

[1] *Repositorio de código: Bandido de K-Brazos.* GitHub. Recuperado de https://github.com/pedrojosefernandez1/k_brazos_FCPSSL

[2] Wikipedia. (2024). *Bernoulli distribution.* Recuperado de https://en.wikipedia.org/wiki/Bernoulli_distribution

[3] Wikipedia. (2024). *Binomial distribution*. Recuperado de https://en.wikipedia.org/wiki/Binomial_distribution

[4] Wikipedia. (2024). *Normal distribution*. Recuperado de https://en.wikipedia.org/wiki/Normal_distribution

[5] Richard S. Sutton and Andrew G. Barto. (2018). *Reinforcement Learning: An Introduction* (2ª ed.).MIT Press.