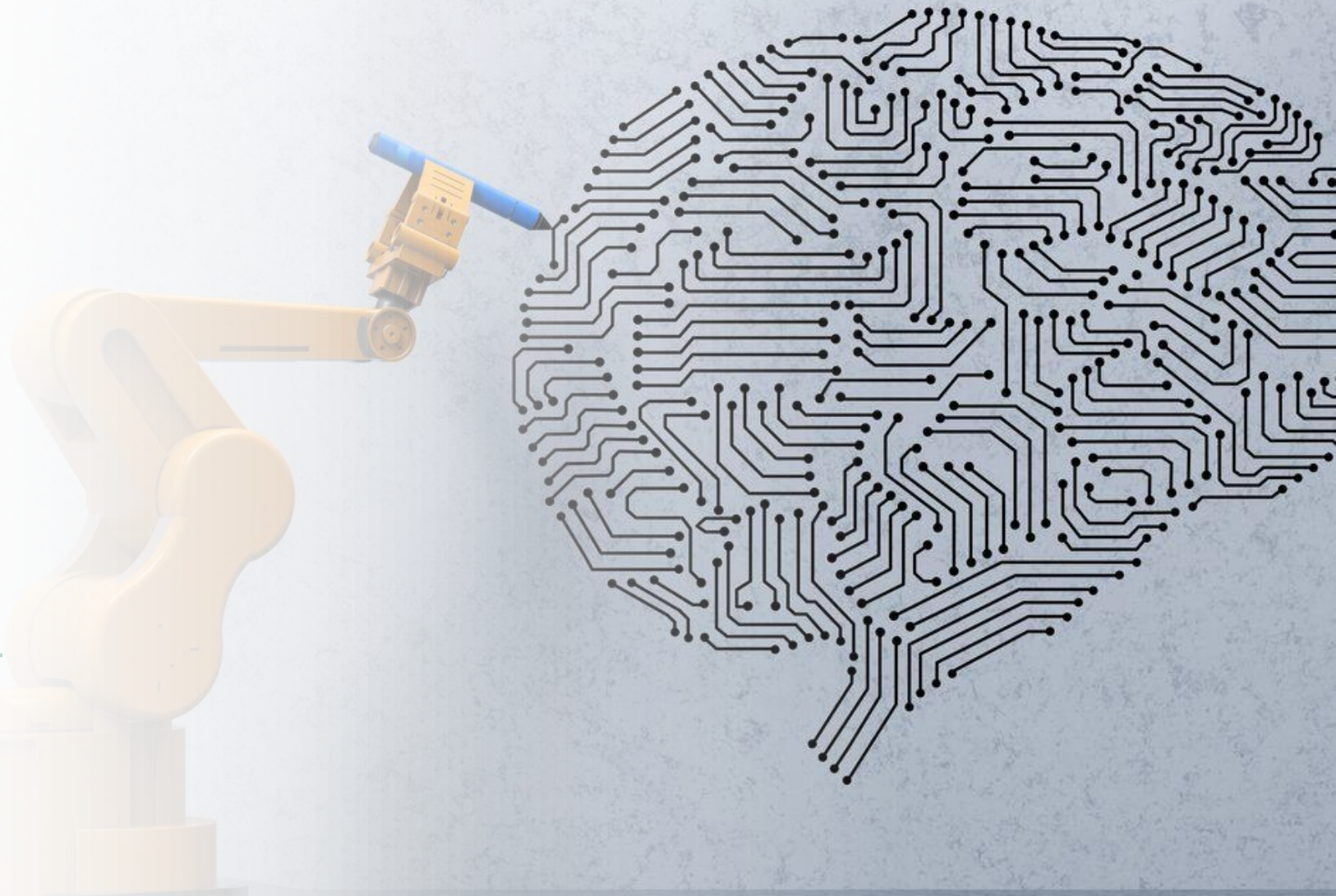
Aprendizaje por refuerzo

Clase 11: Exploración II



Para el día de hoy…

- Exploración

En la clase anterior… UCB1

$$a_t = \arg\max_{a \in \mathcal{A}} Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

- Teorema: el algoritmo UCB obtiene regret asintótico total logarítmico

$$\lim_{t \to \infty} L_t \leq 8 \log t \sum_{(a|\Delta_a > 0)} \Delta_a$$

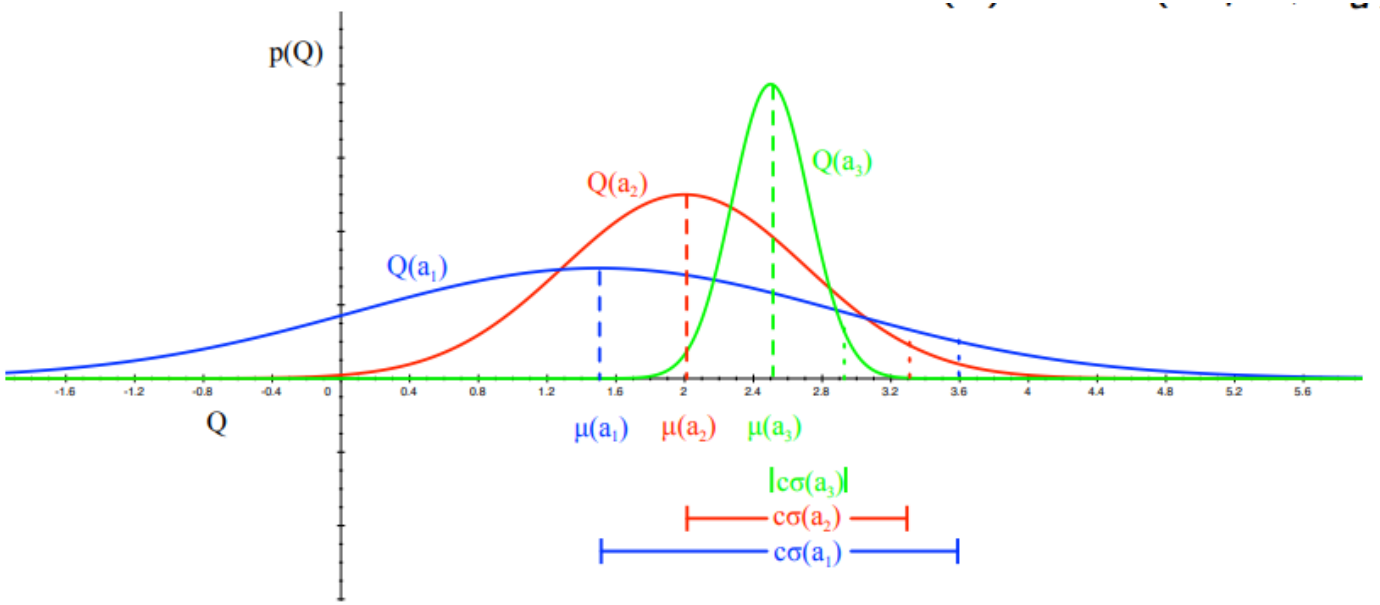
Ahora… Bandidos Bayesianos

- Hasta ahora no hemos hecho suposiciones acerca de la distribución de la recompensa $\mathcal{R}$
- Los bandidos Bayesianos explotan la información a priori de recompensas $p[\mathcal{R}]$
- Calculan la distribución de recompensas a posteriori $p[\mathcal{R}|h_t]$ donde $h_t = a_1, r_1, \ldots, a_{t-1}, r_{t-1}$ es la historia
- Usamos la distribución a posteriori para guiar la exploración
  - Limites de confianza superiores (UCB Bayesianos)
  - Pareo de probabilidades (muestreo de Thompson)
- Mejor desempeño si el conocimiento a priori es preciso

Ejemplo de UCB Bayesianos

- Supongamos que la distribución de recompensa es Gaussiana, $\mathcal{R}_a(r) = \mathcal{N}(r; \mu_a, \sigma_a^2)$
- Calcular la posteriori Gaussiana sobre μ_a y σ_a^2 (por medio del teorema de Bayes)
- $p[\mu_a, \sigma_a^2 | h_t] \propto p[\mu_a, \sigma_a^2] \prod_{(t|a_t = a)} \mathcal{N}(r_t; \mu_a, \sigma_a^2)$
- Elegir la acción que maximice la desviación estándar de $Q(a)$

$$a_t = \arg\max \mu_a + \frac{c\sigma_a}{\sqrt{N(a)}}$$



Pareo de probabilidades

- Selecciona una acción a de acuerdo a la probabilidad que a sea la acción óptima

$$\pi(a|h_t) = \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a|h_t]$$

- El pareo de probabilidad es optimista bajo incertidumbre
- Las acciones inciertas tienen mayor probabilidad de ser seleccionadas
- Puede ser difícil calcular analíticamente de la posteriori

Muestreo de Thompson

- Implementa el pareo de probabilidades

$$\pi(a|h_t) = \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a|h_t]$$
$$= \mathbb{E}_{(\mathcal{R}|h_t)}[1\left(a = \arg\max_{a \in \mathcal{A}} Q(a)\right)]$$

- Utiliza el teorema de Bayes para calcular la distribución a posteriori $p[\mathcal{R}|h_t]$
- Muestrear la distribución de recompensa $\mathcal{R}$ de la posteriori
- Seleccionar la acción que maximiza el valor en la muestra

$$a_t = \arg\max_{a \in \mathcal{A}} Q(a)$$

- El muestro de Thompson alcanza el límite inferior de Lai y Robbins

Valor de la información

La exploración es útil porque obtiene información

¿Es posible cuantificar el valor de la información?

¿Cuánta recompensa puede pagar el tomador de decisiones para obtener información?

Recompensa a largo plazo vs recompensa inmediata

La información es más valiosa en situaciones inciertas

Tiene sentido explorar en situaciones inciertas

Si conocemos el valor de la información, podemos buscar el compromiso óptimo entre exploración y explotación

Espacio de estado de información

- Hasta ahora hemos visto los bandidos como toma de decisión de un paso
- También se puede ver como toma de decisiones secuencial
  - $\tilde{s}$ es la estadística de la historia, $\tilde{s}_t = f(h_t)$
  - Resume toda la información acumulada
- Cada acción a causa una transición a un nuevo estado de información $\tilde{s}'$, con probabilidad $p(\tilde{s}, a, \tilde{s}')$
- Esto define un MDP $\widetilde{\mathcal{M}}$ en el espacio aumentado de información

$$\widetilde{\mathcal{M}} = (\tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma)$$

Bandidos de Bernoulli

- Consideremos un bandido de Bernoulli, tal que $\mathcal{R}^a = \mathcal{B}(\mu_a)$
- Por ej. La probabilidad de ganar o perder un juego μ_a
- Queremos encontrar el brazo que tenga el μ_a más alto
- El estado de información es $\tilde{s} = (\alpha, \beta)$
  - α_a cuenta el número de veces que se tomó a y la recompensa fue 0
  - β_a cuenta el número de veces que se tomó a y la recompensa fue 1

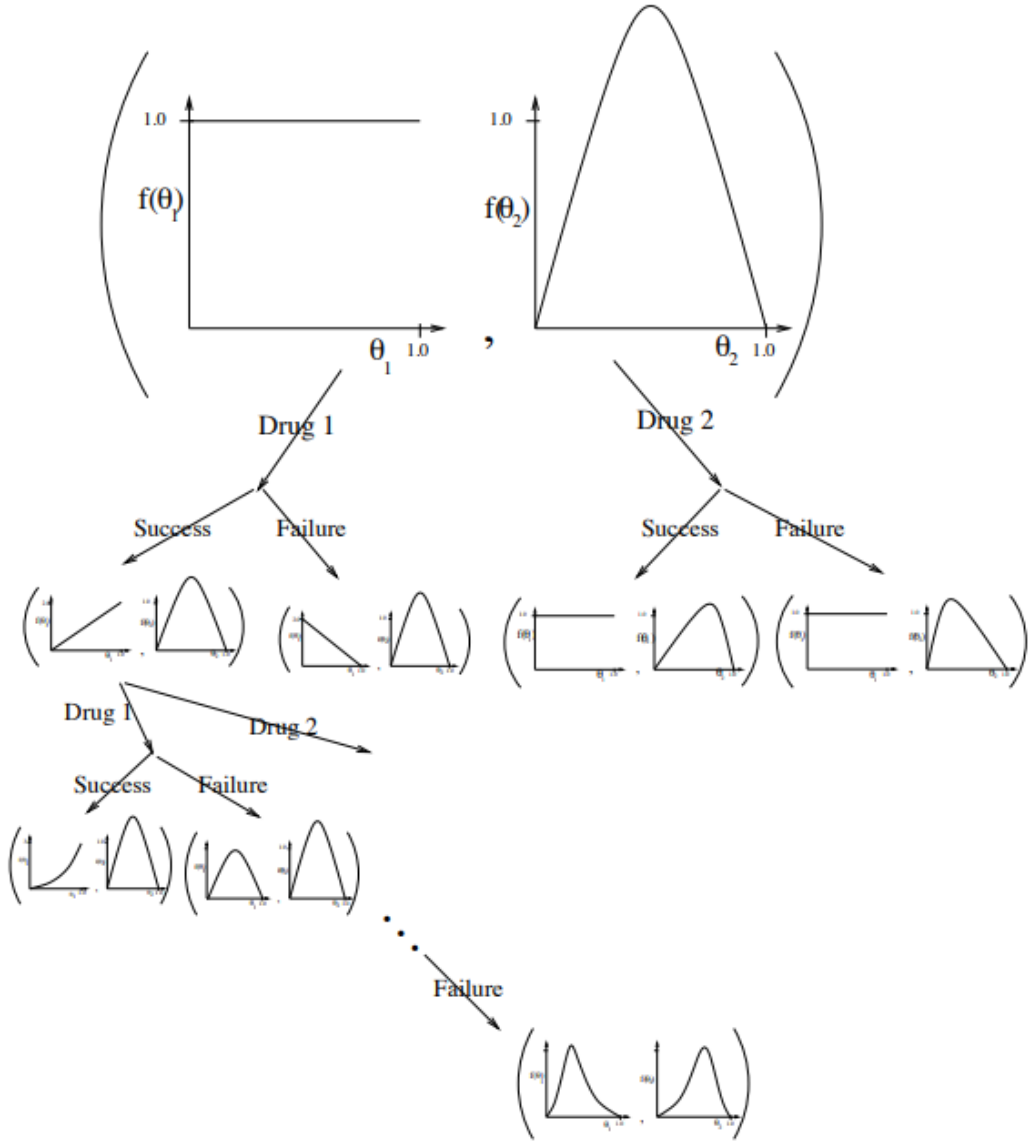
Resolviendo el problema

- Ahora tenemos un MDP infinito sobre estados de información
- Este MDP ´puede ser resuelto con aprendizaje por refuerzo
- RL libre de modelo (Q-learning)
- RL basado en modelo Bayesiano
  - Gittin índices
  - RL Bayes adaptativo
  - Encuentra el compromiso de explotación/explotación óptimo Bayesiano con respecto a la distribución a priori

Bandidos de Bernoulli Bayes adaptativo

- Iniciar con la a priori $Beta(\alpha_a, \beta_a)$ sobre la función de recompensa $\mathcal{R}^a$
- Cada vez que a es seleccionada, actualizar posteriori para $\mathcal{R}^a$
  - $Beta(\alpha_a + 1, \beta_a)$ si $r = 0$
  - $Beta(\alpha_a, \beta_a + 1)$ si $r = 1$
- Esto define una función de transición $\tilde{\mathcal{P}}$ para el MDP de Bayes adaptativo
- El estado de información (α, β) corresponde a un modelo de recompensa $Beta(\alpha, \beta)$
- Cada estado de transición corresponde a un modelo de actualización Bayesiano

$$p(\phi_i|r_i) = \frac{p(r_i|\phi_i)p(\phi_i)}{p(r_i)}$$



Índices de Gittins para bandidos de Bernoulli

- Los MDP de Bayes adaptativos puede ser resuelto por programación dinámica
- La solución exacta es normalmente intratable
- Idea: utilizar búsqueda basada en simulación
  - Búsqueda hacía adelante en el espacio del estado de información
  - Utilizar simulación desde el estado de información actual

Bandidos contextuales

- Un bandido contextual es una tupla $(\mathcal{A}, \mathcal{S}, \mathcal{R})$
- $\mathcal{A}$ es el conjunto de acciones (o brazos)
- $\mathcal{S} = \mathbb{P}[s]$ es la distribución sobre estados (o contextos)
- $r(s, a, r) = \mathbb{P}[r|s, a]$ es la distribución de probabilidad sobre recompensas
- En cada paso t
  - El ambiente genera estados $s_t \sim \mathcal{S}$
  - El agente selecciona la acción $a_t \in \mathcal{A}$
  - El ambiente genera la recompensa $r_t \sim r(s_t, a_t)$
- La meta es maximizar la recompensa cumulativa $\sum_{\tau=1}^{t} r_\tau$

Principios de explotación/explotación en MDPs

- Exploración ingenua
- Inicialización optimista
- Optimismo bajo incertidumbre
- Pareo de probabilidades
- Búsqueda en estados de información

Inicialización optimista: libre de modelo

- Inicializar la función de acción-valor $Q(s,a)$ a $\frac{r_{\max}}{1-\gamma}$
- Ejecutar su algoritmo favorito de RL
  - Monte-Carlo control
  - SARSA
  - Q-learning
  - …
- Motiva la exploración sistemática en estados y acciones

Inicialización optimista: basada en modelo

- Construir un modelo optimista del MDP
- Inicializar transiciones "optimistas" (recompensa a $r_{\max}$)
- Revolver el MDP usando el algoritmo favorito de planeación
  - Iteración de política
  - Iteración de valor
  - Búsqueda en árbol
  - …
- Motiva sistemáticamente la exploración de estados y acciones
- Ejemplo algoritmo RMax

Limites superiores de confianza: libre de modelo

- Maximizar UCB en la función acción-valor $Q^{\pi}(s,a)$

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a) + U(s_t, a)$$

  - Estimar la incertidumbre en la evaluación de política (fácil)
  - Ignorar la incertidumbre de la mejora de política
- Maximizar UCB en la función óptima acción-valor $Q^{*}(s,a)$

$$a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a) + U_1(s_t, a) + U_2(s_t, a)$$

  - Estimar la incertidumbre en la evaluación de política (fácil)
  - Estimar la incertidumbre de la mejora de política (difícil)

RL Bayesiana basada en modelo

- Mantener una distribución a posteriori sobre los modelos MDP
- Estimar transiciones y recompensas $p[\mathcal{P}, \mathcal{R}|h_t]$
- Utilizar la posteriori para guiar la exploración
  - Límites de confianza superior (UCB Bayesiano)
  - Pareo de probabilidades (muestreo de Thompson)

Muestreo de Thompson

- Implementa el pareo de probabilidades

$$\pi(s,a|h_t) = \mathbb{P}[Q^*(s,a) > Q^*(s,a'), \forall a' \neq a|h_t]$$

$$= \mathbb{E}_{(p,\mathcal{R}|h_t)}[1\left(a = \arg\max_{a\in\mathcal{A}} Q^*(a)\right)]$$

- Utiliza el teorema de Bayes para calcular la distribución a posteriori $p[p,\mathcal{R}|h_t]$
- Muestrear la distribución de recompensa $\mathrm{p},\mathcal{R}$ de la posteriori
- Utilizar el algoritmo favorito para obtener el $Q^*(s,a)$ del MDP
- Seleccionar la acción que maximiza el valor en la muestra $a_t = \arg\max_{a\in\mathcal{A}} Q^*(a)$

Búsqueda en estado de información en MDPs

- Los MDPs pueden ser aumentados para incluir el estado de información
- Ahora el conjunto de estados es $(s, \tilde{s})$
  - s es el conjunto de estados original del MDP
  - $\tilde{s}$ es la estadística de la historia
- Cada acción a provoca una transición
  - A un nuevo estado s' con probabilidad $p(s, a, s')$
  - A un nuevo estado de información $\tilde{s}'$
- Define un MDP $\widetilde{\mathcal{M}}$ en un espacio de información aumentado

$$\widetilde{\mathcal{M}} = (\tilde{\mathcal{S}}, \mathcal{A}, \tilde{p}, \tilde{\mathcal{R}}, \gamma)$$

MDPs por Bayes adaptativo

- La distribución a posterior sobre MDP es el estado de información
- $\tilde{s}_t = \mathbb{P}[p, \mathcal{R}|h_t]$
- El MDP aumentado sobre $(s, \tilde{s})$ es llamado MDP Bayes adaptativo
- Resolver el MDP para encontrar la exploración/explotación óptima (con respecto a la distribución a priori)
- Búsqueda basada en simulación puede ser efectiva

Conclusión

- Hemos cubierto los principios para exploración/explotación
  - Exploración ingenua
  - Inicialización optimista
  - Optimismo bajo incertidumbre
  - Pareo de probabilidades
  - Búsqueda en estados de información
- Cada principio fue desarrollado para bandidos
- Pero los mismo principios pueden ser utilizados para MDPs
- Para escalar podemos generalizar usando funciones de aproximación sobre estados y acciones

Lo que hemos visto

MDPs

Programación dinámica:

- evaluación de política
- Iteración de valor/política

Métodos aproximados

- Evaluación de política (MC/TD)
- MC/SARSA/Aprendizaje Q
- Política de gradiente

Métodos basados en modelo

- Modelos
- Planeación
- Arquitecturas
- Simulación

Exploración

- Bandido multi-brazo

Referencias

- Clases
- Sutton-Barto caps 1-13

Lo que sigue

Aprendizaje Q profundo	Optimización de política con Estrategias evolutivas	Implementaciones	RL en optimización	RL Bayesiano
RL para multi-agente	Control como inferencia	RL inverso	Meta aprendizaje	Aprendizaje multi-tarea y transferencia
	Teoría de la información	Control óptimo	RL con recompensas múltiples	

Para la otra vez…

- Q learning profundo