Análisis Comparativo de Algoritmos para el Problema del Bandido de los k-Brazos

Antonio Orenes Lucas - Iván Martínez Cuevas

9 de marzo de 2025

Resumen

En este trabajo se realiza un análisis comparativo de los algoritmos más utilizados para el problema de los k-brazos: Epsilon-Greedy, UCB1, UCB2 y Gradiente de Preferencias. Se evalúa su desempeño en distintos entornos con distribuciones de recompensa normal, binomial y Bernoulli, variando también el número de brazos.

Los resultados muestran que los algoritmos basados en intervalos de confianza, **UCB1 y UCB2**, presentan el mejor rendimiento en la mayoría de los escenarios, logrando una rápida convergencia hacia la recompensa óptima y minimizando el regret acumulado. Sin embargo, en entornos con alta variabilidad, como la distribución de Bernoulli, **Gradiente de Preferencias** demuestra un mejor desempeño, adaptándose mejor a las fluctuaciones en las recompensas.

Los hallazgos de este estudio refuerzan la importancia de seleccionar el algoritmo adecuado según el tipo de distribución y la complejidad del problema.

1. Introducción

El problema del bandido multi-brazo es un **desafío fundamental** en el aprendizaje por refuerzo, donde un agente debe **maximizar** su recompensa acumulada al elegir iterativamente entre mútliples acciones (brazos) con distribuciones de recompensa desconocidas. Su relevancia se extiende a pruebas clínicas, publicidad en línea y sistemas de recomendación.

La motivación para este estudio es entender cómo diferentes algoritmos equilibran **exploración con explotación** en entornos con distribuciones heterogéneas (Normal, Binomial, Bernoulli). Estudiar variantes de **Epsilon-Greedy**, **UCB** y métodos basados en **Ascenso del Gradiente** permite identificar **ventajas y limitaciones** de cada algoritmo para después poder usarlos de manera adecuada en entornos reales.

Los objetivos son los siguientes:

- **Comprar el rendimiento** de epsilon-Greedy, UCB1, UCB2 y metodos de ascenso del gradiente (Softmax, Gradiente de Preferencias) en distribuciones discretas y continuas.
- Analizar el **impacto de hiperparámetros** como alpha (UCB2), epsilon (epsilon-Greedy) y tau (Softmax) en la convergencia.
- **Evaluar la escalabilidad** con distintos números de brazos.

El documento se organiza de la siguiente manera: en la Sección 2 se presenta la definición formal del problema y el contexto teórico. En la Sección 3 se justifican los algoritmos estudiados. La Sección 4 describe la configuración experimental y los resultados obtenidos. Finalmente, en la Sección 5 se presentan las conclusiones del estudio.

2. Desarrollo

El problema del bandido multi-brazo es un modelo fundamental en el aprendizaje por refuerzo, formulado como un problema de **decisión secuencial**. En este marco, un agente debe seleccionar acciones de un conjunto finito de brazos con **recompensas desconocidas**, balanceando la exploración (adquirir información sobre los brazos) con la explotación (seleccionar el brazo que maximiza la recompensa esperada) [1].

2.1. Definición formal

Sea K el número de brazos disponibles. En cada instante de tiempo t, el agente elige un brazo $a_t \in A = \{a_1, \ldots, a_k\}$ y recibe una recompensa r_t, la cual sigue una distribución de probabilidad desconocida $\mathbb{P}(r|a_t)$. El objetivo es maximizar la recompensa acumulada esperada:

$$\sum_{t=1}^{T} r_t \tag{1}$$

donde T es el horizonte temporal. La diferencia entre la recompensa acumulada óptima y la obtenida se conoce como *regret*:

$$\mathbb{E}[R_T] = q^* T - \sum_{t=1}^{T} \mathbb{E}[r_t] \tag{2}$$

donde q^* es la recompensa esperada del mejor brazo [2].

2.2. Exploración vs. Explotación

El dilema de **exploración-explotación** aparece porque seleccionar siempre el brazo con la **mejor recompensa estimada** puede llevar a decisiones subóptimas si no se ha explorado suficientemente. **Diferentes enfoques** han sido propuestos y estudiados para manejar este dilema:

- **ϵ-Greedy**: con probabilidad ε, selecciona un brazo aleatorio para explorar; de lo contrario, selecciona el brazo con la mayor recompensa media estimada [4].

- **Upper Confidence Bound (UCB)**: considera un intervalo de confianza para la recompensa esperada de cada brazo y elige el brazo con la mayor cota superior [3].
- **Métodos basados en gradiente**: ajustan las probabilidades de selección de cada brazo mediante actualización de preferencias usando gradiente de la recompensa [4].

2.3. Distribuciones de recompensa

Las distribuciones de recompensa pueden ser continuas o discretas, afectando el desempeño de los algoritmos. Los casos estudiados son:

- **Distribución Normal**: modela escenarios con ruido gaussiano.
- **Distribución Bernoulli**: representa recompensas binarias (éxito o fracaso).
- **Distribución Binomial**: generaliza Bernoulli a múltiples ensayos.

Comprender cómo estos algoritmos manejan diferentes distribuciones es clave para aplicaciones en sistemas de recomendación, pruebas A/B y asignación de recursos en línea.

3. Algoritmos

En este estudio hemos seleccionado un **conjunto diverso de algoritmos** para abordar el problema de los k-brazos, con el objetivo de comprender mejor sus **fortalezas y debilidades en diferentes escenarios**. La elección se basa en su representatividad de distintas estrategias para el equilibrio entre exploración y explotación.

- **Epsilon-Greedy**
  - Este algoritmo representa una estrategía simple, donde se explora aleatoriamente con una probabilidad ϵ y se explota la mejor acción conocida con probabilidad $1-\epsilon$.
- **Upper Confidence Bound (UCB1 y UCB2)**
  - Estos algoritmos utilizan intervalos de confianza para guiar la exploración, seleccionando acciones con mayor incertidumbre. Su capacidad para equilibrar la exploración y la explotación de manera eficiente los hace ideales para problemas donde se busca maximizar la recompensa rápidamente.
- **Gradiente de Preferencias**
  - Este algoritmo ajusta las preferencias de acción basado en un gradiente de recompensa. Su inclusión en el estudio permite explorar como los enfoques basado en Ascenso de Gradiente se comparan con estrategias de selección de acción mas directas.
- **Softmax**
  - Otro algoritmo de Ascenso de Gradiente, se incluye para poder tener una visión más amplia de los algoritmos de esta familia.

Al comparar el **rendimiento** de estos algoritmos en el contexto del problema de los k-brazos, esperamos obtener información valiosa sobre sus **fortalezas y debilidades**, lo que nos permitirá comprender mejor cómo aplicarlos en problemas más complejos y diversos.

Este estudio se centra en un **conjunto representativo** de algoritmos, pero existen muchas otras variantes y enfoques que podrían explorarse para futuras investigaciones.

4. Evaluación/Experimentos

En este apartado, detallamos la configuración experimental para evaluar el rendimiento de los algoritmos seleccionados. Describimos las herramientas y los entornos de prueba, las métricas empleadas, resultados obtenidos y un análisis crítico de los mismos.

4.1. Configuración experimental

Los experimentos que se realizaron utilizan Python como lenguaje de programacion. Se desarrollaron clases personalizadas para implementar los distintos algoritmos, así como para simular los entornos de k-brazos con distintas distribuciones de recompensa. El entorno de desarrollo principal fue Google Colab, aprovechando las librerías Numpy para la malipulación numérica y Matplotlib para la visualización de resultados.

4.2. Metodológia de Evaluación

La evaluación se llevo a cabo en dos fases principales:

- **Estudio individual de algoritmos por familia**
  - Se realizaron experimentos iniciales para estudiar el comportamiento de cada algoritmo dentro de su respectiva familia.
  - El objetivo de esta fase fue identificar los parámetros óptimos para cada algoritmo que maximizaran su resultado.
  - Se crearon tres notebooks básicos para este propósito: uno para Epsilon-Greedy, otro para UCB1 y UCB2, y un tercero para Softmax y Gradiente de Preferencias.

- **Comparación directa**
  - Una vez identificados los parámetros óptimos, se procedió a comparar directamente el rendimiento de los algoritmos en diversos escenarios.
  - En esta fase, se realizaron comparaciones específicas entre UCB2 y Gradiente de Preferencias, y entre Epsilon-Greedy y Gradiente de Preferencias, antes de llevar a cabo un estudio final con todos los algoritmos juntos.

4.3. Métricas de Rendimiento

Para evaluar el rendimiento de los algoritmos, se utilizaron las siguientes métricas:

- **Recompensa Promedio**: mide la recompensa acumulada y promediada obtenida por el algoritmo a lo largo del tiempo.
- **Regret Acumulado**: cuantifica la pérdida acumulada de recompensa en comparación con la acción óptima.
- **Porcentaje de Selección del Brazo Óptimo**: indica la frecuencia con la que el algoritmo selecciona el brazo con la recompensa promedio más alta.

4.4. Resultados Obtenidos

A continuación se presentan los resultados obtenidos en los experimentos, organizados por familia de algoritmos.

4.4.1. Epsilon-Greedy

En esta sección se analizan los resultados al evaluar el algoritmo **Epsilon-Greedy** bajo distintas configuraciones. Los experimentos se realizaron con 10 brazos, 20 brazos, y 30 brazos, y se utilizó una distribución de recompensa **normal**, **binomial** y **Bernoulli**.

Evaluación con 10 Brazos y Distribución Normal

Los primeros experimentos fueron realizados con un entorno de **10 brazos** y una **distribución normal**. La figura 1 muestra la relación entre el **regret acumulado** y los **pasos de tiempo** para diferentes valores de ϵ (0.01, 0.1 y 1). Como se puede observar, un valor de ϵ **demasiado bajo** impide que el algoritmo descubra el brazo óptimo, lo que resulta en un **alto regret acumulado** y un desempeño pobre a lo largo del tiempo. En contraste, una estrategia con un **nivel moderado** de exploración ($\epsilon = 0.1$) logra identificar rápidamente **la mejor opción** y estabilizarse en una alta recompensa esperada.

Figura 1: Regret Acumulado vs Pasos de Tiempo (Distribución Normal, 10 Brazos)

Evaluación con 10 Brazos y Distribución Binomial

En este caso, se utilizaron 10 brazos con una **distribución binomial**. Los resultados mostraron una mayor **variabilidad** en el rendimiento comparado con la distribución normal. La figura 2 ilustra el **porcentaje de selección del brazo óptimo** a lo largo de los pasos de tiempo. Los valores de $\epsilon = 0.1$ demostraron ser más efectivos para seleccionar el brazo óptimo de manera más frecuente, aunque las fluctuaciones en las recompensas binomiales aumentaron la **incertidumbre**.

Evaluación con 10 Brazos y Distribución de Bernoulli

La Figura 3 ilustra cómo el **regret acumulado** crece con el tiempo en un entorno con **recompensas más estocásticas**. En este caso, el algoritmo muestra un **mayor**

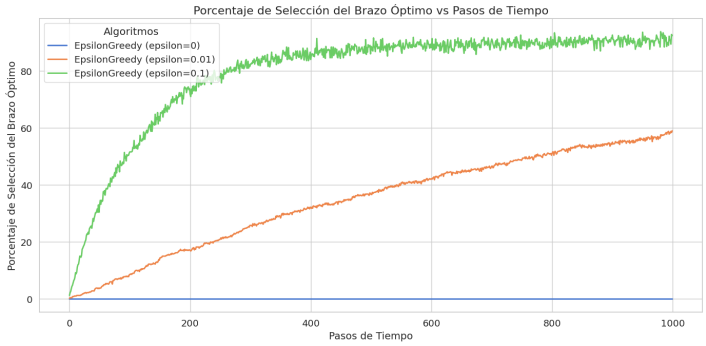

Figura 2: Porcentaje de Selección del Brazo Óptimo vs Pasos de Tiempo (Distribución Binomial, 10 Brazos)

regret inicial para valores bajos de ϵ, debido a que realiza una exploración insuficiente y no encuentra rápidamente el brazo óptimo. Sin embargo, la variante con $\epsilon = 0.1$ sigue mostrando un menor regret a largo plazo, ya que logra encontrar el brazo óptimo más rápido. El comportamiento más **errático** en este entorno refleja la naturaleza menos predecible de las recompensas Binomiales.

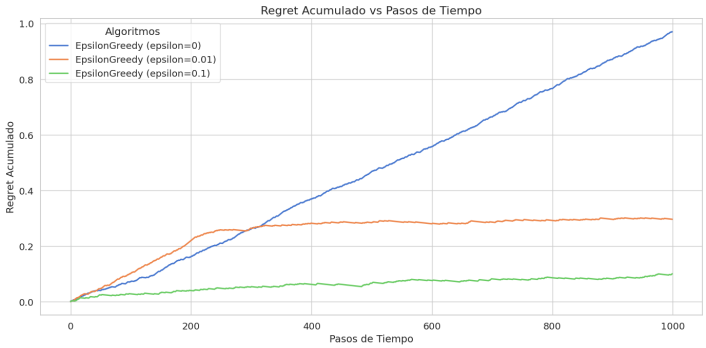

Figura 3: Porcentaje de Selección del Brazo Óptimo vs Pasos de Tiempo (Distribución de Bernoulli, 10 Brazos)

Evaluación con más brazos

Al aumentar el número de brazos en el problema del bandido, observamos ciertos **patrones** recurrentes en el desempeño del algoritmo Epsilon-Greedy. En general, incrementar la cantidad de opciones disponibles hace que la **selección del brazo óptimo sea más difícil** y retrasa la **convergencia** del algoritmo. Sin embargo, los efectos no son **drásticos**, y la estrategia con un valor moderado de exploración ($\epsilon = 0.1$) sigue ofreciendo los mejores resultados en términos de regret y recompensa promedio.

Conclusiones

El algoritmo Epsilon-Greedy muestra un rendimiento competitivo en escenarios con diferentes distribuciones de recompensa y mayor número de brazos. Sin embargo, su desempeño depende fuertemente de la configuración de ϵ, que debe ser ajustada cuidadosamente para equilibrar exploración y explotación. La **variabilidad** de las recompensas,

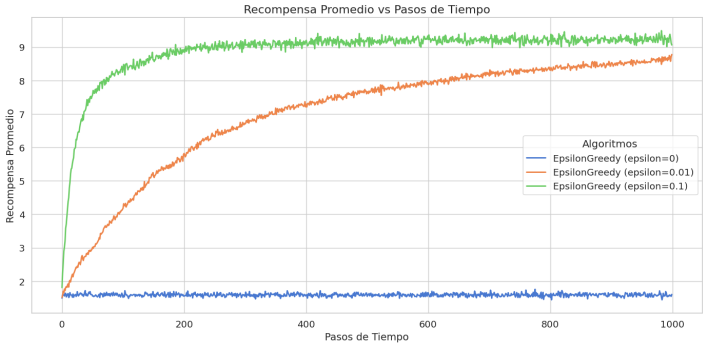

Figura 4: Recompensa promedio vs Pasos de Tiempo (Distribución Normal, 30 Brazos)

especialmente en distribuciones no normales, aumenta la **dificultad de aprendizaje**, lo que puede hacer que el algoritmo tarde más en encontrar el brazo óptimo y acumule un mayor regret en comparación con otros algoritmos más sofisticados.

4.4.2. Upper Confidence Bound

En esta sección se analizan los resultados obtenidos al evaluar los algoritmos **UCB1** y **UCB2** bajo distintas configuraciones. Los experimentos se realizaron con 10 brazos, 20 brazos y 30 brazos, utilizando una distribución de recompensa **normal**, **binomial** y **Bernoulli**.

Evaluación con 10 Brazos y Distribución Normal

Los primeros experimentos se realizaron en un entorno con **10 brazos** y una **distribución normal**. La Figura 5 muestra cómo el **regret acumulado** evoluciona con los pasos de tiempo para los algoritmos UCB1 y UCB2. Como se puede observar, ambos algoritmos logran rápidamente una recompensa óptima cercana al valor máximo. La fase de exploración es corta y eficiente, lo que demuestra la capacidad de los algoritmos para equilibrar exploración y explotación de manera efectiva.

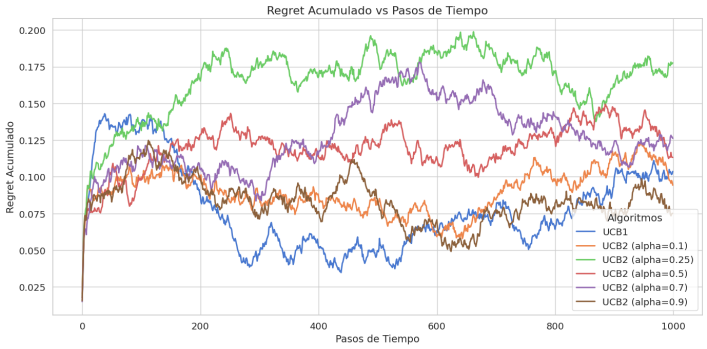

Figura 5: Regret Acumulado vs Pasos de Tiempo (Distribución Normal, 10 Brazos)

Evaluación con 10 Brazos y Distribución Binomial

En este caso, se utilizaron **10 brazos** con una **distribución binomial**. La Figura 6 muestra cómo el regret acumulado evoluciona con los pasos de tiempo para los algoritmos UCB1 y UCB2. Aunque los algoritmos siguen identificando el brazo óptimo rápidamente, las recompensas binomiales introducen fluctuaciones adicionales, especialmente en las primeras iteraciones. También se puede observar como hasta la iteración 40 todos los algoritmos tienen un resultado similar. A partir de ahí, las variantes con alpha bajo (0.1 y 0.25) superan en regret a las demás variantes.

Evaluación con 10 Brazos y Distribución de Bernoulli

En el caso de la **distribución de Bernoulli**, las recompensas binarias aumentan la incertidumbre en el proceso de aprendizaje. La Figura 7 ilustra cómo el porcentaje de selección del brazo óptimo varía con los pasos de tiempo para los algoritmos UCB1 y UCB2. UCB1 muestra menos variabilidad, mientras que UCB2 muestra variabilidad a lo largo de todo el proceso, lo que sugiere una dificultad constante para mantener una selección estable del brazo óptimo.

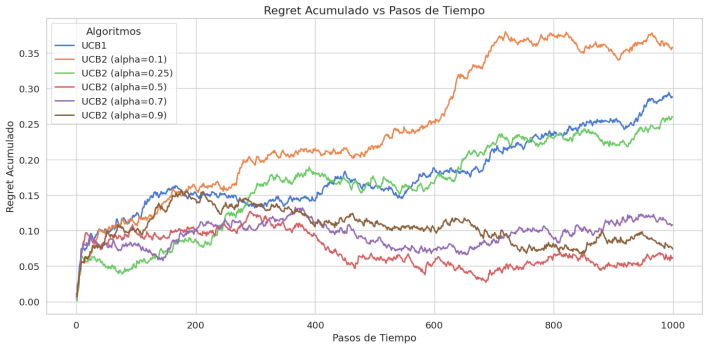

Figura 6: Regret Acumulado vs Pasos de Tiempo (Distribución Binomial, 10 Brazos)

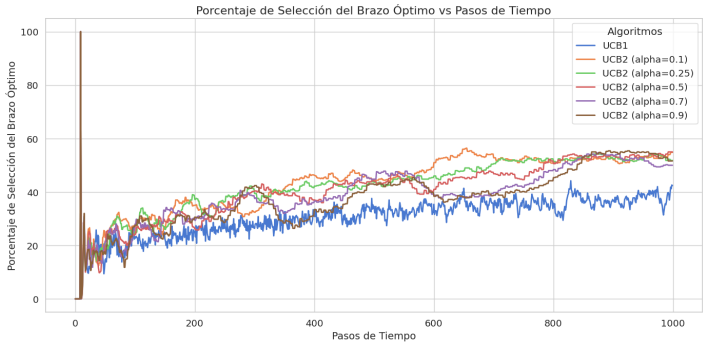

Figura 7: Porcentaje de selección del brazo óptimo (Distribución de Bernoulli, 10 Brazos)

Evaluación con más Brazos

Al aumentar el número de brazos en el problema del bandido (de 10 a 30), la fase de exploración se extiende, ya que el algoritmo debe explorar más opciones antes de

identificar el brazo óptimo. La Figura 8 muestra cómo la **recompensa promedio** evoluciona a lo largo de los pasos de tiempo en un entorno con 30 brazos y una distribución normal. A pesar de la mayor cantidad de brazos, los algoritmos UCB1 y UCB2 siguen mostrando una convergencia rápida hacia la recompensa óptima, aunque con un retraso adicional debido a la exploración inicial más prolongada.

Este inicio oscilante se debe a que, al principio del algoritmo, UCB2 asume que no sabe nada sobre las recompensas de los brazos, por lo que asigna un valor optimista a la recompensa de cada brazo. Esto significa que, para cada brazo, el algoritmo considera no solo la recompensa promedio observada hasta el momento, sino también un "margen de confianza"que puede hacer que inicialmente sobreestime las recompensas de brazos menos explorados.

Este margen de confianza inicial puede llevar a una exploración más agresiva de brazos que aún no se han probado lo suficiente. Como resultado, UCB2 puede seleccionar más brazos de lo necesario en las primeras iteraciones, lo que genera una fluctuación en las recompensas.

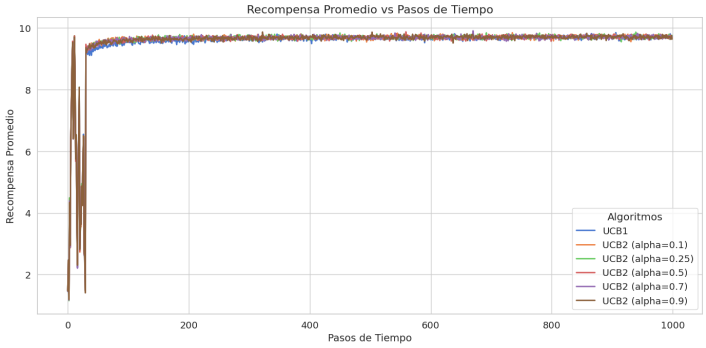

Figura 8: Recompensa Promedio vs Pasos de Tiempo (Distribución Normal, 30 Brazos)

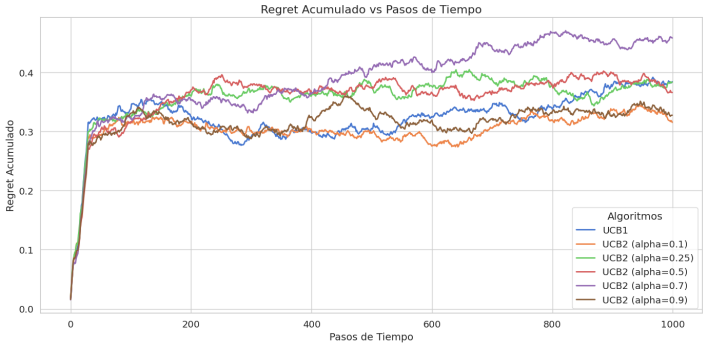

Figura 9: Recompensa Promedio vs Pasos de Tiempo (Distribución Normal, 30 Brazos)

En la figura 9 podemos ver justo el comportamiento contrario al que vimos en la figura 6 para la distribución Binomial. Las variantes que obtienen un regret acumulado más bajo son las variantes con alpha alto (0.5, 0.9).

Conclusiones

Los algoritmos UCB1 y UCB2 muestran un rendimiento eficiente en términos de exploración y explotación en diferentes contextos, con un tiempo de convergencia relativamente rápido hacia la recompensa óptima. Sin embargo, su desempeño varía según la naturaleza de las recompensas y el número de brazos disponibles. En entornos con distribuciones estocásticas como la binomial y la Bernoulli, los algoritmos requieren una fase de exploración más prolongada, lo que puede aumentar el regret inicial. Las variantes de UCB2 con valores bajos de α (0.1, 0.25) son más efectivas para reducir el regret acumulado y optimizar la recompensa promedio al adaptarse rápidamente a los brazos óptimos. Sin embargo, la variabilidad de las recompensas en distribuciones no normales aumenta la dificultad de aprendizaje en comparación con distribuciones más predecibles como la normal. A pesar de estos desafíos, los algoritmos UCB se comportan de manera robusta y eficiente, incluso cuando el número de brazos aumenta.

4.4.3. Ascenso del Gradiente

En esta sección se analizan los resultados obtenidos al evaluar los algoritmos de **Ascenso del Gradiente** con diferentes configuraciones de número de brazos y distribuciones de recompensa. Los experimentos se realizaron con 10 brazos, 20 brazos y 30 brazos, utilizando una distribución de recompensa normal, binomial y Bernoulli.

Evaluación con 10 Brazos y Distribución Normal

Los primeros experimentos se realizaron en un entorno con una **distribución normal**. La Figura 10 muestra cómo la **recompensa promedio** evoluciona con los pasos de tiempo para las variantes del algoritmo de **Ascenso del Gradiente** (Softmax y Gradiente de Preferencias). En este entorno, ambos algoritmos logran una convergencia rápida hacia una recompensa alta, con **Gradiente de Preferencias** superando al algoritmo **Softmax** en cuanto a la **recompensa promedio** y alcanzando el máximo con mayor eficacia. El algoritmo **Gradiente de Preferencias** con $\alpha = 0{,}1$ muestra el mejor desempeño.

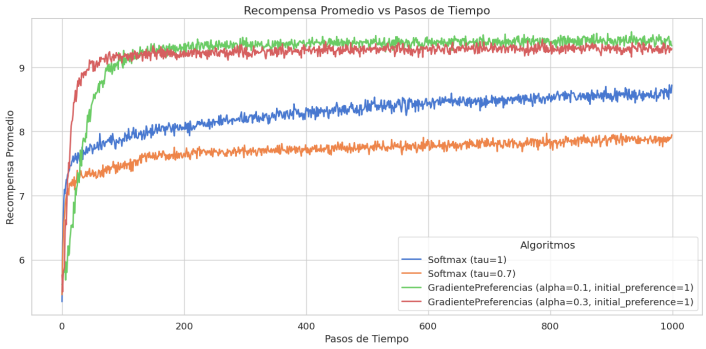

Figura 10: Recompensa promedio vs Pasos de Tiempo (Distribución Normal, 10 Brazos)

Evaluación con 10 Brazos y Distribución Binomial

En el caso de la **distribución binomial**, la Figura 11 ilustra la evolución del **regret acumulado** a lo largo de los pasos de tiempo. A pesar de las fluctuaciones adicionales generadas por la naturaleza discreta de la binomial, **Gradiente de Preferencias** mantiene un desempeño superior al de **Softmax**. El algoritmo **Gradiente de Preferencias** sigue siendo capaz de identificar rápidamente los brazos óptimos, aunque con un pequeño retraso respecto a la distribución normal. Sin embargo, el **regret acumulado** es ligeramente más alto debido a la variabilidad introducida por la distribución binomial, especialmente en las primeras iteraciones.

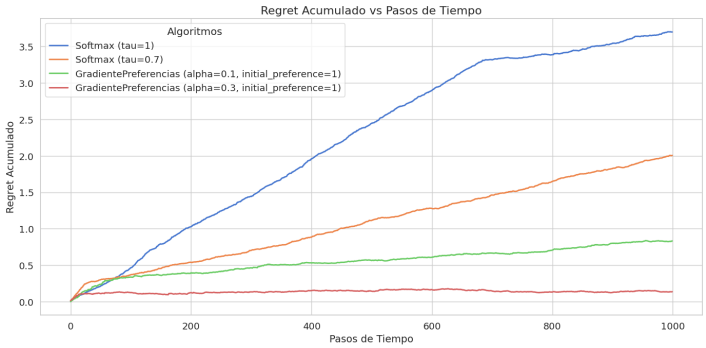

Figura 11: Regret Acumulado vs Pasos de Tiempo (Distribución Binomial, 10 Brazos)

Evaluación con 10 Brazos y Distribución de Bernoulli

La **distribución de Bernoulli** genera un entorno con un alto nivel de incertidumbre debido a las recompensas binarias. La Figura 12 muestra cómo el **porcentaje de selección del brazo óptimo** evoluciona con los pasos de tiempo. Mientras que Gradiente de Preferencias sigue mostrando un rendimiento relativamente superior, Softmax enfrenta dificultades para aprender correctamente el brazo óptimo debido a la alta variabilidad en las recompensas. En este caso, Gradiente de Preferencias con un valor de $\alpha = 0{,}3$ tiene una **convergencia más rápida** que las demás configuraciones. Gradiente de Preferencias se adapta mejor a esta situación porque tiene un enfoque más directo en **ajustar sus preferencias** de forma más eficiente a las recompensas binarias observadas, lo que le permite aprender más rápidamente y converger hacia el brazo óptimo.

Evaluación con más Brazos

Al aumentar el número de brazos (de 10 a 30), la fase de exploración se extiende, ya que el algoritmo necesita explorar más opciones antes de identificar el brazo óptimo. La Figura 13 muestra cómo la **recompensa promedio** evoluciona en un entorno con **30 brazos**. Aunque el número de brazos aumenta, **Gradiente de Preferencias** sigue mostrando una convergencia más rápida y eficiente en comparación con **Softmax**, aunque ambos algoritmos tardan más en converger debido a la mayor cantidad de opciones a explorar. Lo que da lugar a que la recompensa promedio obtenida sea ligeramente inferior.

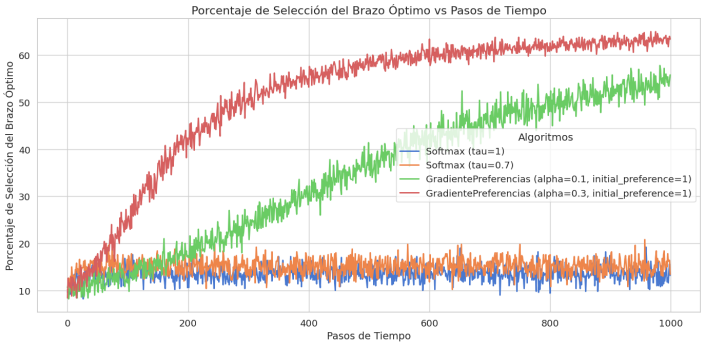

Figura 12: Porcentaje de Selección del Brazo Óptimo vs Pasos de Tiempo (Distribución de Bernoulli, 10 Brazos)

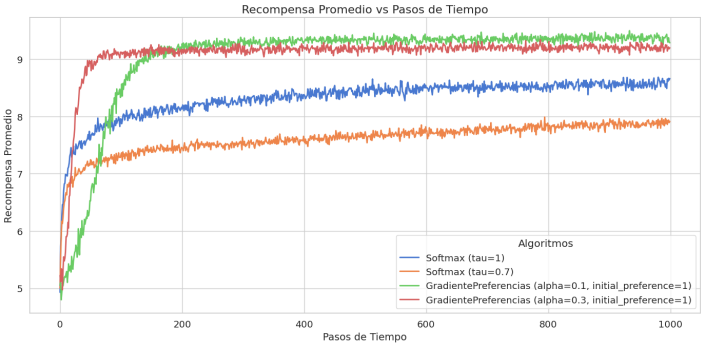

Figura 13: Recompensa Promedio vs Pasos de Tiempo (Distribución Normal, 30 Brazos)

Conclusiones

El algoritmo de **Ascenso del Gradiente** muestra un **rendimiento consistente** en diversos entornos, con una convergencia más rápida y una **mejor estimación de recompensas a largo plazo** en comparación con **Softmax**. En distribuciones con mayor variabilidad, como la **binomial** y la **Bernoulli** el algoritmo de Gradiente de Preferencias sigue demostrando ser más robusto y capaz de adaptarse más rápidamente. En cambio, Softmax tiene dificultades para adaptarse eficientemente a estos entornos ruidosos, lo que resulta en un **mayor regret acumulado** y una **menor tasa de selección del brazo óptimo**.

4.4.4. Comparación Final de Algoritmos

En este estudio, realizamos una comparación exhaustiva del rendimiento de los algoritmos comentados anteriormente: **Epsilon-Greedy**, **UCB1**, **UCB2** y **Gradiente de Preferencias**.

Durante nuestras pruebas, observamos que dentro de la familia de algoritmos de ascenso del gradiente, el **Gradiente de Preferencias** supera significativamente a la estrategia **Softmax**, que mostró un rendimiento deficiente al no alcanzar la recompensa óptima. Por lo tanto, el análisis se centra en Gradiente de Preferencias con $\alpha = 0{,}3$, comparándolo con **Epsilon-Greedy** ($\epsilon = 0{,}1$), **UCB1** y **UCB2** ($\alpha = 1$).

Evaluación con Distribución Normal (10, 20 y 30 brazos)

Para la distribución normal con **10 brazos**, los algoritmos **UCB1** y **UCB2** mostraron el mejor desempeño, alcanzando una **recompensa promedio alta** ($\sim$ 9,56), el **menor regret acumulado** ($\sim$ 0,2) y una selección casi constante del brazo óptimo ($\sim$ 100 %). Esto se debe a que su enfoque basado en **intervalos de confianza** permite una exploración eficiente y una convergencia rápida hacia la mejor opción.

El **Gradiente de Preferencias** presentó un rendimiento inferior, con un **regret acumulado significativamente mayor** ($\sim$ 1,4) y una tasa de selección del brazo óptimo de solo $\sim$ 60 %. Esto sugiere que la actualización de preferencias basada en gradiente no es suficientemente agresiva en este entorno.

Cuando el número de brazos aumenta a **20 y 30**, los resultados indican que:

- El **regret acumulado del Gradiente de Preferencias** sigue siendo bajo, lo que indica que minimiza selecciones subóptimas.
- Sin embargo, su **recompensa promedio es intermedia** y no selecciona consistentemente el brazo óptimo, lo que sugiere que explora demasiado sin enfocarse en la mejor opción.
- **UCB1 y UCB2 siguen siendo las mejores opciones**, equilibrando exploración y explotación mediante la evaluación dinámica de la incertidumbre.

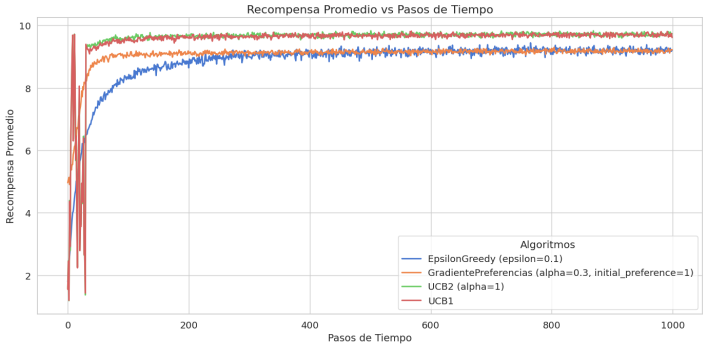

Figura 14: Recompensa Promedio vs Pasos de Tiempo (Distribución Normal, 30 Brazos)

Evaluación con 10 Brazos y Distribución Binomial

Para la **distribución binomial**, los algoritmos **UCB1** y **UCB2** continúan siendo los más efectivos, pero requieren **más tiempo para converger** debido a la mayor variabilidad de la recompensa. El **Gradiente de Preferencias** ($\alpha = 0{,}3$) muestra una **adaptación más rápida**, aunque su selección final del brazo óptimo sigue siendo inferior.

Un hallazgo interesante es que la distribución binomial genera un efecto similar a aumentar la cantidad de brazos: **los algoritmos tardan más en converger** y toman decisiones más conservadoras en la exploración.

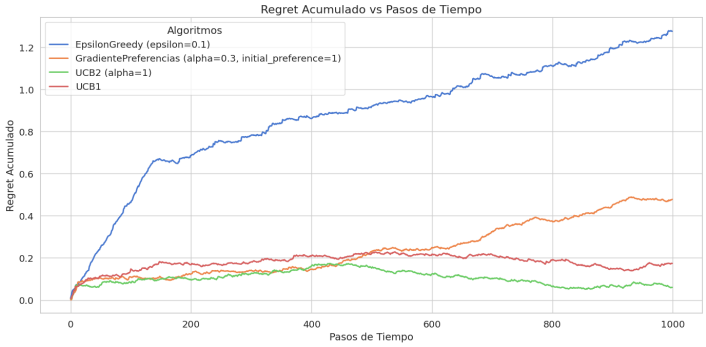

Figura 15: Regret Acumulado vs Pasos de Tiempo (Distribución Binomial, 10 Brazos)

Evaluación con 10 Brazos y Distribución de Bernoulli

En la **distribución de Bernoulli**, el **Gradiente de Preferencias** obtiene el mejor rendimiento. Aunque su convergencia es más lenta, logra la **mayor recompensa promedio** y la **mayor tasa de selección del brazo óptimo** a largo plazo. Esto sugiere que su mecanismo de actualización progresiva es más adecuado para entornos de alta variabilidad.

El algoritmo **Epsilon-Greedy** ($\epsilon = 0{,}1$) también muestra un desempeño sólido, con un buen balance entre exploración y explotación.

Por otro lado, los algoritmos **UCB1 y UCB2** tienen dificultades en esta distribución, mostrando **oscilaciones prolongadas** y menor estabilidad en la selección del brazo óptimo. En particular, **UCB1 tiene el peor rendimiento en términos de regret acumulado**, lo cual es un hallazgo inesperado, ya que generalmente es más efectivo en otras distribuciones.

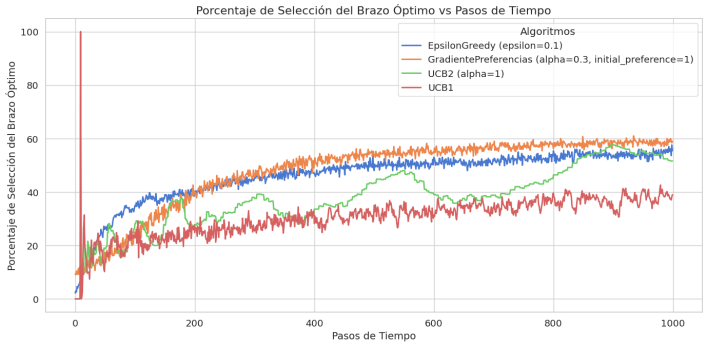

Figura 16: Porcentaje de Selección del Brazo Óptimo vs Pasos de Tiempo (Distribución de Bernoulli, 10 Brazos)

5. Conclusiones

Los resultados obtenidos en los distintos experimentos nos permiten extraer las siguientes conclusiones clave:

- Los algoritmos **UCB1 y UCB2** ofrecen un **rendimiento superior en la mayoría de los escenarios**, especialmente en entornos con un gran número de brazos o

distribuciones normales y binomiales. Son muy eficaces en maximizar la recompensa y minimizar el regret, aunque requieren una fase inicial de exploración más prolongada.
- **Gradiente de Preferencias** muestra un **desempeño variable según la distribución de recompensas**. Si bien no es la mejor opción en distribuciones normales, en la **distribución de Bernoulli** demuestra ser el algoritmo más efectivo.
- **Epsilon-Greedy** es un algoritmo simple y eficiente, con un rendimiento competitivo en la **distribución de Bernoulli** y un desempeño estable en otros escenarios. Sin embargo, su exploración constante puede limitar su capacidad de maximizar la recompensa en ciertos casos.

La elección del mejor algoritmo dependerá del contexto específico:

- **Distribución Normal y Binomial**: UCB1 o UCB2 son las mejores opciones debido a su capacidad para maximizar la recompensa y minimizar el regret, aunque en la distribución binomial convergen más lentamente.
- **Distribución de Bernoulli**: Gradiente de Preferencias destaca, con Epsilon-Greedy también mostrando un desempeño competitivo.
- **Escenarios con muchos brazos**: UCB1 y UCB2 son los más robustos, mientras que Gradiente de Preferencias y Epsilon-Greedy pueden tener dificultades para converger y maximizar la recompensa.

Estos resultados confirman que no existe un único algoritmo óptimo para todos los escenarios, y que la **selección del algoritmo adecuado depende en gran medida de la distribución de recompensas y la cantidad de brazos en el entorno**.

En cuanto a las limitaciones del estudio. Este estudio proporciona un análisis exhaustivo del desempeño de diferentes algoritmos para el problema de los k-brazos, pero tiene algunas **limitaciones** que podrían influir en los resultados y que abren la puerta a futuras investigaciones:

- **Duración de los experimentos**: Los experimentos se realizaron con un número finito de iteraciones. En entornos con un horizonte temporal más largo, algunos algoritmos podrían exhibir un comportamiento diferente, especialmente en términos de convergencia a largo plazo.
- **Falta de análisis en entornos no estacionarios**: Este estudio se centró en entornos donde la distribución de recompensa de cada brazo es fija. Sin embargo, en muchos escenarios reales, las recompensas pueden cambiar con el tiempo. Evaluar estos algoritmos en contextos no estacionarios podría revelar diferencias clave en su desempeño.

Con respecto a las líneas futuras de estudio, hay varias direcciones que pueden tomarse para extender este trabajo:

- **Incorporación de técnicas bayesianas**: Algoritmos como Thompson Sampling han mostrado un gran potencial en la literatura y podrían incluirse en estudios comparativos futuros.
- **Ajuste automático de hiperparámetros**: Desarrollar métodos que ajusten dinámicamente parámetros como ϵ en Epsilon-Greedy o α en Gradiente de Preferencias podría mejorar el rendimiento sin necesidad de un ajuste manual.

En conclusión, este estudio proporciona una base sólida para comprender el comportamiento de diferentes estrategias en el problema de los k-brazos, pero aún quedan muchas áreas por explorar.

Referencias

[1] Robbins, H. (1952). *Some Aspects of the Sequential Design of Experiments*. Bulletin of the American Mathematical Society, 58(5), 527-535.

[2] Bonald, T. (2017). *Reinforcement Learning: Multi-Armed Bandits.*.

[3] Agrawal, R. (1995). *Sample mean based index policies by O(log n) regret for the multi-armed bandit problem*. Advances in Applied Probability, 27(4), 1054-1078

[4] Sutton, R. & Barto, A. (2018). *Reinforcement Learning, second edition: An Introduction*. MIT Press.