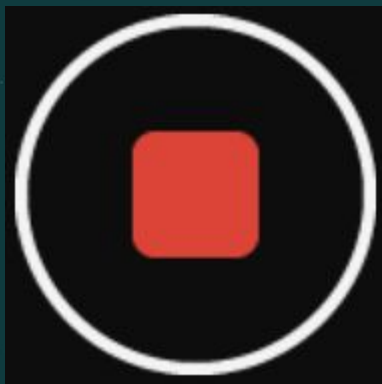


Lecture 21: Learning in MA
systems



Start Recording!

Reminders

- Office Hours tomorrow with Adrien (11-12AM)
- Last Talks this Friday.
- Next two lectures will be on empirical game theory, self-play and other interesting things.

Talk on StarCraft II by Wojciech M. Czarnecki

On Friday 16th **at noon**

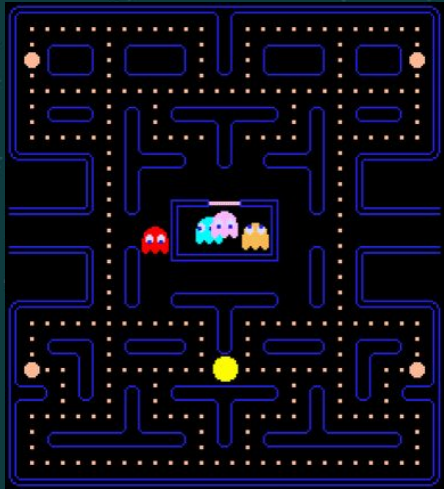
References for this lecture:

1. Balduzzi, David, et al. "Open-ended learning in symmetric zero-sum games." *International Conference on Machine Learning*. PMLR, 2019.

Today: Empirical Games

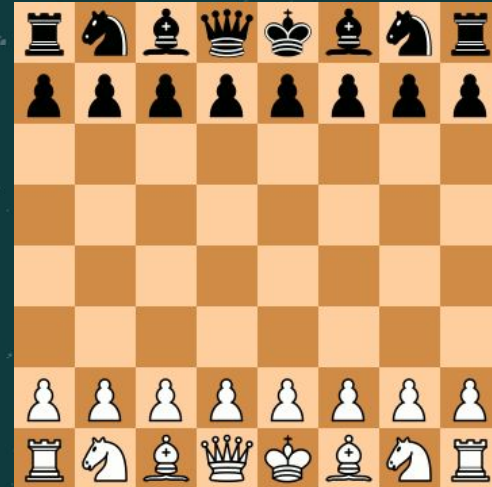
Motivations : Learning Objectives

Single player:



Hand-crafted notion of performance

Multi-player:



Very simple notion of performance
The complexity of the task depends on the **opponent(s)**

Achieving super-human performance in multi-player games is very challenging and requires a deep understanding of the game.

Go



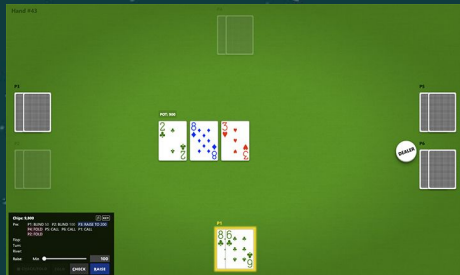
[Silver et al. 2016]
(Picture from DeepMind's blog post)

Dota 2



[OpenAI et al. 2019]
(Picture from OpenAI's Blog post)

Poker



[Brown and Sandholm 2019]
(Picture from FAIR's Blog post)

Starcraft II



[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

AntiSymmetric (zero-sum) Game (Functional Form)

Anti-symmetric Payoff:

$$\varphi : \mathcal{W} \times \mathcal{W} \rightarrow \mathbb{R}$$

Players (example: RL policies)

$$\varphi(u, w) = -\varphi(w, u)$$


Intuition: Switching the roles switches the results.

Example: Chess, Go, Poker (need to randomize who starts)

NB: Can generalize to non-zero sum (just heavier because of the two losses)

Who are the players?

Here we care about the agent/player (same thing)


$$\varphi(w, u)$$

Usually: Parametrized policy u_{θ}

But can be anything that plays the game: (e.g., chess engine, human player)


Interpretation of the payoff

$$\varphi(u, w) > 0 \quad u \text{ beats } w.$$

$$\varphi(u, w) < 0 \quad w \text{ beats } u.$$

$$\varphi(u, w) = 0 \quad \text{it is a tie.}$$

Proba of winning



Example: $\varphi(u, w) = \mathbb{P}(u \succ w) - \frac{1}{2}$

$$\varphi(u, w) = \log \mathbb{P}(u \succ w) - \log \mathbb{P}(w \succ u)$$

Transitive game:

$$\varphi(u, v) > 0, \varphi(v, w) > 0 \Rightarrow \varphi(u, w) > 0$$

Example: $\varphi(u, w) = \sigma(f(u) - f(w))$

Question: Is it the only possible transitive payoff???

Answer: I don't know... Research project!

Example: Elo Rating

$$\mathbb{P}(u \succ w) = \frac{1}{1 + \exp(\alpha \cdot (f(w) - f(u)))}$$

$f(u)$: Elo Rating of u

Problem: $f(u) \gg f(w) \Rightarrow \nabla_u \varphi(u, w) \approx 0$

Intuition: Playing against weaker opponent gives almost no training signal.

Example: Elo Rating

$$\mathbb{P}(u \succ w) = \frac{1}{1 + \exp(\alpha \cdot (f(w) - f(u)))}$$

Solution: Self-play i.e., play against **a copy** of yourself.

Problem: $f(u) \gg f(w) \Rightarrow \nabla_u \varphi(u, w) \approx 0$

Intuition: Playing against weaker opponent gives almost no training signal.

Example: Elo Rating

$$\mathbb{P}(u \succ w) = \frac{1}{1 + \exp(\alpha \cdot (f(w) - f(u)))}$$

Sol Question (Mohamed): When can we approximate Elo Rating?

Prob Answer: See next lecture and <https://arxiv.org/pdf/1806.02643.pdf>

Intuition: Playing against weaker opponent gives almost no training signal.

Open-ended Learning

General Framework to answer the question:

“Who plays against who?”

updated agent \leftarrow **oracle(agent, opp, payoff)**

Stronger agent against the opponent.

Example: Self-play:

$u_{t+1} \leftarrow \text{oracle}(u_t, u_t, \varphi)$

Open-ended Learning

General Framework to answer the question:

“Who

Question (Safwen):
Can we generalize Self-play to more than two players.

upd. Answer: Yes!!!

See: <https://openai.com/blog/emergent-tool-use/>

payoff)

Stronger agent against the opponent.

Example: Self-play:

$$u_{t+1} \leftarrow \text{oracle}(u_t, u_t, \varphi)$$

updated agent \leftarrow **oracle**(agent, opp, payoff)

such that $\varphi(\text{updated agent, opponent}) > \varphi(\text{agent, opponent})$

Examples:

- Gradient ascent method:

$$\mathbf{oracle}(u_t, v_t, \varphi) = u_t + \eta \nabla_u \varphi(u_t, v_t)$$

- RL algorithms (gradient based or not). For instance Q-learning:

$$\underbrace{\mathbf{oracle}(Q_u, Q_v, \varphi)(s, a)}_{\text{New Q function of u}} = \underbrace{Q_u(s, a)}_{\text{Old Q function of u}} + \eta \underbrace{(r_v)}_{\text{Reward against agent v}} + \underbrace{\gamma \max_a Q_u(s^+, a)}_{\text{Discounted estimate of the value at the next state}} - Q_u(s, a)$$

New Q function of u

Old Q function of u

Discounted estimate of the value at the next state

Reward against agent v

- Evolutionary algorithms

updated agent \leftarrow oracle(agent, opp, payoff)

such that $v(\text{updated agent opponent}) > v(\text{agent opponent})$

Exam

Question (Zicong Mo): It seems that the key here is to have a differentiable phi. I am not clear about the definition of the phi.

How can you represent a process of game interaction in one single function phi?

Does it only represent the end result of the game?

ing:

$$\text{oracle}(Q_u, Q_v, \varphi)(s, a) = Q_u(s, a) + \eta(r_v) + \gamma \max_a Q_u(s^+, a) - Q_u(s, a)$$

New Q function of u

Old Q function of u

Discounted estimate of the value at the next state

Reward against agent v

- Evolutionary algorithms

Open-ended Learning

General Framework to answer the question:

“Who plays against who?”

`updated agent ← oracle(agent, opp, payoff)`

Conclusion:

- General framework to understand general algorithm such as self-play or Fictitious self play.

Self-Play

$$u_{t+1} \leftarrow \text{oracle}(u_t, u_t, \varphi)$$

- Play against a copy of yourself
- Well calibrated opponent
- Simple algorithm.
- Successful in Chess, Go and many other applications
- Issue: Assume that the payoff is transitive:

$$\varphi(v_{t+1}, v_t) > 0, \dots, \varphi(v_1, v_0) > 0 \Rightarrow \varphi(v_{t+1}, v_i), i \in [t]$$

- Improvement against v_t implies global improvement.

A simple Example: The bilinear game

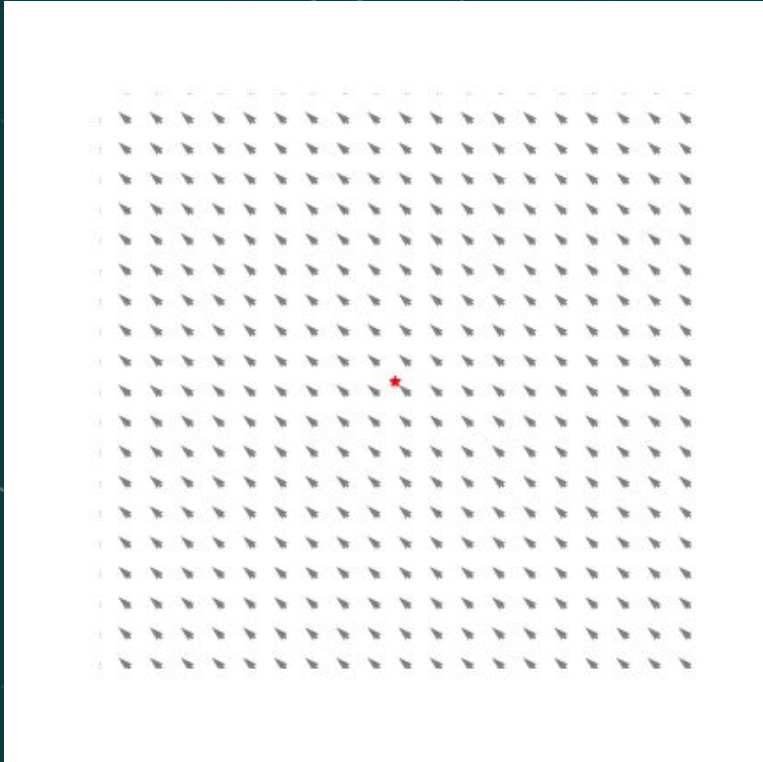
Simple payoff: $\varphi(u, w) = u_1 w_2 - w_1 u_2$

Self play: $u_{t+1} = u_t + \eta \nabla_u \varphi(u_t, \tilde{u}_t)$

Remark: in practice $\theta_{t+1} = \theta_t + \eta \nabla_u \frac{d\varphi(u_{\theta_t}, u_{\tilde{\theta}_t})}{dt}$

Copy of θ_t (do **not** differentiate through it)

A simple Example: The bilinear game



$$u_{t+1} = u_t + \eta \nabla_u \varphi(u_t, \boxed{u_t})$$

The vector field depends on your opponent.

Proposition: The dynamics of self play diverges

A simple Example: The bilinear game

Question (François David):

If I understand it right, the more transitive the payoff matrix is, the less we have to select a group of agents since playing against the best agents is going to result in a good generalization.

On the other hand, the more the payoff matrix is purely random or cyclic, the more we need to evaluate with a larger group of agents for a better generalization when training?

$$\varphi(u_t, u_t)$$

n your opponent.

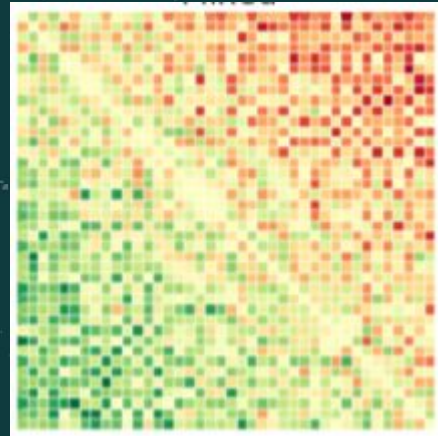
If play diverges

Idea: Playing against a group of agents

Population of agents $\mathcal{B} = (u_i)$

Payoff matrix of the group: $A_{\mathcal{B}}$

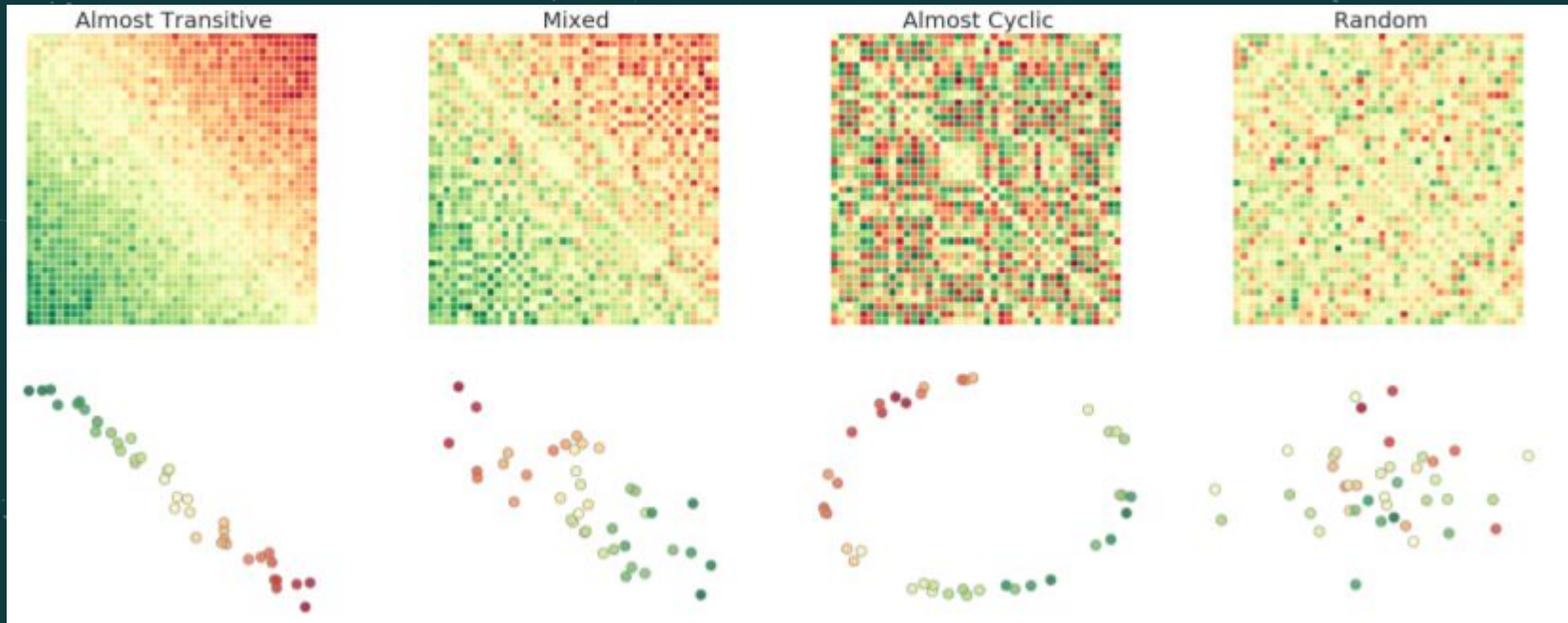
$$[A_{\mathcal{B}}]_{ij} = \varphi(u_i, u_j)$$



Idea: Playing against a group of agents

Population of agents \mathcal{B}

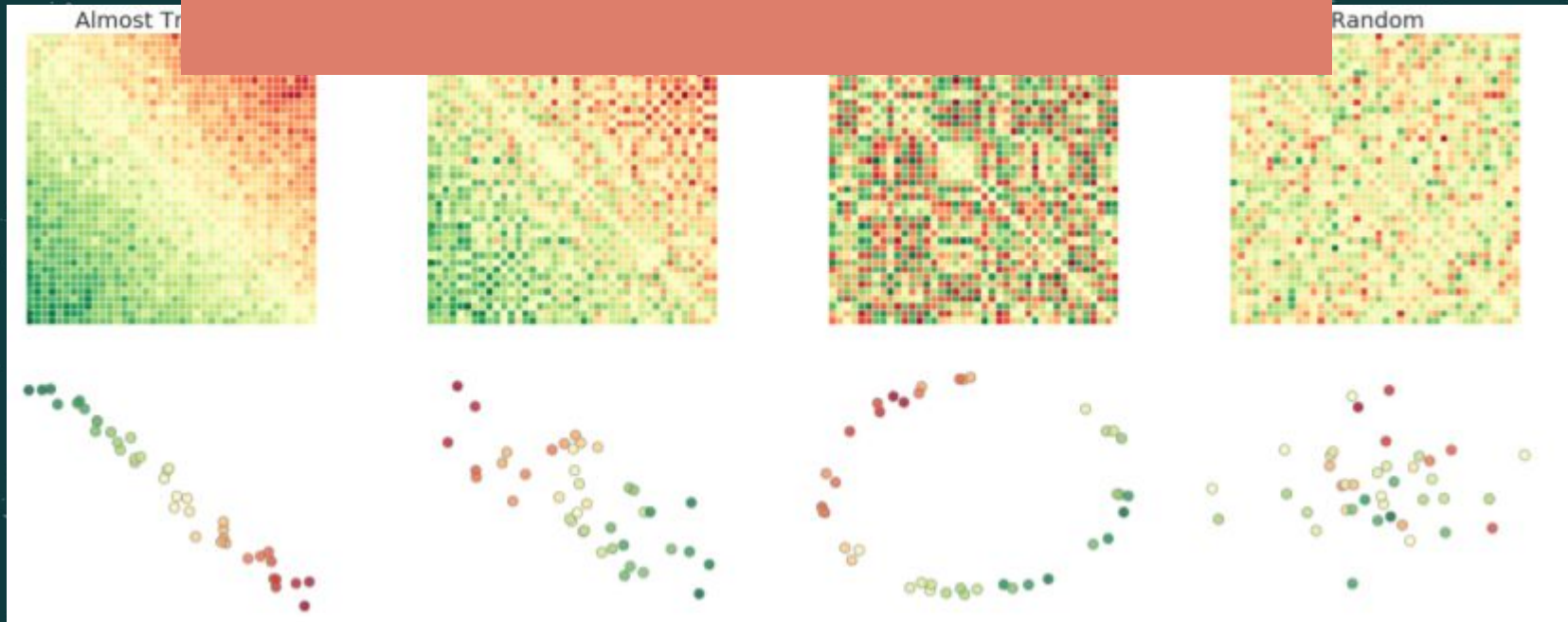
Payoff matrix of the group: $A_{\mathcal{B}}$



Idea: Playing against a group of agents

Popula
Payoff

Question (Olivier): I am a little bit puzzled by this representation, could you explain it in further?



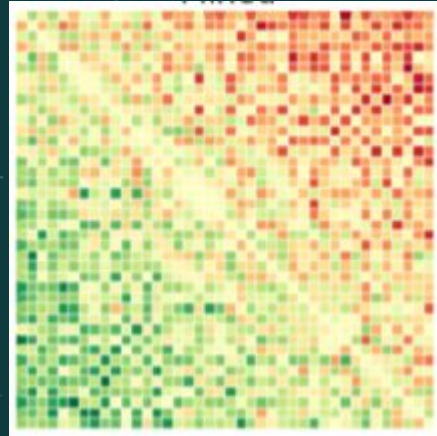
Nash of an Empirical Game

Proposition:

$$Nash = \{p : p^\top A \geq 0, p \geq 0\}$$

Mixture of Agents:

Sample v_i with probability p_i .



Nash of an Empirical Game

Proposition

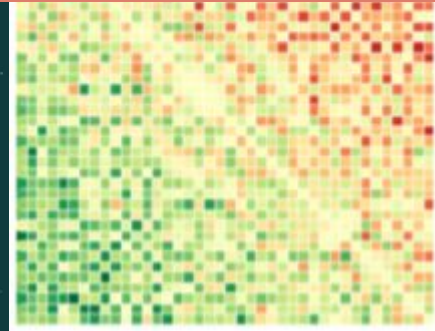
N

Mixture

Sample v_i with probability p_i .

Question (Hattie Zhou):

- 1) How do we find the nash in a population of agents?
- 2) Would we only compare the nash to other agents in the population, or compare the nash to all possible mixtures of agents in the population?



Matrix of the empirical game

We can use this matrix for several purposes:

1. Evaluate (a group of) agents.
2. Evaluate the diversity of a group of agents
3. Setup efficient Training.

Many open questions remaining:

- How to relate the empirical matrix to the real game?
- Are the proposed measures (see next slides) meaningful?

How to evaluate the Performance of a Populations?

Definition 3. Given populations \mathfrak{P} and \mathfrak{Q} , let (\mathbf{p}, \mathbf{q}) be a Nash equilibrium of the zero-sum game on $\mathbf{A}_{\mathfrak{P}, \mathfrak{Q}} := \phi(\mathbf{v}, \mathbf{w})_{\mathbf{v} \in \mathfrak{P}, \mathbf{w} \in \mathfrak{Q}}$. The **relative population performance** is

$$v(\mathfrak{P}, \mathfrak{Q}) := \mathbf{p}^\top \cdot \mathbf{A}_{\mathfrak{P}, \mathfrak{Q}} \cdot \mathbf{q} = \sum_{i,j=1}^{n_1, n_2} A_{ij} \cdot p_i q_j.$$

Proposition 5. (i) Performance v is independent of the choice of Nash equilibrium. (ii) If ϕ is monotonic then performance compares the best agents in each population

$$v(\mathfrak{P}, \mathfrak{Q}) = \max_{\mathbf{v} \in \mathfrak{P}} f(\mathbf{v}) - \max_{\mathbf{w} \in \mathfrak{Q}} f(\mathbf{w}).$$

(iii) If $\text{hull}(\mathfrak{P}) \subset \text{hull}(\mathfrak{Q})$ then $v(\mathfrak{P}, \mathfrak{Q}) \leq 0$ and $v(\mathfrak{P}, \mathfrak{R}) \leq v(\mathfrak{Q}, \mathfrak{R})$ for **any** population \mathfrak{R} .

Proposition: For any population

- $v(B, B) = 0$

How to Evaluate Diversity of a Population?

Definition 4. Denote the rectifier by $[x]_+ := x$ if $x \geq 0$ and $[x]_+ := 0$ otherwise. Given population \mathfrak{P} , let \mathbf{p} be a Nash equilibrium on $\mathbf{A}_{\mathfrak{P}}$. The **effective diversity** of the population is:

$$d(\mathfrak{P}) := \mathbf{p}^\top \cdot [\mathbf{A}_{\mathfrak{P}}]_+ \cdot \mathbf{p} = \sum_{i,j=1}^n [\phi(\mathbf{w}_i, \mathbf{w}_j)]_+ \cdot p_i p_j.$$

Interpretation: How much the **best** agents (i.e. agents in the Nash) exploit each other.

How to train agents Efficiently?

Use this matrix to find who to train against:

- Train against the Nash
- Train against the best Response.
- Many other ways:
 - [Garnelo et al. 2021](to appear at AAMAS)
 - see also SC II paper (league of agents).

Idea:

- Compute the Nash and Play against it: (PSRO)

Algorithm 3 Response to Nash (PSRO_N)

input: population \mathfrak{P}_1 of agents

for $t = 1, \dots, T$ **do**

$\mathbf{p}_t \leftarrow$ Nash on $\mathbf{A}_{\mathfrak{P}_t}$

$\mathbf{v}_{t+1} \leftarrow$ oracle $(\mathbf{v}_t, \sum_{\mathbf{w}_i \in \mathfrak{P}_t} \mathbf{p}_t[i] \cdot \phi_{\mathbf{w}_i}(\bullet))$

$\mathfrak{P}_{t+1} \leftarrow \mathfrak{P}_t \cup \{\mathbf{v}_{t+1}\}$

end for

output: \mathfrak{P}_{T+1}

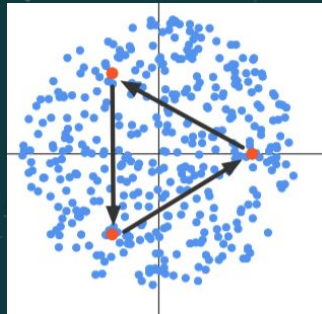
- Seems like a good idea.
- Problem: Sometime provide zero gradient (e.g. Bilinear example)

Fictitious Self-Play

- Group of agents v_i
- Play against to 'best' opponent

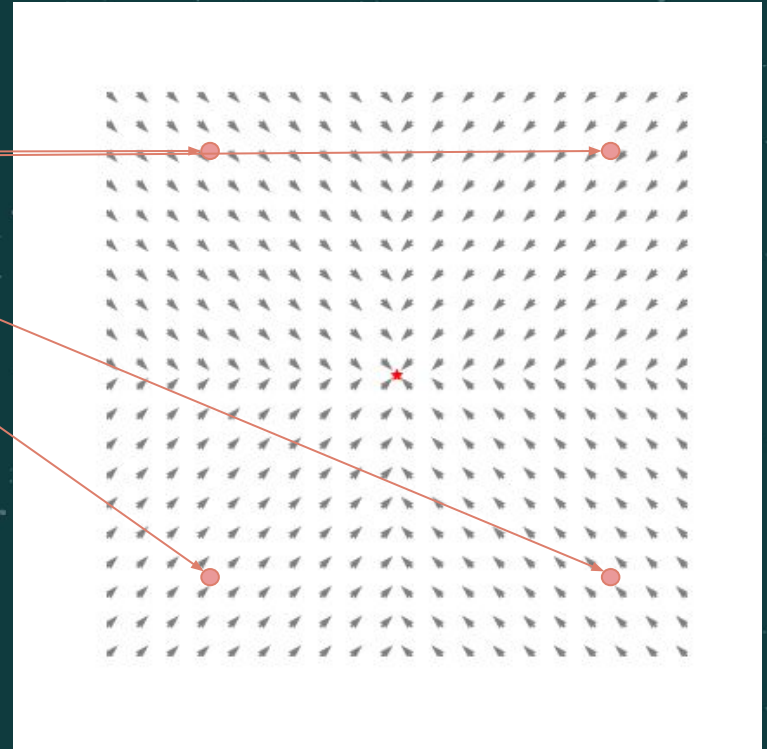
$$u_{t+1} \leftarrow \text{oracle}(u_t, \text{best opp}, \varphi)$$

$$\text{best opp} := \arg \min_{v_i} \varphi(u, v_i)$$



Fictitious Self-Play

- Group of agents v_i
- Play against to 'best' opponent
- Used in Starcraft II [Vinyals · 2019]



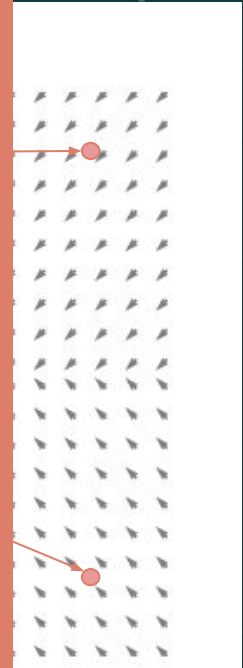
Fictitious Self-Play

- Group of
- Play again
- Used in S

Question (Simon): I'd like to better understand how "groups of players" are used ultimately. Is the end-goal to come up with a single player that does the best against unknown opponents?

I'm thinking of a completely circular game like rock-paper-scissors: even if you find the optimal strategy against each opponent you could face, it still doesn't provide any useful information about what you should do against an unknown opponent.

Is this a special case because it is fully circular?



Conclusion

- Self-play is a very powerful method to train agents in a Multi-Agent framework.
- Sometimes it fails (when we need a diversity of agents to play the game)
- When having a group of agents we can use the empirical payoff to:
 - Evaluate agents
 - Train Agents
 - Evaluate the group (perfs and diversity)