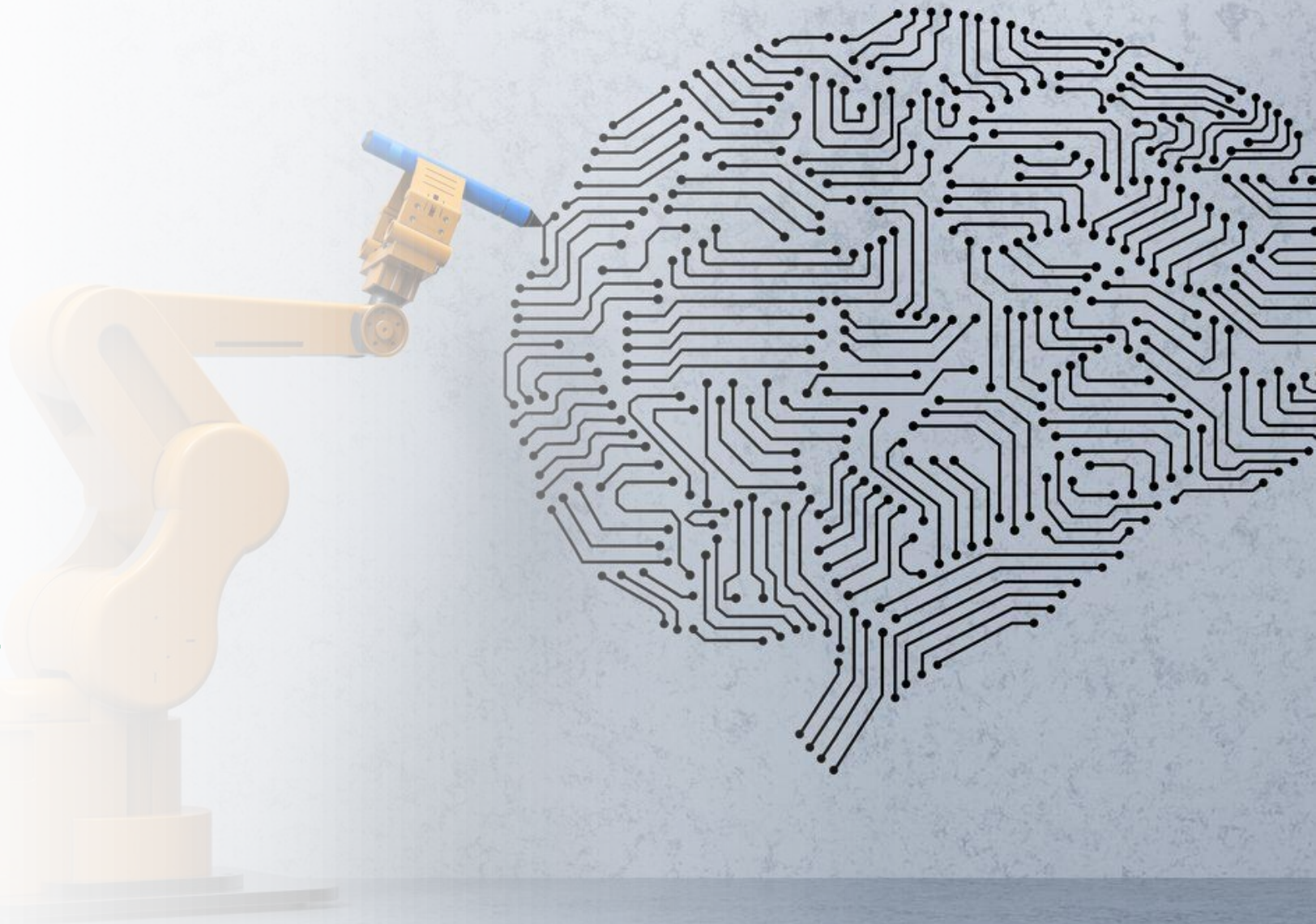
Aprendizaje por refuerzo

Clase 24: multi-modelo (ensambles)



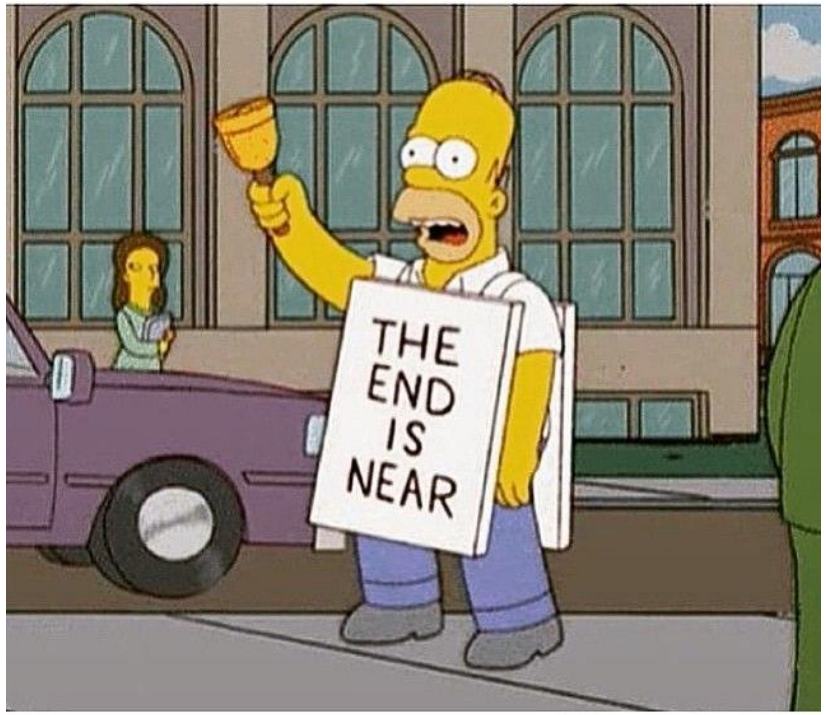
Antes de empezar...

ABRIL 2023

L	M	M	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

MAYO 2023

L	M	M	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	▼	27	28
29	30	31				



Antes de empezar…

- Tarea 4
- Proyecto

Para el día de hoy…

- Ensambles para RL

¿Cuál método deberíamos utilizar?

DQN

DDQN

A2C

A3C

Estrategias evolutivas

Algún otro


¿Por qué no usar todos?

Ensambles

Muy utilizados en aprendizaje supervisado

Entrenan diversos métodos de aprendizaje para resolver el mismo problema

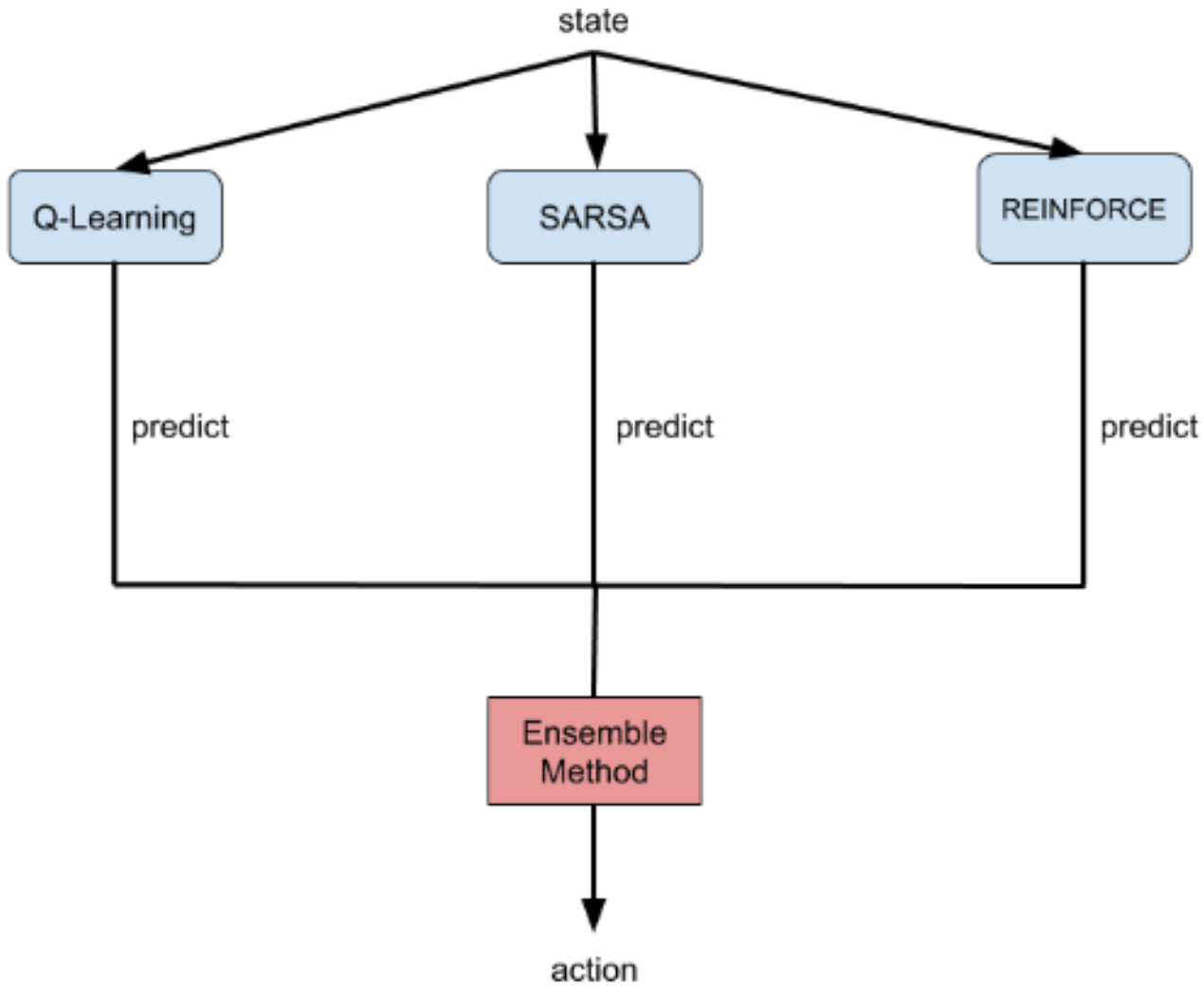
También son llamados aprendizaje basado en comité

¿A quién le creemos?

Algoritmo	Acción	Probabilidad
DDQN	R, U, L, D	0.7, 0.2, 0.08, 0.02
A2C	U, R, D, L	0.5, 0.3, 0.15, 0.05
A3C	R, D, L, U	0.27, 0.26, 0.25, 0.22
Estrategia evolutiva	D, L, R, U	.94, 0.03, 0.02, 0.01

“Tantas personas no pueden estar equivocadas”

- Para que una población se vuelva “sabia” se necesita
  - Diversidad de opinión
  - Independencia
  - Descentralización
  - Agregación



La idea

- La habilidad de generalizar un ensamble es típicamente más fuerte que los algoritmos base
- Son capaces de potenciar métodos de aprendizaje débiles
- Incluso, son capaces de llevar a ser métodos de aprendizaje fuertes

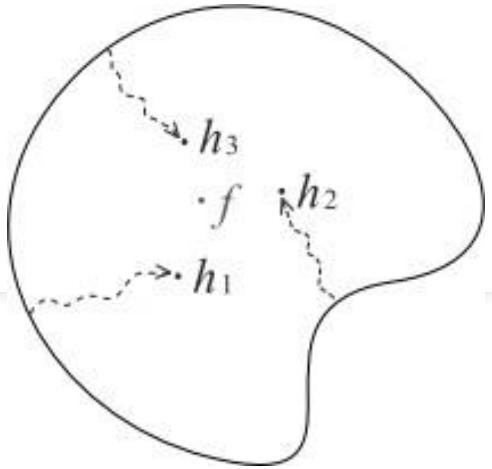
El algoritmo base

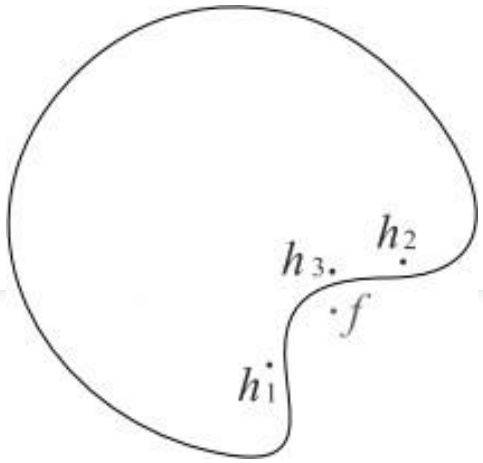
- Un ensamble normalmente consiste en dos pasos
  - Generar métodos de aprendizaje base
  - Combinarlos
- Para obtener un buen ensamble los algoritmos base deben ser tan precisos como sea posible y diversos
- El costo computacional no suele ser más grande que el de los algoritmos base

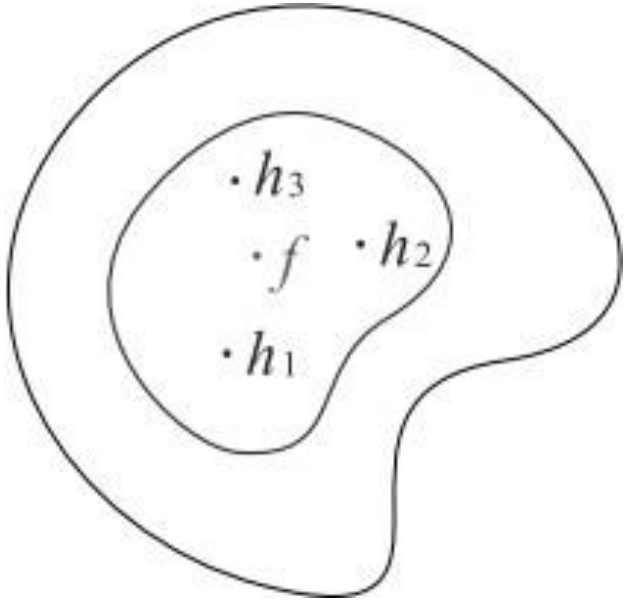
Familias de métodos

Combinación de métodos

- Generamos un conjunto de algoritmos base y en lugar de encontrar el “mejor” los combinamos para obtener una generalización fuerte.
- Normalmente lo hacemos por las siguientes razones
  - Problemas estadísticos
  - Problemas computacionales
  - Problemas de representación







Teoría de votaciones para n candidatos

- Cada algoritmo expresa sus preferencias sobre las acciones

9	0	0	4	0	7
A_1	A_1	A_2	A_2	A_3	A_3
A_2	A_3	A_1	A_3	A_1	A_2
A_3	A_2	A_3	A_1	A_2	A_1

Eligiendo el ganador: por mayoría

- Hacer el conteo de las veces que cada candidato obtiene el primer lugar
- Elegir a aquel candidato que haya obtenido la puntuación más alta
- Ganador: A_1

9	0	0	4	0	7
A_1	A_1	A_2	A_2	A_3	A_3
A_2	A_3	A_1	A_3	A_1	A_2
A_3	A_2	A_3	A_1	A_2	A_1

	1er
A_1	9
A_2	4
A_3	7

Eligiendo el ganador: por conteo de Borda

- Se asignan puntos desde $n, n-1, \ldots, 1$ a cada voto de mayor a menor y suman
- El ganador es aquel con mayor suma
- Ganador: A_3

9	0	0	4	0	7
A_1	A_1	A_2	A_2	A_3	A_3
A_2	A_3	A_1	A_3	A_1	A_2
A_3	A_2	A_3	A_1	A_2	A_1

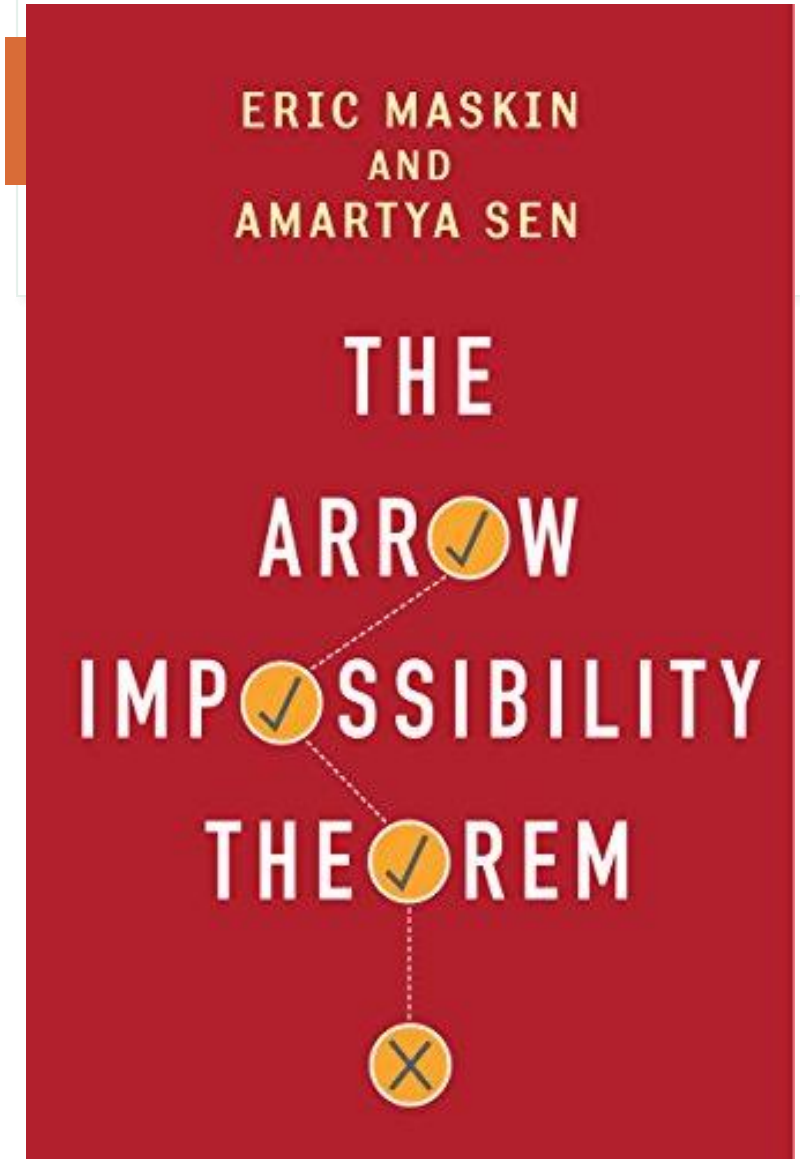
	1er	2do	3er	Borda fórmula	$\sum$
A_1	9	0	11	$3(9)+2(0)+1(11)$	38
A_2	4	11	5	$3(4)+2(11)+1(5)$	39
A_3	7	9	4	$3(7)+2(9)+1(4)$	43

Eligiendo el ganador: ganador de Condorcet

- Se hace una matriz $n \times n$
- Se rellena con las veces que un candidato A_i es preferido a A_j
- Un ganador de Condorcet es aquel que gana todos sus "encuentros" uno a uno
- Ganador: A_2

9	0	0	4	0	7
A_1	A_1	A_2	A_2	A_3	A_3
A_2	A_3	A_1	A_3	A_1	A_2
A_3	A_2	A_3	A_1	A_2	A_1

	A_1	A_2	A_3
A_1	–	9	9
A_2	11	–	13
A_3	11	7	–



Los buenos deseos

- No dictadura: los deseos de múltiples votantes deben ser considerados
- Eficiencia de Pareto: si todos prefieren a A sobre B entonces A debe ganar
- Independencia de alternativas irrelevantes: si se remueve un candidato, no debe alterar el orden del resto
- Universalidad: se deben considerar las preferencias de todos los votantes
- Monotonía: si un votante promueve su preferencia por alguna opción, el resultado no debe degradar esa opción

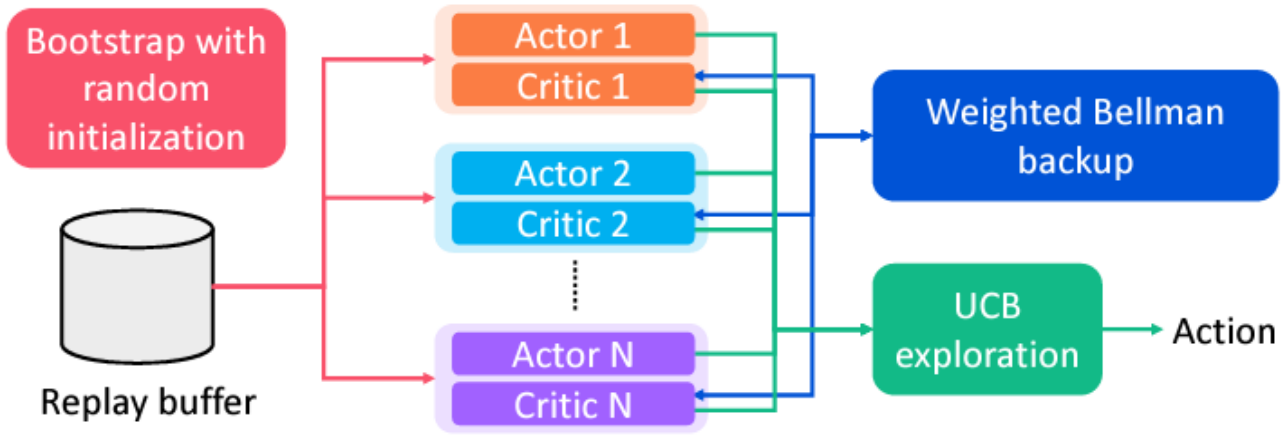
El teorema de imposibilidad de Arrow nos dice que es imposible formular un ordenamiento sin violar alguno de los principios anteriores

Algunas notas

- Con voto por mayoría puede no respetar los deseos de la mayoría
- Con conteo de Borda puede no respetarse la independencia de alternativas irrelevantes
- Con Condorcet pueden no respetarse los deseos de la mayoría
- Intentar buscar un ganador de Condorcet
- En caso de no existir elegir el ganador por conteo de Borda

SUNRISE

- Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning
- Lee et al. (2020)
- Ensamble para métodos fuera de política
- Realiza un ensamble de la función Q con pesos de acuerdo a la desviación estándar de la función Q
- Utiliza UCB para exploración



Resultados

							SUNRISE
Game	Human	Random	SimPLe	CURL	DrQ	Rainbow	SUNRISE
Alien	7127.7	227.8	616.9	558.2	761.4	789.0	872.0
Amidar	1719.5	5.8	88.0	142.1	97.3	118.5	122.6
Assault	742.0	222.4	527.2	600.6	489.1	413.0	594.8
Asterix	8503.3	210.0	1128.3	734.5	637.5	533.3	755.0
BankHeist	753.1	14.2	34.2	131.6	196.6	97.7	266.7
BattleZone	37187.5	2360.0	5184.4	14870.0	13520.6	7833.3	15700.0
Boxing	12.1	0.1	9.1	1.2	6.9	0.6	6.7
Breakout	30.5	1.7	16.4	4.9	14.5	2.3	1.8
ChopperCommand	7387.8	811.0	1246.9	1058.5	646.6	590.0	1040.0
CrazyClimber	35829.4	10780.5	62583.6	12146.5	19694.1	25426.7	22230.0
DemonAttack	1971.0	152.1	208.1	817.6	1222.2	688.2	919.8
Freeway	29.6	0.0	20.3	26.7	15.4	28.7	30.2
Frostbite	4334.7	65.2	254.7	1181.3	449.7	1478.3	2026.7
Gopher	2412.5	257.6	771.0	669.3	598.4	348.7	654.7
Hero	30826.4	1027.0	2656.6	6279.3	4001.6	3675.7	8072.5
Jamesbond	302.8	29.0	125.3	471.0	272.3	300.0	390.0
Kangaroo	3035.0	52.0	323.1	872.5	1052.4	1060.0	2000.0
Krull	2665.5	1598.0	4539.9	4229.6	4002.3	2592.1	3087.2
KungFuMaster	22736.3	258.5	17257.2	14307.8	7106.4	8600.0	10306.7
MsPacman	6951.6	307.3	1480.0	1465.5	1065.6	1118.7	1482.3
Pong	14.6	-20.7	12.8	-16.5	-11.4	-19.0	-19.3
PrivateEye	69571.3	24.9	58.3	218.4	49.2	97.8	100.0
Qbert	13455.0	163.9	1288.8	1042.4	1100.9	646.7	1830.8
RoadRunner	7845.0	11.5	5640.6	5661.0	8069.8	9923.3	11913.3
Seaquest	42054.7	68.4	683.3	384.5	321.8	396.0	570.7
UpNDown	11693.2	533.4	3350.3	2955.2	3924.9	3816.0	5074.0

Forma normal de un juego

- Un conjunto de jugadores/agentes $\mathcal{I}$
- Un conjunto de acciones conjuntas $a = (a_i)$, $a_i \in \mathcal{A}$, es la acción del agente $i \in \mathcal{I}$
- Recompensa/pagos $r_i(a)$ es la recompensa recibida por el agente i con la acción a
- Cuando un juego en su forma normal se repite un número de veces (finito/infinito) se llama juego repetido

Estrategias

- Estrategia/política: $\pi_i \in \Delta(\mathcal{A}_i)$: $\pi_i(a_i)$ es la probabilidad que un agente i seleccione la acción a_i
  - Pura (determinista): solo se juega una acción
  - Mixta (estocástica): una distribución sobre un conjunto de acciones
- Perfil: una estrategia para cada jugador $\pi = (\pi_i)_i$
- Cada jugador desea maximizar su pago/recompensa
- El pago esperado de cada jugador i cuando se usa un perfil π

$$r_i(\pi) = \sum_a r_i(a) \prod_{j \in \mathcal{J}} \pi_j(a_j)$$

Un caso especial: juegos de dos jugadores

- El pago de juegos de dos jugadores puede ser representado con una matriz
- Dilema del prisionero: cada agente elige cooperar o acusar al otro

		Bob	
		cooperate	defect
Alex	cooperate	$1, 1$	$-1, 2$
	defect	$2, -1$	$0, 0$

Estrategia dominante

- Una estrategia dominante π_i para un jugador i es una estrategia que es la mejor respuesta a todo π_{-i}
- $r_i(\pi_i, \pi_{-i}) \geq r_i(\tilde{\pi}_i, \pi_{-i}), \forall \tilde{\pi}, \pi_{-i}$
- En un equilibrio, cada jugador adopta una estrategia dominante
- Es posible que no exista una estrategia dominante ni un equilibrio

		Bob	
		cooperate	defect
Alex	cooperate	$1, 1$	$-1, 2$
	defect	$2, -1$	$0, 0$

Equilibrio de Nash

- En un equilibrio de Nash π^*, ningún jugador puede mejorar su recompensa esperada cambiando su política, si el resto mantiene la suya
- π^* es la mejor respuesta para cada agente i si los otros agentes se quedan con π^*_{-1}
- Para cada agente

$$r_i(\pi^*) \geq r_i(\pi_i, \pi^*_{-1}) \; \forall \pi_i$$

- Toda estrategia dominante es un equilibrio de Nash

Para la otra vez…

- RL multi-agente