

# **APRENDIZAJE REFORZADO OFFLINE**

## ***EL7021: Seminario de robótica y sistemas autónomos***

Francisco Leiva<sup>2</sup>    Javier Ruiz-del-Solar<sup>1,2</sup>

<sup>1</sup>Departamento de Ingeniería Eléctrica, Universidad de Chile

<sup>2</sup>Advanced Mining Technology Center (AMTC), Universidad de Chile

Mayo, 2023

# Motivación

- ▶ El aprendizaje reforzado profundo (RL+DL) ha permitido resolver problemas en múltiples dominios.
- ▶ Un problema del deep RL es que el aprendizaje es “*online*”:
  - ▶ En muchos casos adquirir datos interactuando con el ambiente es poco práctico y/o costoso y/o riesgoso.
  - ▶ El uso de simuladores para sustituir la interacción real da lugar al problema del “*reality-gap*”.
  - ▶ Incluso si la interacción es posible, poder usar un *dataset* puede ser más práctico.

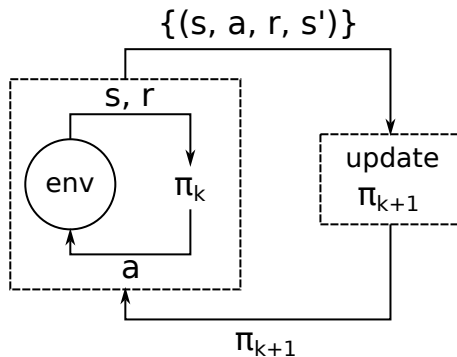
# Nota

- ▶ Esta clase está basada en Levine et al., «Offline reinforcement learning: Tutorial, review, and perspectives on open problems».
- ▶ Para más detalles sobre los contenidos expuestos, además de tópicos adicionales relacionados que no han sido considerados en esta presentación, revisar dicho estudio.

# Aprendizaje reforzado online

## *Online Reinforcement Learning*

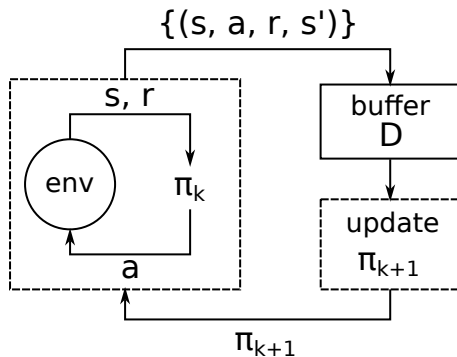
- ▶ Las experiencias generadas a través de interacciones agente-ambiente son utilizadas para actualizar la política.
- ▶ Tras actualizar  $\pi_k$  se deben generar nuevos datos para poder realizar una nueva actualización.
- ▶ Las experiencias para actualizar a  $\pi_k$  son generadas por  $\pi_k$ .



# Aprendizaje reforzado off-policy

## Off-Policy Reinforcement Learning

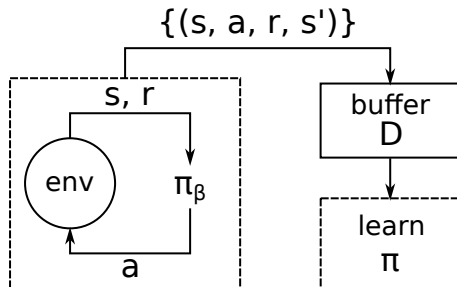
- ▶ Las interacciones agente-ambiente son guardadas en un *replay buffer*  $\mathcal{D}$ .
- ▶ Experiencias de  $\mathcal{D}$  son muestreadas para realizar actualizaciones de la política.
- ▶ De este modo  $\mathcal{D}$  presenta experiencias generadas por  $\pi_0, \dots, \pi_k$ .



# Aprendizaje reforzado offline

## Offline Reinforcement Learning

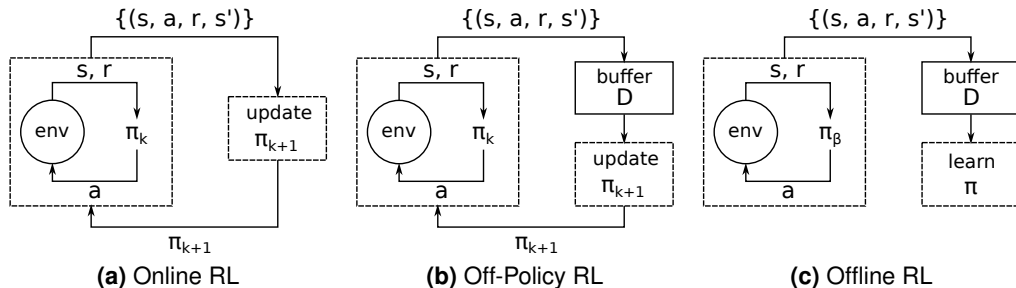
- ▶ Las experiencias para aprender una política son muestreadas de un *buffer*  $\mathcal{D}$ .
- ▶ Las experiencias en  $\mathcal{D}$  fueron generadas por una política  $\pi_\beta$  (que puede ser desconocida).
- ▶ No existe interacción agente-ambiente durante el aprendizaje de  $\pi$ .



# Aprendizaje reforzado offline

## Offline Reinforcement Learning

Comparación general entre aprendizaje reforzado online (on-policy), off-policy, y offline.<sup>1</sup>



<sup>1</sup>Sergey Levine et al. «Offline reinforcement learning: Tutorial, review, and perspectives on open problems». En: *arXiv preprint arXiv:2005.01643* (2020).

# Aprendizaje reforzado offline

## *Offline Reinforcement Learning*

Desafíos:

- ▶ Es muy difícil aprender puramente desde un *dataset*, sin interacciones en línea.
- ▶ *Distributional shift*.

Potencial:

- ▶ En SL: de datos a reconocedores de patrones.
- ▶ En offline RL: De datos a motores de decisión.



# Formulación del problema asociado a offline RL

Mismo objetivo que el del aprendizaje reforzado, es decir, aprender una política  $\pi$  para maximizar:

$$J_{\text{RL}}(\pi) = \mathbb{E}_{\tau \sim p_{\pi}(\tau)} \left[ \sum_{t=1}^T \gamma^{t-1} r(s_t, a_t) \right]$$

No obstante, no hay interacciones agente-ambiente en línea:

- ▶ Las experiencias provienen de un dataset fijo  $\mathcal{D} = \{(s_t^i, a_t^i, r_t^i, s_{t+1}^i)\}_{i=1}^N$ .

Se asume que en  $\mathcal{D}$  las tuplas  $(s, a)$  son tales que:

- ▶  $s \sim d^{\pi_{\beta}}(s)$
- ▶  $a \sim \pi_{\beta}(a|s)$

La idea es aprender una política que maximice el retorno esperado del agente al interactuar con el ambiente (aunque durante el aprendizaje no exista interacción).

# Formulación del problema asociado a offline RL

El problema de offline RL puede ser abordado (en principio) con cualquier clase de algoritmo visto con anterioridad:

- ▶ *Policy-based*
- ▶ *Value-based*
- ▶ *Actor-Critic*
- ▶ *Model-based*

Por ejemplo, aplicando Q-Learning para un *dataset* fijo.

El desafío, es que la aplicación de estos algoritmos “*online*” no siempre funcionan bien en un *setting offline*.

# Desafíos del offline RL

- ▶ Exploración:
  - ▶ El *dataset*  $\mathcal{D}$  es fijo, por lo que la exploración queda fuera de lo que un algoritmo de offline RL puede abordar.
- ▶ Consultas contrafactuales:
  - ▶ En offline RL se requiere formular y responder preguntas del tipo “¿Qué hubiese ocurrido si...?” pues se quiere obtener una política  $\pi$  que sea mejor que  $\pi_\beta$ , es decir, queremos aprender algo mejor que lo que se observa en  $\mathcal{D}$ .
  - ▶ Lo anterior se traduce en un problema de cambio de distribución.

# Algoritmos de offline RL

Se revisarán brevemente los siguientes enfoques:

- ▶ Offline RL usando *importance sampling*
- ▶ Offline RL usando programación dinámica aproximada (value-based / actor-critic)

# Off-Policy Evaluation usando *importance sampling*

Recordatorio de *importance sampling*:

$$\begin{aligned}\mathbb{E}_{x \sim p(x)}[f(x)] &= \int p(x)f(x)dx \\ &= \int \frac{q(x)}{q(x)}p(x)f(x)dx = \int q(x)\frac{p(x)}{q(x)}f(x)dx \\ &= \mathbb{E}_{x \sim q(x)}\left[\frac{p(x)}{q(x)}f(x)\right]\end{aligned}$$

Luego, si los datos de  $\mathcal{D}$  son generados a partir de  $\pi_\beta$ :

$$\begin{aligned}J_{\text{RL}}(\pi_\theta) &= \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}[R(\tau)] \\ &= \mathbb{E}_{\tau \sim p_{\pi_\beta}(\tau)}\left[\frac{p_{\pi_\theta}(\tau)}{p_{\pi_\beta}(\tau)}R(\tau)\right]\end{aligned}$$

# Off-Policy Evaluation usando *importance sampling*

Recordando que:

$$\underbrace{p_{\pi}(s_1, a_1, \dots, s_T, a_T)}_{p_{\pi}(\tau)} = p(s_1) \prod_{t=1}^T \pi(a_t | s_t) p(s_{t+1} | s_t, a_t)$$

Entonces:

$$\begin{aligned} J_{\text{RL}}(\pi_{\theta}) &= \mathbb{E}_{\tau \sim p_{\pi_{\beta}}(\tau)} \left[ \frac{p_{\pi_{\theta}}(\tau)}{p_{\pi_{\beta}}(\tau)} R(\tau) \right] \\ &= \mathbb{E}_{\tau \sim p_{\pi_{\beta}}(\tau)} \left[ \prod_{t=1}^T \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\beta}(a_t | s_t)} \sum_{t=1}^T \gamma^{t-1} r(s_t, a_t) \right] \\ &\approx \sum_{k=1}^N \left[ \prod_{t=1}^T \frac{\pi_{\theta}(a_t^{(k)} | s_t^{(k)})}{\pi_{\beta}(a_t^{(k)} | s_t^{(k)})} \left( \sum_{t=1}^T \gamma^{t-1} r(s_t^{(k)}, a_t^{(k)}) \right) \right] \end{aligned}$$

# Off-Policy Evaluation usando *importance sampling*

Observaciones:

- ▶ Muy alta varianza debido al producto de pesos de importancia.
- ▶ Normalizar por dicho producto da lugar a un estimador de menor varianza, pero que tiene sesgo (*weighted importance sampling estimator*).
- ▶ Una forma de mejorar el estimador es descartar los pesos de importancia asociados a tuplas  $(s_{t'}, a_{t'})$  para  $t' > t$  al considerar  $r_t$ , es decir:

$$J_{\text{RL}}(\pi_{\theta}) \approx \sum_{k=1}^N \left[ \sum_{t=1}^T \left( \prod_{t'=1}^t \frac{\pi_{\theta}(a_{t'}^{(k)} | s_{t'}^{(k)})}{\pi_{\beta}(a_{t'}^{(k)} | s_{t'}^{(k)})} \right) \left( \gamma^{t-1} r(s_t^{(k)}, a_t^{(k)}) \right) \right]$$

A este estimador se le conoce como “*per-decision importance sampling estimator*”.

# Off-policy Policy Gradient

Análogamente a la derivación anterior, se pueden computar gradientes de la política usando *importance sampling*, según:

$$\begin{aligned}\nabla_{\theta} J_{\text{RL}}(\pi_{\beta}) &= \mathbb{E}_{\tau \sim p_{\pi_{\beta}}(\tau)} \left[ \prod_{t=1}^T \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\beta}(a_t | s_t)} \left( \sum_{t=1}^T \nabla_{\theta} \log(\pi_{\theta}(a_t | s_t)) \right) A^{\pi}(s_t, a_t) \right] \\ &\approx \frac{1}{N} \sum_{k=1}^N \left[ \prod_{t=1}^T \frac{\pi_{\theta}(a_t^{(k)} | s_t^{(k)})}{\pi_{\beta}(a_t^{(k)} | s_t^{(k)})} \left( \sum_{t=1}^T \nabla_{\theta} \log(\pi_{\theta}(a_t^{(k)} | s_t^{(k)})) \right) A^{\pi}(s_t^{(k)}, a_t^{(k)}) \right]\end{aligned}$$

Al igual que en los casos anteriores, esta estimación de gradiente puede ser mejorada normalizando por el producto de los pesos de importancia (“*weighted importance sampling*” policy gradient) y dependiendo de la elección de  $A^{\pi}$ , formular un “*per-decision importance sampling estimator*”.



# Off-policy Policy Gradient

Una forma alternativa de derivar un gradiente *off-policy* consiste en considerar a la distribución de estados inducida por  $\pi_\beta$ , en lugar de  $\pi_\theta$ . Esto genera un sesgo en la estimación del gradiente, pero es útil en la práctica.

Con lo anterior, el objetivo de RL se puede reescribir como:

$$J_{\pi_\beta}(\pi_\theta) = \mathbb{E}_{s \sim d^{\pi_\beta}(s)} [V^\pi(s)]$$

- Notar que  $J_{\text{RL}}(\pi) \neq J_{\pi_\beta}(\pi_\theta)$ .
- Notar también que es posible calcular esperanzas bajo  $d^{\pi_\beta}(s)$  al muestrear datos desde  $\mathcal{D}$ .

# Off-policy Policy Gradient

Con lo anterior es posible derivar un gradiente de la política, definido por:

$$\begin{aligned}\nabla_{\theta} J_{\pi_{\beta}}(\pi_{\theta}) &= \mathbb{E}_{s \sim d^{\pi_{\beta}}(s)} \left[ \mathbb{E}_{a \sim \pi_{\theta}(a|s)} [Q^{\pi_{\theta}}(s, a) \nabla_{\theta} \log(\pi_{\theta}(a|s)) + \nabla_{\theta} Q^{\pi_{\theta}}(s, a)] \right] \\ &\approx \mathbb{E}_{s \sim d^{\pi_{\beta}}(s)} \left[ \mathbb{E}_{a \sim \pi_{\theta}(a|s)} [Q^{\pi_{\theta}}(s, a) \nabla_{\theta} \log(\pi_{\theta}(a|s))] \right]\end{aligned}$$

- Notar que para calcular este gradiente es necesario tener una estimación de  $Q^{\pi_{\theta}}(s, a)$  a partir de trayectorias *off-policy*.

# Métodos de offline RL basados en funciones de valor

- ▶ La idea general es aproximar una función de valor y emplear dicha función para derivar una política.
- ▶ Se incluirá en esta parte a algoritmos de tipo actor-crítico.
- ▶ Técnicamente, algoritmos de tipo *value-based* y actor-crítico vistos previamente pueden utilizarse en un contexto puramente offline (y en algunos casos esto funciona).
- ▶ No obstante, estos métodos sufren de varios problemas, principalmente asociados a *distributional shift*.
- ▶ En general existen dos mecanismos para aliviar lo anterior:
  - ▶ Métodos basados en la utilización de restricciones para la política.
  - ▶ Métodos que consideran la incerteza de las estimaciones de los valores Q.

# Distributional shift

- ▶ Ya se discutió que en despliegue, existe un claro problema de cambio de distribución en los estados.
- ▶ Los métodos basados en funciones de valor, no obstante, también sufren de cambio en la distribución de acciones durante el entrenamiento, al computar los objetivos en la regla de actualización de Bellman:
  - ▶  $\pi_\theta(a|s)$  produce acciones fuera de distribución, para la estimación de  $Q^{\pi_\theta}(s', a')$ .
  - ▶ Si  $\pi_\theta(a|s)$  produce acciones fuera de distribución que  $Q^{\pi_\theta}(s, a)$  considera que son de buena calidad, va a aprender a hacerlas.

# Métodos de offline RL basados en restricciones

- ▶ Estos métodos se basan en la utilización de restricciones para hacer que la política usada para calcular las acciones necesarias para el cálculo de los *target values* en la ecuación de Bellman, estén cerca de la distribución asociada a la política usada para generar los datos en  $\mathcal{D}$ .
  - ▶ Es decir, que  $\pi_\theta(a'|s')$  esté cerca de  $\pi_\beta(a'|s')$ .
- ▶ Notar que esto es suficiente, pues  $Q^{\pi_\theta}(s, a)$  se evalúa en los mismos estados en los que se entrena, es decir, en entrenamiento solo hay cambio de distribución en las acciones al calcular  $\mathbb{E}_{a' \sim \pi_\theta(a'|s')} [Q^{\pi_\theta}(s', a')]$ .
- ▶ Los métodos que fuerzan esto puede clasificarse fundamentalmente por dos características:
  - ▶ La métrica de distancia probabilística usada.
  - ▶ Como la restricción deseada es impuesta.

# Métodos de offline RL basados en restricciones

- ▶ Alternativamente a imponer restricciones sobre la política, es regularizar las actualizaciones sobre la función Q para evitar sobre estimación en pares estado-acción asociados a acciones fuera de distribución.
- ▶ Ventajas de este enfoque:
  - ▶ Aplicable a algoritmos basados en funciones de valor y actor-crítico.
  - ▶ No requiere una política explícita.
  - ▶ No requiere modelar  $\pi_\beta$ .

# Conservative Q-Learning (CQL)

- Dado un *batch* de transiciones  $\mathcal{B} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ , y definiendo la regla de actualización de la función Q como:

$$L(\mathcal{B}, \phi) = \sum_{i=1}^N \left( r_i + \gamma \max_{a' \in \mathcal{A}} Q_{\phi}(s'_i, a') - Q_{\phi}(s_i, a_i) \right)^2$$

- En CQL se busca modificar este objetivo agregando una penalización  $\mathcal{C}(\mathcal{B}, \phi)$  para obtener un nuevo objetivo de minimización, dado por:

$$\tilde{L}(\mathcal{B}, \phi) = L(\mathcal{B}, \phi) + \alpha \mathcal{C}(\mathcal{B}, \phi)$$

- Ejemplo de penalización:

$$\mathcal{C}(\mathcal{B}, \phi) = \mathbb{E}_{s \sim \mathcal{B}} \left[ \mathbb{E}_{a \sim \mu(a|s)} [Q_{\phi}(s, a)] \right]$$

# Conservative Q-Learning (CQL)<sup>2</sup>

- ▶ Con una correcta elección de  $\mu(a|s)$ , sería posible reducir Q-values altos, y mantener aproximaciones razonables para aquellos asociados a acciones que están dentro de la distribución.
- ▶ La elección  $\mu(a|s) \propto \exp(Q_\phi(s, a))$  permite que la estimación regularizada de la función Q sea una cota inferior de la función Q real.
- ▶ En la práctica la elección anterior de  $\mathcal{C}(\mathcal{B}, \phi)$  puede resultar en subestimaciones muy grandes, por lo que otra opción es definirla de manera diferente:

$$\mathcal{C}(\mathcal{B}, \phi) = \mathbb{E}_{s \sim \mathcal{B}} [\mathbb{E}_{a \sim \mu(a|s)} [Q_\phi(s, a)]] - \mathbb{E}_{(s,a) \sim \mathcal{B}} [Q_\phi(s, a)]$$



- ▶ Intuitivamente, esto permite que valores Q altos puedan ser asignados a pares  $(s, a)$  que estén dentro de la distribución.

---

<sup>2</sup>Aviral Kumar et al. «Conservative q-learning for offline reinforcement learning». En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 1179-1191.



# Referencias

-  Kumar, Aviral et al. «Conservative q-learning for offline reinforcement learning». En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 1179-1191.
-  Levine, Sergey et al. «Offline reinforcement learning: Tutorial, review, and perspectives on open problems». En: *arXiv preprint arXiv:2005.01643* (2020).