Aprendizaje por refuerzo multiagente

Sistemas Inteligentes Distribuidos

Sergio Alvarez
Javier Vázquez

Bibliografía

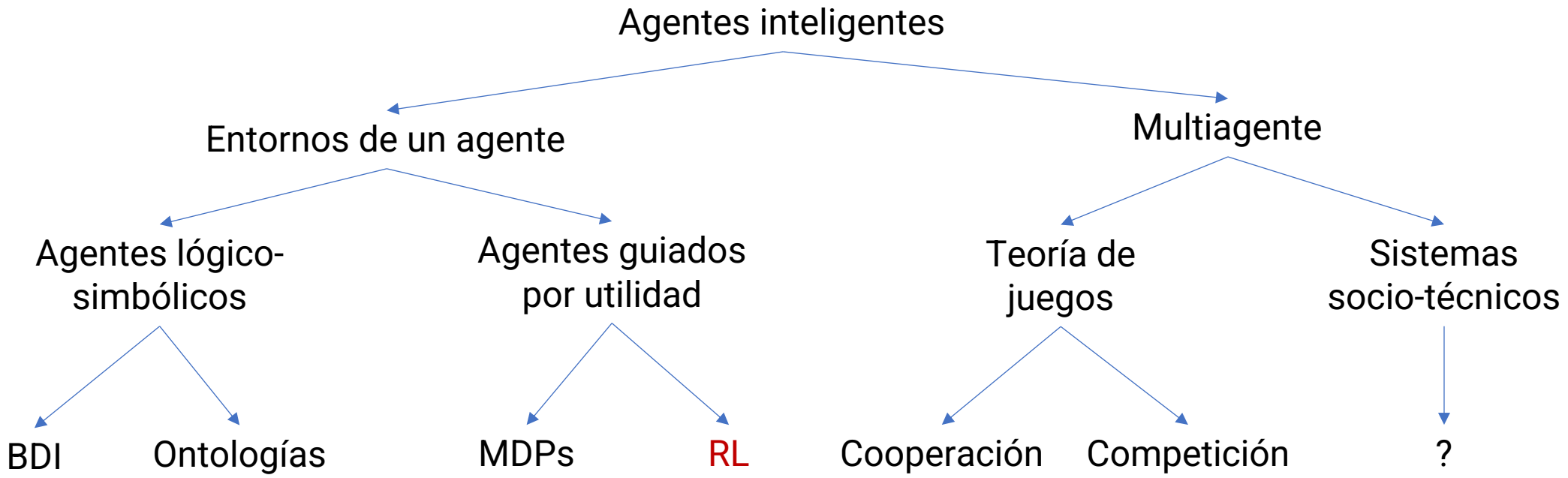
- *Multiagent systems: algorithmic, game-theoretic, and logical foundations* (Shoham & Leyton-Brown), cap. 3, 4, 5, 6

Introducción

Aprendizaje por refuerzo multiagente

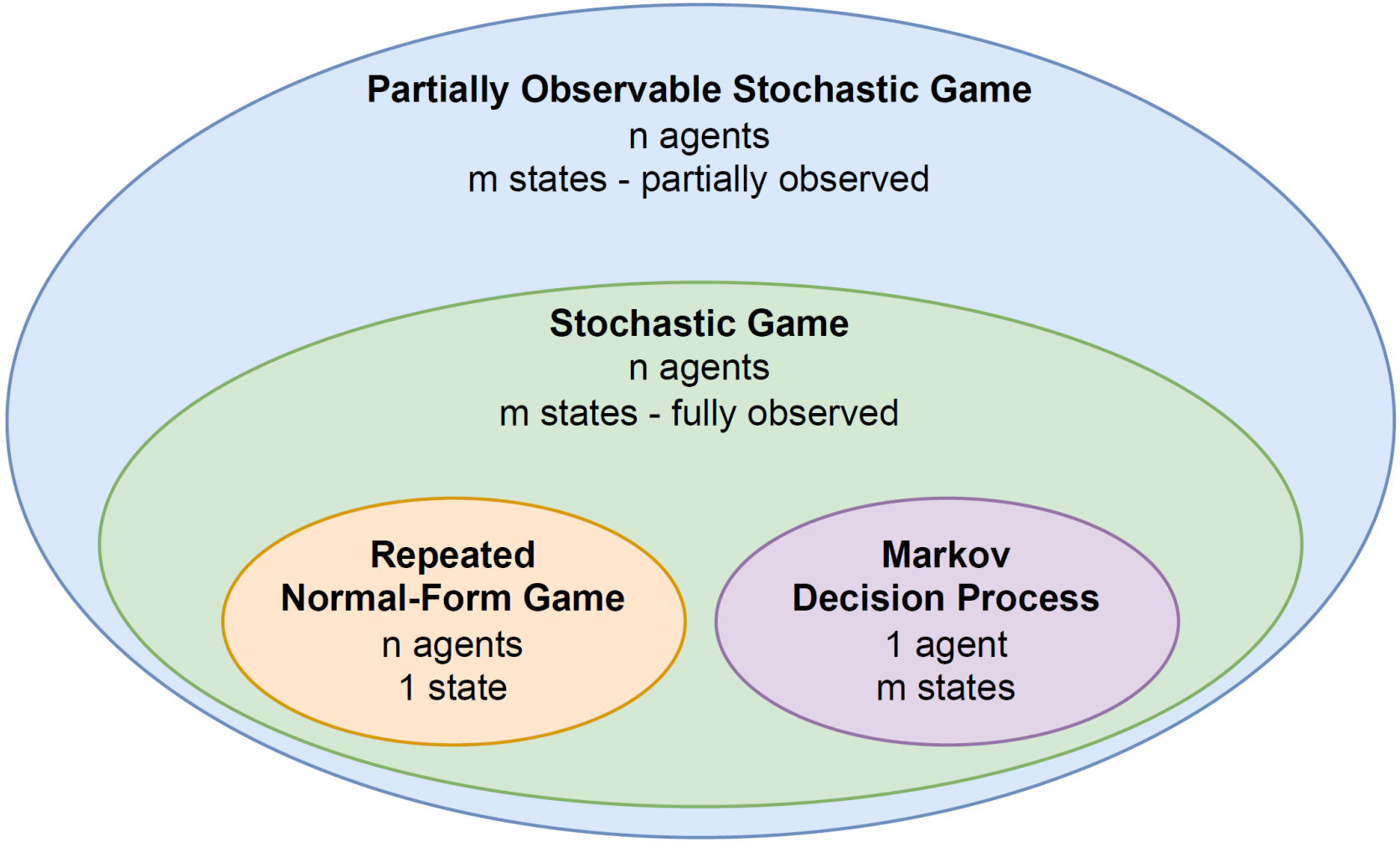
De teoría de juegos a MARL

¿Qué hemos visto hasta ahora?



¿Podemos aprender políticas de manera automática en entornos multiagente?

De teoría de juegos a MARL



Juego en forma normal

- Un **juego en forma normal** tiene forma matricial, relacionando agentes (N) con acciones disponibles (S), además de una función de recompensa sobre las acciones de todos los agentes ($\mathcal{R}$):

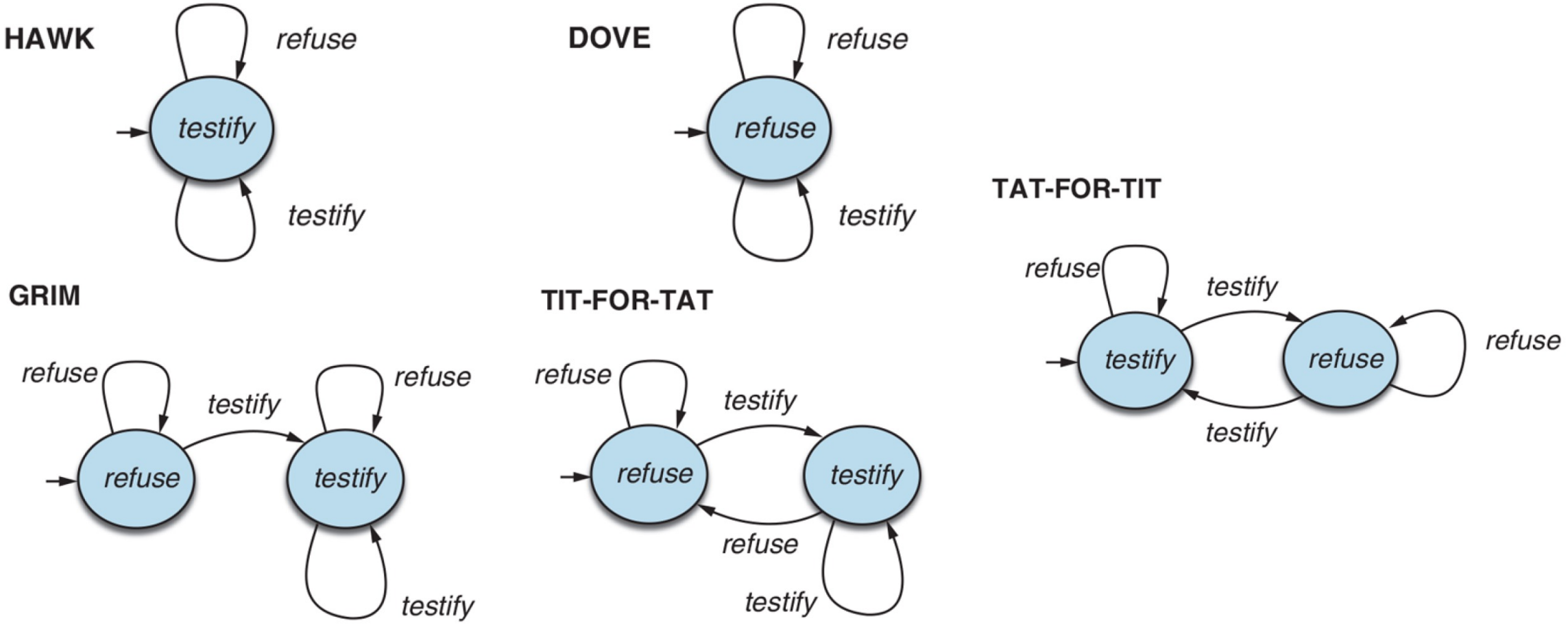
$$G = \langle N, \{A^i\}_{i \in N}, \{\mathcal{R}^i\}_{i \in N} \rangle$$

- Acción conjunta (*joint action*): $a = (a^1, \ldots, a^n)$
- Selección de estrategia mixta: $\pi^i : A^i \to [0, 1]$
- Recompensa para cada agente: $r^i = \mathcal{R}^i(a)$
- Casos especiales (extremos):
  - Juego de suma cero: $\forall a : \sum_{i \in N} r^i(a) = 0$
  - Juego de recompensa común: $\forall a, i, j : r^i(a) = r^j(a)$

Juego en forma normal repetido

- Dado un juego en forma normal G, un **juego en forma normal repetido** es la iteración de G durante T pasos ($t = 0, 1, 2, \dots T - 1$), pudiendo ser T un valor finito o infinito
- Acción conjunta (*joint action*): $a_t = (a_t^1, \dots, a_t^n)$
- Histórico de acciones conjuntas: $\mathrm{h_t} = (a_0, \dots, a_{t-1})$
- Selección de estrategia mixta: $\pi^i(a_t^i | h_t)$
- Recompensa para cada agente: $r_t^i = \mathcal{R}^i(a_t)$

Ejemplo: dilema del prisionero



Juegos estocásticos

Aprendizaje por refuerzo multiagente

Juego estocástico

- n agentes, m estados, observabilidad total
  - Generalizamos juegos en forma normal y MDPs
- Un **juego estocástico** se define como $G = \langle N, S, A, \mathcal{R}, \mathcal{T}, \mu \rangle$:
  - $N = \{1, \dots, n\}$ es el conjunto de agentes
  - S es el conjunto de estados, siendo $\bar{S} \subseteq S$ los estados finales
  - A^i es el conjunto de acciones disponibles para el agente $i \in N$
  - $\mathcal{R}^i: S \times A \times S \to \mathbb{R}$ es la función de recompensa
  - $\mathcal{T}: S \times A \times S \to [0, 1]$ es la función de transición entre estados, siendo $\mathcal{T}(s, a, s')$ equivalente a $p(s'|s, a)$
  - $\mu: S \to [0,1]$ es la distribución de probabilidad para el estado inicial

Juego estocástico

- A medida que el juego avanza, se construye un histórico de acciones conjuntas tal que $h_t = (s_0, a_0, s_1, a_1, \dots, s_t)$, compartido entre todos los agentes (observabilidad total)
- Suponemos que cada agente $i \in N$ tiene una política $\pi_i\left(a_i^t \middle| h^t\right)$
- El juego empieza en un estado inicial $s^0 \in S$, escogido según μ
- Cada tiempo t, hasta alcanzar un estado $s_t \in \bar{S}$ o hasta $t = T$:
  - Cada agente $i \in N$ observa el estado actual $s^t \in S$ y escoge una acción $a_i^t \in A_i$ maximizando la distribución de probabilidad $\pi_i(\cdot\,|h^t)$
  - Dado el estado s_t y la acción conjunta a_t, el juego transiciona al estado $s_{t+1} \in S$ siguiendo la distribución de probabilidad $p(\cdot|s_t, a_t)$
  - Cada agente recibe una recompensa $r_t^i = \mathcal{R}(s_t, a_t, s_{t+1})$

Juego estocástico

- Un juego estocástico es un juego en forma normal repetido si:
  - Sólo hay un estado en S: $|S| = 1$
  - No hay estados finales: $\bar{S} = \emptyset$
  - La función de recompensa no depende de s_{t+1}:

$$\forall s, a, s', s'' : \mathcal{R}(s, a, s') = \mathcal{R}(s, a, s'')$$

- Un juego estocástico es un MDP (en su definición general) si:
  - Sólo hay un agente en N: $|N| = 1$
  - Los conjuntos de estados y acciones pueden ser continuos

Juego estocástico parcialmente observable

- En los juegos estocásticos parcialmente observables (**POSGs**), los agentes reciben observaciones que pueden ser incompletas sobre el estado del entorno y las acciones de los otros agentes:

$$p(s_t, o_t | s_{t-1}, a_{t-1})$$

- $o^t = (o_t^1, \dots, o_t^n)$ es la observación conjunta en tiempo t conteniendo todas las observaciones individuales o_t^i
- Para simplificar, se suele suponer que las observaciones sólo dependen del nuevo estado s_t y la acción conjunta a_{t-1} que condujo al nuevo estado (y no del estado anterior s_{t-1})

Juego estocástico parcialmente observable

- Un **juego estocástico parcialmente observable** (**POSG**) se define como una tupla $G = \langle N, S, O, \mathcal{O}, A, \mathcal{R}, \mathcal{T}, \mu \rangle$, donde:
  - $N, S, A, \mathcal{R}, \mathcal{T}, \mu$ se definen del mismo modo que en un juego estocástico
  - $O = (O^1, \dots, O^n)$ es el conjunto de observaciones accesibles para cada agente i
  - $\mathcal{O} = (\mathcal{O}^1, \dots, \mathcal{O}^n)$ es el conjunto de funciones de observación para cada agente i: $\mathcal{O}^i$: $A \times S \times O^i \rightarrow [0,1]$, tal que

$$\forall a \in A, s \in S: \sum_{o^i \in O^i} \mathcal{O}^i(a, s, o^i) = 1$$

Juego estocástico parcialmente observable

- Un POSG se resuelve de manera similar a un juego estocástico:
  - Los agentes no reciben estados sino observaciones $o_t^i \in O^i$ según la distribución de probabilidad $\mathcal{O}^i(o_t^i|a_{t-1}, s_t)$
  - La política $\pi^i(a_t^i|h_t^i)$ está condicionada por la historia privada de observaciones de i: $h_t^i = (o_0^i, \dots, o_t^i)$
  - Las acciones a_{t-1}^j de los demás agentes no siempre son observables
  - Sin embargo, $\mathcal{T}$ y $\mathcal{R}$ siguen dependiendo de los estados, no de las observaciones: $p(s_t|s_{t-1}, a_{t-1})$ y $\mathcal{R}(s_{t-1}, a_{t-1}, s_t')$
- Un POSG es un juego estocástico cuando $o_t^i = (s_t, a_{t-1})$
- Un POSG es un POMDP cuando $|N| = 1$

Juego estocástico parcialmente observable

- El concepto de función de observación $\mathcal{O}$ es **flexible** ya que permite modelar escenarios de interés en sistemas multiagente:
  - **Falta de observabilidad de acciones de otros agentes**: robótica, mercados con transacciones o negociaciones privadas
  - **Rango de observación limitado**: sensores, juegos con *fog of war*
- Un POMDP (un agente) normalmente se resuelve mediante la creación y actualización de creencias sobre estados
  - Esto hace que el problema sea **difícil de tratar** (ver tema 5, apéndice)
  - El problema se agrava en POSGs ($|N| > 1$)
  - El enfoque será similar a model-free RL: **no depender de $\boldsymbol{S}$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{\mathcal{O}^i}$**

Comunicación

- Los juegos estocásticos y POSGs permiten modelar **comunicación entre agentes**:

$$A^i = X^i \times M^i$$

  donde X^i es el conjunto de acciones disponibles en el entorno para i y M^i es el conjunto de acciones comunicativas (e.g. mensajes, longitud de un vector de tokens, etc) disponibles para i

- El estado del entorno no depende de M^i, pero los otros agentes pueden recibir el producto de estas acciones comunicativas
- La función de transición en este caso tiene la forma:

$$\forall s, s' \in S, \forall a \in A, m \in M: p(s'|s,a) = p(s'|s, \langle (a^1, m^1), \ldots, (a^n, m^n) \rangle)$$

Conocimiento sobre el entorno

- En **Teoría de Juegos**, la suposición estándar es que **todos los agentes tienen conocimiento de todos los componentes que definen el juego**
  - Incluyendo los espacios de acciones y recompensas propios y de todos los agentes
  - Esta suposición permite resolver el juego en base al **análisis de conceptos de solución** (mejores respuestas, equilibrios)
- En **aprendizaje por refuerzo multiagente**, la suposición de partida es que **los agentes no tienen conocimiento de las funciones de recompensa, de transición ni de observación**
- Cada agente i sólo percibe la experiencia inmediata de
  - La recompensa inmediata r_t^i
  - (Juegos estocásticos) Las acciones conjuntas a_t y el siguiente estado s_{t+1}
  - (POSGs) La observación o_{t+1}^i

MARL

Aprendizaje por refuerzo multiagente

Definición

- Un **problema de aprendizaje por refuerzo multiagente** (***MARL problem***) se define por la combinación de **un modelo de juego** y **un concepto de solución**
  - El modelo de juego define las mecánicas del sistema multiagente y las interacciones entre agentes
  - El concepto de solución define las propiedades que debería cumplir una solución válida para el problema
- La solución a un problema MARL es una **política conjunta** (***joint policy***) $\pi = (\pi^1, \dots, \pi^n)$ que satisface los requerimientos definidos por la solución de concepto con respecto de la recompensa esperada $U^i(\pi)$ para cada agente i

Utilidad esperada

- Para generalizar, definimos sobre un POSG
- El **histórico completo hasta tiempo t** se define como:

$$\hat{h}_t = \{(s_\tau, o_\tau, a_\tau)_{\tau<t}, s_t, o_t\}$$

- La **utilidad esperada para el agente i bajo la política conjunta π** es la suma de utilidades de todos los históricos que se pueden alcanzar con la política en el juego, ponderadas por su probabilidad:

$$U^i(\pi) = \lim_{t\to\infty} \mathbb{E}_{\hat{h}_t\sim(\mu,\mathcal{T},\mathcal{O},\pi)}\left[u^i(\hat{h}_t)\right] = \sum_{\hat{h}_t\in\hat{H}} p\left(\hat{h}_t\middle|\pi\right)\cdot u^i\left(\hat{h}_t\right)$$

Utilidad esperada

$$U^i(\pi) = \lim_{t\to\infty} \mathbb{E}_{\hat{h}_t \sim (\mu,\mathcal{T},\mathcal{O},\pi)}\left[\boldsymbol{u^i(\hat{h}_t)}\right] = \sum_{\hat{h}_t \in \hat{H}} \boldsymbol{p(\hat{h}_t|\pi)} \cdot \boldsymbol{u^i(\hat{h}_t)}$$

- La **probabilidad de alcanzar un histórico $\hat{h}_t$ con la política π** es:

$$\boldsymbol{p(\hat{h}_t|\pi)} = \mu(s_0) \cdot \mathcal{O}(o_0|\emptyset, s_0) \cdot \prod_{\tau=0}^{t-1} \pi(a_\tau|h_\tau) \cdot p(s_{\tau+1}|s_\tau, a_\tau) \cdot \mathcal{O}(o_{\tau+1}|a_\tau, s_{\tau+1})$$

- La **utilidad de un histórico $\hat{h}_t$ para un agente i** es:

$$\boldsymbol{u^i(\hat{h}_t)} = \sum_{\tau=0}^{t-1} \gamma_\tau \cdot \mathcal{R}^i(s_\tau, a_\tau, s_{\tau+1})$$

Política conjunta y política individual

- Si suponemos que **los agentes actúan de manera independiente**, siguiendo las reglas del cálculo de la probabilidad conjunta se puede **descomponer la política conjunta en el producto de las políticas individuales**:

$$\pi(a_t|h_t) = \prod_{i \in N} \pi^i(a_t^i|h_t^i)$$

Ecuaciones de Bellman

- Definiciones alternativas y equivalentes, basadas en las **ecuaciones de Bellman**, como funciones sobre históricos en lugar de sobre estados:

$$V_i^{\pi}(\hat{h}) = \sum_{a \in A} \pi(a|(o_0, \dots, o_t)) \cdot Q_i^{\pi}(\hat{h}, a)$$

$$Q_i^{\pi}(\hat{h}, a) = \sum_{s' \in S} p(s'|s(\hat{h}), a) \left[\mathcal{R}^i(s(\hat{h}), a, s') + \gamma \sum_{o' \in O} \mathcal{O}(o'|a, s') V_i^{\pi}(\langle \hat{h}, a, s', o' \rangle) \right]$$

donde $s(\hat{h})$ es el último estado de $\hat{h}$, y $\langle\rangle$ actúa como operador de concatenación

Utilidad esperada (ecuaciones de Bellman)

- La **utilidad esperada con respecto de V_i^π** es:

$$U^i(\pi) = \mathbb{E}_{s_0 \sim \mu, o_0 \sim \mathcal{O}(\cdot|\emptyset, s_0)}[V_i^\pi(\langle s_0, o_0 \rangle)]$$

Conceptos de solución más habituales

- **Mejores respuestas**
- **Minimax**
- **Equilibrio de Nash**
- **Equilibrio de ϵ-Nash**: *si los demás siguen haciendo lo que están haciendo, yo no puedo mejorar, por más que ϵ, lo que estoy haciendo*

  $\forall i, \pi'^i : U^i\left(\pi'^i, \pi^{-i}\right) - \epsilon \leq U^i(\pi)$

- **Pareto-eficiencia**
- Bienestar social (***welfare***):

$$W(\pi) = \sum_{i \in N} U^i(\pi)$$

- Justicia social (***fairness***):

$$F(\pi) = \prod_{i \in N} U^i(\pi)$$

- **Mínimo arrepentimiento**

Mínimo arrepentimiento

- Todos los conceptos de solución vistos se basan en un análisis de las políticas de todos los agentes, excepto la del mínimo arrepentimiento (*no-regret*)
- Arrepentimiento: diferencia entre la recompensa recibida por un agente y la recompensa que podría haber recibido si hubiese escogido otra acción
- En un juego en forma normal, dada una secuencia de episodios $e = 1, \dots, z$:

$$\text{Regret}_z^i = \max_{a^i \in A^i} \sum_{e=1}^{z} \left[\mathcal{R}^i\left(\langle a^i, a_e^{-i} \rangle\right) - \mathcal{R}^i(a_e) \right]$$

Mínimo arrepentimiento

- Se dice que hay *no-regret* cuando el arrepentimiento medio en el infinito es cero:

$$\forall i: \lim_{z \to \infty} \left[\frac{1}{z} \cdot \mathrm{Regret}_z^i \leq 0 \right]$$

- La forma general para juegos estocásticos y POSGs es:

$$\mathrm{Regret}_z^i = \max_{\pi^i \in \Pi^i} \sum_{e=1}^{z} \left[U^i\left(\langle \pi^i, \pi_e^{-i} \rangle \right) - U^i(\pi_e) \right]$$

Entrenamiento

Aprendizaje por refuerzo multiagente

Proceso general de aprendizaje

- Un proceso de aprendizaje para MARL parte de estos elementos:
  - Modelo de juego
  - Conjunto de entrenamiento $\mathcal{D}_z$: un conjunto de z históricos h_{t_e} producidos por políticas conjuntas π_e para cada episodio e

$$\mathcal{D}_z = \{h_{t_e} | \, e = 1, \dots, z\}, z \geq 0$$

  - Algoritmo de aprendizaje $\mathbb{L}$ que usa los datos y una política para generar una nueva política:

$$\pi_{z+1} = \mathbb{L}(\mathcal{D}_z, \pi_z)$$

  - Objetivo de aprendizaje (que representa el concepto de solución)
- $\mathbb{L}$ puede consistir en una combinación de algoritmos, por ejemplo: un algoritmo diferente para cada agente

Tipos de convergencia

- La forma más básica de convergencia es la convergencia teórica a una política óptima con datos infinitos:

$$\lim_{z \to \infty} \pi_z = \pi^*$$

- Un ejemplo de forma alternativa es la convergencia de la utilidad esperada:

$$\lim_{z \to \infty} U^i(\pi_z) = U^i(\pi^*), \forall i \in N$$

Reducción a un agente

- La forma más sencilla de entrenar agentes para resolver un problema MARL es reducir el problema colectivo a un problema de un solo agente: **aprendizaje centralizado**
  - Un único agente entrena una política central π^c a partir de las observaciones de todos los agentes
  - El algoritmo **CQL (Central Q-Learning)** es un ejemplo de esta metodología
- Cada agente puede aprender su propia política π^i usando únicamente su propio histórico de observaciones, acciones y recompensas, ignorando la existencia de otros agentes: **aprendizaje independiente**
  - Un ejemplo de esta metodología es **IQL (Independent Q-Learning)**

Reducción a un agente

- Estos enfoques *single-agent* son vulnerables a ciertos problemas:
  - **Falta de estacionariedad**: la acción óptima no depende sólo de la observación sino también de las demás acciones individuales simultáneas
  - **Múltiples equilibrios**: si puede haber múltiples equilibrios según el concepto de solución, puede haber inestabilidad en el entrenamiento o en la política
  - **Reparto de recompensas**: decidir cómo repartir una recompensa colectiva entre los agentes puede complicar el entrenamiento
  - **Falta de escalabilidad**: el espacio de acciones conjuntas depende del espacio de acciones de cada agente, causando un crecimiento exponencial del espacio de búsqueda y dificultando el equilibrio exploración – explotación
- Técnicas como *self-play* o *mixed-play* pueden ser una solución

Algoritmos multiagente

- Basados en modelo
  - **Iteración de valor**
  - Minimax
  - Expectiminimax
- Aprendizaje por diferencias temporales (**aprendizaje de acciones conjuntas – JAL-GT**): estimación del valor (según el concepto de solución) de las posibles acciones conjuntas
  - Minimax Q-Learning
  - Nash Q-Learning
  - Correlated Q-Learning

Algoritmos multiagente

- **Modelado de agentes**: construcción de modelos de los otros agentes para predecir su comportamiento, modelar el juego y aplicar conceptos de solución (e.g. mejores respuestas)
  - Reconstrucción de política: aprendizaje supervisado, utilizando los pares estado/observación-acción capturados
  - **Juego ficticio** (juegos en forma normal)
  - **Aprendizaje de acciones conjuntas – JAL-AM** (juegos estocásticos)
  - **Aprendizaje bayesiano**: inferencia probabilística de creencias, calculando el **valor de la información** obtenida de los otros agentes para guiar el equilibrio explotación-exploración

Multi-agent Reinforcement Learning in Sequential Social Dilemmas

Joel Z. Leibo[1]
DeepMind, London, UK
jzl@google.com

Vinicius Zambaldi[1]
DeepMind, London, UK
vzambaldi@google.com

Marc Lanctot
DeepMind, London, UK
lanctot@google.com

Janusz Marecki
DeepMind, London, UK
tartel@google.com

Thore Graepel
DeepMind, London, UK
thore@google.com

ABSTRACT

Matrix games like Prisoner's Dilemma have guided research on social dilemmas for decades. However, they necessarily treat the choice to cooperate or defect as an atomic action. In real-world social dilemmas these choices are temporally extended. Cooperativeness is a property that applies to policies, not elementary actions. We introduce sequential social dilemmas that share the mixed incentive structure of matrix game social dilemmas but also require agents to learn policies that implement their strategic intentions. We analyze the dynamics of policies learned by multiple self-interested independent learning agents, each using its own deep Q-network, on two Markov games we introduce here: 1. a fruit

The theory of repeated general-sum matrix games provides a framework for understanding social dilemmas. Fig. 1 shows payoff matrices for three canonical examples: Prisoner's Dilemma, Chicken, and Stag Hunt. The two actions are interpreted as cooperate and defect respectively. The four possible outcomes of each stage game are R (reward of mutual cooperation), P (punishment arising from mutual defection), S (sucker outcome obtained by the player who cooperates with a defecting partner), and T (temptation outcome achieved by defecting against a cooperator). A matrix game is a social dilemma when its four payoffs satisfy the following *social dilemma inequalities* (this formulation from [3]):

https://arxiv.org/abs/1702.03037