Reinforcement Learning

Multi-Agent Learning I

Stefano V. Albrecht

THE UNIVERSITY *of* EDINBURGH
informatics

Lecture Outline

Today:

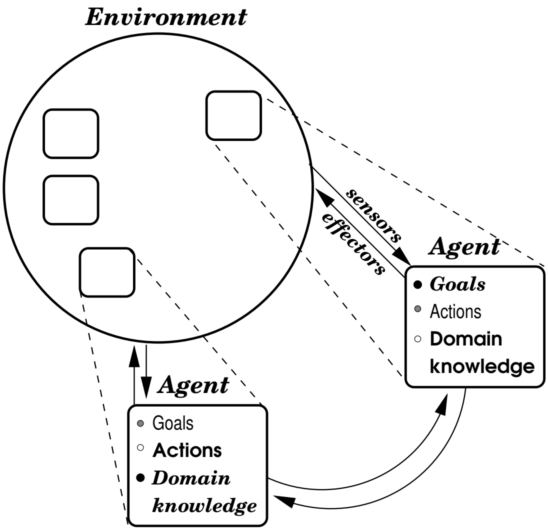
- Multi-agent systems
- Multi-agent learning and challenges
- Models of interaction
- Learning goals

Next time:

- Learning algorithms

Multi-Agent Systems

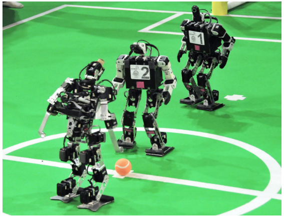
- Multiple agents interact in shared environment
- Each agent with own observations, actions, goals, ...
- Agents must coordinate actions to achieve their goals



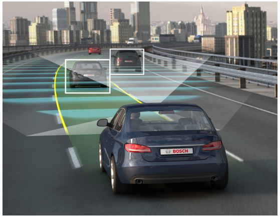
Multi-Agent Systems – Applications

Games

Robot soccer



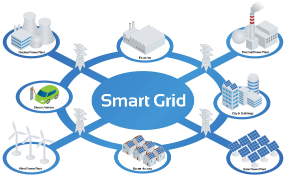
Autonomous cars



Negotiation/markets

Wireless networks

Smart grid

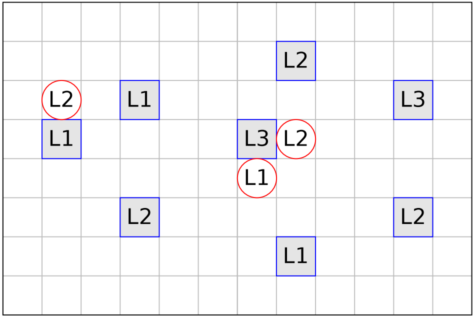


Why Multi-Agent Systems?

Example: Level-based foraging

- 3 robots (circles) must collect all items in minimal time
- Robots can collect item if sum of their levels $\geq$ item level
- Action is tuple (rob_1, rob_2, rob_3) with $rob_i \in \{\text{up, down, left, right, collect}\}$

  $\Rightarrow$ 125 possible actions!



Why Multi-Agent Systems?

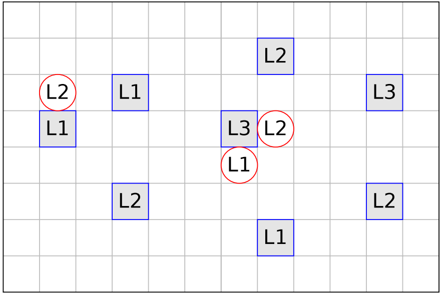
Idea of multi-agent systems:

Decompose intractable decision problem into smaller decision problems

- Use 3 agents, one for each robot

  Each agent has only 5 possible actions!

  $\Rightarrow$ *Factored action space*



New challenge:

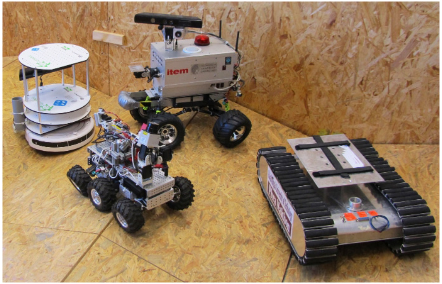
- Agents must *coordinate* actions with each other to accomplish goals

Why Multi-Agent Systems?

More reasons for multi-agent systems:

Decentralised control: may not be able to control system in one central place (e.g. multiple robots working together, without communication)

State-space reduction: multi-agent decomposition may also reduce size of state space for individual agents (e.g. if only a subset of state features are relevant for an agent)



Multi-agent learning:

- Learning is process of improving performance via experience
- Can agents *learn* to coordinate actions with other agents?
- What to learn?

  $\Rightarrow$ How to select own actions

  $\Rightarrow$ How other agents select actions

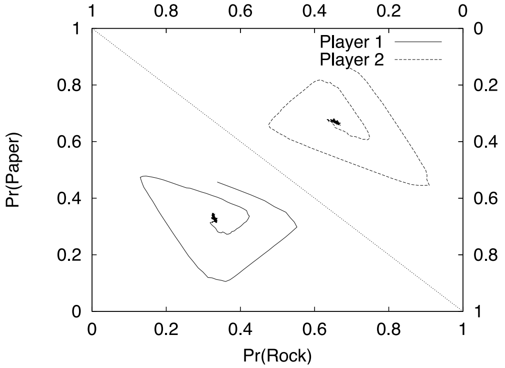
  $\Rightarrow$ Other agents' goals, plans, beliefs, ...

Challenges of Multi-Agent Learning

Non-stationary environment:

- MDP assumes stationary environment: environment dynamics do not change over time
- If environment includes learning agents, environment becomes non-stationary from the perspective of individual agents

  ⇒ Markov assumption broken



Moving target problem

Challenges of Multi-Agent Learning

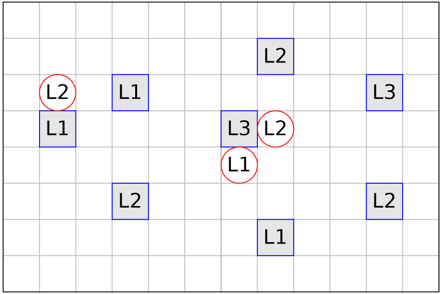
Multi-agent credit assignment:

- We know (temporal) credit-assignment problem from standard RL

  $\Rightarrow$ What past actions led to current reward?

- Now we must also ask: *whose* actions led to current reward?

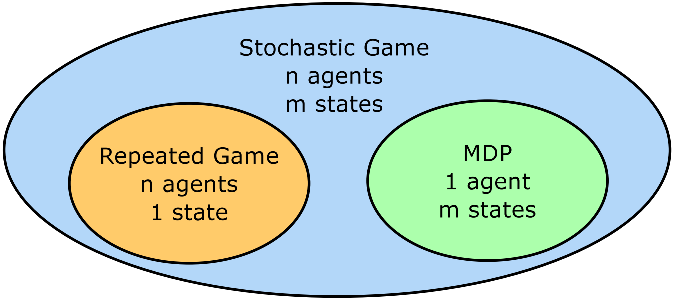
Example: If the two agents in centre collect L3 item, everyone gets $+1$ reward. How do agents know that the agent on the left did not contribute to the reward?



Multi-Agent Models

Standard models of multi-agent interaction:

- Normal-form game
- Repeated game
- Stochastic game



Normal-Form Game

Normal-form game consists of:

- Finite set of agents $N = \{1, ..., n\}$
- For each agent $i \in N$:
  - Finite set of actions A_i
  - Reward function $u_i : A \to \mathbb{R}$, where $A = A_1 \times ... \times A_n$ (joint action space)

Each agent i selects policy $\pi_i : A_i \to [0, 1]$, takes action $a_i \in A_i$ with probability $\pi_i(a_i)$, and receives reward $u_i(a_1, ..., a_n)$

Given policy profile $(\pi_1, ..., \pi_n)$, expected reward to i is

$$U_i(\pi_1, ..., \pi_n) = \sum_{a \in A} u_i(a) \prod_{i \in N} \pi_i(a_i)$$

Normal-Form Game: Prisoner's Dilemma

Example: Prisoner's Dilemma

- Two prisoners are interrogated in separate rooms
- Each prisoner can **Cooperate** (C) or **Defect** (D)
- Reward matrix:

Agent 2

Agent 1		C	D
	C	-1,-1	-5,0
	D	0,-5	-3,-3

Normal-Form Game: Rock-Paper-Scissors

Example: Rock-Paper-Scissors

- Two players, three actions
- Rock beats Scissors beats Paper beats Rock
- Reward matrix:

Agent 2

Agent 1		R	P	S
	R	0,0	-1,1	1,-1
	P	1,-1	0,0	-1,1
	S	-1,1	1,-1	0,0

Repeated Game

Learning is to improve performance via experience

- Normal-form game is single interaction $\Rightarrow$ *no experience!*
- Experience comes from repeated interactions

Repeated Game

Learning is to improve performance via experience

- Normal-form game is single interaction $\Rightarrow$ *no experience!*
- Experience comes from repeated interactions

Repeated game:

- Repeat the same normal-form game for time steps $t = 0, 1, 2, 3, \ldots$
- At time t, each agent i...
  - selects policy π_i^t
  - samples action a_i^t with probability $\pi_i^t(a_i^t)$
  - receives reward $u_i(a^t)$ where $a^t = (a_1^t, \ldots, a_n^t)$
- Learning: modify policy π_i^t based on history $H^t = (a^0, a^1, \ldots, a^{t-1})$

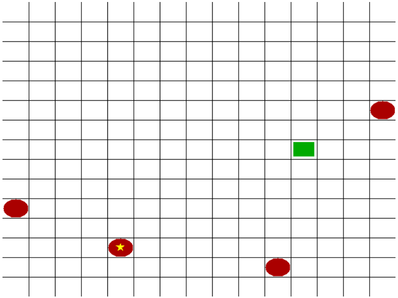
Stochastic Game

Agents interact in shared environment

- Environment has states, and actions have effect on state
- Agents choose actions based on observed state

Example: Predator-prey

- Predator agents (red) must capture prey
- State: agent positions
- Actions: up, down, left, right



Stochastic Game

Stochastic game (or Markov game) consists of:

- Finite set of agents $N = \{1, ..., n\}$
- Finite set of states S
- For each agent $i \in N$:
  - Finite set of actions A_i
  - Reward function $u_i : S \times A \to \mathbb{R}$, where $A = A_1 \times ... \times A_n$
- State transition probabilities $T : S \times A \times S \to [0, 1]$

Generalises MDP to multiple agents

Stochastic Game

Game starts in initial state $s^0 \in S$

At time t, each agent i...

- Observes current state s^t
- Chooses action a_i^t with probability $\pi_i(s^t, a_i^t)$
- Receives reward $u_i(s^t, a_1^t, ..., a_n^t)$

Then game transitions into next state s^{t+1} with probability $T(s^t, a^t, s^{t+1})$

Repeat T times or until terminal state is reached

$\Rightarrow$ Learning is now based on *state-action history* $H^t = (s^0, a^0, s^1, a^1, ..., s^t)$

Stochastic Game — Expected Return

Given policy profile $\pi = (\pi_1, ..., \pi_n)$, what is expected return to agent i in state s?

$$U_i(s, \pi) = \sum_{a \in A} \left(\prod_{j \in N} \pi_j(s, a_j) \right) \left[u_i(s, a) + \gamma \sum_{s' \in S} T(s, a, s')\, U_i(s', \pi) \right]$$

- Analogous to Bellman equation
- Discount rate $0 \le \gamma < 1$ makes return finite

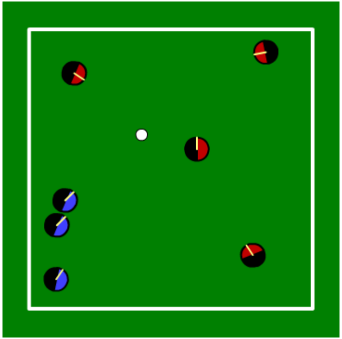
Stochastic Game: Soccer Keepaway

Example: Soccer Keepaway

- "Keeper" agents must keep ball away from "Taker" agents
- State: player positions & orientations, ball position, ...
- Actions: go to ball, pass ball to player, ...

Video: Keepaway

Source: `http://www.cs.utexas.edu/~AustinVilla/sim/keepaway`



What does it mean to solve a game?

- If game has *common rewards*, $\forall i : u_i = u$, then solving game is like solving MDP

  $\Rightarrow$ Find policy profile $\pi = (\pi_1, ..., \pi_n)$ that maximises $U_i(s, \pi)$ for all s

Solving Games

What does it mean to solve a game?

- If game has *common rewards*, $\forall i : u_i = u$, then solving game is like solving MDP

  $\Rightarrow$ Find policy profile $\pi = (\pi_1, ..., \pi_n)$ that maximises $U_i(s, \pi)$ for all s

- But if agent rewards differ, $u_i \neq u_j$, what should π optimise?

  Many solution concepts exist:

  - Minimax solution
  - Nash/correlated equilibrium
  - Pareto-optimality
  - Social welfare & fairness
  - No-regret
  - Targeted optimality & safety

Minimax

Two-player zero-sum game: $u_i = -u_j$

- e.g. Rock-Paper-Scissors, Chess

Minimax

Two-player zero-sum game: $u_i = -u_j$

- e.g. Rock-Paper-Scissors, Chess

Policy profile (π_i, π_j) is minimax profile if

$$U_i(\pi_i, \pi_j) = \max_{\pi_i'} \min_{\pi_j'} U_i(\pi_i', \pi_j') = \min_{\pi_j'} \max_{\pi_i'} U_i(\pi_i', \pi_j') = -U_j(\pi_i, \pi_j)$$

Reward that can be guaranteed against *worst-case* opponent

Minimax

Two-player zero-sum game: $u_i = -u_j$

- e.g. Rock-Paper-Scissors, Chess

Policy profile (π_i, π_j) is minimax profile if

$$U_i(\pi_i, \pi_j) = \max_{\pi_i'} \min_{\pi_j'} U_i(\pi_i', \pi_j') = \min_{\pi_j'} \max_{\pi_i'} U_i(\pi_i', \pi_j') = -U_j(\pi_i, \pi_j)$$

Reward that can be guaranteed against *worst-case* opponent

- Every two-player zero-sum normal-form game has minimax profile (von Neumann and Morgenstern, 1944)
- Every finite or infinite+discounted zero-sum stochastic game has minimax profile (Shapley, 1953)

Nash Equilibrium

Policy profile $\pi = (\pi_1, ..., \pi_n)$ is Nash equilibrium (NE) if

$$\forall i \; \forall \pi'_i : U_i(\pi'_i, \pi_{-i}) \leq U_i(\pi)$$

No agent can improve reward by unilaterally deviating from profile (every agent plays best-response to other agents)

Nash Equilibrium

Policy profile $\pi = (\pi_1, ..., \pi_n)$ is Nash equilibrium (NE) if

$$\forall i \; \forall \pi'_i : U_i(\pi'_i, \pi_{-i}) \leq U_i(\pi)$$

No agent can improve reward by unilaterally deviating from profile (every agent plays best-response to other agents)

Every finite normal-form game has at least one NE (Nash, 1950) (also stochastic games, e.g. Fink (1964))

- Standard solution in game theory
- In two-player zero-sum game, minimax is same as NE

Nash Equilibrium – Example

Example: Prisoner's Dilemma

- Only NE in normal-form game is (D,D)
- Normal-form NE are also NE in infinite repeated game
- Infinite repeated game has many more NE $\rightarrow$ "Folk theorem"

	C	D
C	-1,-1	-5,0
D	0,-5	-3,-3

Nash Equilibrium – Example

Example: Prisoner's Dilemma

- Only NE in normal-form game is (D,D)
- Normal-form NE are also NE in infinite repeated game
- Infinite repeated game has many more NE $\rightarrow$ "Folk theorem"

	C	D
C	-1,-1	-5,0
D	0,-5	-3,-3

Example: Rock-Paper-Scissors

- Only NE in normal-form game is $\pi_i = \pi_j = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

	R	P	S
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

The Equilibrium Legacy

The “Equilibrium Legacy” in multi-agent learning:

- Quickly adopted equilibrium as standard goal of learning
- But equilibrium (e.g. NE) has many limitations...

The Equilibrium Legacy

The "Equilibrium Legacy" in multi-agent learning:

- Quickly adopted equilibrium as standard goal of learning
- But equilibrium (e.g. NE) has many limitations...

1. Non-uniqueness

   Often multiple NE exist; how should agents choose same one?

The Equilibrium Legacy

The “Equilibrium Legacy” in multi-agent learning:

- Quickly adopted equilibrium as standard goal of learning
- But equilibrium (e.g. NE) has many limitations...

1. Non-uniqueness

   Often multiple NE exist; how should agents choose same one?

2. Sup-optimality

   NE may not give highest rewards to agents

The Equilibrium Legacy

The "Equilibrium Legacy" in multi-agent learning:

- Quickly adopted equilibrium as standard goal of learning
- But equilibrium (e.g. NE) has many limitations...

1. Non-uniqueness

   Often multiple NE exist; how should agents choose same one?

2. Sup-optimality

   NE may not give highest rewards to agents

3. Incompleteness

   NE does not specify behaviours for off-equilibrium paths

The Equilibrium Legacy

The "Equilibrium Legacy" in multi-agent learning:

- Quickly adopted equilibrium as standard goal of learning
- But equilibrium (e.g. NE) has many limitations...

1. Non-uniqueness

   Often multiple NE exist; how should agents choose same one?

2. Sup-optimality

   NE may not give highest rewards to agents

3. Incompleteness

   NE does not specify behaviours for off-equilibrium paths

4. Rationality

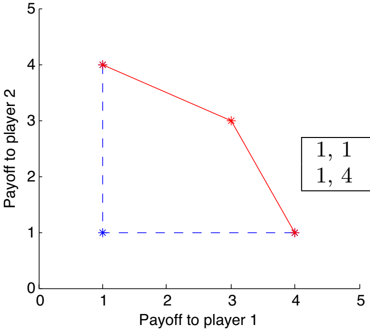
   NE assumes all agents are rational (= perfect reward maximisers)

Pareto Optimum

Policy profile $\pi = (\pi_1, ..., \pi_n)$ is Pareto-optimal if there is no other profile π' such that

$$\forall i : U_i(\pi') \geq U_i(\pi) \quad \textbf{and} \quad \exists_i : U_i(\pi') > U_i(\pi)$$

Can't improve one agent without making other agent worse off



1, 1	4, 1
1, 4	3, 3

Pareto-front is set of all Pareto-optimal rewards (red line)

Social Welfare & Fairness

Pareto-optimality says nothing about social welfare and fairness

Welfare and fairness of profile $\pi = (\pi_1, ..., \pi_n)$ often defined as

$$Welfare(\pi) = \sum_i U_i(\pi) \qquad Fairness(\pi) = \prod_i U_i(\pi)$$

π is welfare/fairness-optimal if it maximises $Welfare(\pi)/Fairness(\pi)$

$\Rightarrow$ Any welfare/fairness-optimal π is also Pareto-optimal (Why?)

No-Regret

Given history $H^t = (a^0, a^1, ..., a^{t-1})$, agent i's regret for not having taken action a_i is

$$R_i(a_i|H^t) = \sum_{\tau=0}^{t-1} u_i(a_i, a_{-i}^{\tau}) - u_i(a_i^{\tau}, a_{-i}^{\tau})$$

Policy π_i achieves no-regret if

$$\forall a_i : \lim_{t \to \infty} \frac{1}{t} R_i(a_i|H^t) \leq 0$$

(Other variants exist)

No-Regret

Like Nash equilibrium, no-regret widely used in multi-agent learning

But, like NE, definition of regret has conceptual issues

No-Regret

Like Nash equilibrium, no-regret widely used in multi-agent learning

But, like NE, definition of regret has conceptual issues

- Regret definition assumes other agents don't change actions

$$R_i(a_i|H^t) = \sum_{\tau=0}^{t-1} u_i(a_i, a_{-i}^{\tau}) - u_i(a_i^{\tau}, a_{-i}^{\tau})$$

$\Rightarrow$ But: entire history may change if different actions taken!

No-Regret

Like Nash equilibrium, no-regret widely used in multi-agent learning

But, like NE, definition of regret has conceptual issues

- Regret definition assumes other agents don't change actions

$$R_i(a_i|H^t) = \sum_{\tau=0}^{t-1} u_i(a_i, a_{-i}^{\tau}) - u_i(a_i^{\tau}, a_{-i}^{\tau})$$

$\Rightarrow$ But: entire history may change if different actions taken!

- Minimising regret not generally same as maximising reward e.g. (Crandall, 2014)

Targeted Optimality & Safety

Many algorithms designed to achieve some version of targeted optimality and safety:

- If other agent's policy π_j is in a defined class, agent i's learning should converge to best-response

$$U_i(\pi_i, \pi_j) \approx \max_{\pi_i'} U_i(\pi_i', \pi_j)$$

Targeted Optimality & Safety

Many algorithms designed to achieve some version of targeted optimality and safety:

- If other agent's policy π_j is in a defined class, agent i's learning should converge to best-response

$$U_i(\pi_i, \pi_j) \approx \max_{\pi_i'} U_i(\pi_i', \pi_j)$$

- If π_j not in class, π_i should at least achieve safety (maximin) reward

$$U_i(\pi_i, \pi_j) \approx \max_{\pi_i'} \min_{\pi_j'} U_i(\pi_i', \pi_j')$$

Policy classes: non-learning, memory-bounded, finite automata, ...

Reading (Optional)

- G. Laurent, L. Matignon, N. Le Fort-Piat. The World of Independent Learners is not Markovian. International Journal of Knowledge-Based and Intelligent Engineering Systems, 15(1):55–64, 2011
- Our RL reading list contains many survey articles on multi-agent learning: `https://eu01.alma.exlibrisgroup.com/leganto/public/44UOE_INST/lists/22066371180002466?auth=SAML&section=22066371280002466`
- AIJ Special Issue *"Foundations of Multi-Agent Learning"* (2007) `https://www.sciencedirect.com/journal/artificial-intelligence/vol/171/issue/7`

References

J. Crandall. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research*, 49:111–142, 2014.

A. Fink. Equilibrium in a stochastic n-person game. *Journal of Science of the Hiroshima University*, 28(1):89–93, 1964.

J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

L. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100, 1953.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.