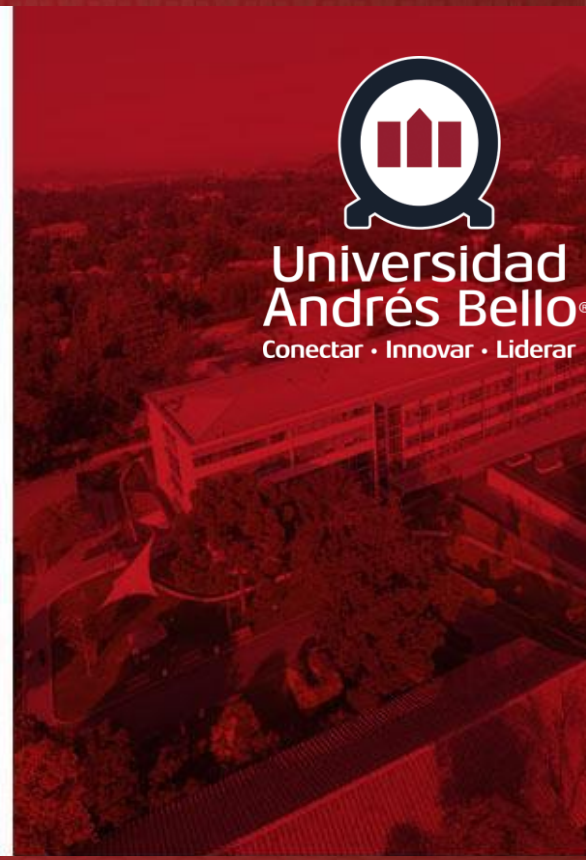
Escuela Internacional de Primavera sobre Entornos Ubicuos y Aplicaciones de Robots Sociales



Tutorial #2 – Aprendizaje Reforzado con Gym (material en www.miguelsolis.info)

Dr. Miguel A. Solis
Universidad Andrés Bello
16 de octubre de 2023

Áreas en robótica

Áreas en robótica

- Robótica móvil
- Robótica bioinspirada
- Robótica cognitiva
- Robótica evolutiva
- Interacción humano-robot
- Microrobótica
- Robótica tele-operada
- Robótica de enjambre

Percepción y acción

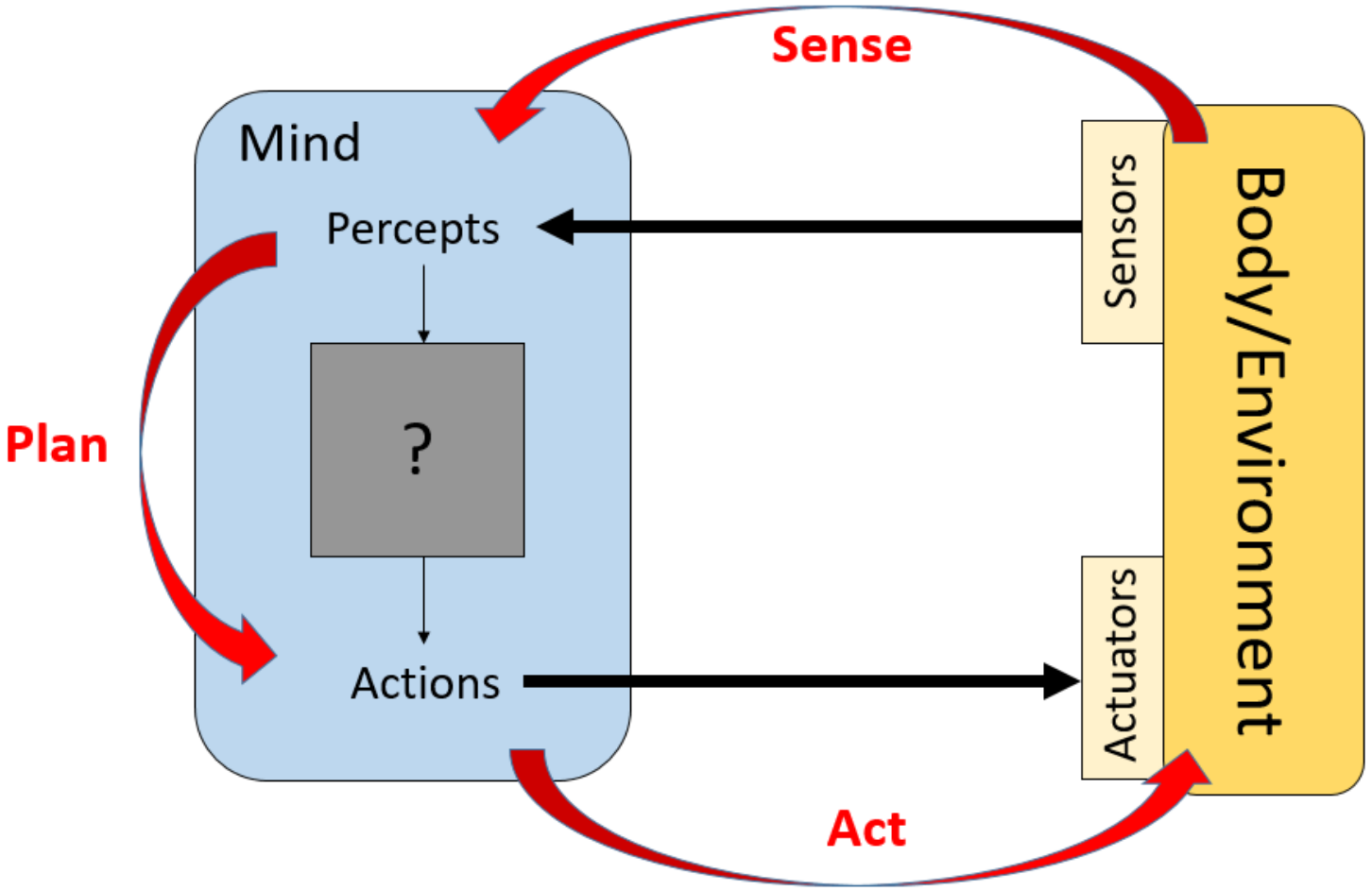


Foto extraída de “Robotic Systems” de Kris Hauser, University of Illinois at Urbana-Champaign.

Toma de decisiones

Ejemplos de formas de interactuar entre el agente y el entorno:

- de manera reactiva: reaccionando ante ciertos estímulos del entorno. La manera más simple de su implementación es a través de una máquina de estados finitos. (*Si ocurre A estando en X, entonces ejecuto B y quedaré en Y*).

Toma de decisiones

Universidad Andrés Bello

Ejemplos de formas de interactuar entre el agente y el entorno:

- de manera inteligente: existen técnicas de inteligencia artificial que permiten generar cierto comportamiento en base a heurísticas.
- con aprendizaje incremental: existen técnicas computacionales para dotar de aprendizaje automático a cierto agente, que puede ser un dispositivo con actuaciones físicas.

UBIBOT

Toma de decisiones

- Técnicas basadas en sistema de reglas (if-else , navegación reactiva).
- Métodos estadísticos.
- Métodos basados en modelos.
- Métodos libres de modelos.

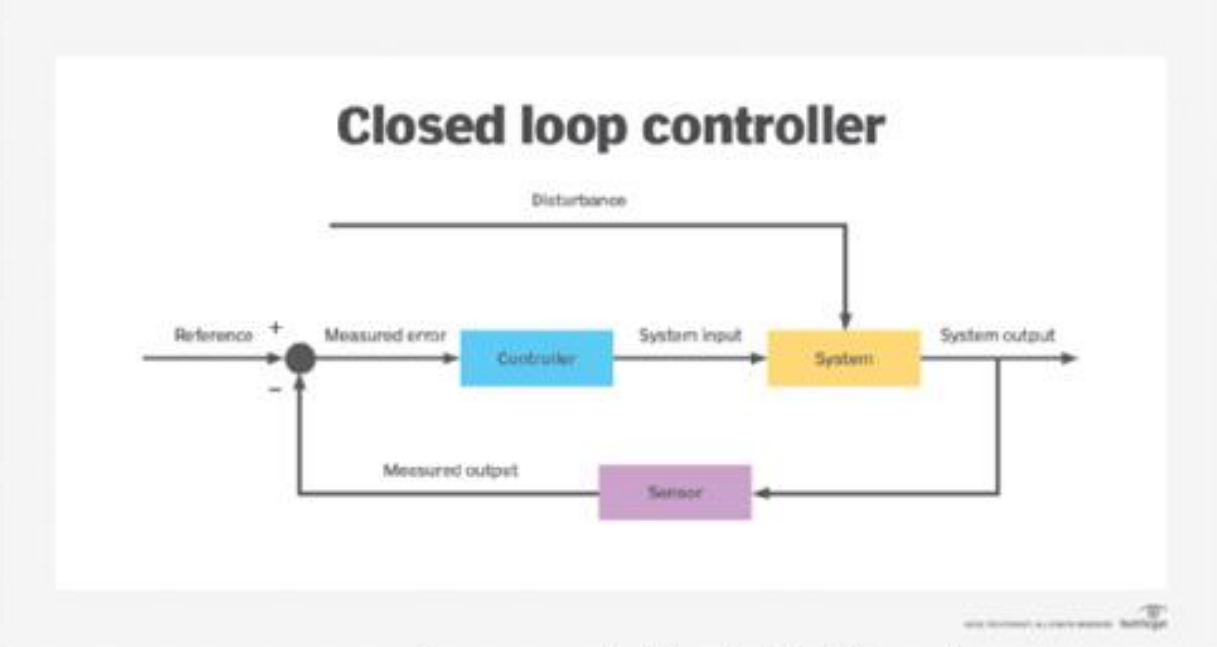
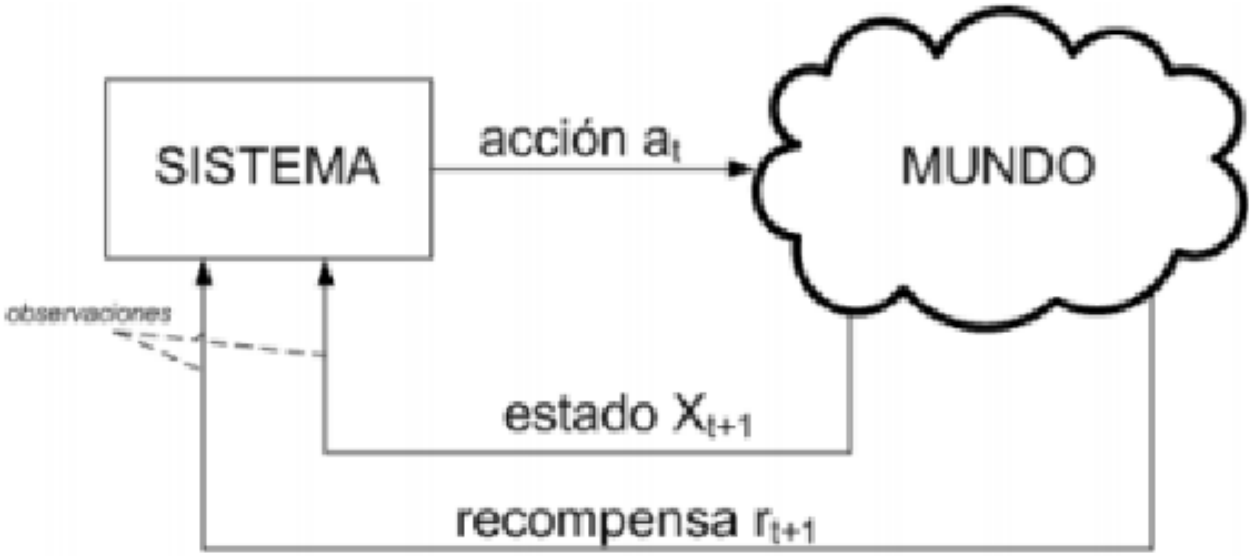


Imagen extraída de TechTarget

Aprendizaje reforzado



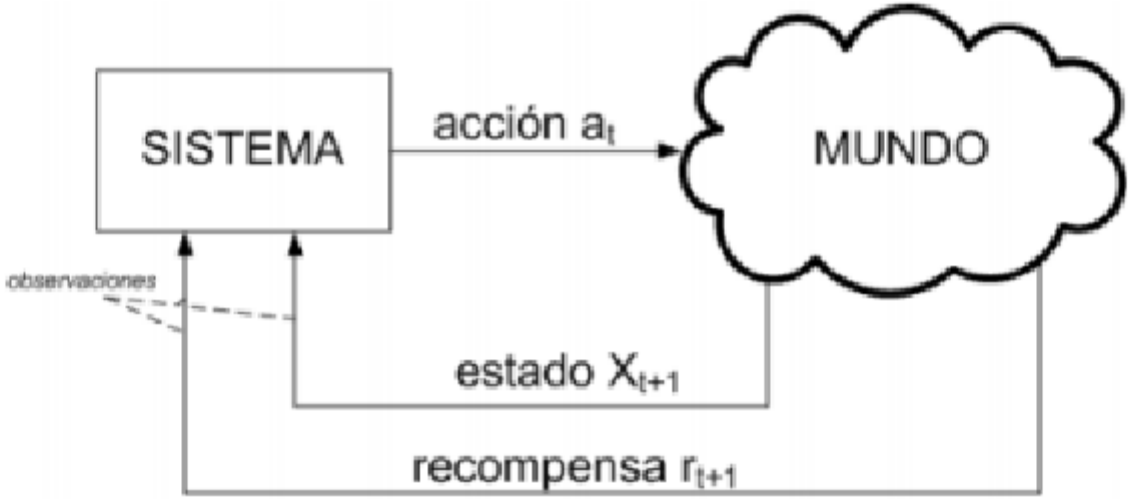
- Aprende de interacciones con el entorno.
- Aplicaciones:
  - Problemas de decisión secuencial.
  - Sistemas adaptivos.

Aprendizaje reforzado y conductivismo

- Edward L. Thorndike - Animal Intelligence: An experimental study of the associate processes in animals (1898).
- El animal modifica su conducta según interacción de prueba y error con el medio ambiente.

Aprendizaje reforzado y conductivismo

El **agente** interactúa con el entorno a través de percepciones y acciones.



- Recibe como entrada (percibe), el estado actual del entorno, *s*.
- Luego, genera una acción (ejecuta) *a* como salida.
- Recibe una señal de refuerzo *r* (recompensa).

Proceso de Markov

- Un problema de aprendizaje reforzado se formula de manera formal mediante un MDP.
- MDP: Markov Decision Process.

Un estado s_k se dice que obedece a un Proceso de Markov (de primer orden) si y sólo si:

$$Pr\{s_{k+1}|s_k\} = Pr\{s_{k+1}|s_1, ..., s_k\}.$$

Elementos en un problema de aprendizaje reforzado

Un problema de aprendizaje reforzado, formulado como un MDP está compuesto por la tupla (S,A,T,R) donde

- S: conjunto de estados
- A: conjunto de acciones
- $T : S \times A \times S \rightarrow [0, 1]$ (*función de transición, desconocida*)
- R: $S \times A \times S \rightarrow \mathbb{R}$ (*función de recompensas*)
- $\pi : S \rightarrow A$ (*política*)

Elementos en un problema de aprendizaje reforzado

Universidad Andrés Bello

- Estado en instante k:

$$s_k \in \mathcal{S}$$

- Acción ejecutada en instante k:

$$a_k \in \mathcal{A}$$

- Función de transición de estado:

$$s_{k+1} = T(s_k, a_k)$$

UBIBOT

Recompensas

$$r_k = R(s_k, a_k, s_{k+1})$$

- $r_k > 0$
- $r_k = 0$
- $r_k < 0$

- Política π:

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

- Una política es óptima si maximiza la recompensa a **largo plazo**.

Retorno

Función de valor:

$$V^{\pi}(s_k) = r_k + \gamma r_{k+1} + \gamma^2 r_{k+2} + \dots$$
$$= \sum_{i=0}^{\infty} \gamma^i r_{k+i}.$$

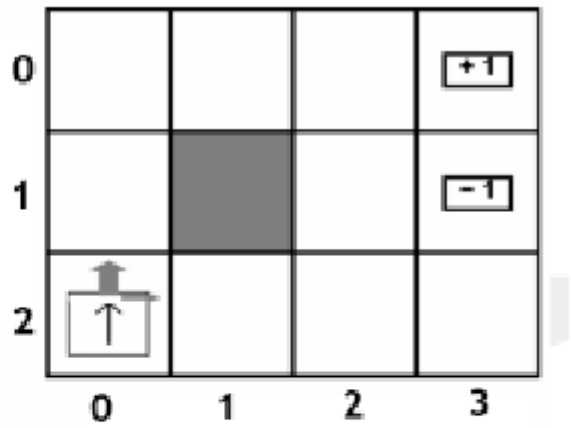
Restricciones:

- $0 \leq \gamma < 1$.
- r_k acotado.

Una política π^* es óptima si la función de valor obtenida para esa política es óptima:

$$V^*(s) = V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s, \pi$$

Ejemplo: Grid World



- Estados: ubicación en la grilla.
- Acciones: arriba, izquierda, derecha, abajo.
- Recompensa: +1, -1, -0.1.

Algoritmos clásicos

Universidad Andrés Bello

Provenientes de Programación Dinámica:

- Value Iteration
- Policy Iteration
- Temporal Difference
  - Q-Learning
  - SARSA

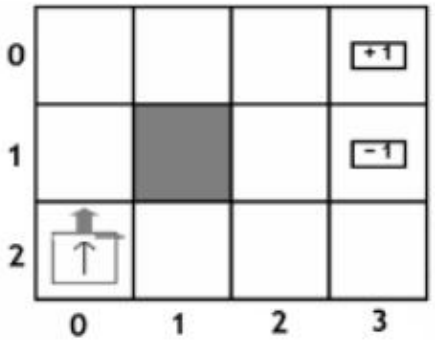
UBIBOT

Value Iteration

Consiste en realizar k iteraciones, hasta que $V_k(s) - V_{k-1}(s)$ es suficientemente pequeño, con actualización según:

$$V_k(s) = \max_a \sum_{s'} Pr\{s', s, a\} \cdot (R(s', s, a) + \gamma V_{k-1}(s'))$$

Ejemplo



- Después de 1 iteración:

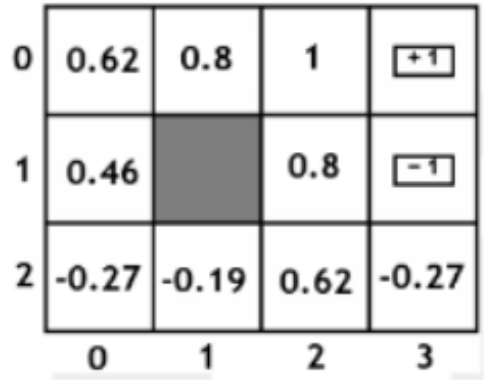
0	-0.1	-0.1	1	+1
1	-0.19		-0.1	-1
2	-0.1	-0.1	-0.19	-0.1
	0	1	2	3

- Después de 2 iteraciones:

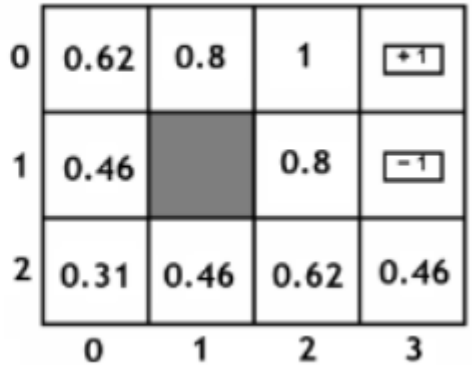
0	0.62	0.8	1	+1
1	0.46		0.8	-1
2	-0.27	-0.19	0.62	-0.27
	0	1	2	3

Ejemplo

- Después de 2 iteraciones:



- En convergencia:



Policy Iteration

Universidad Andrés Bello

Función de valor:

$$V_k(s) = \max_a \sum_{s'} Pr\{s', s, a\} \cdot (R(s', s, a) + \gamma V_{k-1}(s'))$$

- La política no necesariamente será ejecutar la acción que maximiza la función de valor.
  - comportamiento *greedy*.
  - comportamiento ϵ-*greedy*.

UBIBOT

Ejemplo

- De todas formas se evalua la función de valor $V(s)$.
- Se encuentra de forma directa la política.

0	der.	der.	der.	meta
1	arr.		arr.	prohib
2	arr.	der.	arr.	izq.
	0	1	2	3

Valor Q

- Así como $V(s)$ corresponde a la función que valoriza el estado, $Q(s,a)$ corresponde a la función que valoriza el tomar cierta acción en ese estado.
- Para una política óptima π^*, se cumple

$$Q^{\pi^*}(s,a) \geq Q^{\pi}(s,a) \quad \forall s, a, \pi$$

Q-learning

- Consiste en iterar sobre cada par (estado,acción), para α y γ fijos.
- Regla de actualización:

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha\left(r_{k+1} + \gamma \max_a Q(s_{k+1,a})\right)$$

Suponiendo $\hat{Q} = Q^*$, entonces la acción óptima para cada estado se puede obtener maximizando:

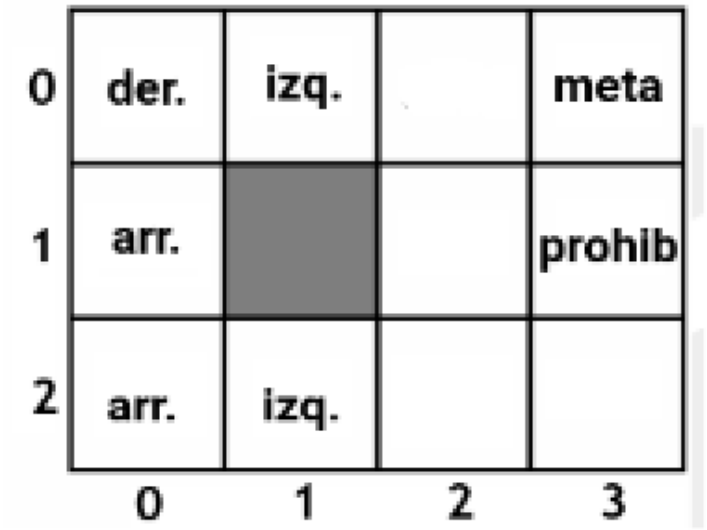
$$\pi^*(s_k) = arg\max_{a_k} Q^*(s_k, a_k)$$

Imagen extraída de “Introduction to Q-learning with OpenAI Gym”, Gelana Tostaeva, Medium.

Ejemplo

- No se realizan iteraciones sobre todo el espacio de estados, sólo cuando el estado se visita.
- Este ejemplo es repetitivo (cuando se llega a un estado final, vuelve al inicio).
- **atención** con los óptimos locales.

Estancamiento en óptimo local:



Ejemplo

Estancamiento en óptimo local:

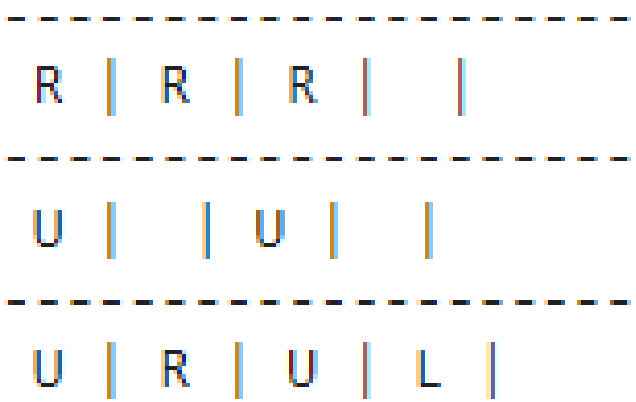
0	der.	izq.		meta
1	arr.			prohib
2	arr.	izq.		
	0	1	2	3

Incorporando exploración

0	der.	der.	der.	meta
1	arr.			prohib
2	arr.	izq.		
	0	1	2	3

Python Notebook

16102023_RL_Parte1.ipynb



Abrir con visor de Python Notebook o colab.research.google.com

Después de almuerzo:

Aprendizaje reforzado profundo

(Aula N401)