Aprendizaje Reforzado

Maestría en Data Mining, Universidad Austral

Javier Kreiner

Repaso de Bandidos Multi-brazo

- En cada paso tomo una acción
- Cada acción tiene asociada una distribución de recompensas diferente
- Luego de tomar la acción el ambiente nos da una recompensa $A_t \rightarrow R_t$
- Pero no conocemos la distribución $q_*(a) = \mathbb{E}[R_t | A_t = a]$ de cada acción $a = 1, \ldots, k$
- El objetivo es actuar de tal manera de maximizar la recompensa acumulada esperada
- O sea usar una política $\pi(a_1, r_1, \ldots a_{t-1}, r_{t-1}) = a_t$ que maximice la recompensa acumulada
- También podemos verlo como minimizar la 'regret':

$$R_T := \max_{i=1,\ldots,K} \mathbb{E}\left[\sum_{t=1}^{T} r_{i,t} - \sum_{t=1}^{T} r_{a_t,t}\right]$$

Bandidos Multibrazo Contextuales

- Antes de tomar una acción en cada paso el ambiente nos da un contexto x_t
- Por ejemplo características de un usuario y un anuncio/artículo/medicina
- En cada paso tomo una acción y luego de tomar la acción el ambiente nos da una recompensa
- Ahora la recompensa depende de la acción tomada y el contexto recibido
- Formalmente:
  - El ambiente 'sortea' (x_t, $\mathbf{r}_t$) de un distribución $\mathcal{D}$ sobre $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
  - Observamos contexto x_t.
  - Elegimos a_t de $\mathcal{A}$.
  - Recibimos recompensa $\mathbf{r}_t[a_t]$
- Objetivo: minimizar la 'regret': $R_{\mathrm{A}}(T) \stackrel{\text{def}}{=} \mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right] - \mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t}\right].$

  Importante: no recibimos componentes de $\mathbf{r}_t$ para las acciones no seleccionadas

Comparación de los modelos

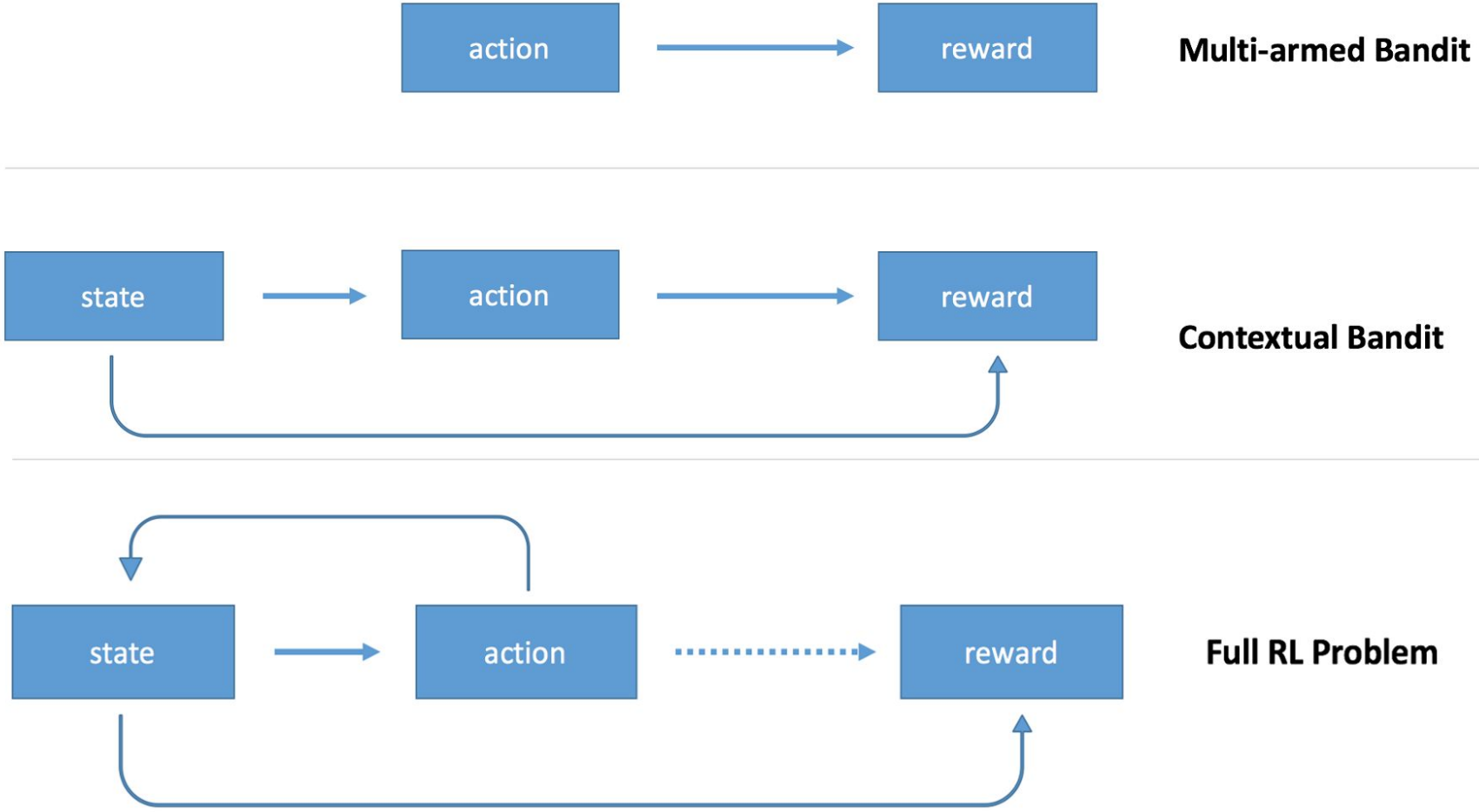

Tabla comparando las diferentes situaciones

	Recompensa depende el estado	Acción modifica el mundo	Balance exploración/explotación
Bandidos multibrazo	No	No	Sí
Bandidos contextuales	Sí	No	Sí
Aprendizaje reforzado (full)	Sí	Sí	Sí

Relación con aprendizaje supervisado

- Muchas veces bandidos contextuales es tratado como un problema de aprendizaje supervisado
- El problema es que si usamos una política basada en una función de mapeo aprendida con datos pasados no aprendemos a tomar acciones fuera de esa política
- Cómo convertir un problema supervisado arbitrario en bandidos contextuales:
    - presentar un ejemplo al agente
    - el agente selecciona un label
    - si es el label correcto se le da recompensa 1, caso contrario 0
    - no se muestra el label correcto, sólo la recompensa para el label elegido

Aplicaciones de bandidos y bandidos contextuales

(fuente: A Survey on Practical Applications of Multi-Armed and Contextual Bandits)

- Recomendación de artículos de noticias personalizados
- Diseño de ensayos clínicos
- Selección de portafolio
- Sistemas de recomendación en situaciones en que los items y usuarios cambian dinámicamente
- Pricing dinámico
- Sistemas de diálogo (dialogue systems)
- Detección de Anomalías (para detectar fraude con tarjetas de crédito)

Recomendación de artículos de noticias personalizados

(fuente: A Contextual-Bandit Approach to Personalized News Article Recommendation)

- Objetivo: Seleccionar artículos para presentar a usuarios secuencialmente, considerando información contextual de usuarios y artículos, y utilizando el feedback en forma de clicks para maximizarlos
- Desafiante:
  - el pool de contenido, usuarios e intereses cambia dinámicamente (approaches tradicionales no funcionan)
  - la escala de lo servicios involucrados requiere soluciones con aprendizaje y cómputo veloces
- Hay que balancear dos objetivos: maximizar la satisfacción a largo plazo y explorar para obtener información sobre cuán bueno es el match entre usuario y contenido
- Los brazos serían los artículos a mostrar.
- Si un artículo es cliqueado, hay una recompensa de 1, si no de 0. Con esta definición la recompensa esperada de un artículo es su click through rate (CTR)

Recomendación de artículos de noticias personalizados

- Cómo las noticias cambian frecuentemente, es necesario explorar la reacción de los usuarios a las novedades, mientras también se 'explotan' los items que sabemos que funcionan
- Los usuarios son representados con un vector de características: demográficas, geográficas, actividades históricas agregadas, etc.
- Los artículos con un vector de características: información descriptiva, categorías, etc.
- El sistema debe reconocer similaridades entre usuarios y entre artículos, porque son numerosos y la interacción por lo tanto es esparsa
- En este caso el contexto que recibe el agente es el usuario que está visitando
- Cada brazo sería el artículo a mostrar (con sus características)
- Cada vez que actuamos acumulamos experiencia, dado un contexto y una acción (elección de artículo), se recibió una cierta recompensa

Ejemplo de Yahoo!

Cómo evaluar una política cuando tengo datos del pasado aleatorios

- Supongamos que actuamos según una política aleatoria que toma todas las acciones con igual probabilidad (logging policy)
- El objetivo es evaluar una política determinada π
- Tenemos una secuencia de tuplas (x_t, a_t, r_t)
- Si $\pi(h_{t-1}, x_t) = a_t$ entonces consideramos la tupla para evaluar la política

Ejemplo con Vowpal Wabbit

- sudo apt install libssl-dev
- sudo apt install libboost-dev
- sudo apt install libboost-all-dev
- instalar https://cmake.org/install/. Bajar, descomprimir y en la consola:
  - `./bootstrap`
  - `make`
  - `make install`
- pip3 install vowpalwabbit

Ejemplo con Tensorflow

- git clone https://github.com/tensorflow/models.git

Lecturas recomendadas:

- A Contextual-Bandit Approach to Personalized News Article Recommendation
- Deep Bayesian Bandits Showdown