Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC3675 - Aprendizaje Reforzado
21 de agosto, 2024

Tarea 1: Bandits

Javier Campos A. & Pedro Palma V.

Introducción

Todos los experimentos de replicación de resultados (secciones a, c y f) fueron hechos con 1000 steps y 2000 runs. Para referencia rápida, se incluyen las implemetaciones de los agentes. Para mayor detalle sobre los algoritmos implementados, referirse a nuestro código en GitHub.

a)

Los resultados del experimento se muestran en las figuras 1 y 2. Observamos que el comportamiento del agente, según ambas métricas, es prácticamente idéntico al esperado (al comparar con la fig. 2.2 del libro).

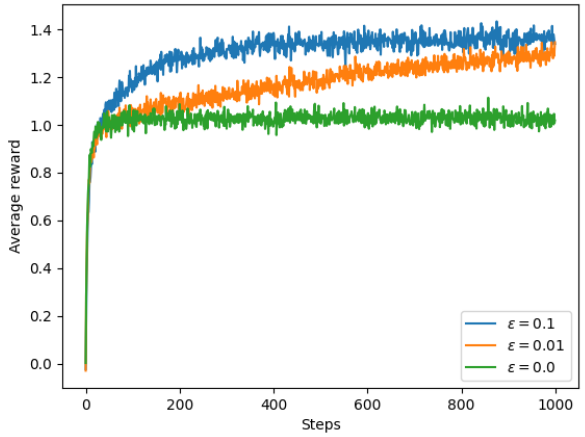

Figura 1: Average rewards - réplica de fig. 2.2 del libro

b)

Contrario a la teoría, observamos que el rendimiento del agente es sub-óptimo más de un 10 % de las veces para un $\epsilon = 0.1$. Creemos que el problema se debe a que tanto los valores esperados de cada *arm* como las recompensas de cada una de estas acciones son muestreadas desde distribuciones normales de varianza $\sigma^2 = 1$. Al ser ambas varianzas idénticas, se hace muy difícil para el agente

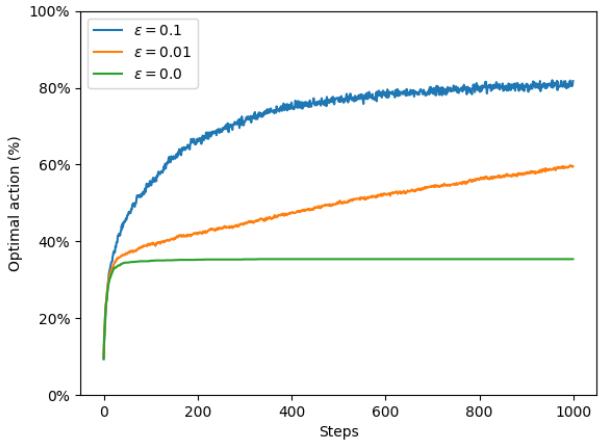

Figura 2: Optimal action % - réplica de fig. 2.2 del libro

estimar el valor real de la esperanza de cada acción en base a recompensas aleatorias porque existe mucho espacio para *overlap* entre las distribuciones de cada *arm*.
En otras palabras, las esperanzas de cada *arm* son demasiado cercanas entre sí en comparación a la varianza de cada acción, por lo que es muy posible que las muestras (recompensas) utilizadas por el agente para estimar el valor de cada acción, lo guíen a cometer errores en esta estimación (*Q-values*). **Estos errores de estimación causan que el agente seleccione acciones sub-óptimas en más del 10 % de los casos, bajo la creencia de que en realidad se trata de la acción óptima debido al *overlap* de las distribuciones desde las que se muestrean las recompensas.**
Hipótesis: Si este es el caso, el agente debería poder alcanzar un rendimiento cercano al 90 % (con $\epsilon = 0.1$) si realizamos las siguientes modificaciones a ciertos hiper-parámetros:

1. Aumentar la varianza de la distribución desde la cual se muestrean los valores esperados de cada *arm* del bandit.

2. Aumentar el número de *steps* del experimento, ya que con suficientes muestras de cada acción el agente debería ser capaz de distinguir estas (relativamente) pequeñas diferencias entre los valores esperados de cada *arm*.

Experimento 1: aumentar varianza

Aumentamos, arbitrariamente, la desviación estándar desde $\sigma = 1$ a $\sigma = 5$. Esto provocará una mayor diferencia (en promedio) entre los valores esperados de cada acción.
Según nuestra hipótesis, esto haría más fácil para el agente estimar los *Q-values* y con ello, facilita la identificación de la acción óptima. Deberíamos observar un rendimiento cercano al 90 % para $\epsilon = 0.1$.

Modificando el parámetro *scale=5.0* en la clase *BanditEnv* se obtienen los resultados de la figura 3.

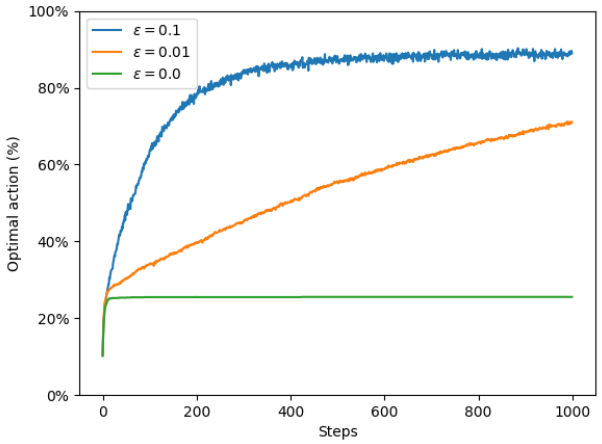

Figura 3: Optimal action %, con $\sigma = 5$

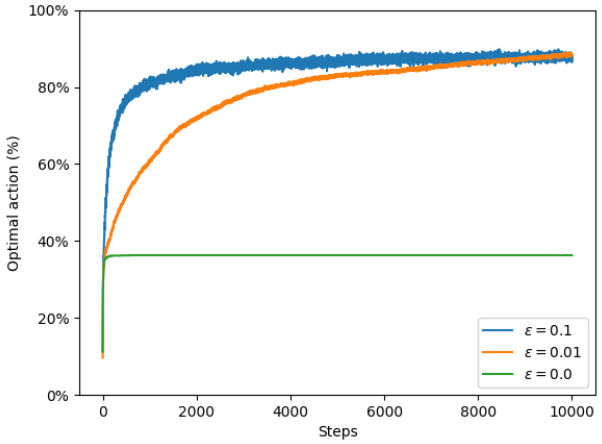

Figura 4: Optimal Action %, Steps= 10.000

Experimento 2: aumentar *steps*

Aumentamos el número de *steps* arbitrariamente por un factor x10 (manteniendo, obviamente, el valor de la varianza original , $\sigma^2 = 1$). Con esta modificación, deberíamos observar un rendimiento cercano al 90 % (con $\epsilon = 0.1$). Esto se debe a que con un alto número de muestras, el agente contaría con más información para encontrar la acción óptima y seleccionarla con probabilidad $1 - \epsilon$.
Los resultados del experimento se muestran en la figura 4.

Conclusión:

Observamos que, en los resultados de ambos experimentos, el agente logra un rendimiento cercano al 90 %, superando al desempeño obtenido con $\sigma = 1$ y solo 1000 steps. Esta evidencia apoya nuestra hipótesis, indicando que la selección de acciones sub-óptimas en más de el 10 % de las veces por parte del agente se debe al alto grado de dificultad de estimar los valores esperados de cada acción en base a muestras obtenidas de distribuciones con mucho *overlap* entre sí. Hemos mostrado que el agente puede lograr un desempeño óptimo ya sea tomando más muestras o reduciendo la

dificultad del problema al aumentar la varianza de la distribución desde la cual se toman los valores esperados de cada acción.

c)

Los resultados del experimento con un agente con *step_size* fijo replican exitosamente aquellos presentados en la figura 2.3 del libro (ver figura 5) .

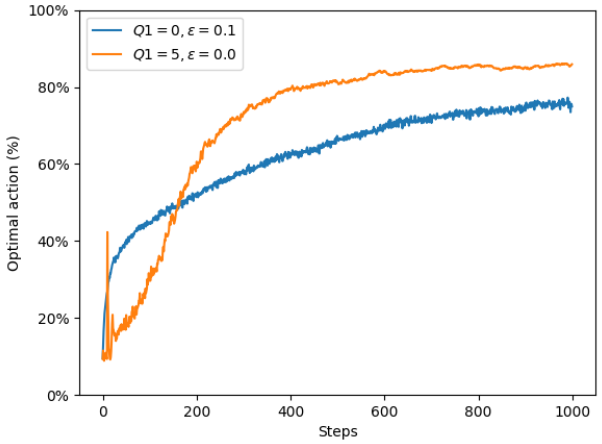

Figura 5: Optimal action % - réplica de fig. 2.3 del libro

d)

Al inicializar todas las acciones con un Q *value* optimista $Q_0 = 5$, ocurre que el agente comenzará eligiendo aleatoriamente una acción por un empate de Q *values*, existiendo mayor probabilidad de escoger una acción sub-óptima. En este caso, el agente procederá a castigar la acción escogida disminuyendo su Q *value*, para luego encontrarse nuevamente con un empate entre varias acciones. Este fenómeno ocurrirá hasta que exista una única acción con Q *value* máximo. Dado que el agente con $\epsilon = 0$ es completamente codicioso (i.e no explora), continuará seleccionando, repetidamente, esta acción que maximiza Q. Esto explica la repentina subida en el gráfico. Sin embargo, debido a la aleatoriedad de las recompensas, esta acción óptima eventualmente entregará una serie de recompensas bajas que harán descender su Q *value*, forzando al agente a seleccionar alguna de las acciones sub-óptimas. Esto explica el rápido descenso en la curva.
A medida que el agente adquiera más experiencias, continuará disminuyendo los Q *values* de las acciones sub-óptimas, convergiendo, con el tiempo, a seleccionar la acción de máximo valor esperado.

e)

La razón por la que el agente optimista($\epsilon = 0$, $Q_0 = 5$) no se acerca a rendimientos cercanos al 100 % (ver figura 5) es que, al ser completamente codicioso, no dedica nada de su tiempo a la exploración del espacio de acciones. Es su optimismo lo único que le ayuda a 'explorar' las distintas

acciones y así actualizar sus *Q values* debido a que se va 'decepcionando' iterativamente de cada una hasta eventualmente encontrar la acción óptima.

Sin embargo, por la naturaleza aleatoria de las recompensas que entregan las acciones, es probable que en algunas ocasiones la acción que tiene mayor *Q value* no se corresponda con la acción óptima, debido a que esta última entregó inicialmente malas recompensas y una acción sub-óptima quedó con mayor *Q value*. Puesto que el agente con $\epsilon = 0$ no explora, seleccionará esta acción sub-óptima por siempre.

f)

Los resultados obtenidos al replicar el experimento de la figura 2.5 del libro se muestran en la figura 6. Observamos que el comportamiento del agente es muy similar a lo esperado. De todas maneras, notamos que la curva correspondiente a $\alpha = 0.4$ con *baseline* (Verde) difiere en menos de un 10 % (aprox.) respecto a la curva presentada en el libro.
Cabe notar que estos resultados corresponden a un agente de gradiente estocástico que implementa fielmente el algoritmo visto en clases (que también se describe en el libro). En particular, la actualización del *baseline* $\bar{R}$ se define como el promedio de las recompensas obtenidas por el agente hasta el tiempo actual, sin incluirla.
Sin embargo, los autores del libro mencionan en una nota al pie en la página 37 que su gráfica de la figura 2.5 fue obtenida considerando un *baseline* que **SI** incorpora la recompensa actual, en contra de lo sugerido por ellos en el texto y también contrario a la regla de actualización del *baseline* sugerida por el algoritmo visto en clases.

En nuestro código, esto se traduce llamar al método *update.baseline()* antes de llamar al método *learn()* - que actualiza la política del agente-. Con esta modificación al agente, obtuvimos nuevos resultados para el experimento (ver figura 7), **los cuales si logran replicar con éxito la figura 2.5 del libro**.

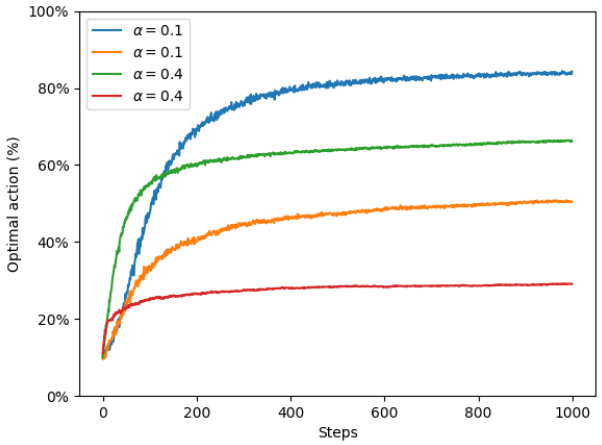

Figura 6: Optimal action %.
Leyenda:
- Curva azul y verde: con *baseline*
- Curva amarilla y roja: sin *baseline*

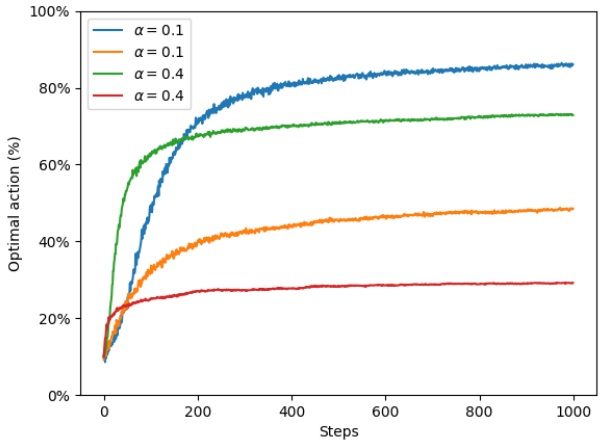

Figura 7: Optimal action % - réplica de fig. 2.5 del libro.
Leyenda:
- Curva azul y verde: con baseline
- Curva amarilla y roja: sin baseline

g)

Replicando nuevamente el experimento de la figura 2.5 del libro, esta vez con media $\mu = 0$, se obtuvieron los resultados en la figura 8. Notamos que, con esta modificación, los resultados con y sin *baseline* para cada α son prácticamente iguales, es decir, el uso de *baseline* no afecta en nada al rendimiento del agente.

Para entender mejor este comportamiento, realizamos el mismo experimento, pero con $\mu = 1$, obteniendo los resultados de la figura 9.
Es evidente que mientras a mayor μ (valor esperado de cada arm), mayor es la diferencia en el desempeño de un agente con y sin *baseline*, para ambos valores de α. En particular, es clave notar que el desempeño de los agentes sin *baseline* disminuye cuando aumenta sistemáticamente el valor esperado de las recompensas de todas las acciones.
Esto se debe a que el *baseline* funciona como un estimador del valor esperado de las recompensas del sistema, ya que está definido como la media de las recompensas obtenidas hasta la actualidad. El *baseline* le permite al agente **adaptarse** a situaciones donde los valores esperados de las recompensas estan *shifteados* por una constante arbitraria, es decir, que las muestras obtenidas al interactuar con los *k-arms* provienen de distribuciones cuyos valores medios no varían en torno a 0, actuando como una normalización para distinguir recompensas 'positivas' de 'negativas'.
Un agente sin *baseline* asume que las recompensas que está observando al ejecutar acciones provienen de una distribución de media 0 por lo que su desempeño es deficiente en situaciones donde las recompensas tienen este sesgo/*shift*. En estos casos, el agente es incapaz de estimar correctamente la política óptima a partir de las recompensas obtenidas, simplemente porque **NO** cuenta con una estimación de la media para ayudarle a definir **que es una recompensa 'buena'/'mala'**

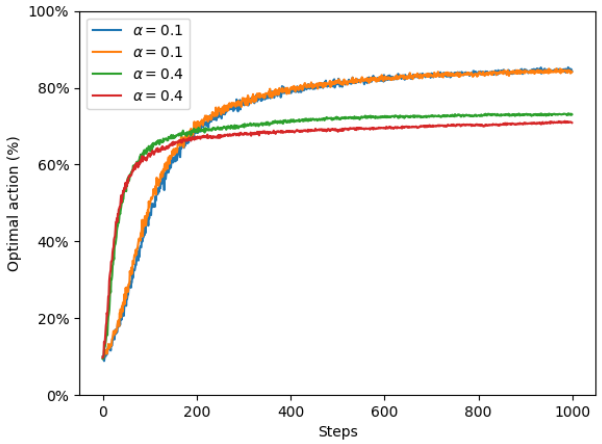

Figura 8: Optimal action %
Leyenda:
- Curva azul y verde: con *baseline*
- Curva amarilla y roja: sin *baseline*

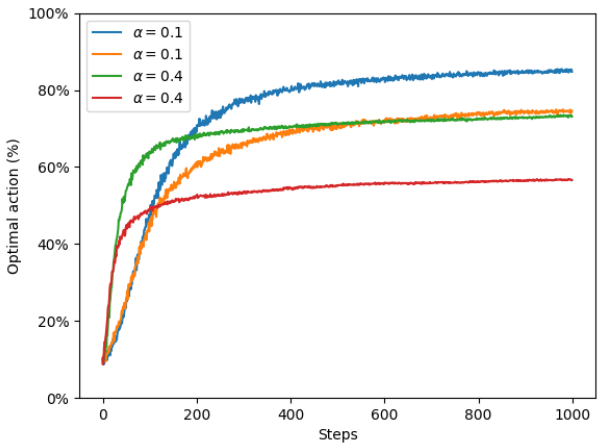

Figura 9: Optimal action %
Leyenda:
- Curva azul y verde: con *baseline*
- Curva amarilla y roja: sin *baseline*