

Ejercicio 1

Demostrar que, si conociéramos exactamente el valor de cada acción, es decir, si $Q_t(a) = E[R_t | A_t = a]$, entonces la acción *greedy* $A_t = \operatorname{argmax}_a Q_t(a)$ es la acción óptima en el sentido de que permite maximizar las recompensas totales.

Solución

Sea,

a : Acción arbitraria

A_t : Acción tomada en el tiempo t

R_t : Recompensa obtenida en el tiempo t

$Q_t(a)$: Valor estimado de la acción a en el tiempo t

$q_*(a)$: Recompensa esperada al tomar la acción a

Acción greedy: acción que maximiza el valor esperado de la recompensa

El valor $q_*(a)$ de una acción a equivale a la recompensa esperada si se selecciona dicha acción.

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a]$$

El algoritmo *greedy, no exploratorio*, estima $Q_t(a)$ como el promedio de las recompensas obtenidas, es decir,

$$Q_t(a) \doteq \frac{\text{suma de las recompensas obtenidas al seleccionar } \mathbf{a}}{\text{número de veces que se seleccionó } \mathbf{a}}$$

En cada tiempo t , el algoritmo *greedy* selecciona la acción a que maximiza el valor esperado de la recompensa, es decir, la acción a tal que $Q_t(a) = q_*(a)$. En otras palabras, en cada paso, el algoritmo elige la acción que tiene la mayor recompensa esperada, matemáticamente se expresa como

$$\begin{aligned} A_t &= \operatorname{argmax}_a Q_t(a) \\ &= \operatorname{argmax}_a q_*(a) \end{aligned}$$

Dado que $q_*(a)$ es la recompensa esperada real, al seleccionarla se obtiene la mayor recompensa posible. Por lo tanto, la acción *greedy* $A_t = \operatorname{argmax}_a Q_t(a)$ es la acción óptima en el sentido de que permite maximizar las recompensas totales.

Ejercicio 2

En una selección de acciones tipo ε -*greedy* con dos acciones posibles y $\varepsilon = 0.1$, ¿Cuál es la probabilidad de seleccionar la acción *greedy*?

Solución

En una estrategia de acciones tipo ε -*greedy*, se selecciona una acción aleatoria con probabilidad ε y se selecciona la acción *greedy* con probabilidad $1 - \varepsilon$. Es decir, si $\varepsilon = 0.1$, entonces la probabilidad de seleccionar la acción *greedy* es de $1 - \varepsilon = 0.9$. O bien,

$$P(\text{acción greedy}) = 1 - \varepsilon = 1 - 0.1 = 0.9$$

Por lo tanto, la probabilidad de seleccionar la acción *greedy* es 0.90 o el 90 %.

Ejercicio 3

Demostrar que el valor de una acción después de haber sido seleccionada $n - 1$ veces, definido como

$$Q_n = \frac{R_1 + R_2 + \dots + R_{n-1}}{n-1}$$

puede calcularse incrementalmente con la siguiente fórmula:

$$Q_{n+1} = Q_n + \frac{1}{n} [R_n - Q_n]$$

Describe la ventaja de esta fórmula desde un punto de vista computacional.

Solución

Sea,

Q_n : Valor estimado de la acción después de haber sido seleccionada $n-1$ veces

R_i : Recompensa obtenida después de la i -ésima selección de la acción

n : Número de veces que la acción ha sido seleccionada

Dado que Q_n es el promedio de las recompensas obtenidas después de haber seleccionado la acción $n-1$ veces, se tiene que

$$Q_n = \frac{R_1 + R_2 + \dots + R_{n-1}}{n-1}$$

O bien puede escribirse como

$$Q_n = \frac{1}{n-1} \sum_{i=1}^{n-1} R_i$$

O su equivalente en términos de Q_{n+1} , al sustituir n por $n+1$ en la fórmula anterior

$$Q_{n+1} = \frac{1}{n} \sum_{i=1}^n R_i$$

Si se desea calcular el valor de la acción después de haber sido seleccionada n veces, se puede hacer de manera incremental, es decir, se puede calcular el valor de la acción después de haber sido seleccionada n veces a partir del valor de la acción después de haber sido seleccionada $n-1$ veces, de la siguiente manera:

$$\begin{aligned} Q_{n+1} &= \frac{R_1 + R_2 + \dots + R_{n-1} + R_n}{n} \\ &= \frac{n-1}{n} \frac{R_1 + R_2 + \dots + R_{n-1}}{n-1} + \frac{1}{n} R_n \end{aligned}$$

Tras simplificar la expresión anterior, se obtiene

$$\begin{aligned} Q_{n+1} &= \frac{n-1}{n} Q_n + \frac{1}{n} R_n \\ &= Q_n + \frac{1}{n} [R_n - Q_n] \end{aligned}$$

Ventajas de la fórmula incremental

Desde un punto de vista computacional, la fórmula incremental más eficiente que recalcular el promedio desde cero en cada paso. Las principales ventajas son:

- **Memoria constante:** La fórmula incremental no almacena todas las recompensas anteriores, solo se necesita almacenar el valor actual de Q_n y el número de veces que la acción ha sido seleccionada, lo que reduce el requerimiento de $O(n)$ a $O(1)$, lo que es mucho más eficiente cuando n es grande.
- **Eficiencia:** El cálculo del valor de Q_{n+1} únicamente requiere una operación de suma y una operación de división, lo que es computacionalmente más eficiente que recalcular el promedio desde cero en cada paso.

Ejercicio 4

Considere un problema *k-armed bandit* con $k = 4$ acciones. Considere la aplicación de un algoritmo *bandit* usando selección de acciones ε - *greedy*, estimación incremental de los valores de cada acción y valores iniciales nulos $Q_1(a) = 0 \forall a$. Suponga la siguiente secuencia de acciones y recompensas:

- $A_1 = 1, R_1 = 1$
- $A_2 = 2, R_2 = 1$
- $A_3 = 2, R_3 = -2$
- $A_4 = 2, R_4 = 2$
- $A_5 = 3, R_5 = 0$

En algunos de estos pasos se ha tomado una decisión aleatoria.

- ¿En qué pasos definitivamente se tomaron decisiones aleatorias?
- ¿En qué pasos es posible que la decisión haya sido aleatoria?

Solución

Sea,

$k = 4$: Número de acciones

$Q_1(a) = 0 \forall a$: Valor inicial de cada acción

La secuencia de acciones y recompensas [1] se muestra en la siguiente tabla:

n	A_n	R_n	Q_{n+1}	Aleatoria
0	-	-	0.0	-
1	1	1	1.0	Sí
2	2	1	1.0	Sí
3	2	-2	0.0	Posiblemente
4	2	2	0.5	Posiblemente
5	3	0	0.4	Posiblemente

Paso 1:

Este es el primer paso y no hay información previa. Por lo que la decisión fue aleatoria.

$$Q_2 = Q_1 + \frac{1}{1}[R_1 - Q_1] = 0 + \frac{1}{1}[1 - 0] = 1.0$$

Paso 2:

Al ser el segundo paso y aún no tener información suficiente sobre las recompensas esperadas, es probable que esta decisión también haya sido aleatoria.

$$Q_3 = Q_2 + \frac{1}{2}[R_2 - Q_2] = 1 + \frac{1}{2}[1 - 1] = 1.0$$

Paso 3:

Aunque se ha seleccionado la acción 2 nuevamente, la recompensa anterior fue positiva. No obstante, considerando el algoritmo ε -greedy, existe la posibilidad de que se haya tomado una decisión aleatoria.

$$Q_4 = Q_3 + \frac{1}{3}[R_3 - Q_3] = 1 + \frac{1}{3}[-2 - 1] = 0.0$$

Paso 4:

Aunque se ha repetido la acción con una recompensa positiva en el paso anterior, podría haber habido un componente aleatorio en la elección.

$$Q_5 = Q_4 + \frac{1}{4}[R_4 - Q_4] = 0 + \frac{1}{4}[2 - 0] = 0.5$$

Paso 5:

Considerando las acciones anteriores, las cuales fueron sobre la acción 2, principalmente, la elección en cuestión podría ser el resultado de una decisión aleatoria.

$$Q_6 = Q_5 + \frac{1}{5}[R_5 - Q_5] = 0.5 + \frac{1}{5}[0 - 0.5] = 0.4$$

Ejercicio 5

[Programación] Aplique el algoritmo bandit ε -greedy con $\varepsilon \in \{0, 0.01, 0.10\}$ a un problema k -armed bandit con $k = 10$ acciones. Considere recompensas con medias aleatorias y desvío estándar constante σ . Analice experimentalmente el efecto del desvío estándar σ evaluando tres casos: $\sigma \in \{0, 1, 10\}$ ¿Qué conclusiones puede sacar?

Solución

El algoritmo bandit ε -greedy consiste en los siguientes pasos¹:

Algoritmo 1 Algoritmo ε -greedy para el problema de Multi-Armed Bandit [2]

- 1: **Inicializar:** para $a = 1$ hasta k :
 - 2: $Q(a) \leftarrow 0$ // Valor estimado de la acción a
 - 3: $N(a) \leftarrow 0$ // Número de veces que se ha seleccionado la acción a
 - 4: **Iterar n veces:**
 - 5: Seleccionar una acción A :
 - 6: $A \leftarrow \begin{cases} \operatorname{argmax}_a Q(a) & \text{con probabilidad } 1 - \varepsilon \\ \text{una acción aleatoria} & \text{con probabilidad } \varepsilon \end{cases}$
 - 7: $R \leftarrow \text{bandit}(A)$: Obtener la recompensa R de la acción seleccionada.
 - 8: $N(A) \leftarrow N(A) + 1$: Actualizar el número de selecciones de A .
 - 9: $Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$: Actualizar la estimación de valor de A .
-

Donde,

a : Acción arbitraria

¹La implementación del algoritmo se realizó en Python utilizando la librería *numpy* para el cálculo de las recompensas y se encuentra disponible en el repositorio de GitHub.

A : Acción seleccionada

R : Recompensa obtenida

$Q(A)$: Valor estimado de la acción A

$N(A)$: Número de veces que se ha seleccionado la acción A

$bandit(A)$: Función que devuelve la recompensa de la acción A

Sean las constantes,

$k = 10$: Número de brazos

$n = 20000$: Número de iteraciones

$\varepsilon \in \{0, 0.01, 0.10\}$: Probabilidad de exploración

$\sigma \in \{0, 1, 10\}$: Desvío estándar de las recompensas

Se inicializó la recompensa media de cada brazo tal que se distribuya normalmente $\mu_i \sim \mathcal{N}(0, 1)$.

$$\mu_k = [\mu_0, \mu_1, \dots, \mu_{k-1}] \quad \text{con} \quad \mu_i \sim \mathcal{N}(0, 1) \quad \forall \quad i = 0, 1, 2, \dots, k-1$$

Teniendo en cuenta las recompensas medias μ_k y los desvíos estándar σ , se calcularon las recompensas para cada brazo R_k :

$$R_k \sim \mathcal{N}(\mu_k, \sigma^2) \quad \forall i, \sigma$$

Las recompensas para cada caso se muestran en la Figura 1.

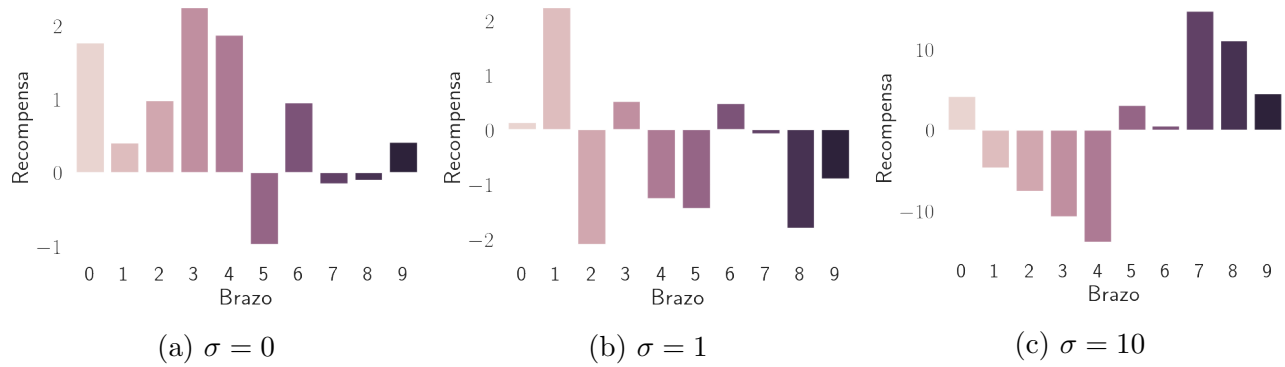


Figura 1: Recompensa para cada brazo²

Se aplicó el algoritmo $\varepsilon - greedy$ con distintas combinaciones de ε y σ . La Figura 2 muestra el número de veces que se seleccionó cada brazo para los distintos valores de ε y σ .

²Notar que la numeración de los brazos comienza en 0.

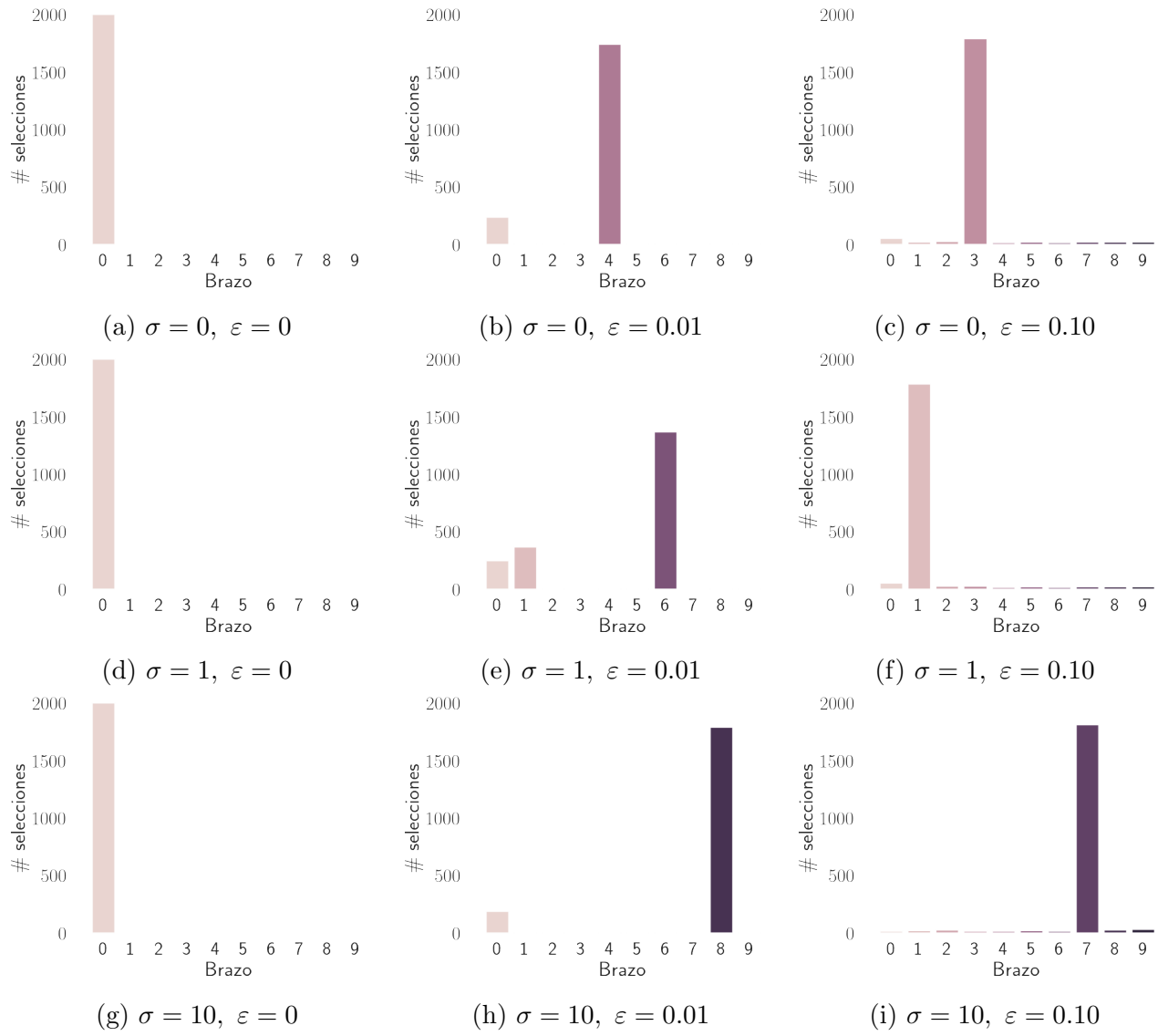


Figura 2: Número de veces que se seleccionó cada brazo, para $\epsilon \in \{0, 0.01, 0.10\}$ y $\sigma \in \{0, 1, 10\}$

Los resultados de los Q -valor de las acciones se muestran en la Figura 3.

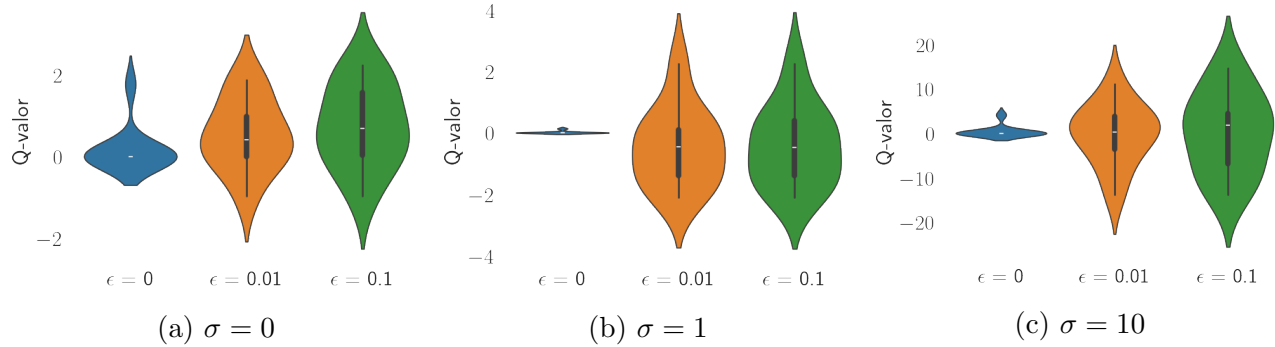


Figura 3: Q -Valor de las acciones

Finalmente, se analizó la recompensa promedio obtenida en función del número de iteraciones para los distintos valores de ϵ y σ , cuando $n \rightarrow \infty$. Los resultados se muestran en la Figura 4.

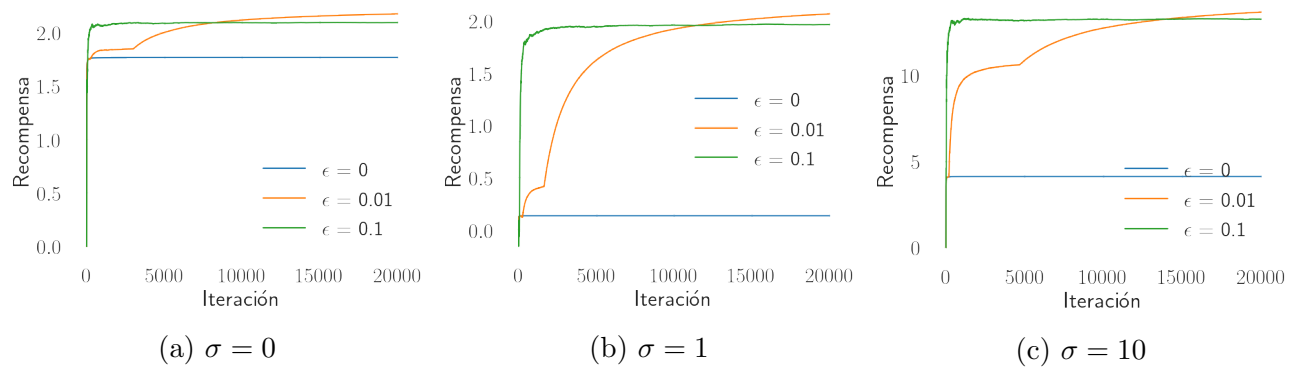


Figura 4: Recompensa promedio obtenida

Conclusiones

Efecto de σ : El desempeño del algoritmo *k-armed bandits* varía en función del desvío estándar σ , como se observa en las Figuras 4a, 4b y 4c. A medida que σ aumenta, la recompensa estimada aumenta. En particular:

- Para $\sigma = 0$ (Figura 2a), el brazo con mayor recompensa es el brazo 3, seguido de los brazos 4 y 0.
- Para $\sigma = 1$ (Figura 2d), el brazo con mayor recompensa es el brazo 1, seguido de los brazos 3 y 6.
- Para $\sigma = 10$ (Figura 2g), el brazo con mayor recompensa es el brazo 7, seguido de los brazos 8 y 9.

Al aumentar σ , Figuras 4a, 4b, 4c, la brecha entre los valores estimados de la recompensa para $\epsilon = 0.0$ y $\epsilon \in \{0.01, 0.10\}$ se amplía, lo que indica que la exploración impacta directamente en el valor estimado de la recompensa, siendo $\epsilon = 0.0$ aquella que converge a un valor subóptimo más rápidamente, mientras que $\epsilon \in \{0.01, 0.10\}$ logran una mejor estimación de la recompensa a largo plazo.

La variable/parámetro σ no puede ser optimizada en la vida real, da cuenta de la variabilidad que pueda existir en las recompensas y puede ser desconocida.

Exploración vs. Explotación (i.e., ϵ) : La elección de ϵ afecta el balance entre la exploración y la explotación del algoritmo *k-armed bandits*.

- Para $\epsilon = 0$, el algoritmo se comporta como un método de explotación pura, seleccionando siempre la misma acción, como se observa en las Figuras 2a, 2d y 2g. Esto resulta en una rápida convergencia a un valor subóptimo (Figuras 4a, 4b y 4c).
- Para $\epsilon = 0.01$, se observa un balance entre exploración y explotación, con una variabilidad moderada en los *Q-valores*. La recompensa estimada crece lentamente, pero a largo plazo supera la obtenida con $\epsilon = 0.10$, Figuras 4a, 4b y 4c. Además se observa la presencia de escalones en los que se realiza la explotación de la mejor acción, hasta el momento. Como se aprecia en las Figuras 2b, 2e y 2e, se realiza la explotación de brazos con recompensas bajas antes de explorar otras opciones que puedan tener una recompensa mayor.
- Para $\epsilon = 0.10$, el algoritmo realiza una exploración más agresiva, con una mayor variabilidad en los *Q-valores*, convergiendo rápidamente a un valor constante como se aprecia en las Figuras 4a, 4b y 4c. Como se mencionó, la exploración se realiza de manera más agresiva, por lo que se aprecia la

exploración de todas las acciones, Figuras 2c, 2f y 2i, eventualmente explorando la mejor acción a largo plazo.

Al comparar los resultados obtenidos con $\varepsilon = 0.01$ y $\varepsilon = 0.10$, se puede concluir que un mayor valor de ε (e.g., $\varepsilon = 0.10$) permite una exploración más agresiva, resultando en una mayor recompensa en el corto plazo. Por otro lado, $\varepsilon = 0.01$ ofrece saltos que dan cuenta del balance entre exploración y explotación, lo que resulta en una recompensa promedio más alta a largo plazo. Aunque al principio explora menos, a largo plazo logra una mejor estimación de la recompensa, como se muestra en las Figuras 4a, 4b y 4c.

Distribución de Q -valores:

- Para $\varepsilon = 0.0$ (Figura 3a), la distribución de los Q -valores es estrecha, lo que refleja poca variabilidad y una tendencia a explotar la mejor acción conocida.
- Para $\varepsilon = 0.01$ (Figura 3b), la distribución se ensancha, indicando una mayor exploración.
- Para $\varepsilon = 0.10$ (Figura 3c), la distribución de los Q -valores es aún más amplia, lo que sugiere una exploración más frecuente de diferentes acciones.

Lo anterior coincide con las conclusiones relacionadas al balance entre exploración y explotación.

Ejercicio 6

Dada la fórmula adaptativa del valor $Q_{n+1} = Q_n + \alpha [R_n - Q_n]$ con $\alpha \in (0, 1]$, demostrar que

- $Q_{n+1} = (1 - \alpha)^n Q_n + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i$
- $(1 - \alpha)^n + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} = 1$, es decir, Q_{n+1} es un promedio pesado de $Q_n, R_1, R_2, \dots, R_n$.

Solución

Sea,

Q_n : Valor estimado de la acción después de haber sido seleccionada $n - 1$ veces

R_i : Recompensa obtenida después de la i -ésima selección de la acción

n : Número de veces que la acción ha sido seleccionada

$\alpha \in (0, 1]$: *step-size* constante

Q_{n+1} : Valor estimado de la acción después de haber sido seleccionada n veces

Primera parte

La fórmula adaptativa para calcular el valor Q_{n+1} se puede expresar como

$$\begin{aligned} Q_{n+1} &= Q_n + \alpha [R_n - Q_n] \\ &= Q_n + (1 - \alpha) Q_n + \alpha R_n \end{aligned}$$

Si $n = 1$,

$$Q_2 = (1 - \alpha) Q_1 + \alpha R_1$$

Si $n = 2$,

$$Q_3 = (1 - \alpha) Q_2 + \alpha R_2$$

Lo que coincide con la fórmula en cuestión,

$$Q_2 = (1 - \alpha)Q_1 + \sum_{i=1}^1 \alpha(1 - \alpha)^{1-i} R_i$$

$$Q_2 = (1 - \alpha)Q_1 + \alpha R_1$$

.....

$$Q_3 = (1 - \alpha)^2 Q_2 + \sum_{i=1}^2 \alpha(1 - \alpha)^{2-i} R_i$$

$$Q_3 = (1 - \alpha)^2 Q_2 + \alpha(1 - \alpha)R_1 + \alpha R_2$$

Si se generaliza para n , se tiene que

$$\begin{aligned} Q_{n+1} &= (1 - \alpha)^n Q_n + \sum_{i=1}^n \alpha(1 - \alpha)^{n-i} R_i \\ &= (1 - \alpha)^n Q_n + \alpha(1 - \alpha)^{n-1} R_1 + \alpha(1 - \alpha)^{n-2} R_2 + \dots + \alpha R_n \end{aligned}$$

Segunda parte

Se desea demostrar que $(1 - \alpha)^n + \sum_{i=1}^n \alpha(1 - \alpha)^{n-i} = 1$. La expresión se puede reescribir como

$$\sum_{i=1}^n \alpha(1 - \alpha)^{n-i} = \alpha \sum_{i=1}^n (1 - \alpha)^{n-i}$$

Se sustituye $j = n - i$, tal que cuando $i = 1$, $j = n - 1$ y cuando $i = n$, $j = 0$.

$$\sum_{j=0}^{n-1} (1 - \alpha)^j$$

El último término de la expresión anterior es la suma de una serie geométrica, la cual se puede expresar como

$$\sum_{j=0}^{n-1} (1 - \alpha)^j = \frac{1 - (1 - \alpha)^n}{\alpha}$$

Al sustituir $(1 - \alpha)^n + \alpha \left(\frac{1 - (1 - \alpha)^n}{\alpha} \right)$ en la expresión original, se obtiene

$$(1 - \alpha)^n + \frac{1 - (1 - \alpha)^n}{\alpha} = 1$$

Esto implica que Q_{n+1} es un promedio pesado de los valores $Q_n, R_1, R_2, \dots, R_n$, donde los pesos están determinados por los términos de la serie geométrica.

Ejercicio 7

Demostrar que fórmula adaptativa para calcular el valor $Q_{n+1} = Q_n + \alpha [R_n - Q_n]$ con *step-size* $\alpha \in (0, 1]$ constante no verifica las hipótesis del teorema de convergencia y, por lo tanto, no está garantizada su convergencia.

Solución

La fórmula adaptativa para calcular el valor Q_{n+1} se puede expresar como

$$Q_{n+1} = (1 - \alpha)Q_n + \alpha R_n$$

Es decir, el valor de Q_{n+1} es una combinación lineal de los valores anteriores Q_n y la recompensa R_n .

Análisis de convergencia

Para que la secuencia Q_n converja se deben cumplir las condiciones:

- **Condición de convergencia:** El *step-size* α debe decrecer a lo largo del tiempo, es decir, $\alpha_n \rightarrow 0$ cuando $n \rightarrow \infty$.
- **Acotamiento:** La secuencia Q_n debe estar acotada, es decir, Q_n debe estar acotada para todo n .

El caso en cuestión plantea que α es constante, es decir, $\alpha_n = \alpha, \forall n$. Por lo tanto, el término $R_n - Q_n$ no se reduce a lo largo del tiempo, lo que impide que la secuencia Q_n converja, o bien, su valor presente oscilaciones.

En conclusión, teniendo en cuenta los teoremas sobre convergencia de sucesiones:

- Si el término general no tiende a cero, la secuencia Q_n no converge.
- El caso en cuestión, con $\alpha \in (0, 1]$, es contante y no decrece.

Por lo tanto, no se garantiza la convergencia de la secuencia Q_n .

Ejercicio 8

En la *Figura 2.3* del libro *Sutton&Barto (2018)* [3], se observa un *spike* en el paso número 11 cuando se utiliza inicialización optimista. De una explicación de este fenómeno.

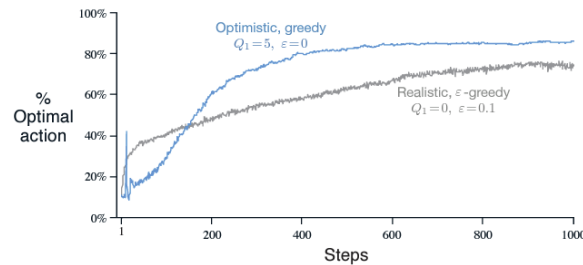


Figure 2.3: The effect of optimistic initial action-value estimates on the 10-armed testbed. Both methods used a constant step-size parameter, $\alpha = 0.1$.

Figura 5: Figura 2.3 - Sutton&Barto (2018)

Solución

Sea,

Inicialización optimista: Al inicio de cada episodio, se inicializan los valores de las acciones con un valor alto, por lo que el agente tenderá a probar todas las acciones. Esta inicialización fomenta/induce a la exploración.

Inicialización realista: Los valores de las acciones se inicializan con un valor realista, como si no se tuviera información previa sobre las recompensas esperadas. Esta inicialización fomenta/induce a la explotación.

El spike en cuestión se debe a la inicialización optimista de los valores de las acciones. Al corresponder a valores altos, el agente tenderá a probar todas las acciones, lo que puede llevar a obtener recompensas altas en los primeros pasos. En otras palabras, el agente lleva a cabo una búsqueda más agresiva al inicio del episodio.

Ejercicio 9

Demuestre que la función *SOFTMAX*: $p(a) = \frac{e^{H(a)}}{\sum_{a'=1}^K e^{H(a'')}}$, define una distribución de probabilidades discreta válida.

Solución

Para demostrar que la función *SOFTMAX* define una distribución de probabilidades discreta válida, se deben cumplir las siguientes condiciones:

No negatividad

La probabilidad de cada acción debe ser no negativa, es decir, $p(a) \geq 0$. Es decir,

$$p(a) = \frac{e^{H(a)}}{\sum_{a'=1}^K e^{H(a')}} \geq 0, \quad \forall a$$

Notar que,

- **Numerador:** $e^{H(a)} \geq 0$, ya que $e^x > 0$ para todo x .
- **Denominador:** $\sum_{a'=1}^K e^{H(a')} > 0$, ya que es la suma de los valores exponenciales de las acciones.
- **División:** La división de dos valores positivos es positiva.

Por lo tanto, la probabilidad de cada acción es no negativa, es decir, $p(a) \geq 0$.

Suma unitaria

La suma de las probabilidades de todas las acciones debe ser igual a uno, es decir, $\sum_{a=1}^K p(a) = 1$.

Para tal efecto se demuestra que la suma de las probabilidades de todas las acciones es igual a uno.

$$\sum_{a=1}^K p(a) = 1$$

O bien,

$$\sum_{a=1}^K \frac{e^{H(a)}}{\sum_{a'=1}^K e^{H(a')}} = 1$$

El cociente puede expresarse como,

$$\frac{1}{\sum_{a'=1}^K e^{H(a')}} \sum_{a=1}^K e^{H(a)} = 1$$

La suma de los valores exponenciales de las acciones es igual a la suma de los valores exponenciales de las acciones, lo que implica que la suma de las probabilidades de todas las acciones es igual a uno. En otras palabras, la suma en el numerador es igual a la suma en el denominador.

$$\begin{aligned} \frac{\sum_{a=1}^K e^{H(a)}}{\sum_{a'=1}^K e^{H(a')}} &= 1 \\ \sum_{a=1}^K p(a) &= \frac{\sum_{a=1}^K e^{H(a)}}{\sum_{a'=1}^K e^{H(a')}} = 1 \\ \sum_{a=1}^K p(a) &= 1 \end{aligned}$$

Se concluye que la función *SOFTMAX* define una distribución de probabilidades discreta válida.

Ejercicio 10

Demostrar que las derivadas de la función *SOFTMAX* $p(x)$ respecto de sus parámetros $H(a)$, $a = 1, 2, \dots, K$, son iguales a:

$$\frac{\partial p(x)}{\partial H(a)} = \begin{cases} p(x)(1 - p(x)) & \text{si } x = a \\ -p(x)p(a) & \text{si } x \neq a \end{cases}$$

Solución

Se calcula la derivada cuando $a = b$,

$$\frac{\partial p(a)}{\partial H(a)} = \frac{\partial}{\partial H(a)} \left(\frac{e^{H(a)}}{\sum_{a'=1}^K e^{H(a')}} \right)$$

Se substituyó $Z = \sum_{a'=1}^K e^{H(a')}$ previamente,

$$\begin{aligned} \frac{\partial p(a)}{\partial H(a)} &= \frac{\partial}{\partial H(a)} \left(\frac{e^{H(a)}}{Z} \right) \\ &= \frac{Z \cdot e^{H(a)} - e^{H(a)} \cdot \frac{\partial Z}{\partial H(a)}}{Z^2} \end{aligned}$$

Se calcula la derivada de Z respecto de $H(a)$, es decir, $\frac{\partial Z}{\partial H(a)}$

$$\begin{aligned} Z &= \sum_{a'=1}^K e^{H(a')} \\ \frac{\partial Z}{\partial H(a)} &= e^{H(a)} \end{aligned}$$

Se substituye en la expresión anterior,

$$\begin{aligned} \frac{\partial p(a)}{\partial H(a)} &= \frac{Z \cdot e^{H(a)} - e^{H(a)} \cdot e^{H(a)}}{Z^2} \\ &= \frac{Z \cdot e^{H(a)} - e^{2H(a)}}{Z^2} \\ &= \frac{e^{H(a)}(Z - e^{H(a)})}{Z^2} \end{aligned}$$

Además, $p(a) = \frac{e^{H(a)}}{Z}$, por lo que

$$\begin{aligned} Z &= p(a) + \sum_{b \neq a} e^{H(b)} \\ &= p(a) + (1 - p(a))Z \end{aligned}$$

Por lo tanto,

$$Z - e^{H(a)} = (1 - p(a))Z = (1 - p(x))Z$$

Finalmente,

$$\frac{\partial p(a)}{\partial H(a)} = p(x)(1 - p(x))$$

Se calcula la derivada cuando $a \neq b$. Sea,

$$p(b) = \frac{e^{H(b)}}{Z}$$

Se calcula la derivada de $p(b)$ respecto de $H(a)$,

$$\begin{aligned} \frac{\partial p(b)}{\partial H(a)} &= 0 - 0 + (-1) \cdot Z^{-2} \cdot e^{H(b)} \cdot e^{H(a)} \\ &= -p(b)p(x) \end{aligned}$$

Se concluye que las derivadas de la función SOFTMAX $p(x)$ [3] respecto de sus parámetros $H(a)$, $a = 1, 2, \dots, K$, son iguales a:

$$\frac{\partial p(x)}{\partial H(a)} = \begin{cases} p(x)(1 - p(x)) & \text{si } x = a \\ -p(x)p(a) & \text{si } x \neq a \end{cases}$$

Ejercicio 11

Demostrar que la regla de actualización por gradiente ascendente estocástico:

$$H_{t+1}(a) = H_t(a) + \alpha \frac{\partial E[R_t]}{\partial H_t(a)},$$

con $E[R_t] = \sum_{x=1}^K p_t(x)q_*(x)$, puede escribirse de la siguiente manera:

$$H_{t+1}(a) = \begin{cases} H_t(a) + \alpha(R_t - C)(1 - p_t(a)) & \text{si } a = A_t \\ H_t(a) - \alpha(R_t - C)p_t(a) & \text{si } a \neq A_t \end{cases}$$

Solución

Se calcula la derivada de $E[R_t]$ respecto de $H_t(a)$,

$$\frac{\partial E[R_t]}{\partial H_t(a)} = \sum_{x=1}^K q_*(x) \frac{\partial p_t(x)}{\partial H_t(a)}$$

Además, $p_t(x) = \frac{e^{H_t(x)}}{\sum_{x'=1}^K e^{H_t(x')}}$, por lo que si $x = a$,

$$\frac{\partial p_t(x)}{\partial H_t(a)} = p_t(x)(1 - p_t(x))$$

Si $x \neq a$,

$$\frac{\partial p_t(x)}{\partial H_t(a)} = -p_t(x)p_t(a)$$

Por lo tanto, si $a = A_t$ la derivada de $E[R_t]$ respecto de $H_t(a)$ es

$$\begin{aligned}\frac{\partial E[R_t]}{\partial H_t(a)} &= q_*(a)p_t(a)(1 - p_t(a)) + \sum_{x \neq a} q_*(x)(-p_t(b)p_t(a)) \\ &= p(a) \left(q_*(a)(1 - p_t(a)) - \sum_{x \neq a} q_*(x)p_t(a) \right)\end{aligned}$$

El término entre paréntesis es igual a $R_t - C$, siendo C el costo esperado,

$$C = E[R_t] = \sum_{x=1}^K p_t(x)q_*(x)$$

Al sustituir en la expresión anterior, se obtiene

$$\frac{\partial E[R_t]}{\partial H_t(a)} = p(a)(R_t - C)$$

Lo anterior es válido para $R = q_*(A)$, es decir, la recompensa esperada es igual a la recompensa óptima. Por lo tanto, tras remplazar en la regla de actualización por gradiente ascendente estocástico [3], se obtiene

$$H_{t+1}(a) = \begin{cases} H_t(a) + \alpha(R_t - C)(1 - p_t(a)) & \text{si } a = A_t \\ H_t(a) - \alpha(R_t - C)p_t(a) & \text{si } a \neq A_t \end{cases}$$

Referencias

- [1] M. Kubat, *An Introduction to Machine Learning*. Springer International Publishing, 2015.
- [2] A. Majumder, *Deep Reinforcement Learning in Unity: With Unity ML Toolkit*. 01 2021.
- [3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.