Tutorial:
Multi-Agent Learning
D Balduzzi, T Graepel, E Hughes, M Jaderberg, S Omidshafiei, J Perolat, K Tuyls
Joint work with many great collaborators, including:
We won't cover ...

- Single Agent Reinforcement Learning
  - Markov Decision Processes
  - Algorithms
- A good resource though
Part I. Background & Theory

1. Introduction
2. NFGs and Markov Games
3. Social Learning
Part I: Background & Theory

- Motivation
- What is Multi-Agent Learning?
  - General Setup
  - Different Realizations: RL-based, Swarms, Evo-based
  - Role of (Evolutionary) Game Theory
- Game Theoretic Intuitions: NFG and Replicator Dynamics
- Opportunities & Challenges
Motivation

- Re-thinking fundamentals of whole area
  - Special issue Shoham 2007
  - AI Magazine article (Weiss & Tuyls)
  - The rise of Deep Learning and building AGI

- A unified formal framework
- Better understanding/theoretical underpinnings
- Application to complex systems

Based on a recent paper:
K. Tuyls and P. Stone: *Multiagent Learning Paradigms*. To Appear
Motivation

**Surge in Autonomous Systems and Artificial Intelligence Research**

Deep reinforcement learning

RoboCup@work (smARTLab)
Motivation

Surge in Autonomous Systems and Artificial Intelligence Research

Deep reinforcement learning

RoboCup@work (smARTLab)

On the verge of huge changes in AUTOMATION: Industry 4.0

O. Scalabre: “the next manufacturing revolution is here”

Report of the 100 Year Study of AI (released Sept 1st ’16, AAAI)
Motivation

**Surge in Autonomous Systems and Artificial Intelligence Research**

Deep reinforcement learning

RoboCup@work (smARTLab)

**On the verge of huge changes in AUTOMATION: Industry 4.0**

O. Scalabre: “the next manufacturing revolution is here”

Report of the 100 Year Study of AI (released Sept 1st ’16, AAAI)

BUT
Motivation

**Surge in Autonomous Systems and Artificial Intelligence Research**

Deep reinforcement learning

RoboCup@work (smARTLab)
Motivation

**Surge in Autonomous Systems and Artificial Intelligence Research**

Deep reinforcement learning

RoboCup@work (smARTLab)
Motivation

Actions

Control

Environment

Perception

Sensory Input
Motivation

Control

Perception

Actions

Environment

Sensory Input
We live in a multi-agent world and to be successful in that world, agents will need to learn to take into account the agency of others.
Example (RoboCup)

The remaining items needed are picked
The black aluminium profile and the M20 nut are picked up
Example warehouse commissioning

The robots decide autonomously which actions to take. They receive the global state from the warehouse management software. The global state consists of the currently active orders and approximate positions of the other robots.
What is Multi-Agent Learning?
Multi-Agent Learning lacks a Foundation, or Theory, of its own
What is Multi-Agent Learning?
The study of multi-agent systems in which one or more of the autonomous entities improves automatically through experience.

What is Multi-Agent Learning?

- RL towards individual utility
- RL towards social welfare
- Co-evolutionary learning
- Swarm Intelligence
- Adaptive mechanism design

Tools
- EGT
- (Opponent Modelling)
What is Multi-Agent Learning?

“Perhaps a thing is simple if you can describe it fully in several different ways, without immediately knowing that you are describing the same thing” R. Feynman
Several Realizations

1. Online RL towards individual utility
2. Online RL towards social welfare
3. Co-Evolutionary approaches
4. Swarm Intelligence
5. Adaptive Mechanism Design
“Perhaps a thing is simple if you can describe it fully in several different ways, without immediately knowing that you are describing the same thing” R. Feynman
EGT: unified theory (Role of EGT)

- Individual Learners
  - Reinforcement Learning

- Mechanism Design
- Protocol Learning

- Multi-Agent Learning

- Evolutionary Computation
- Swarm Intelligence
- Population Learners
EGT: unified theory (Role of EGT)

- Individual Learners
  - Reinforcement Learning

- Mechanism Design
- Protocol Learning

- Multi-Agent Learning

- Evolutionary Computation

- Swarm Intelligence
  - Population Learners
EGT: unified theory (Role of EGT)

Unified

Individual Learners
Reinforcement Learning

Mechanism Design
Protocol Learning

Multi-Agent Learning

Evolutionary Game Theory

Swarm Intelligence
Neural Computation

DeepMind
EGT: Towards a Unified Theory (Role of EGT)
EGT: Towards a Unified Theory (Role of EGT)

- CENTRAL CONCEPT:
  - Nash equilibrium
- Normative theory
- Rational players
EGT: Towards a Unified Theory (Role of EGT)

Strategic decision making

Classical Game Theory
- CENTRAL CONCEPT: Nash equilibrium
- Normative theory
- Rational players

Key technique for adaptation
Reinforcement Learning
EGT: Towards a Unified Theory (Role of EGT)

Behavioral assumptions

Evolutionary Game Theory

Key technique for adaptation

Reinforcement Learning
EGT: Towards a Unified Theory (Role of EGT)
EGT: Towards a Unified Theory (Role of EGT)

\[ \frac{dp_i}{dt} = p_i [e_i A_q - p A_q] \]
\[ \frac{dq_i}{dt} = q_i [p B e_i - q B p] \]
Game Theoretic Intuitions

- Evolutionary Game Theory (EGT), 1
  - Application of game theory to evolving populations of lifeforms in biology (1973, Smith & Price)
  - EGT differs from classical GT by focusing more on the dynamics of strategy change (quality, frequency)
  - Common approach: **replicator equations**, describing growth rate of the proportion of organisms using a certain strategy
Evolutionary Game Theory (EGT), 2

- **Extension** to two-player game situations, coupled replicator equations:

\[
\frac{dx_i}{dt} = x_i[(Ay)_i - x^T Ay] \\
\frac{dy_i}{dt} = y_i[(Bx)_i - y^T Bx]
\]

- **Example**: Prisoner’s dilemma

\[
A = \begin{pmatrix} -2 & -10 \\ -1 & -5 \end{pmatrix} \quad B = \begin{pmatrix} -2 & -1 \\ -10 & -5 \end{pmatrix}
\]

**Cooperate (deny)**

**Defect (confess)**

A is row player
B is column player
Game Theoretic Intuitions

- There are strong formal links between EGT and multi-agent RL [e.g., AAMAS09/10/12/14, IAT08, ECML, AAAI’14, JAIR’15 etc.]
  - Learning dynamics corresponds to replicator dynamics
  - The concept of evolutionary stable strategies (ESS) can be transferred to multi-agent RL (Nash equilibria)
- Multi-agent RL methods and evolutionary models
- Recently connection between PG and RD (Neural Replicator Dynamics)
Game Theoretic Intuitions

- We showed that there are strong formal links between EGT and multi-agent RL [e.g., AAMAS09/10/12/14, IAT08, etc.]
- Learning dynamics corresponds to replicator dynamics
- The concept of evolutionary stable strategies (ESS) can be transferred to multi-agent RL (≡ Nash equilibria)
- Multi-agent RL methods and evolutionary models: Game Theoretic Intuitions

FAQ
\[ \frac{dx_i}{dt} = \frac{\alpha x_i}{\tau} [ (Ay)_i - x^T Ay ] + x_i \alpha \sum_j x_j \ln \left( \frac{x_j}{x_i} \right) \]

LFAQ
\[ u_i = \sum_j \frac{A_{ij} y_j \left[ \left( \sum_{k: A_{ik} \leq A_{ij}} y_k \right)^{\kappa} - \left( \sum_{k: A_{ik} < A_{ij}} y_k \right)^{\kappa} \right]}{\sum_{k: A_{ik} = A_{ij}} y_k} \]
\[ \frac{dx_i}{dt} = \frac{\alpha x_i}{\tau} (u_i - x^T u) + x_i \alpha \sum_j x_j \ln \left( \frac{x_j}{x_i} \right) \]

FALA
\[ \frac{dx_i}{dt} = \alpha x_i [(Ay)_i - x^T Ay] \]

RM
\[ \frac{dx_i}{dt} = \frac{\lambda x_i [(Ay)_i - x^T Ay]}{1 - \lambda [\max_k (Ay)_k - x^T Ay]} \]
Game Theoretic Intuitions

FAQ and Prisoner’s Dilemma
### Game Theoretic Intuitions

<table>
<thead>
<tr>
<th></th>
<th>Football</th>
<th>Ballet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Football</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Ballet</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

**FAQ self play**

**LFAQ self play**
Game Theoretic Intuitions

**FAQ vs. LFAQ mixed play**

Battle of the Sexes  
Stag Hunt
# Game Theoretic Intuitions

## Switching dynamics

<table>
<thead>
<tr>
<th>Rewards</th>
<th>State 1</th>
<th>State 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>0.3, 0.3</td>
<td>0.4, 0.4</td>
</tr>
<tr>
<td></td>
<td>0.2, 0.2</td>
<td>0.1, 0.1</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>1, 0</td>
<td>1, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transitions</th>
<th>2 State PD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C,C)→(0.9,0.1)</td>
<td>(C,C)→(0.1,0.9)</td>
</tr>
<tr>
<td>(C,D)→(0.1,0.9)</td>
<td>(C,D)→(0.9,0.1)</td>
</tr>
<tr>
<td>(D,C)→(0.1,0.9)</td>
<td>(D,C)→(0.9,0.1)</td>
</tr>
<tr>
<td>(D,D)→(0.9,0.1)</td>
<td>(D,D)→(0.1,0.9)</td>
</tr>
</tbody>
</table>
Game Theoretic Intuitions
Other paradigms

- Swarm Intelligence: Haitham Bou-Ammar, Karl Tuyls, Michael Kaisers: Evolutionary Dynamics of Ant Colony Optimization. MATES 2012: 40-52

(Some) References

2. From Normal Form to Markov Games
Game Theory 101

- Game theory's role in multi-agent learning:
  - Model of agent interactions
  - Analytic toolkit for evaluating agents
  - Consistent driver of innovations in learning algorithms

- **Objective:**
  
  Provide foundational & intuitive understanding of key game theory concepts
From Normal Form to Markov Games

Normal Form Games
- Definitions:
  - Model
  - Solution concepts

Markov Games
- Definitions:
  - Model
  - Optimal policy

- Algorithms Based on Best Response
- Learning in Markov Games (Part II)
From Normal Form to Markov Games

Normal Form Games

Definitions:
- Model
- Solution concepts

Algorithms Based on Best Response

Markov Games

Definitions:
- Model
- Optimal policy

Learning in Markov Games (Part II)
Normal Form Games: Formal Description

Let’s start with a two-player Normal Form Game (NFG):

### Player 1 payoff table $R^1$

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1_1$</td>
<td>$a^2_1$</td>
</tr>
<tr>
<td>$a^1_2$</td>
<td>$a^2_2$</td>
</tr>
<tr>
<td>$a^1_n$</td>
<td>$a^2_n$</td>
</tr>
</tbody>
</table>

### Player 2 payoff table $R^2$

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^1_1$</td>
<td>$r^1(a^1_i,a^2_j)$</td>
</tr>
<tr>
<td>$a^1_2$</td>
<td>$r^2(a^1_i,a^2_j)$</td>
</tr>
<tr>
<td>$a^1_n$</td>
<td></td>
</tr>
</tbody>
</table>

If **pure** strategies are selected according to **mixed strategies** $\pi^1$ and $\pi^2$ (i.e., $a^1 \sim \pi^1$ and $a^2 \sim \pi^2$):

Player 1 will receive $E_{\pi_1,\pi_2}[r^1(a^1,a^2)] = \pi^1^T R^1 \pi^2$

Player 2 will receive $E_{\pi_1,\pi_2}[r^2(a^1,a^2)] = \pi^1^T R^2 \pi^2$
Normal Form Games: Solution Concept

- Next step: analyze agent behaviors given this model of interactions
- A **solution concept** is a formal set of principles that can be:
  - Descriptive: forecasts how agents **will** behave
  - Prescriptive: suggests how agents **should** behave
Normal Form Games: Solution Concept

Best response (BR): the strategy with highest payoff for a player, given knowledge of the other players’ strategies

\[ \pi^{2,\text{BR}} = \text{BR}(\pi^1=(1,0)) = (1,0) \]

\[ \pi^{2,\text{BR}} = \text{BR}(\pi^1=(0,1)) = (0,1) \]
Normal Form Games: Solution Concept

● **Nash Equilibrium:**

A strategy profile where all players in simultaneous best responses to each other

\[
\max_{\pi} \begin{pmatrix} \pi^T \end{pmatrix} R \begin{pmatrix} \pi \end{pmatrix}^2 = \begin{pmatrix} \pi^1 \end{pmatrix}^T R \begin{pmatrix} \pi^1 \end{pmatrix}^2 \quad \text{and} \quad \max_{\pi} \begin{pmatrix} \pi^1 \end{pmatrix}^T R^2 \begin{pmatrix} \pi \end{pmatrix} = \begin{pmatrix} \pi^1 \end{pmatrix}^T R^2 \begin{pmatrix} \pi \end{pmatrix}^2
\]

i.e., no player can do better by unilaterally deviating

● **Nash’s theorem [1950]:**

Every finite game has a mixed strategy Nash equilibrium

● Not unique in general → equilibrium selection problem
Normal Form Games: Solution Concept

Nash equilibria and their expected payoffs:

1. $\pi^1, \pi^2 = (1,0), (1,0) \rightarrow (2,1)$
2. $\pi^1, \pi^2 = (0,1), (0,1) \rightarrow (1,2)$
3. $\pi^1, \pi^2 = (\frac{2}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{2}{3}) \rightarrow (\frac{2}{3}, \frac{2}{3})$

- Very different outcomes!
- Intractable in general [Daskalakis et al., 2009]
  - Though polynomial-time computable for two-player zero-sum games
Nash equilibria and their expected payoffs:

1. $\pi^1,\pi^2 = (0,1), (1,0) \rightarrow (1,0)$
2. $\pi^1,\pi^2 = (1,0), (0,1) \rightarrow (0,1)$
3. $\pi^1,\pi^2 = \left(\frac{100}{101}, \frac{1}{101}\right), \left(\frac{100}{101}, \frac{1}{101}\right) \rightarrow (0,0)$

3rd equilibrium may seem reasonable, but >0 probability of (-100,-100) reward for both players!
Normal Form Games: Solution Concept

A better alternative might be to play the distribution on the right:

Unfortunately, no set of independent mixed strategies can result in this joint distribution!
Normal Form Games: Solution Concept

- **Idea:** address the issue of independent randomness by using a joint distribution
  - Correlated equilibria

A correlated equilibrium is a distribution, $D$, over strategy profiles such that for every player $i$:

$$E_{a \sim D} [r^i(a^i, a^{-i}) \mid a^i] \geq \max_a E_{a \sim D} [r^i(a, a^{-i}) \mid a^i]$$

<table>
<thead>
<tr>
<th></th>
<th>Stop</th>
<th>Go</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Go</td>
<td>1</td>
<td>-100</td>
</tr>
</tbody>
</table>

Sampled action for player $i$

Joint action samples
Normal Form Games: Solution Concept

- **Idea:** address the issue of independent randomness by using a joint distribution
  - Correlated equilibria
Topology of Solution Concepts

- Pure Nash equilibrium
- Mixed Nash equilibrium
- Correlated Equilibrium
- Coarse Correlated Equilibrium
From Normal Form to Markov Games

Normal Form Games
Definitions:
- Model
- Solution concepts

Markov Games
Definitions:
- Model
- Optimal policy

Algorithms Based on Best Response

Learning in Markov Games (Part II)
So far: solution concepts (e.g., Nash Equilibria) given full knowledge of game

Learning dynamics: do the dynamical interactions of players with limited knowledge lead to these solution concepts?
Let’s weaken our assumptions:

- Players interact in rounds
- Each player knows their own strategy, but not the full payoff table
- After each round, each player observes their pure strategies’ expected payoffs:
  
  Player 1 observes vector $\mathbf{R}_1 \pi^2$
  
  Player 2 observes vector $\pi^1 \mathbf{R}_2$
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:
  - Play a best response w.r.t. history of play in the $T$ previous rounds

\[
\pi^1 \in \arg\max_{\pi_\epsilon} \pi^T \left( \frac{1}{T} \sum_t R^1 \pi^2_t \right)
\]

- “Fictitious” in the sense that each player maintains a belief over opponent strategies according the play history

\[
\pi^1 \in \arg\max_{\pi_\epsilon} \pi^T R^1 \left( \frac{1}{T} \pi^2_t \right)
\]

- Observed payoff vector in round $t$

- Time-average opponent play
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Unique mixed Nash:

\[ \pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right), \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right) \]
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Unique mixed Nash:

\[ \pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right), \quad \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right) \]
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Unique mixed Nash:
\[ \pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right), \quad \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right) \]
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Player 2 chooses $\text{argmin}$ of $n_{t-1}^1$

Player 1 chooses $\text{argmax}$ of $n_{t-1}^2$

Unique mixed Nash:

$\pi_t^1 = (\frac{1}{2}, \frac{1}{2})$, $\pi_t^2 = (\frac{1}{2}, \frac{1}{2})$

<table>
<thead>
<tr>
<th></th>
<th>$\pi_t^1$</th>
<th>$\pi_t^2$</th>
<th>$n_t^1 (H,T)$</th>
<th>$n_t^2 (H,T)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
<td>(1, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Player 2

<table>
<thead>
<tr>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Player 1

<table>
<thead>
<tr>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Unique mixed Nash:

\[ \pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right), \quad \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right) \]

<table>
<thead>
<tr>
<th>t</th>
<th>( \pi_t^1 )</th>
<th>( \pi_t^2 )</th>
<th>( n_t^1 (H, T) )</th>
<th>( n_t^2 (H, T) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
<td>(1, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Player 2 chooses argmin of \( n_{t-1}^1 \)

Player 1 chooses argmax of \( n_{t-1}^2 \)
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Unique mixed Nash:

\[ \pi^1_t = (\frac{1}{2}, \frac{1}{2}), \quad \pi^2_t = (\frac{1}{2}, \frac{1}{2}) \]
**Normal Form Games: Fictitious Play**

- Fictitious Play [Brown, 1951]:

![Game Matrix]

<table>
<thead>
<tr>
<th></th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Player 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Unique mixed Nash:**

\[ \pi_t^1 = (\frac{1}{2}, \frac{1}{2}), \quad \pi_t^2 = (\frac{1}{2}, \frac{1}{2}) \]

<table>
<thead>
<tr>
<th>t</th>
<th>(n_t^1) (H,T)</th>
<th>(n_t^2) (H,T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Player 2 chooses argmin of \(n_{t-1}^1\)
- Player 1 chooses argmax of \(n_{t-1}^2\)
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Unique mixed Nash:

\[ \pi^1_t = \left( \frac{1}{2}, \frac{1}{2} \right), \quad \pi^2_t = \left( \frac{1}{2}, \frac{1}{2} \right) \]
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

![Game Matrix]

Player 2 chooses \( \text{argmin} \) of \( n_{t-1}^1 \)

Player 1 chooses \( \text{argmax} \) of \( n_{t-1}^2 \)

Unique mixed Nash:

\( \pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right) \), \( \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right) \)
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

$$\begin{array}{cccc}
\text{Player 2} & \text{Player 1} & H & T \\
H & 1 & -1 & -1 \\
T & -1 & 1 & 1 \\
\end{array}$$

Unique mixed Nash:

$$\pi_t^1 = \left( \frac{1}{2}, \frac{1}{2} \right), \quad \pi_t^2 = \left( \frac{1}{2}, \frac{1}{2} \right)$$

<table>
<thead>
<tr>
<th>t</th>
<th>$\pi_t^1$</th>
<th>$\pi_t^2$</th>
<th>$n_t^1$ (H,T)</th>
<th>$n_t^2$ (H,T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
<td>(1, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>H</td>
<td>(2, 2)</td>
<td>(2, 0)</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>T</td>
<td>(3, 2)</td>
<td>(2, 1)</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
<td>T</td>
<td>(4, 2)</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Player 2 chooses $\text{argmin of } n_{t-1}^1$

Player 1 chooses $\text{argmax of } n_{t-1}^2$
Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Player 2 chooses $\arg\min n^{t-1}$
Player 1 chooses $\arg\max n^{t-1}$

Unique mixed Nash:

$$\pi^1_t = (\frac{1}{2}, \frac{1}{2}), \pi^2_t = (\frac{1}{2}, \frac{1}{2})$$

<table>
<thead>
<tr>
<th>$t$</th>
<th>$\pi^1_t$</th>
<th>$\pi^2_t$</th>
<th>$n^1_{t-1}(H,T)$</th>
<th>$n^2_{t-1}(H,T)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
<td>(1, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>H</td>
<td>(2, 2)</td>
<td>(2, 0)</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>T</td>
<td>(3, 2)</td>
<td>(2, 1)</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
<td>T</td>
<td>(4, 2)</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5</td>
<td>T</td>
<td>T</td>
<td>(4, 3)</td>
<td>(2, 3)</td>
</tr>
<tr>
<td>6</td>
<td>T</td>
<td>T</td>
<td>(4, 4)</td>
<td>(2, 4)</td>
</tr>
<tr>
<td>7</td>
<td>T</td>
<td>H</td>
<td>(4, 5)</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>8</td>
<td>T</td>
<td>H</td>
<td>(4, 6)</td>
<td>(4, 4)</td>
</tr>
</tbody>
</table>
Normal Form Games: Fictitious Play

- **Fictitious Play [Brown, 1951]:**

<table>
<thead>
<tr>
<th>t</th>
<th>$\pi^1_t$</th>
<th>$\pi^2_t$</th>
<th>$n^1_t (H,T)$</th>
<th>$n^2_t (H,T)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>(0, 2)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>H</td>
<td>(1, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>2</td>
<td>H</td>
<td>H</td>
<td>(2, 2)</td>
<td>(2, 0)</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>T</td>
<td>(3, 2)</td>
<td>(2, 1)</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
<td>T</td>
<td>(4, 2)</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>5</td>
<td>T</td>
<td>T</td>
<td>(4, 3)</td>
<td>(2, 3)</td>
</tr>
<tr>
<td>6</td>
<td>T</td>
<td>T</td>
<td>(4, 4)</td>
<td>(2, 4)</td>
</tr>
<tr>
<td>7</td>
<td>T</td>
<td>H</td>
<td>(4, 5)</td>
<td>(3, 4)</td>
</tr>
<tr>
<td>8</td>
<td>T</td>
<td>H</td>
<td>(4, 6)</td>
<td>(4, 4)</td>
</tr>
</tbody>
</table>

Unique mixed Nash:

$\pi^1_t = (\frac{1}{2}, \frac{1}{2})$, $\pi^2_t = (\frac{1}{2}, \frac{1}{2})$

Play will continue to cycle deterministically, with time-average strategies converging to Nash.
Normal Form Games: Fictitious Play

- When does Fictitious Play converge, and to what?

- Average-time strategies of fictitious players converge to a Nash in:
  - Two-player zero-sum games
  - 2x2 games
  - Potential games
  - ...

- Not guaranteed in general! Try it on modified RPS:
Normal Form Games: Oracle Algorithms

- **Goal**: compute a Nash equilibrium of the game (AKA “solve” the game)
- **Insight**: computing a best response is generally cheaper than solving the game

  - Reduction to a single-player optimization problem
  - Due to their efficiency, BR algorithms sometimes called “oracles”

- Oracle algorithms use BR to solve the game:
  - Single/double oracle: one/both player(s) use the oracle algorithm
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

![Diagram showing full game and restricted game, with arrows indicating arbitrary initial policies and the matrices representing the game states.](image-url)
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

\[
\begin{array}{cc}
\pi^2_0 & \pi^2_1 \\
5 & 8 \\
3 & 2 \\
\end{array}
\]

Arbitrary initial policies

Compute restricted game
Nash equilibrium \((p^0,q^0)\)
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

<table>
<thead>
<tr>
<th>( \pi^2_0 )</th>
<th>( \pi^2_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Arbitrary initial policies

Compute restricted game
Nash equilibrium \((p^n, q^n)\)

Compute BR to \((p^n, q^n)\), given access to strategies in full game

Full Game

Restricted game
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

  
  \[
  \begin{array}{c|ccc}
  & \pi_1^0 & \pi_1^1 & \pi_2^1 \\
  \hline
  \pi_0^1 & 5 & 8 & 10 \\
  \pi_1^1 & 3 & 2 & 2 \\
  \pi_2^1 & 7 & 8 & 0 \\
  \end{array}
  \]

  Expand restricted game, adding BR strategies \( \pi_2^1 \) and \( \pi_2^2 \).

  Compute restricted game, Nash equilibrium \((p^n,q^n)\).

  Compute BR to \((p^n,q^n)\), given access to strategies in full game.

  Full Game

  Restricted game

  Arbitrary initial policies
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:
  - Iteration 0: restricted game of R vs. R
  - Iteration 1:
    - Solve restricted game:
      - $(1, 0, 0), (1, 0, 0)$
    - Unrestricted $BR^1_1, BR^2_1 = P, P$

```
   R
R | 0
```

DeepMind
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:
  - Iteration 0: restricted game of R vs. R
  - Iteration 1:
    - Solve restricted game:
      
      \[
      (1, 0, 0), (1, 0, 0)
      \]
    - Unrestricted BR
      
      \[
      1\text{, }1, 1\text{, }2 = P, P
      \]
  - Iteration 2:
    - Solve restricted game:
      
      \[
      (0, 1, 0), (0, 1, 0)
      \]
    - Unrestricted BR
      
      \[
      1\text{, }2, 2\text{, }2 = S, S
      \]
Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

<table>
<thead>
<tr>
<th>R</th>
<th>P</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Iteration 0: restricted game of R vs. R
- Iteration 1:
  - Solve restricted game:
    - \((1, 0, 0), (1, 0, 0)\)
  - Unrestricted BR\(_1\), BR\(_2\) = P, P
- Iteration 2:
  - Solve restricted game:
    - \((0, 1, 0), (0, 1, 0)\)
  - Unrestricted BR\(_1\), BR\(_2\) = S, S
- Iteration 2:
  - Solve restricted game:
    - \((\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})\)
Normal Form Games: Oracle Algorithms

- Computation time improvements vs. solving full game [McMahan et al., 2003]:

Table 1. Sample problem discretizations, number of sensor placements available to the opponent, solution time using Equation 4, and solution time and number of iterations using the Double Oracle Algorithm.

<table>
<thead>
<tr>
<th></th>
<th>grid size</th>
<th>k</th>
<th>LP</th>
<th>Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>54 x 45</td>
<td>32</td>
<td>56.8 s</td>
<td>1.9 s</td>
</tr>
<tr>
<td>B</td>
<td>54 x 45</td>
<td>328</td>
<td>104.2 s</td>
<td>8.4 s</td>
</tr>
<tr>
<td>C</td>
<td>94 x 79</td>
<td>136</td>
<td>2835.4 s</td>
<td>10.5 s</td>
</tr>
<tr>
<td>D</td>
<td>135 x 113</td>
<td>32</td>
<td>1266.0 s</td>
<td>10.2 s</td>
</tr>
<tr>
<td>E</td>
<td>135 x 113</td>
<td>92</td>
<td>8713.0 s</td>
<td>18.3 s</td>
</tr>
<tr>
<td>F</td>
<td>269 x 226</td>
<td>16</td>
<td>-</td>
<td>39.8 s</td>
</tr>
<tr>
<td>G</td>
<td>269 x 226</td>
<td>32</td>
<td>-</td>
<td>41.1 s</td>
</tr>
</tbody>
</table>
Normal Form Games: Algorithms

- When does Double Oracle converge, and to what?

- Convergence guaranteed for two-player finite games
  - Proof: worst case, the restricted game just expands to the full game

- Convergence to minimax equilibrium in finite games [McMahan et al. 2003]
From Normal Form to Markov Games

Normal Form Games

Definitions:
- Model
- Solution concepts

Markov Games

Definitions:
- Model
- Optimal policy

Algorithms Based on Best Response

Learning in Markov Games (Part II)
Markov Games: Description

Setting (e.g., in a 2-player game):
- Agents in environment with state $s$
Markov Games: Description

Setting (e.g., in a 2-player game):
- Agents in environment with state \( s \)
- Simultaneously select actions \( a^1 \) & \( a^2 \)
- Receive rewards \( r^1(s,a^1,a^2) \) & \( r^2(s,a^1,a^2) \)
Markov Games: Description

Setting (e.g., in a 2-player game):
- Agents in environment with state $s$.
- Simultaneously select actions $a^1$ & $a^2$.
- Receive rewards $r^1(s,a^1,a^2)$ & $r^2(s,a^1,a^2)$.
- Move to state $s' \sim p(.|s,a^1,a^2)$. 

\[ r^1(s,a^1,a^2), r^2(s,a^1,a^2) \]
Markov Games: Description

Setting (e.g., in a 2-player game):
- Agents in environment with state $s$
- Simultaneously select actions $a^1$ & $a^2$
- Receive rewards $r^1(s,a^1,a^2)$ & $r^2(s,a^1,a^2)$
- Move to state $s' \sim p(.|s,a^1,a^2)$

Goal: find the “optimal” policy

If actions are selected according to policies $\pi^1(.|s)$ & $\pi^2(.|s)$, i.e., $a^1 \sim \pi^1(.|s)$ and $a^2 \sim \pi^2(.|s)$:

Player 1 receives $v_{\pi_1,\pi_2}^1(s_0) = E_{\pi_1,\pi_2} \left[ r^1(s_0,a^1_0,a^2_0) + \gamma r^1(s_1,a^1_1,a^2_1) + \ldots \right]$

Player 2 receives $v_{\pi_1,\pi_2}^2(s_0) = E_{\pi_1,\pi_2} \left[ r^2(s_0,a^1_0,a^2_0) + \gamma r^2(s_1,a^1_1,a^2_1) + \ldots \right]$

Discount factor $\in [0,1)$
From Normal Form to Markov Games

Normal Form Games

Definitions:
- Model
- Solution concepts

Algorithms Based on Best Response

Markov Games

Definitions:
- Model
- Optimal policy

Learning in Markov Games (Part II)
References

3. Social Learning
Social dilemmas

Situations where any individual may profit from selfishness unless too many individuals choose the selfish option, in which case the whole group loses.

“Social dilemmas expose tensions between collective and individual rationality”

-Anatol Rapoport (1974)
Social dilemmas \cite{Liebrand1983, Macy2002}

- **Reward** for mutual cooperation
- **Sucker** for cooperating with defector
- **Punishment** for mutual defection
- **Temptation** to defect on a cooperator

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>R, R</td>
<td>S, T</td>
</tr>
<tr>
<td>D</td>
<td>T, S</td>
<td>P, P</td>
</tr>
</tbody>
</table>

1. \( R > P \) (mutual cooperation better than mutual defection)
2. \( R > S \) (mutual cooperation better than being exploited)
3. \( T > P \) (being greedy better than being punished)
4. either (fear) \( S < P \) (being sucker worse than mutual defection) … or (greed) \( T > R \) (being greedy better than mutual cooperation)
Sequential Social Dilemmas

- MGSDs are defined as repeated matrix games for which the social dilemma inequalities hold.
- The social dilemma inequalities enforce the mixed motivation structure of the game: both competition and cooperation are motivated.
- SSDs are defined by an EGTA mapping to an associated MGSD.

- MGSD = Matrix Game Social Dilemma
- SSD = Sequential Social Dilemma
- EGTA = Empirical Game Theory Analysis
Can we design an agent that can promote cooperation and take fairness into account in SSDs?

Can we do this based on the Fehr and Schmidt model of inequity aversion?

- MGSD = Matrix Game Social Dilemma
- SSD = Sequential Social Dilemma
- EGTA = Empirical Game Theory Analysis
Examples (Leibo et al. 2017)

**Gathering**
- Cooperation = not tagging
- Defection = tagging

**Wolfpack**
- Cooperation = team capture
- Defection = individual capture
Proving that these are SSDs (by Schelling diagrams)
Examples

- Each line shows the payoff to an individual agent (y) for choosing C or D as a function of number of others that chose C (x).
The Fehr and Schmidt model (Fehr and Schmidt, 1999)

\[ U_i(r_i, \ldots r_N) = r_i \]

\[ - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max(r_j - r_i, 0) \] ← envy

\[ - \frac{\beta_i}{N-1} \sum_{j \neq i} \max(r_i - r_j, 0) \] ← guilt
The inequity-averse agent model (Hughes, Leibo, Tuyls et al. 2018)

\[ u_i(s^t_i, a^t_i) = r_i(s^t_i, a^t_i, \theta_{ii}) \]
\[ - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max(e^t_j r_j(s^t_j, a^t_j, \theta_{ij})) \]
\[ - e^t_i r_i(s^t_i, a^t_i, \theta_{ii}), 0) \]
\[ - \frac{\beta_i}{N-1} \sum_{j \neq i} \max(e^t_j r_j(s^t_j, a^t_j, \theta_{ii})) \]
\[ - e^t_j r_j(s^t_j, a^t_j, \theta_{ij}), 0) , \]

---

envy

guilt
Envy and guilt

![Graph showing the relationship between smoothed reward and timestep with annotations for envy and guilt.](image-url)
The Tragedy of the Commons (Hardin 1968)

Tension between collective and individual rationality.
1. Agents move around on a grid world.
2. Agents are only rewarded when they collect an apple.
3. The apple growth rule is density dependent. So apples grow more quickly adjacent to nearby apples.
4. If all the apples in a local patch are removed then none grow back.
5. Episodes last 1000 steps, after which the game resets to its initial condition.
6. Agents have a “time-out beam” with which they can zap one another. A zapped agent gets removed from the game for 25 steps.
The Commons Game

- $N = 10$ players
- Each agent can individually profit from selfishness, but the group is doomed if all elect that option.
- There can be a “tragedy of the commons” (G. Hardin 1968)
Multiple social outcome metrics

Societal-level measurement is complicated!

1. **Utilitarian efficiency (U)** = total reward (sum over all players)
2. **Sustainability (S)** = average time of reward collection in episode
3. **Peacefulness (P)** = average number of unzapped agent steps

Only illustrate a couple of experiments
Envious agents become police
Envious agents become police
The Public Goods Game  (Hughes, Leibo, Tuyls et al. 2018)
1. Agents move around on a grid world.
2. Agents are only rewarded when they collect an apple.
3. The apple growth rule is dependent on the waste density. The lower the waste, the higher the apple growth.
4. Initially the waste density is so high that no apples can spawn.
5. Episodes last 1000 steps, after which the game resets to its initial condition.
6. Agents have a “fining beam” with which they can zap one another. Fining costs -1 reward, and causes the fined agent -50 reward.
Guilty agents provide public goods
Guilty agents provide public goods
Take home

● Understanding several MAL paradigms within 1 framework

● EGT as a tool to capture MAL dynamics

● Deep Reinforcement Learning opens new possibilities in many respects, revisiting some of the old results

● Evaluation, Dynamics, and new Algorithmics
Part II. Evaluation & Learning

4. Evaluation
5. Gradients in Games
6. Multi-agent Learning at Scale
7. The Importance of Games
4. Evaluation
How to evaluate agents in a multi-agent context?
Overview

Elo Rating
- Static score
- Cannot capture dynamics
- Cannot deal with intransitivities

Empirical Game Theory

Continuous-time Evolutionary Dynamics
- Limited to evaluating 3/4 agents
- Stable/unstable Nash equilibria
- Generally intractable to compute & select

Discrete-time Evolutionary Dynamics
- Many-agent interactions
- Stable agents & Markov-Conley Chains
- Unique, tractable to compute & select

Little hope for a general predictive theory in terms of Nash equilibrium

Elo Evaluation

“The logic of the equation is evident without algebraic demonstration: a player performing above his expectancy gains points, and a player performing below his expectancy loses points.” – Arpad E. Elo

- Update rule:
  \[ R_{t+1}^i = R_t^i + K[S_{t+1}^i - E_t^i] \]

- Win probability:
  \[ p_{ij} = \frac{1}{1 + e^{-\alpha(R_i - R_j)}} \]

- Chess:
  \[ p_{ij} = \frac{1}{1 + 10^{(R_i - R_j)/400}} \]

Elo picked 10 as basis and 400 as the denominator because then a difference of 400 points corresponds to a 90% winning probability.
**Elo Evaluation**

\[
\begin{bmatrix}
2&0&0
1&0&1
0&2&0
1&1&0
0&0&2
0&1&1
\end{bmatrix} \quad \begin{bmatrix}
U_{i1} \\
U_{i2} \\
U_{i3}
\end{bmatrix}
\begin{bmatrix}
0.5 & 0 & 0 \\
0.014 & 0 & 0.986 \\
0 & 0.5 & 0 \\
0.03 & 0.97 & 0 \\
0 & 0 & 0.5 \\
0 & 0.3 & 0.7 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
2&0&0
1&0&1
0&2&0
1&1&0
0&0&2
0&1&1
\end{bmatrix} \quad \begin{bmatrix}
U_{i1} \\
U_{i2} \\
U_{i3}
\end{bmatrix}
\begin{bmatrix}
0.5 & 0 & 0 \\
0.54 & 0 & 0.46 \\
0 & 0.5 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0.5 \\
0 & 0.45 & 0.55 \\
\end{bmatrix}
\]

8: 1330  
10: 1927  
21: 2069

In reality: 8>21, 21>10 and 10>8
Empirical Game Theory Analysis

- A symmetric multi-agent *Meta-Game*: 
  
  \((S, A, M, p\text{-type})\)

- Policies are atomic actions, \(|A|=n\)
- \(n\) does not need to equal \(p\)
- \(S\) and \(A\) can coincide
- E.g. Go dataset: \((S, A, M, 2\text{-type})\)
  - \(|A|=30\) and \(S=A\)

### Payoff table from data

\[
P = \begin{pmatrix}
N_{i1} & N_{i2} & N_{i3} & u_{i1} & u_{i2} & u_{i3} \\
6 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 2 & -0.5 & 0 & 1 \\
0 & 0 & 6 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
N_{i1,j1} & N_{i2,j2} & N_{i3,j3} & u_{i1,j1} & u_{i2,j2} & u_{i3,j3} \\
(1,1) & 0 & 0 & (2,3) & 0 & 0 \\
(1,0) & (0,1) & 0 & (0.5,0) & (0,0.5) & 0 \\
(0,1) & (1,0) & 0 & (0,0.4) & (0.3,0) & 0 \\
0 & 0 & (1,1) & 0 & 0 & (3,2) \\
\end{pmatrix}
\]
Meta-Game analysis

- Example Rock-Paper-Scissors

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>(0,0)</td>
<td>(-1,1)</td>
<td>(1,-1)</td>
</tr>
<tr>
<td>Paper</td>
<td>(1,-1)</td>
<td>(0,0)</td>
<td>(-1,1)</td>
</tr>
<tr>
<td>Scissors</td>
<td>(-1,1)</td>
<td>(1,-1)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

- Strategy Space Consumption:
  - Use *sizes of basins* of attraction to rate strategies
  - Combine with *curl* and *sizes of differential*
Experiments

AlphaGo, Colonel Blotto, Leduc Poker
AlphaGo data set

Set of 30 strategies.

\[
\begin{pmatrix}
\alpha_{rp} & \alpha_{vp} & \alpha_{rp} \\
2 & 0 & 0 \\
1 & 0 & 1 \\
0 & 2 & 0 \\
1 & 1 & 0 \\
0 & 0 & 2 \\
0 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
U_1 \\
0.5 \\
0.95 \\
0 \\
0.99 \\
0 \\
0 \\
\end{pmatrix}
\begin{pmatrix}
U_2 \\
0 \\
0 \\
0.01 \\
0 \\
0 \\
0.39 \\
\end{pmatrix}
\begin{pmatrix}
U_3 \\
0 \\
0.05 \\
0 \\
0 \\
0.5 \\
0.61 \\
\end{pmatrix}
\]
This meta-analysis does not only show the attractor(s) and its (their) stability, but also how the multi-agent interaction flows through strategy space, and what the basins of attraction look like.
AlphaGo data set

The curl, size and direction of the differential play a role in the determination of the strength and weakness of a strategy in strategy space, and will be useful for the strategy space consumption concept.
AlphaGo data set

Go Leaderboard
Colonel Blotto Game

See [https://github.com/deepmind/open_spiel](https://github.com/deepmind/open_spiel) for description / implementation

- 2 players, 100 troops each
- Divide over 5 lands

```
[[20, 20, 20, 20, 20]]
[[33, 1, 32, 1, 33]]
```
Colonel Blotto

Examined 10 most played strategies

Also in the case of mixed Nash equilibria, the concepts are still eligible, and we can determine the strength of a strategy by computing how much it pulls the mixed equilibrium towards itself.
Leduc Poker (PSRO)

In asymmetric games we get a coupled system of replicator equations, resulting in a simplex for each player over its respective strategy sets. The dynamics are now more complex (and coupled), but still these plots provide insightful information w.r.t. equilibria and the flow of dynamics.

An interesting, previously unknown result, is that a mixed Nash Equilibrium \((x,y)\) in the asymmetric game is also a mixed Nash Equilibrium in the symmetrised games, i.e., the y-component for the row player’s game, and the x-component in the column player’s game. The reverse is also true.
In Conclusion

- EGT/meta-games well suited for both **symmetric** and **asymmetric games**
  - Poker, Go, Auctions, Robotics
- Provide bounds that tell you how reliable the estimated game is
- Limited to 3/4 strategies
Multi-Agent Evaluation

**Elo Rating**
- Static score
- Cannot capture dynamics
- Cannot deal with intransitivities

**Empirical Game Theory**

**Continuous-time Evolutionary Dynamics**
- Limited to evaluating 3/4 agents
- Stable/unstable Nash equilibria
- Generally intractable to compute & select

**Discrete-time Evolutionary Dynamics**
- Many-agent interactions
- Stable agents & Markov-Conley Chains
- Unique, tractable to compute & select

Little hope for a **general predictive theory** in terms of Nash equilibrium
Analogous to Nash using Kakutani’s fixed point theorem as a basis for his solution concept, we use Conley’s Fundamental Theorem of Dynamical Systems (Conley, 1978):

“All flow on a compact metric space decomposes into a gradient-like part that leads to a recurrent part.”

Markov-Conley Chains (MCCs) are the discrete analogs of the recurrent set above:

- Capture irreducible long-term dynamical interactions between agents
- Correspond to the unique stationary distribution of an underlying discrete-time evolutionary process
- Pinpoint diverse set of agents that are evolutionarily stable (cannot be mutated or invaded)
A Dynamical Solution Concept

- Caveat: difficult to study these recurrent sets theoretically
  - We need a **meaningful approximation** that can be tractably analyzed
- **Response graph**: directed graph where nodes correspond to pure strategy profiles, and directed edges if the deviating player’s new strategy is a better-response

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>C</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
<td>(2,0)</td>
<td>(0,2)</td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>M</strong></td>
<td>(0,2)</td>
<td>(2,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>
A Dynamical Solution Concept

- Caveat: difficult to study these recurrent sets theoretically
  - We need a **meaningful approximation** that can be tractably analyzed
- **Response graph**: directed graph where nodes correspond to pure strategy profiles, and directed edges if the deviating player’s new strategy is a better-response
- **Markov-Conley chains (MCCs)**:
  - Markov chains over the sink strongly connected components of response graph
  - Our dynamical solution concept!

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>C</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
<td>(2,0)</td>
<td>(0,2)</td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>M</strong></td>
<td>(0,2)</td>
<td>(2,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>
Markov-Conley chains (MCCs):

- Markov chains over the \textit{sink} strongly connected components of response graph
- \textit{Hint}: a directed graph is strongly connected if there is a path between all pairs of its vertices.

How many MCCs exist in the below response graph?

A. 0
B. 1
C. 2
D. 9

<table>
<thead>
<tr>
<th>Player 1</th>
<th>(L)</th>
<th>(C)</th>
<th>(R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U)</td>
<td>(2,0)</td>
<td>(0,2)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>(M)</td>
<td>(0,2)</td>
<td>(2,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>(D)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>
Quiz Question

- **Markov-Conley chains (MCCs):**
  - Markov chains over the sink strongly connected components of response graph

How many MCCs exist in the below response graph?

A. 0  
B. 1  
C. 2  
D. 9

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
<th>L</th>
<th>C</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td></td>
<td>(2,0)</td>
<td>(0,2)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>(0,2)</td>
<td>(2,0)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>
MCCs are computationally attractive, but face equilibrium selection issues akin to Nash.
MCCs are computationally attractive, but face equilibrium selection issues akin to Nash.

**Solution:** perturb the response graph such that a random walk can *climb upward* on the potential hills and hop between MCCs (sinks) with a very small probability.

- Irreducible Markov chain → unique stationary distribution → unique MCC rankings.
Linking MCCs and Evolution

- Remarkably, our perturbed model is equivalent to a **discrete-time evolutionary process**
  - Well-studied in the literature for pairwise/symmetric games
  - Generalized in our work to $K$-player asymmetric games
- **Basic idea:** model a selection-mutation process over a set of interacting populations
Remarkably, our perturbed model is equivalent to a discrete-time evolutionary process

- Well-studied in the literature for pairwise/symmetric games
- Generalized in our work to $K$-player asymmetric games

**Basic idea:** model a selection-mutation process over a set of interacting populations

- Strong agents (i.e., those resistant to mutants) propagate via a selection function:

  \[ \mathbb{P}(\tau \rightarrow \sigma, S^{-k}) = \left(1 + e^{\alpha(f^k(\tau, s^{-k}) - f^k(\sigma, s^{-k}))}\right)^{-1} \]

  - **Probability of competing agent $\sigma$ taking over**
  - **Fitness of resident agent $\tau$ vs. competing agent $\sigma$**

  - **Small $\alpha$**
  - **Weak selection**

**Ranking-intensity value $\alpha$**

- **Large $\alpha$**
- **Strong selection**
- **MCC solution concept**
- **$\alpha$-Rank**
**α-Rank Algorithm**

1. Construct the meta-game payoff tables from multi-agent simulations
2. Define a Markov chain where states are the agents being evaluated
3. Compute transition matrix $C$ according to an evolutionary process with selection-intensity parameter $\alpha$
4. Compute the unique stationary distribution $\pi$ of $C$
5. Agent rankings/scores correspond to the ordered masses of $\pi$

- Ranking guaranteed to exist and is unique
- Handles cycles/intransitivities
- Scalable and applies to general-sum, symmetric/asymmetric, many-player games
Unified View of Multi-agent Evaluation by Evolution

**Macro-model: Discrete-time Dynamics**

**Analytical toolkit:**
- Markov chain
- Stationary distribution
- Fixation probabilities

**Applicability:**
- K-wise interactions
- Symmetric and asymmetric games

**Foundations:**
- Conley’s Fundamental Theorem
- Chain recurrent sets and components

**Advantages:**
- Captures dynamic behavior
- More tractable to compute than Nash
- Filters out transient agents
- Involves only a single hyperparameter, $\alpha$

---

**Micro-model: Continuous-time Dynamics**

**Analytical toolkit:**
- Flow diagrams sub-graph
- Attractors, equilibria

**Applicability:**
- 3 to 4 agents max
- Symmetric games and 2-population asymmetric games

---

**Unifying ranking model:**
**Markov Conley Chains & $\alpha$-Rank**

**Selection-intensity parameter $\alpha$**

**Agent Ranking**

- Agent | Rank | Score
- A1 | 1 | 0.18
- A2 | 2 | 0.23
- A3 | 3 | 0.18
- A4 | 4 | 0.18
- A5 | 5 | 0.09
- A6 | 6 | 0.00
- A7 | 7 | 0.00
- A8 | 8 | 0.00

---

**Foundations:**
- Conley’s Fundamental Theorem
- Chain recurrent sets and components

---

DeepMind
Demonstrations

- Rock-Paper-Scissors (2-player, symmetric, 3 agents)

<table>
<thead>
<tr>
<th>Agent</th>
<th>Rank</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1</td>
<td>0.33</td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>0.33</td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Demonstrations

- AlphaZero Chess (2-player game, 56 agent snapshots taken during training)

Top-8 agents shown

<table>
<thead>
<tr>
<th>Agent</th>
<th>Rank</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ(99.4)</td>
<td>1</td>
<td>0.39</td>
</tr>
<tr>
<td>AZ(93.9)</td>
<td>2</td>
<td>0.22</td>
</tr>
<tr>
<td>AZ(98.7)</td>
<td>3</td>
<td>0.19</td>
</tr>
<tr>
<td>AZ(94.7)</td>
<td>4</td>
<td>0.14</td>
</tr>
<tr>
<td>AZ(86.4)</td>
<td>5</td>
<td>0.05</td>
</tr>
<tr>
<td>AZ(88.8)</td>
<td>6</td>
<td>0.01</td>
</tr>
<tr>
<td>AZ(90.3)</td>
<td>7</td>
<td>0.00</td>
</tr>
<tr>
<td>AZ(93.3)</td>
<td>8</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Top-8 agents (training percent complete in parentheses)
Demonstrations

- Kuhn Poker (4-player, asymmetric, 256 agent profiles)

<table>
<thead>
<tr>
<th>Agent</th>
<th>Rank</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,3,3,2)</td>
<td>1</td>
<td>0.08</td>
</tr>
<tr>
<td>(2,3,3,1)</td>
<td>2</td>
<td>0.07</td>
</tr>
<tr>
<td>(2,3,3,2)</td>
<td>3</td>
<td>0.07</td>
</tr>
<tr>
<td>(3,3,3,1)</td>
<td>4</td>
<td>0.06</td>
</tr>
<tr>
<td>(3,3,3,3)</td>
<td>5</td>
<td>0.06</td>
</tr>
<tr>
<td>(3,2,3,3)</td>
<td>6</td>
<td>0.05</td>
</tr>
<tr>
<td>(2,3,2,1)</td>
<td>7</td>
<td>0.04</td>
</tr>
<tr>
<td>(2,3,2,2)</td>
<td>8</td>
<td>0.04</td>
</tr>
<tr>
<td>(2,2,3,1)</td>
<td>9</td>
<td>0.04</td>
</tr>
<tr>
<td>(2,2,3,3)</td>
<td>10</td>
<td>0.03</td>
</tr>
<tr>
<td>(2,2,2,1)</td>
<td>11</td>
<td>0.03</td>
</tr>
<tr>
<td>(2,2,2,2)</td>
<td>12</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Top-12 profiles shown
Summary

- **α-Rank**: principled multi-agent evaluation method
  - To appear in Nature’s Scientific Reports journal, check out [arXiv draft](https://arxiv.org) for more:

  - **AlphaGo results**
  - **MuJoCo Soccer results**
  - **α-Rank vs. Nash in two-player games**
5. Gradients in Games
“If you have a large big dataset, and you train a very big neural network, then success is guaranteed!”
-- Ilya Sutskever (NIPS 2014)
The central dogma of deep (supervised) learning:

- compose **differentiable modules** into a neural net;
- convert data into a differentiable **objective function**;
- add **backprop**; and
- press go.

“If you have a large big dataset, and you train a very big neural network, then success is guaranteed!”

-- Ilya Sutskever (NIPS 2014)
How’d we get here?

Lots of “small” things:

- **differentiable modules:**
  - CNNs, LSTMs, ResNets, ReLUs, clever initializations, BatchNorm, ...

- **objective functions:**
  - datasets $\rightarrow$ losses

- **backprop:**
  - momentum, Adam, RMSProp, learning rates, hyper-parameters

- **press go:**
  - libraries (TensorFlow, PyTorch, ...) and GPUs take care of almost everything
One big thing: the loss landscape

Everything depends on gradient descent finding (good) local minima in the loss landscape
Trouble in paradise

- Modules aren't actually modules:
  - Trained NNs are nowhere near plug-and-play
  - NNs are invariably (re)trained from scratch
  - Not data-efficient

- Rampant overfitting
  - Transfer learning is extremely difficult
  - Adversarial examples

End-to-end learning doesn't scale
What’s next?

William Gibson: “The future is already here — it's just not very evenly distributed.”
What’s next?

William Gibson: “The future is already here — it’s just not very evenly distributed.”

- Generative Adversarial Networks (Goodfellow et al, NIPS 2014)
- Cycle-consistent adversarial nets (Zhu et al, ICCV 2017)
- Synthetic gradients (Jaderberberg et al, ICML 2017)
- Deep learning and neurosci (Marblestone et al, 2016)
- Intrinsic curiosity (Pathak et al, ICML 2017)
Generative adversarial networks

Image Credit - deeplearning4j.org
Cycle-GANs

cycle-consistency = \{ \text{learning a commutative diagram} \}
What’s next?

William Gibson: “The future is already here — it’s just not very evenly distributed.”

- Generative Adversarial Networks (Goodfellow et al, NIPS 2014)
- Cycle-consistent adversarial nets (Zhu et al, ICCV 2017)
- Synthetic gradients (Jaderberberg et al, ICML 2017)
- Deep learning and neurosci (Marblestone et al, 2016)
- Intrinsic curiosity (Pathak et al, ICML 2017)

Themes:

- Interacting losses and datasets
- It's hard work and ad hoc
A brief history of ML

- **Learning:**
  - **Why?** Don't want to hand-code behaviors
  - **Catch:** Weaker guarantees
A brief history of ML

- **Learning:**
  - *Why?* Don’t want to hand-code behaviors
  - *Catch:* Weaker guarantees

- **Learning representations:**
  - *Why?* Don’t want to hand-design features
  - *Catch:* Non-convex optimization
A brief history of ML

● Learning:
  ○ Why? Don't want to hand-code behaviors
  ○ Catch: Weaker guarantees

● Learning representations:
  ○ Why? Don't want to hand-design features
  ○ Catch: Non-convex optimization

● Learning losses:
  ○ Why? Don't want to hand-label data
  ○ Catch: ...
What’s the problem?
Minimal example

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

- Dynamics cycle around origin
But there’s no landscape

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

- Dynamics cycle around origin
- There’s **no** consistent “down direction”
But there's no landscape

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

- Dynamics cycle around origin
- There's no consistent "down direction"

**Technical problem:**
- Vector field isn't a gradient vector field
Three problems

1. Gradient descent isn’t guaranteed to converge (to anything, at all)
2. Even if it does, it can be very unstable and slow
3. Actually, can’t even measure progress

| Learning rate | 0.01 | 0.032 | 0.1 |
Which geometry?

Mathematicians and physicists have been studying geometry for centuries. There must be something on-the-shelf that we can use.
Div, grad, and curl

Helmholtz decomposition:
Any vector field in \( \mathbb{R}^3 \) decomposes as a sum of a gradient vector field (a curl-free or irrotational component) and a divergence-free component:

\[
\xi = \nabla \phi + \text{curl}(\rho)
\]

Escher-ish (measures infinitesimal tendency to rotate)

landscape-ish
Minimal example

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x, 0) \]
Minimal example

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x, 0) \]

- Vector field is divergence-free
  - There's no function that is being optimized
Minimal example

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x, 0) \]

- Vector field is divergence-free
  - There's no function that is being optimized

\[ \xi = \text{curl}(-xz, -yz, 0) \]

- ???
Minimal example

\[ \xi = \left( \frac{\partial l_1}{\partial x}, \frac{\partial l_2}{\partial y} \right) = (y, -x, 0) \]

- Vector field is divergence-free
  - There's no function that is being optimized

\[ \xi = \text{curl}(-xz, -yz, 0) \]

- ???
Which geometry?

Mathematicians and physicists have been studying geometry for centuries. There must be something on-the-shelf that we can use.

Actually, those cycles look like planetary orbits ...
Classical mechanics (in one slide)

Canonical coordinates: position $q$ and momentum $p = mv$

Hamiltonian: total (potential + kinetic) energy $\mathcal{H}(q, p)$

Dynamics: $\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{\partial p_i}$, $\frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{\partial q_i}$, $\xi = (\nabla_p \mathcal{H}, -\nabla_q \mathcal{H})$

Conservation of energy: $\langle \xi, \nabla \mathcal{H} \rangle = 0$

The dynamics lives on the level sets of the Hamiltonian.
Position, momentum, and conservation of energy don’t feature in good old fashioned game theory.
Eg: zero-sum bimatrix games

\[ \ell_1(x, y) = x^\top A y \quad \ell_2(x, y) = -x^\top A y \]

Singular value decomposition:

\[ A = U^\top D V \]

Change of coordinates:

\[ u = D^{\frac{1}{2}} U x \quad v = D^{\frac{1}{2}} V y \]

New losses:

\[ \ell_1(u, v) = u^\top v \quad \ell_2(u, v) = -u^\top v \]
Hamiltonian:

$$\mathcal{H}(u, v) = \frac{1}{2} (u^T u + v^T v)$$

Level sets are ellipses (in original coordinates)

Hamiltonian dynamics:

$$\xi = (\nabla_v \mathcal{H}, -\nabla_u \mathcal{H})$$

Eg: zero-sum bimatrix games

$$\ell_1(u, v) = u^T v \quad \ell_2(u, v) = -u^T v$$

$$\xi = (v, -u)$$
How to solve Hamiltonian games

- Level sets of the Hamiltonian (ellipses) are conserved by simultaneous gradient descent on the losses.

- Gradient descent on the Hamiltonian (not the losses) finds Nash equilibrium.
Game over?

- Constructing the Hamiltonian relied on simultaneously SVD-ability of losses.

- Can something like this be done in general? **No.**
The big picture

Potential games

Hamiltonian games

general games
The big picture

- PPAD hard (even for good old fashioned games)
- No tractable, general purpose method
The big picture

- Studied by game theorists for 30+ years
- “cooperative”
- Simultaneous gradient descent on **losses** finds local Nash

Potential games

Hamiltonian games
The big picture

- (supervised) deep learning lives here

Potential games

Hamiltonian games
The big picture

- New class of games
- "hyper-adversarial"
- Gradient descent on Hamiltonian finds local Nash
The big picture

- Gradient descent on Hamiltonian finds local Nash
- Gradient descent on losses finds local Nash

Hamiltonian games

Potential games

general games
Infinitesimal structure of gradients

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

**Game Hessian:**

\[ H_\xi = \begin{pmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{pmatrix} \]
Infinitesimal structure of gradients

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

Generalized Helmholtz decomposition:

\[ H_\xi = \begin{pmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{pmatrix} = \frac{1}{2} \left( S \overset{H+H^T}{\text{H+H}_2^T} + A \overset{H-H^T}{\text{H-H}_2^T} \right) \]
Infinitesimal structure of gradients

\[ \ell_1(x, y) = xy \quad \ell_2(x, y) = -xy \]

\[ \xi = \left( \frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x) \]

Generalized Helmholtz decomposition:

\[ H_\xi = \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}}_{S} + \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}}_{A} \]
Div, grad, and curl (again)

**Functions**
- Scalar-valued

**1-forms**
- Vector-valued
- "Curl"

**2-forms**
- Antisymmetric matrix-valued (Lie algebra of infinitesimal rotations)

**Diagrams**
- $d_0 \rightarrow \text{grad} \rightarrow 1\text{-forms} \rightarrow \text{exterior derivative} \rightarrow d \rightarrow \text{2-forms}
Div, grad, and curl (again)

functions  
scalar-valued \[ \xrightarrow{d_0} \] grad \[ \rightarrow \] 1-forms \nvector-valued \[ \xrightarrow{d} \] “curl” \[ \rightarrow \] 2-forms  
antisymmetric matrix-valued (Lie algebra of infinitesimal rotations)

\[ \xi = \left( \frac{\partial l_1}{\partial x}, \frac{\partial l_2}{\partial y} \right) = (y, -x) \]

\[ d_1 \xi = \left( \begin{array}{cc} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{array} \right) = A \]

2-form measures failure to be a gradient vector field
The generalized Helmholtz decomposition:

The game Hessian decomposes as \( H = S + A \)

- \( S \) is the "gradient component"
- \( A \) is the "curl component"
The big picture

Hamiltonian games

Potential games

$S \equiv 0$

$A \equiv 0$

$H = S + A$

general games
Symplectic Gradient Adjustment (SGA)

\[ \dot{\xi} + \lambda \cdot A^T \xi \]

- \( \lambda = \pm 1 \)
- computational cost is 2x backprop
Symplectic Gradient Adjustment (SGA)

\[ \dot{\xi} + \lambda \cdot A^T \xi \]

**Properties:**

- \( \xi \perp A^T \xi \): adjustment is compatible with original dynamics
  - Related: consensus optimization (Mescheder et al, NIPS 2017), \( \dot{\xi} + \lambda \cdot H^T \xi \) which is attracted to local maxima

- if potential game then SGA is gradient descent \( \rightarrow \) finds local min
- if Hamiltonian game then SGA finds local Nash equilibrium
- behaves correctly near stable and unstable equilibria
SGA allows higher learning rates

Learning rate

0.01

0.032

0.1

Gradient descent

SGA
Comparison with Optimistic Mirror Descent: 2-players
Comparison with Optimistic Mirror Descent: 4-players

![Graph showing time to converge vs learning rate](image-url)
Performance on synthetic GAN
Performance on synthetic GAN
Performance on synthetic GAN
Summary

● Deep (supervised) learning is gradient descent on a loss
  ○ Simple, effective, one-concept-fits-all
  ○ Compositionality comes for free

● We're starting to work with interacting losses
  ○ We don't really know when or how to compose losses
  ○ There's real thinking to be done
6. Multi-agent Learning at Scale
Multi-agent Reinforcement Learning (MARL)

Objective: find policy that maximizes local or joint value:

- Competitive
- Cooperative

\[ V^* = \max_{\pi} \mathbb{E} \left[ \sum_t \gamma^t R(s_t, a_t) | P(s_0), \pi \right] \]
MARL: Training and Execution
Independent Q-Learning Approaches

**Independent Q-learning [Tan, 1993]**

\[
Q(x, a) \leftarrow Q(x, a) + \beta(r + \gamma V(y) - Q(x, a))
\]

\[
V(x) = \max_{b \in \text{actions}} Q(x, b)
\]

**Independent Deep Q-Networks [Tampuu et al., 2015]**

![Diagram of Deep Q-Networks]

**Table 1: Average Number of Steps to Capture a Prey**

<table>
<thead>
<tr>
<th>N-of-prey/N-of-hunters</th>
<th>1/1</th>
<th>1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random hunters</td>
<td>123.08</td>
<td>56.47</td>
</tr>
<tr>
<td>Learning hunters</td>
<td>25.32</td>
<td>12.21</td>
</tr>
</tbody>
</table>

![Graph showing Evolution of Q-value]
Lenient Learning Approaches

- **Issue:** Non-stationarities $\rightarrow$ policy/Q-value degradation and destabilization
- **Idea:** learners should be lenient against/ignore Q-value degradation
  - See Lenient Deep Q-Networks (Palmer et al., 2018) and Hysteretic Q-Networks (Omidshafiei et al., 2017)

**Hysteretic Q-Networks:**

$$L(\theta_j^i) = (r_t^i + \gamma \max_{a'} Q(o_{t+1}^i, h_t^i, a'; \hat{\theta}_j^i) - Q(o_t^i, h_{t-1}^i, a_t^i; \theta_j^i))^2$$

Local TD error $\delta_t^i$

$$\theta_j \left\{ \begin{array}{ll} \theta_j - \alpha \nabla_{\theta_j} L(\theta_j^i) & \delta_t^i > 0 \text{ (underestimate)} \\ \theta_j - \beta \nabla_{\theta_j} L(\theta_j^i) & \delta_t^i \leq 0 \text{ (overestimate/degradation)} \end{array} \right.$$  

where $0 < \beta < \alpha$
Lenient Learning Approaches

● **Issue**: Non-stationarities $\rightarrow$ policy/Q-value degradation and destabilization

● **Idea**: learners should be lenient against/ignore Q-value degradation

- Converges to optimal in deterministic cooperative MDPs [Lauer et al., 2000]
Centralized Critic Decentralized Actor Approaches

- **Idea:** reduce nonstationarity & credit assignment issues using a central critic
- **Examples:** MADDPG [Lowe et al., 2017] & COMA [Foerster et al., 2017]
- Apply to both cooperative and competitive games

Centralized critic trained to minimize loss:
\[ \mathcal{L}(\theta_i) = \mathbb{E}_{x,a,r,x'}[(Q^\pi(x, a_1, \ldots, a_N) - y)^2], \]
\[ y = r_i + \gamma Q^\pi_i(x', a'_1, \ldots, a'_N)|_{a'_i = \pi'_i(o_j)} \]

Decentralized actors trained via policy gradient:
\[ \nabla_\theta_i J(\theta_i) = \mathbb{E}_{s \sim p^\mu, a_i \sim \pi_i} [\nabla_\theta_i \log \pi_i(a_i|o_i) Q^\pi_i(x, a_1, \ldots, a_N)] \]
Opponent-aware Models

- **Idea:** account for beliefs, models, and/or learning algorithms of other agents

**Interactive POMDPs [Gmytrasiewicz & Doshi, 2005]**

Maintain a belief over environment state and the other agents’ models (e.g., learning algorithms, observation functions, their beliefs over other agents, etc.)

**Extended Replicator Dynamics [Tuyls et al., 2003]**

In standard replicator dynamics (RD), player strategies evolve greedily w.r.t. current payoff:

\[
\frac{dx_i}{dt} = [(Ax)_i - x \cdot Ax] x_i
\]

In the extended RD, players take into account payoff growth in the future:

\[
f(x) = RD(x) + (dRD(x)/dt) * \eta
\]

**Learning with Opponent-Learning Awareness (LOLA) [Foerster et al., 2018]**

“Naive” learner policy gradient update for agent 1:

\[
\theta_{i+1} = \theta_i + f^{1}_{nl}(\theta_i^1, \theta_i^2),
\]

\[
f_{nl}^1 = \nabla_{\theta_i^1} V^1(\theta_i^1, \theta_i^2) \cdot \delta
\]

Taylor-expand agent 1’s value given agent 2’s update:

\[
V^1(\theta^1, \theta^2 + \Delta \theta^2) \approx V^1(\theta^1, \theta^2) + (\Delta \theta^2)^T \nabla_{\theta^2} V^1(\theta^1, \theta^2)
\]

Assuming agent 2 is a naive learner with update

\[
\Delta \theta^2 = \nabla_{\theta^2} V^2(\theta^1, \theta^2) \cdot \eta
\]

then we arrive at the LOLA update rule:

\[
f^{1}_{lola}(\theta^1, \theta^2) = \nabla_{\theta^1} V^1(\theta^1, \theta^2) \cdot \delta
\]

\[
+ \left( \nabla_{\theta^2} V^1(\theta^1, \theta^2) \right)^T \nabla_{\theta^1} \nabla_{\theta^2} V^2(\theta^1, \theta^2) \cdot \delta \eta
\]
Games and Reinforcement Learning

**Game theory**
- Solutions are *strategy profiles*
  specifying joint actions at all
  possible *information sets*

**Reinforcement learning**
- Solutions are *joint policies*
  specifying joint actions at all
  possible *partially observed states*
Neural Fictitious Self-Play [Heinrich & Silver 2016]

- **Idea**: Fictitious self-play (FSP) + deep reinforcement learning
- Approximate NE via two neural networks:

1. **Best response net (BR):**
   - Estimate a best response
   - Trained via RL

2. **Average policy net (AVG):**
   - Estimate the time-average policy
   - Trained via supervised learning
Neural Fictitious Self-Play [Heinrich & Silver 2016]

- Leduc Hold’em poker experiments:
  - 1st scalable end-to-end approach to learn approximate Nash equilibria w/o prior domain knowledge
    - Competitive with superhuman computer poker programs when it was released
Learning under Nonstationarity

**Policy Gradient** (Advantage Actor-Critic)

\[ \nabla_\theta J(\theta) = \mathbb{E}_\pi [\nabla_\theta \log \pi(a_t \mid s_t; \theta) A(s_t, a_t; w, \theta)] \]

**Replicator Dynamics**

\[ \hat{\pi}(a) = \pi(a)A(a) \]

Logit space \( \pi = \text{softmax}(y) \) stateless tabular case

\[ y_t(a) = y_{t-1}(a) + \eta \pi(a)A(a) \]

- Nash
- PG policy
- RD policy

![Diagram showing Nash, PG policy, and RD policy trajectories]
Neural Replicator Dynamics (NeuRD)

- Policy Gradient handles high-dimensional state- and action-spaces seamlessly
  - Replicator Dynamics are limited to tabular settings
- Replicator Dynamics are no-regret (time-average convergence to Nash)
  - Policy Gradient has no such guarantees

Neural Replicator Dynamics: best of both worlds!
Neural Replicator Dynamics (NeuRD)

\[ \theta_t = \theta_{t+1} + \eta \sum_{s,a} \nabla_{\theta} y_{t-1}(s_t, a_t; \theta) A(s_t, a_t; \theta, w) \]

Logits, where policy is
\[ \pi = \text{softmax}(y) \]

Advantage \[ q(s,a) - v(s) \]
Results

Biased Rock-Paper-Scissors

Leduc Poker

DeepMind
# A MARL Retrospective

<table>
<thead>
<tr>
<th>Foundational Algorithm</th>
<th>Modern and/or Deep RL Counterpart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fictitious Play [Brown, 1951]</td>
<td>Extensive-form Fictitious Play [Heinrich et al., 2015]</td>
</tr>
<tr>
<td>Independent Q-learning [Tan, 1993]</td>
<td>Neural Fictitious Self-Play [Heinrich &amp; Silver, 2016]</td>
</tr>
<tr>
<td>Double Oracle [McMahan et al., 2003]</td>
<td>Multi-agent Deep Q-Networks [Tampuu et al., 2015]</td>
</tr>
<tr>
<td>Hysteretic Q-learning [Matignon et al., 2007]</td>
<td>Policy-Space Response Oracles [Lanctot et al., 2017]</td>
</tr>
<tr>
<td>Extended Replicator Dynamics [Tuyls et al., 2003]</td>
<td>Recurrent Hysteretic Q-Networks [Omidshafiei et al., 2017]</td>
</tr>
<tr>
<td>Lenient Learning [Panait et al., 2006; Panait, Tuyls, Luke, 2008]</td>
<td>Learning with Opponent-Learning Awareness [Foerster et al., 2017]</td>
</tr>
<tr>
<td></td>
<td>Neural Replicator Dynamics [Omidshafiei et al., 2019]</td>
</tr>
</tbody>
</table>

Non-exhaustive list! For more, check out:

“Deep Reinforcement Learning for Multi-Agent Systems: A Review of Challenges, Solutions and Applications” (Nguyen et al., 2019)

“Is multiagent deep reinforcement learning the answer or the question? A brief survey” (Hernandez-Leal et al., 2018)


“Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems.” (Matignon et al., 2008)
References


7. Why are Games Important? 
Wrap-up
Games as a Multi-Agent Platform

How Life Imitates Chess G. Kasparov

“Unfortunately, the number of ways to do something wrong always exceeds the number of ways to do it right”

“A CEO must combine analysis and research with creative thinking to lead his company effectively”
Games for AI

Good controlled model for Multi-Agent Learning

- Simple rules, deep concepts
- Studied for hundreds or thousands of years
- Co-evolution artifact -> Learning
- ‘Drosophila’ of artificial intelligence
- Microcosmos encapsulating real world issues
- Games are fun!
Games for AI - A theory of Games

- Concept from traditional Game Theory
- Hyper-rational players
- Static concept

Intuitively: A Nash Equilibrium is a strategy profile for a game, such that no player can increase its payoff by unilaterally changing its strategy.

- Players are not hyper rational, but also biologically and socially conditioned
Zero-Sum Games for AI

- Why are zero-sum games of interest?
  - Many standard AI benchmark domains are inherently zero-sum
  - Strong theoretical guarantees for zero-sum games
  - Strict relations over outcomes → strategize by maximizing wins/rewards
  - Existence of standard algorithm evaluation methods
AlphaGo Zero

Mastering Go without Human Knowledge
AlphaZero: One Algorithm, Three Games

Chess  Shogi  Go
Video Games

Started with **toy MDPs**.

**Grid worlds** starting to feel like games.

**Atari** - very engaging for humans.

*Mnih et al, 2018.*
Video Games

Started with toy MDPs.

Grid worlds starting to feel like games.

Atari - very engaging for humans.

3D single-player - even richer potential task space. (DeepMind Lab, VizDoom, Minecraft)

A3C Vmnh et al 2016,
Video Games: **Multi-agent**

Much richer task space with simple rules: competitive and cooperative

Diversity of solution: robustness

Auto-curricula

Non-stationary: continual learning

*Video Games: Multi-agent*  
Dorer vs Stone, 2017.
The Importance of Games

- Development of **general applicable** techniques in
  - Controlled **environments**
  - Fast **simulations**
  - Principled **evaluation** and **understanding**
  - Drives the **AI Frontiers**

- Can be deployed in various **other domains**
  - Fraud detection systems
  - Auction agents
  - Energy systems (smart grid)
  - Industry 4.0 systems