Bandidos Multibrazos

Alvaro J. Riascos Villegas
Universidad de los Andes y Quantil

Agosto de 2024

Contenido

El problema formal

- Sean K variables aleatoria $X_{i,t}$ (i.e., armas) con soporte en $[0,1]$.
- $X_{i,t}$ indica el resultado de la i-esima arma en el t-esimo disparo.
- Asumimos que las variables $\{X_{i,t} \mid t \geq 1\}$ son i.i.d en t de acuerdo con una distribución con media desconocida μ_i, $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$.
- Obsérvese que las variables $X_{i,t}$ pueden estar correlacionadas en i.
- Cada ronda t el agente elige una arma i para disparar y observa la recompensa $X_{i,t}$.
- El objetivo el valor acumulado de la recompensas en T rondas: $\sum_{t=1}^{T} X_{i,t} I_{[A_t=i]}$ donde A_t es la acción (arma) que se elige disparar en t.

Ejemplo: Bernoulli

- Supongamos que tres armas $X_{i,t}$ que se distribuyen Bernoulli, $Bern(\theta_i)$.
- Cuando se dispara una arma se recibe una recompensa de 1 de lo contrario cero.

Ejemplo: Bernoulli

- Supongamos que despues de interactuar con el sistema se tiene el siguiente conocimiento sobre la recompensa promedio de cada acción:

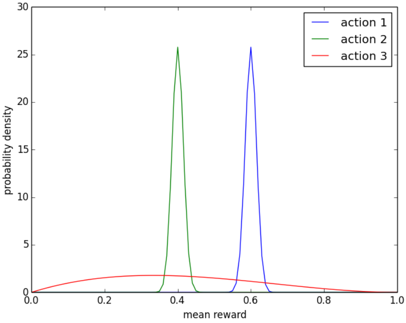

Figura: Densidad de probabilidad sobre recompensas promedio

- En promedio las acción 2 es más alta pero la acción 3 tiene mucha incertidumbre.

Ejemplo: Bernoulli

- Esta incertidumbre se puede deber a que se ha disparado poco y un algoritmo que no tenga en consideración esto podría no elegir, eventualmente, la mejor acción.
- Un algoritmo ϵ-codicioso explora com la misma probabilidad cada una de la acciones. Esto puede ser ineficiente porque la acción 2 parece estar dominada or la acción 1 mientras que la acción 3 es promisoria.
- UCB y Thompson Sampling son formas de atacar ese problema.

Ejemplo: Caminos más cortos en un grafo

- Una persona desea ir del punto 1 al 12 y los tiempos de desplazamiento $X_{i,t}$ son inciertos.

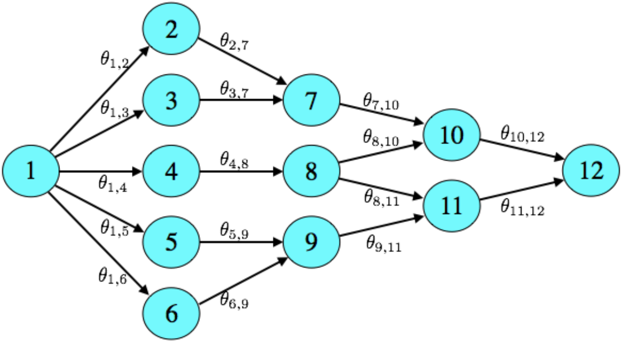

Figura: Camino más corto

- Las acciones son caminos en el grafo entre 1 y 12. Un *camino* a_t es una sucesion de enlaces $a_t = (e_1, ..., e_k)$.
- El objetivo es minimizar: $c_t = \sum_{e \in a_t} X_{t,e}$

Explorar primero

1. Fase de exploración: disparar cada arma N-veces (donde N es un parémetro del algortimo).
2. Elegir la accion con mayor recompensa.
3. Jugar esa accion en todas las rondas que faltan.

Explorar primero

- La forma estándar de evaluar un algoritmo es usando el concepto de arrepentimiento (*regret*):

$$R(T) = T \max_a q_t(a) - \sum_{t=1}^{T} q_t(a) \qquad (1)$$

  donde en esta notación $q_t(a) = E[X_{t,a}]$
- Obsérvese que entre mayor el arrepentimiento pero es la estrategia que ha generado el algortimo.
- Para el algortimo explorar primero se conoce el siguiente resultado:

$$E[R(T)] \leq T^{\frac{2}{3}} \times O(Klog(T))^{\frac{1}{3}} \qquad (2)$$

Algortimos más eficientes

- La cota superior sobre el arrepentimiento del algoritmo anterior es bastante grande.
- ϵ - greedy y UCB1 son mejoras sutanciales.

Bandidos con Mochilas

- Los problemas de bandidos con mochilas (*bandits with knapsacks*) es un modelo general para imponer restricciones al problema de bandidos estudiado hasta este punto.

Ejemplo: pricing dinámico con oferta limitada

- El problem es:
  - Un vendedor tiene B productos identicos para vender.
  - Debe intentar hacer en T rondas de interaccion con los consumidores. Cada interacción es una oferta tomelo o dejelo con un unico consumidor.
  - En cada ronda el vendedor anuncia el precio de venta $p_t \in [0, 1]$. El comprador recibe una valoración privada v_t realización de una distribucion desconocida D.
  - El comprador compra el producto si $p_t < v_t$.
  - El objetivo del vendedor es maximizar las ventas en la T rondas.
- Si $B = T$ realmente no hay restrición porque igual el máximo número de items que se pueden vender en T interacciones es T.
- Si $B < T$, si hay una restricción de oferta limitada en todas las rondas (i.e., global).
- Este es un problema de bandidos con un continuo de armas todas con la misma distribucion de recompensas. Una accion es $p_t \in [0, 1]$

Ejemplo: pricing dinámico de varios productos con oferta limitada:

- El problem es:
  - Un vendedor tiene n productos para vender y un inventario limitado $B_1, ..., B_n$ de ada uno.
  - Debe intentar venderlos en T rondas de interacción con los consumidores. Cada interacción es una oferta tomelo o dejelo con un unico consumidor.
  - En cada ronda el vendedor anuncia el precio de los n productos $p_{t,i} \in [0,1]$. El comprador recibe una valoración privada $v_{t,i}$ de cada producto, realización de una distribución desconocida D.
  - El comprador compra el producto si $p_{t,i} < v_{t,i}$.
  - El objetivo del vendedor es maximizar las ventas en la T rondas.
- Si $B = T$ realmente no hay restrición porque igual el máximo número de items que se pueden vender en T interacciones es T.
- Si $B < T$, si hay una restriccion de oferta limitada en todas las rondas (i.e., global).

Bandidos con Mochilas Formalmente

- Supongamos que tenemos K armas y d recursos con presupuestos $B_1, .., B_d \in [0, T]$.
- En cada ronda:
  1. Elegir $a_t \in \{1, ..., K\}$
  2. Se observa $(r_t; c_{t,1}, ..., c_{t,d}) \in [0, 1]^{d+1}$ donde r_t es la recompensa y $c_{t,1}$ es consumo del recurso i.
  3. El algoritmo para cuando algun recurso consume la totalidad de su presupuesto.

Aplicaciones

- Pricing dinamico uno o varios productos.
- Pricing dinamico para contratar.
- Pago por click en subastas de avisos publicitarios.
- Subastas repetidas.
- Ofertas dinámicas en una subasta con un presupuesto.

Aplicaciones: Pricing dinamico

- El problema anterior se puede escribir como un problema con dos recursos, el tiempo T y el inventario del producto.
- Se observa $(p, 1, 1)$ si el precio es aceptado y $(0, 0, 1)$ si no es aceptado.

Aplicaciones: Pricing dinamico de varios productos

- El problema anterior se puede escribir como un problema con $K+1$ recursos, el tiempo T y el inventario de cada producto.
- Se observa $(r, 1, 1, .., 0, 1, ..,1)$ si se vende algun producto, donde r son los ingresos por las ventas totalees. $(0, 0, ..., 0, 1)$ si no es aceptado.

UcbBwK

- El algoritmo que resuelve este problema con una buena garantía de desempeño es una extensión de UCB1.

Contenido

El problema formal

- Sean M variables aleatoria $X_{i,t}$ (i.e., armas) con soporte en $[0,1]$.
- $X_{i,t}$ indica el resultado de la i-esima arma en el t-esimo disparo.
- Asumimos que las variables $\{X_{i,t} \mid t \geq 1\}$ son i.i.d en t de acuerdo con una distribución con media desconocida μ_i, $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$.
- Obsérvese que las variables $X_{i,t}$ pueden estar correlacionadas en i.
- Cada ronda del juego el agente elige un subconjunto $S \subset \{1, ..., M\}$ de armas para disparar. Decimos que S es una super arma.
- En cada ronda solo se observa el disparo de las armas en la super arma.
- Por simplicidad suponemos que $\mid S \mid = k$ para un k conocido.

El problema formal

- Para cada arma $i \in \{1, ..., M\}$, donde M es el número total de armas, sea $T_i(t)$ el número de veces la arma i se ha disparado en t rondas. Si una arma no se dispara en la ronda t entoncdes $T_{i,t} = T_{i,t-1}$.
- Sea $R_t(S)$ las varibles aleatorias no negativas que representan la recompensa en t cuando la super arma S se juega.
- Suponemos $R_t(S)$ es de la forma $R_t(S) = \sum_{i \in S} X_{i,T_{i,t}}$.
- $R_t(S)$, $E[R_t(S)]$, es una función de S y los parámetros μ_i de las armas en S.
- El problema que queremos resolver es encontrar un algoritmo, un método para elegir una super arma en cada ronda t, tal que maximice el valor esperado de la recompensa en cada ronda t: $E[R_t(S)] = \sum_{i \in S} \mu_i$, para un vector de parámetros $\boldsymbol{\mu}$ desconocidos.

CUCB

1. For each arm i, maintain: (1) variable T_i as the total number of times arm i is played so far; (2) Variables $\hat{\mu}_i$, as the mean of all outcomes $X_{i,t}$ for $1 \leq i \leq M$ that have been observed up to round t (initially 1).
2. $t \leftarrow t + 1$. For each arm i, set $\bar{\mu}_i = \text{mín}\left\{\hat{\mu}_i + \sqrt{\frac{3 \ln t}{2 T_i}}, 1\right\}$. $S = \mathrm{Oracle}(\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_m)$. Play S. Observe outcomes of played base arms i, and update all T_i's and $\hat{\mu}_i$'s.